UNIVERSITY OF
CAMBRIDGE

Interpretable Deep Learning:

Beyond Feature-Importance

with

Concept-based Explanations

Botty Todorov Dimanov

Girton College

This dissertation is submitted on December 30, 2020 for the degree of Doctor of Philosophy

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. It is not substantially the same as any that I have submitted, or am concurrently submitting, for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or is being concurrently submitted, for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. This dissertation does not exceed the prescribed limit of 60 000 words.

Botty Todorov Dimanov
May 25, 2021

Abstract

Interpretable Deep Learning

Botty Todorov Dimanov

Deep Neural Network (DNN) models are challenging to interpret because of their highly complex and non-linear nature. This lack of interpretability (1) inhibits adoption within safety critical applications, (2) makes it challenging to debug existing models, and (3) prevents us from extracting valuable knowledge. Explainable AI (XAI) research aims to increase the transparency of DNN model behaviour to improve interpretability. Feature importance explanations are the most popular interpretability approaches. They show the importance of each input feature (e.g., pixel, patch, word vector) to the model's prediction. However, we hypothesise that feature importance explanations have two main shortcomings concerning their inability to describe the complexity of a DNN behaviour with sufficient (1) fidelity and (2) richness. Fidelity and richness are essential because different tasks, users, and data types require specific levels of trust and understanding.

The goal of this thesis is to showcase the shortcomings of feature importance explanations and to develop explanation techniques that describe the DNN behaviour with greater richness. We design an adversarial explanation attack to highlight the infidelity and inadequacy of feature importance explanations. Our attack modifies the parameters of a pre-trained model. It uses fairness as a proxy measure for the fidelity of an explanation method to demonstrate that *the apparent importance* of a feature does not reveal anything reliable about the fairness of a model. Hence, regulators or auditors should not rely on feature importance explanations to measure or enforce standards of fairness.

As one solution, we formulate five different levels of the semantic richness of explanations to evaluate explanations and propose two function decomposition frameworks (DGINN and CME) to extract explanations from DNNs at a semantically higher level than feature importance explanations. Concept-based approaches provide explanations in terms of atomic human-understandable units (e.g., wheel or door) rather than individual raw features (e.g., pixels or characters). Our function decomposition frameworks can extract specific class representations from 5% of the network parameters and concept representations with an average-per-concept F1 score of 86%. Finally, the CME framework makes it possible to compare concept-based explanations, contributing to the scientific rigour of evaluating interpretability methods.

Acknowledgements

My favourite part of completing any piece of work is experiencing the gratitude that flows when writing the acknowledgements section! Given that this is an almost half-a-decade project, I wanted to make this section the best I have written so far! I am fortunate to have many people to thank. However, before we go onto them, first and foremost I have something else to appreciate – all the problems, challenges, hurdles, insurmountable obstacles and other wonderful emotions, such as anger, frustration, rejection, disappointment, and discouragement that the Stoic universe managed to throw my way. I am so grateful for all of these 'little teasers" along the way because they made this project the one from which I have grown the most and learnt immeasurably about myself! I was hoping I would come out in the end and say: "well it was easy", maybe, fortunately, it was not easy at all!

By a happy chance, I was blessed with countless lifesavers, who are among the most outstanding and generous people in the universe, including, but not limited to sensational supervisors, phenomenal collaborators, unwavering bedrock supporters, unconditionally loving sweetheart, family, and friends!

To begin with, no word of appreciation could possibly measure up to the support of my supervisors **Mateja Jamnik** and **Adrian Weller**! My immense gratitude for their unwavering support, profound wisdom, devotion to serving!

The person deserving most credit for completing this thesis and forging my character is without doubt **Mateja**. Her generosity to give me the opportunity to undertake a Ph.D. in the first place and to serve as my unwavering guiding beacon every single week for the past four years make me feel immensely grateful! Very few students would be fortunate to have a committed and dedicated supervisor like Mateja, who would without hesitation devote her time and efforts in paying attention to the slightest details of any of our joint works and have special dedicated time for our conversation every week for the duration of 4 long years. She helped me develop habits that will serve me for the rest of my life, such as punctuality, the emphasis on doing my best at everything I do, and upholding absolute excellence in every aspect of my work. Her teachings helped me on top of all else to learn to listen, trust, have faith, and be an outstanding team player, while "obey[ing] your leaders and submit[ing] to them, for they are keeping watch over your souls, as those who will have to give an account. *Let them* do this *with joy and not with groaning*, for that would be of no advantage to you" (Hebrews 13:17).

Another person, who has made an immeasurable contribution to my success, has been without doubt **Adrian**. Not surprisingly, finding a Jedi master can unquestionably lead to breakthroughs! The most rewarding time of my career as a researcher has been the possibility to work with and experience Adrian's scientific shrewdness, profound knowledge, exceptional astuteness, and passion for making a significant contribution to the scientific community! I hope that Adrian accepts my vehement protestations of gratitude for recognising the virtue of zest in me, taking me under his wing, and giving me the thought love that helped me get my greatest scientific achievement to date! From him I learned to question and be cautious, I learned that the quality of your questions, determines the quality of your answers and that it takes much more wisdom to ask the right questions than to answer them. Moreover, I learned that outstanding performance requires that you expect from yourself more than anybody else could expect from you!

Yet another person deserving equal credit is **Pietro Lio**, thanks to whom I had the blessing to experience Cambridge, and thanks to whose wisdom of visualising failure and using the time machine to go back in time to fix the past, I graduated once! From Pietro, I learned to keep my spirit high no matter what happens, and that in life it is not necessary to do everything yourself to achieve your result!

This, maybe most valuable, lesson I learned during my Ph.D., was thankfully reiterated from my dear friends **Ahmed Zaidi** and **Youmna Farag**, who helped me see that the Ph.D. has nothing to do with working alone! In fact, it is the privilege of working with exceptional people that makes the entire endeavour worth it! They helped me realise that above all else I craved for meaningful work and meaningful relationships with exceptional people! One such person is my first co-author, **Umang Bhatt**, whose excitement, support, time, and dedication in completing the adversarial explanation project cannot be appreciated enough.

However, another person upon whose appearance in my life hinges the entirety of my Ph.D. (and its worthiness), has been my collaborator, business partner, friend, my yin-yang – **Dmitry Kazhdan**, who created the polarity of our mastermind group and made the going ten times faster, better and most importantly unprecedentedly fun!

Yet another examples of the power of teamwork are **Zohreh Shams**, **Daniel Dimanov**, and **Dmitry Kazhdan** who deserve special gratitude for proofreading this manuscript and providing invaluable suggestions that made this thesis a reality!

Additionally, there are many people whose support means more to me than they can ever imagine, and I am humbled and appreciative of all they have done to balance my life between the science of achievement and the art of fulfilment. Let me now mention some of these astonishing individuals: Los señores **Dedalo** and **Bruno** for teaching me that there is more to life than work and for the invaluable lesson that your *surroundings* determine your mood (which turns out to be an entire philosophy "The Life-Changing Magic of Tidying Up"); **Maksim Dodonov** for the business acumen, determination and the iconoclastic execution that reminded me that energy comes when one enjoys the competitive nature of life because it feels amazing to be the winner; the Philosophers from the Hugh Akston's club: **Mr Jack Reidan** and **Dr Roeland Decorte-Galt** for pushing the boundaries of my thinking and my mind beyond the imaginable, and inviting me into the world of Atlas Shrugged; **Jack Reid** and **Obi-wan Kenobi** for their ability to tell and marvel at stories that make any conversation with them simply remarkable; **Alexander Dimitrov** for being a great friend, helping me rekindle the spark to complete this thesis, and for introducing me to three of the most exceptional friends and sources of laughter I am blessed to have – **Rasho and Stefcho Ikonomovi**, and **Philip-Totue**; the support, constant encouragement and cheerleading for "Doctor Botty" of another one of my dearest friends, **Norbert Naskov**, brings tears of gratitude to my face; laughter is the antipode of fear, and there are two friends who are supreme masters of this craft, and have brought many smiles to my face and cherished moments to my heart – thank you Vasko and Deian; my office-mate, **Aaron Stockdill**, who deserves a special round of applause for inspiring me to action, every time I had the pleasure of spending time with him; the astonishing people in the Computer Lab, who helped me learn immeasurably and made going to the lab an unforgettable experience: **Simeon Spasov**, **Tiago Azevedo**, **Giovanna Maria Dimitri**, **Duo Wang**, **Andrei Margeloiu**, **Ben Day**, **Pietro Barbiero**, **Helena Andres Terre**, **Agnieszka Slowik**, **Daniel Raggi**, **Luca Zanetti**, **Chaitanya Mangla**, **Felix Opolka**, **Jin Zhu**, **Urška Matjašec**, **Dobrik Georgiev**, **Hayk Saribekyan**, **Emma Rocheteau**, **Chris Bodnar**, **Cătălina Cangea**, **Paul Scherer**; **Sebastian Bianchi** for rekindling my passion for learning and making Spanish irresistibly enticing; I am also grateful to the students I had a chance to supervise – **Harry Graham** and **Sumaiyah Kola**, from whom I learned more about people and life than everything they probably learned from me; finally a special thank you goes to **Dhruv Ghulati** for sharing his Good Readings Reading Challenge that lead me to seek new voices of wisdom, without which the completion of this almost half-a-decade project might have been put in jeopardy!

Some of these new voices of wisdom, **Napoleon Hill**, **Jim Rohn**, **Tony Robbins**, and **Brian Johnson** have my eternal gratitude for opening my eyes to the importance of the voices you let inside and for their inspiration in the darkest moments! I am indebted to Tony Robbins and **Joseph McClendon III** for finally taking me out of the matrix and into the realm of state-control; and for teaching me the importance of having fun in everything you do to be *happily achieving.* And of course, I am thankful to **Alexander the Great** for having the audacity to cut the Gordian knot and inspiring me to do the same! I am obliged to my coaches, **Nano Gazze - The Ph.D.-Maker** for challenging me to develop personal accountability and always focus on the most important goal, **Brian**

Johnson and **Michael Balchan** for nourishing my energy and prompting me to read the best machine learning book out there – Deep Learning by **Ian Goodfellow, Yoshua Bengio and Aaron Courville**, **Scott Walker** for showing me the RPM system that helped me focus on the outcomes, and **Lanny Basham** for his wisdom on cultivating your self-image!

The **NVC Gang** has been a special place, where **Lisie**, **Gregorio Armani**, **Peter Pan**, and **King Arthur** bestowed upon me the gifts of developing the emotional intelligence to build lasting relationships and the ability to council with my inner board of directors. Mostly I would like to share my heartfelt appreciation for my first mentor, **Elizabeth**, who gave me the key to my secret inner world, and **Charlie** for giving me the map to navigate it!

Alice and Mat deserve a note of appreciation for helping me to discover who I truly want to be, and where my people are! I will be forever grateful to **Greg Law, Jason du Preez, and Martin Carroll**, who supported me to pursue entrepreneurship and who first planted in my mind the idea to turn my Ph.D. into a practical innovation, and then "coaxed, nursed, and enticed" it to remain alive. "Gradually, [this] idea [has] became a giant under its own power", and it coaxes, nurses, and drives me!

A special thank you to **James Thomas** who was the first one to entrust us his with fortune[1], and **Amy Weatherup**, the Impulse team, of **Yupar** and **Alexandra**, and the CDL team, including **Kristjan**, **Bogdan**, **Adam**, and **Ezra**, as well as our mentors, **William TP, Ash Fontana, Steve Mann, Michael Lyon, and Andra Keay**, who gave life to the inception of Tenyks.

Two people among all others created the spark and the powerful motivation to complete this thesis! I will long remember and cherish the help of **Hanadi Jabado** and **Shaw Chuang**, who helped me to visualise vividly the power of my decisions and actions!

I am deeply grateful to the UVH team, who challenged me beyond everything I could have ever imagined. A special thank you goes to **Plamen**, **Ines**, **Viktor**, and **Marto** for jumping from a rock with me and building an aeroplane on the go! I am grateful to Plamen not only for joining me on this adventure, but also for being one of my longest-standing swim-buddies in all sorts of precarious exploits! The magnificence of the Vensy gentlemen, **Tunde** and **Rodney**, cannot go unmentioned, as they made one of my dreams for this Ph.D. come true!

By good luck, the Graduate Admissions team not only saw something in me and gave me a chance to join the Cambridge ranks, but **Lisa**, **Marketa**, and **Joy** have done so much on my behalf in helping me to complete and enjoy both my MPhil and my Ph.D. that it is difficult to express the value I attribute to their generosity. In addition, their invitation to the graduate student forum has been a blessing as it set me on the path of

[1] "from the one who has been entrusted with much, much more will be asked" Luke 12:48 NIV

generosity and service. This path bestowed upon me the pleasure of meeting the fabulous members of the Mindfulness Society – **Josephina**, **Ana**, and **Carin**. This experience catapulted me into one of the greatest adventures of my Ph.D. journey, namely leading the Cambridge University Entrepreneurs (CUE)! I cannot express enough the gratitude I have for the CUE team who did so much work in the background so that I could have the time to work on my Ph.D. Special thanks go to **Nina Warner** - my saviour angel, **Dragomir** - my right hand, **Florian** for giving me the pep-talk about focus and execution of well-proven formulae, Stewart McTavish for inspiring me to join CUE in the first place, **Max Ge** for his wisdom, guidance, and attention to detail, **Sehaj** for our spiritual journey to meet **Sukhi**, **Shahang**, and **Akhil**.

And now is time to say thank you to the most special out of all these people because ”any scientist would be lucky to have one person in the world who loves him and also genuinely understands, appreciates and contributes to his work;” and I am fortunate to have at least four such people.

My **parents Daniela and Todor** have devoted everything to make the idea of walking on the Cambridge grass a reality for their son. Their dedication can be recognised in the supreme creativity of inspiring, questioning, listening, nagging, teasing, stimulating, and imagining all sorts of possible ways to help me get to the end of this unforgettable journey! No words, actions or achievements can ever even remotely describe the burning love and gratitude that I feel for them! However, I can always endeavour to make them at least as proud of me as the immeasurable love, appreciation, support, encouragement, kindness, and care they have given me, despite any price of efforts or sacrifices they had to endure to make my life a magnificent one!

The size of the heart of my wonderfully loving and high-spirited brother, **Danny**, has once again amazed me beyond anything that my imagination could phantom. I am grateful to him for bringing the tears of pride of seeing him win the best student award at Bournemouth that inspired me to continue my endeavours if only for the possibility of bringing the same tears of joy to him and our parents! But most of all, I am grateful for his unconditional love to kindly offer any assistance necessary regardless of time or circumstances. It is an absolute blessing to have such an unshakeable bedrock ally in life as him!

To love and be loved is one of the most cherished gifts of life that colours every journey, imbuing the spirit of passion and desire in every single action! I have been lucky enough to worship one such gift, **Diana**, whose discipline and playful spirit have served both as an inspiration to me and as a conclusive illustration that you can always have both, for example, you can be gloriously achieving and experiencing a profound feeling of love and passion! She taught me the value of doing the “small things” in life, which, without one’s noticing, stack up and compound over time to take proportions that dwarf the supposedly "big things". She thought me to leap-frog and to love, to love myself, to love her, to love others, and to love conflicts because your muscles grow exponentially when you push them beyond their limits.

I would also like to thank **God** for giving me the grace to complete this thesis and the courage to change what I could, the serenity to accept what I could not, and the wisdom to tell the difference.

Finally, thank you to my examiners, **Professor Neil Lawrence** and **Dr. Siddharth Narayanaswamy** for taking the time to read this thesis and preparing thoughtful questions and suggestions that have made this work even stronger!

Contents

Notation

This section provides a concise reference describing notation used throughout this document, which is consistent with Goodfellow, Bengio, and Courville (2016a). If you are unfamiliar with any of the corresponding mathematical concepts, Goodfellow, Bengio, and Courville (2016a) describe most of these ideas in chapters 2–4.

Numbers and Arrays

a	A scalar (integer or real)
$\boldsymbol{a}$	A vector
$\boldsymbol{A}$	A matrix
$\mathsf{\mathbf{A}}$	A tensor
$\boldsymbol{I}_n$	Identity matrix with n rows and n columns
$\boldsymbol{I}$	Identity matrix with dimensionality implied by context
$\boldsymbol{e}^{(i)}$	Standard basis vector $[0, \dots, 0, 1, 0, \dots, 0]$ with a 1 at position i
$\mathrm{diag}(\boldsymbol{a})$	A square, diagonal matrix with diagonal entries given by $\boldsymbol{a}$
a	A scalar random variable
$\mathbf{a}$	A vector-valued random variable
$\mathbf{A}$	A matrix-valued random variable

Sets and Graphs

$\mathbb{A}$	A set
$\mathbb{R}$	The set of real numbers
$\{0, 1\}$	The set containing 0 and 1
$\{0, 1, \dots, n\}$	The set of all integers between 0 and n
$[a, b]$	The real interval including a and b
$(a, b]$	The real interval excluding a but including b
$\mathbb{A} \backslash \mathbb{B}$	Set subtraction, i.e., the set containing the elements of $\mathbb{A}$ that are not in $\mathbb{B}$
$\mathcal{G}$	A graph
$Pa_{\mathcal{G}}(\mathrm{x}_i)$	The parents of x_i in $\mathcal{G}$

Indexing

a_i	Element i of vector $\boldsymbol{a}$, with indexing starting at 1
a_{-i}	All elements of vector $\boldsymbol{a}$ except for element i
$A_{i,j}$	Element i, j of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{i,:}$	Row i of matrix $\boldsymbol{A}$
$\boldsymbol{A}_{:,i}$	Column i of matrix $\boldsymbol{A}$
$\mathsf{A}_{i,j,k}$	Element (i, j, k) of a 3-D tensor $\boldsymbol{\mathsf{A}}$
$\boldsymbol{\mathsf{A}}_{:,:,i}$	2-D slice of a 3-D tensor
a_i	Element i of the random vector $\mathbf{a}$

Linear Algebra Operations

$\boldsymbol{A}^\top$	Transpose of matrix $\boldsymbol{A}$
$\boldsymbol{A}^+$	Moore-Penrose pseudoinverse of $\boldsymbol{A}$
$\boldsymbol{A} \odot \boldsymbol{B}$	Element-wise (Hadamard) product of $\boldsymbol{A}$ and $\boldsymbol{B}$
$\det(\boldsymbol{A})$	Determinant of $\boldsymbol{A}$

Calculus

$\dfrac{dy}{dx}$	Derivative of y with respect to x
$\dfrac{\partial y}{\partial x}$	Partial derivative of y with respect to x
$\nabla_{\boldsymbol{x}} y$	Gradient of y with respect to $\boldsymbol{x}$
$\nabla_{\boldsymbol{X}} y$	Matrix derivatives of y with respect to $\boldsymbol{X}$
$\nabla_{\mathsf{\mathbf{X}}} y$	Tensor containing derivatives of y with respect to $\mathsf{\mathbf{X}}$
$\dfrac{\partial f}{\partial \boldsymbol{x}}$	Jacobian matrix $\boldsymbol{J} \in \mathbb{R}^{m \times n}$ of $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$
$\nabla_{\boldsymbol{x}}^2 f(\boldsymbol{x})$ or $\boldsymbol{H}(f)(\boldsymbol{x})$	The Hessian matrix of f at input point $\boldsymbol{x}$
$\int f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral over the entire domain of $\boldsymbol{x}$
$\int_{\mathbb{S}} f(\boldsymbol{x}) d\boldsymbol{x}$	Definite integral with respect to $\boldsymbol{x}$ over the set $\mathbb{S}$

Probability and Information Theory

$\mathrm{a} \perp \mathrm{b}$	The random variables a and b are independent
$\mathrm{a} \perp \mathrm{b} \mid \mathrm{c}$	They are conditionally independent given c
$P(\mathrm{a})$	A probability distribution over a discrete variable
$p(\mathrm{a})$	A probability distribution over a continuous variable, or over a variable whose type has not been specified
$\mathrm{a} \sim P$	Random variable a has distribution P
$\mathbb{E}_{\mathrm{x} \sim P}[f(x)]$ or $\mathbb{E} f(x)$	Expectation of $f(x)$ with respect to $P(\mathrm{x})$
$\mathrm{Var}(f(x))$	Variance of $f(x)$ under $P(\mathrm{x})$
$\mathrm{Cov}(f(x), g(x))$	Covariance of $f(x)$ and $g(x)$ under $P(\mathrm{x})$
$H(\mathrm{x})$	Shannon entropy of the random variable x
$D_{\mathrm{KL}}(P \Vert Q)$	Kullback-Leibler divergence of P and Q
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution over $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

Functions

$f : \mathbb{A} \rightarrow \mathbb{B}$	The function f with domain $\mathbb{A}$ and range $\mathbb{B}$
$f \circ g$	Composition of the functions f and g
$f(\boldsymbol{x}; \boldsymbol{\theta})$	A function of $\boldsymbol{x}$ parametrized by $\boldsymbol{\theta}$. (Sometimes we write $f(\boldsymbol{x})$ and omit the argument $\boldsymbol{\theta}$ to lighten notation)
$\log x$	Natural logarithm of x
$\sigma(x)$	Logistic sigmoid, $\dfrac{1}{1+\exp(-x)}$
$\zeta(x)$	Softplus, $\log(1+\exp(x))$
$\lvert\lvert\boldsymbol{x}\rvert\rvert_p$	L^p norm of $\boldsymbol{x}$
$\lvert\lvert\boldsymbol{x}\rvert\rvert$	L^2 norm of $\boldsymbol{x}$
x^+	Positive part of x, i.e., $\max(0, x)$
$\mathbf{1}_{\text{condition}}$	is 1 if the condition is true, 0 otherwise

Sometimes we use a function f whose argument is a scalar but apply it to a vector, matrix, or tensor: $f(\boldsymbol{x})$, $f(\boldsymbol{X})$, or $f(\mathsf{\mathbf{X}})$. This denotes the application of f to the array element-wise. For example, if $\mathsf{\mathbf{C}} = \sigma(\mathsf{\mathbf{X}})$, then $\mathsf{C}_{i,j,k} = \sigma(\mathsf{X}_{i,j,k})$ for all valid values of i, j and k.

Datasets and Distributions

p_{data}	The data generating distribution
$\hat{p}_{\text{data}}$	The empirical distribution defined by the training set
$\mathbb{X}$	A set of training examples
$\boldsymbol{x}^{(i)}$	The i-th example (input) from a dataset
$y^{(i)}$ or $\boldsymbol{y}^{(i)}$	The target associated with $\boldsymbol{x}^{(i)}$ for supervised learning
$\boldsymbol{X}$	The $m \times n$ matrix with input example $\boldsymbol{x}^{(i)}$ in row $\boldsymbol{X}_{i,:}$

CHAPTER 1

Introduction

You can do it if you believe you can!

Napoleon Hill

Deep learning models are difficult to interpret because of their highly complex and non-linear nature. A model is interpretable when there is a human-understandable explanation about the model predictions. The lack of interpretability is a threefold problem:

1. It inhibits adoption of deep learning models, especially in industries under heavy regulation and with a high cost of errors.
2. It makes it difficult to debug existing models, which hampers development progress.
3. It prevents us from utilising the insights learned from the models for knowledge discovery and advancing scientific progress.

Deep Learning In the context of interpretability it has been argued that instead of using Deep Neural Networks (DNNs) we can rely exclusively on simpler models (e.g., logistic regression, decision trees, or decision lists) (Letham et al., 2015). In contrast to these simpler models, DNNs represent information using distributed representations, which can encode exponentially more regions than non-distributed representations[1]. Distributed representations encode implicit generic regularisation strategies that yield better generalisation and statistical efficiency properties for a particular family of AI-hard task in complex real-world domains, such as images, video, audio, and natural language (Bengio, Courville, and Vincent, 2013). The superiority of distributed representations is uncontested across domains such as end-to-end natural speech synthesis (Sotelo et al., 2017), image

[1] Section 2.5 defines and discusses distributed representations in more depth and demonstrates their superiority in terms of expressive power.

recognition (Szegedy et al., 2017; Szegedy et al., 2015), machine translation (Sutskever, Vinyals, and Le, 2014), and speech recognition (Graves and Jaitly, 2014).

Interpretability However, the generalisation and statistical efficiency come at the price of unexpected and challenging to interpret behaviour. The terms of interpretability, explanation, and transparency are still loosely defined. **Interpretability** and **explainability** are often used interchangeably (Lipton, 2016; Adadi and Berrada, 2018; Carvalho, Pereira, and Cardoso, 2019; Hall, 2019) to characterise the property of a model "to *explain* or to present in understandable terms to humans" (Doshi-Velez and Kim, 2017). In this thesis we use Adel, Ghahramani, and Weller (2018)'s definition – an explanation is "a *simple relationship* to something that [humans] can understand". While sometimes **transparency** and interpretability are also used interchangeably (GB, 2017), here we use the term transparency more specifically to denote the ability to understand the internal operations and confidence of the model (Lipton, 2016; Zhou and Chen, 2018a).

A recent surge in Explainable AI (XAI) research aims to increase the DNN transparency to improve interpretability. Transparency of algorithmic systems has also been discussed as a way for end-users and regulators to develop appropriate trust in machine learning models (Adadi and Berrada, 2018; Carvalho, Pereira, and Cardoso, 2019; Guidotti et al., 2018; Murdoch et al., 2019). Interpretability approaches can be generally divided into (1) extrinsic, or post-hoc, and (2) intrinsic, or interpretable-by-design. The latter approaches build models that inherently have high transparency, whereas the former analyse the behaviour of pre-built models. The most popular family of extrinsic interpretability approaches that increase DNN transparency are importance-based explanations (Bhatt et al., 2020). Importance-based explanations may be divided into three main categories: feature importance, sample importance, or hybrids of the two (case-based reasoning) (Adadi and Berrada, 2018; Carvalho, Pereira, and Cardoso, 2019; Guidotti et al., 2018; Murdoch et al., 2019).

Feature importance or saliency methods provide scores for a given input that shows how important each feature (e.g., pixel, patch, word vector) of the input was to the algorithm's decision. Sample importance methods indicate the most relevant samples for a particular decision. Case-based reasoning describes the most important features of the most relevant samples.

Hypothesis 1: Inadequacy of importance-based explanations In this thesis, we focus on extrinsic interpretability of DNNs. We hypothesise that importance-based explanations can describe neither the behaviour of deep learning models with sufficient fidelity, nor the richness and complexity of the learned behaviour. The term *fidelity* is used here to refer to the ability of an explanation method to describe accurately the behaviour of the underlying model. The term *richness* refers to the ability of an explanation method to describe as many different aspects and complexities of the underlying model's behaviour in order to increase a user's semantic understanding of the model. There are three reasons for this inadequacy. First, these methods could be fragile to input (Ghorbani, Abid, and Zou, 2019; Kindermans et al., 2019) or model parameter (Adebayo et al., 2018; Dimanov et al., 2020) perturbations (see Chapter 4). Human experiments demonstrate that feature importance explanations do not necessarily increase human understanding, trust, or ability to correct mistakes in a model (Poursabzi-Sangdeh et al., 2018; Kim et al., 2018). Finally, importance-based explanations are designed to provide explanations for a wide range of models (e.g., random forests, DNNs, ensemble models), which limits their ability to describe behaviours idiosyncratic to a particular model family, such as DNNs. In Chapter 3, we define five different levels of interpretability to measure and benchmark the semantic richness of various explanation methods. We argue that importance-based explanation provide only level 1 explainability.

Fidelity and Comprehensiveness In Chapter 4, we assess the fidelity of feature importance explanations. We use fairness as a proxy measure for the fidelity of an explanation method, and we show that *the apparent importance* of a feature does not reveal anything reliable about the fairness of a model in connection to that feature. We explain how this can happen with an instructive example demonstrating that a model could have arbitrarily high levels of unfairness across a range of popular fairness metrics, even while appearing to have zero dependence on the relevant sensitive feature. Next, we design an adversarial explanation attack to modify the parameters of a pre-trained model that demonstrates that in practice, as well as in theory, *the apparent importance* of a feature does not reveal anything reliable about the fairness of a model. To modify the model parameters, our approach retrains an existing model with a modified loss objective function. Within the modified loss function, we add an 'explanation loss' term to the original loss in the form of the gradient of the original loss with respect to a chosen target feature. The resulting models obtain low local sensitivity to the chosen feature with little loss of accuracy. The low sensitivity generalises to unseen test points for ten features across four datasets according to seven feature importance explanation methods. Our work raises concerns for regulators or auditors hoping to rely on feature importance explanation methods to measure or enforce standards of fairness.

Hypothesis 2: Concept-based Model Extraction for Semantically Higher Level Explanations The second hypothesis of this thesis is that specialised explanation methods can be developed to explain in a semantically higher level the information captured in distributed representation than feature importance explanations. To assess the semantic level of the captured information, we proposed five different levels of interpretability in Chapter 3: (1) feature importance, (2) feature interactions, (3) interpretable factor descriptions, (4) functional descriptions, (5) causal graphs. As an explanation progresses along the semantic levels, it takes into consideration feature interactions, groups different configurations of features and feature values into single factors and assigns meaning to each factor, describes the mathematical functions that map these factors to the model's output, and elicits the causal directions between different factors.

Two independent strands of research have emerged to build on the requirement of enhancing the semantic level of explanations – model extraction and concept-based explanations. Model extraction, or model translation, approaches approximate black-box complex models with simpler models to increase the model transparency. Provided the approximation quality (referred to as *fidelity*) is high enough, the extracted models could preserve many statistical properties of the complex model, while remaining open to interpretation. On the other hand, concept-based approaches aim to provide explanations of a DNN model in terms of human-understandable units, rather than individual features, pixels, or characters. For example, the concepts of a *wheel* and a *door* are important for the detection of cars.

Therefore, we propose two novel function decomposition frameworks for interpreting neural networks with richer semantics using model extraction, bridging the fields of model extraction and concept-based explanations. Both frameworks use model functional decomposition, which is a form of model extraction, to provide different forms of concept-based explanations

Concept-based Function Decomposition Frameworks for Model Extraction Specifically, we consider two different types of model functional decomposition: (1) *(D)ependency (G)raphs for (I)nterpreting (N)eural (N)etworks (**DGINN**)* and (2) (C)oncept-Based (M)odel (E)xtraction (CME). While DGINN extracts class-specific representation using a series of function decompositions, CME extracts more fine-grained concept-based representations using functional decomposition of two functions.

One the one hand, the DGINN framework produces two types of class-specific dependency graphs: (1) layer-wise and (2) neuron-specific. The layer-wise dependency graph indicates the relevant neurons to the specific class in each layer, while the neuron-specific dependency graph indicates the pertinent neurons between a pair of layers given the target class. On the other hand, the CME framework produces a new interpretable model consisting of two functions: (1) input-to-concept function; and (2) concept-to-output function. The extracted model can be used instead of the original model or just to mimic the behaviour of the original model to enhance interpretability.

The design of our frameworks relies on the sparsity of hidden representations model property and other general assumptions about the internal operation of DNNs such as manifolds, natural clustering, and shared factors assumptions (Section 2.2 describes these assumptions in more detail). Given these assumptions, we hypothesise that there are very few neurons that describe well-defined variations within the data that can be encoded within particular concepts or concept values. Our findings suggest that very sparse (5% of the total neurons) representations define two types of low-dimensional manifolds. The first type describes the general variance within a class or concept on a single manifold. The second type of manifolds spatially separates distinct concept values on disjoint manifolds. The benefits of our frameworks are that they can provide both global (describing overall model behaviour) and local (describing model behaviour for a particular instance) explanations with richer semantics and a higher level of interpretability because these low-dimension manifolds can be associated with a human-understandable meaning. Moreover, the CME framework can be used to compare concept-based explanations, thus paving the way towards quantifying, axiomatising, and benchmarking future concept-based explanation approaches.

1.1 Contributions

In summary, the contributions of this thesis are:

1. a rational reconstruction of the Explainable AI field presenting a high-level guideline for measuring the semantic richness of explanations (see Section 3.3.2) and a novel taxonomy of interpretability methods (see Section 3.4);

2. an adversarial approach demonstrating the infidelity of feature importance regarding the fairness of the explained models (Chapter 4);

3. DGINN - a novel framework for interpreting DNNs classification decisions using class-specific representations (Chapter 5);

4. CME - a concept-based model extraction framework, which generates both local and global explanations of DNN models, by approximating DNNs with models grounded in human-understandable concepts and their interactions (Chapter 6).

1.2 Overview

The remainder of this thesis is organised as follows: Chapter 2 describes the foundations of representation learning – the superset of deep learning models and the assumptions we implicitly make about deep learning models. It goes on to make a case for distributed representations. We argue that to explain, we first need to understand the assumptions, behaviour, properties, and conditions that govern the entity which we are explaining. Chapter 3 defines and motivates the terms of interpretability, explanation, and transparency.

It then outlines the characteristics of "good" explanations and presents a novel taxonomy of explanation methods. Chapter 4 portrays an adversarial explanation attack method for modifying a pre-trained model to manipulate the output of many popular feature importance explanation methods with little change in accuracy, thus demonstrating the danger of trusting such explanation methods. We show how this explanation attack can mask a model's discriminatory use of a sensitive feature, raising substantial concerns about using such explanation methods to check model fairness.

Next, we take a step towards concept-based model extraction, demonstrating that specific class (Chapter 5) and concept representation (Chapter 6) can be successfully extracted from DNNs using function decomposition using the DGINN and CME frameworks, respectively. Chapter 7 concludes the thesis and puts forward a position about the future of interpretability research, in which we propose to seriously rethink the evaluation procedures of the field. Doshi-Velez and Kim (2017) and Lipton (2016) argue for human-participant experiments to determine whether a type of explanation effectively communicates the model behaviour to the user. Until the psychological suitability of an explanation is confirmed, we want to start from the fundamentals of the scientific method and define control variables. For example, we want to isolate the effects of the learning process and we want to incorporate ways of evaluating the relationships between features.

1.3 List of publications

The material presented in this thesis has in parts been published in the following publications:

1. You shouldn't trust me: Learning models which conceal unfairness from multiple explanation methods (Dimanov et al., 2020) (Chapter 4).

2. Step-wise Sensitivity Analysis: Identifying Partially Distributed Representations for Interpretable Deep Learning (Dimanov and Jamnik, 2019) (Chapter 5).

3. Now You See Me (CME): Concept-based Explanations via Model Extraction (Kazhdan et al., 2020)[2] (Chapter 6).

The following publications formed part of this PhD research project and present results that are supplementary to this work or build upon it. The work within these publications has been lead by collaborators, and they are not covered in this thesis:

1. MEME: Generating RNN Model Explanations via Model Extraction (Dmitry et al., 2020).

[2] Equal contribution with Dmitry Kazhdan.

2. REM: An Integrative Rule Extraction Methodology for Explainable and Interpretable Data Analysis in Healthcare (Shams et al., 2021).

3. Is Disentanglement all you need? Comparing Concept-based & Disentanglement Approaches (Kazhdan et al., 2021).

4. Failing Conceptually: Concept-Based Explanations of Dataset Shift (Wijaya et al., 2021).

CHAPTER 2

Representation Learning

What I do not understand, I cannot explain.

Chatbot child of Richard Feynman
and Albert Einstein

In this chapter, we argue that a deep understanding of how DNNs work is necessary to explain their predictions. For this purpose, we review the core notions in representation learning that are pivotal to the design of interpretable models and demonstrate their value in describing our expectations about the real-world. These expectations impose eleven implicit assumptions about the data distribution of real-world problems (Section 2.2), which mandate six key requirements of an ideal representation (Section 2.3). The ideal data representation should be (i) expressive, (ii) abstract, (iii) disentangling, (iv) easy to model, (v) compact, and (vi) robust. We highlight partially-distributed representations (Section 2.4) as the best instantiation of these requirements from both a statistical and computational point of view. Nevertheless, partially-distributed representations have three main limitations in terms of their ability to (1) provide interpretable to humans information (interpretability); (2) resist minor corruptions or data distribution shifts (robustness); and (3) generalise to unseen distributions or represent relationships between multiple entities (generalisation) (Section 2.5). In this thesis, we address the interpretability limitation of distributed representations and argue that it can be enhanced when the assumptions encoded in representation learning models are considered more carefully (see Chapters 5 & 6).

2.1 Purpose of Representations

The performance of information processing systems depends on the way of representing information. Defining the best way to represent information begs the question: *What makes a good representation?* The choice of representation depends on the subsequent information processing task and the agent performing the task. For example, the operation of finding an element in a list has a computational complexity of $O(n)$ when the list is represented as a linked list, but $O(\log n)$ when the list is represented as a binary tree.

On the other hand, representing a number in Arabic or Roman numeral form could affect significantly the time required for a human to perform the simple multiplication of 7×14 rather than $VII \times XIV$. Most people would need to convert the Roman numeral into decimal representation, perform the calculation, and convert back to the original format. While humans prefer the decimal domain, computers thrive in binary representations. However, if the relevant information for a particular task is well-separated, we could design ways to translate between different forms of representations. This is why the point of representation learning algorithms, such as linear factor models, autoencoders, Boltzmann machines, neural networks, and probabilistic models with latent variables, is to build a representation that can *untangle the underlying factors of variation, which are relevant to the subsequent task.* Next we explore the assumptions about the real world that facilitate the disentangling of these factors of variation.

2.2 Prior Assumptions

Currently, there are two main strategies to discover the underlying factors of variation. Depending on the availability of additional signal in the form of labels, the strategies can be divided into supervised and an unsupervised. In the supervised learning case, the labels contain a powerful signal about the importance of various features. However, in the more general case of unsupervised learning, where there are no labels available, we can only rely on more indirect clues in the form of prior beliefs, or assumptions, that a developer can impose on the algorithm. Unfortunately, according to the **no free lunch theorem** there is no universally better machine learning model or regularisation technique averaged over *all data distributions* (Wolpert, 1996).

Representation learning imposes a set of assumptions, which encode prior beliefs that make it more manageable to learn and represent real-world data-generating distributions rather than any data-generating distribution, thus tackling the no free lunch theorem (Bengio, Courville, and Vincent, 2013; Goodfellow, Bengio, and Courville, 2016a). There is a family of challenging AI-related tasks, such as computer vision, natural language processing, robotics, or information retrieval that involve complex behaviours that can be described through highly non-linear mathematical functions. These functions have a large number of variations (ups and downs) across their input space, but simple underlying structure (Yao, 1985; Hastad, 1986; Håstad and Goldmann, 1991; Bengio, Delalleau, and Roux, 2006; Bengio, LeCun, et al., 2007; Delalleau and Bengio, 2011; Braverman, 2011).

For instance, the knowledge of the underlying data aspects such as the *position*, *lighting*, and *orientation* of 3D objects, can be enough to describe all pixel intensities within an image. These aspects, called **factors of variation**, describe the changes in the behaviour of the data separately from each other and are often independent. Separate is to say that each factor encodes an individual variation in the data disentangling it from the others. Independent implies that changing one factor does not affect the other factors since their interactions are limited.

Next, we discuss eight[1] of eleven implicit assumptions encoded within representation learning algorithms to disentangle the factors of variation (Bengio, Courville, and Vincent, 2013; Goodfellow, Bengio, and Courville, 2016a). An enhanced understanding of the implicit assumptions of representation learning algorithms can help us leverage, or even manipulate[2], particular model properties to enhance the interpretability, generalisation, and robustness of representation learning algorithms. For example, Chapter 5 demonstrates that leveraging the sparsity, manifolds, natural clustering, and hierarchical organisation assumptions leads to the extraction of class-specific representations, which describe how each output is represented within a DNN. Chapter 6 builds on these findings and leverages the multiple factors and shared factors assumptions to produce explanations in the form of high-level semantic units, termed concepts (Kim et al., 2018; Ghorbani et al., 2019), which are more readily interpretable from a cognitive perspective (Poursabzi-Sangdeh et al., 2018; Kim et al., 2018; Ghorbani et al., 2019).

[1] A more extensive list of implicit assumptions can be found in Appendix B.1.

[2] See Appendix C.

Remark 2.2.1

Before we proceed, let us make the distinction between features and factors. Although sometimes used interchangeably, the two terms differ depending on the setting they are used. In a deterministic setting a *feature*, or *attribute*, describes a distinctive characteristic of the input that differentiates different types of data (e.g., clusters, classes or labels) (Murphy, 2012). In a probabilistic setting, a *factor*, or *clique potential*, is a fundamental building block for representing high-dimensional probability distributions. A mathematical way to represent a factor is through a table or a function that takes a set of random variables (the scope) as the argument and produces a real number, affinity, as the output. The affinity describes the probability of occurrence for all different configurations of the random variables. Hence, factors describe the data generating process to obtain the observed data, while a feature describes a characteristic of the observed data. Sometimes, a characteristic could be a factor, as is the case in the causal factor assumption[a] (Goodfellow, Bengio, and Courville, 2016a).

[a] See Appendix A.

- **Multiple factors assumption**: Assumes that there are more than one factors of variation that explain the observed data. For example, if we take the 3D objects example, the lighting factor on its own would not be enough to explain the pixel intensities. This assumption allows us to easily solve any task provided we can capture and disentangle its key explanatory factors. Section 2.5 describes how this assumption motivates distributed representations with separate control over directions in representation space, such that each entry represents a factor of variation.

- **Causal factors assumption**: Assumes that the generative process is such that the observed data is an effect of the underlying factors of variation, and not vice versa. In this case, if the learned representation truly captures the factors of variation, then its elements represent the causes of the observed data (Schölkopf et al., 2012; Erhan et al., 2010). Hence, the 3D object lighting causes the pixel intensity increase rather than the pixel intensities causing the object to appear brighter. When this assumption holds, the learned model is more robust to changes in the input distribution because these changes are driven by shifts in the distribution of the underlying causal factors. For example, if we assume that $p(\mathbf{x})$ and $p(\mathbf{y}|\mathbf{x})$ are independent (i.e., the exogeneity assumption[3]), then changes in $p(\mathbf{x})$ do not interfere with our model of $p(\mathbf{y}|\mathbf{x})$ (Lasserre, Bishop, and Minka, 2006).

- **Shared factors assumption**: Assumes that different tasks share factors across a common pool of reusable latent factors of variation. Therefore, using one task to

[3] See Appendix A.

extract underlying factors of variation should be beneficial to discover factors relevant to other tasks. Transferring statistical strength of reusable features across tasks and domains motivates the successful application of representation learning algorithms to multi-task learning (Collobert et al., 2011), transfer learning (Goodfellow, Courville, and Bengio, 2012), and domain adaptation (Glorot, Bordes, and Bengio, 2011). As we will see in the next two assumptions and through this chapter, the ability to represent many examples with reusable features projects the input into a rich similarity space, where multiple examples are not constrained to be only *local* neighbours in input space. Therefore, this assumption results in exponential gain in the expressivity of the representation[4].

- **Hierarchical organisation assumption**: Assumes that the world is described by highly complex functions with a considerable degree of variation (ups and downs), but with an *underlying simple structure*, which is hierarchical. The rationale behind this assumption is that humans often describe concepts hierarchically with multiple levels of abstraction. For example, a software engineer prefers to represent information with a hierarchy of reusable components such as functions and modules rather than with one flat main program.

  While the shared factors assumption supposes the existence of reusable components, the hierarchical organisation assumption incorporates the belief that *a hierarchy of reusable components* can describe abstract ideas more easily. For example, we can describe the concept of cars through relationships about objects such as its parts (e.g., tires, windshields and doors). We can represent each of these objects with simpler shapes, such as rectangles, circles, and squares. The shapes can be represented through relationships between straight and curved lines. Naturally, concepts become more abstract as they become increasingly invariant to local input transformations, which are uninformative to the subsequent task.

  Assuming a hierarchical structure has a threefold benefit: (1) contributes to disentangling of factors of variation; (2) leads to exponential gains in representation power because it promotes the reuse of features; (3) induces a prior of building invariant features[5].

- **Manifolds assumption**: Assumes that the probability density of real-world high-dimensional data is highly concentrated along (often non-linear) connected regions of tiny volume (of much smaller dimensionality that the original space), called manifolds (Cayton, 2005; Narayanan and Mitter, 2010; Schölkopf, Smola, and Müller, 1998; Saul and Roweis, 2003; Tenenbaum, De Silva, and Langford, 2000; Brand,

[4] See Section 2.5.

[5] In Appendix B.4.3, we discuss these benefits in more detail.

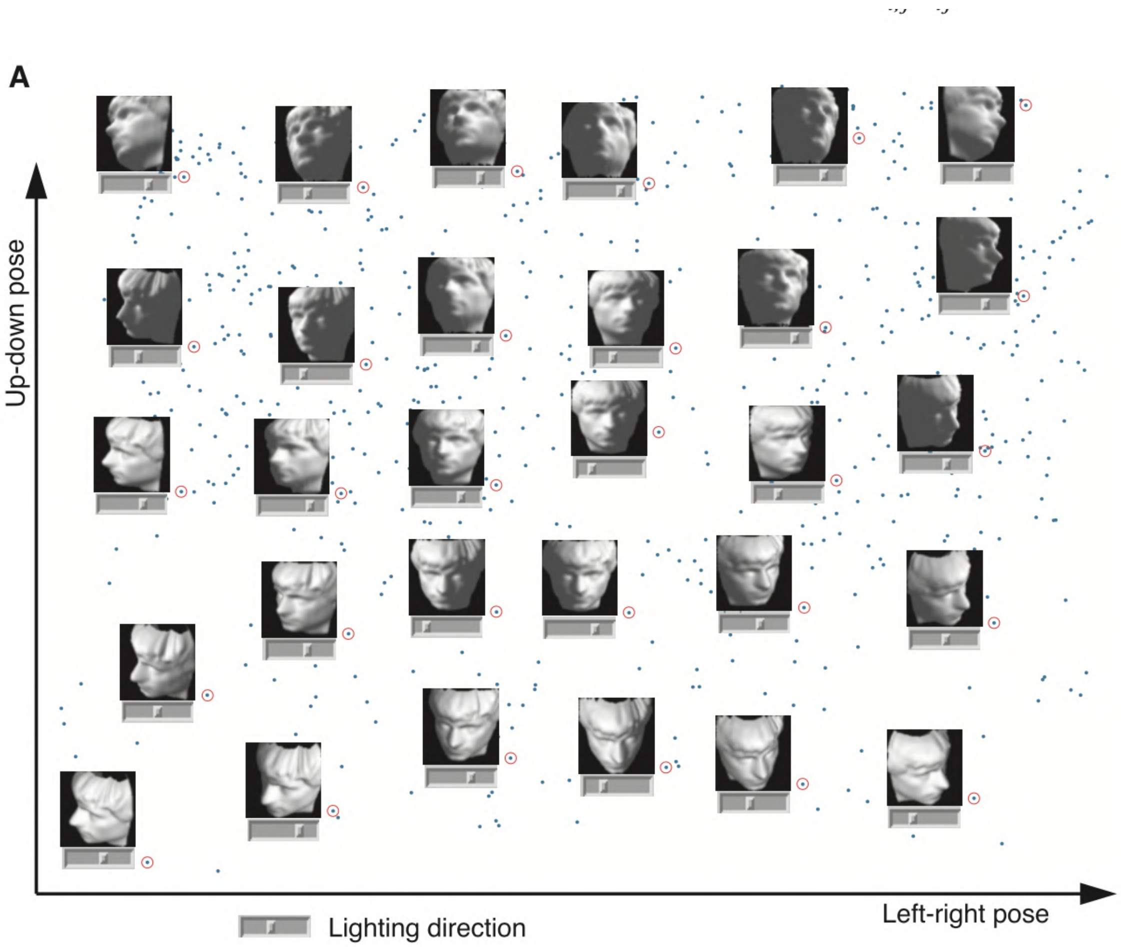


Figure 2.1: Illustration of the manifold assumption in a visual perception problem, with three examples of factors of variation (degrees of freedom): (1) left-right poses; (2) up-down poses; and (3) lighting. A manifold learning algorithm (Isomap (Tenenbaum, De Silva, and Langford, 2000)) learns a three dimensional embedding, the separate dimensions of which correlate highly with the degrees of freedom observed in the data, suggesting the algorithm has learned the intrinsic geometric factors of variation. All data points (blue) are represented in two-dimensional space, with particular samples visualised (red circles). The horizontal sliders represent the third dimension corresponding to lighting. Image reproduced from (Tenenbaum, De Silva, and Langford, 2000).

2003; Belkin and Niyogi, 2003; Donoho and Grimes, 2003; Weinberger, Sha, and Saul, 2004). A manifold is a region consisting of connected data points, such that one point is similar to its surrounding points. Movements along the manifold correspond to specific allowable transformations in input space, which describe the local variations of the input. For example, Figure 2.1 demonstrates how transitions along the y-axis of the learned manifold [6] correspond to up-down pose changes in the original space. The highest variance is observed along directions tangent to the manifold, while directions orthogonal to the manifold have minimal variance. In addition, interpolating between points along the tangent directions can yield new valid points,

[6]

which were not part of the original dataset. However, most of the input space consists of invalid datapoints because there are very few directions tangent to a low-dimensional manifold.

There are five important factors related to learning the structure of a manifold (Bengio and Monperrus, 2005; Rifai et al., 2011b; Verma et al., 2019): (1) noise (i.e., datapoints might lie slightly outside the manifold); (2) curvature (i.e., the degree to which the geometry of the manifold deviates from being a straight line), (3) dimensionality, (4) density (i.e., how sparsely populated is the manifold), (5) number of the manifolds, and (6) curvature of the high-entropy regions between the manifolds (i.e., transitions). In Chapters 5 & 6, we show that we can associate these manifold structures within DNNs to concepts or particular outputs; therefore, enhancing our ability to understand these algorithms.

- **Natural clustering assumption**: Assumes that the points of different classes, or with distinct characteristics, are likely to concentrate along separate manifolds, whereas similar points concentrate along connected manifolds, such that local variations within a manifold do not change the class identity (Rifai et al., 2011b).

  Low-density regions in input space separate the manifolds in a way that the distances between manifolds carry information regarding the difference between the points. Due to this fact, this assumption is sometimes referred to as the **"disconnected manifolds assumption"** because small input perturbations should not be able to transition between manifolds (Rifai et al., 2011b; Bengio and Delalleau, 2011; Bengio, Courville, and Vincent, 2013).

  This manifold geometry induces a **rich similarity space**, in which objects distant apart in input space, come together to form clusters. The rich similarity space yields potent generalisation properties because we can now transfer the knowledge about one point to exponentially many more points on the corresponding manifold[7]. Although originally it is assumed that a manifold corresponds to a single class (Bengio and Delalleau, 2011; Verma et al., 2019), meaning class manifolds do not overlap much, results in Chapters 5 & 6 suggest the presence of overlapping manifolds.

- **Simple factor relationships assumption**: Assumes that simple dependencies describe the relations between factors. For example, the simplest form of relationship is marginal independence. When the explanatory factors are independent of each other, the knowledge of the distribution of one factor generalises to various configurations of the others. We make this assumption when we use a linear classifier such as the softmax final layer in neural networks on top of a linear combination of a learned

[7] See Appendix B.4.

representation (Goodfellow, Bengio, and Courville, 2016a). Hence, we expect that the deeper layers of the networks have learned more abstract and linearly separable features. More sophisticated forms of dependence (e.g., polynomials of low order such as linear, quadratic, cubic, or even quartic) are also reasonable assumptions. Although the degree of the polynomials that usually describes physical properties ranges between two and four (Lin and Tegmark, 2016), currently these high order dependencies are rarely used in practice because of the computational and statistical challenges they introduce[8].

- **Sparsity assumption**: Assumes that the learned features have a high correlation with very few explanatory factors and are invariant to others; consequently, most of the time a feature will not be used to describe an input. For instance, a feature describing a steering wheel, will not be active for an image of a bird. That is to say, if the features describe a binary state – ”present” or ”absent”, we assume that most of the features are absent most of the time. This assumption motivates sparse representations, the intuition for which is that the degree of sparsity controls the insensitivity of a model to small input changes[9].

2.3 The Ideal Data Representation Properties

The ideal data representation would describe the world in view of our beliefs of how the observed data would behave. In contrast to supervised or reinforcement learning, representation learning does not necessarily have clear objectives for training. Therefore, Bengio, Courville, and Vincent (2013) propose that the goal of representation learning is to: *“disentangle as many factors as possible, [while] discarding as little information about the data as is practical”*. A key challenge is how to determine what is *possible* and how much is *practical.* One way to determine at least the practicality is to set the purpose of representations as making subsequent processing tasks easier, more efficient and more robust to noise or changes in the data (Goodfellow, Bengio, and Courville, 2016a).

Based on the ability to facilitate subsequent tasks, there are six primary requirements of an ideal representation[10] (Hinton, McClelland, and Rumelhart, 1986; Elman, 1991; Plate, 2006; Goodfellow, Bengio, and Courville, 2016a):

1. **expressive**: the representation can distinguish between the greatest number of possible input configurations based on the underlying factors of variation that are salient to the subsequent task;

[8] In fact, modern DNNs have been shown to exhibit a strong bias towards simple functions (Pérez, Camargo, and Louis, 2019).

[9] In Appendix C we develop the relationship between sparsity and invariance further.

[10] Appendix B.2 describes each of these requirements in more details.

2. **abstract**: the representation builds different levels of abstraction to facilitate the control of sensitivity and invariance to the underlying factors of variation, depending on their relevance to the subsequent task;

3. **disentangling**: *separates* the underlying factors of variation;

4. **easy to model**: represents sparse and independent factors, or simple factor relationships;

5. **compact**: smaller representations are more efficient both from a computational standpoint (smaller vectors to multiply) and statistical standpoint (fewer parameters to learn, many of which can be reused over many different inputs);

6. **robust**: the representation is (1) not vulnerable to noise, missing data, local perturbations, transformations, and corruptions; and (2) can facilitate out-of-distribution generalisation.

This expressivity of a representation also known as representational power or representational capacity. There is a noteworthy distinction between representational capacity and effective capacity. While the former refers to the theoretical maximum number of encodable regions, the latter refers to the practically resulting capacity after training a model. Notice that these requirements may not necessarily be satisfied simultaneously. A representation that is easy to model (e.g., mutually independent features) might not cleanly separate the underlying causal factors or preserve as much information as possible. Alternatively, an extremely compact representation might not be completely expressive (in Appendix C.1.2 we demonstrate that compactness and robustness are at odds).

Now that we have introduced the prior assumptions about the real-world data distributions and the ideal criteria to represent these distributions, let us turn our attention to the characteristics of representations capable of meeting our criteria.

2.4 Representation Characteristics

The goal of representation learning is to build representations that *disentangle the underlying factors of variation, which are relevant to our subsequent task.* A natural question that follows is: **How do we design representations that disentangle the factors of variation?** Here we describe the characteristics of representations that can disentangle the maximum number of factors in the most practical way possible.

Local or Distributed There has been a long-standing debate whether neural networks represent information in “local”, or “distributed” fashion. In a local, or symbolic, setting the activation of one neuron encodes one concept (Feldman and Ballard, 1982). In contrast, in a distributed setting, a particular activation pattern over a larger group of units represents a concept. In the latter case various concepts are represented by different patterns of activity across the same units (Hinton and Anderson, 1981; Hinton, Sejnowski, and Ackley, 1984; Hinton, McClelland, and Rumelhart, 1986).

Remark 2.4.1

Sometimes the distinction between local and distributed representation is not entirely clear. For example, Van Gelder (2013) illustrate that an 8-bit number can be interpreted as a distributed representation of the numbers 0-255 since this piece of information is contained in the pattern activity across multiple units. However, the number also forms a local representation of the powers of 2, since each bit represents a different power: $2^0, 2^1 ... 2^7$. Therefore, the *interpretation* of representations depends mainly on our perception.

Density The density spectrum of distributed representations refers to the total activity level within an activation pattern. At one extreme we have dense distributed representations, while at the other extreme we have purely local representations. In the middle of the spectrum, we have sparse distributed representations, or partially-distributed representation[11]. A simple way to describe partially-distributed representation is that only a few units are active at any given time, while the inactive units are equal or close to 0. If a dense distributed representation contains N not mutually exclusive, elements, then a sparse representation will have at most $k : k < N$ units active at any one time. At the same time, a local representation has $k = 1$.

Naturally, a smaller number of active neurons decreases the representational power of sparse representations. Nevertheless, even with very low values of k, partially-distributed representations still have an exponentially higher representational capacity (order of $\binom{N}{k}$) than local representation measured as the number of regions that can be carved out in input space (Bengio, 2009). Sparse representations are also biologically plausible since biological neurons form representations that are distributed and sparse (Olshausen and Field, 1997), with 1-4% active neurons at any one time (Attwell and Laughlin, 2001; Lennie, 2003). Furthermore, we will see in our discussion on Superposition that sparse representations also lead to the desirable property of increased robustness of the representation.

[11] In this dissertation, the term "partially-distributed representation" to refer to "sparse distributed representations" and "sparse representations".

Remark 2.4.2

Sparse representations are different from sparse parameterisation. Sparse parameterisation entails that the model **parameters**, or weights, are mostly zeros, while the representations entails that the **activations** of the units are close to zero. Sparse parameterisation does not imply sparse activations, as the non-sparse parameters affect the values of most of the units. In fact, sparse representations impose a complicated implicit prior over the model parameters.

Multifaceted neurons Recent findings (Li et al., 2016; Fong and Vedaldi, 2018; Bau et al., 2017b; Bau et al., 2019) suggest that the interactions between a mixture of local and *partially-distributed* representations (PDRs) govern the DNN decision process. Nguyen, Yosinski, and Clune (2016) provide evidence in support of the claim that alternative neuron activity patterns represent different concepts. These activity patterns make high-level neurons multifaceted. That is, the neurons respond to different types of stimulus (facets) related to the same concept. For example, a high-level neuron responds to both human and lion faces (Yosinski et al., 2015), or an outside and inside view of a movie theatre during different times of the day (see Figure 2.2) (Nguyen, Yosinski, and Clune, 2016).

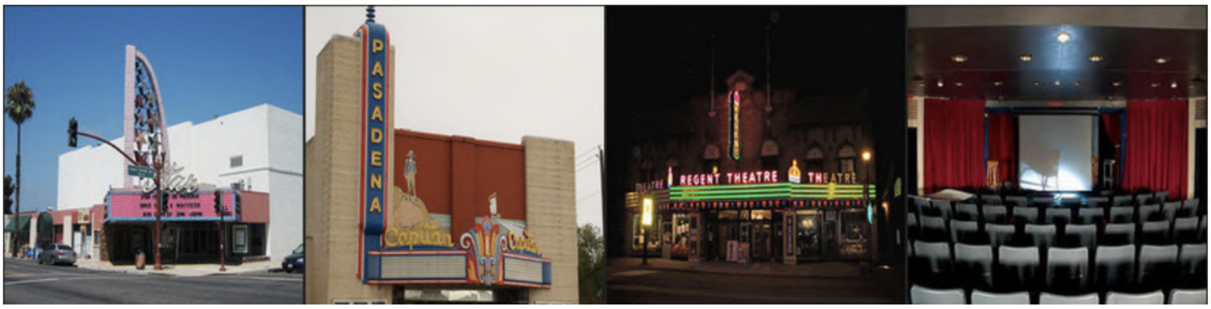


Figure 2.2: High-level neuron responding to different facets of a movie theatre encoding two correlated factors of variation – environment (day / night / cloudy) and location (inside / outside). The figure illustrates the multifaceted nature of neurons (i.e., the same neurons recognise the concept of a "movie theater", regardless of different factors of variation). Image reproduced from Nguyen, Yosinski, and Clune (2016).

Biological neurons are similarly multifaceted (Quiroga et al., 2005). The same neuron can respond to different representations of the same concept: the name of a famous actress ("Halle Berry"), a picture of the actress, and a picture of the actress in movie costume (cat-woman in Batman)[12]. What is noteworthy from a biological standpoint is that visual

[12] In biology, such neuron cells are called *grandmother cells* since a grandmother cell responds to any signal that sensibly discriminates the entity, just as someone would recognise their grandmother in various situations. A grandmother is a common ancestor between many grandchildren cells. In that sense, grandmother cells become *invariant* to various transformations like changing the position, lighting, or orientation of a visual object (Quiroga et al., 2005).

neurons in the inferotemporal cortex[13] of monkeys fire selectively for the general concepts of hands and faces (Quiroga et al., 2005). That is, different neurons fire for hands than for faces. Interestingly, the same neurons fire for different faces, but the activation patterns for each face are different (Freiwald, Tsao, and Livingstone, 2009). This finding hints that the function of the biological representation is to distinguish between different objects of the same kind. Different cells detect constellations of diverse facial parts. In Chapter 6.3.3.1, we demonstrate that CNNs exhibit similar behaviour. Critically, the biological activation is interlinked with the presence of an entire face[14].

Superposition Another term to describe the multifaceted nature of neurons is **superposition** (Plate, 2006). Superposition describes the ability of distributed representations to represent multiple concepts by putting different activation patterns on top of the same neurons ("superimposing") at the same time. Hence, the same set of neurons is representing multiple distinct entities. A challenge to the interpretation of neural networks is that the existence of superimposed patterns makes it difficult to ascertain whether a particular pattern is the result of superposition or not.

A naive way to determine whether superposition is taking place is to compare the similarity between a known pattern a and the current pattern b. If the two patterns are similar, we "conjecture" that a was superimposed with other patterns to form b.

As the number of superimposed patterns increases, the robustness of a representation decreases because the additive effect of multiple different patterns can (1) obscure the original patterns – a phenomenon known as **interference**; or (2) cause the invalid appearance of a known pattern – a phenomenon known as **ghosting** (Plate, 2006).

The interference and ghosting phenomena illustrate that the main representational power of distributed representations, their multifacetedness, is also one of their main weaknesses. Naturally, a denser representation fills up faster and has a higher likelihood of unwanted superposition. Generally, the likelihood of ghosting and interference is proportional to the density and number of distinct patterns and the degree of noise tolerance allowed within the representation, but inversely proportional to the size of the representation (Plate, 2006). Therefore, the emergence of this phenomenon denotes the limit of the effective representational capacity: *"the number of symbols it can store simultaneously and reliably"* (Rosenfeld and Touretzky, 1987; Plate, 2006).

These rough guidelines give us two important intuitions. First, sparse representations have a higher likelihood of being more robust because there is "more room" to store patterns without as much interference, so a larger number of patterns can be superimposed.

[13] The part of the visual system that plays a role most similar to present-day CNNs.

[14] The lack of the ability to fire only in response to the presence of the whole is another form of spurious correlations. It is another one of the main shortcomings of current deep learning visual systems (Arjovsky et al., 2019; Sabour, Frosst, and Hinton, 2017).

Second, sparse, large and non-robust representations are easier to interpret because there is a smaller likelihood of multiple superimposed patterns. We develop these intuitions further in Appendix C.

2.5 Partially-Distributed Representations

Partially-distributed representations are a sparse form of distributed representations, which have been the key building block of deep learning approaches. The purpose of distributed representations is to learn to encode a "complicated" target function with a high degree of variation. This is of paramount importance in the domains of visual and times series data (text, audio, forecasting). The special ingredient of this types of representations is that they are composed of numerous elements that can be *controlled separately*, (i.e., they are not mutually exclusive). Models with distributed representations, such as neural networks with hidden units or probabilistic models with latent variables, leverage these numerous elements to capture the underlying factors of variation that explain the observed data. Theoretically, the power of distributed representations arises from the assumption that if each element represents a different factor of variation (feature), then n features with k values can represent k^n difference concepts (Goodfellow, Bengio, and Courville, 2016a)[15].

Distributed representation have been designed to incorporate multiple of the assumptions described in Section 2.2 using the characteristics described in Section 2.4 to fulfil the requirements set out in Section 2.3. In particular, distributed representations are designed to: (1) disentangle independent, invariant and linearly separable factors (disentangling); (2) form a natural clustering in a rich similarity space of reusable factors connected in a hierarchical structure of simple relationships (abstract). These two design considerations give distributed representations (3) exponential gains in representation power over non-distributed representations (expressive & compact) (Bengio, Courville, and Vincent, 2013; Goodfellow, Bengio, and Courville, 2016a).

Unfortunately, these benefits come at a price. Some of the main shortcomings of distributed representations are the lack of:

- **interpretability**: the ability to be understandable to a human (Doshi-Velez and Kim, 2017);
- **robustness**: the ability to resist minor corruptions and distribution shifts (Hendrycks and Dietterich, 2019);
- **generalisation**: the ability to generalise to unseen distributions or to handle the binding problem (i.e., the ability to maintain associations between multiple concepts (Plate, 2006)).

[15] See Appendix B.4 for more details.

These challenges are exasperated by the lack of well-established benchmarks and methodologies for comparison and evaluation (Andreas, 2019; Do and Tran, 2020; Kornblith et al., 2019). Appendix B.3 describes these shortcomings in more detail.

2.6 Conclusions

In this chapter, we introduced the primary assumptions behind representation learning, which presume that the world can be described by a family of functions that exhibit a well-structured behaviour. These assumptions drive the design of modern representation learning algorithms. We argue that since these assumptions play a pivotal role in designing DNNs, they should also play an equally critical role in explaining modern learning algorithms. We discussed that the ideal representation should be expressive and should build different levels of abstraction to capture and disentangle the highly salient variations in data. DNNs learn partially-distributed representations that superimpose activation patterns to represent an exponential number of concepts, requiring fewer parameters and less training data than non-distributed representation algorithms. These partially-distributed representations define a hierarchical structure of rich similarity spaces, in which meaningful similarities can more easily disentangle and cluster concepts together. However, the expressive power of distributed representations comes at the cost of their interpretability. Appendix B expands on the ideas presented here.

In this thesis, we demonstrate that designing interpretability methods for distributed representations in light of the sparsity, manifolds, and hierarchical organisation assumptions, yields techniques that can describe more aspects of the model behaviour at a semantically higher level that existing approaches. In the next chapter, we introduce a high-level guideline to measure the semantic level of explanations and introduce a taxonomy of the existing approaches. We hypothesise that a deep understanding of the connected manifolds, the separating regions between them, and the mapping between manifolds and human-understandable concepts will lead to more accurate explanations that are more widely accessible (i.e., more intuitive for a broader range of stakeholders). In Chapters 5 & 6, we take two steps towards developing this understanding.

CHAPTER 3

Explainable AI

Success is neither magical nor mysterious. Success is the natural consequence of consistently applying the basic fundamentals.

Jim Rohn

The goal of the Explainable AI (XAI) research field is to *"produce more explainable models, while maintaining a high level of learning performance (prediction accuracy); and enable human users to understand, appropriately trust, and effectively manage the emerging generation of artificially intelligent partners"* (Gunning, 2018). **Interpretable**, or **explainable machine learning (ex-ML)** is a subset of the field of XAI that is dedicated to developing a suite of methods and models that make the behaviour and predictions of *machine learning* systems understandable to humans while achieving high predictive accuracy (Adadi and Berrada, 2018; Molnar, 2019). Interpretable ML, **interactive ML** and **human-in-the-loop ML** are interrelated. Interactive ML is the field of machine learning systems created to work side-by-side with human users. Having a person using interpretable ML to interact with a machine learning model is referred to as a **human-in-the-loop** learning model.

The contribution of this chapter is a rational reconstruction of the Explainable AI field presenting a high-level guideline for measuring semantic richness of the explanations in order to benchmark the state-of-the-art (see Section 3.3.2) and a novel taxonomy of interpretability methods (see Section 3.4). Using our taxonomy we identify four main limitations of current interpretability methods, which we address in Chapters 5 & 6.

The outline of this chapter is as follows: we begin with a survey of the definitions and aims of interpretability, explanation, and transparency (Section 3.1). We then go on to motivate the need for interpretability (Section 3.2). Section 3.3 outlines the characteristics of "good" explanations. Section 3.4 proposes a taxonomy of explanation methods. Sections 3.5 & 3.6 illuminate the limitations of importance-based explanations and representation analysis techniques, respectively, to position the work in the remainder of this thesis.

3.1 Definitions

Interpretability Despite the pressing need, the community still lacks a consensus around the precise definition and objectives of **interpretability** and **explainability** (Lipton, 2016; Adadi and Berrada, 2018; Carvalho, Pereira, and Cardoso, 2019; Hall, 2019). The terms are often used interchangeably to characterise a property of the model (Abdollahi and Nasraoui, 2018). Doshi-Velez and Kim (2017) define interpretability as "the ability to *explain* or to present in understandable terms to humans". The ability to explain can be described as the model's property of *predictability* (Kim, Khanna, and Koyejo, 2016b), or *"simultability"* (Lipton, 2016): "a method is interpretable if a user can correctly and efficiently predict the method's results" (Kim, Khanna, and Koyejo, 2016b). Miller (2019) and Zhou and Chen (2018a) emphasise the importance of the context (i.e., target audience) and define interpretability, or explainability, as how well a human can understand a proposed decision from an AI system in a given *context*.

Aims & Objectives Lei (2017) defines the goals of interpretability as the dual ability to *explain* the model's design and decisions. Ribeiro, Singh, and Guestrin (2016) see value in "understanding the reasons behind predictions" and propose that a technique can gain such understanding if it *explains* the predictions. Zhou and Chen (2018a) highlight the goal of an explanation is to facilitate trustworthy decision making.

Aamodt (1991) suggests that an explanation in the context of an AI system has two purposes: (1) to increase the system's transparency to the user, and (2) to serve as a method for inference and reasoning. As a method for inference, an explanation is a natural way to reason about the world and refers to the internal inference process of an agent. For instance, humans generate a hypothesis and verify its validity internally to interpret an observation. Consequently, Aamodt (1991) advocates that an AI system must be able to explain to itself to conduct higher-level cognitive tasks[1] (Aamodt, 1991). Consequently, the advancement of the XAI field will result not only in an increased interpretability, but also potentially in increased capabilities of modern deep learning systems.

[1] According to Schank (1986), a requirement for understanding and intelligent behaviour is the ability to *internally* rationalise a decision and the process that was used to derive the result. For example, explaining an *expectation failure* (i.e., a situation in which an expected result did not occur), is of paramount importance to understanding and learning (Schank, 1986).

Explanation The aim of interpretability to explain the model's design and decisions begs another question, which philosophers have grappled with for centuries: *What is an explanation?* One proposition is that an **explanation** is the answer to the question "Why?" (Miller, 2019; Weller, 2019). Weller (2019) makes the distinction that this could be about the future ("to what purpose") or about the past ("to what cause"). Another possible definition is that an explanation is an explicit description of the reasons behind a decision to human (Miller, 2019). Yet another more descriptive definition is: "A collection of visual and/or interactive artifacts that provide a user with sufficient description of the model behaviour to accurately perform tasks like evaluation, trusting, predicting, or improving the model" (Hall, Gill, and Schmidt, 2019).

Here, we adopt Adel, Ghahramani, and Weller (2018)'s definition of explanation – "a simple relationship to something that [humans] can understand". Let us now look into the purpose of transparency.

Transparency There is a degree of uncertainty around the terminology in the term of **transparency** (Weller, 2019). The definitions range from the nebulous "the opposite of opacity or blackbox-ness [of a model]" (Lipton, 2016) to the more specific "the process of illuminating how a certain result was produced" (Aamodt, 1991).

Some authors use the terms "interpretable" and "transparent" interchangeably. The Royal Society defines interpretable, or transparent machine learning to be "systems whose workings, or outputs, can be understood or interrogated by human users, so that a human-friendly explanation of a result can be produced" (GB, 2017). On the other hand, Lipton (2016) and Zhou and Chen (2018a) contrast interpretability, which is concerned with the ability to provide explanations about the generated decisions, and transparency, which is a more specific term that describes the ability to understand the *internal* operations and confidence of the model.

Transparency is not a binary property, and some models are inherently more transparent than others (Weller, 2019; Lipton, 2016; Hall, 2019)[2]. Transparency might not always be desirable since there are various types and levels of transparency that depend on the target audience (Weller, 2019; Lipton, 2016). For example, making the decision making process of a loan application system completely transparent makes it vulnerable to malicious applicants, who aim to game the system.

We can "look" at the model's mechanism from four different perspectives (Lipton, 2016): simultability, decomposability, algorithmic and data transparency. **Simultability** refers to the transparency of the reasoning process (i.e., can a human replicate the reasoning of the system). **Decomposability** describes the ability to split the model parameters such that each of its parts corresponds to a description in natural language (this is the

[2] We discuss inherently, or intrinsically transparent models in Section 3.4.2.1.

approach we adopt in Chapters 5 & 6). **Algorithmic transparency** is associated with the understanding of the training process and the trade-offs that the resulting model is making in comparison to similar models in parameter space. This set of models, referred to as the **Rashomon set**, contains models with similar accuracy, but different properties such as interpretability, or robustness (Semenova and Rudin, 2019)[3]. Finally, **data transparency** refers to the knowledge and understanding of the data collection process (Weller, 2019).

3.2 Motivation

The lack of interpretability could have potential risks in three general areas (Goodfellow, Shlens, and Szegedy, 2015; Amodei et al., 2016): (1) mismatch between an AI system's intended and actually learned goals; (2) (a) unawareness of undesirable behaviours such as an AI system's inability to detect context change and generalise to unseen data distributions, and (b) inability to prevent undesirable behaviours even if the awareness was available; (3) inability to extract actionable insights. The mitigation of these risks can be translated into three main benefits of interpretability: value alignment, model improvement, and increased model utility.

Value alignment As ML systems continue to penetrate ubiquitous complex applications and automate human action, they need to adhere to the same principles that govern ethical and legal social functions. That is, ML systems need to be **safe** (Varshney and Alemzadeh, 2016), **predicatable** (Bostrom and Yudkowsky, 2014), **auditable** (Bostrom and Yudkowsky, 2014), and **fair** (Goodman and Flaxman, 2016; Weller, 2017). For example, a robotic arm conducting a brain surgery must aim to minimise all possible harmful effects, not just the cost of morbidity (**safe**) (Varshney and Alemzadeh, 2016). A loan-approval system must be transparent to inspection (**auditable**) (Bostrom and Yudkowsky, 2014) so that we can establish whether it is **predictable** (Bostrom and Yudkowsky, 2014) (i.e., renders similar decisions in similar circumstances) and **fair** (Goodman and Flaxman, 2016).

In Recommender Systems, explanations increase the user's trust, ability to improve the system, and the overall utility and satisfaction (Abdollahi and Nasraoui, 2016; Zanker, 2012). A user's trust in an ML system is crucial when deciding whether to maintain human supervision or hand over complete control to the system.

Model improvement, maintenance, and security Interpretability is the key to improving model performance both through debugging and hyper-parameter optimisation (Freitas, 2014; Liu et al., 2017; Ribeiro, Singh, and Guestrin, 2016; Dimanov and

[3] We discuss potential properties that could influence trade-offs among members of the Rashomon set in Appendix C.

Jamnik, 2019). The world is constantly changing, which means we need to continually monitor and maintain the behaviour of deployed models (**maintainable**) (Sculley et al., 2015). An increase in transparency will enable the more accurate extrapolation of the algorithm's behaviour to unseen situations, thereby increasing our confidence in and ability to maintain the system. Model transparency can increase the robustness of the model against manipulation from external (hackers) and internal (unethical or lazy developers) agents (Bostrom and Yudkowsky, 2014).

Utility Interpretability should be as an important a criterion as predictive accuracy when considering the utility of a model (Freitas, 2014). As the volume of data increases, so does the need for its more sophisticated understanding. An increased understanding of the model leads to valuable insights or the discovery of new knowledge (Quinlan, 1999). Additionally, industry experts have a strong need for actionable insights because they need to know the underlying reasons for a prediction in order to take corrective action.

3.3 What is a "good" explanation?

Now that we have defined and motivated explanations, let us turn our attention to evaluating them. A precise definition of good explanations remains elusive. Multiple efforts have been made to describe the properties of explanation from different angles.

Gilpin et al. (2018) define a good explanation as the situation "when you can no longer keep asking why". This is vague and difficult to measure. Slightly more concrete requirements could be that an explanation needs to be faithful (Fong and Vedaldi, 2019), interpretable (Fong and Vedaldi, 2019), and expressive (Robnik-Šikonja and Bohanec, 2018).

To quantify the faithfulness, interpretability, and expressivity of an explanation, we can rely on model extraction evaluation criteria (Andrews, Diederich, and Tickle, 1995; Jacobsson, 2005; Lughofer et al., 2017; Robnik-Šikonja and Bohanec, 2018) (Štrumbelj, Kononenko, and Šikonja, 2009; Guidotti et al., 2018; Fong and Vedaldi, 2019; Murdoch et al., 2019; Robnik-Šikonja and Bohanec, 2018):

1. **comprehensibility**: the extent to which the model and the predictions are intuitive and informative to humans.

2. **predictive accuracy**: to what extent can we solve the original task using the explanations (maybe in the form of an extracted interpretable model).

3. **fidelity** (or descriptive accuracy): the extent to which we can extract a model to *mimic* the behaviour of another *model* under interpretation.

4. **computational complexity**: the time and space complexity of the explanation model.

5. **stability (or robustness)**: the degree to which explanation methods provide similar explanations for *similar data points*.

6. **consistency**: the degree to which the explanation is the same for *similar models*.

In Section 3.3.2 we propose a high-level guideline to measure *comprehensibility*, which we use to assess state-of-the-art approaches, in Chapter 4 we measure the *consistency* of importance-based explanations, and in Chapters 5 & 6 we measure the predictive accuracy, fidelity, and computational complexity of our explanations.

From a user perspective in the context of recommender systems, the aim of an explanation is to help us make better decisions. Hence, we can measure the **utility** of a "good" explanation, which needs to facilitate the decision-making process in seven ways (Tintarev and Masthoff, 2011):

1. effectiveness (better user decisions),

2. efficiency (faster user decisions),

3. transparency (reasoning behind decisions),

4. scrutability (ability to provide feedback to the system),

5. persuasiveness (convinces the user to make the recommended decision),

6. trust (confidence in recommendations),

7. satisfaction (ease of usability or enjoyment).

From a philosophical, social, and cognitive perspective, we can deduce that humans prefer explanations that are (Grice, 1975; Miller, 2019):

1. simple, but informative (Harman, 1965; Read and Marcus-Newhall, 1993) (c.f. Occam's razor[4] (Thorburn, 1918)),

2. containing the appropriate amount of detail (Keil, 2006)[5];

3. contrastive/counterfactual explanations: the explanation describes why an event P happened instead of another event Q (Q is termed "foil"),

[4] The principle of parsimony, widely known as Occam's razor (c.1287-1347), states that among competing hypothesis of varying complexity that fit the data equally well, we should choose the simplest one because it is more likely to generalise.

[5] This could lead to confirmation bias – the selection of a small, biased subset of reasons for an event rather than the complete set of causes.

4. describing certainty over probability: "The most likely explanation is not always the best explanation for a person". This is know as the Certainty effect in Prospect theory (Kahneman and Tversky, 2013),

5. the result of an interaction, or exchange of knowledge.

Using the utility of explanations in the context of recommender systems and the philosophical, social, and cognitive preferences of users, we can now examine ways of evaluating the comprehensibility of explanations.

3.3.1 Comprehensibility

In Section 3.3, we defined the comprehensibility of explanations as the extent to which the model and the predictions are **intuitive** and **informative**. What is not clear yet is what each of these terms entails. The former measures psychological preferences, including the amount and complexity of the information, while the latter describes "the appropriate *amount of detail*" for a *particular person*. In this section we present ways to measure intuitiveness and demonstrate that the amount of detail of informativeness depends on the target audience (expert vs layman), the task, and the dataset. Consequently, we propose a high-level guideline to help us measure the semantic informativeness of explanations.

Intuitive A number of studies have examined the comprehensibility of explanation from the intuitive side (Miller, 1956; Doshi-Velez and Kim, 2017; Freitas, 2014). Three main factors influence the intuitiveness of explanations as measured in cognitive chunks (Doshi-Velez and Kim, 2017):

1. form of cognitive chunks – the basic unit of the explanation (e.g., raw features, semantically meaningful concepts, datapoints);

2. cognitive load of chunks:

   (a) number of cognitive chunks: our cognitive load is limited to a maximum of 7 items at a time (Miller, 1956);

   (b) level of compositionality: the cognitive load can be managed through organising the chunks in a structured way;

   (c) interactions between cognitive chunks: capture relationships and combinations of chunks[6];

3. uncertainty: how much does the "certainty effect" tax cognitive processing ability.

[6] Doshi-Velez and Kim (2017) proposes this attribute as a separate factor. In Section 3.3.2, we argue that these three aspects jointly determine the cognitive load.

Cognitive load Freitas (2014) proposes measuring the cognitive load of an explanation using the complexity of the structured relationship, or the model, that describes the explanation in terms of three metrics: (1) model size (number of parameters), (2) complexity of the function in terms of (a) degrees of the polynomial (e.g., linear, quadratic, cubic) or (b) feature interactions (Lou, Caruana, and Gehrke, 2012), and **monotonicity** (i.e., a montonic function always varies in the same direction (increasing or decreasing) with any single input variable) (Freitas, 2014).

Task The complexity of an explanation depends both on the user and the context of the decision-making process. For example, if a disaster is imminent, and a prompt decision is needed, a simpler explanation is preferred. However, when time is not a factor, and there could be ethical considerations (e.g., loan application), then a more exhaustive explanation might be preferable (Guidotti et al., 2018).

Target Audience Weller (2019) and Tomsett et al. (2018) make the distinction that there are different stakeholders involved in the explanation of machine learning models. Depending on the target audience, we want to balance the level of detail or information content: (scientific realism vs simplification) (Forster, 1986; Forster and Sober, 1994; "The promise and peril of human evaluation for model interpretability"):

1. **scientific realism (descriptive explanations)**: the most exhaustive possible description of the model behaviour. This level of detail fulfils the goal of transparency and is useful for detailed evaluation in the cases of *debugging*, *auditing*, and *verification.*

2. **simplification** (persuasive explanations): the goal is to communicate effectively to non-technical or general audience users, to influence their decisions.

We unify the stakeholders identified in Weller (2019) and Tomsett et al. (2018) to reach 8 distinct stakeholders:

- developer: the agent creating the system (i.e., designing, training, and testing) (Weller, 2019; Tomsett et al., 2018)

- operator: the agent using the system to produce outputs (Tomsett et al., 2018)

- decision maker: the agent using the outputs of the system to draw conclusions (Tomsett et al., 2018)

- owner: the agent owning the system (Tomsett et al., 2018; Weller, 2019)

- decision subject: the agent about whom a decision has been made (e.g., a loan applicant) (Tomsett et al., 2018; Weller, 2019)

- data subject: the agent whose data was used to train the model (Tomsett et al., 2018)
- examiner: the agent who is validating and verifying the operation of the system (e.g., auditor) (Tomsett et al., 2018; Weller, 2019)
- society: the general public that needs to become comfortable with the strengths and limitations of the system (Weller, 2019)

Data Type We can broadly classify data types into four categories: tabular (e.g., credit scoring), sequential (e.g., text, audio or time series), visual (image data), or spatio-temporal (e.g., video, brain scans). Different data types have varying degrees of comprehensibility, depending on the type of explanation. For example, Huysmans et al. (2011) demonstrate that when presenting *rules*, the most comprehensible form of presentation is a table rather than a list. Here we focus on tabular and visual data.

Next, we propose that a different **level of explainability** in terms of cognitive load, or semantic richness, might be required depending on the task, target audience, and data type.

3.3.2 Levels of Explainability

We conjecture that to extract powerful knowledge from neural networks, we want to increase our level of understanding towards more descriptive and scientifically realistic explanations. In order to measure our degree of understanding and the expressive complexity of the provided explanations, we propose five distinct levels of interpretability:

- Level 1: **feature importance** – the knowledge about the contribution of each feature. Think of this as if we are extracting the simplest possible model linear regression to approximate our black-box predictor.
- Level 2: **feature interactions** (combinations) – a more advanced form of explanations to elicit the interactions between features. Level 1 interpretability assumes marginal independence (i.e., the joint distribution over the input data factorises into independent components). Level 2 interpretability introduces linear factor dependencies between the features. Think of this as if we have included extra feature terms of the form x_1x_2 to our linear regression model approximation.
- Level 3: **interpretable factor descriptions** (e.g., concepts) – this stage uses atomic human-understandable units to describe interactions between the raw variables or the underlying factors of variation. We give an example of Level 3 interpretability in Section 3.4.4.1.

- Level 4: **functional descriptions** (relationships) – the next stage is the ability to describe the functions that govern the factor dependencies from Level 3. This means that instead of saying the contribution of x_1 is some arbitrary value v, we are able to describe the functional family that governs the contribution of factor x_1 or combination of factors x_1x_2 (e.g., $y \propto \sin(x_1)$, $y \propto x_1^2$, $y \propto x_1^{\sqrt{x_2}}$). Think of this as if we are able to extract both the factor interactions and the functions that describes the behaviour of the output w.r.t. these interaction. Essentially, we would be able to approximate our black box predictor through a Generalised Additive Model (McCullagh et al., 1986). The difference between Level 2 and Level 4 ex-ML is that Level 2 checks for the existence of a dependence, whereas Level 4 describes the relationship between the dependence and the output.

- Level 5: **causal graphs** – an even higher form of understanding would be to describe the causal relationships that the model learned about the data. This stage involves the use of explanations in multiple scenarios: (1) associating the functions and feature interactions between all inputs to domain knowledge (e.g., ontologies or knowledge bases), (2) planning to achieve future goals, and (3) reasoning about what would have happened in hypothetical situations (Pearl, 2009; Pearl and Mackenzie, 2018). We conjecture that reaching this level of explainability will require close interactions with domain experts.

A natural question that follows is whether these levels should follow a linear progression. To illustrate the consecutive progression of the levels let us take an example. Let's consider the equation for distance travelled given velocity, time and acceleration: $s = v_it + \frac{1}{2}at^2$. To begin with, the design matrix consists of 3 unknown columns x_1, x_2, x_3. Level 1 feature importance can signify that all three features are important, x_3 is the most important and x_1 is the least important. Level 2 feature interaction description signifies the importance of x_1x_3, x_2x_3, and x_3^2. Level 3 would assign human understandable meaning to these variables as initial velocity (v_i), time (t), and acceleration (a). While Levels 1-3 only indicate the importance of each entity (e.g. features, interactions, concepts) to the model's output, Level 4 describes the functional relationship which defines how the model's output varies with the entity. Hence, Level 4 would explicitly produce the underlying equation $s = v_it + \frac{1}{2}at^2$. Finally, Level 5 would describe the intuition that it is the time and acceleration that determine the travelled distance, and not vice versa.

Some might argue that Level 4 might be achieved without reaching Level 3, as in $s = x_1x_3 + \frac{1}{2}x_2x_3^2$. However, we argue that for Level 4 to bring the necessary comprehensibility sufficient for a diverse set of target audience stakeholders, Level 4 requires **both** the high scientific realism of Level 2's functional descriptions, and the cognitive simplification of Level 3's intuitively interpretable factors.

Section 3.4 illustrates that the vast majority of explanation methods currently fall under Levels 1 & 2, whereas in Chapters 5 & 6 we propose explanation methods that fall under Levels 3 & 4, respectively.

3.3.3 Evaluation

One of the key challenges for XAI research is the lack of well-established evaluation methodologies. Doshi-Velez and Kim (2017) propose three types of evaluation depending on human engagement and end task complexity – application-grounded, human-grounded, and functionally-grounded evaluation:

Application-grounded (real task) Application-grounded evaluation involves conducting domain expert experiments to measure the quality of the system on a a real-world end task. For instance, if an application is designed to facilitate medical diagnosis, it should be evaluated on results such as correctly diagnosed patients, identification of new important facts, and time to correct diagnosis. There are two difficulties with this approach: (1) time and cost of recruiting a sufficiently large pool of domain experts, requesting approvals and conducting the experiments; (2) lack of a baselines, although a starting point could be the performance of the domain expert on the task with and without explanations, or the explanation of a human vs. the generated explanation.

Human-grounded (simple task) Human-grounded evaluation also involves human experiments, but they are conducted on *synthetic tasks* with *laypeople*. The benefit of this evaluation is twofold. First, the subject pool increases due to lower expertise requirements. Second, the general quality of an explanation can be measured through controlled tasks.

Functionally-grounded (proxy tasks) Functionally-grounded evaluation does *not involve human participation*. Hence, this evaluation is much cheaper to conduct because it relies on predefined and measurable notions of interpretability on proxy tasks. Functional evaluation can reliably measure the predictive accuracy, fidelity, computational complexity, stability and consistency of the explanations; however, the evaluation of comprehensibility and utility is less reliable and requires human-based evaluation. Another options is to define axioms that interpretability methods need to abide to across the entire dataset and all possible models of a given model class (e.g., neural networks) (Lundberg and Lee, 2017). We describe the currently used axioms for a particular family of explanation methods in Section 3.5.2.2.

Chapters 5 & 6 rely on functionally-grounded evaluation. Since functionally-grounded evaluation is the most prominent form of assessing the quality of interpretability methods, in Chapter 6 we propose a framework that can be used to benchmark a particular family of explanations, concept-based explanations (see Section 3.4.4.1), on functionally-grounded tasks.

Guidelines Finally, Tintarev and Masthoff (2011) propose four design guidelines for developing and evaluating explanations:

1. Design with target user benefits in mind and develop evaluation metrics to measure the extent to which they are accomplished.

2. Presentation of the explanations is critical – it could either strongly enforce a point or obscure it in confusion.

3. "Be aware that the evaluation of explanations is related to, and may be confounded with, the functioning of the [underlying model behaviour]".

4. Consider whether the relationship between the algorithm and the presentation of the explanation faithfully reflect the underlying behaviour.

In Chapter 4, we highlight that Guidelines 3) & 4) are violated for feature-importance explanation because they do not reliably reflect the underlying model behaviour and that to some extent this might be the result of confounding factors. Therefore, we propose that one way to fulfill these requirements is to evaluate explanation methods in well-controlled experimental settings – **fixed controlled datasets** and **fixed controlled models**. In Chapter 6, we introduce one fixed controlled dataset evaluation. A fruitful area for further work would be to define deep learning models, whose behaviour is manually crafted, in order to evaluate explanations.

3.4 Taxonomy

This thesis focuses on interpretable ML and particularly on interpretability methods for feed-forward and convolutional neural networks. There are some techniques to enhance the transparency of naive-Bayes (Kulesza et al., 2011; Becker, Kohavi, and Sommerfield, 2001), decision trees (Ankerst et al., 1999), support vector machines (Fung, Sandilya, and Rao, 2005), and hidden Markov models (Baum and Petrie, 1966) that we do not discuss here. Although we briefly mention some techniques particular to recurrent neural networks (RNNs), this is by far a non-exhaustive list. Research on reinforcement learning interpretability (e.g., (Kazhdan, Shams, and Liò, 2020)) is also outside the scope of this survey. For further information on these subjects, we refer the reader to some excellent surveys (Guidotti et al., 2018; Carvalho, Pereira, and Cardoso, 2019; Adadi and Berrada, 2018; Murdoch et al., 2019) and books (Zhou and Chen, 2018b; Samek et al., 2019; Hall, 2019; Molnar, 2019).

Several taxonomies for interpretable ML have been developed in parallel with ours (Samek et al., 2017; Murdoch et al., 2019; Hall, 2019; Adadi and Berrada, 2018; Carvalho, Pereira, and Cardoso, 2019; Mojsilovic and Mojsilovic, 2020; Guidotti et al., 2018). Here we unify, expand and reorganise disparate terms from these taxonomies. We are the first to propose that existing classifications describe the extremes of particular spectra, overlooking important ideas such as semi-local and network-agnostic explanations. Further, we are the first to propose that explanations can be seen as functions, and as such they have different functional domains and ranges.

Our taxonomy describes interpretability approaches in terms of:

- The focus of the explanation: data vs model (Section 3.4.1).
  - The stage of development of model-based explanations: intrinsic vs extrinsic (Section 3.4.2.1).
  - The families of algorithms model-based explanations can be applied to: model-agnostic, model-specific, network-agnostic, network-specific (Section 3.4.2.2).
  - The entity of interpretation of the internal workings for network-specific approaches: neuron, neuron-interactions, layer (Section 3.4.2.3).
- The scope of the information: local, semi-local, global (Section 3.4.3).
- The domain and range of the explanation function: input space, output space, hidden space, and concept space (Section 3.4.4).
- The presentation of the explanation: importance, mathematical, visual (Section 3.4.5).

Outline In this section we introduce the different categories of explanations, whereas Sections 3.5 & 3.6 focus on particular families of interpretability approaches relevant to the remainder of this thesis. Specifically, Chapter 4 expands on the limitations of feature importance explanations (introduced in Section 3.5). Chapters 5 & 6 rely on the framework of model extraction (introduced in Section 3.4.5.2) to develop semantically-higher-level forms of representation analysis explanations (introduced in Section 3.6). In contrast to existing representation analysis approaches, our CME framework (Chapter 6) highlights the relationship between hidden representations and *concepts* rather than the relationship between hidden representations and inputs or outputs.

3.4.1 Data-based vs Model-based Explanations

Interpretability methods can be broadly divided into two categories depending on whether we want to understand the variation in the data, in isolation of the model (1) **data-based**; or understand the behaviour of a model (2) **model-based**. In this thesis we focus on model-based explanations; therefore, we briefly survey two examples of data-based explanations – (1) case-base reasoning and (2) disentangling interpretable representations, before focusing on model-based explanations in more depth.

3.4.1.1 Data-based explanations

Case-based reasoning Case-based reasoning, or exemplar-based reasoning, is an intuitive part of the human decision making process (Aamodt and Plaza, 1994; Cohen, Freeman, and Wolf, 1996; Newell, Simon, et al., 1972; Cunningham, Doyle, and Loughrey, 2003). The idea is to elicit representative examples, or *prototypes*, that describe a group (cluster) of samples that share a certain characteristic (e.g., class label). For example, k-Nearest Neighbours (KNN) (Fix and Hodges Jr, 1951), K-Means clustering, other types of Mixture models (Everitt, 1985) can all be seen as forms of case-based reasoning data explanations. We can also explain the data in terms of the most important features, as in the case of Latent Direchlet Allocation (LDA) (Blei, Ng, and Jordan, 2003), or sparse principal components analysis (sprase PCA) (Zou, Hastie, and Tibshirani, 2006). Subject experiments suggest that humans find case-based reasoning more intuitive than feature importance, even though both methods convey similar information (Cunningham, Doyle, and Loughrey, 2003).

There are four broad types of case-based reasoning: (1) samples, (2) features, (3) important samples and features (prototypes), and (4) contrastive prototypes (criticisms) (Blei, Ng, and Jordan, 2003; Kim, Rudin, and Shah, 2014; Kim, Shah, and Doshi-Velez, 2015; Kim, Khanna, and Koyejo, 2016a).

Figure 3.1 demonstrates the difference between samples, features, and prototypes. While prototypes boost the *intra*-group interpretability (i.e., the commonalities between particular instances that have been classified similarly), criticisms elicit differentiating factors between prototypes to boost *inter*-group interpretability (i.e., the aspects of the data that distinguish one decision from another). For example, the Bayesian Case Model (BCM) (Kim, Rudin, and Shah, 2014) extracts a set of the most important prototype features alongside important samples. On the other hand, Mind the Gap (MGM) (Kim, Shah, and Doshi-Velez, 2015) and MMD-critique framework (Kim, Khanna, and Koyejo, 2016a) describe the separation between groups and the intra-group variation with separating features and outliers (termed criticisms), respectively.

Remark 3.4.1

User studies (Kim, Shah, and Doshi-Velez, 2015), as well as, philosophy discourses (Miller, 2019), suggest that humans find *separation rather than variation* more informative. This type of explanation is termed, contrastive, or counterfactual.

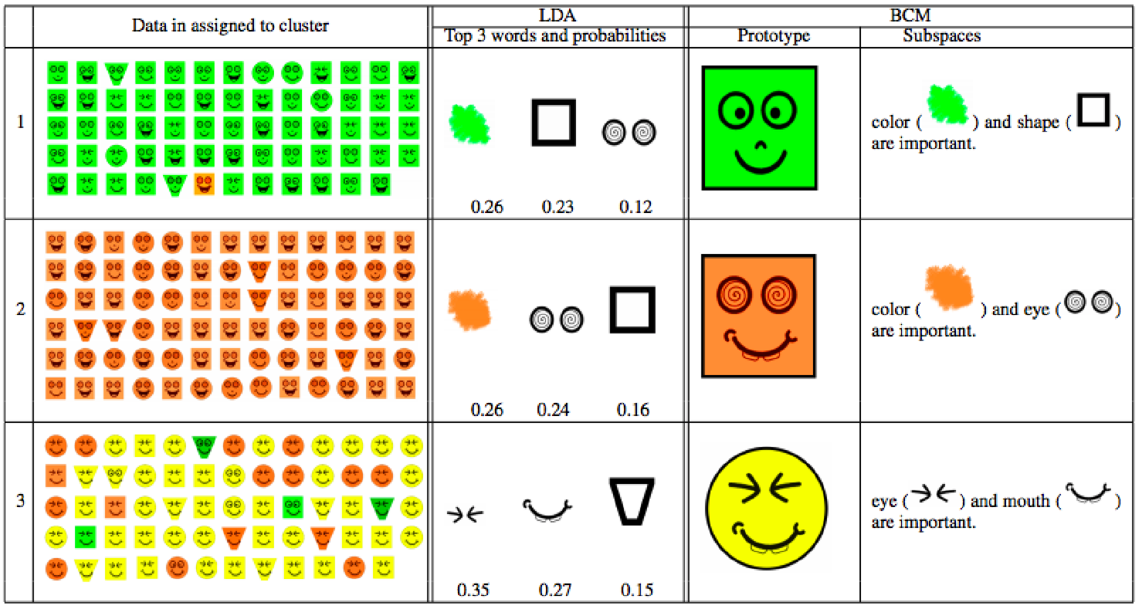

Data in assigned to cluster
LDA
Top 3 words and probabilities
BCM
Prototype
Subspaces
1
0.26
0.23
0.12
color ( ) and shape ( ) are important.
2
0.26
0.24
0.16
color ( ) and eye ( ) are important.
3
0.35
0.27
0.15
eye ( ) and mouth ( ) are important.

Figure 3.1: Comparison between **Feature Importance** explanations computed using Latent Dirichlet Allocation (LDA) and **Prototype** explanations computed with a Bayesian Case Model (BCM). Reproduced from (Kim, Rudin, and Shah, 2014).

Interpretable Representations Autoencoders (Hinton and Salakhutdinov, 2006) and generative adversarial networks (GANs) present ways to learn the underlying data distribution and to disentangle the factors of variation that describe the generative process. Two prominent examples of this are Info-Gan (Chen et al., 2016) and variational autoencoders VAEs (Kingma and Welling, 2014; Rezende, Mohamed, and Wierstra, 2014; Higgins et al., 2017). Info-GANs maximise the variational (lower-bounded) mutual information between a small subset of latent variables and the input data in a min-max game framework. On the other hand, VAEs use gradient-based optimisation to learn an approximate distribution q, which maximises the evidence lower bound (i.e., lower bound of the log-likelihood of the observed data). This technique is called learned approximate inference. Both approaches extract latent representations that encode independent factors of variation. Each factor describes the generative process of the observed data in a human-interpretable way. For example, a factor can be the angle or thickness in digit classification or hairstyle in facial recognition.

Interpretable representation approaches are powerful tools for gaining information about the data distribution. However, there might be a potentially infinite number of factors of variation, such that only distinct subsets are relevant for particular tasks [7]. GANs and VAEs essentially encode an implicit prior over the possible tasks that might concern us. This prior might introduce blind spots and biases that could be difficult to

[7] For more details, see Appendix B.2.2.

detect, or might not be applicable for certain tasks. For example, there maybe be three factors of variation in a dataset of 2D objects – shape, rotation, and scale. Without supervision, there is no guarantee that a GAN or a VAE would learn all three factors. Even if shape and rotation are correctly discovered, there is no way of telling whether there is a third factor (Locatello et al., 2020). In Chapter 6, we illustrate that the same scenario can be observed for feed-forward DNNs.

3.4.2 Model-based explanations

Model-based explanations focus on explaining the behaviour of the model explicitly, in the context of the data. In the remainder of this section, we will describe different types of model-based interpretability.

3.4.2.1 Intrinsic vs Extrinsic Explanations

Linear sparse models, rule lists (Clancey, 1983; Steels, 1985; Van Melle, 1980; Lakkaraju, Bach, and Leskovec, 2016), decision trees (Quinlan, 1986), and case-based reasoning approaches (e.g., KNN) (Fix and Hodges Jr, 1951) have higher comprehensibility (i.e., they are more readily interpretable) (Freitas, 2014; Huysmans et al., 2011; Ribeiro, Singh, and Guestrin, 2016; Kim, Rudin, and Shah, 2014; Hall, 2019). The explanation of such models is part of their internal operation. As such, these explanations cannot be immediately transferred to different types of models. Hence, we call methods that provide explanations on the basis of their inherent design **intrinsic**, **intrinsically transparent**, white-box (Abdollahi and Nasraoui, 2018), glass-box (Zahavy, Ben-Zrihem, and Mannor, 2016), or transparent (Hall, 2019) models. On the other hand, neural networks involve complex behaviours with billions of parameters, which makes them difficult to interpret intrinsically. Therefore, we need to develop post-hoc methods that explain the behaviour of such models after they have been trained. We refer to such explainability methods as **extrinsic**. In this thesis, we focus on extrinsic interpretability, thus, we briefly list the recent developments of intrinsic methods.

Recent intrinsic methods Intrinsically transparent models trade off model complexity (consequently performance) for increased interpretability. Recently, more complex models that maintain a higher level of performance while remaining relatively interprertable have been proposed. Examples include Supersparse Linear Integer Models (SLIM) (Ustun and Rudin, 2016), Explainable Neural Networks (XNNs) (Vaughan et al., 2018), Generalised Additive Models with interactions (GA2Ms) (Lou et al., 2013), and Bayesian Rule Lists (Letham et al., 2015; Wang et al., 2016).

On the other hand, neural networks have a very low intrinsic comprehensibility; hence, to increase their transparency, we usually require extrinsic, or post hoc, methods, which are developed separately from the model.

Attention Intrinsic techniques have been developed to increase the transparency of neural networks. For instance, recently it has been proposed that attention mechanisms (Bahdanau, Cho, and Bengio, 2015; Xu et al., 2015; Paulus, Xiong, and Socher, 2017; Lei, 2017) can be used to provide explanations in the context of RNNs on NLP tasks and CNNs on vision tasks (Zhou et al., 2016; Rocktäschel et al., 2016; Walker, Ji, and Stent, 2018; Thorne et al., 2019). However, it is still unclear whether attention mechanism can be faithfully used to explain RNN models (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019).

Another example of more intrinsically explainable architectures are RNNs with adaptive computation time (Graves, 2016) and Differential Neural Computers (DNCs) (Graves et al., 2016). These models encode internal states that give their developer an intuition about the information that the network is using and the current beliefs of the system.

Let us now turn our attention to extrinsic approaches.

3.4.2.2 Model-agnostic vs Model-specific Extrinsic Explanations

Extrinsic methods range from model-agnostic to model-specific. Model-agnostic approaches are applicable across a wide range of ML models, albeit at the cost of explanations which have lower complexity. On the other hand, model-specific approaches increase the transparency of the examined model, but are only applicable to specific model families.

In this thesis, we focus on neural-network specific approaches, which fall in-between model-agnostic and model-specific. Neural-network specific approaches can be broadly categorised into two groups: **network**, or architecture, **agnostic**; and **network**, or architecture,-**specific**. For example, Deconvolution (Zeiler and Fergus, 2014) only works for CNNs with ReLU activations, while Layer-wise Relevance Propagation (LRP) (Bach et al., 2015) can be applied to any network with monotonous activations. In contrast, model-agnostic approaches have access only to the inputs and outputs of the model, so they usually train a **surrogate model** to approximate the behaviour of the original model (Baehrens et al., 2010; Wang et al., 2012; Ribeiro, Singh, and Guestrin, 2016). As such, the model-agnostic approaches fall under the category of **functional** approaches because they consider the model as a black-box and assume access only to the input-output relationship. On the other hand, **topological** approaches are a type of neural-network specific approaches, which assume access to and use the topology of the network to produce explanations with higher fidelity.

3.4.2.3 Unit-wise vs Layer-wise

Topological approaches can be further separated into **unit-**, or **neuron-wise**, and **layer-wise**. Unit-wise approaches examine individual neurons (Erhan et al., 2009; Girshick et al., 2014; Goodfellow et al., 2009), whereas layer-wise methods investigate the behaviour of the entire representation within a layer (i.e., treat all neurons as a group) (Girshick et al., 2014; Yosinski et al., 2014; Williams, 1986; Mahendran and Vedaldi, 2015; Alain and Bengio, 2017).

If unit-wise methods sit at one extreme of the spectrum, and layer-wise ones sit at the other, we argue that there is a lack of methods that sit in between, which we term *neuron-interaction approaches.* In Section 2.4, we described the hypothesis that information in a DNN is represented in the form of partially-distributed representations, such that groups of inter-neuron interactions encode the relevant information. We argue that very few network-specific approaches consider neuron-interactions. In contrast, in Chapter 5, we propose a method to analyse neural networks leveraging precisely these interactions.

3.4.3 Granularity/Scope of information

The task and stakeholder determine the granularity (coarseness), or the scope of the information for an explanation (Weller, 2017; Tomsett et al., 2018). The spectrum of explanation granularity ranges from local, or instance-specific, to global, or model-centric explanations. The terms local and global refer to the size of the neighbourhood that the explanation is describing. As such, the scope of information can be seen as the region within which an explanation is valid (also known as coverage) (Ribeiro, Singh, and Guestrin, 2018). For instance, **local** explanations describe the behaviour of the model or the characteristics of the data for a particular sample, or instance; hence, the term **instance-specific** (e.g., feature importance) (Simonyan, Vedaldi, and Zisserman, 2013; Zeiler and Fergus, 2014; Zintgraf et al., 2017; Shrikumar, Greenside, and Kundaje, 2017; Landecker et al., 2013; Bach et al., 2015; Montavon et al., 2017).

On the other hand, **global** explanations give an overall description of model behaviour across the entire dataset (e.g., activation maximisation (Erhan et al., 2009) and model extraction (Zilke, Mencía, and Janssen, 2016; Chen et al., 2017; Krishnan, Sivakumar, and Bhattacharya, 1999; Sato and Tsukimoto, 2001; Kazhdan, Shams, and Liò, 2020)). Once again, we argue that local and global explanations are the extremes of a spectrum, such that in between there exist semi-local explanations. We understand **semi-local** explanations to mean descriptions of groups of points, or sub-populations. In Chapter 5, we propose one type of semi-local explanations – class-specific explanations. That is, explanations of the model's behaviour in relation to all datapoints of the same class label.

The benefit of instance-based explanations is that they provide information about a specific point of interest, which is obscured in the global view. Local explanations are useful for the decision subjects, who receive valuable feedback about the decision for their case. In contrast, global explanations are useful for developers and examiners, who can verify that the system is operating as intended, identify biases, and alleviate potential problems. However, global explanations could give too high level of an understanding. Therefore, we propose semi-local explanations, as a means to increase the granularity of our understanding. Additionally, in Chapter 6, we propose a framework that can provide both local and global explanations, thus getting the best of both worlds.

3.4.4 Domain space of explanations

An alternative way to look at explanations is as mappings between different spaces. In particular, explanations can map between any combination of the following spaces: a) **input** space; b) **output** space; c) **hidden** space; d) **concept** space.

For example, case-based reasoning gives explanations in terms of the input space. Importance-based explanations describe the input-output mapping of a model (i.e., what kind of inputs are important for specific outputs). Here we describe two additional spaces that can be used to enhance our interpretation of DNNs: the concept space and the hidden space.

3.4.4.1 Concept-based Explanations

A concept is a human-understandable unit, rather than a raw variable, single feature, pixel, or character. For example, the concepts of a *wheel* and a *door* are important for the detection of cars. Concept-based approaches aim to provide explanations of a DNN model in terms of these human-understandable units. Figure 3.2 illustrates an example of using concept explanations for bird recognition.

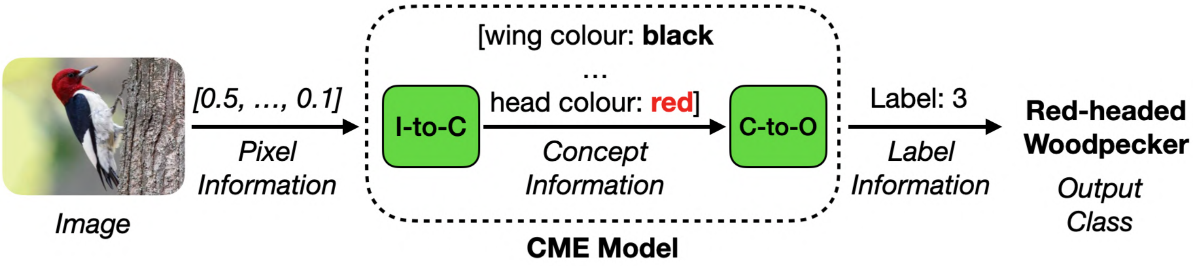


Figure 3.2: Concept-based model extraction describes the decision making process of a bird classifier in terms of human-understandable units such as head and wing colour. Image reproduced from (Kazhdan et al., 2020).

Remark 3.4.2

The understanding of the term "feature" has evolved. The term "variable" used to describe the raw input, whereas the term "feature" represented more general or useful information that is extracted from raw input variables (Guyon and Elisseeff, 2003). More recently, the idea of "concepts" emerged to describe the previous meaning of a feature, while the term "feature" is now usually used to describe a "variable".

Concept-based explanations have been used in a wide range of different ways, including: inspecting what a model has learned (Ghorbani et al., 2019; Yeh et al., 2019), providing class-specific explanations (Kim et al., 2017), and discovering causal relations of concepts (Goyal, Shalit, and Kim, 2019). For example, Testing with Concept Activation Vectors (TCAV) (Kim et al., 2018) examines the behaviour of a hidden representations within a particular model layer in directions of manually pre-defined concepts. Automatic Concept Extraction (ACE) (Ghorbani et al., 2019) is a way to extract such concept directions automatically using superpixel image patches. Interpretable basis functions (Zhou et al., 2018) use only the penultimate layer of a neural network to define a context-specific concept space as a linear combination of basis input space vectors. Network Dissection (Bau et al., 2017a) and Net2Vec (Fong and Vedaldi, 2018) use the convolutional layers to perform concept-based segmentation using concept bounding box annotations.

Similarly to Concept-based Model Extraction (CME – described in Chapter 6), Net2Vec proposes to classify rather than segment concepts. In parallel with our work several approaches have been proposed that also fall under Levels 3 & 4 explainability categories. Concept Bottleneck Models (CBMs) (Koh et al., 2020) and Concept Whitening (Chen, Bei, and Rudin, 2020) also produce an intermediate set of human-specified concepts given a particular input. ProtoPNet introduces an intrinsic concept-based explanation method that uses case-based reasoning to compare image patches to prototypes in training set (Chen et al., 2019a). As such, the image patches can be seen as another form of concepts that provide local explanations. CBMs and Concept Whitening methods regularise a CNN to output a concept representation within one of their layers, whereas ProtoPNet introduces a new architecture. Hence, these methods can be classified as intrinsic, while CME is an extrinsic approach since it does not require any model alterations. In addition, these methods provide only local explanations. On the other hand, CME provides both local and global explanations because it can describe the relationships between concepts and the model outputs in general, as well as for individual predictions.

There are three main limitations to current concept-based explanation approaches: First, existing concept-based explanation approaches are capable of handling binary-valued concepts only, which means that multi-valued concepts have to be binarised first. For instance, given a concept such as "shape", with possible values 'square' and 'circle', these approaches have to convert “shape” into two binary concepts ‘is_square’, and ‘is_circle’. Therefore, the concept space of these approaches encodes the presence or absence of every possible concept value in a separate dimension, using negative sampling. This definition has three important implications: (1) these approaches are computationally expensive because the concept space has an extremely high cardinality; (2) mutually exclusive concepts can be assigned to a single datapoint; (3) mappings from concept space to output space are highly error-prone because negative sampling is only capable of describing directions rather than regions in hidden space. In contrast, the concept space we define in Chapter 6 is axis-aligned with concept variation, decreasing the cardinality, accounting for mutually exclusive concepts, and resulting in better mapping quality.

Second, extracting concepts from a single layer imposes an *unnecessary* trade-off between low- and high-level concepts. Chapter 6 demonstrates that different layers of the network have varying sensitivity concerning different concepts. Hence, we can extract concepts with higher accuracy by focusing on multiple layers.

Third, these methods can only describe concept importance for particular outputs, whereas our method, CME, can describe the functional relationship between concepts and outputs. Consequently, our approach provides Level 4 explainability and makes a substantial increase in the level of semantic information provided in comparison to input-output explanation methods such as importance-based explanations (See Section 3.5).

3.4.5 Presentation

Explanation methods may be classified depending on the medium through which they communicate the learned information to the user as importance-based, mathematical, or visual.

3.4.5.1 Importance-based

Importance-based explanation describe the contribution of a particular entity (e.g., sample or feature) to a specific outcome. Importance-based explanations are usually local explanations, although in Chapters 5 & 6 we demonstrate that it is possible to aggregate importance-based explanations to provide global explainability. We introduce importance-based explanations in Section 3.5.

3.4.5.2 Mathematical

Mathematical explanations are typically global explanations that describe the functional properties of the model using rules, decision tress, or polynomials. A prominent example of mathematical explanations is the model extraction vein of work.

Model Extraction We can view concept-based explanation methods as a way of communicating the transformation that a DNN applies between the input space and concept space. On the other hand, model extraction techniques extract rules (Andrews, Diederich, and Tickle, 1995; Jacobsson, 2005; Zilke, Mencía, and Janssen, 2016; Chen et al., 2017), decision trees (Krishnan, Sivakumar, and Bhattacharya, 1999; Sato and Tsukimoto, 2001), or other more readily interpretable models (Kazhdan, Shams, and Liò, 2020) to describe how the **model's output** varies w.r.t. or across the entire domain of the **input features** (Tan et al., 2019). Provided the approximation quality (referred to as *fidelity*) is high enough, an extracted model can preserve many statistical properties of the original model, while remaining open to interpretation.

Similarly, CME (the method we propose in Chapter 6) approximates complex models with simpler, more interpretable ones. However, our extracted models present the variation of model output w.r.t. human-interpretable **concepts**, not input features.

Model extraction approaches are useful because they provide global explanations about the model behaviour, so that a wide range of stakeholders, such as developers, operators, decision makers, examiners, and owners, can make sure the decision making process is aligned with their expectations. Additionally, the complex black-box model can be replaced with the extracted model to provide higher predictability or the decision maker can adopt their choice based on the learned information and not use any model all together. For example, in the healthcare and criminal justice systems simple checkbox-style scoring systems can be extracted to standardise decision making (Rudin and Ustun, 2018).

3.4.5.3 Visual

Visualisation approaches can be split on the basis of the context domain they are portraying into: synthetic input generation (e.g., Activation Maximisation (Erhan et al., 2009) , Inversion (Williams, 1986; Mahendran and Vedaldi, 2015)), dimensonality reduction (e.g., PCA (Hotelling, 1933), t-SNE (Maaten and Hinton, 2008)), functional description (e.g., Partial dependence plots (PDP) (Friedman, 2001), Accumulated local effects (ALE)) (Apley and Zhu, 2016), importance heatmaps, and architecture visualisation.

Synthetic input generation methods interpret the hidden-to-input relationship. These methods solve an optimisation problem to produce inputs that describe the stimulus, which maximally activates a neuron or group of neurons. **Dimensonality reduction** techniques project the hidden space into lower dimensions to investigate the properties of the internal representations. **Functional description** visualisations describe the behaviour of the model or its internal workings across a range of inputs. They depict the relationship between a specific feature and the output of the model by marginalising the effect of the remaining features. As such, the functional description explanations give global explanations of the model behaviour in terms of a particular feature. An

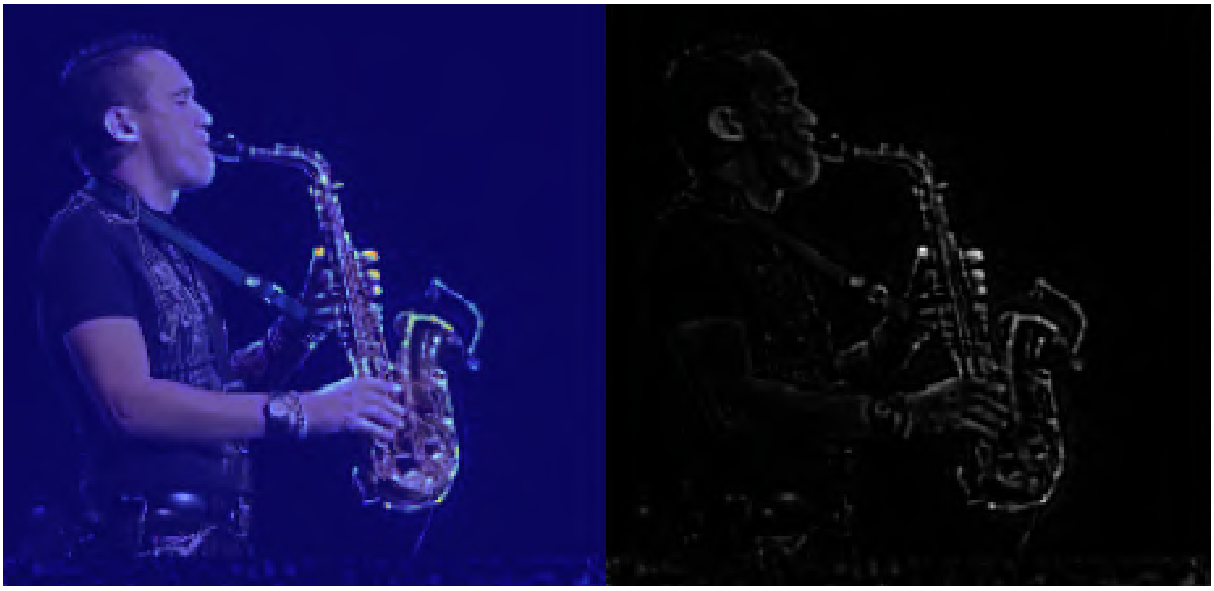


Figure 3.3: An example of heatmap explanation using Guided-backpropagation (Springenberg et al., 2015). Left: superimposed importance heatmap over an input image to a DNN. The different colours indicate whether a particular pixel provides evidence in favour of, or against a particular decision. Right: the absolute values of the heatmap pixels, demonstrating the magnitude of the importance of each pixel. Image reproduced from (Grün et al., 2016).

importance heatmap, also called class-saliency heatmap, sensitivity map, saliency map, or pixel attribution map (Smilkov et al., 2017), is a popular technique for communicating the contribution of each pixel to a model's final decision (Samek et al., 2017; Grün et al., 2016; Simonyan, Vedaldi, and Zisserman, 2013; Zeiler and Fergus, 2014; Bach et al., 2015; Li et al., 2015). Figure 3.3 shows an example of an importance heatmap. Finally, **architecture visualisation** techniques depict the information flow from the input to the output in terms of relevant neurons and the properties of these neurons. As such these techniques are part of the network-specific category.

We argue that a high level of comprehensibility about the model's behaviour requires the use of all three forms of explanation presentation. Existing approaches predominantly rely on only one presentation medium at a time, as we will demonstrate in the following sections. However, we argue that explanations may be enhanced when multiple mediums are combined. Hence, Chapter 4 elucidates the limitations of importance-based explanations using dimensionality reduction and functional description explanations. Additionally, in Chapter 5 we combine importance-based explanations, heatmap visualisations, and architecture visualisations to illustrate the limitations of relying on a single presentation medium to provide explanations. Finally, in Chapter 6, we combine mathematical explanations with architecture visualisations to describe the model behaviour.

Next, we look at importance-based explanations in more details, whereas in Section 3.6 we look at some of the visual explanations in the context of hidden representation analysis.

3.5 Importance- / Contribution-based Explanations

The most popular family of approaches for interpretability in practise are importance-based explanations (Bhatt et al., 2020). Importance-based explanations may be divided into three main categories: feature importance (Landecker et al., 2013), sample importance, and hybrids (e.g., case-based reasoning, which we discussed in Section 3.4.1). In Chapter 4, we illustrate that despite their ubiquitous application (Bhatt et al., 2020), feature importance explanations should not be used to assess the fairness of a model. Hence, for brevity we mention sample importance in Section 3.5.1. Then Section 3.5.2 describes the different families of feature importance explanations methods that we evaluate in Chapter 4.

3.5.1 Sample Importance

Sample importance explanations are type of importance based explanations that indicate the influence of different training points to the final decision. The best example of this type of explanation is the k-Nearest Neighbours algorithm (KNN) (Fix and Hodges Jr, 1951). Post-hoc model-based version of sample importance methods include influence functions (Koh and Liang, 2017), influential samples (Anirudh et al., 2017), representer points (Yeh et al., 2018). These methods provide rankings of examples that most positively and negatively influenced the decisions.

Sample importance explanations are useful for machine learning engineers to fine-tune and debug the system and guide future data acquisition efforts. However, depending on the privacy context, it might not be appropriate to share sample importance explanations with the end-users.

Remark 3.5.1

Importance can only be an absolute value. For example, a feature is important or not. On the other hand, the contribution, or attribution, describes how much the feature is contributing positively or negatively towards the output. Positive implies that increasing the feature will increase the likelihood of the outcome, while negative contribution implies the opposite (Samek et al., 2019).

3.5.2 Feature Importance / Contribution

Feature importance methods provide scores for a given data point that show the contribution of each feature (e.g., pixel, patch, word vector) of the input to the algorithm's decision. Several taxonomies for feature importance explanations have been developed (Ancona et al., 2019; Fong and Vedaldi, 2019; Samek et al., 2017; Grün et al., 2016; Adadi and Berrada, 2018). Although numerous terms have been used to describe the category of feature-importance explanations, such as sensitivity analysis, saliency-based, attribution methods, backpropagation-based, deconvolutional, or gradient-based (Samek et al., 2017; Grün et al., 2016; Adadi and Berrada, 2018), the majority of feature importance explanations can be described with a single equation)[8] (Ancona et al., 2018; Lundberg and Lee, 2017). Hence, we propose to categorise these taxonomies in terms of the properties they describe: (1) mathematical properties; (2) produced information properties.

On the basis of the mathematical formulation, feature importance explanations may be divided into two groups (Fong and Vedaldi, 2019): (1) **gradient-based**, (2) **perturbation-based**. Perturbation-based approaches apply discrete alterations to the feature values to estimate the contribution of each feature. In contrast, gradient-based approaches rely on gradient information, so they can be seen as the local infinitesimally small version of perturbation-based approaches. Due to this subtle difference Ancona et al. (2019) proposes another system of classification, which distinguishes between the type of information that feature importance methods produce: (1) **sensitivity analysis**, (2) **salience**.

Sensitivity analysis describes how the output changes due to infinitesimally small perturbations in one or more input variables. Since these methods approximate the first-order Taylor expansion, they are only accurate within infinitesimal small neighbourhoods around a target point. In contrast, the salience measures the *marginal* effect of each feature to the output with respect to a particular reference point. That is, the explanation describes the change in the outcome that follows from removing or changing one particular feature to a different value (Ancona et al., 2019). Since salience measures the marginal effect, the sum of the contributions of each feature need to sum to one. In other words, sensitivity analysis describes the magnitude and direction of the change in the prediction within very local neighbourhoods, whereas salience methods describe the contribution of a significant feature change to the output.

Essentially, gradient-based methods provide sensitivity analysis, while perturbation-based approaches measure the salience. In Chapter 4, we propose a method that can mask the underlying importance of a feature from both gradient-based and perturbation-based methods.

3.5.2.1 Gradient-based methods

Gradient-based methods evaluate the gradient of the DNN output with respect to the features at a particular point (Samek et al., 2017; Grün et al., 2016; Simonyan, Vedaldi, and Zisserman, 2013; Zeiler and Fergus, 2014; Bach et al., 2015; Li et al., 2015). Gradient-based methods may be categorised into two groups (1.1) functional and (1.2) topological.

[8] In Chapter 5.2, equation 5.1 presents this unifying equation in more detail.

Functional Functional methods treat the model as a black-box function. Hence, they explain the relationship the model has learned between inputs and outputs in the form of individual or group of *samples* or *features* (Zeiler and Fergus, 2014; Simonyan, Vedaldi, and Zisserman, 2013; Zintgraf et al., 2017; Ribeiro, Singh, and Guestrin, 2016). Examples of functional methods include Sensitivity Analysis (Simonyan, Vedaldi, and Zisserman, 2013; Zurada, Malinowski, and Cloete, 1994), SmoothGrad (Smilkov et al., 2017)[9], Gradients × Input (Shrikumar et al., 2016), and Integrated Gradients (Sundararajan, Taly, and Yan, 2017).

Topological On the other hand, topological, or contribution propagation, methods are network-specific approaches that treat the model as a graph and redistribute the effect of the lower layers on the output in a layer-by-layer fashion (Landecker et al., 2013; Bach et al., 2015; Montavon et al., 2017). The basic idea is to traverse the network in a layer-by-layer fashion and compute a relevance, or importance, score of each neuron (Landecker et al., 2013; Bach et al., 2015; Montavon et al., 2017; Ancona et al., 2019). Examples of this approach include contribution propagation (Landecker et al., 2013), Layer-wise-relevance propagation (LRP) (Bach et al., 2015), Deep Taylor Decomposition (Montavon et al., 2017), Excitation Propagation (Zhang et al., 2016), Guided-Backpropagation (Springenberg et al., 2015), Grad-CAM (Selvaraju et al., 2016), DeepLift (Shrikumar, Greenside, and Kundaje, 2017), PatternNet (Kindermans et al., 2017), and Pattern Atttribution (Kindermans et al., 2017). Since a weight of zero can be used to represent missing or blocked connection, the contribution propagation approach is usually (with the exception of Excitation Propagation and Guided-backpropagation, which are defined explicitly for ReLU activations) applicable to any architecture (e.g., fully-connected, convolutional, recurrent).

Contribution propagation is comparable to the DGINN approach, which we propose in Chapter 5, in that we also propagate the contributions. Contrary to the majority of propagation contribution methods, we do not impose the constraint that each neuron has to have a contribution. Approaches that do not distribute contribution to every neuron, such as our approach, Excitation Propagation (Zhang et al., 2016), and PatternNet (Kindermans et al., 2017) fall under the sub-category of topological approaches called **constrained redistribution**.

The premises behind constrained redistribution approaches are the sparsity and manifold assumptions, which mandate that very few neurons participate in the representation of each factor of variation. While Excitation Backpropagation uses a probabilistic winner-take-all sampling across the neurons that is limited to ReLU activations and positive weight connections between adjacent layers, we use outlier analysis to select multiple relevant neurons that allows for various activation functions and parameter settings. PatternNet

[9] Some authors consider SmoothGrad both a gradient-based and perturbation-based approach because it samples points in the neighbourhood of the target point to approximate the gradient.

extracts a denoised signal from the data based on the covariance for each neuron called pattern, and exchanges the neuron weights for the pattern weights.

3.5.2.2 Perturbation-based methods

Perturbation-based techniques apply discrete modifications to each feature to measure its contribution. We propose that these methods may be divided into two sub-groups: (2.1) surrogate models and (2.2) ablation-based. Surrogate models approximate the output of a black-box model with a linear (or higher comprehensibility) classifier on datapoints sampled in the local neighbourhood of the target point. Ablation-based approaches remove or mask a feature at a particular point to measure the feature's contribution. A certain subset of surrogate models can be categorised under contrastive and counterfactual explanations. These types of explanations answer the question of why "x" and not "y" and describe the minimum changes of the features that would have lead to this different outcome. Finally, perturbation-based methods must follow well-prescribed axioms since they describe the marginal contribution of each feature.

Surrogate models The best example of surrogate models is the Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin, 2016) approach. LIME samples points within the neighbourhood of the target point and trains a linear model to approximate the original model's output. The challenge with linear approximations is that they only provide relative feature importance. Hence, the set of sufficiently important features is not clear (Kim et al., 2018). For this reason, Anchors (Ribeiro, Singh, and Guestrin, 2018) and Local foil trees (Waa et al., 2018) approximate the model with if-then rule lists and one-versus-all decision trees respectively, which describe more fully all the sufficiently important features.

The main benefit of surrogate model approaches is that they are model-agnostic and do not require access to the model. The major drawback of surrogate models is their exceedingly high computational cost to explain just a single point[10].

Remark 3.5.2

It is noteworthy that the contribution of a variable sometimes depends on interactions with other variables. For instance, in the AND problem, we would need to perturb both features (e.g., $x_1 = 0, x_2 = 0$) to observe their influence (Robnik-Šikonja and Kononenko, 2008). Still, this does not provide a higher level of explainability (level 2) since it only measure the importance rather than communicating the dependence between the features.

[10] The computation takes several minutes per datapoint for a GoogleNet model (Lapuschkin, 2019).

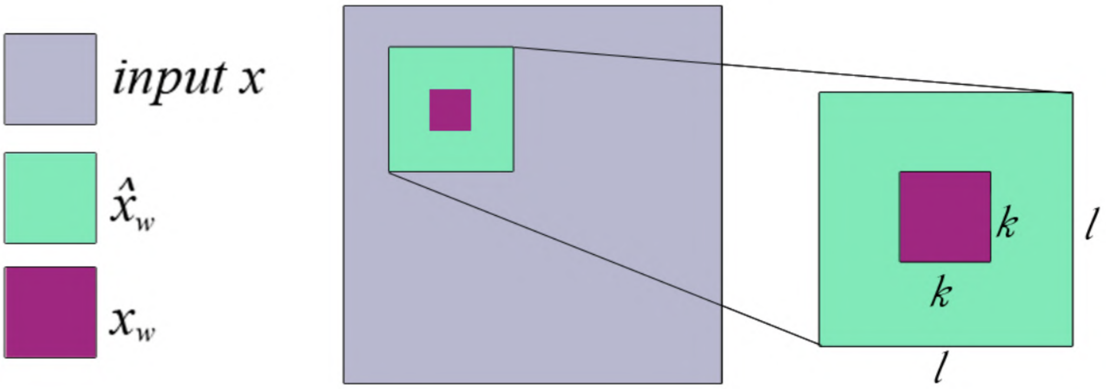


Figure 3.4: Illustration of the occlusion principle in prediction difference analysis. For an input image x, a patch x_w of size k is masked, where the mask is conditioned on the neighbourhood of the patch $\hat{x_w}$ with size $l > k$. Image reproduced from (Zintgraf et al., 2017).

Contrastive and Counterfactual Explanations The outputs of Anchors and Local foil trees (Ribeiro, Singh, and Guestrin, 2018) are a type of counterfactual explanations because they describe all the minimum changes that would have lead to a different outcome (Wachter, Mittelstadt, and Russell, 2017; Adadi and Berrada, 2018; Lipton, 1990; Hendricks et al., 2018). This set of minimum changes is known as contrastive perturbations (Dhurandhar et al., 2018). Counterfactual explanations answer the question: "Why this output (the fact) instead of another (the foil)" (Waa et al., 2018; Miller, 2019). Feature importance explanations answer this question with an answer to another question: Which feature and by how much do we need to change to affect the outcome?

On the other hand, **contrastive** explanations describe not only the minimal and sufficient features that need to be present, but also the minimal and necessarily absent features (Dhurandhar et al., 2018). Examples of contrastive approaches include (Wachter, Mittelstadt, and Russell, 2017), Contrastive Explanation Method (CEM) (Dhurandhar et al., 2018), and Model Agnostic Contrastive Explanations (MACEM) (Dhurandhar et al., 2020), which uses an optimisation procedure to adversarial samples that describe the set of contrastive perturbations.

Ablation-based Ablation-based methods rely on the principle of occlusion, which removes inputs (or patches of pixels in the case of images) to measure the change in the prediction. The idea is that irrelevant parts results in relatively smaller prediction differences (Robnik-Šikonja and Kononenko, 2008; Zeiler and Fergus, 2014; Goyal et al., 2016; Grün et al., 2016).

One of the most theoretically sound approaches in this vein of work are **Shapley values**, which fulfil all feature importance axioms that we describe later (Shapley, 1953; Strumbelj and Kononenko, 2010). Shapley values are a game-theory approach that computes the contribution of every possible feature combination, which gives a theoretical guarantee that all feature interactions have been accounted for (Shapley, 1953). IME (Štrumbelj, Kononenko, and Šikonja, 2009; Strumbelj and Kononenko, 2010) is an approach that uses Shapley values to compute all possible feature combination sets. Unfortunately, this approach is not feasible for high-dimensional data such as images.

For this reasons, different approximations have been developed that approximate Shapley values directly (e.g., SHAP (Lundberg and Lee, 2017)) or that approximate the possible feature combinations with various sampling techniques. The most challenging part is determining the masking procedure. For example, Occlusion masks image patches with constant pixel values (Zeiler and Fergus, 2014) or randomised pixel values (Zhou et al., 2014), whereas prediction difference analysis (Zintgraf et al., 2017) masks image patches with conditional multivariate sampling (see Figure 3.4). Other examples of perturbation methods change one-variable-at-a-time (e.g., EXPLAIN (Robnik-Šikonja and Kononenko, 2008), leave-one-covariate-out (LOCO) (Lei et al., 2018), and Feedback (Cao et al., 2015)).

One variable at a time approaches are simple and computationally cheap, however, these benefits come at the price of missing feature interactions.

Remark 3.5.3

SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017) assumes feature independence and model linearity, which are the main benefits that stem from the full Shapley computation. In Chapter 4 we demonstrate that due to these assumptions, SHAP is equally fragile as other gradient-based methods to model perturbations.

Axioms Since salience methods measure the marginal contribution of each feature with respect to a reference point, they must abide by specific requirements:

A salience explanation approach needs to fulfil the following properties:

- **sensitive**: a feature is assigned importance if there exists a perturbation of this feature, which affects the output (Sundararajan, Taly, and Yan, 2017),
- **additive**: the feature importance values should sum to the total change in prediction (also referred to as conservation axiom, or summation to delta) (Lundberg and Lee, 2017; Shrikumar, Greenside, and Kundaje, 2017),
- **locally faithful**: the explanation accurately describes changes in the output within the neighbourhood of the target point (also referred to as continuity)[11] (Lundberg

[11] We can think of this property as the analogy of adversarial examples for explanation methods. Assuming an explainer g, then $g(x) \approx g(x + \epsilon)$.

and Lee, 2017; Montavon, Samek, and Müller, 2018),

- **symmetry**: features containing identical information are assigned equal contribution (Lundberg and Lee, 2017),
- **null**: features that do not contain information are not attributed any value (also known as the dummy axiom) (Strumbelj and Kononenko, 2010; Lundberg and Lee, 2017),
- **consistency**: the explanation does not vary between different models (also referred to as implementation invariance) (Sundararajan, Taly, and Yan, 2017).

The benefit of the axiomatic approach is that we can theoretically study the properties of explanation techniques. For example, it can be demonstrated theoretically Gradient × input is not locally faithful, while LRP is (Montavon, 2019). In Chapter 4, we demonstrate that both gradient-based and perturbation-based approaches do not fulfil the consistency requirement.

3.5.2.3 Limitations

Despite their ubiquitous application and significant contribution to the field, feature importance explanations remain lacking in three main aspects: methodological, adversarial, and cognitive.

Methodological fragility It has been demonstrated that many gradient-based explanations do not change when the predictions change (Adebayo et al., 2018). One possible explanation for this finding comes from the fact that Guided backpropagation and Deconvolution conduct partial image recovery, which is independent of the output. In fact, gradient-based methods are exponentially less sensitive or even independent of the parameters of later layers (Adebayo et al., 2018; "When Explanations Lie: Why Many Modified BP Attributions Fail").

One reasons for the decreased sensitivity of higher layers might be that lower layers seem to play a more important role in the decision making process (Raghu et al., 2017). Another possibility might be that DNNs with ReLUs have highly fluctuating partial derivatives (Smilkov et al., 2017). These violent oscillations are due to the fact that techniques that rely on functional gradient or simple Taylor decomposition are sensitive to noise in the derivatives and gradient shattering (i.e., the exponential increase of regions with network depth leads to highly varying and discontinuous gradient values) (Montavon, Samek, and Müller, 2018). This is one of the reasons why heatmaps produced with sensitivity analysis are noisy. Yet another possible reason could be that feature importance explanations predominantly describe only very local model behaviour (Jiang et al., 2018).

This scenario makes any conclusions about the generalisation of the decision or the explanation potentially useless.

Adversarial fragility Adversarial examples have been show to fool *classification* accuracy by perturbing data points (Szegedy et al., 2014). Later it was observed that many *explanation* methods are fragile with respect to small changes in the raw features of a data point, even if the classification is unaffected (Adebayo et al., 2018; Alvarez-Melis and Jaakkola, 2018; Kindermans et al., 2019; Alvarez-Melis and Jaakkola, 2018). Tiny adversarial perturbations to data inputs can be generated so that the classification remains unchanged, but the explanation returned is very different (Ghorbani, Abid, and Zou, 2019). The reason for this phenomenon seems to be an excessively large curvature (Dombrowski et al., 2019) [12].

In contrast, in Chapter 4 we do not perturb the data. Instead, we modify the *model* in order to manipulate the explanations of conventional saliency methods. In particular, we aim to modify the model so that for any given data point, multiple explanation methods will not show the sensitive feature as important - even if in fact it is. Very recently, some works explored similar ideas. Pruthi et al. (2019) examined how attention-based methods could be fooled. Jain and Wallace (2019) showed that "attention is not explanation"', demonstrating that attention maps could be manipulated after training without altering predictions. Heo, Joo, and Moon (2019) considered modifying vision models to control explanations. Slack et al. (2019) employed a 'scaffolding' construction specifically to fool a small subset of the methods we investigatted – Local Interpretable Model-Agnostic Explanations 'LIME' (Ribeiro, Singh, and Guestrin, 2016) and Shapley Values 'SHAP' (Lundberg and Lee, 2017) explanation methods.

Cognitive fragility Human experiments also demonstrate that feature importance explanations do not necessarily increase human understanding, trust, or ability to correct mistakes in a model (Poursabzi-Sangdeh et al., 2018; Kim et al., 2018). This is because humans are subject to different biases (Adebayo et al., 2018; Abdollahi and Nasraoui, 2018; Pohl and Pohl, 2004):

1. **selection bias**: select or exclude certain sample when collecting data due to the remaining biases,

2. **confirmation bias**: search for reasons that validate initial believes and conclusions,

3. **implicit bias**: unconscious tendency to favour a particular sub-population,

4. **over-generalisation bias**: making general conclusions from small and overly specific sample sizes,

[12] See Appendix C.2.2 for more details.

5. **automation bias**: tendency to favour decisions from automated systems,

6. **reporting bias**: disclose only positive rather than negative results.

Because of the automation, selection, and confirmation bias people would accept "sensible" explanation for models producing random outputs (Adebayo et al., 2018). Additionally, feature importance explanations provide only relative importance of features, which does not communicate any information about feature interactions. This limitation prevents them from describing model behaviour at levels of explainability higher than level 1. Feature importance explanations often lead to over-generalisation bias of extrapolating overall model behaviour based on explanations about a single instance. In order to resolve this issue, in Chapter 5 we propose semi-local explanations that report model behaviour across a wider range of samples. Furthermore, Chapter 6 introduces a framework for both local and global explanations.

3.6 Representation Analysis

Representation analysis, or hidden space analysis, aims to increase the transparency of the latent representations in DNNs. A variety of techniques has been developed to analyse and visualise the *hidden representations* of DNN models in relation to their *inputs* or *output labels* (Alain and Bengio, 2017; Montavon, Braun, and Müller, 2011; Duch, 2003; Tenenbaum, De Silva, and Langford, 2000; Tenenbaum and Freeman, 1997). In contrast, in Chapter 6 we study the relationship between hidden representations and *concepts*, showing that representations gradually build sensitivity to relevant concepts and invariance to irrelevant concepts (see Section 6). Similarly, an invertible generative model can be trained to learn an intermediate representation that could translate between the latent space and a more human-interpretable space (Adel, Ghahramani, and Weller, 2018). The rest of this section describes techniques for interpreting the hidden space using the input or output space.

3.6.1 Dimensionality Reduction (output space)

Projections of the hidden space in 2D (dimensionality reduction) (Duch, 2003; Tenenbaum, De Silva, and Langford, 2000; Tenenbaum and Freeman, 1997) or visualisations of data point perturbation trajectories (Cantareira, Paulovich, and Etemad, 2020) have been applied to study the learning process, layer transformations, and regularisation effects in relation to the output. For instance, word embeddings project hidden representations of an RNN language model to demonstrate that these representations define rich semantic relationships (Mikolov, Yih, and Zweig, 2013a). Word embeddings are one of the most widely used methods for model validation and hidden layer semantic exploration in NLP (Li et al., 2015; Rauber et al., 2017; Donahue et al., 2014; Mnih et al., 2015). The same approach can be used in reinforcement learning to map states to sub-manifolds of the hidden space (Zahavy, Ben-Zrihem, and Mannor, 2016). This analysis can be taken further, as in Alain and Bengio (2017) and Montavon, Braun, and Müller (2011), to train linear classifiers that **predict the output labels** from each hidden layer or from a kernel PCA projection of the layer, respectively. This is useful when it is important to understand which parts of the hidden representation are pertinent to the decision making process. In contrast, CME (presented in Chapter 6) trains classifiers to extract concepts from the hidden representations before mapping these concepts to the output. In this way, CME adds an additional layer of interpretation that is more natural to comprehend.

3.6.2 Component Visualisation (input space)

Component visualisation is a popular approach for DNN interpretability, which provides some intuition about the decision making process. The drawback is that the input space needs to be intuitively comprehensible, as in the case of images, which is not always the case for complicated domains such as drug discovery.

Activation maximisation Activation maximisation (Erhan et al., 2009) treats the explainability of DNNs as an optimisation problem and synthesises the optimal input (usually image) that maximally excites (i.e., activates) a hidden unit. The technique is to start from random noise and use the derivative of the neuron activation with respect to every raw feature (in the case of images, these are pixel values) to find the optimal synthetic image. This synthesised image is the preferred input stimulus for the target unit, and it therefore may be the case that the image describes what the hidden unit represents. A variety of methods have been proposed to improve the quality of the synthesised image using different regularisation schemes such as adversarial examples (Szegedy et al., 2014), total variation (Mahendran and Vedaldi, 2015), blurring (Nguyen, Yosinski, and Clune, 2015), jitter and scaling Mordvintsev, Olah, and Tyka, 2015, bilateral filters (Tyka, 2016), GANs (Nguyen et al., 2016) or denosing autoencoders (Nguyen et al., 2017). Other techniques reveal different aspects about the multifaceted nature of neurons s (Nguyen, Yosinski, and Clune, 2016; Mahendran and Vedaldi, 2015).

Inversion Naturally, a single neuron within the hidden layers might not contain all the relevant information. Hence, we can gain additional insight by looking at the entire layer. Maximising all neurons within a layer would not produce anything sensible, therefore, representation inversion, or code inversion (Williams, 1986; Mahendran and Vedaldi, 2015), finds an image which *sets the neuron activations at particular values*, corresponding to a target input. In that sense, activation maximisation is a global method, while inversion

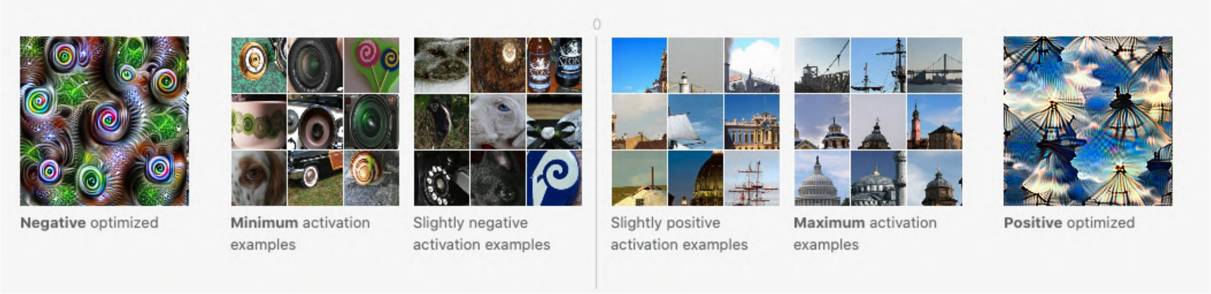

0
Negative optimized
Minimum activation examples
Slightly negative activation examples
Slightly positive activation examples
Maximum activation examples
Positive optimized

Figure 3.5: Illustration of unit-wise visualisation for InceptionV1's (Szegedy et al., 2015) layer mixed 4a, unit 492. Figure reproduced from (Olah, Mordvintsev, and Schubert, 2017). In addition to using activation maximisation to interpret a neural network, we can search for input samples, to which the activation of the neuron is highest or lowest (i.e., the images, or words (Hermans and Schrauwen, 2013; Karpathy, Johnson, and Li, 2015), which maximally "activate" or "deactive" a neuron (Olah, Mordvintsev, and Schubert, 2017). The "activation examples" panels demonstrate the idea.

is a local method. On the other hand, activation maximisation is a unit-wise approach, whereas inversion is a layer-wise one. In contrast, DGINN and CME (which we introduce in Chapters 5 & 6) are both unit interactions approaches and offer semi-local explanations or both local and global explanations, respectively.

Advantages and Disadvantages The benefit of the component visualisation approach is that a developer can gain an intuition behind the types of features that a DNN is picking up. For example, we can learn about local units (i.e., a unit that is sufficient to describe a factor of variation), dataset deficiencies, and the robustness of representations across the layers (Yosinski et al., 2015; Karpathy, Johnson, and Li, 2015). A drawback is that many non-technical users do find these explanations subjective or not completely informative. Another limitation is that the approach is more suitable for domains which are readily interpretable, such as images. The approach is hardly applicable to domains involving multi-dimensional inputs such as DNA sequencing or drug discovery, in which even expert users do not have a highly-developed intuition. In contrast, concept-based model extraction provides more formal explanations in the form of rules that describe explicitly the model behaviour across a wide range of inputs.

Neuron Importance The vast number of neurons makes component visualisation approaches infeasible for manual human inspection. For this reason, it is sometimes useful to identify only the relevant for inspection neurons using ablation experiments (Girshick et al., 2014), transfer learning (Yosinski et al., 2014), or attribution flow through each neuron, termed conductance (Dhamdhere, Sundararajan, and Yan, 2018). Similarly, in Chapter 5 we propose the DGINN framework for measuring the importance of neurons. Our results are complementary with a phenomenon discovered in parallel – the lottery ticket hypothesis (LT) (Frankle and Carbin, 2018), which determines the importance of a neuron by setting all weights with negligible contribution to zero and retraining the network. Both DGINN and LT suggest the existence of specialised sub-networks within a larger DNN. The lottery ticket hypothesis suggests that DNNs contain subnetworks, which when trained in isolation achieve comparable accuracy to the entire network. In contrast, DGINN demonstrates that there are different sub-networks, which without retraining are biased towards particular classes.

3.6.3 Architecture Visualisation

Architecture visualisation approaches visualise the network's topology and augment it with additional techniques from component visualisations to increase model transparency. For example, several works have been developed in parallel with DGINN (which we introduce in Chapter 5) such as trees of relevant neurons (Zhang and Zhu, 2018) or directed acyclic graphs (DAGs) (Liu et al., 2017). While the former use contribution propagation to determine the relevant neurons, the latter rely on activation clustering (Liu et al., 2017). DGINN combines the benefits of both approaches. First, it leverages the *multi-clustering* assumption from representation learning. That is a neuron can participate in more than one partially distributed representation. Hence, we use a graph rather than a tree. Second, it applies statistical outlier analysis only to the activations of the relevant neurons to *select only the most pertinent* paths for investigation.

3.7 Conclusion

In this chapter, we proposed to categories existing work on interpretability based on seven not-mutually exclusive groups. Namely, each of the groups describes a spectrum on the basis of: (1) the focus of the explanation (data-based vs model-based), (2) the stage of development (intrinsic vs extrinsic), (3) the families of algorithms model-based explanations can be applied to (model-agnostic, model-specific, network-agnostic, network-specific), (4) the entity of interpretation of the internal workings for network-specific approaches (neuron, layer), (5) the scope of the information (local, semi-local, global), (6) the domain and range of the explanation function (input space, output space, hidden space, and concept space), (7) the presentation of the explanation (importance, mathematical, visual).

We argue that the majority of the existing effort dedicated to interpretability focuses exclusively on the extremes of the spectra proposed in our taxonomy. That is, current methods look at explanation techniques primarily as an either-or instances of our categories, which leads to four main limitations. First, there are numerous methods for local (e.g., feature importance) (Simonyan, Vedaldi, and Zisserman, 2013; Zeiler and Fergus, 2014; Zintgraf et al., 2017; Shrikumar, Greenside, and Kundaje, 2017; Landecker et al., 2013;

Bach et al., 2015; Montavon et al., 2017) or global (e.g., activation maximisation (Erhan et al., 2009) explainability, but there is little published data on semi-local (e.g., class-specific, or concept) explanations. Second, far too little attention has been paid to the fact that DNNs use sparse, or partially distributed representations. Consequently, network-specific methods have been primarily unit-wise (Erhan et al., 2009; Girshick et al., 2014; Goodfellow et al., 2009) or layer-wise (Girshick et al., 2014; Yosinski et al., 2014; Williams, 1986; Mahendran and Vedaldi, 2015; Alain and Bengio, 2017). However, recent studies suggest that all neurons are not equal, neither are all layers (Raghu et al., 2017; Frankle and Carbin, 2018; Andreas, 2019; Do and Tran, 2020; Kornblith et al., 2019; Zhang, Bengio, and Singer, 2019). Hence, future approaches should focus on partially-distributed representations at key layers. In Chapters 5 & 6, we illustrate that this strategy leads to considerable improvements to the semantic level of explanations. Third, the majority of explainability methods describe the input-output relationship (e.g., importance-based explanations, component visualisation, model extraction, and representation analysis). At the same time very little attention has been paid to the role of the high-level semantic units such as concept explanations. Fourth, due to the unreliability of feature importance methods, and the low comprehensibility of activation maximisation approaches, there has been a deluge of methods that focus on improving the quality of these methods rather than increasing the level of explainability (see Section 3.3.2).

We demonstrate additional limitations of feature importance methods in Chapter 4. Therefore, we propose class-specific and mathematical concept-based explanations that are extracted from groups of neurons within relevant layers in Chapters 5 & 6. Concept-based explanations move us to level 4 explainability, in which the role of feature interactions and their relationship to the outcome are more readily understandable.

CHAPTER 4

Adversarial model perturbations to manipulate explanations

What gets us into trouble is not what we don't know. It's what we know for sure that just ain't so.

Mark Twain

In Chapter 3, we introduced some of the limitations of feature importance explanations. Here, we expand this discussion and focus on the first research question of this thesis, namely evaluating the fidelity of feature importance explanations. Specifically, we investigate the ability of feature importance methods to provide reliable information about the fairness of a model. Fairness is part of a larger research agenda of building models that are 'Fair, Accountable, and Transparent' (Diakopoulos et al., 2017; Weller, 2019). Fairness is a key concern in many application areas including selecting candidates for hire, approving loans in banking, and selecting recipients of organ donations.

Transparency has emerged as a way to aid our understanding of the inner workings of a machine learning model and ensure model fairness. In practice, the most popular family of approaches for transparency are feature importance[1], or saliency, methods (Bhatt et al., 2020). It has been common to suggest that such saliency methods can be used to inspect a model for fairness as follows. We observe if a model's outputs depend significantly on a protected feature such as gender or race, which are termed *sensitive*. When there is a high dependence on a sensitive attribute then the model appears to be unfair.

In this chapter, we show that *the apparent importance of a sensitive feature does not reliably reveal anything about the fairness of a model.* We explain how this can happen with an instructive example demonstrating that a model could have arbitrarily high levels of

[1] See Section 3.5.2.

unfairness across a range of popular metrics, even while appearing to have zero dependence on the relevant sensitive feature. We introduce a practical approach to modify an existing model in order to downgrade the apparent importance of a sensitive feature according to explanation methods.

Specifically, we answer the following questions:

1. How badly can we fool fairness measures, as perceived by various importance-based explanation methods?
2. Are all fairness measures equally prone to fooling?
3. Can multiple importance-based methods be fooled simultaneously?
4. Are all explanation methods equally vulnerable?

While previous work has focused on a model's vulnerability to adversarially perturbed *input data* (Ghorbani, Abid, and Zou, 2019), and considered robust training with respect to the *input* to mitigate this susceptibility, here we show that the *model parameters* can be modified so as to lead to a desirable misleading explanation. Consequently, the insight that feature-importance is not useful for fairness would not be limited only to input perturbations, but to parameter modifications as well.

To the best of our knowledge, we are the first to focus on the fairness of a model concerning popular explanation methods. We published this work in collaboration with Umang Bhatt and Adrian Weller in Dimanov et al. (2020)(as the main contributor). Section 4.1 introduces the subject of fairness and it shows how unfairness can be arbitrarily high, despite no dependence on a sensitive feature. We describe our approach to modifying a model in order to hide unfairness in Section 4.2. Section 4.3 presents our evaluation methodology. Finally, in Section 4.4 we show empirically that our approach has little impact on a model's accuracy while being able to fool simultaneously seven popular feature-importance approaches to explanation (See Section 3.5.2).

Our observations raise serious concerns for organisations or regulators who hope to rely on feature importance interpretability methods to validate the fairness of models. For example, a malicious agent (e.g., bank) might conceal the unfairness of their models from regulators relying on feature importance explanation for their auditing. We focus here on deep learning models, but our ideas extend naturally to other model classes.

4.1 Fairness

A key question when examining whether an explanation method reliably reveals information about fairness of a model is whether or not in fact the model is fair. We assess the fairness using standard definitions from the literature (Beutel et al., 2017; Hardt, Price, and Srebro, 2016), used within the IBM AI Fairness 360 Toolkit (Bellamy et al., 2018):

1. Demographic Parity (DP): the predicted *positive rates* for both groups should be the same.

2. Equal Opportunity (EQ): the *true positive rates (TPR)* for both groups should the same.

3. Equal Accuracy (EA): the classifier accuracy for both groups should be the same.

4. Equal Odds (EO): the *true positive rates (TPR)* and the *true negative rates (TNR)* for both groups should the same.

5. Disparate Impact (DI): the ratio between the *positive rates* for the unprivileged and privileged groups.

6. Theil Index (TI): between-group unfairness based on generalised entropy indices (Speicher et al., 2018).

Note that it is typically not possible to satisfy many fairness notions simultaneously (Kleinberg, 2018).

How extreme could unfairness be, yet still be hidden? Let us first consider the limits of how unfair a model might be, yet still appear to be fair according to explanation methods. Worryingly, and perhaps surprisingly, we show that in fact a model can be arbitrarily unfair with respect to a feature, yet appear to have no sensitivity at all to the feature (i.e., low to no gradients in the direction of the feature).

Consider an arbitrary classification problem, shown in Figure 4.1. Each data point has two features: a continuous x_1 and a binary x_2. Let x_2 be a sensitive feature, such as age, given by the shape of the point: assume young and mature people. The true label y for each point is indicated by its colour: blue for positive and orange for negative.

The black curve indicates the model's softmax predicted label value $\hat{y}$ as a function of the features (x_1, x_2). If the function value is above 0.5, then the output is 1, else the output is 0; this is shown by the pale blue/orange boundary in the background colour. Further, assume the model does not vary in the direction of x_2 (hence it has 0 gradient).

Five data points are shown. The model makes only one classification mistake (the blue young person receives $\hat{y} = 0$ yet has $y = 1$). However, this model is highly unfair with respect to the sensitive feature for three metrics described in Section 4.1. Equal Opportunity is maximally violated: for young people, $0/1 = 0\%$ deserving points get the good (blue) outcome; for mature people, $2/2 = 100\%$ deserving points get the good (blue)

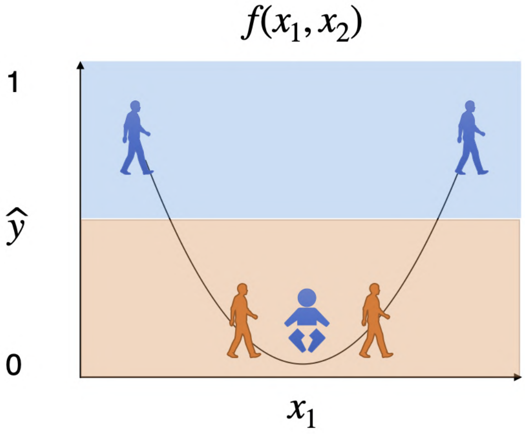


Figure 4.1: This example illustrates a function with no dependence on target feature yet extreme unfairness, showing the softmax predicted label $\hat{y}$ versus an input feature x_1, which is not the target feature. Each shape shown is a data point. The colour indicates the true label, i.e., blue means $y = 1$ and orange means $y = 0$. The shape shows the value of the target feature: young and mature people. The black curve shows a function mapping from features to estimated output label $\hat{y}$. Assume the function is constant across age. The blue young person is in the orange zone, whereas it should be in the blue zone (see Section 4.1). Best viewed in colour.

outcome. Equal Accuracy is also maximally violated: for young people, $0/1 = 0\%$ points are accurate (blue young person should be placed in the blue zone); for mature people, $4/4 = 100\%$ points are accurate (correctly, blue mature people are in the blue zone, orange mature people are in the orange zone).

Finally, consider demographic parity (DP): for young people, $0/1 = 0\%$ get the good outcome; for mature people, $2/4 = 50\%$ get the good outcome. Observe that if we keep adding more blue mature people data points near the ones already shown then the young people ratio stays unchanged while the mature people ratio tends to 1. Thus, we can obtain any arbitrarily high level of DP unfairness. Similar results can be derived for the other metrics. This demonstrates the extreme unfairness that could occur in a model. But how could this be achieved?

4.2 Method: Learning a Modified Model with Concealed Unfairness

The aim of our approach is to modify an existing model so that multiple explanation methods will not show a particular target feature as important without considerably affecting the accuracy of the model. Our approach retrains an existing model with a modified loss objective function: we add an "explanation loss" term to the original loss in the form of the gradient of the original loss with respect to a chosen target feature. Our attack method achieves three objectives:

1. We obtain a model with low local sensitivity to the chosen feature, yet with little loss of accuracy;
2. the low sensitivity generalises to unseen test points; and
3. low feature sensitivity leads to low attribution for the target feature across all seven feature importance explanation methods that we experimented with (see Section 4.4).

Let us now describe the method formally.

Notation We consider differentiable functions $f : \mathbb{R}^m \mapsto \mathbb{R}^d$; and a dataset of an input matrix in $\boldsymbol{X} \subseteq \mathbb{R}^{n\times m}$ with n samples and m features (attributes) and an output matrix $\boldsymbol{Y} \subseteq \mathbb{R}^{n\times d}$, where each row is a 1-hot encoded vector of d output classes. While our approach applies to arbitrary d, here we focus on the binary classification case of $d = 2$ corresponding to 'positive' and 'negative' output classes (e.g., receive a loan or not). Concretely, we focus on neural network functions $\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta})$ parameterised by $\boldsymbol{\theta}$, which we shorthand to f_θ. We write $\mathbf{x}^{(i)}$ for the input vector row i with m feature columns, and $\boldsymbol{X}_{:,j}$ for an entire feature j column vector.

We write g for a local feature explanation function which takes as input a model f and an input $\mathbf{x}$, and returns feature importance scores $g(f, \mathbf{x}) \in \mathbb{R}^m$, where $g(f, \mathbf{x})_j$ is the importance of (or attribution for) feature x_j for the model's prediction $f(\mathbf{x})$. We encode categorical features (e.g., male or female) as discrete values and normalise continuous variables in the range $[0, 1]$.

Formal Objectives Suppose we have trained a model f_θ with acceptable performance but with undesirably high target feature attribution. We would like to find a **modified classifier** $f_{\theta+\delta}$, with the following properties:

1. *Performance similarity:* e.g., the new model has similar accuracy

$$\forall i, \ \ f_{\theta+\delta}(\mathbf{x}^{(i)}) \approx f_\theta(\mathbf{x}^{(i)}).$$

2. *Low target feature attribution:* the importance of the target feature j (e.g., gender or race), as given by a chosen explanation method g, decreases significantly

$$\forall i, \ \ |g(f_{\theta+\delta}, \mathbf{x}^{(i)})_j| \ll |g(f_\theta, \mathbf{x}^{(i)})_j|.$$

Learning a Modified Model with Concealed Unfairness To manipulate the feature importance explanations, we begin with a pre-trained model and then modify it by optimising with an extra penalty term, *explanation loss*, weighted by a hyperparameter α, which is normalised over all n training points (full batch):

$$\mathcal{L}' = \mathcal{L} + \frac{\alpha}{n} \left|\left|\nabla_{\boldsymbol{X}_{:,j}} \mathcal{L}\right|\right|_p , \tag{4.1}$$

where j is the index of the target feature that we want to appear as the model is avoiding to use, and $\nabla_{\boldsymbol{X}_{:,j}} \mathcal{L}$ is the gradient vector of the original cross-entropy loss $\mathcal{L}$ with respect to the entire feature column vector $\boldsymbol{X}_{:,j}$. We apply the L^p norm.[2] We define a new objective that regularises for low derivative with respect to the target feature across the training points, and results in the modified classifier, $f_{\theta+\delta}$. We outline the procedure in Algorithm 1, where we used $\tau = 100$ iterations consistently since this was sufficient for convergence across runs. We ran hyper-parameter search (discussed in Section 4.3.1) to set $\alpha = 3$ for all experiments.

Algorithm 1 Adversarial Explanation Attack

```
Input: Original classifier $f_\theta$, target feature's index $i$, input matrix $\boldsymbol{X} \in R^{n \times m}$ with
corresponding targets $\boldsymbol{y} \in \mathbb{R}^d$, and number of iterations $\tau$.
Initialise $\boldsymbol{\delta} = \mathbf{0}$
for $t \in [0, \tau]$ iterations do
    Calculate the cross entropy loss $\mathcal{L}$ with respect to $f_{\theta+\delta}$
    Calculate the explanation loss
        $\zeta = \frac{1}{n} \times L^p \left( \left[ \left| \frac{\partial \mathcal{L}}{\boldsymbol{X}_{1,i}} \right|, \left| \frac{\partial \mathcal{L}}{\boldsymbol{X}_{2,i}} \right|, \ldots, \left| \frac{\partial \mathcal{L}}{\boldsymbol{X}_{n,i}} \right| \right] \right)$
    Calculate the total loss $\mathcal{L}' = \mathcal{L} + \alpha \times \zeta$ (equation 4.1)
    Update model parameters with $\nabla_{\boldsymbol{\theta}} \mathcal{L}'$ using Adam
end for
Output: Modified classifier $f_{\theta+\delta}$
```

Remark 4.2.1

In Appendix C.2.2, we clarify the difference between our approach for explanation loss and the recent method of Heo, Joo, and Moon (2019). While their approach takes the gradient of the one correct label element from the logits layer just before the softmax output, we take the gradient of the cross-entropy loss.

Taking the gradient of the loss, rather than only the correct label element, contains extra information about the other classes, with the potential to improve generalisation across explanation methods and test points.

[2] We use $p = 1$ since it led to rapid convergence and good results.

4.3 Evaluation

In this Section we describe the experimental set-up of our evaluation and define measurable evaluation criteria to assesses the objectives postulated in Section 4.2.

4.3.1 Experimental Set-up

Datasets Unless stated otherwise, we conduct experiments on four datasets with sensitive features – three from the UCI machine learning repository (Dua and Graff, 2017) adult (*Adult*) – gender, race; German credit (*German*) – age, gender; bank market (*Bank*) – age, marital; and the dataset for Correctional Offender Management Profiling for Alternative Sanctions (Larson et al., 2019) (*COMPAS*) – gender, race, age.

Models For each dataset we train 0-9 hidden layer multilayer perceptrons (MLPs) with 100 units in each layer, regularised with a layer-wise L^2-norm penalty weighted by 0.03 for up to 1,000 epochs with early stopping and patience of 100 epochs with 10 random initialisations. We use L^2-norm regularisation because we want to have as many parameters active as possible so that there would be more directions to manipulate. The penalty 0.03 was empirically validated to give the best validation accuracy. We use Tensorflow (Abadi et al., 2016) to conduct the original optimisation with Adam (Kingma and Ba, 2014), a global learning rate of 0.01 and 0.005 learning rate decay over each update and with full batch gradient descent. We conducted hyper-parameter optimisation to determine that optimisation with L^1-norm and $\alpha = 3$ converges slightly faster and to better configurations in terms of performance similarity and low feature attribution.

Feature Attribution Methods We evaluate seven popular feature attribution methods described in detail in Section 3.5.2:

1. **Gradients**: Sensitivity analysis gradients (Simonyan, Vedaldi, and Zisserman, 2013),
2. **Gradients × input** (Shrikumar et al., 2016),
3. **Integrated Gradients** (Sundararajan, Taly, and Yan, 2017),
4. **SHAP**: approximation of Shapley values (Lundberg and Lee, 2017) – Expected Gradients (Erion et al., 2019),
5. **LIME**: Local Interpretable Model-Agnostic Explanations (Ribeiro, Singh, and Guestrin, 2016),
6. **GB**: Guided-backpropagation (Springenberg et al., 2015), and

7. **Dependency Graphs** (Chapter 5).

A Dependency Graph resulting from the DGINN framework ran with the gradients relevance functions (introduced in Chapter 5) corresponds to a sub-graph of the DNN that contains only the relevant neurons for each decision. Since we can propagate the relevance of every neuron through the sub-graph, we can compute relevance scores for each of the input neurons. These relevance scores can be treated as attribution values, as explained in Chapter 5. We use the authors' repositories of SHAP and LIME and Ancona et al. (2018)'s implementation for the remaining methods. We conceal unfairness using the training data and report evaluations both on the training data, and on a test set that was not used neither for training the original model, nor for the modified model.

Fairness For the fairness evaluation, we use the implementation of IBM AI360 Toolkit (Bellamy et al., 2018). We consider model predictions for two primary sub-groups based on a sensitive feature, designating the sub-groups as privileged or unprivileged following (Bellamy et al., 2018), and binarise each sensitive features in the following fashion: Gender: Male - privileged, Female - unprivileged; Age: $25 >$ x privileged, $25 <$ x unprivileged; Race: White - privileged, Non-white - unprivileged; Martial status: Single - privileged, Not single - unprivileged. We evaluate across the six fairness metrics described in Section 4.1.

Hyper-parameter Investigation In all experiments, we use L^1-norm for equation 4.1, we minimise using Adam (Tieleman and Hinton, 2012), and $\alpha = 3$. These are careful design choices that we made after an empirical investigation, which we discuss next.

Explanation Loss Norm We observe that the L^1-norm converged slightly faster and to slightly better configurations both in terms of performance similarity and low target feature attribution metrics across different settings in comparison to both the L^2 and L^∞ norms.

We can develop some intuition about these results if we interpret the L^p as a regulariser of the explanations[3]. The backpropagated gradient of the L^1-norm is constant regardless of the norm's parameter value; hence, the feature importance explanations of the target feature ($|\frac{\partial \mathcal{L}}{\partial \boldsymbol{X}_{i,j}}|$) with magnitudes both much greater than and closer to 0 are equally penalised, resulting in "sparse explanations" (i.e., most of the explanations are 0 or close to 0). On the other hand, the backpropagated gradient of the L^2-norm is linear with the norm's parameter and penalises explanations with large magnitudes, but does not affect as much explanations with relatively small values.

The effect on explanations with relatively small values is even more pronounced for the L^∞-norm, where the backpropagated gradient is non-zero only for the highest explanation

[3] We look at more similarities to regularisation in Section D.

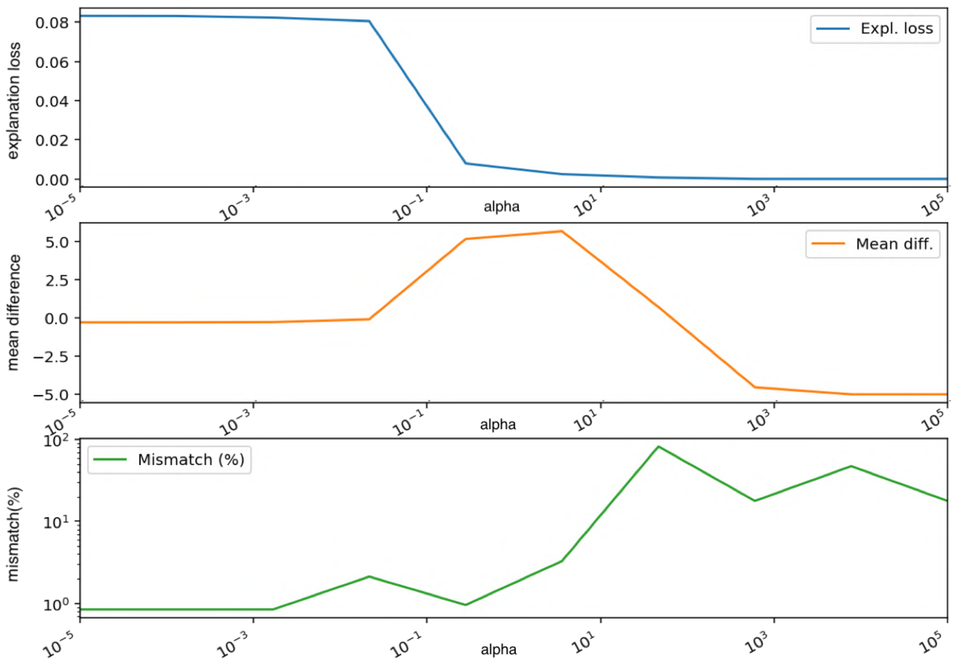


Figure 4.2: Illustration of the effects of $\alpha \in 10^{[-5,5]}$ (x-axis) on the performance similarity and low target feature attribution metrics (y-axis): (top) average explanation loss per sample (Expl. loss); (middle) the mean of the sensitive property importance ranking distribution (Mean diff.); and (bottom) the percentage difference between the two models' predictions (Mismatch). Notice that optimal α values lie in the range $[10^{-1}, 10^{1}]$.

value. Hence, training with L^{∞} norm resembles a single sample gradient descent and results in significantly slower convergence. Further, we observed that the choice of the explanation loss norm is strongly coupled with the value of the explanation penalty term α. All three norms converge to very similar configurations with the appropriate α. Since the L^2-norm over emphasises explanations with an extremely high value, it requires a lower α. In contrast, the L^{∞}-norm reflects the loss of a single example and requires an α of orders of magnitude higher than the L^1-norm. Taken together, these results suggest that L^1-norm is the optimum norm.

Explanation Penalty Term (α) Figure 4.2 demonstrates that the learning dynamics of the adversarial explanation attack vary with the explanation penalty term (α.) At one extreme, the penalty term α corresponds to unnoticeable changes in the explanation loss (see Figure 4.2(top)), while at the other extreme α corresponds to a catastrophic change that leads to a constant model which ignores all features and drastically changes the model predictions (see Figure 4.2 (bottom)). Within the optimum range ($\alpha \in [10^{-1}, 10^{1}]$), we can minimise the explanation loss significantly while keeping the model prediction dissimilarity relatively low. For these reasons, we recommend a value of $\alpha = 3$ and set it for all experiments.

Learning Rate We observed that parameter learning approaches could make a significant difference in the stability of the optimisation process. Similarly to regular training, adaptive learning rate algorithms achieve significantly better results. A vanilla-SGD optimisation is much more likely to converge to constant classifiers that predict the label distribution and requires bespoke learning rate scheduling routines similar to Smith (2018), where the learning rate is adopted dynamically based on the explanation loss. Specifically, every time the explanation loss (ζ) at epoch t goes above the previous explanation loss ($\zeta^t > \zeta^{t-1}$), we decay the learning rate based on the following step decay formula: $\eta^t = \eta^0 \times 0.9^{1+t}$.

4.3.2 Evaluation Criteria

Performance Similarity We consider the concealing procedure successful when both properties from Section 4.2 are satisfied. We measure **performance similarity** between the modified model and the original model through three metrics:

- **Loss diff.**: Difference between the categorical cross entropy losses ($\mathcal{L}$) of both models averaged over all test points.
- **Accuracy Change (Acc Δ)**: Difference in the accuracy of both models.
- **Mismatch (%)**: Difference in the output of the two models, as measured by the percentage of datapoints, where the predictions of the two models differ. This metric is a proxy for the fidelity with which the modified model approximates the performance of the original model.

Low Target Feature Attribution Measuring the effect of the concealing procedure on feature importance is more complex. We want to avoid the pathological case of the attack shrinking the importance of all features and inducing a random classifier. Therefore, we introduce four metrics based on relative feature importance on the ranking histograms[4], which describes the probability mass distribution of the target feature importance in comparison to the remaining features. We show a case where the initial model had a low target feature gradient, demonstrating that even in this case, the attack was successful. An effective attack shifts the distribution from left to right. We use five metrics to measure this distribution shift and assess the **low target feature attribution**:

- **Top k:** the number of datapoints where the sensitive feature received rank k or above.
- **Mode shift: (Avg. #shifts)** the difference between the modes of the distribution.
- **Mean shift:** the difference between the means.

[4] See Figure 4.3 for an example of a relative feature importance ranking histogram.

- **Highest rank:** the highest rank that the sensitive feature received across all datapoints.
- **Highest ranking datapoints (HRD):** the number of datapoints where the sensitive feature received the highest rank. This is the same as Top k, where $k =$ highest rank.

4.4 Results

This Section measures the degree to which the adversarial model explanation attack objectives (set out in Section 4.2) can be achieved across 4 datasets, 10 sensitive features, and 10 model architectures across 10 different initilisations.

4.4.1 Attack Evaluation

Figure 4.3 illustrates three important points. First, our method significantly decreases the relative importance of the target feature, effectively making it the least important of all features with little change in accuracy[5]. Second, the attack transfers across seven different explanation methods. Third, the attack generalises for unseen, held-out test datapoints.

Transferability Tables 4.1 and 4.2 illustrate that the explanation attack transfers across explanation methods. That is, the explanation loss is designed to decrease the gradient, which is essentially a targeted attack against the Gradients explanation method.

However, the attack transfers to both gradient-based and perturbation-based explanation methods and significantly decreases the importance for all investigated explanation methods. This finding suggests that we can simultaneously conceal the unfairness of a model from multiple explanation techniques.

Notice in Table 4.1 that in the case of the Adult dataset and gender target feature for all explanation methods, the attack has moved down the target feature importance out of the highest ranking features for thousands of data points, demonstrating that the attack works even when the target feature has high relative importance.

Generalisation The generalisation of the attack to test points is noteworthy since we might expect that the decision boundary would be perturbed locally around the training points, affecting only training point explanations, without significant change for test points, especially if far away in feature space. We investigate the hypothesis of strictly local changes to decision boundaries and other possible explanations for this result in Section 4.4.3.1.

[5] We explore different reasons why the accuracy does not drop in Appendix C.

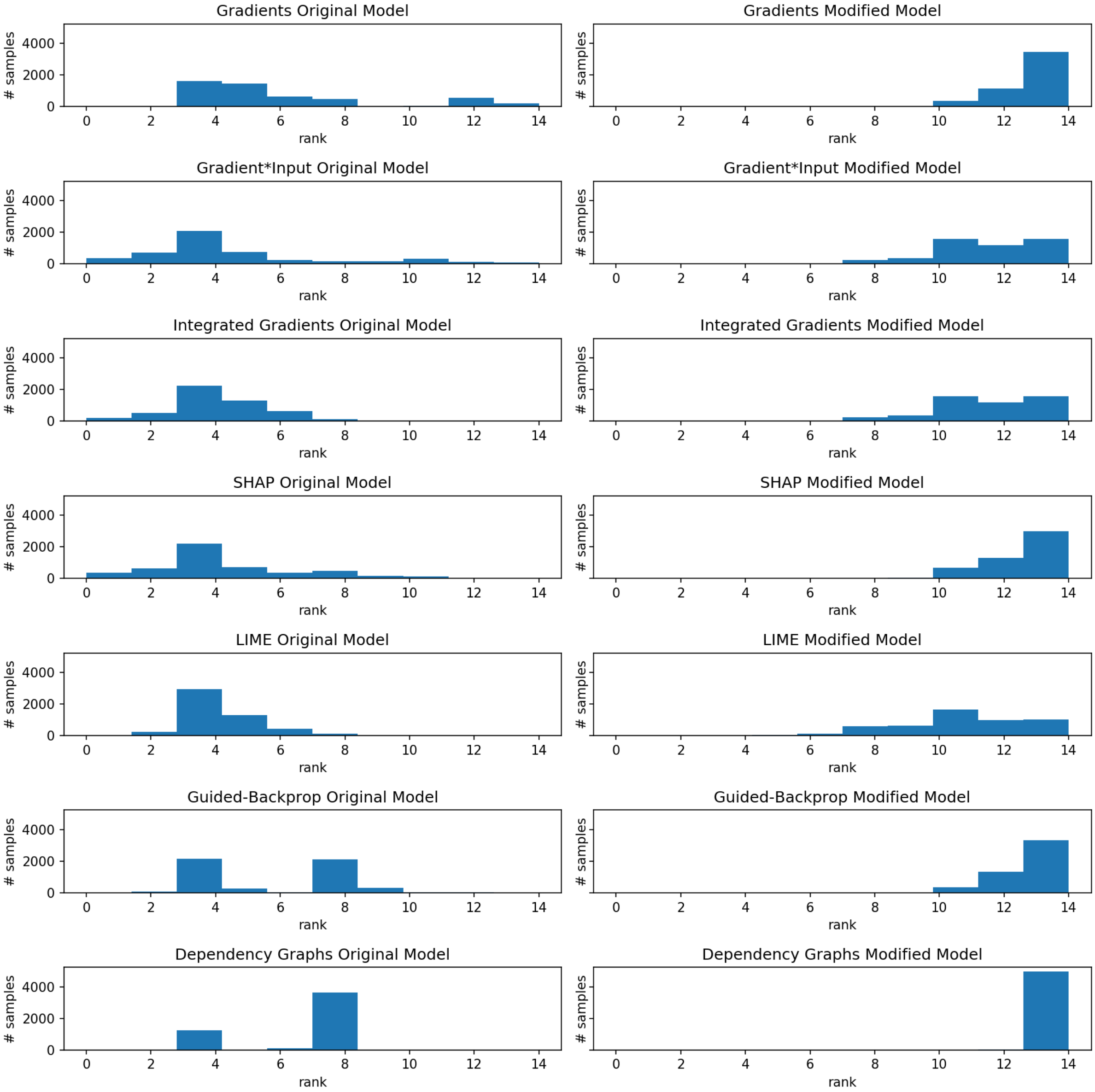


Figure 4.3: Importance ranking histograms for gender as the sensitive feature on the adult test set of the original (left) and modified (right) models. Each histogram represents the ranking across the test set assigned by the designated feature importance method. A *higher ranking number* (further to the right) indicates *smaller feature importance*. Observe that the **modified model** has successfully **shifted** the **ranking** for **all explanation methods**. At the same time the test accuracy between the two types of models has remained almost unchanged (differing at most between 2%-6%).

	Mode (O)	Mode (M)	# shifts	Mean (O)	Mean (M)	Mean Diff	Highest Rank(O)	Highest Rank(M)	HRD_O (O)	HRD_O (M)	Top-5 (O)	Top-5 (M)	Top-1 (O)	Top-1 (M)
Gradients	5	13	8	6.03	12.59	6.56	3	8	821	0	1599	0	0	0
Gradient*Input	4	13	9	4.64	11.39	6.75	0	5	29	0	3141	0	29	0
Integrated Gradients	4	13	9	4.08	11.39	7.31	0	5	38	0	2956	0	38	0
SHAP	3	13	10	4.23	12.40	8.17	0	7	119	0	3178	0	119	0
LIME	4	13	9	4.21	10.63	6.42	0	3	1	0	3162	17	1	0
Guided-Backprop	7	13	6	5.69	12.58	6.89	2	8	67	0	2202	0	0	0
Dependency Graphs	7	13	6	6.39	12.99	6.60	4	12	1241	0	1241	0	0	0

Table 4.1: Evaluation of performance similarity and low feature attribution after an adversarial explanation attack for seven explanation methods on Adult Gender Train ('O' is original model, 'M' is modified model). Notice that after our attack the mode and mean ranking of the sensitive feature have shifted significantly ("# shifts"). For nearly all datapoints, the sensitive feature moves out of the top five most important features ("Top-5 (M)"). The results are averaged over 10 random initialisation of a 5 hidden-layer model.

		Train ζ (10^{-2})	Test ζ (10^{-2})	Train Acc Δ	Test Acc Δ	Tr. Mis (%)	Ts. Mis (%)
Dataset	Feature						
adult	age	9.79±3.61	9.82±3.59	-2.76±1.03	-3.07±1.16	10.88±1.67	10.72±1.66
	gender	11.03±3.36	11.11±3.38	-2.43±0.86	-2.71±0.94	10.37±2.44	10.29±2.49
	race	10.1±2.75	10.18±2.76	-2.47±0.85	-2.78±0.9	10.24±1.31	10.37±1.35
bank	age	**12.79±4.12**	**13.39±4.17**	-1.81±0.35	**-2.23±0.4**	7.35±0.73	7.5±0.75
	marital	12.5±5.26	12.96±5.46	-1.73±0.34	-2.27±0.4	**7.25±0.71**	**7.43±0.7**
compas	age	4.0±1.69	4.34±1.82	-2.23±0.66	-3.2±0.91	**19.83±1.68**	**18.96±1.6**
	race	3.4±1.9	3.62±1.97	**-1.54±0.75**	-2.7±0.87	18.85±2.48	18.38±2.82
	sex	3.01±1.53	3.2±1.59	-1.9±0.83	-2.78±0.99	19.46±2.85	18.39±3.02
german	age	**1.77±1.34**	**1.82±1.43**	**-7.38±6.38**	**-5.83±6.6**	18.59±10.33	17.72±10.25
	gender	2.21±1.31	2.24±1.38	-6.07±3.27	-4.21±4.01	17.14±4.84	15.88±4.87

Table 4.2: Summary of performance similarity and low target feature attribution metrics over four **train** and **test** datasets and six features averaged over 10 different complexities. We find that the explanation loss (ζ) for **both** the train and test sets is low. Also the change in accuracy (Acc Δ) and the percentage of mismatch points (Mis (%)) between the original and modified model over both datasets are similar – min and max values in bold. These results suggest that our attack successfully generalises to unseen test points.

Further, Table 4.2 confirms that the attack generalises across datasets and features since it is capable of shifting the importance ranking distribution considerably for a total of 10 features over 4 datasets. The table indicates that the test values for both the performance similarity and low target feature attribution are either similar or lower to their training counterparts, meaning that the attack generalises to unseen test points.

4.4.2 Fairness Evaluation

Remark

Signed and Absolute Unfairness For the purposes of this Section, we measure the unfairness of a model by observing the bias of a model between two groups (privileged or unprivileged). A perfectly fair model has a bias of 0. The sign of bias, or the **signed unfairness**, signifies for which of the two groups, the model has a preference. Discarding the sign and taking the **absolute unfairness** helps us to get a different understanding of the unfairness when comparing two models.

Let us assume two groups A & B, a modified model with a 0.4 bias (i.e., bias towards group A), and an original model with a -0.6 bias (i.e., bias towards group B). Then the signed fairness difference $(0.4 - (-0.6) = 1)$ evaluates the size of the unfairness amplitude between the two models, whereas the unsigned, or absolute unfairness, $(|0.4| - |-0.6| = -0.2)$ measures whether the modified model has become more or less unfair overall. A positive value indicates that the modified model has gained fairness, whereas a negative means it has become more unfair. Hence, the signed fairness measures the difference between the two models. On the other hand, the absolute fairness evaluates the degree of unfairness in light of the fact that the model has no "apparent" reliance on the sensitive feature.

Unfair models appear fair Figure 4.4 illustrates one example of model complexity and initialisation. It depicts that our approach can hide a sensitive feature in such a way that the modified model would appear fair using local-sensitivity explanation techniques, yet actually it could become more or less unfair according to multiple fairness measures. The low local-sensitivity can result in a decision boundary that varies irrespective of the sensitive feature values, such as the one illustrated in Figure 4.1. We investigate the effects of the adversarial explanation attack on the decision boundary in Section 4.4.3.1.

Unpredictable impact on fairness We run experiments across model complexities and different initialisations. Figure 4.5 shows that *the adversarial explanation attack does not have a consistent impact on the fairness metrics, even though the apparent importance of the feature is negligible.* The attack causes the resulting model to have unpredictable unfairness behaviour, becoming more unfair for some features, less unfair for others or maintaining relatively similar fairness levels to the original model. The unpredictability of the unfairness argues strongly against relying solely on transparency to verify model fairness.

Nevertheless, in most cases, the fairness metrics are affected similarly in the sense that

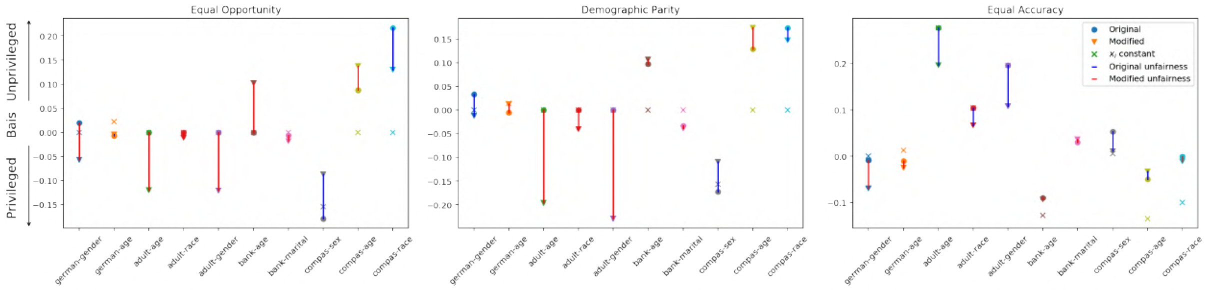


Figure 4.4: Unfairness across 3 metrics: Equal Opportunity, Demographic Parity and Equal accuracy. Each plot depicts how each of the fairness metrics is affected after an attack across 4 datasets and their sensitive features for a 5-hidden layer MLP. The y-axis illustrates whether the model is more biased towards the privileged or unprivileged group. Blue lines indicate that the modified model has become less biased, while the red lines indicate that the modified model has become more biased. We find no consistent pattern of bias towards a particular group. The crosses indicate the bias according to fairness via unawareness (see Section 4.4.3.2). We find no consistent pattern. To some extent, we see that the unfairness with respect to Equal Opportunity is higher for the original model and behaves similarly to removing the feature. Similarly for demographic parity, we find that the modified model is less biased than the original model with respect to the sensitive feature. Equal accuracy (of subgroups between both models) was least affected by our attack.

if one of the models becomes more unfair according to one metric, most of the remaining metrics vary accordingly. One possible explanation for the inconsistent behaviour of the fairness metrics after the attack could be the presence of confounding factors. Although the "explained" importance of a feature could be low, the model might have learned to rely on other features, which could be used to infer the target feature (e.g., someone's gender can be inferred from their marital status of a husband or wife). We investigate this possibility in Appendix C.2.1.

Another possibility is that the adversarial explanation attack results in a model that: a) effectively keeps the same model, but flattens the derivatives to make it locally insensitive to a feature; or b) ignores the feature altogether. We discuss evidence in favour of a) over b) in Section 4.4.3.2 and give further details in Appendix C.2.2.

Fairness and Representational Capacity Figure 4.5 demonstrates that the signed unfairness magnitude between the modified and the original models varies across fairness metrics; however, the direction of unfairness change is consistent across fairness metrics (i.e., the majority of the fairness metrics indicate the same direction).

Most importantly, the signed unfairness difference **varies arbitrarily across datasets and features**, showing an unpredictable pattern. In contrast, Figures 4.6a and 4.6b demonstrate that the absolute unfairness difference is highly dependent on the model complexity. That is, for models of lower complexities the attack makes the modified model less unfair for the majority of datasets, features and fairness metrics. However, for models of higher complexities, the attack leads to a model that is more unfair according to some

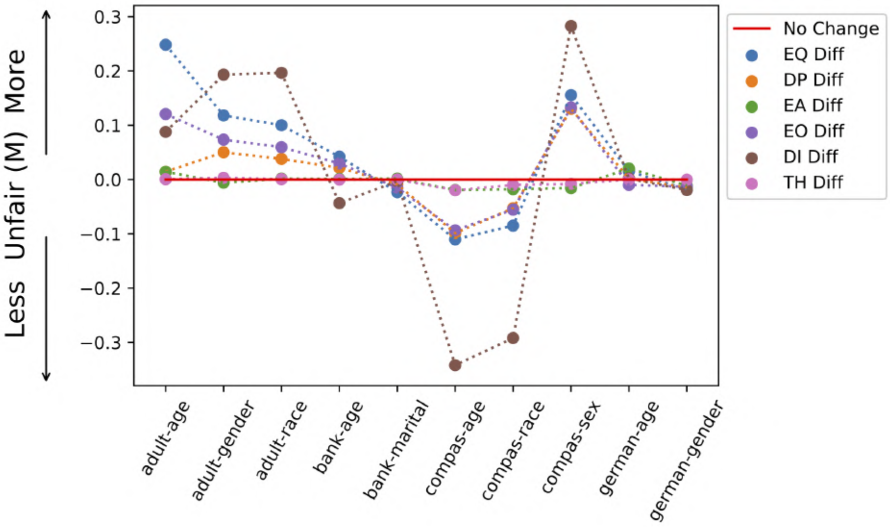


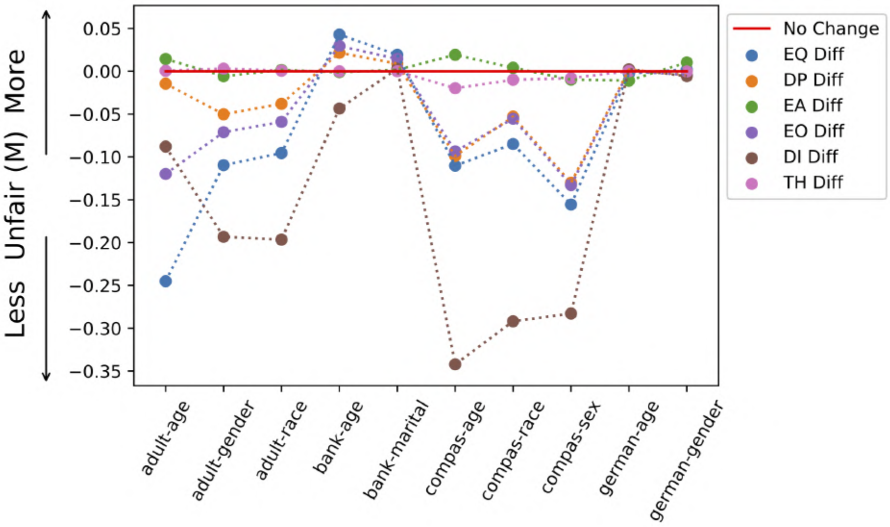
Figure 4.5: Evaluation of the impact our explanation attack has on unfairness (*signed unfairness* of modified model minus *signed unfairness* of original). We show all fairness metrics used by IBM AI Fairness 360 (Bellamy et al., 2018) across four datasets and their sensitive features, averaged over 10 model complexities (number of hidden layers) and 10 random initialisations. We find no consistent pattern of impact, though Disparate Impact (DI) appears to vary the most.

fairness measure, but less unfair according to others.

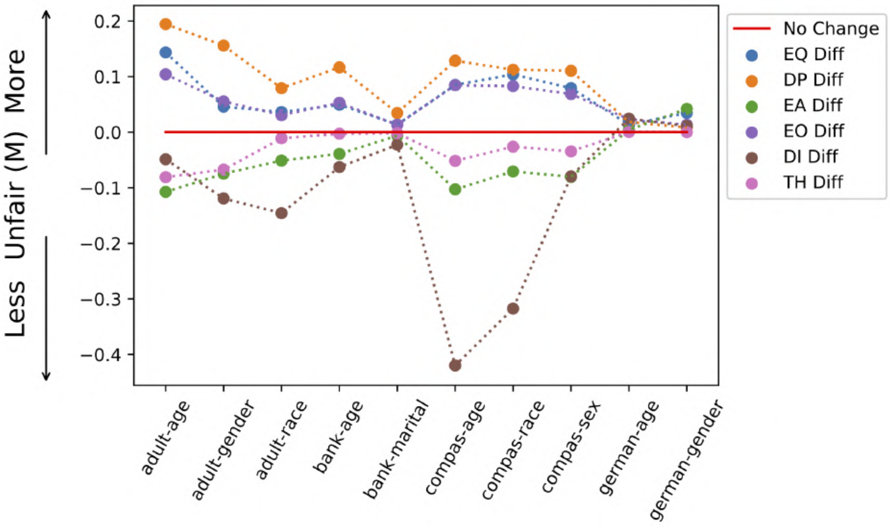
There are two crucial implications of this finding. First, the attack causes significant alternations to the fairness profile of the model despite its "apparent" insensitivity to the feature. Second, the fact that model complexity can clearly distinguish the effect the adversarial explanation attack on the change in fairness strengthens the conjecture about the critical role of representational capacity and decision boundary curvature in understanding the behaviour of neural networks. In Appendix C we demonstrate further results, which support this conjecture.

4.4.3 Model Comparison

We now investigate the similarity between the original and modified models beyond accuracy and fidelity in order to assess whether our manipulation is easy to detect. In particular, we investigate the degree to which the modified model has changed in three ways. First, we visualise the decision boundaries in 2D PCA projected space of both the original and the modified models. Second, we visualise how the output varies with respect to the target feature through partial dependence plots. Third, we compare the accuracy between three models (a) the original model, (b) the modified model, and (c) a model that ignores the feature. The aim of this study is to investigate the possibility that the modified model is completely ignoring the target feature.



(a) Evaluation of the impact our explanation attack has on unfairness (*absolute unfairness* of modified model minus *absolute unfairness* of original). We show six fairness metrics across 4 datasets and their sensitive features, averaged over model complexities **0-5 number of hidden layers** and 10 random initialisations. We find that different fairness metrics are affected differently, however, in a fashion that makes the resulting model less unfair overall.



(b) Evaluation of the impact our explanation attack has on unfairness (*absolute unfairness* of modified model minus *absolute unfairness* of original). We show six fairness metrics across 4 datasets and their sensitive features, averaged over model complexities **6-9 number of hidden layers** and 10 random initialisations. We find that different fairness metrics are affected differently, but consistently. That is a particular metric generally assigns higher or lower unfairness.

Figure 4.6: Comparison of the effect of model capacity on fairness. There are important differences to Figure 4.5. Namely, the change in unfairness seems to be much more predictable because each fairness metric is affected similarly across different features. That is, the “line of the metric never crosses the red “no change” line; consequently, according to a particular metric the model consistently appears more fair, or consistently more unfair.

4.4.3.1 Decision Boundary: How much does the model really change?

We visualise global geometry changes in the decision boundary with a 2D PCA projected space of both the original and the modified models (see Figure 4.7). Moreover, we

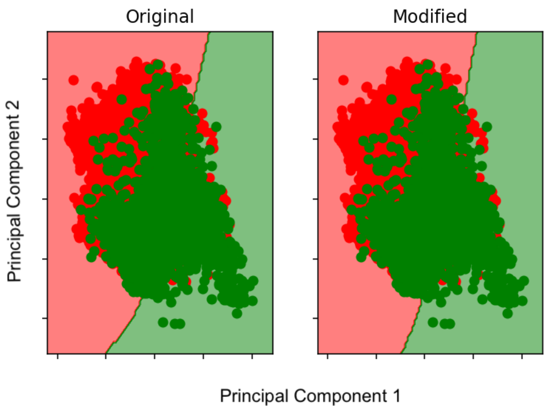


Figure 4.7: Comparison of the decision boundary between the original (left) and modified (right) classifier after an attack on Adult capital gains (most important feature) in 2D reduced input space (scikit-learn Pedregosa et al., 2011's PCA implementation). Red and green backgrounds indicate negative and positive predictions, respectively. Notice the slightly modified boundary in the lower end region with few datapoints. The circles represent the 2D projections of each point in the training and the test set, while their colour indicates the true label.

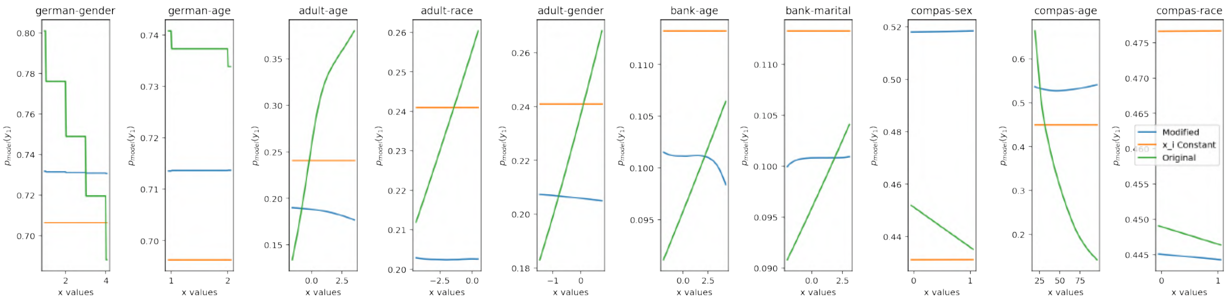


Figure 4.8: Partial dependence plots showing how the predicted output varies according to the sensitive feature shown for the original (green), modified (blue), and constant (orange) models. Results shown are for 5 hidden layers. Best viewed in digital.

measure the effect of the sensitive feature on different models through a partial dependence plot (Friedman, 2001), which plots $f(\boldsymbol{x_i})$ vs $\boldsymbol{x_i}$, where $f(\boldsymbol{x_i})$ is the response to $\boldsymbol{x_i}$ with the other attributes averaged out[6].

The small number of mismatches shown in Table 4.2 (Mis %), coupled with the small change to the decision boundary, as illustrated in Figure 4.7 suggest that *overall* the model has not changed significantly, despite the significant changes in explanation. However, Figure 4.8 suggests that the model can change significantly with respect to the target attribute. For example, a rather disappointing result is that the decision boundaries of the modified models seem excessively flat. This finding seems to refute the hypothesis that the boundary of the modified model becomes flat only in the vicinity of training points, while maintaining curvature outside of this range. One possible explanation for this result could be that partial dependence plots do not actually depict the boundary at the training

[6] We refer the reader to Chapter 3 Section 3.4.5.3 for further information.

points, but at unrealistically averaged points.

Interestingly, we observe the greatest curvature in the age feature, which has the highest mutual information with the remaining features[7]. High curvature for highly confounded features could suggest that confounders make it more difficult for our attack to flatten the decision boundary around training points w.r.t. the target feature.

It may be the case therefore that the overall geometry of the modified model does not change significantly. However, it exhibits considerable alterations in ways that suggest that the feature is completely ignored or inferred from other variables.

4.4.3.2 Fairness via unawareness

Another way to view the example in Section 4.1 is that we have a model which by construction ignores the sensitive feature x_2. This is sometimes considered a form of process fairness via unawareness (Chen et al., 2019b; Grgić-Hlača et al., 2018). It is known that even if a model cannot access a sensitive feature, it may still be unfair with respect to it. For example, the model might be able to reconstruct the sensitive feature with high accuracy from other features. This may lead one to wonder how our approach differs from simply removing the target feature.

The difference is that our approach attempts to learn a function which has very low derivative with respect to the sensitive feature at training points – hence, we might learn a function which varies significantly between the two possible sensitive feature settings. If we consider the example from Section 4.1, the function would be very flat just within the young person region, but excessively curved outside of this region, still yielding different outputs for young versus mature people.

We explore how our approach differs from simply removing the target feature in two ways. First, Figure 4.4 supports the argument that our method is different to fairness via unawareness. It shows that the unfairness of our modified model does not match that of a model which simply ignores the target feature (i.e., the crosses and the arrows do not occur in the same location). Second, we compared the performance between our method and simply ignoring the feature to demonstrate that the resulting models exhibit different behaviours. We describe the results of these observations next.

4.4.3.3 Does the model ignore the feature?

We explored whether the modified model ignores the feature by comparing modified models learned with our approach against models where the sensitive feature was held constant (we did this, rather than simply remove the feature, in order to maintain model complexity). Figure 4.9 suggests that the **modified models do not rely solely on**

[7] See Appendix C.

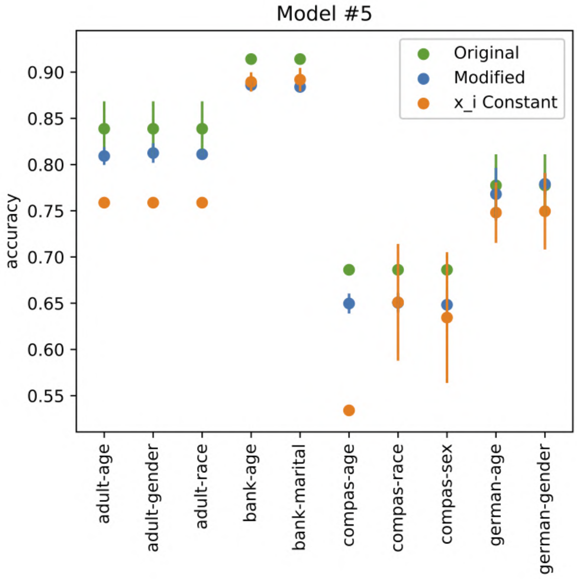


Figure 4.9: A comparison of accuracies of the modified model, a model trained with the target feature held at constant x_2, and the original model. Observe that across datasets and target features, our method achieves an accuracy comparable to the one of the original model and significantly higher than that of the constant model, demonstrating that the modified model is not merely ignoring the target feature. Results are averaged across 10 initialisations for a model with 5 hidden layers. Best viewed in colour.

correlated features. It seems they are using information from the target feature because the modified models perform better than models where the target feature is held constant. Indeed, as shown, modified models can achieve accuracy close to the original model accuracy.

Closer inspection of Figure 4.10 reveals further performance differences across model complexities, suggesting that the representational capacity might play a role in determining the success of our attack. Models of lower representational capacity are performing worse than the models, which ignore the feature altogether.

The attack seems to boost performance for higher capacity, suggesting the attack can have a regularising effect. Heo, Joo, and Moon (2019) showed a similar trend for CNNs. One possible explanation for this phenomenon could be that more complex models are better capable of extracting useful information from the target feature (while they still appear not to use the target feature according to the explanation methods we considered). We investigate the regularising effect of the adversarial explanation attack in more depth in Appendix D.

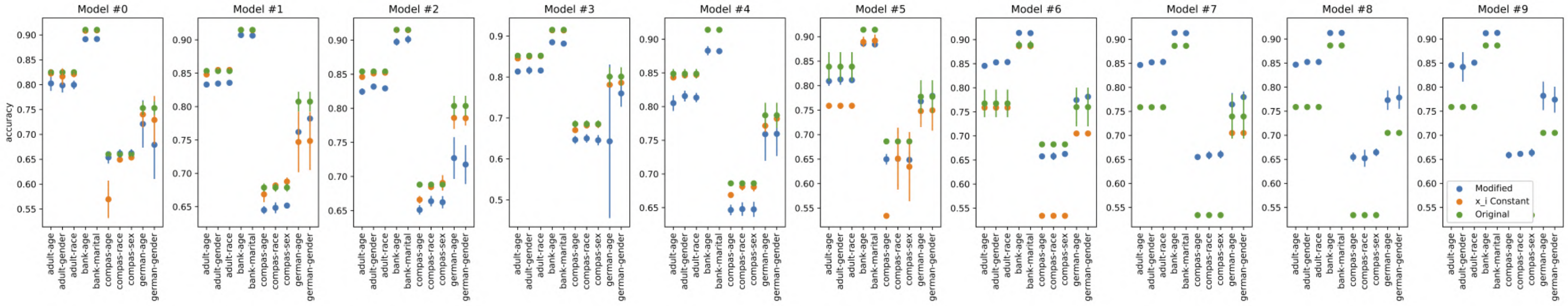


Figure 4.10: Predictive performance comparison (accuracy) of the modified model, a model trained with the target feature x_2 held at constant, and the original model across 10 random initialisations for models of increasing complexity (number of hidden layers from 0-9) on a held-out test set. Notice that for higher complexities, the model constant and the original model overlap. For higher complexities, the adversarial attack achieves better results than both the original and constant models, which suggests that our approach can also be used as a regulariser. Notice that the variance across initialisations decreases with deeper models suggesting that higher model complexity results converge to more similar regions in parameter space.

4.5 Conclusions

In this chapter, we demonstrated a limitation of many popular explanation methods – their inability to reliably indicate whether or not a model is fair. We make two arguments to support our claim. First, Section 4.1 provided an intuitive explanation to show how explainability methods might fail to describe the unfairness of a model. Second, Section 4.2 introduced a method to modify an existing model and downgrade the feature importance of key sensitive features across seven explanation methods and unseen test points across four datasets, while having little effect on model accuracy (as shown in Section 4.4). The implications of our results are twofold. First, regulators and auditors of machine learning systems should consider different methods for verifying the fairness of models. Second, our results show the inadequacy of feature-importance explanations to describe with enough fidelity and richness the behaviour of DNNs. Let us now turn our attention to developing explainability techniques that can describes DNN model behaviour with greater richness.

CHAPTER 5

Dependency Graphs for Interpreting Neural Networks

Knowledge is power only if you can act on it!

A wise man

So far we have discussed that feature importance explanations, or saliency methods, are fragile from a statistical (Ghorbani, Abid, and Zou, 2019; Kindermans et al., 2019) and adversarial (Adebayo et al., 2018; Dimanov et al., 2020) perspectives, and unsatisfactory from a cognitive (Poursabzi-Sangdeh et al., 2018; Kim et al., 2018) perspective (See Section 3.5.2.3 and Chapter 4). In Chapter 2, we saw that DNNs represent information based on particular assumptions about the world (see Section 2.2) and in a particular form using a combination of local representations and sparse, or *partially-distributed representation* (PDR) (Li et al., 2016; Fong and Vedaldi, 2018). At the same time, recent studies suggest that the human cognition prefers and operates more readily with explanations in the form of high-level semantic units, termed **concepts** (Kim et al., 2018; Ghorbani et al., 2019).

As a consequence, two strands of research have emerged to build on saliency methods – model extraction and concept-based explanations. Model extraction, or model translation, approaches approximate black-box complex models with simpler models (such as decision trees, lists, or linear models) to increase the model transparency. Provided the approximation quality (referred to as *fidelity*) is high enough, extracted models could preserve many statistical properties of the original black-box model, while remaining open to interpretation. On the other hand, concept-based approaches aim to provide explanations of a DNN model in terms of human-understandable units, rather than individual features, pixels, or characters. For example, the concepts of a *wheel* and a *door* are important for the detection of cars.

Aims & Hypothesis The purpose of the investigation in the next two chapters is to explore the possibility of gaining additional insights into the neural network's internal operation in terms of human interpretable concepts. Recently, it has been suggested that the identification and interpretation of partially-distributed representation will enhance our understanding of this internal operation (Olah et al., 2018). We hypothesise that the "interface for communication" between DNNs and humans will happen on the level of partially-distributed representations and concept-based explanations. We propose that model extraction of the DNN's functional decomposition is one way to achieve the goal of building this "interface".

Methodology: Model approximation using functional decomposition We consider two different types of model functional decomposition: (1) decomposing the model into a series of functions, which identifies relevant neuron to extract high-level concept representations in the form of class-specific representations; (2) decomposing the model into two functions, such that the extracted model operates on an interpretable representation in concept space. In this chapter, we introduce the former approach, whereas in Chapter 6 we introduce the latter.

*(D)ependency (G)raphs for (I)nterpreting (N)eural (N)etworks (**DGINN**)*, which we present in this chapter, is a novel framework for interpreting DNN classification decisions through identifying class-specific representations. Our framework can distil the quintessential part of the network related to a particular class, demonstrating that class-specific representations emerge in the form of sub-networks, or sub-graphs, which we term class-specific dependency graphs.

Contributions To the best of our knowledge, we are the first to propose generating semi-local explanations to increase the level of explainability by aggregating results from importance-based explanations, thus paving the way towards concept-based explanations. Our findings contest the claim that feature importance of individual neurons are a reliable way to debug and analyse the behaviour of neural networks, as recently proposed (Zintgraf et al., 2017). We caution against interpretations of single neurons in isolation and make a case for labelled datasets that allow for controlled qualitative evaluation of explainability techniques.

We provide additional insights about the shared factors, natural clustering, and sparsity assumptions, described in Chapter 2.2, and demonstrate that class-specific dependency graphs identify parts of the internal representation, sub-graphs, that are shared across different classes and cluster into semantically meaningful groups. A class-specific dependency graph can extract a binary classifier for the corresponding class, from a fraction of the original DNN parameters. Surprisingly, we find the existence of static class representations, which are input invariant for hard-pruned networks (> 80% of parameters removed). These findings suggest that class-specific dependency graphs identify the partially-distributed representations that encode the low-dimensional manifolds, along which the internal representation represents the underlying factors of variation related to a particular class. Our method can be used to extract interpretable models that are capable of translating black-box DNN decision into human-understandable concept explanations. We demonstrate one example of extracting concept-based models in Chapter 6.

Our framework can be used for research, auditability, and model enhancement. For example, we demonstrate that the framework can be used to compare the quality of the extracted dependency graphs to evaluate neuron importance methods, thus contributing to the enhancement of evaluation of importance-based techniques. Our framework has implications for those developing monitoring techniques for measuring the reliability of DNN decisions, as well as DNN developers conducting data augmentation, who could examine the features captured by a DNN.

5.1 Framework

DGINN is a DNN interpretability framework that decomposes the model into a series of functions to extract class-specific representations. Figure 5.1 illustrates the high-level idea. In summary, DGINN can produce two types of dependency graphs: (1) class-specific layer-wise dependency graphs; and (2) neuron-specific dependency graph. The layer-wise dependency graph indicates the neurons relevant to the specific class in each layer, while the neuron-specific dependency graph indicates the pertinent neurons between a pair of layers given the target class. A layer-wise dependency graph contains a set of relevant neurons in each layer, while a neuron-specific dependency graph includes a set of neurons pertinent to an upper-layer target neuron. Next we describe the precise process of extracting dependency graphs.

5.2 Mathematical Formulation

Before we introduce DGINN formally, let us first define the mathematical formulation used throughout this chapter. We consider a pre-trained DNN classifier, $f : \mathcal{X} \to \mathcal{Y}$, ($\mathcal{X} \subset \mathbb{R}^{H \times W \times C} \subset \mathbb{R}^m$, $\mathcal{Y} \subset \mathbb{R}^{d^\circ}$), where H, W, K are respectively the height, width, and channels of an image, m is the cardinality of the input space, equal to $H \times W \times C$, and d° is the cardinality of the output space, equal to the number of classes. Hence, $f(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{y}$ is a $\boldsymbol{\theta}$-parameterised function, mapping from an input $\mathbf{x} \in \mathbb{X} \subset \mathcal{X}$ to an output $\mathbf{y} \in \mathcal{Y}$, where y_i corresponds to a particular class. For every DNN layer $l \in \{1..L\}$ of an L layers

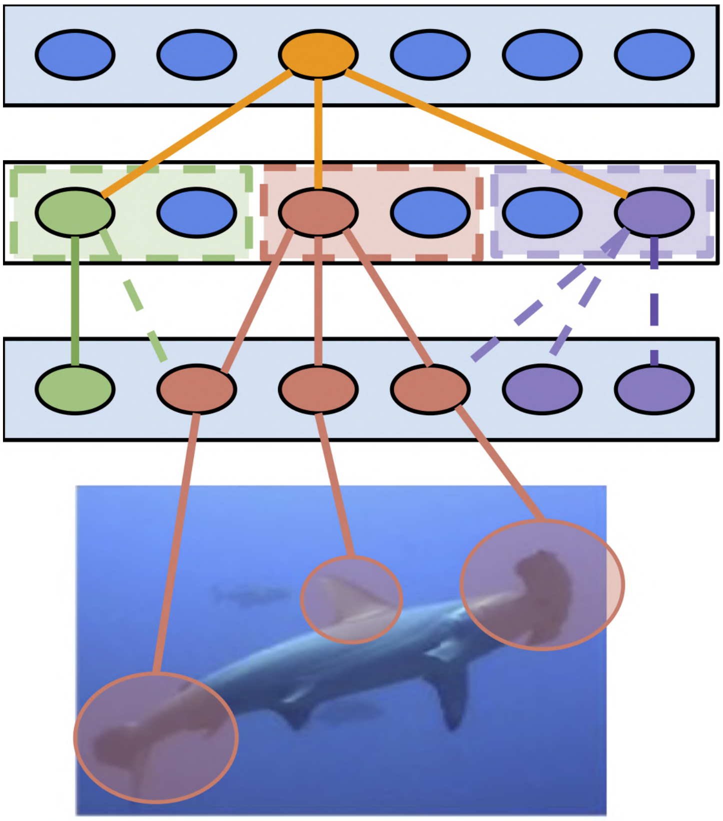


Figure 5.1: Schematic Representation: A sketch of how our framework Dependency Graphs for Interpreting Neural Networks (DGINN) can be used to provide interpretation for a shark prediction (for actual output examples, see Fig. 5.8). Each step identifies partially distributed representations (PDRs) of relevant neurons, and the algorithm recursively traverses the lower layers for each PDR.

deep network, we denote the function $f^l : \mathcal{X} \rightarrow \mathcal{H}^l$, ($\mathcal{H}^l \subset \mathbb{R}^d$) as the mapping from the input space $\mathcal{X}$ to the hidden representation space $\mathcal{H}^l$, where d denotes the number of hidden units and can be different for each layer. Finally, the function $h^l : \mathcal{H}^{l-1} \rightarrow \mathcal{H}^l$ maps between the hidden representations between two layers, such that $h^0 = \mathbf{x}$.

Here we define the two equations that describe the majority of feature importance methods. Given an image, we approximate the output $\mathbf{y}$ of the DNN model f in the neighbourhood of $\mathbf{x}$ with a linear function using a first-order Taylor approximation:

$$\mathrm{y}_i = f_i(\mathbf{x}) \approx \boldsymbol{\omega}^T \mathbf{x} + b, \tag{5.1}$$

where $\boldsymbol{\omega}$ is the gradient of $f_i(\mathbf{x})$ with respect to an image $\mathbf{x}$, evaluated at $\mathbf{x}$:

$$\boldsymbol{\omega} = \frac{\partial f_i(\mathbf{x})}{\partial \mathbf{x}}\bigg|_{\mathbf{x}}. \tag{5.2}$$

We can interpret the magnitude of the values of $\boldsymbol{\omega}$ as an *importance metric* of each pixel. Each value indicates which *pixels* of $\mathbf{x}$ need to be changed the least to cause the greatest increase in $\mathbf{y}_i$. This process is known as sensitivity analysis (Simonyan, Vedaldi, and Zisserman, 2013) (see Section 3.5.2). Most feature importance metrics can be derived using Equation 5.1 and a slight modifications to the formulation of Equation 5.2 (Ancona et al., 2018).

We propose a much more fine-grained analysis based on the hypothesis that sensitivity analysis can be used in an analogous way to determine the relevance between adjacent layers. Instead of trying to approximate $\mathbf{y}_i$ directly, we consider f to be defined as the successive composition of simpler functions that represent the transformations of data between layers:

$$\begin{aligned} f(\mathbf{x}) &= h^l(f^{l-1}(\mathbf{x})) \\ &= h^l \circ h^{l-1} \circ h^{l-2} ... \circ h^1(\mathbf{x}) \end{aligned} \tag{5.3}$$

Hence, we can evaluate the Taylor approximation at image $\mathbf{x}^{(i)}$ between a higher and lower layer, respectively l and j:

$$\boldsymbol{\omega}^l_{n,i,:} = \left. \frac{\partial h^l_n\big(f^j(\mathbf{x})\big)}{\partial \mathbf{x}} \right|_{\mathbf{x}^{(i)}} \tag{5.4}$$

Further, given the sparsity assumption from Section 2.2, we hypothesise that there are very few neurons that encode particular concepts or concept values. We conjecture that the relevance values of these vital neurons are significantly higher than the relevance values of other neurons. We demonstrate results that support this conjecture in Section 5.5. Hence, we propose an outlier detection technique to discover the neurons that have the highest likelihood of encoding concepts (described in Section 5.3.2). We conjecture that this small set of relevant neurons defines the low-dimensional manifolds, which describe the data variations that are idiosyncratic to specific classes. Section 5.5 provides evidence that support this conjecture.

5.3 DGINN Framework Details (Algorithm 2)

Given a DNN classifier f, a set of datapoints $\mathbb{X}$, and a set of target labels in the form of relevant neurons in the output (top) layer $\mathbb{S} \ni n.n \in h^l$, start from the top layer and follow the four steps in Algorithm 2 to produce a set of b relevant neurons $\mathbb{S}^{l-1}$ in the lower layer. Then set $\mathbb{S} := \mathbb{S}^{l-1}$ and repeat until the input layer.

Algorithm 2 consists of the following steps. Step I. computes the relevance between all neurons between two adjacent layers. Step II. aggregates across datapoints to weight the neuron relevance w.r.t the datapoints under investigation. Step III. aggregates across upper-layer neurons to weigh the layer-wise importance of a neuron to compute a proxy

Algorithm 2 DGINN framework – Identifying partially-distributed representations

INPUT: DNN classifier f, a layer l ($h^l \in \mathbb{R}^d$) from f, a set of relevant neurons $\mathbb{S} \ni n.n \in h^l$, and a set of images $\mathbf{x}^{(i)} \in \mathbb{X}$.

STEP I: Compute $\boldsymbol{\Omega}^l \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{X}|\times d'}$ relevance of neurons in layer $h^{l-1} \in \mathbb{R}^{d'}$ to each n and $\mathbf{x}^{(i)}$ using Equation 5.4 so that if f^{l-1} is a:

- Fully-connected layer: stack results into a relevance tensor $\boldsymbol{\Omega}^l \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{X}|\times d'}$;
- Convolutional layer: spatially average the omega tensor elements $\boldsymbol{\omega}^l_{n,i,h,w}$ into a relevance tensor $\boldsymbol{\omega}^l \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{X}|\times d'}$;
- Pooling-layers: directly compute for $l-2$: $\boldsymbol{\omega}^l = \nabla_{f^{l-2}} h^l \circ f^{l-2}|_{\mathbf{x}^{(i)}}$

STEP II: Aggregate a relevance tensor $\boldsymbol{\Omega}^l$ across data points to produce a relevance matrix $\boldsymbol{\Omega}^l \in \mathbb{R}^{|\mathbb{S}|\times d'}$ that indicates the relevance between the neurons in layers l and $l-1$. Aggregation can be either 1) an averaging aggregation function that yields a continuous output; or 2) an outlier aggregation function (Tukey, 1977) that yields a binary output.

STEP III: Aggregate a relevance matrix $\boldsymbol{\Omega}^l$ across upper layer neurons to produce relevance vector $\boldsymbol{\omega}^l \in \mathbb{R}^{d'}$. The output is the overall importance ranking of all neurons in a **layer** in the form of a relevance vector $\boldsymbol{\omega}^l$ (*layer-wise* mode). Alternatively, we can skip this step to preserve the relevance with respect to a particular neuron in the form of **neuron-specific** relevance vector $\boldsymbol{\omega}^l_{n,:}$.

STEP IV: Threshold b relevant neurons. For *layer-wise output*, perform statistical thresholding of all neurons above a certain percentile such that the resulting number of neurons equals b. For *neuron-specific output*, select top b neuron values $\boldsymbol{\omega}^l_{n,:}$ for each n in $\mathbb{S}$.

OUTPUT: $\mathbb{S}^{l-1}$ with b relevant neurons.

for a neuron's "reusability" across upper-layer neurons. Hence, this estimate is a proxy for the likelihood of a neuron to be part of a PDR. Step IV. thresholds relevant neurons based on statistical analysis to get a total of predefined number of b relevant neurons.

We can skip Step III. to preserve the mapping between an upper-layer neuron and its relevant neurons and separate the relevant neurons into distinct PDRs. This flexibility enables us to investigate the distribution of relevant neurons across the entire layer with respect to upper-layer neurons. Therefore, DGINN is capable of producing dependency graphs both across layers of the entire network (layer-wise) and between pairs of layers, indicating the neurons pertinent to the activation of an upper layer neuron (neuron-specific). The result of the layer-wise execution is a set of relevant neurons in each layer, while the result of the neuron-specific execution is a set of neurons pertinent to an upper-layer target neuron.

Since DGINN is a framework, different equations in each of the steps could achieve various goals. For example, the first step can apply any method that computes relevance scores, including gradient- (Ancona et al., 2018), statistical- (Zintgraf et al., 2017), or game-theory- (Chen et al., 2019c) based approaches. Here we demonstrate a simple method for computing the relevance importance, *sensitivity analysis* (Baehrens et al., 2010; Simonyan, Vedaldi, and Zisserman, 2013), yields high-quality results. In Chapter 6, we show that even the most straightforward way of using the activations is enough to extract concepts from hidden representations.

The rest of this Section formally presents the four steps of the DGINN framework, illustrated in Figure 5.2.

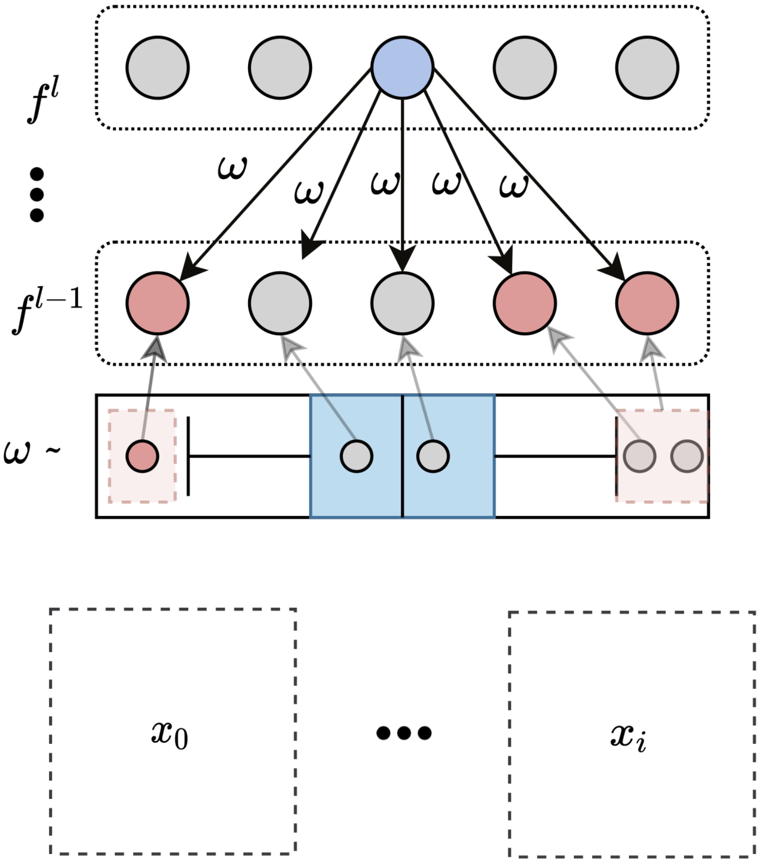


Figure 5.2: Visual Abstract of the Methodology: Visual Representation of DGINN's two novelties. First, the outliers of the $\boldsymbol{\omega}$ value distribution($\boldsymbol{\omega}$ $\sim$), represented with a boxplot, determine the relevant neurons across the layers. Second, the analysis is aggregated across instance-specific computations (on datapoints $x_0...x_i$) to gain model-centric results.

5.3.1 STEP I: Compute Relevance Tensor

Input: This step requires a network (f), a layer l, a set of relevant neurons $\mathbb{S} \ni n.n \in h^l$, and a set of images $\mathbf{x}^{(i)} \in \mathbb{X}$..

Output: $\mathbf{\Omega}^l \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{X}|\times d'}$ relevance scores of neurons in layer $l-1$ ($f^{l-1} \in \mathbb{R}^{d'}$) with respect to a neuron n in layer l (h^l) as a gradient at $\mathbf{x}^{(i)}$ using Equation 5.4. Essentially, this produces the relevance of all neurons in layer f^{l-1} to the activation of neuron n.

Method: The relevance for DCNN is computed differently depending on the type of layer h^l.

If h^l is **fully-connected**, the result is a relevance vector $\boldsymbol{\omega}^l_{n,i,:} \in \mathbb{R}^{d'}$. Repeating this process for all images and neurons in $\mathbb{S}$ yields a relevance tensor $\mathbf{\Omega}^l \in \mathbb{R}^{|\mathbb{S}|\times|\mathbb{X}|\times d'}$.

If h^l is a **convolutional** layer, the result of Equation 5.4 is a 3D relevance tensor $\mathbf{\Omega}^l_{n,i,\dots} \in \mathbb{R}^{H \times W \times K}$, where H, W, K are respectively the height, width, and number of activation maps in $l-1$. Since every activation map k ($h^l_{:,:,k} \in \mathbb{R}^{H' \times W'}$) is produced by convolving identical weights onto a lower layer activation map p ($h^{l-1}_{:,:,p} \in \mathbb{R}^{H \times W}$), k represents an identical feature across the activation map p. Hence, the vector $\boldsymbol{\omega}^l_{n,i,h,w,:} \in \mathbb{R}^K$ represents the relevance of all lower level activation maps (features) at a location (h, w) to the activation of unit n.

We apply spatial-averaging over all locations (h, w) to obtain the relative importance of a feature. That is, we convert $\boldsymbol{\omega}^l_{n,i,h,w,:}$ into a relevance vector $\boldsymbol{\omega}^l_{n,i,:} \in \mathbb{R}^K$, where each element $\omega^l_{n,i,k}$ indicates the relative importance of an activation map k across all locations.

We can repeat the process for all images and relevant neurons to obtain a 3D relevance tensor $\mathbf{\Omega}^l \in \mathbb{R}^{|S| \times |\mathbb{X}| \times K}$.

The **pooling** layers can be seen as a filter of their predecessors since $\frac{df^l}{d\mathbf{x}^{(i)}} = \mathbf{c} \times \frac{df^{l-1}}{d\mathbf{x}^{(i)}}$, where $\mathbf{c} \in \{0, 1\}^{d'}$. Hence, if f^{l-1} is a pooling layer, we compute the relevance tensor directly w.r.t $l-2$:

$$\boldsymbol{\omega}^l = \nabla_{f^{l-2}(\mathbf{x})} h^l(f^{l-2}(\mathbf{x}))|_{\mathbf{x}^{(i)}}$$

.

We can change Equation 5.4 and experiment with different ways to compute relevance values. For example, instead of gradients, Equation 5.4 can use the weights (similar to the lottery ticket hypothesis (Frankle and Carbin, 2018)), the activations ($f^j(\mathbf{x})$) or the element-wise product between the weights and the activations[1]. In the case of using the weights as relevant values, the next step is redundant since they do not vary with the input samples. Section 5.5 presents a comparison between these alternatives.

5.3.2 STEP II: Aggregate Across Datapoints

Input: This steps requires a relevance tensor $\mathbf{\Omega}^l \in \mathbb{R}^{|\mathbb{S}| \times |\mathbb{X}| \times d'}$.
Output: The result is a relevance matrix $\mathbf{\Omega}^l \in \mathbb{R}^{|\mathbb{S}| \times d'}$ that indicates the relevance between the neurons in layers l and $l-1$.
Method: This step aggregates across the dataset dimension (i) of the relevance tensor $\mathbf{\Omega}^l_{:,i,:}$. We find that each row $\boldsymbol{\omega}^l_{n,i,:} \in \mathbb{R}^{d'}$ follows a normal distribution, and consistently exhibits a small number of outliers across the data dimension i (see Section 5.5). Assuming the sparsity assumption holds, we hypothesise that these outliers are the only relevant neurons since they describe the low-dimensional manifolds, which capture the class- or concept-specific variation. Consequently, we use the Tukey's fences (1.5× Inter-Quartile Range) outlier detection method (Tukey, 1977) to select relevant neurons from each row

[1] Alternatively, one can learn an additional signal on top of the weights and activations, as in Pattern Attribution (Kindermans et al., 2017) (see Section 3.5.2).

$\boldsymbol{\omega}^l_{n,i,:}$. Section 5.5 provides evidence supporting this hypothesis.

Alternatives to the outliers aggregation exist and could be investigated in future work. One example is an averaging strategy, which takes the mean over the datapoints dimension to produce a relevance matrix $\boldsymbol{\Omega}^l \in \mathbb{R}^{|\mathbb{S}| \times d'}$. This matrix indicates the average relevance across datapoints between neurons in layers l and $l-1$. Observe that in the case of the averaging strategy, the relevance matrix $\boldsymbol{\Omega}^l$ contains continuous values, while in the outliers case, it contains binary values.

Note that the aggregation functions estimate the empirical relative relevance of neurons. That is, they operate across data points, and as such, they yield relative, not absolute results. Since scaling the weights in the model results in an absolute change in all ω values (without affecting the relative values), this step is invariant to weight scaling.

5.3.3 STEP III: Aggregating across upper layer neurons

Input: Relevance matrix $\boldsymbol{\Omega}^l \in \mathbb{R}^{|\mathbb{S}| \times d'}$.
Output: Relevance vector $\boldsymbol{\omega}^l \in \mathbb{R}^{d'}$.
Method: Here we aggregate across the dimension of upper layer neurons $(n.\boldsymbol{\Omega}^l_{n,:}\ )$ to produce a global layer ranking in the form of a relevance vector $\boldsymbol{\omega}^l \in \mathbb{R}^{d'}$. We use mean averaging aggregation for our experiments; however, many different alternatives exist and could be investigated in future work (e.g., median, mode). Notice that it is possible to preserve the local relevance of the neurons in this step. When we skip this step, the local relevance is preserved and the result is a neuron-specific relevance vector $\boldsymbol{\omega}^l_{n,:}$. These vectors can be used to explore PDRs as we demonstrate in Section 5.6.

5.3.4 STEP IV: Threshold

Input: Branching factor b, and a relevance vector $\boldsymbol{\omega}^l$ or relevance matrix $\boldsymbol{\Omega}^l \in \mathbb{R}^{|\mathbb{S}| \times d'}$.
Output: Set $\mathbb{S}^{l-1} \ni n'.n' \in h^{l-1}$ of all relevant neurons for the lower layer.
Method: For the layer-wise relevance case, we perform statistical thresholding of all neuron relevance values $\boldsymbol{\omega}^l$ above a certain percentile t such that we get a set $\mathbb{S}^{l-1}$ of b relevant neurons for the lower layer. In Section 5.5, we investigate the effect of t on the quality of dependency graphs.

For the neuron-specific relevance case, we follow the sparsity assumption to select the top b relevant neurons $\mathbb{B}^n$ for each $n \in \mathbb{S}^l$ using the outlier statistical thresholding (mentioned in STEP II.) on $\boldsymbol{\Omega}^l_{n,:}$. Then we count the number of occurrences of lower layer relevant neurons n' across all sets of upper layer relevant neurons $\mathbb{B}^n$ and return a set $\mathbb{S}^{l-1}$ of the most frequently relevant n' neurons.

Time complexity The time complexity of our approach in the worst-case is $O(c*d*n)$, where c is the time to perform the relevance computation, d is the depth, and n is the maximum number of neurons in any layer.

The approach is still practical since it is not supposed to be executed every time that an explanation is necessary, just as a network is not retrained every time before a prediction. On Tesla P100 it takes 6.5 seconds to generate a dependency graph for a Conv-Net model on 5000 CIFAR images[2].

5.4 Experimental Set-up

Here we conduct a quantitative and qualitative evaluation of the DGINN framework. We carry out a quantitative evaluation on two datasets: (1) Circles dataset - a toy non-linear binary classification problem of two circles, one smaller circle inside a bigger one; and (2) CIFAR-10 (Krizhevsky, Hinton, et al., 2009). We use the following models: 2 hidden-layer Multi-Layer Perceptron (MLP) (with 8 and 16 neurons respectively) for circles[3]; and a convolutional network (Conv-Net) that achieves **88.19%** and **83.85%** accuracy on the CIFAR-10 training and test sets respectively with the following layers: conv 3x3x64, max-pool, conv 3x3x64, fully-connected 328 units, fully-connected 194 units, soft-max 10 units with RELU (Nair and Hinton, 2010) activations).

For the qualitative evaluation, we present results from Conv-Net on CIFAR and VGG16 (Simonyan and Zisserman, 2014)[4] on ImageNet (Russakovsky et al., 2015). We investigate the following threshold values – $t \in \{10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%\}$.

5.5 Quantitative Evaluation

This section presents four quantitative results. First, it presents pieces of evidence that support the sparsity assumption[5] and justify the choice of outlier detection as a technique for relevant neuron selection. Second, it presents a comparison between four alternative techniques for Equation 5.4, which we call **relevance functions**. Third, we use ablation studies to evaluate the ability of class-specific dependency graphs (with various relevance functions) to extract class-relevant information across threshold values. Specifically, we compare the accuracy of the original network to that of a "pruned network", where all weights that are not part of the dependency graph are masked to zero as irrelevant. Fourth,

[2] Section 5.4 describes the model and dataset.

[3] An MLP with a single hidden layer of 3 neurons can solve the problem. Since we want to evaluate whether DGINN can distinguish relevant from irrelevant neurons, we intentionally train a more complex 2 hidden-layer MLP.

[4] The publicly available pre-trained model implemented in keras (Chollet et al., 2015).

[5] See Section 2.2.

we highlight that the performance of the input-variant class-specific dependency graphs generalises to unseen datapoints. Finally, we demonstrate the presence of input invariant class-specific dependency graphs, which suggest that the class-specific dependency graph has isolated the manifold representing the corresponding class. Further, we discuss the relation of this finding to the lottery ticket hypothesis.

We compare across the following relevance functions:

- *weights_abs*: the absolute value of the weights;
- *activations_abs*: the absolute average activation of a neuron over a target dataset;
- *weight_act_abs*: the absolute average activation of a neuron over a target data multiplied by the absolute value of the weight;
- *gradients_abs*: the absolute gradient values of a neuron w.r.t to the activation of an upper-layer neuron averaged across the target data.

We ran a hyperparameter investigation on the Circles dataset to determine whether or not to use **absolute values** for the relevance functions. We demonstrate that the ReLU activations make the absolute values redundant for activations and gradients since their values are always non-negative. However, absolute values make a significant difference when the weights are used since they could contain negative numbers. Therefore, we use the absolute values for all functions. Additionally, unless stated otherwise, we use a threshold value of 50% of the network parameters.

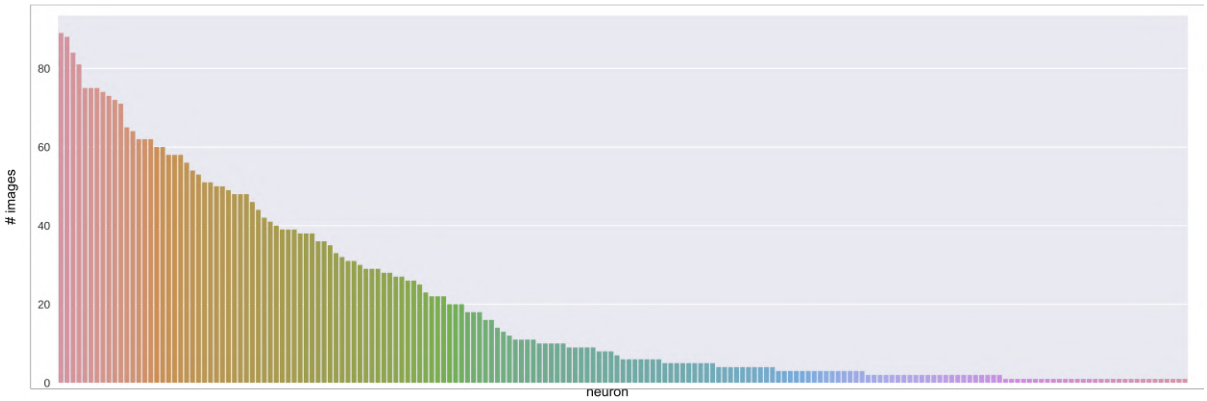


Figure 5.3: Barplot representing the frequency of occurrence of outliers in layer f^{fc2} for the Hammerhead shark class. The y-axis represents the number of images, in which a neuron was an outlier. There are 189 unique outliers (4.6% of the total 4096 neurons). Notice that the first 3 outliers occur in almost all images and that the relevance follows a power-law distribution.

Sparsity Assumption Investigation Here we conduct a statistical analysis of $\boldsymbol{\omega}$ computed for a VGG16 model trained on ImageNet and evaluated on the Hammerhead shark class. We make three important findings: (1) outlier ω values emerge consistently

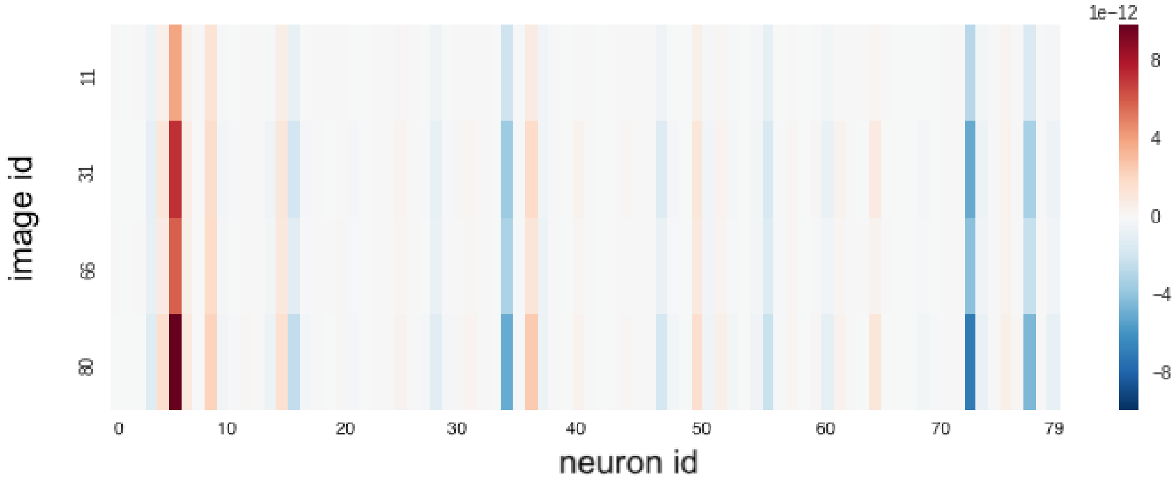


Figure 5.4: A heatmap of amplified (cubed) $\mathbf{\Omega}^{fc2}_{0:4,n,0:200}$ values for 4 Hammerhead shark images. The x-,y-,z-axes represent $\omega^{fc2}_{i,n,k}$, which is the relevance (z-axis) of neuron k (x-axis) to an arbitrary neuron n for an image i (y-axis). Observe that the images share exactly the same small number of positive and negative outliers with varying degrees of intensity. Notice this is different to a Hinton diagram (Hinton, McClelland, and Rumelhart, 1986), which visualises the weights and biases.

across inputs; (2) the relevance of a neuron always has the same sign across datapoints of the same class; (3) the frequency with which a neuron has an outlier-high relevance follows a power-law distribution.

Our analysis reveals the consistent presence of a small number of outlier ω values (less than 6%) across layers. Figure 5.3 depicts that a small number of neurons have considerably higher relevance values that the rest of the neurons for a large number of semantically similar images, which is in accord with the sparsity and manifold assumptions[6].

Figure 5.4 shows that when the extreme ω values are amplified, similar patterns appear with varying degrees of strength across a small number of images. The figure demonstrates that not only do the same neurons share outlier-high relevance values, but also that these relevance values have the same sign across different input stimulus of the same class. This finding suggests that the outlier values could correspond to neurons, which are relevant to the representation of a particular class (i.e., that define the dimensions of the class manifold). It may be the case therefore that Figure 5.4 is a visualisation of part of the PDR for a Hammerhead shark in layer f^{fc2}. We investigate this hypothesis in Paragraph “Input Invariance”.

At the same time, Figure 5.4 portrays that the absolute values of the outlier ω values are not identical across the inputs. This finding indicates that our approach is not equivalent to merely selecting the neurons with the highest weights, which would yield the same neurons across images. On the contrary, Figure 5.3 shows the frequency of relevance

[6] See Section 2.2.

follows a power-law distribution. There are several possible explanations for this result. One explanation might be that a few concepts, or particular concept values, frequently appear because they are highly characteristic of the particular image class. Hence, they are common for most images of the same class. Another explanation could be that a long tail of concept values distinguishes various instances of the same class.

Another possible explanation might be that the information in a neural network is represented in two different ways: a) as pockets or blobs of manifolds that separate different concept values (e.g., value red and value blue of the concept colour); and b) continuous manifolds describe the general variance within a class, or concept. We investigate this hypothesis further in Section 6.3.3.1. Either case suggests the existence of a long tail of infrequently used concept values that could be represented with a sparse representation.

Relevance Functions The purpose of the following set of experiments is to benchmark the performance of our proposed technique for neuron importance estimation (gradients) against other alternatives.

A well-established metric for examining the importance of pixels (Dabkowski and Gal, 2017) or concepts (Ghorbani et al., 2019) is the *smallest sufficient units* (SSU) metric which looks for the smallest set of units (pixels, concepts, neurons) that are enough for predicting the target class. Here we propose to follow the same methodology to approximate the overall importance of a neuron through ablation experiments, in which we disconnect relevant neurons from the network according to their importance. In this respect, our methodology is similar to Bau et al. (2019), who mask the activation of a neuron to measure its importance. Similarly, Hinton, Osindero, and Teh (2006) and Bengio et al. (2007) evaluate the quality and utility of representations by training a linear classifier on top of them. In contrast, we do not retrain the classifier since we are interested in the information it has already learned.

Tables 5.1 & 5.2 demonstrate that even though the absolute weights strategy (c.f. lottery ticket hypothesis) exhibits the highest performance across all points (train and test), it is unable to extract class-specific information (Class 2 Table 5.1 & Ts Class* Table 5.2). This inability is because the *static nature* of weights does not carry information relevant to individual datapoints or distinct sets of datapoints (e.g., classes).

In contrast, on both the Circles (Table 5.1) and the CIFAR (Table 5.2) datasets, the class-specific performance of the gradients, activations, and absolute weights_activations relevance functions at threshold $t = 50\%$ is significantly higher than that of the weight function. These three functions are much more dynamic and input-dependent strategies that capture more class-specific information. While activations are faster to compute, gradients provide additional information in the form of interdependence between neurons of different layers.

	Train	Test	Class 1	Class 2
weight	82.72±12.58	82.23±13.0	96.17±12.17	69.37±26.41
weight_abs	**87.12**±10.08	**86.47**±10.59	97.21±5.63	76.57±22.2
activations	71.36±15.58	72.27±15.91	**100.0**±0.0	71.23±27.84
activations_abs	71.36±15.58	72.27±15.91	**100.0**±0.0	71.23±27.84
grads	74.14±13.63	73.82±13.94	**100.0**±0.0	71.81±26.0
grads_abs	74.14±13.63	73.82±13.94	**100.0**±0.0	71.81±26.0
weight_act	81.41±13.35	80.88±13.29	95.05±13.56	69.38±26.28
weight_act_abs	75.36±10.39	75.92±10.91	**100.0**±0.0	**79.88**±16.82

Table 5.1: Circles Dataset. The table demonstrates the mean±standard deviation performance of the dependency graphs at threshold $t = 50\%$ over 100 different model initialisations. The columns indicate the dataset used for evaluation. The class columns indicate which of the two classes is considered.

	Train	Tr Class*	Test	Ts Class*
weight_abs	**51.60**	51.6±23.53	**46.86**	46.86±23.07
weight_act_abs	51.26	91.54±13.25	45.79	83.22±19.81
gradients_abs	45.28	93.12±8.76	41.54	85.53±17.02
activations_abs	45.79	**93.75**±8.46	42.14	**86.3**±16.67

Table 5.2: CIFAR-10 Dataset. The table demonstrates the mean±standard deviation accuracy of the dependency graphs at threshold $t = 50\%$ over 100 different model initialisations. The columns indicate the training dataset for the relevance functions and the evaluation. The Class* columns indicate the average true positive rate (TPR) of class-specific dependency graphs across the 10 classes, while Tr and Ts indicate training and test sets respectively. Compare this to the original accuracy of **88.19%** and **83.85%** training and test respectively, reported in Section 5.4.

Generalisation Tables 5.1 & 5.2 demonstrate that the performance of the extracted sub-networks generalises to unseen datapoints. Activations, gradients, and absolute weight_activations have comparable performances reaching 86% accuracy for unseen data averaged across class-specific sub-networks that contain *less than 80% of the total network parameters.* This finding suggests that the DGINN framework can successfully identify the class relevant manifolds within the network.

On the Circles dataset, the activations, gradients, and weight_act strategies extract the class 1 ideally over 100 models, while for class 2 weight_act outperforms gradients with 79.88% to 71.81% A possible explanation for this result might be that DNNs solve binary problems by learning more about one of the two classes. This type of shortcut learning is a well-documented problem (Geirhos et al., 2020), and our results demonstrate that dependency graphs can be used to identify such occurrences.

Threshold Next, we compare the different relevance functions across threshold values. The objective of these experiments is threefold: (1) select the highest performing relevance functions; (2) select the highest performing threshold value; (3) investigate the class information captured within the class-specific dependency graphs. Figure 5.5 demonstrates that all four relevance functions perform consistently above random and have a comparable performance for both the training and test sets. The most striking result is that the class predictability (column 2 in Figure 5.5) improves as we decrease the number of neurons while keeping only the relevant neurons. Figure 5.5 demonstrates the sharp difference between the *static* nature of weights (left & right) and the *dynamic* nature of the other three relevance functions (center). Across all classes the weights function performs best; however, it is not capable of detecting class-specific information. Specifically, Figure 5.5 (center) shows that at 50% ablation, the performance of the weights drastically begins to drop, while the remaining relevance functions increase to a staggering 100% true positive rate (TPR). Figure 5.6 demonstrates that this behaviour occurs consistently across classes for both the training and the test sets. We perform additional experiments with L1-,and L2- regularisation on all layers (norm penalty parameter $\alpha = 0.001$), which yield the same results[7]. Since the 50% mark denotes a major inflexion point, we select a threshold value of 50% for the rest of the experiments.

These findings have three implications. First, the fact that class predictability increases as we remove irrelevant neurons supports the hypothesis that the DGINN framework is capable of identifying class-specific representations because it determines the neurons, pertinent to a particular class, despite the decrease in representation size. Second, the results support the low-dimensional manifolds and natural clustering assumptions since they demonstrate that very few neurons are responsible for the representation of each class. Third, these findings help us understand the degree of sparsity within PDRs, suggesting that 10% of the total layer capacity is enough to represent different classes.

Input Invariance Figures 5.6 & 5.7 present the surprising result that *without retraining* the class-specific dependency graphs enter a "*biased mode*" of operation as the number of parameters decreases. In this *biased mode*, the dependency graphs progressively predict the same output for which they have been specialised, thereby turning into constant classifiers. Figures 5.6 shows that at 20% of the network parameters, 9 of the specialised dependency graphs become *invariant to any input* (see Appendix E Figure E.1 for a visualisation of the dependency graphs across all classes). These results illustrate that the dependency graphs are learning input-invariant representations, which corroborate the idea that the DGINN framework is capable of extracting class-specific representations.

Figure 5.7 demonstrates that the only exception to the specialised dependency graphs

[7] Further details can be found in Appendix E.

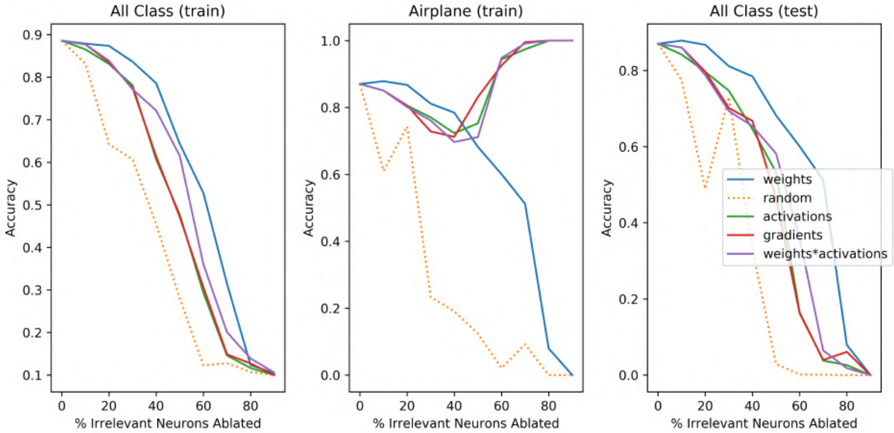


Figure 5.5: A comparison between 4 relevance functions and an additional random function (determining the relevance of neurons arbitrarily) across decreasing thresholds at 10% of all neurons apart. The lines indicate the accuracy of the network after masking all neurons outside the dependency graph. Each column indicates the dataset for which the dependency graph has been specialised and evaluated. For the last column, the dependency graph is specialised for the training set and evaluated on the test set. Notice that across all classes (left & right) the weights function slightly outperforms the rest. However, when the performance is related to a particular class (center), as in the case of airplane, the static nature of the weights function does not allow it to determine the class-specific information.

is the *cat* class, which exhibits a bimodal prediction distribution. 99% of cat predictions are invariant to input, while the remaining less than 1% were dog predictions. Out of this 1%, 32% of the images were correctly classified (i.e., the input was indeed a dog image).

One possible explanation for the bimodal prediction distribution might be that the dependency graphs share sub-structures with semantically similar classes. This explanation is in line with the shared factors assumptions, We present further evidence that supports this hypothesis in Section 5.6. Another possible cause for this discrepancy could be the fact that the cat class has the most substantial generalisation error, which could mean that the dependency graph is indicating the fragility of this classification decision. There is ample room for further progress in determining how the properties of dependency graphs relate to the robustness of predictions for particular classes.

A natural question following these results could be: "*Isn't it natural that if we mask all neurons related to a class, the network will know only about this class?*". The point here is that this would be very natural as long as we have identified precisely the neurons relevant to the class. Hence, since the class is recognised, this is evidence that we have correctly identified the neurons pertinent to that class.

On the flip side, it could be argued that learning a class-conditional input-invariant representations (i.e., constant classifier predicting the same class) will increase the class predictability without learning anything meaningful. We dispute this claim since a class-conditional input-invariant representation is a very natural result of the manifold and

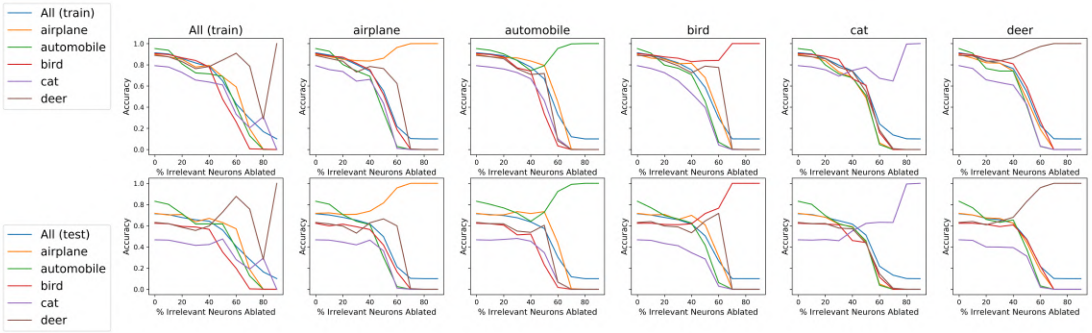


Figure 5.6: A comparison of the true positive rates (TPRs) between 5 class-specific dependency graphs extracted with the gradients relevance function. The lines indicate the accuracy of network once all neurons outside the dependency graph have been masked. Each column indicates the dataset, for which the dependency graph has been specialised. The row indicates whether the datasets belong the training (row 1) or test set (row 2).

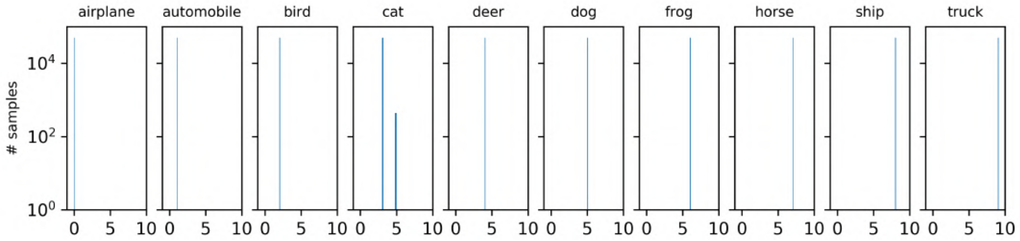


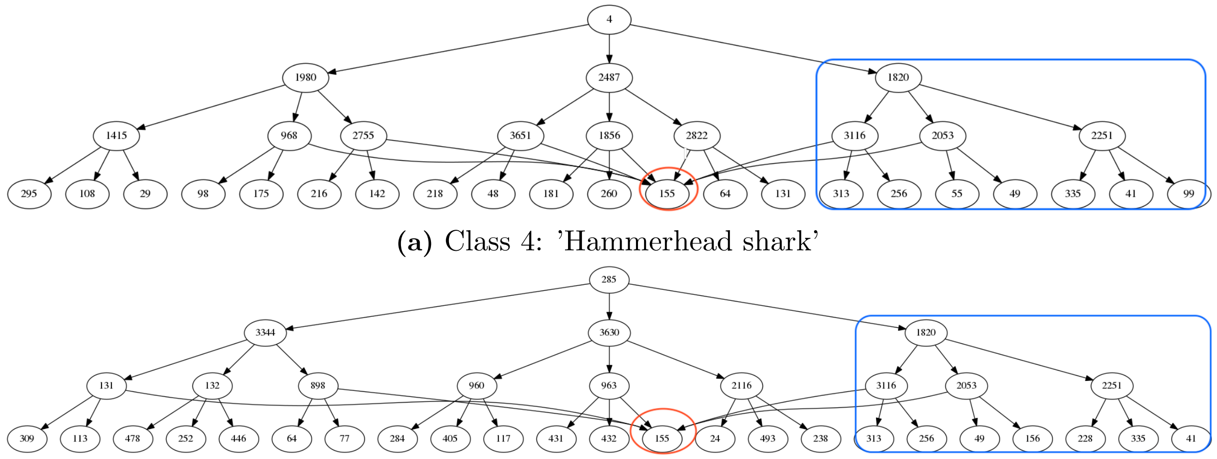
Figure 5.7: The distribution of a cat class-specific dependency graph predictions extracted with gradients at threshold $t = 20\%$. Each column indicates the predictions of a class-specific dependency graph, while the y-axis indicates number of samples and x-axis indicates the predicted class id given the following array: ['airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck']. Notice that all, but the cat class produce the same output for every single data-point.

natural clustering assumptions. An input-invariant representation might correspond to a different class-specific manifold such that any movement along this manifold does not change the output of the network. The class identity would only be changed when we transition across manifolds. We present further evidence that the geometry of the hidden space contains input-invariant representation in Section 6.3.3.1.

Relation to the Lottery Ticket Hypothesis Our results are complementary to the lottery ticket hypothesis study, which proposes that stochastic gradient descent (SGD) seeks out and *trains* a subset of well-initialised weights (Frankle and Carbin, 2018). We take this conjecture one step further by demonstrating that SGD results in class-specific sub-networks, which, without *retraining*, maintain their performance for the corresponding class, and gradually become more biased towards this class, irrespective of the input. The limitation of our study is that it cannot be directly compared to the lottery ticket phenomenon experiments (published in parallel with our research) since the model is not retrained and also because we measure the TPR rather than the accuracy.

5.6 Qualitative Evaluation

In this evaluation, we compare class-specific dependency graphs to investigate the shared factors assumption[8]. Our findings contest the claim that feature importance of individual neurons as proposed by Zintgraf et al. (2017) are a reliable way to debug and analyse the behaviour of neural networks. Concretely, the identification of shared neurons for classes that are not naturally considered similar, portrays the fact that feature importance visualisations of a specific neuron activate equally for distinct input types. Hence, these visualisations cannot be used to make general conclusions about the behaviour of an individual neuron.



(a) Class 4: 'Hammerhead shark'

(b) Class 285: 'Egyptian cat'

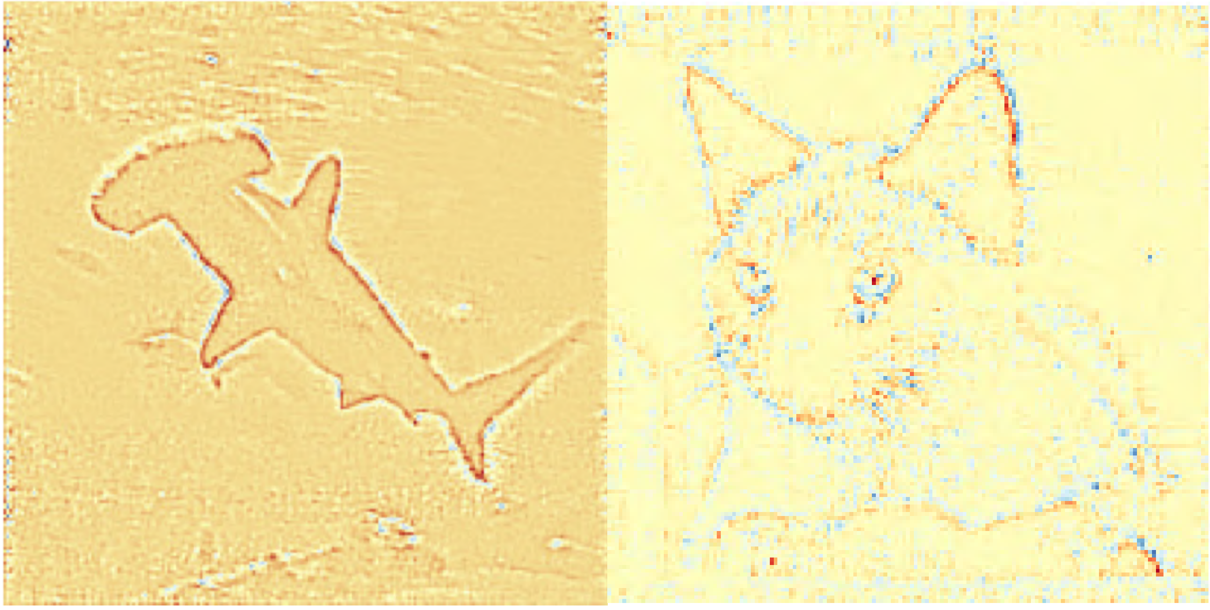
Figure 5.8: Dependency graphs at threshold $t = 5\%$ computed with the gradients relevance function for Hammerhead shark and Egyptian cat classes (penultimate 4 layers of VGG16, excluding the pooling layer) expose a surprising degree of similarity.

Shared sub-graphs We visualise two examples of class-specific dependency graphs in Figure 5.8 to illustrate that classes that are typically considered different may share significant similarities. Specifically, the dependency graphs share 6 out of the 8 most relevant activation maps and relevance connections in `block_5_conv3` (blue rectangle). This finding is in line with the shared factors assumption, demonstrating that there are *shared factors* in the most critical dimensions of their class-specific representations even between *classes* of *arbitrary semantic similarity*. One possible way that the information is represented in latent space could be that some variations in the data lie on shared low-dimensional manifolds in latent space. In contrast, other more class-specific variations could reside on distinct unimodal manifolds (i.e., manifolds that encode only data variations that describe the same class).

[8] See Section 2.2.

Single Shared Neuron Additionally, both dependency graphs share multiple incoming relevance connections to the same neuron f_{155}^{b5c3} (red circle). This neuron is equally important for both classes and forms a part of a shared sub-structure. It might therefore be the case that the neuron encodes a more abstract concept shared between both classes, as expected in the shared factors assumption. What is surprising is that neuron f_{155}^{b5c3} is among the top 3 most important neurons for multiple upper layer neurons and it is the only such neuron. This pattern also occurs exclusively in one out of four portrayed layers (the last convolutional layer). One possible explanation could be that the hard-pruning regime of $t = 5\%$ is eliminating many more, but slightly less relevant patterns of this kind. Another possibility is that dense layers learn more sparser and more specialised representations. In Section 6.3.3.1, we demonstrate that CNNs encode concept values in well-separable regions in the hidden space of their dense layers.

While the investigation in Section 6.3.3.1 leverages concept labels, further work needs to be undertaken to identify concepts in an unsupervised fashion. One way to accomplish this goal could be to identify shared PDRs class-specific dependency graphs using network motifs (Milo et al., 2002). Next



(a) Class 4: 'Hammerhead shark'

(b) Class 285: 'Egyptian cat'

Figure 5.9: Pixel importance heatmaps of activation map f_{155}^{b5c3}, computed using guided-backpropagation. Figures (a) & (b) indicate the importance heatmaps for an image from the corresponding class. Red and blue correspond to respectively positive or negative contribution to the activation of the activation map f_{155}^{b5c3}.

Unreliability of Feature Importance Recently, it has been suggested that one way to debug and understand the behaviour of neural networks is to compute pixel importance heatmaps for particular neurons (Zintgraf et al., 2017). We caution against relying on this approach to gain trustworthy and informative information.

To illustrate, consider the pixel importance heatmaps of neuron f_{155}^{b5c3} in Figure 5.8. According to Figure 5.8, neuron f_{155}^{b5c3} (red circle) plays a vital role in the decisions related to both sharks and cats. This is because the neuron is part of the very few most important elements of both class-specific representations and because it is a shared component for multiple upper-layer neurons.

Figures 5.9a & 5.9b display the pixel importance heatmaps (generated using Springenberg et al. (2015)'s guided backpropagation) of neuron f_{155}^{b5c3}. Had we relied on a single visualisation, we would have erroneously presumed that the neuron perfectly encodes either the concept of a shark or a cat. However, pixel importance heatmaps seem to be equally active for both classes. One possible explanation for this result could be that these heatmaps capture primarily information about edges and contours irrespective of the input (Adebayo et al., 2018). Further results (Raghu et al., 2017) corroborate this hypothesis since they advocate that the lower layers, in which edge-related information is represented, play a much more critical role in the pixel importance computations. One of the issues that emerge from these findings is that neuron investigation in isolation is overly simplistic because it is quite likely that a single neuron is part of a much more complex interaction between multiple units as part of a PDR. Consequently, we propose that investigation of neuron behaviour should be conducted on different input types and in conjunction with other related neurons. DGINN helps identify the sub-sets of neurons that should be studied together.

Semantic Similarity To investigate the semantic similarity between multiple class-specific dependency graphs, we perform clustering analysis. We construct a distance matrix, where the distance is inversely proportional to the number of shared nodes between class-specific dependency graphs computed at threshold $t = 0.5$. Hence, a higher number of shared nodes leads to a smaller distance, so that dependency graphs of two classes with multiple shared nodes are closer together.

We use the UPGMA (unweighted pair group method with arithmetic mean) agglomerative hierarchical clustering method (Sokal, 1958) with Euclidean distance. Figure 5.10 depicts two noteworthy results in the form of a cluster heatmap. First, the objects seem to cluster based on the semantically interpretable dimension animals vs vehicles. Second, it demonstrates the likelihood that semantically meaningful pairs are grouped together – ship & truck (cargo vehicles), cats & dogs (pets), birds & frogs (wild animals). These findings support the hypothesis that the dependency graphs extract semantically relevant information. One way to build on these qualitative results could be to use labelled datasets that explicitly encode the semantic similarity to generate quantitative evaluations. We demonstrate the benefits of having a labelled set of concepts in the next chapter.

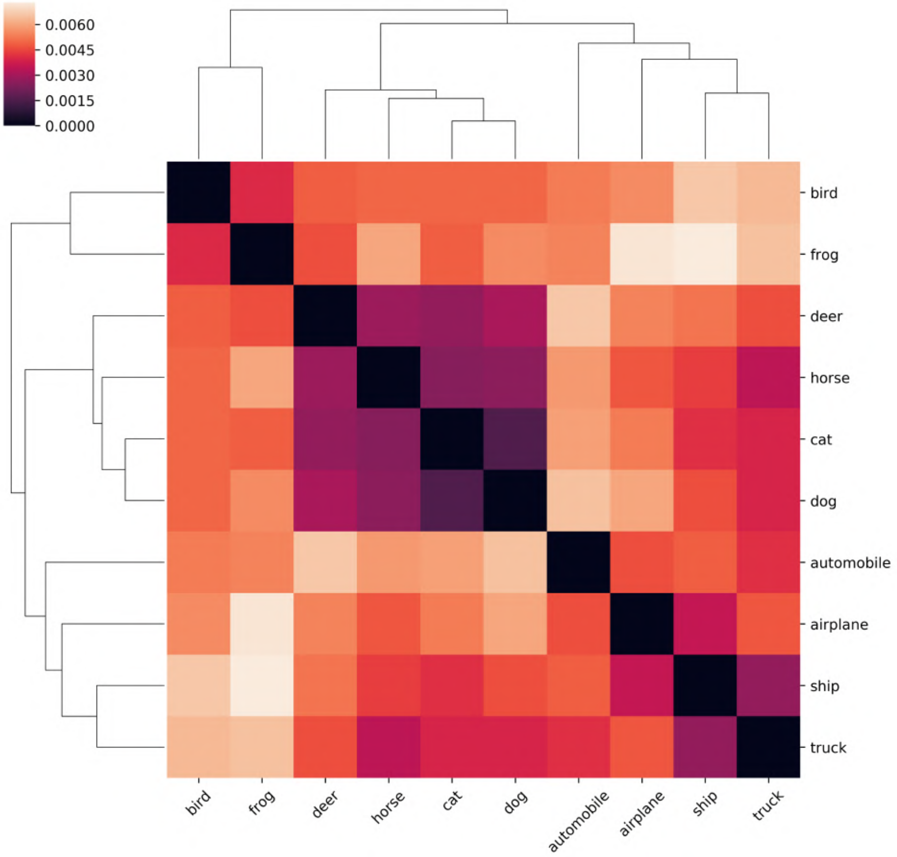


Figure 5.10: A cluster-heatmap between class-specific dependency graphs computed at threshold $t = 50\%$. The tree-like structures adjacent to the heat map indicates the hierarchical relationships between classes, while the colour patches indicate the similarity between each pair of classes. Notice that semantically similar classes are grouped together (e.g., cat and dog, ship and truck).

5.7 Conclusions

In this chapter, we introduce a novel framework for interpreting Deep Neural Networks (DNNs) classification decisions. *(D)ependency (G)raphs for (I)nterpreting (N)eural (N)etworks (**DGINN**)* identifies class-specific representations using model decomposition into a series of functions.

We find that class-specific representations appear within a fraction of the latent space. These class-specific representations seem to capture information about their corresponding class since they can act as binary classifiers for that class. Surprisingly, a subspace of these class-specific representations corresponds to tiny latent space manifolds that are input invariant. These findings give tangible evidence to the sparsity, manifolds, natural clustering, and shared factors assumptions from Section 2.2 and support the conjecture that partially-distributed representations (1) can be identified and (2) contain information pertinent to the decision making process. In the next chapter, we build on these findings and demonstrate that we can extract human-interpretable concepts from partially-distributed representations.

Future work can investigate ways to exploit the approach in areas such as error explanation, adversarial examples detection, or out-of-distribution sample detection by detecting subtle deviations outside the expected class-specific dependency graphs. For example, we could monitor for deviations anomalies from the dependency graph at inference time to detect potential susceptibility to adversarial attacks, when the network is making decisions for the wrong reasons, or to detect out of distribution samples. For instance, an indication that the wrong partially distributed representation is activated (i.e., wrong concept detected) might inform a human operator that a prediction is not trustworthy. The same analysis could be performed at train time to conduct error explanation for misclassified examples or study whether the network has captured robust or brittle features. This analysis can inform a machine learning engineer on how to augment their dataset to mitigate any issues.

CHAPTER 6

Now You See Me (CME): Concept-based Model Extraction

Ask, and it shall be given you;
seek, and ye shall find;
knock, and it shall be opened unto you.

Mathew 7:7 KJV

In this chapter, we continue our investigation of gaining additional insights into the DNN's internal operation in terms of human interpretable concepts using DNN functional decomposition and mappings between PDRs and concepts. Concept-based explanations are superior to feature importance explanations for three main reasons. First, concepts provide explanations at a level of abstraction that is more readily understandable by a human (Kim et al., 2018). They describe meaningful interactions between low-level features, thus achieving a higher level of explanation. Second, concepts can be used to provide both global and local explainability. Since concepts provide explanations for groups of data-points that share common atomic and human understandable characteristics they are an example of semi-local explanations. Since concepts can be used for local, semi-local, and global explainability they can be used more effectively within interactive machine learning applications.

For example, an expert can observe the model behaviour and change concept predictions to influence the model's output effectively. Imagine a doctor, who is including the presence of a particular clinical artefact, which the model did not detect. Another example of the enhanced interactivity enabled by concepts is the development of effective *what-if-tools*, which could allow an expert to ask questions like "what would the output be if a clinical artefact was positioned differently".

In the previous chapter, we demonstrated only the possibility of discovering PDRs

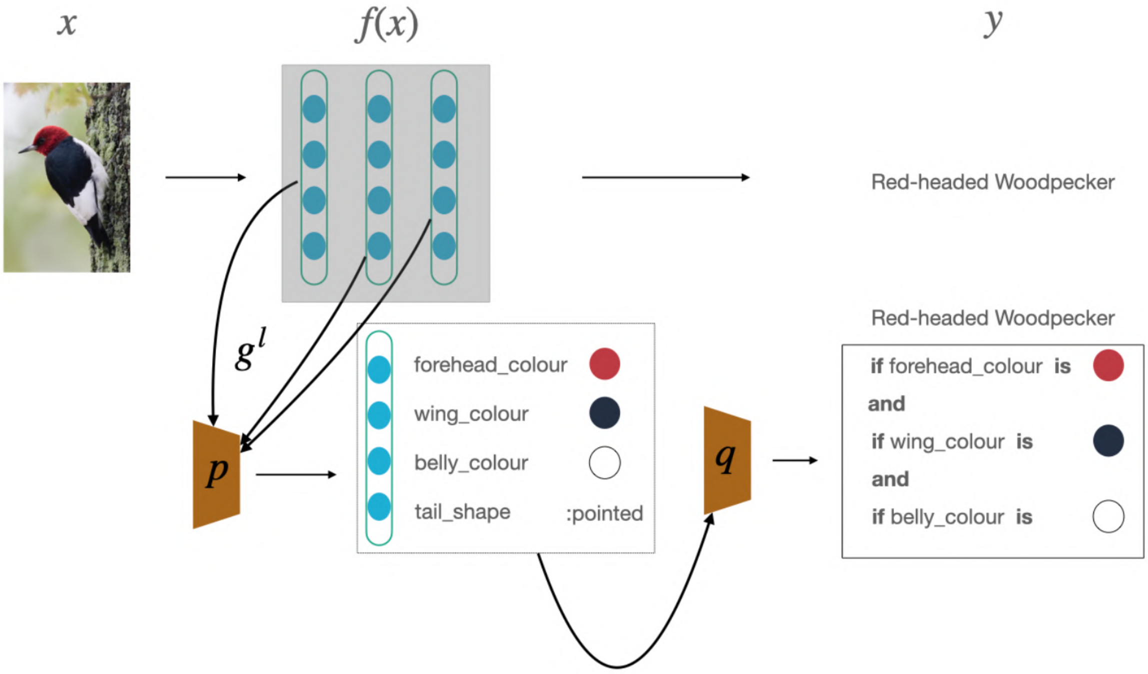


Figure 6.1: Visual summary of CME: (C)oncept-based (M)odel (E)xtraction framework. Given inputs $\mathbf{x}$, a model $\mathbf{y} = f(\mathbf{x})$, and outputs $\mathbf{y}$, we construct a series of functions g^l that take a hidden representation and produce concept labels. The output of these functions is aggregated within a input-to-concept function $p(\mathbf{x})$, which produces concept labels for a given input. These concept labels are consumed by a concept-to-output function q that generates interpretable reasoning behind the model's output. Combining functions p and q results in a new model that approximates the original DNN in a human-understandable way.

that are associated with concepts. In this chapter, we introduce CME[1]: a (C)oncept-based (M)odel (E)xtraction framework. CME generates global explanations of DNN models by approximating DNNs with models grounded in human-understandable concepts and their interactions. Figure 6.1 summarises our approach. Instead of relying on a decomposition of a series of functions, we hypothesise that a DNN can be decomposed into two key functions: one function mapping inputs to concepts, and another function mapping concepts to outputs. This function decomposition extracts a model that approximates the original DNN, while enhancing the richness of interpretations and enabling interactive machine learning applications.

CME takes a step towards quantifying and axiomatising concept-based explanation approaches and might have implications for researchers investigating the psychology of human concept learning. This chapter is the result of joint work with Dmitry Kazhdan, which concluded with a publication (Kazhdan et al., 2020). In particular, we make the following contributions:

[1] Pronounced "See Me."

- We present a novel model extraction framework CME, capable of approximating DNN models with interpretable models that represent their decision-making process using human-understandable concepts.

- We demonstrate, using two use cases, that it is possible to approximate a DNN with a decomposition of two functions. The interpretability and fidelity of these functions can be measured more efficiently, allowing us to compare existing concept-based explanation methods with our novel semi-supervised concept-based extraction technique.

- We demonstrate, using two case-studies, how CME can analyse (both quantitatively and qualitatively) the concept information a DNN model has learned, how this information is represented across the DNN layers, and how a DNN uses concept information when predicting output labels

- Our framework can be used to: (1) provide both **global** (i.e., explaining overall model behaviour) and **local explanations** (i.e., explaining individual predictions) of DNN models through concepts; and (2) investigate the link between the geometry of the hidden space and the information flow in concept space (rather than output space).

6.1 Methodology

In this section, we present our CME approach, describing how it can be used to extract interpretable concept-based models from DNNs. We consider DNN approximation as a function composition of two simpler functions. The first function "translates" from input space to concept space (concept-based explanation), while the second one "translates" from concept space to prediction space (model extraction).

6.1.1 Formulation

We consider a pre-trained DNN classifier $f : \mathcal{X} \to \mathcal{Y}$, ($\mathcal{X} \subset \mathbb{R}^n$, $\mathcal{Y} \subset \mathbb{R}^o$), where $f(\mathbf{x}) = y$ is mapping an input $\mathbf{x} \in \mathcal{X}$ to an output class $y \in \mathcal{Y}$. For every DNN layer l, we denote the function $f^l : \mathcal{X} \to \mathcal{H}^l$, ($\mathcal{H}^l \subset \mathbb{R}^m$) as the mapping from the input space $\mathcal{X}$ to the hidden representation space $\mathcal{H}^l$, where m denotes the number of hidden units, and can be different for each layer.

We assume the existence of a *concept representation* $\mathcal{C} \subset \mathbb{R}^k$, defining k distinct concepts associated with the input data. $\mathcal{C}$ is defined such that every basis vector in $\mathcal{C}$ spans the space of possible values for one particular concept. We further assume the existence of a function $p^\star : \mathcal{X} \to \mathcal{C}$, where $p^\star(\mathbf{x}) = \mathbf{c}$ is mapping an input $\mathbf{x}$ to its concept representation $\mathbf{c}$. Thus, $p^{\star}$ defines the ground-truth concepts and their values for every input point.

6.1.2 Hypothesis

We hypothesise that any DNN model f can be decomposed into functions p and q, such that $f(\mathbf{x}) = q(p(\mathbf{x}))$. In this definition, the function $p : \mathcal{X} \rightarrow \mathcal{C}$ is an *input-to-concept* function, mapping data-points from their input representation $\mathbf{x} \in \mathcal{X}$ to their concept representation $\mathbf{c} \in \mathcal{C}$. The function $q : \mathcal{C} \rightarrow \mathcal{Y}$ is a *concept-to-output* function, mapping data-points in their concept representation $\mathcal{C}$ to output space $\mathcal{Y}$. Thus, when processing an input $\mathbf{x}$, a DNN f can be seen as converting this input into an interpretable concept representation using p, and then using q to predict the output from this representation.

The aim of CME is to approximate the behaviour of f with an extracted model $\hat{f} : \mathcal{X} \rightarrow \mathcal{Y}$, by approximating p and q with $\hat{p}$ and $\hat{q}$, so that $\hat{f}$ is defined as $\hat{f}(\mathbf{x}) = \hat{q}(\hat{p}(\mathbf{x}))$. Next, we describe our approach for extracting $\hat{p}$ and $\hat{q}$ from a pre-trained DNN.

6.1.3 Input-to-Concept ($\hat{p}$)

When extracting $\hat{p}$, we assume we have access to the DNN training data and labels $\{(\mathbf{x}^{(0)}, y^{(0)}), ..., (\mathbf{x}^{(d)}, y^{(d)})\}$. Furthermore, we assume partial access to $p^{\star}$[2], such that a small set[3] of i training points $\{\mathbf{x}^{(0)}, ..., \mathbf{x}^{(i-1)}\}$ have concept labels $\{\mathbf{c}^{(0)}, ..., \mathbf{c}^{(i-1)}\}$ associated with them, while the remaining u points $\{\mathbf{x}^{(i)}, ..., \mathbf{x}^{(i+u)}\}$ do not (in this case $u = d - i$). We refer to these subsets respectively as the *concept labelled dataset* and *concept unlabelled dataset*. Using these datasets, we generate $\hat{p}$ by aggregating concept label predictions across multiple layers of the given DNN model, as described below.

Given a DNN layer l with m hidden units, we compute the layer's representation of the input data $\mathbf{h} = f^{l}(\mathbf{x})$, obtaining $(\mathbf{h}^{(0)}, ..., \mathbf{h}^{(i+u)})$. Using this data and the concept labels, we construct a semi-supervised dataset, consisting of labelled data $\{(\mathbf{h}^{(0)}, \mathbf{c}^{(0)}), ..., (\mathbf{h}^{(i-1)}, \mathbf{c}^{(i-1)})\}$, and unlabelled data $\{\mathbf{h}^{(i)}, ..., \mathbf{h}^{(i+u)}\}$.

Next, we rely on Semi-Supervised Multi-Task Learning (SSMTL) (Liu, Liao, and Carin, 2008), in order to extract a function $g^{l} : \mathcal{H}^{l} \rightarrow \mathcal{C}$, which predicts concept labels from layer l's hidden space. We treat each concept as a separate, independent task. Hence, $g^{l}(\mathbf{h})$ is decomposed into k separate tasks, and is defined as $g^{l}(\mathbf{h}) = (g^{l}_{1}(\mathbf{h}), ..., g^{l}_{k}(\mathbf{h}))$ where each $g^{l}_{i}(\mathbf{h})$ ($i \in \{1..k\}$) predicts the value of concept i from $\mathbf{h}$.

Repeating this process for all model layers L, we obtain a set of functions $G = \{g^{l}_{i} \mid l \in \{1..L\} \ \wedge i \in \{1..k\}\}$. For every concept i, we define the "best" layer l^{i} for predicting that

[2] It is reasonable to expect that a domain expert could label a small number of points (approximately 50 - 100 per concept) that would provide the partial signal for $p^{\star}$.

[3] In Section 6.3, we show that 100 samples suffice to learn $\hat{p}$ at a satisfactory level.

concept as shown in equation 6.1:

$$l^i = \arg\min_{l \in L} \ell(g_i^l, i) \tag{6.1}$$

where ℓ is a loss function (in this case the error rate), computing the predictive loss of function g_i^l wrt to a concept i. Finally, we define $\hat{p}$ as shown in equation 6.2:

$$\hat{p}(\mathbf{x}) = (g_1^{l^1} \circ f^{l^1}(\mathbf{x}), ..., g_k^{l^k} \circ f^{l^k}(\mathbf{x})) \tag{6.2}$$

Thus, for every concept $i \in \{1..k\}$, given an input $\mathbf{x}$, the value computed by $\hat{p}(\mathbf{x})$ is equal to the value computed by $g_i^{l^i}$ from that input's hidden representation in layer l^i. Overall, $\hat{p}$ encapsulates concept information contained in a given DNN model, and can be used to analyse how this information is represented, as well as to predict concept values for new inputs.

6.1.4 Concept-to-Label ($\hat{q}$)

We set extraction of $\hat{q}$ as a classification problem, in which we train $\hat{q}$ to predict output labels y from concept labels $\mathbf{c}$. We use $\hat{p}$ to generate concept labels for all training data points, obtaining a set of concept labels $\{\mathbf{c}^{(0)}, ..., \mathbf{c}^{(i+u)}\}$. Next, we produce a labelled dataset, consisting of concept labels and corresponding DNN output labels $\{(\mathbf{c}^{(0)}, y^{(0)}), ..., (\mathbf{c}^{(i+u)}, y^{(i+u)})\}$, and use it to train $\hat{q}$ in a supervised manner. We experimented with using Decision Trees (DTs), and Logistic Regression (LR) models for representing $\hat{q}$, as discussed in Section 6.3. Overall, $\hat{q}$ can be used to analyse how a DNN uses concept information when making predictions.

6.2 Experimental Set-up

We use two case studies – dSprites (Matthey et al., 2017), and Caltech-UCSD birds (Wah et al., 2011), which have slightly different set-ups in terms of classification tasks, models, and concept labels. Next, we discuss each use case separately. Afterwards, we describe the benchmarks, against which we compare our concept-based model extraction technique.

6.2.1 dSprites Dataset

The dSprites dataset (Matthey et al., 2017) is a well-established dataset for unsupervised latent factor disentanglement. dSprites is a dataset of 2D shapes, procedurally generated from 6 ground truth independent concepts. Table 6.1 lists the concepts, and corresponding values. Figure 6.2 presents some examples. dSprites consists of 64 × 64 pixel black-and-white images, generated from all possible combinations of these concepts, for a total of

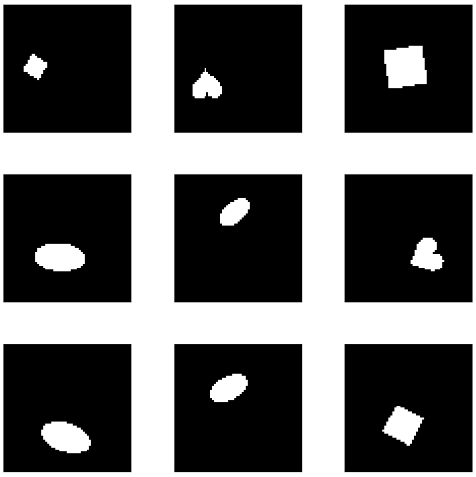


Figure 6.2: Example images from the dSprites dataset.

$1 \times 3 \times 6 \times 40 \times 32 \times 32 = 737280$ total images.

Table 6.1: dSprites concepts and values

Name	**Values**
Color	white
Shape	square, ellipse, heart
Scale	6 values linearly spaced in $[0.5, 1]$
Rotation	40 values in $[0, 2\pi]$
Position X	32 values in $[0, 1]$
Position Y	32 values in $[0, 1]$

For computational reasons, we down-sample the dataset to 36864, while preserving its statistical properties, such as concept value ranges and diversity. We retain only 16 of the 32 values for *Position X* and *Position Y* (keeping every other value only), and retain only 8 of the 40 values for *Rotation* (retaining every 5th value).

Classification Tasks We define 2 classification tasks, used to evaluate our framework:

- **Task 1**: This task consists of determining the shape concept value from an input image. For every image sample, we define its task label as the shape concept label of that sample.
- **Task 2**: This task consists of discriminating between all possible *shape* and *scale* concept value combinations. We assign a distinct identifier to each possible combination of the shape and scale concept labels, resulting in $6 \times 3 = 18$ classes. For every image sample, we define its task label as the identifier corresponding to this sample's shape and scale concept values.

These tasks permit us to explore the quality of models extracted by CME when used in progressively more complex scenarios. Task 1 explores a scenario in which a DNN has to learn to recognise a specific concept. Task 2 explores a more complex scenario, in which a DNN has to learn to recognise combinations of concepts.

Model We trained a Convolutional Neural Network (CNN) model (LeCun et al., 1990) for each task. Both models had the same architecture, consisting of 3 convolutional layers, 2 dense layers with ReLUs, 50% dropout (Srivastava et al., 2014) and a softmax output layer. The models were trained using categorical cross-entropy loss, and achieved $100.0 \pm 0.0\%$ classification accuracies on their respective held-out test sets. We refer to these models as the *Task 1 model* and the *Task 2 model* in the rest of this work.

Ground-truth Concept Information Importantly, the task and dataset definitions described in this section imply that we know precisely which concepts the models had to learn, in order to achieve $100.0 \pm 0.0\%$ task performances (*shape* for Task 1, and *shape* and *scale* for Task 2). We refer to this as the *ground truth* concept information learned by these models.

6.2.2 Caltech-UCSD Birds (CUB)

For our second dataset, we used Caltech-UCSD Birds 200 2011 (CUB) (Wah et al., 2011). This dataset consists of 11,788 images of 200 bird species with every image annotated using 312 binary concept labels (e.g., beak and wing colour, shape, and pattern). We relied on concept pre-processing steps defined in (Koh et al., 2020) (used for de-noising concept annotations, and filtering out outlier concepts), which produces a refined set of $k = 112$ binary concept labels for every image sample.

Classification Task We relied on the standard CUB classification task, which consists of predicting the bird species from an input image.

Model We used the Inception-v3 architecture (Szegedy et al., 2016), pretrained on ImageNet (Krizhevsky, Sutskever, and Hinton, 2012) (except for the fully-connected layers) and fine-tuned end-to-end on the CUB dataset, following the preprocessing practices described in (Cui et al., 2018). The model achieved $82.7 \pm 0.4\%$ classification accuracy on a held-out test set. We refer to this model as the *CUB model* in the rest of this work.

Ground-truth Concept Information Unlike dSprites, the CUB dataset does not explicitly define how the available concepts relate to the output task. Thus, we *do not* have access to the ground truth concept information learned by the CUB model. Instead, we use human concept-annotations such as wing colour and tail shape, which are provided as part of the dataset. The annotations describe the inputs, but do not necessarily describe the relationship between the concepts and the classification task.

Additionally, in contrast, to the dSprites concept labels, the concept annotations for CUB are binary. We use this dataset to benchmark directly against the Concept Bottleneck Model (CBM) and to illustrate that CME can handle both multi-valued and binary concept labels, contrary to other approaches.

6.2.3 Benchmarks

Net2Vec We benchmark the $\hat{p}$ functions for the three tasks against Net2Vec (Fong and Vedaldi, 2018). As discussed in Section 3.4.4.1, Net2Vec attempts to predict presence/absence of concepts from spatially-averaged hidden layer activations of convolutional layers of a CNN model. Given a binary concept c, this approach trains a logistic regressor, predicting the presence/absence of this concept in an input image from the latent representation of a given CNN layer. In case of multi-valued concepts, the concept space has to be binarised. For instance, given a concept such as “shape”, with possible values ‘square’ and ‘circle’, these approaches have to convert “shape” into two binary concepts ‘is_square’, and ‘is_circle’. For a fair comparison with Net2Vec, for each concept value, we split the labelled training points set into a positive set, containing instances of a particular concept value, and a negative set containing all other examples. In this case, the binarised concept value with the highest likelihood is returned.

Unlike CME, Net2Vec does not provide a way of selecting the convolutional layer to use for concept extraction. We consider the best-case scenario by selecting the convolutional layers yielding the best concept extraction performance. For all tasks, these layers were convolutional layers closest to the output (the 3rd conv. layer in case of dSprites tasks, and the final inception block output layer in case of the CUB task).

Concept Bottleneck Model (CBM) As discussed in Section 6.2.2, we do not have access to ground truth concept information between the concepts and outputs for the CUB model. Instead, we define an upper bound on the amount of concept information available using a pre-trained *sequential bottleneck model* defined in Koh et al. (2020) (referred to as CBM in the rest of this work). CBM is a bottleneck model, obtained by resizing one of the layers of the CUB model to match the number of concepts provided (we refer to this as the *bottleneck layer*), and training the model in two steps. First, the input-to-concept sub-model, consisting of the layers between the input layer and the bottleneck layer (inclusive), is trained to predict concept values from input data. Next, the concept-to-output sub-model, consisting of the layers between the layer following the bottleneck layer and the output layer, is trained to predict task labels from the concept values predicted by the input-to-concept sub-model. Since this bottleneck model is explicitly trained to rely on concept information when making task label predictions, it serves as an *upper bound* for the concept information learnable from the dataset, and for the task performance achievable using this information. A key difference between CME and CBM, is that CBM does not attempt to approximate, or analyse, the CUB model behaviour, but instead attempts to solve the same classification task using concept information only.

We use the input-to-concept CBM submodel as a $\hat{p}$ benchmark, representing the upper bound of concept information learnable from the data. We use the output-to-concept submodel as a $\hat{q}$ benchmark, representing the upper bound of task performance achievable from only predicted concept information. Finally, we use the entire model as an $\hat{f}$ benchmark. We make use of the saved trained model from Koh et al. (2020), available in the official repository[4].

6.3 Results

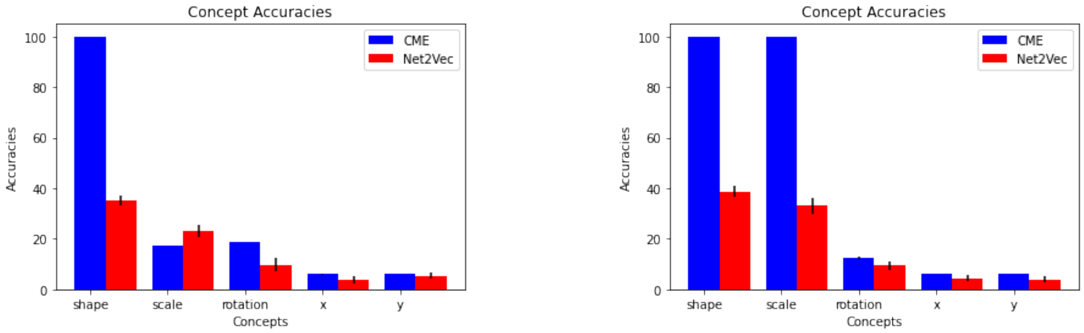
This section presents the results obtained by evaluating our approach using the two case studies described above. Section 6.3.1 measures the concept prediction performance of $\hat{p}$. Section 6.3.2 measures the end-to-end task performance of $\hat{f}$. Section 6.3.3 performs inspection of our extracted models and their constituent parts to provide insights into the behaviour of the original model.

We obtain the concept labelled dataset by returning the ground-truth concept values for a random set of samples in the model training data. For dSprites, we found that a concept labelled dataset of a 100 samples or more worked well in practice for both tasks. Thus, we fix the size of the concept labelled dataset to 100 in all of the dSprites experiments. For CUB, we found that a concept labelled dataset containing 15 or more samples per concept class worked well in practice. Thus, we fix the size of the concept labelled dataset to 15 samples per concept class in all of the CUB experiments.

6.3.1 Concept Prediction Performance – Input-to-Concept ($\hat{p}$)

First, we evaluate the quality of $\hat{p}$ functions produced by CME, Net2Vec, and CBM by measuring their predictive performance on concept labels using a held-out sample test set. For both dSprites tasks, we relied on the *Label Spreading* semi-supervised model (Zhou et al., 2004), provided in scikit-learn (Pedregosa et al., 2011), when learning the g_i^l functions for CME. For CUB, we used logistic regression functions instead, as they gave better performance.

[4] https://github.com/yewsiang/ConceptBottleneck



(a) Task 1

(b) Task 2

Figure 6.3: Evaluation of $\hat{p}$. We show the predictive accuracy of $\hat{p}$, computed using our approach and Net2Vec, for every concept and task averaged over 5 runs.

dSprites Figure 6.3 shows predictive performance of the $\hat{p}$ functions on all concepts for the two dSprites tasks (averaged over 5 runs). As discussed in Section 6.2.1, we have access to the ground-truth task relevant concept information (i.e., *shape* concept information for Task 1, and *shape* and *scale* concept information for Task 2).

For both tasks, $\hat{p}$ functions extracted by CME successfully achieved high predictive accuracy on concepts relevant to the tasks, whilst achieving a lower performance on concepts irrelevant to the tasks. Thus, CME was able to successfully extract the concept information contained in the task models. This finding also illustrates the selective salience property of the internal representations[5], demonstrating that the information, which is not relevant to the task, is not learned by the original model. Additionally, for both tasks, Net2Vec achieved a much lower performance on the relevant concepts, depicting that the superiority of CME's ability to dynamically determine the layers most pertinent to particular concepts. Notice that we do not report CBM performance because CBMs are not defined for multi-valued concepts.

CUB As discussed in Section 6.2.2, the CUB dataset does not explicitly define how the concepts relate to the output task labels. Thus, we do not know how relevant different concepts are to the task label prediction. In this section, we make the conservative assumption that all concepts are relevant, when evaluating $\hat{p}$ functions.

We rely on the "average-per-concept" metrics introduced in Koh et al. (2020) when evaluating the $\hat{p}$ function performances. That is, we compute the $F1$ predictive scores for each concept, and then average over all concepts. We obtained $F1$ scores of 92 ± 0.5%, 86.3 ± 2.0%, and 85.9 ± 2.3% for CBM, CME, and Net2Vec $\hat{p}$ functions, respectively (averaged over 5 runs). We observe that CME performs slightly, but not significantly better than Net2Vec. Interestingly, both approaches achieve performance that is substantially lower than the upper bound of CBM. There could be two possible explanations for this

[5] For more information see Appendix B.2.

result: (1) not all concepts are relevant to the task; hence, the original models are not learning anything about these concepts because the models are not explicitly trained to recognise concepts, so these concepts cannot be extracted; (2) the relationship between hidden representations and concepts is non-linear, which means that the g_i^{li} functions are too simple to capture this behaviour. Therefore, we argue that for fair comparison of concept-based explanations it is crucial to measure the concept prediction performance on relevant concepts only.

Moreover, we argue that in case of a large number of concepts, it is crucial to measure how concept mis-predictions are distributed across the test samples. For instance, consider a dSprites Task 2 $\hat{p}$ function that achieves 90% predictive accuracy on both *shape* and *scale* concepts. The average predictive accuracy on relevant concepts achieved by this $\hat{p}$ will therefore be 90%. However, if the two concepts are mis-predicted for strictly different samples (i.e., none of the samples have both *shape* and *scale* predicted incorrectly at the same time), this means that 20% of the test samples will have one relevant concept predicted incorrectly. Given that both concepts need to be predicted correctly when using them for task label prediction, this implies that consequent task label prediction will not be able to achieve over 80% task label accuracy. This effect becomes even more pronounced in case of a larger number of relevant concepts. Consequently, we suggest that future work in concept-based explanations should develop specific metrics that take into account the number of correctly classified relevant concepts.

6.3.2 Task Performance – End-to-End ($\hat{f}$)

In this section, we evaluate the fidelity and performance of the extracted $\hat{f}$ models. For all CME and Net2Vec $\hat{p}$ functions evaluated in the previous section, we trained output-to-concept functions $\hat{q}$, predicting class labels from the $\hat{p}$ concept predictions. Next, for every $\hat{p}$, we defined its corresponding $\hat{f}$ as discussed in Section 6.1, via a composition of $\hat{p}$ and its associated $\hat{q}$. For every $\hat{f}$, we evaluated its fidelity and its task performance, using a held-out sample test set. Table 6.2 shows the fidelity of extracted models, and Table 6.3 shows the task performance for these models (averaged over 5 runs). The original Task 1, Task 2, and CUB models achieved task performances of 100±0%, 100±0%, and 82.7±0.4%, respectively, as described in Section 6.2.

For both dSprites tasks, CME $\hat{f}$ models achieved high (99%+) fidelity and task performance scores, indicating that CME successfully approximated the original dSprites models. Furthermore, these scores were considerably higher than those produced by the Net2Vec $\hat{f}$ models.

For the CUB task, both CME and Net2Vec $\hat{f}$ models achieved relatively lower fidelity and task performance scores (in this case, performance of CME was very similar to that of Net2Vec). Crucially, the upper bound of CBM *also* achieved relatively low fidelity

Table 6.2: Fidelity of extracted $\hat{f}$ models. Note that CME has been weakly-supervised (100 concept labels for dSprites & 225 concept labels for CUB), whereas CBM has been fully-supervised.

	CME	**CBM**	**Net2Vec**
dSprites Task 1	100.0±0.0%	–	24.5±3.6%
dSprites Task 2	99.3±0.5%	–	38.3±4.0%
CUB	74.42±3.1%	77.5±0.2%	73.8±2.8%

Table 6.3: Task performance of extracted $\hat{f}$ models. Note that CME has been weakly-supervised (100 concept labels for dSprites & 225 concept labels for CUB), whereas CBM has been fully-supervised.

	CME	**CBM**	**Net2Vec**
dSprites Task 1	100.0±0%	–	24.5±3.6%
dSprites Task 2	99.3±0.5%	–	38.3±4.0%
CUB	70.8±1.8%	75.7±0.6%	69.8±1.5%

and accuracy scores. This implies that concept information learnable from the data is insufficient for achieving high task accuracy. These findings imply that the relatively high CUB model accuracy has to be caused by the CUB model relying on other non-concept information. Consequently, the low fidelity of CME and Net2Vec is a consequence of the CUB model being not completely *concept-decomposable* given the available concepts, indicating that it's behaviour cannot be explained by the desired concepts.

6.3.3 Explainability

We present several ways to analyse $\hat{p}$ and $\hat{q}$ to characterise the behaviour of the original model f. Since the two functions can be studied separately, we gain additional insights about what concept information the original model learned and how this concept information is used to make predictions.

Overall, inspection of $\hat{p}$ and $\hat{q}$ can increase our understanding of the global behaviour and decision-making process of a model. Furthermore, by observing the outputs of both $\hat{p}$ and $\hat{q}$ on a single new data-point $\mathbf{x}$, we can also obtain local explanations for specific model predictions. Here we present a case study on dSprites because of the more manageable number of concepts, classes, and model size, whereas Appendix F gives more details on CUB.

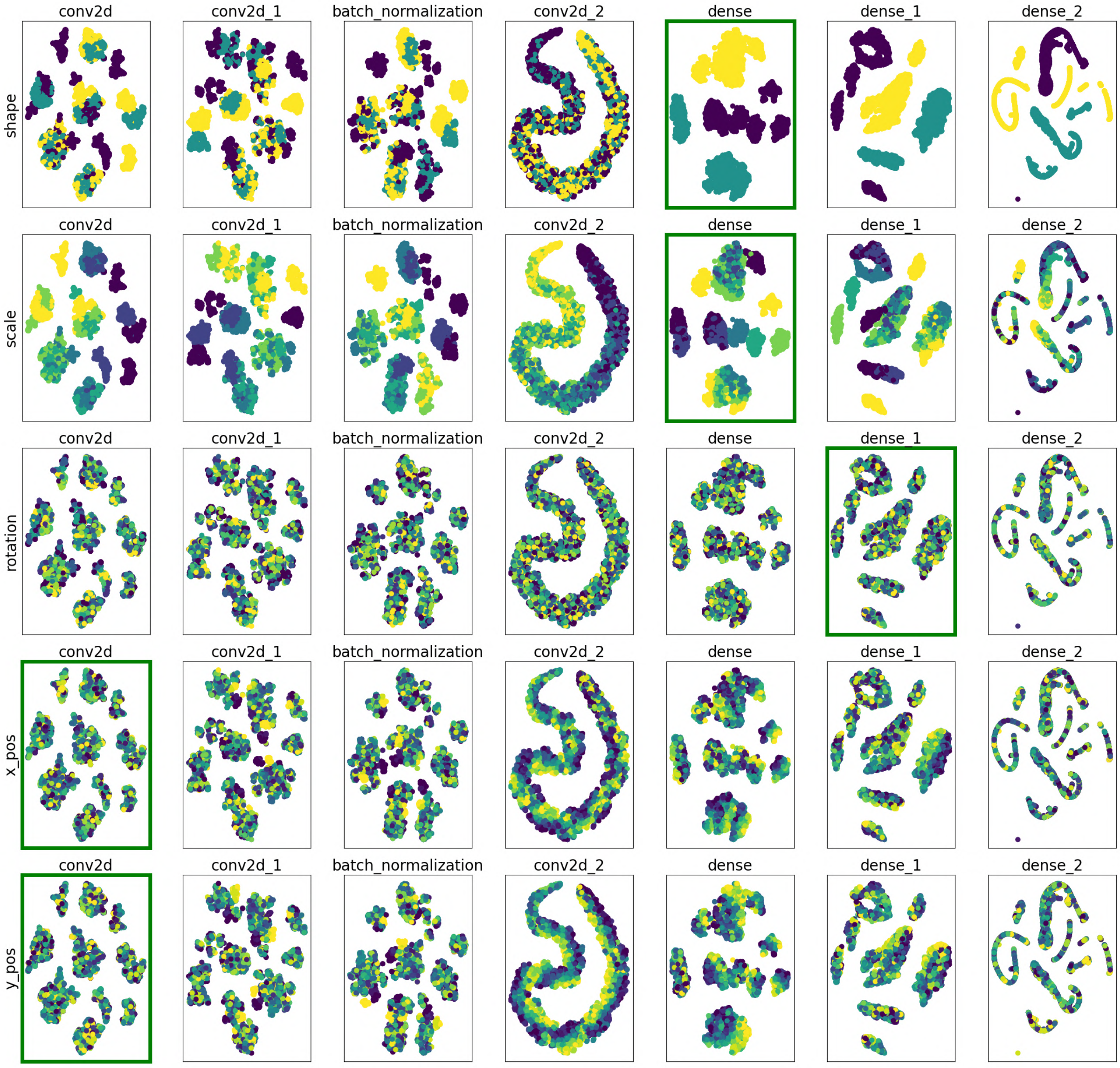


Figure 6.4: Concept labels across the layers of the dSprites Task 2 model in t-SNE 2D projected hidden space. Each row corresponds to a different concept, each column corresponds to a different layer, and colour represents different concept labels. For every concept row, the subplot with a green border indicates the layer $\hat{p}$ uses for predicting the value of that concept. Notice that the concepts get progressively easier to separate with layers closer to the output.

6.3.3.1 Input-to-Concept ($\hat{p}$)

Here we inspect $\hat{p}$ and the layers $\hat{p}$ utilises for concept prediction to explore the relationship between the concept space ($\mathcal{C}$) and the hidden space of the DNN layers.

Figure 6.4 shows a t-SNE (Maaten and Hinton, 2008) 2D projected plot of every layer's hidden space of the Task 2 model, highlighting different concept values. This analysis is complementary to existing approaches for hidden space analysis (see Section 3.6). Three important findings stand out in Figure 6.4: (1) there are different types of manifolds; (2) higher layers disentangle concept values; (3) the highest separability of concept values occurs at different layers. We discuss each of these next.

Types of Manifolds Not surprisingly latent space manifolds come in different shapes and sizes, but a rather surprising result is that manifolds come in different types, which we term "*blobs*" and "*paths*". A blob manifolds contain a distinct concept value (see row `shape`, columns `dense`, `dense_1`, `dense_2`), while path manifolds represent the variation of the entire concept along the manifold structure (see row `scale`, column `conv2d_2`). For example, the results for *scale* concept (row 2, columns 4 & 5) demonstrate a limitation of linear and clustering-based concept extraction approaches, since the scale variation is well-represented across the manifold structure, but in a curved, non-linear way.

Smooth, Spread-out, and Unimodal Manifolds In accordance with previous studies (Kim et al., 2018; Bengio and Delalleau, 2011; Bengio et al., 2013), Figure 6.4 (rows `shape`,`scale`) illustrates that the manifolds of higher layers become smoother (less curved), more spread-out (taking more continuous space), and more unimodal (correspond to a single concept value). Smoother, spread-out and unimodal manifolds facilitate the interpolation between high-probability samples, making classification of unseen samples possible. These findings confirm the supposition that DNNs are capable of disentangling highly curved input manifolds into flat hidden space manifolds (Poole et al., 2016).

Additionally, our results suggest that multiclustering properties (overlapping clusters and partial membership) are more likely at the layers closest to the input, since the hierarchical organisation has not yet built features that are invariant to all uninformative variations in the data, as theoretically predicted (Kim et al., 2018; Bengio and Delalleau, 2011; Bengio et al., 2013). Hence, single class membership with well-separated concept values emerges closer to the output layers.

Concept separability and invariance vary across the layers The concept representation varies significantly across layers and that highest separability (see Figure 6.5) across all concepts is not necessarily achieved in a single layer.

Similarly to TCAV (Kim et al., 2018), we find that the separability of relevant concepts (e.g., shape, and scale) increases in higher layers of the network. In contrast, and in line with Bengio (2009), we find that the network gradually develops an invariance towards irrelevant concepts (e.g., position) as depth increases in the absence of skip connections. These findings imply that it is beneficial to consider multiple layers simultaneously, when performing concept extraction, instead of focusing on a single layer, as is done in existing work.

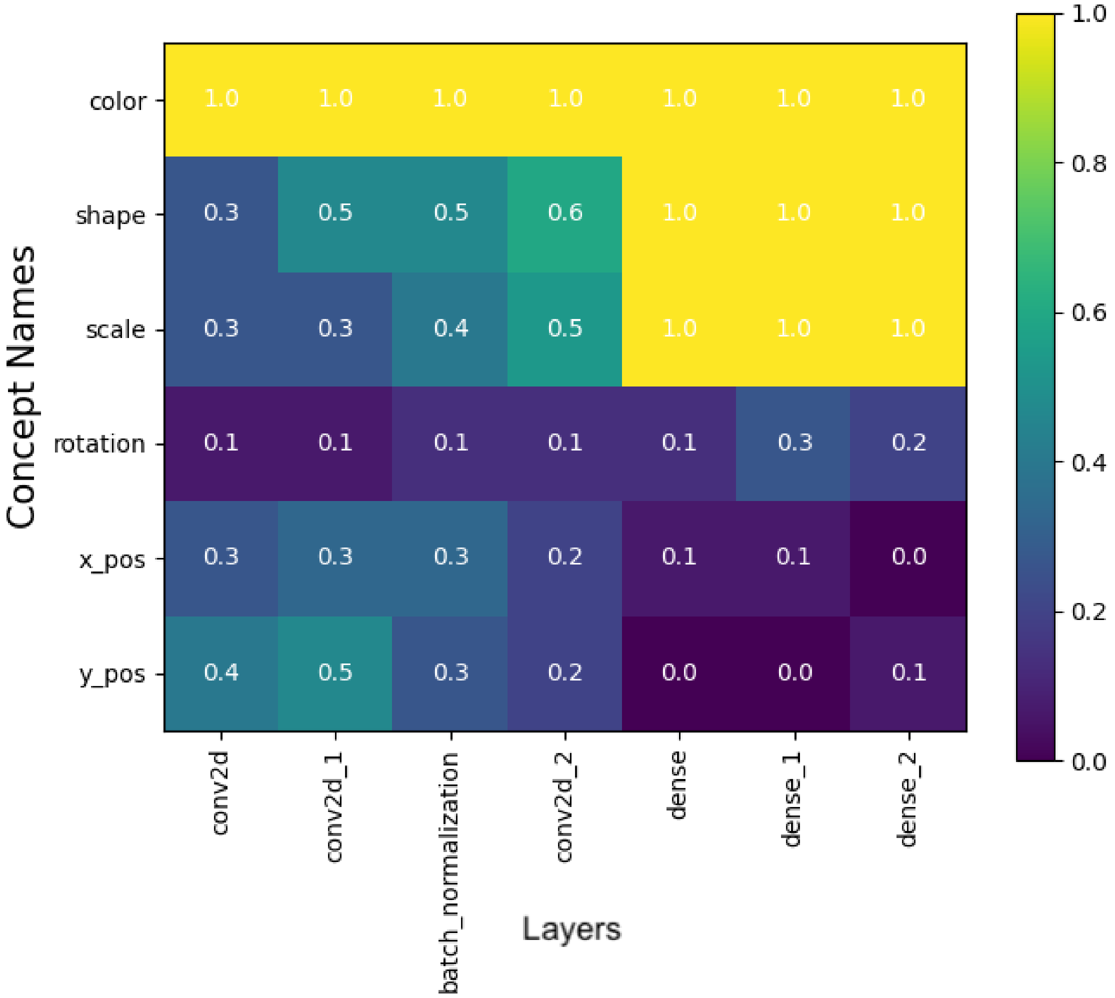


Figure 6.5: Visualisation of g^l for every layer (l) and concept of the dSprites Task 2 model. Each cell represents the accuracy of g^l for a particular concept (rows) at a specific layer (columns). Notice that some concepts are more predictable than others.

6.3.3.2 Concept-to-Output ($\hat{q}$)

An analysis of $\hat{p}$ and $\hat{q}$ can be used to inspect the global behaviour of a DNN model, building an understanding of not only which concepts the DNN learns to extract, but also how the DNN uses these concepts for classification.

The concept-to-output functions ($\hat{q}$) are classifiers trained to predict output labels from concept labels. As discussed in Section 6.1, we can choose the $\hat{q}$ functions to be more easily interpretable (e.g., linear models, decision trees, or decision list). Hence, these functions can more easily communicate how a DNN uses concept information when making predictions to build an understanding of model behaviour. We can analyse or plot the behaviour of $\hat{q}$ (e.g., inspect the coefficients of the linear model or plotting the decision tree rules). Figure 6.6 presents one example of this analysis, which can provide insights into how $\hat{f}$ uses the concepts during it decision-making process. Specifically, Figure 6.6

portrays that the $\hat{q}$ function used by CME for Task 1 in the form of a decision tree. The decision tree provides a global interpretation of the model behaviour.

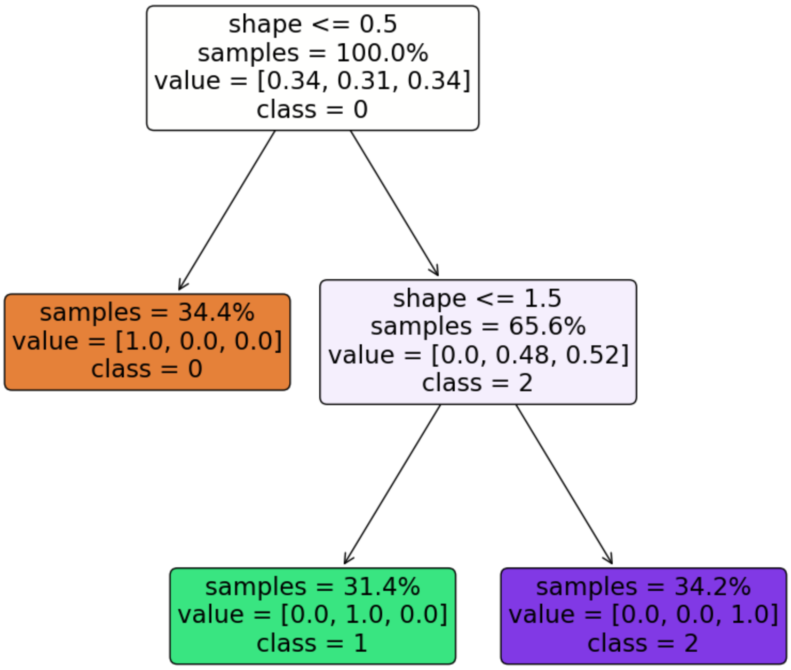


Figure 6.6: Visualisation of a decision tree $\hat{q}$ extracted for Task 1 on dSprites. Notice that the leaves have correctly learned to differentiate between the classes based on the concept of shape.

In addition to global interpretability, we can use our approach to achieve local interpretability of DNN models, allowing us to inspect their instance-specific prediction. After we approximate a DNN by $\hat{f}$, any prediction produced by $\hat{f}$ can be directly traced back to concepts recognised by its corresponding $\hat{p}$, and to functional relation between concepts and the output class label, represented by $\hat{q}$. Finally, concept explanations describe the expected model behaviour across well-specified groups of data points. This allows us to make more fine-grained inferences about the expected output for sub-populations of instances with greater trust and comprehensibility. The semi-local explanation is more trustworthy than a local explanation because it is more likely to hold for a wider range of circumstances. Additionally, it is more informative than a global explanation because it can elucidate edge cases or clusters of points, for which the model behaviour deviates from the typical case.

Overall, the inspection of $\hat{q}$ functions can be used for (i) verifying that a DNN uses concept information correctly during decision-making, and that its high-level behaviour is consistent with user expectations using simple observations of the extracted model or conducting what-if-analysis (*model verification*), (ii) identifying specific concepts or concept interactions (if any) causing incorrect behaviour (*model debugging*), (iii) extracting new knowledge about how concept information can be used for solving a particular task (*knowledge extraction*), (iv) modifying the behaviour of the model in run-time by interactively changing the values of incorrectly predicted concepts (intervening).

6.4 Conclusions

In this chapter, we proposed a novel framework for interpreting neural networks in the medium of concept-based explanations: (C)oncept-based (M)odel (E)xtraction framework (**CME**). In contrast to DGINN, CME extracts concept-based representations using model extraction through functional decomposition of two functions rather than a series of multiple functions. Both CME and DGINN move the field of interpretability one step forward on the levels of explainability beyond *importance* into the realm of *functional relationship* description. We argue that to continue to evolve, the field of XAI has to continue to move up the ladder of levels of explanation sophistication described in Section 3.3.2. Concept-based explanations are the first form of semi-local explanations, and as such they form an essential part of future interactive machine learning systems.

The findings presented here will also be of interest to researchers aiming to quantify and axiomatise concept-based explanation approaches because we cast the field into a well-defined mathematical formulation and propose a way to compare alternative techniques. Finally, our study raises questions regarding the psychology of human concept learning. In Section 3.4.4.1, we discussed the computational, statistical, and cognitive advantages of mathematically representing the concept space as a set of dimensions encoding the variation for a single concept type rather than as a one-hot encoding of all possible concept values. More research using controlled human experiments is needed to investigate which of the two, if any, is the more realistic and user-friendly definition.

Limitations Two limitations of this study are that (1) we assume the availability of a fixed set of k concept labels before model extraction begins, and (2) we assume we know the concept space. In Section 6.3.3.1, we demonstrated that CNNs encode concept values in well-separable regions in the hidden space of their dense layers. This result suggests that unsupervised or active learning approaches could be a fruitful area for further work in alleviating these challenges.

Future Work A natural progression of this work is to explore techniques to reduce the costs of extracting and labelling concepts. Active-learning approaches can be used to obtain maximally-informative concept labels from the user in an interactive fashion. On the other hand, when the number of concepts is unknown $\hat{p}$ has to be extracted in an unsupervised fashion. One way to identify concepts in an *unsupervised fashion* could be to identify shared PDRs across class-specific dependency graphs using network motifs (Milo et al., 2002) or other pattern matching techniques. We hypothesise that there is a high likelihood a shared PDR corresponds to a concept or particular concept value.

CHAPTER 7

Conclusions & Future Work

Good artists copy. Great artists steal.
Real artists ship.

Steve Jobs

7.1 Conclusions

This thesis set out to investigate two hypotheses: (1) the inadequacy of importance-based explanations to describe the behaviour of deep learning models with sufficient fidelity and semantic richness; and (2) the development of specialised explanation methods that can explain in a cognitively better way the information captured in distributed representation of DNNs.

In Chapter 3, we introduced a new taxonomy of explainability methods that takes into account the level of semantic information provided from a particular interpretability method, so that we can assess the semantic richness of explanation methods (see Section 3.3.2). Our taxonomy identifies four main limitations of existing approaches: (1) the lack of semi-local explainability; (2) excessive focus on unit-wise and layer-wise techniques, despite evidence suggesting partially distributed representations; (3) interpretation limited to the input-output relationships to the exclusion of intermediate pieces of information, such as concepts; and (4) we introduce guidelines to measure the sophistication of explanation to show that existing methods focus exclusively on level 1 explainability (i.e., feature importance). We emphasise the statistical Ghorbani, Abid, and Zou, 2019; Kindermans et al., 2019, adversarial Adebayo et al., 2018; Dimanov et al., 2020 and cognitive Poursabzi-Sangdeh et al., 2018; Kim et al., 2018 limitations of feature importance explanations.

We demonstrated additional limitations of feature importance methods in Chapter 4.

Consequently, we proposed class-specific and mathematical concept-based explanations that are extracted from groups of neurons within relevant layers in Chapters 5 (DGINN) and 6 (CME). While DGINN identifies the parts of the network associated with different concepts to provide semi-local level 3 explainability, CME extracts both concepts and the functional relationship between concepts and outputs to provide local and global level 4 explainability. Hence, concept-based explanations provide local, semi-local, and global explanations to move the level of explainability to level 4, in which the role of feature interactions and their relationship to the outcome are more readily understandable. Next, we summarise each contribution.

Adversarial model perturbations to manipulate explanations More concretely, Chapter 4 examined the fidelity of explanation methods to demonstrate that many feature importance explanation methods used in real-world settings are not able to indicate reliably whether or not a model is fair. We provided both theoretical intuition why this is possible and a practical method to modify an existing model to downgrade the feature importance of key sensitive features across seven explanation methods with little effect on model accuracy.

Concept-based explanations As an alternative to feature importance explanation, we propose that concept-based model extraction techniques based on function decomposition and layer-wise model extraction (rather than input-output analysis) yield model interpretations of higher fidelity that are semantically more meaningful. Therefore, we introduced two novel frameworks for interpreting neural networks using model extraction through the medium of concept-based explanations: (D)ependency (G)raphs for (I)nterpreting (N)eural (N)etworks (**DGINN**) in Chapter 5 and (C)oncept-based (M)odel (E)xtraction framework (**CME**) in Chapter 6. While DGINN takes an intermediate step in extracting class-specific representation using a series of function decompositions, CME extracts concept-based representations using a compositions of two functions. Our techniques move the field of interpretability on step higher on the levels of explainability sophistication beyond *importance* into the realm of *functional relationship* description.

Our DGINN and CME frameworks confirm two conjectures. First, class-specific representations appear within a fraction of the latent space. These class-specific representations seem to capture information about their corresponding class since they can act as binary classifiers for that class, and surprisingly, a subspace of these class-specific representations corresponds to tiny latent space manifolds that are input invariant. These findings give tangible evidence to the sparsity, manifolds, natural clustering, and shared factors assumptions. Second, we confirm the conjecture that PDRs describe fine-grained variation in the data, which can be associated with human-understandable concepts. The results shed new light on how information is represented in the DNN hidden space. Moreover, it might be the case that there are at least two distinct types of hidden space manifolds: blobs and paths. Blob manifolds encode the variation concerning a distinct concept value, while path manifolds represent the variation of the entire concept along the manifold structure. These results caution against interpretations of single neurons in isolation using feature importance methods and make a case for well-controlled datasets that allow for rigorous quantitative evaluation. We give recommendation for the future of XAI evaluation in Section 7.2.3.

Implications Our work raises concerns for those hoping to rely on feature importance explanation methods to measure or enforce standards of fairness. For example, a trained loan scoring system might be unfair with respect to a sensitive feature such as gender. The model's parameters might be modified in such a way that a feature importance explanation could falsely suggest that the output does not depend on this sensitive feature.

Additionally, the findings presented here will be of interest to researchers aiming to quantify and axiomatise concept-based explanation approaches because we cast the field into a well-defined mathematical formulation and propose a way to compare alternative techniques. Finally, our study raises questions regarding the psychology of human concept learning. For example, do people think of concepts as continuous spectra of variation (e.g., small-medium-large), or as binary categories (e.g., large vs not-large)? How do people make decisions based on concepts, if at all?

7.2 Future Work

Here we describe four different strands of research that naturally follow from our work: (1) investigating the conditions that lead to the success of the adversarial explanation attack; (2) alleviating the limitations of concept-based explanations; (3) developing more rigorous forms for evaluating explanation methods; and (4) future research of explainability methods.

7.2.1 Adversarial Explanation Attack

There are many interesting questions to explore in future work. How is the attack succeeding, how can it be refined (e.g., by better understanding the learning dynamics, or by exploring how well it might be used against multiple target variables), and how might it be well defended against? We discuss them next.

Representational Capacity and Dataset Complexity One could further explore how the attack relates to the dataset complexity, model capacity, and explanation method (e.g., the ratio between the model's capacity and the dataset's complexity, or the degree of confounding information). For example, one could investigate formal metrics of dataset complexity (c.f. Semenova and Rudin, 2019) and investigate the correlation between the smoothness of convergence and dataset complexity. Another exciting area of further exploration would be to understand the relation between local/global curvature and robust/adversarial explanation training. Appendix C provides more details as to how these concepts could be investigated further.

Virtual Adversarial Training A fruitful area for further work is to improve the model similarity in terms of output similarity rather than performance similarity. We conjecture that the attack will be more successful when the modified model is trained to match the prediction's of the original model rather than to fit the training data. A convenient way to achieve this goal is to minimise the KL-divergence between the output distributions of the modified and original models, instead of minimising the divergence between the modified model output distribution and the empirical training distribution.

7.2.2 Concept-based Explanations

Two limitations of concept-based explanations are that (1) we assume the availability of a fixed set of k concept labels before model extraction begins, and (2) we assume we know the concept space. In Section 6.3.3.1, we demonstrate that CNNs encode concept values in well-separable regions in the hidden space of their dense layers. This result suggests that unsupervised or active learning approaches could be a fruitful area for further work in alleviating these challenges.

Concept Labelling A natural progression of this work is to explore active-learning based approaches to obtain maximally-informative concept labels from the user in an interactive fashion. These approaches may be used to reduce manual concept labelling effort significantly and improve extracted model fidelity. An active-learning approach for concept labelling is one step towards the vision of interactive machine learning. However, a further study would need to assess the minimum number of concept labels required for the task and dataset at hand.

Concept Extraction Another way to decrease the cost of acquiring concept annotations is to improve the concept extraction process. We compute $\hat{p}$ by extracting and combining concept information from individual layers, using semi-supervised methods. Exploring other approaches to extracting $\hat{p}$, such as considering combinations of multiple layers as an ensemble of concept predictors, or weakly-supervised methods, are exciting avenues for further exploration. Furthermore, it might be the case that concepts have a non-linear mapping with outputs. Hence, investigating techniques to capture such relationships in an interpretable fashion would be a fruitful area of exploration.

Automatic Concept Extraction Concept extraction seems to work reliably when the concept space is well-understood, and the different concepts are known in advance. However, it is not realistic to expect that this knowledge would be available for a wide variety of tasks. In the cases when we have no information about the concepts, $\hat{p}$ has to be extracted in an *unsupervised fashion*. Automatic Concept Extraction Ghorbani et al., 2019 are limited to super-pixels. Another possibility could be to identify shared PDRs across class-specific dependency graphs using network motifs Milo et al., 2002 or other pattern matching techniques. We conjecture that there is a high likelihood a shared PDR corresponds to a concept or particular concept value.

Encoding Sensitivity and Invariance Knowledge of concepts can be used to improve model performance or encode domain information. Tangent propagation Simard et al., 1992 is a regularisation technique that forces the model to become invariant to variations outside of the class manifold[1]. The main limitation of tangent propagation is that it requires the user to define vectors that are tangent to the class manifold manually. Similarly to Rifai et al., 2011b, we conjecture that DGINN and CME can be used to elucidate information about the class and concept manifolds automatically. These pieces of information can contribute to the regularisation of more accurate and robust models. For example, DNN developers could control the salient factors, which a DNN needs to develop sensitivity to, while managing the invariance to noise and spurious correlations.

Verification and Robustness We mentioned that the end task or auxiliary tasks across the layers could be an effective way to control the salience of the network to particular factors. A greater focus on the end task design could give further insights into the information that the network is learning or discarding that are relevant for *model robustness*, and *model verification*.

For example, DGINN and CME could be used in conjunction with novel loss functions or auxiliary task to inspect the learned concepts and ensure the model relies on desired concepts for its decision-making. Another example application of DGINN and CME might be the detection of adversarial explanation attacks (discussed in Chapter 4) and adversarial examples, or unintended model behavioural, such as model bias. In this case, we can monitor a DNN for “unexpected” concept associated with a sample or a decision. That is, we can use $\hat{q}$ to directly compare a user’s mental model of a task (i.e., how concepts should be used during decision making) with the model learned by a DNN.

[1] See Appendix D.

7.2.3 XAI Evaluation

One of the most significant challenges for the field of XAI is the lack of well-established and rigorous methods to evaluate and compare explainability methods. Doshi-Velez and Kim Doshi-Velez and Kim, 2017 proposes to include user participation as part of the evaluation protocol. We argue future studies need to establish controlled datasets and controlled models for two reasons. On the one hand, when the provided explanations do not make sense, it is not always clear whether the explanation is wrong, or whether the model has learned the wrong signal. On the other hand, when explanations are meaningful, it cannot definitively be determined whether they faithfully represent model behaviour.

Controlled Datasets The common concept of *"garbage in, garbage out"* means that spurious correlations or noise are likely to be learned; however, a human observer would question even the most trustworthy explanation method when the explanation presents contradicting or noisy data. The challenge now is to develop and design datasets using complex, but controlled generative processes. We suggest starting by focusing on exploring the effects of feature interactions. In this way, we can isolate the effects of confounding factors and accurately measure the capabilities of our explanation techniques, given particular variations and dependencies in the data[2]. We can also begin to move up the ladder of the levels of explainability understanding. Controlled datasets help us disambiguate whether the problem is with the explanation technique or the human interpreter.

Controlled Models In contrast to controlled datasets, well-controlled models help us discern whether the problem is with the explanation or with the model. Model parameters could have a considerable influence on the extracted explanations, and it is very misleading to develop explanation techniques on poor-performing models. Worryingly, due to confirmation bias[3] humans would accept explanations, which intuitively make sense, but do not reflect the model's behaviour faithfully Adebayo et al., 2018; Miller, 2019. In Chapter 4, we demonstrated one example of masking the relative importance of a set of sensitive features, although the model behaviour indicated that information regarding these features was still used.

Fixed controlled models can be designed such that the internal representations and the interactions between them are well-understood. For example, the first representation learning algorithms were manually designed to encode family free relationships Hinton and Anderson, 1981. Equipped with a gold standard to compare with, researchers will be able to distinguish between an inadequately trained model and a faulty explanation

[2] For more information see Appendix C.2.1.

[3] See Section 3.5.2.3, "Cognitive fragility".

method more clearly.

Future of Explainability Finally, the main limitation of the majority of explainability methods is the ability to describe only very local model behaviour Jiang et al., 2018. This is one of the reasons why we can alter the decision boundary to affect the interpretability and the apparent fairness of a model, with little change in accuracy. However, both the local and global curvature of the decision boundary play an important part in defining the model performance and interpretability. As we mentioned, the model performance directly affects interpretability. Hence, accuracy is not a variable to trade-off with trustworthiness. On the contrary, it contributes to the increased trust in the model. Therefore, future interpretability research should focus not on finding a compromise between accurate and interpretable models, but on describing both the local and global curvature of models.

Bibliography

Aamodt, Agnar (1991). "A knowledge-intensive, integrated approach to problem solving and sustained learning". In: *Knowledge Engineering and Image Processing Group. University of Trondheim*, pp. 27–85.

Aamodt, Agnar and Enric Plaza (1994). "Case-based reasoning: Foundational issues, methodological variations, and system approaches". In: *AI communications* 7.1, pp. 39–59.

Abadi, Martín et al. (2016). "Tensorflow: A system for large-scale machine learning". In: *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283.

Abdollahi, Behnoush and Olfa Nasraoui (2016). "Explainable Restricted Boltzmann Machines for Collaborative Filtering". In: *arXiv preprint arXiv:1606.07129*.

— (2018). "Transparency in fair machine learning: the case of explainable recommender systems". In: *Human and Machine Learning*. Springer, pp. 21–35.

Adadi, Amina and Mohammed Berrada (2018). "Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI)". In: *IEEE Access* 6, pp. 52138–52160.

Adebayo, Julius et al. (2018). "Sanity checks for saliency maps". In: *Advances in Neural Information Processing Systems*, pp. 9505–9515.

Adel, Tameem, Zoubin Ghahramani, and Adrian Weller (2018). "Discovering interpretable representations for both deep generative and discriminative models". In: *International Conference on Machine Learning*, pp. 50–59.

Ahuja, Kartik et al. (2020). "Invariant Risk Minimization Games". In: *CoRR* abs/2002.04692. arXiv: `2002.04692`. URL: `https://arxiv.org/abs/2002.04692`.

Al-Ani, Ahmed and Mohamed Deriche (2002). "Feature selection using a mutual information based measure". In: *Object recognition supported by user interaction for service robots*. Vol. 4. IEEE, pp. 82–85.

Alain, Guillaume and Yoshua Bengio (2014). "What regularized auto-encoders learn from the data-generating distribution". In: *The Journal of Machine Learning Research* 15.1, pp. 3563–3593.

Alain, Guillaume and Yoshua Bengio (2017). “Understanding intermediate layers using linear classifier probes”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings.* OpenReview.net. URL: https://openreview.net/forum?id=HJ4-rAVtl.

Alayrac, Jean-Baptiste et al. (2019). “Are Labels Required for Improving Adversarial Robustness?” In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada.* Ed. by Hanna M. Wallach et al., pp. 12192–12202. URL: http://papers.nips.cc/paper/9388-are-labels-required-for-improving-adversarial-robustness.

Alvarez-Melis, David and Tommi S Jaakkola (2018). “Towards robust interpretability with self-explaining neural networks”. In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems.* Curran Associates Inc., pp. 7786–7795.

Amodei, Dario et al. (2016). “Concrete Problems in AI Safety”. In: *CoRR* abs/1606.06565. arXiv: 1606.06565. URL: http://arxiv.org/abs/1606.06565.

Ancona, Marco et al. (2018). “Towards better understanding of gradient-based attribution methods for Deep Neural Networks”. In: *6th International Conference on Learning Representations (ICLR 2018).*

Ancona, Marco et al. (2019). “Gradient-based attribution methods”. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning.* Springer, pp. 169–191.

Andreas, Jacob (2019). “Measuring Compositionality in Representation Learning”. In: *International Conference on Learning Representations.* URL: https://openreview.net/forum?id=HJz05o0qK7.

Andrews, Robert, Joachim Diederich, and Alan B Tickle (1995). “Survey and critique of techniques for extracting rules from trained artificial neural networks”. In: *Knowledge-based systems* 8.6, pp. 373–389.

Anirudh, Rushil et al. (2017). *Influential sample selection: A graph signal processing approach.* Tech. rep. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States).

Ankerst, Mihael et al. (1999). “Visual classification: an interactive approach to decision tree construction”. In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 392–396.

Apley, Daniel W and Jingyu Zhu (2016). “Visualizing the effects of predictor variables in black box supervised learning models”. In: *arXiv preprint arXiv:1612.08468.*

Arjovsky, Martín et al. (2019). “Invariant Risk Minimization”. In: *CoRR* abs/1907.02893. arXiv: 1907.02893. URL: http://arxiv.org/abs/1907.02893.

Arpit, Devansh, Caiming Xiong, and Richard Socher (Sept. 2019). “Predicting with High Correlation Features”. In: *arXiv e-prints*, arXiv:1910.00164, arXiv:1910.00164. arXiv: `1910.00164 [stat.ML]`.

Attwell, David and Simon B Laughlin (2001). “An energy budget for signaling in the grey matter of the brain”. In: *Journal of Cerebral Blood Flow & Metabolism* 21.10, pp. 1133–1145.

Bach, Sebastian et al. (2015). “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation”. In: *PloS one* 10.7, e0130140.

Baehrens, David et al. (2010). “How to explain individual classification decisions”. In: *Journal of Machine Learning Research* 11.Jun, pp. 1803–1831.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: `http://arxiv.org/abs/1409.0473`.

Barron, Andrew R (1993). “Universal approximation bounds for superpositions of a sigmoidal function”. In: *IEEE Transactions on Information theory* 39.3, pp. 930–945.

Bartlett, Peter L and Wolfgang Maass (2003). “Vapnik-Chervonenkis dimension of neural nets”. In: *The handbook of brain theory and neural networks*, pp. 1188–1192.

Bartlett, Peter L and Shahar Mendelson (2002). “Rademacher and Gaussian complexities: Risk bounds and structural results”. In: *Journal of Machine Learning Research* 3.Nov, pp. 463–482.

Bartlett, Peter and John Shawe-Taylor (1999). “Generalization performance of support vector machines and other pattern classifiers”. In: *Advances in Kernel methods—support vector learning*, pp. 43–54.

Bau, Anthony et al. (2019). “Identifying and Controlling Important Neurons in Neural Machine Translation”. In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=H1z-PsR5KX`.

Bau, David et al. (2017a). “Network Dissection: Quantifying Interpretability of Deep Visual Representations”. In: *Computer Vision and Pattern Recognition*.

— (2017b). “Network dissection: Quantifying interpretability of deep visual representations”. In: *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*. IEEE, pp. 3319–3327.

Baum, Leonard E and Ted Petrie (1966). “Statistical inference for probabilistic functions of finite state Markov chains”. In: *The annals of mathematical statistics* 37.6, pp. 1554–1563.

Becker, Barry, Ron Kohavi, and Dan Sommerfield (2001). “Visualizing the simple Bayesian classifier”. In: *Information visualization in data mining and knowledge discovery* 18, pp. 237–249.

Becker, Suzanna and Geoffrey E Hinton (1992). “Self-organizing neural network that discovers surfaces in random-dot stereograms”. In: *Nature* 355.6356, pp. 161–163.

Belkin, Mikhail and Partha Niyogi (2003). “Laplacian eigenmaps for dimensionality reduction and data representation”. In: *Neural computation* 15.6, pp. 1373–1396.

Bellamy, Rachel K. E. et al. (Oct. 2018). *AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias*. URL: `https://arxiv.org/abs/1810.01943`.

Bengio, Yoshua (2009). “Learning deep architectures for AI”. In: *Foundations and trends in Machine Learning* 2.1, pp. 1–127.

Bengio, Yoshua, Aaron Courville, and Pascal Vincent (2013). “Representation learning: A review and new perspectives”. In: *IEEE transactions on pattern analysis and machine intelligence* 35.8, pp. 1798–1828.

Bengio, Yoshua and Olivier Delalleau (2011). “On the expressive power of deep architectures”. In: *International Conference on Algorithmic Learning Theory*. Springer, pp. 18–36.

Bengio, Yoshua, Olivier Delalleau, and Nicolas L Roux (2006). “The curse of highly variable functions for local kernel machines”. In: *Advances in neural information processing systems*, pp. 107–114.

Bengio, Yoshua, Olivier Delalleau, and Clarence Simard (2010). “Decision trees do not generalize to new variations”. In: *Computational Intelligence* 26.4, pp. 449–467.

Bengio, Yoshua, Réjean Ducharme, and Pascal Vincent (2000). “A Neural Probabilistic Language Model”. In: *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*. Ed. by Todd K. Leen, Thomas G. Dietterich, and Volker Tresp. MIT Press, pp. 932–938. URL: `http://papers.nips.cc/paper/1839-a-neural-probabilistic-language-model`.

Bengio, Yoshua, Yann LeCun, et al. (2007). “Scaling learning algorithms towards AI”. In: *Large-scale kernel machines* 34.5, pp. 1–41.

Bengio, Yoshua and Martin Monperrus (2005). “Non-local manifold tangent learning”. In: *Advances in Neural Information Processing Systems*, pp. 129–136.

Bengio, Yoshua et al. (2007). “Greedy layer-wise training of deep networks”. In: *Advances in neural information processing systems*, pp. 153–160.

Bengio, Yoshua et al. (2013). “Better mixing via deep representations”. In: *International conference on machine learning*, pp. 552–560.

Beutel, Alex et al. (2017). “Data Decisions and Theoretical Implications when Adversarially Learning Fair Representations”. In: *CoRR* abs/1707.00075. arXiv: 1707.00075. URL: http://arxiv.org/abs/1707.00075.

Bhatt, Umang et al. (2020). “Explainable machine learning in deployment”. In: *FAT* ’20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*. Ed. by Mireille Hildebrandt et al. ACM, pp. 648–657. DOI: 10.1145/3351095.3375624. URL: https://doi.org/10.1145/3351095.3375624.

Biggio, Battista et al. (2013). “Evasion attacks against machine learning at test time”. In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer, pp. 387–402.

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). “Latent dirichlet allocation”. In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Bostrom, Nick and Eliezer Yudkowsky (2014). “The ethics of artificial intelligence”. In: *The Cambridge Handbook of Artificial Intelligence*, pp. 316–334.

Brand, Matthew (2003). “Charting a manifold”. In: *Advances in neural information processing systems*, pp. 985–992.

Braverman, Mark (2011). “Poly-logarithmic independence fools bounded-depth boolean circuits”. In: *Communications of the ACM* 54.4, pp. 108–115.

Breiman, Leo (2001). “Random Forests”. In: *Machine Learning* 45.1, pp. 5–32. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

Cantareira, Gabriel Dias, Fernando V. Paulovich, and Elham Etemad (2020). “Visualizing Learning Space in Neural Network Hidden Layers”. In: *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, VISIGRAPP 2020, Volume 3: IVAPP, Valletta, Malta, February 27-29, 2020*. Ed. by Andreas Kerren, Christophe Hurter, and José Braz. SCITEPRESS, pp. 110–121. DOI: 10.5220/0009168901100121. URL: https://doi.org/10.5220/0009168901100121.

Cao, Chunshui et al. (2015). “Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks”. In: *Proceedings of the IEEE international conference on computer vision*, pp. 2956–2964.

Caruana, Richard A (1993). “Multitask connectionist learning”. In: *In Proceedings of the 1993 Connectionist Models Summer School*. Citeseer.

Carvalho, Diogo V, Eduardo M Pereira, and Jaime S Cardoso (2019). “Machine learning interpretability: A survey on methods and metrics”. In: *Electronics* 8.8, p. 832.

Cayton, Lawrence (2005). “Algorithms for manifold learning”. In: *Univ. of California at San Diego Tech. Rep* 12.1-17, p. 1.

Chen, Chaofan et al. (2019a). “This Looks Like That: Deep Learning for Interpretable Image Recognition”. In: *Advances in Neural Information Processing Systems 32: An-*

nual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada. Ed. by Hanna M. Wallach et al., pp. 8928–8939. URL: `https://proceedings.neurips.cc/paper/2019/hash/adf7ee2dcf142b0e11888e72b43fcb75-Abstract.html`.

Chen, Daizhuo et al. (2017). "Enhancing transparency and control when drawing data-driven inferences about individuals". In: *Big data* 5.3, pp. 197–212.

Chen, Jiahao et al. (2019b). "Fairness under unawareness: Assessing disparity when protected class is unobserved". In: *Proceedings of the Conference on Fairness, Accountability, and Transparency.* ACM, pp. 339–348.

Chen, Jianbo et al. (2019c). "L-Shapley and C-Shapley: Efficient Model Interpretation for Structured Data". In: *International Conference on Learning Representations.* URL: `https://openreview.net/forum?id=S1E3Ko09F7`.

Chen, Xi et al. (2016). "Infogan: Interpretable representation learning by information maximizing generative adversarial nets". In: *Advances in Neural Information Processing Systems*, pp. 2172–2180.

Chen, Zhi, Yijie Bei, and Cynthia Rudin (2020). *Concept Whitening for Interpretable Image Recognition.* arXiv: `2002.01650 [cs.LG]`.

Chollet, François et al. (2015). *Keras.* `https://github.com/fchollet/keras`.

Clancey, William J (1983). "The epistemology of a rule-based expert system: a framework for explanation". In: *Artificial intelligence* 20.3, pp. 215–251.

Cohen, Marvin S, Jared T Freeman, and Steve Wolf (1996). "Metarecognition in time-stressed decision making: Recognizing, critiquing, and correcting". In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 38.2, pp. 206–219.

Collobert, Ronan et al. (2011). "Natural language processing (almost) from scratch". In: *Journal of machine learning research* 12.Aug, pp. 2493–2537.

Cui, Yin et al. (2018). "Large scale fine-grained categorization and domain-specific transfer learning". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4109–4118.

Cunningham, Pádraig, Dónal Doyle, and John Loughrey (2003). "An evaluation of the usefulness of case-based explanation". In: *Case-Based Reasoning Research and Development*, pp. 1065–1065.

Dabkowski, Piotr and Yarin Gal (2017). "Real time image saliency for black box classifiers". In: *Advances in Neural Information Processing Systems*, pp. 6967–6976.

Delalleau, Olivier and Yoshua Bengio (2011). "Shallow vs. deep sum-product networks". In: *Advances in neural information processing systems*, pp. 666–674.

Deng, Li (2012). "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]". In: *IEEE Signal Process. Mag.* 29.6, pp. 141–142. DOI: `10.1109/MSP.2012.2211477`. URL: `https://doi.org/10.1109/MSP.2012.2211477`.

Dhamdhere, Kedar, Mukund Sundararajan, and Qiqi Yan (2018). “How Important is a Neuron”. In: *International Conference on Learning Representations.*

Dhurandhar, Amit et al. (2018). “Explanations based on the missing: Towards contrastive explanations with pertinent negatives”. In: *Advances in Neural Information Processing Systems*, pp. 592–603.

Dhurandhar, Amit et al. (2020). *Model agnostic contrastive explanations for structured data.* US Patent App. 16/217,574.

Diakopoulos, Nicholas et al. (2017). “Principles for accountable algorithms and a social impact statement for algorithms”. In: *FAT/ML.*

Dimanov, Botty and Mateja Jamnik (2019). “Step-Wise Sensitivity Analysis: Identifying Partially Distributed Representations for Interpretable Deep Learning”. In: *ICLR 2019 Workshop Debugging Machine Learning Models.*

Dimanov, Botty et al. (2020). “You shouldn’t trust me: Learning models which conceal unfairness from multiple explanation methods”. In: *European Conference on Artificial Intelligence.*

Dmitry, Kazhdan et al. (2020). “MEME: A Concept-based Model Extraction Approach to RNN Explainability”. In: *NeurIPS 2020 Workshop on Human And Model in the Loop Evaluation and Training Strategies.*

Do, Kien and Truyen Tran (2020). “Theory and evaluation metrics for learning disentangled representations”. In: *8th International Conference on Learning Representations (ICLR 2020.*

Dombrowski, Ann-Kathrin et al. (2019). *Explanations can be manipulated and geometry is to blame.* arXiv: `1906.07983 [stat.ML]`.

Donahue, Jeff et al. (2014). “DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition.” In: *International conference on machine learning (ICML).* Vol. 32, pp. 647–655.

Donoho, David L and Carrie Grimes (2003). “Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data”. In: *Proceedings of the National Academy of Sciences* 100.10, pp. 5591–5596.

Doshi-Velez, F. and B. Kim (Feb. 2017). “Towards A Rigorous Science of Interpretable Machine Learning”. In: *ArXiv e-prints.* arXiv: `1702.08608 [stat.ML]`.

Doshi-Velez, Finale and Been Kim (2017). “Towards a rigorous science of interpretable machine learning”. In: *arXiv preprint arXiv:1702.08608.*

Drucker, Harris and Yann Le Cun (1992). “Improving generalization performance using double backpropagation”. In: *IEEE Transactions on Neural Networks* 3.6, pp. 991–997.

Dua, Dheeru and Casey Graff (2017). *UCI Machine Learning Repository.* URL: `http://archive.ics.uci.edu/ml`.

Duch, Włodzisław (2003). “Coloring black boxes: visualization of neural network decisions”. In: *Proceedings of the International Joint Conference on Neural Networks, 2003.* Vol. 3. IEEE, pp. 1735–1740.

Ebrahimi, Javid et al. (2018). “HotFlip: White-Box Adversarial Examples for Text Classification”. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers.* Ed. by Iryna Gurevych and Yusuke Miyao. Association for Computational Linguistics, pp. 31–36. DOI: `10.18653/v1/P18-2006`. URL: `https://www.aclweb.org/anthology/P18-2006/`.

Elman, Jeffrey L (1990). “Finding structure in time”. In: *Cognitive science* 14.2, pp. 179–211.

— (1991). “Distributed representations, simple recurrent networks, and grammatical structure”. In: *Machine learning* 7.2-3, pp. 195–225.

Erhan, Dumitru et al. (2009). “Visualizing higher-layer features of a deep network”. In: *University of Montreal* 1341, p. 3.

Erhan, Dumitru et al. (2010). “Why does unsupervised pre-training help deep learning?” In: *Journal of Machine Learning Research* 11.Feb, pp. 625–660.

Erion, Gabriel G. et al. (2019). “Learning Explainable Models Using Attribution Priors”. In: *CoRR* abs/1906.10670. arXiv: `1906.10670`. URL: `http://arxiv.org/abs/1906.10670`.

Etmann, Christian (2019). “A Closer Look at Double Backpropagation”. In: *arXiv preprint arXiv:1906.06637*.

Everitt, Brian S (1985). *Mixture Distributions.* I. Wiley Online Library.

Feldman, Jerome A and Dana H Ballard (1982). “Connectionist models and their properties”. In: *Cognitive science* 6.3, pp. 205–254.

Fisher, Aaron, Cynthia Rudin, and Francesca Dominici (2019). “All Models are Wrong, but Many are Useful: Learning a Variable’s Importance by Studying an Entire Class of Prediction Models Simultaneously”. In: *Journal of Machine Learning Research* 20.177, pp. 1–81. URL: `http://jmlr.org/papers/v20/18-760.html`.

Fix, Evelyn and Joseph L Hodges Jr (1951). *Discriminatory analysis-nonparametric discrimination: consistency properties.* Tech. rep. DTIC Document.

Földiák, Peter and Peter Fdilr (1989). “Adaptive network for optimal linear feature extraction”. In:

Fong, Ruth and Andrea Vedaldi (2018). “Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8730–8738.

— (2019). “Explanations for attributing deep neural network predictions”. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning.* Springer, pp. 149–167.

Forster, Malcolm R (1986). “Unification and scientific realism revisited”. In: *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*. Vol. 1986. 1. Philosophy of Science Association, pp. 394–405.

Forster, Malcolm and Elliott Sober (1994). “How to tell when simpler, more unified, or less ad hoc theories will provide more accurate predictions”. In: *The British Journal for the Philosophy of Science* 45.1, pp. 1–35.

Frankle, Jonathan and Michael Carbin (2018). “The Lottery Ticket Hypothesis: Training Pruned Neural Networks”. In: *CoRR* abs/1803.03635. arXiv: `1803.03635`. URL: `http://arxiv.org/abs/1803.03635`.

Freitas, Alex A (2014). “Comprehensible classification models: a position paper”. In: *ACM SIGKDD explorations newsletter* 15.1, pp. 1–10.

Freiwald, Winrich A, Doris Y Tsao, and Margaret S Livingstone (2009). “A face feature space in the macaque temporal lobe”. In: *Nature neuroscience* 12.9, p. 1187.

Friedman, Jerome H (2001). “Greedy function approximation: a gradient boosting machine”. In: *Annals of statistics*, pp. 1189–1232.

Fung, Glenn, Sathyakama Sandilya, and R Bharat Rao (2005). “Rule extraction from linear support vector machines”. In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 32–40.

GB, Royal Society (2017). *Machine Learning: The Power and Promise of Computers that Learn by Example: an Introduction*. Royal Society.

Geirhos, Robert et al. (2019). “ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness.” In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=Bygh9j09KX`.

Geirhos, Robert et al. (2020). “Shortcut Learning in Deep Neural Networks”. In: *CoRR* abs/2004.07780. arXiv: `2004.07780`. URL: `https://arxiv.org/abs/2004.07780`.

Ghorbani, Amirata, Abubakar Abid, and James Zou (2019). “Interpretation of neural networks is fragile”. In: *AAAI*.

Ghorbani, Amirata et al. (2019). “Towards automatic concept-based explanations”. In: *Advances in Neural Information Processing Systems*.

Gilpin, Leilani H et al. (2018). “Explaining explanations: An overview of interpretability of machine learning”. In: *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*. IEEE, pp. 80–89.

Girshick, Ross et al. (2014). “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). “Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach”. In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington,* *USA, June 28 - July 2, 2011*. Ed. by Lise Getoor and Tobias Scheffer. Omnipress, pp. 513–520. URL: https://icml.cc/2011/papers/342_icmlpaper.pdf.

Goodfellow, Ian J., Aaron C. Courville, and Yoshua Bengio (2012). "Spike-and-Slab Sparse Coding for Unsupervised Feature Discovery". In: *NIPS Wokrshop Challenges in Learning Hierarchical Models*. NIPS. eprint: 1201.3382. URL: http://arxiv.org/abs/1201.3382.

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy (2015). "Explaining and Harnessing Adversarial Examples". In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1412.6572.

Goodfellow, Ian J. et al. (2014a). "Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1312.6082.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016a). *Deep Learning*. MIT Press.

— (2016b). *Deep learning*. MIT press.

Goodfellow, Ian et al. (2009). "Measuring Invariances in Deep Networks". In: *Advances in Neural Information Processing Systems 22*. Ed. by Y. Bengio et al. Curran Associates, Inc., pp. 646–654. URL: http://papers.nips.cc/paper/3790-measuring-invariances-in-deep-networks.pdf.

Goodfellow, Ian et al. (2014b). "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., pp. 2672–2680. URL: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf.

Goodman, Bryce and Seth Flaxman (2016). "European Union regulations on algorithmic decision-making and a" right to explanation"". In: *arXiv preprint arXiv:1606.08813*.

Goyal, Yash, Uri Shalit, and Been Kim (2019). "Explaining Classifiers with Causal Concept Effect (CaCE)". In: *arXiv preprint arXiv:1907.07165*.

Goyal, Yash et al. (2016). "Towards Transparent AI Systems: Interpreting Visual Question Answering Models". In: *arXiv preprint arXiv:1608.08974*.

Graves, Alex (2016). "Adaptive Computation Time for Recurrent Neural Networks". In: *CoRR* abs/1603.08983. URL: http://arxiv.org/abs/1603.08983.

Graves, Alex and Navdeep Jaitly (2014). "Towards End-To-End Speech Recognition with Recurrent Neural Networks." In: *ICML*. Vol. 14, pp. 1764–1772.

Graves, Alex et al. (2016). "Hybrid computing using a neural network with dynamic external memory". In: *Nature* 538.7626, pp. 471–476.

Gretton, Arthur et al. (2009). "Covariate shift by kernel mean matching". In: *Dataset shift in machine learning* 3.4, p. 5.

Grgić-Hlača, Nina et al. (2018). "Beyond distributive fairness in algorithmic decision making: Feature selection for procedurally fair learning". In: *Thirty-Second AAAI Conference on Artificial Intelligence.*

Grice, Herbert P (1975). "Logic and conversation". In: *Speech acts.* Brill, pp. 41–58.

Grün, Felix et al. (2016). "A Taxonomy and Library for Visualizing Learned Features in Convolutional Neural Networks". In: *arXiv preprint arXiv:1606.07757.*

Guidotti, Riccardo et al. (2018). "A survey of methods for explaining black box models". In: *ACM computing surveys (CSUR)* 51.5, pp. 1–42.

Gunning, D (2018). *Explainable artificial intelligence (XAI). Defense Advanced Research Projects agency.*

Guyon, Isabelle and André Elisseeff (2003). "An introduction to variable and feature selection". In: *Journal of machine learning research* 3.Mar, pp. 1157–1182.

Hall, Patrick (2019). *An introduction to machine learning interpretability.* O'Reilly Media, Incorporated.

Hall, Patrick, Navdeep Gill, and Nicholas Schmidt (2019). "Proposed Guidelines for the Responsible Use of Explainable Machine Learning". In: *arXiv preprint arXiv:1906.03533.*

Hardt, Moritz, Eric Price, and Nati Srebro (2016). "Equality of Opportunity in Supervised Learning". In: *Advances in Neural Information Processing Systems (NeurIPS).*

Harman, Gilbert H (1965). "The inference to the best explanation". In: *The philosophical review* 74.1, pp. 88–95.

Håstad, Johan and Mikael Goldmann (1991). "On the power of small-depth threshold circuits". In: *Computational Complexity* 1.2, pp. 113–129.

Hastad, John (1986). "Almost optimal lower bounds for small depth circuits". In: *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pp. 6–20.

Heckert, Nathanael A et al. (2002). *Handbook 151: NIST/SEMATECH e-Handbook of Statistical Methods.* DOI: `https://doi.org/10.18434/M32189`. URL: `http://www.itl.nist.gov/div898/handbook/`.

Heinze-Deml, Christina, Jonas Peters, and Nicolai Meinshausen (2018). "Invariant causal prediction for nonlinear models". In: *Journal of Causal Inference* 6.2.

Hendricks, Lisa Anne et al. (2018). "Generating Counterfactual Explanations with Natural Language". In: *ICML Workshop on Human Interpretability in Machine Learning*, pp. 95–98.

Hendrycks, Dan and Thomas Dietterich (2019). "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *International Conference on Learning Representations.* URL: `https://openreview.net/forum?id=HJz6tiCqYm`.

Heo, Juyeon, Sunghwan Joo, and Taesup Moon (2019). “Fooling neural network interpretations via adversarial model manipulation”. In: *Advances in Neural Information Processing Systems*, pp. 2921–2932.

Herman, Bernease. “The promise and peril of human evaluation for model interpretability”. In: *Proceedings of NIPS 2017 Symposium on Interpretable Machine.*

Hermans, Michiel and Benjamin Schrauwen (2013). “Training and analysing deep recurrent neural networks”. In: *Advances in neural information processing systems*, pp. 190–198.

Higgins, Irina et al. (2017). “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.* OpenReview.net. URL: `https://openreview.net/forum?id=Sy2fzU9gl`.

Hinton, GE, JL McClelland, and DE Rumelhart (1986). “Distributed representations. In: Parallel distributed processing: Explorations in the microstructure of cognition, vol. 2, Psychological and biological models”. In:

Hinton, Geoffrey E (1990). “Connectionist learning procedures”. In: *Machine learning.* Elsevier, pp. 555–610.

Hinton, Geoffrey E and James A Anderson (1981). “Implementing semantic networks in parallel hardware”. In: *Parallel models of associative memory.* Psychology Press, pp. 201–232.

Hinton, Geoffrey E, Simon Osindero, and Yee-Whye Teh (2006). “A fast learning algorithm for deep belief nets”. In: *Neural computation* 18.7, pp. 1527–1554.

Hinton, Geoffrey E and Ruslan R Salakhutdinov (2006). “Reducing the dimensionality of data with neural networks”. In: *science* 313.5786, pp. 504–507.

Hinton, Geoffrey E, Terrence J Sejnowski, and David H Ackley (1984). *Boltzmann machines: Constraint satisfaction networks that learn.* Carnegie-Mellon University, Department of Computer Science Pittsburgh.

Ho, Tin Kam and Mitra Basu (2002). “Complexity measures of supervised classification problems”. In: *IEEE transactions on pattern analysis and machine intelligence* 24.3, pp. 289–300.

Hotelling, Harold (1933). “Analysis of a complex of statistical variables into principal components.” In: *Journal of educational psychology* 24.6, p. 417.

Hurri, Jarmo and Aapo Hyvärinen (2003). “Temporal coherence, natural image sequences, and the visual cortex”. In: *Advances in Neural Information Processing Systems*, pp. 157–164.

Huysmans, Johan et al. (2011). “An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models”. In: *Decision Support Systems* 51.1, pp. 141–154.

Jacobsson, Henrik (2005). “Rule extraction from recurrent neural networks: Ataxonomy and review”. In: *Neural Computation* 17.6, pp. 1223–1263.

Jain, Sarthak and Byron C Wallace (2019). “Attention is not Explanation”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 3543–3556.

Jiang, Heinrich et al. (2018). “To Trust Or Not To Trust A Classifier”. In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.* Ed. by Samy Bengio et al., pp. 5546–5557. URL: `http://papers.nips.cc/paper/7798-to-trust-or-not-to-trust-a-classifier`.

Kahneman, Daniel and Amos Tversky (2013). “Prospect theory: An analysis of decision under risk”. In: *Handbook of the fundamentals of financial decision making: Part I*. World Scientific, pp. 99–127.

Karpathy, Andrej, Justin Johnson, and Fei-Fei Li (2015). “Visualizing and Understanding Recurrent Networks”. In: *CoRR* abs/1506.02078. URL: `http://arxiv.org/abs/1506.02078`.

Kazhdan, Dmitry, Zohreh Shams, and Pietro Liò (2020). “MARLeME: A Multi-Agent Reinforcement Learning Model Extraction Library”. In: *arXiv preprint arXiv:2004.07928*.

Kazhdan, Dmitry et al. (2020). “Now You See Me (CME): Concept-based Model Extraction”. In: *Proceedings of the CIKM 2020 Workshops co-located with 29th ACM International Conference on Information and Knowledge Management (CIKM 2020), Galway, Ireland, October 19-23, 2020.* Ed. by Stefan Conrad and Ilaria Tiddi. Vol. 2699. CEUR Workshop Proceedings. CEUR-WS.org. URL: `http://ceur-ws.org/Vol-2699/paper02.pdf`.

Kazhdan, Dmitry et al. (2021). “Is Disentanglement all you need? Comparing Concept-based & Disentanglement Approaches”. In:

Keil, Frank C (2006). “Explanation and understanding”. In: *Annu. Rev. Psychol.* 57, pp. 227–254.

Kim, Been, Rajiv Khanna, and Oluwasanmi O Koyejo (2016a). “Examples are not enough, learn to criticize! criticism for interpretability”. In: *Advances in neural information processing systems*, pp. 2280–2288.

Kim, Been, Rajiv Khanna, and Sanmi Koyejo (2016b). “Examples are not Enough, Learn to Criticize! Criticism for Interpretability”. In: *Advances in Neural Information Processing Systems.*

Kim, Been, Cynthia Rudin, and Julie A Shah (2014). “The Bayesian Case Model: A generative approach for case-based reasoning and prototype classification”. In: *Advances in Neural Information Processing Systems*, pp. 1952–1960.

Kim, Been, Julie A Shah, and Finale Doshi-Velez (2015). "Mind the gap: A generative approach to interpretable feature selection and extraction". In: *Advances in Neural Information Processing Systems*, pp. 2260–2268.

Kim, Been et al. (2017). "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)". In: *arXiv preprint arXiv:1711.11279*.

Kim, Been et al. (2018). "Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 2673–2682. URL: `http://proceedings.mlr.press/v80/kim18d.html`.

Kindermans, Pieter-Jan et al. (2017). "Learning how to explain neural networks: PatternNet and PatternAttribution". In: *arXiv preprint arXiv:1705.05598*.

Kindermans, Pieter-Jan et al. (2019). "The (un) reliability of saliency methods". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, pp. 267–280.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kingma, Diederik P and Max Welling (2014). "Auto-Encoding Variational Bayes". In: *stat* 1050, p. 1.

Kleinberg, Jon (2018). "Inherent trade-offs in algorithmic fairness". In: *ACM SIGMETRICS Performance Evaluation Review*. Vol. 46. 1. ACM, pp. 40–40.

Koh, Pang Wei and Percy Liang (2017). "Understanding Black-box Predictions via Influence Functions". In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 1885–1894. URL: `http://proceedings.mlr.press/v70/koh17a.html`.

Koh, Pang Wei et al. (2020). "Concept Bottleneck Models". In: *Proceedings of Machine Learning and Systems 2020*. International Conference on Machine Learning, pp. 11313–11323.

Kornblith, Simon et al. (2019). "Similarity of Neural Network Representations Revisited". In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 3519–3529. URL: `http://proceedings.mlr.press/v97/kornblith19a.html`.

Kouw, Wouter M. (2018). "An introduction to domain adaptation and transfer learning". In: *CoRR* abs/1812.11806. arXiv: `1812.11806`. URL: `http://arxiv.org/abs/1812.11806`.

Krishnan, R, G Sivakumar, and P Bhattacharya (1999). “Extracting decision trees from trained neural networks”. In: *Pattern recognition* 32.12.

Krizhevsky, Alex, Geoffrey Hinton, et al. (2009). “Learning multiple layers of features from tiny images”. In:

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012). “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems*, pp. 1097–1105.

Kulesza, Todd et al. (2011). “Why-oriented end-user debugging of naive Bayes text classification”. In: *ACM Transactions on Interactive Intelligent Systems (TiiS)* 1.1, pp. 1–31.

Lake, Brenden M et al. (2017). “Building machines that learn and think like people”. In: *Behavioral and brain sciences* 40.

Lakkaraju, Himabindu, Stephen H Bach, and Jure Leskovec (2016). “Interpretable decision sets: A joint framework for description and prediction”. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1675–1684.

Landecker, Will et al. (2013). “Interpreting individual classifications of hierarchical networks”. In: *Computational Intelligence and Data Mining (CIDM), 2013 IEEE Symposium on.* IEEE, pp. 32–38.

Lapuschkin, Sebastian (2019). “Opening the machine learning black box with layer-wise relevance propagation”. In:

Larson, Jeff et al. (2019). *How We Analyzed the COMPAS Recidivism Algorithm.* URL: `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm`.

Lasserre, Julia A, Christopher M Bishop, and Thomas P Minka (2006). “Principled hybrids of generative and discriminative models”. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06).* Vol. 1. IEEE, pp. 87–94.

LeCun, Yann et al. (1990). “Handwritten digit recognition with a back-propagation network”. In: *Advances in neural information processing systems*, pp. 396–404.

Lee, Wee Sun, Peter L Bartlett, and Robert C Williamson (1995). “Lower bounds on the VC dimension of smoothly parameterized function classes”. In: *Neural Computation* 7.5, pp. 1040–1053.

Lei, Jing et al. (2018). “Distribution-free predictive inference for regression”. In: *Journal of the American Statistical Association* 113.523, pp. 1094–1111.

Lei, Tao (2017). “Interpretable neural models for natural language processing”. PhD thesis. Massachusetts Institute of Technology.

Lennie, Peter (2003). “The cost of cortical computation”. In: *Current biology* 13.6, pp. 493–497.

Letham, Benjamin et al. (2015). "Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model". In: *The Annals of Applied Statistics* 9.3, pp. 1350–1371.

Li, Jiwei et al. (2015). "Visualizing and understanding neural models in nlp". In: *arXiv preprint arXiv:1506.01066*.

Li, Yixuan et al. (2016). "Convergent Learning: Do different neural networks learn the same representations?" In: *Proceedings of International Conference on Learning Representation (ICLR)*.

Lin, Henry W. and Max Tegmark (2016). "Why does deep and cheap learning work so well?" In: *CoRR* abs/1608.08225. arXiv: 1608.08225. URL: http://arxiv.org/abs/1608.08225.

Lipton, Peter (1990). "Contrastive explanation". In: *Royal Institute of Philosophy Supplement* 27, pp. 247–266.

Lipton, Zachary C (2016). "The mythos of model interpretability". In: *arXiv preprint arXiv:1606.03490*.

Lipton, Zachary C., Yu-Xiang Wang, and Alexander J. Smola (2018). "Detecting and Correcting for Label Shift with Black Box Predictors". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 3128–3136. URL: http://proceedings.mlr.press/v80/lipton18a.html.

Liu, Mengchen et al. (2017). "Towards better analysis of deep convolutional neural networks". In: *IEEE Transactions on Visualization and Computer Graphics* 23.1, pp. 91–100.

Liu, Qiuhua, Xuejun Liao, and Lawrence Carin (2008). "Semi-supervised multitask learning". In: *Advances in Neural Information Processing Systems*.

Locatello, Francesco et al. (2020). "A Sober Look at the Unsupervised Learning of Disentangled Representations and their Evaluation". In: *Journal of Machine Learning Research* 21.209, pp. 1–62. URL: http://jmlr.org/papers/v21/19-976.html.

Lopez-Paz, David (2016). "From dependence to causation". In: *arXiv preprint arXiv:1607.03300*.

Lorena, Ana C et al. (2019). "How Complex is your classification problem? A survey on measuring classification complexity". In: *ACM Computing Surveys (CSUR)* 52.5, pp. 1–34.

Lotter, William, Gabriel Kreiman, and David D. Cox (2015). "Unsupervised Learning of Visual Structure using Predictive Generative Networks". In: *CoRR* abs/1511.06380. arXiv: 1511.06380. URL: http://arxiv.org/abs/1511.06380.

Lou, Yin, Rich Caruana, and Johannes Gehrke (2012). “Intelligible models for classification and regression”. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, pp. 150–158.

Lou, Yin et al. (2013). “Accurate intelligible models with pairwise interactions”. In: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 623–631.

Lughofer, Edwin et al. (2017). “Explaining classifier decisions linguistically for stimulating and improving operators labeling behavior”. In: *Information Sciences* 420, pp. 16–36.

Lundberg, Scott M and Su-In Lee (2017). “A unified approach to interpreting model predictions”. In: *Advances in Neural Information Processing Systems*, pp. 4765–4774.

Maaten, Laurens van der and Geoffrey Hinton (2008). “Visualizing data using t-SNE”. In: *Journal of Machine Learning Research* 9.Nov, pp. 2579–2605.

MacKay, David JC and David JC Mac Kay (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Madry, Aleksander et al. (2018). “Towards Deep Learning Models Resistant to Adversarial Attacks”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. URL: `https://openreview.net/forum?id=rJzIBfZAb`.

Mahendran, Aravindh and Andrea Vedaldi (2015). “Understanding deep image representations by inverting them”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5188–5196.

Matthey, Loic et al. (2017). *dSprites: Disentanglement testing Sprites dataset*. URL: `https://github.com/deepmind/dsprites-dataset/`.

McAllester, David A. (1999). “PAC-Bayesian Model Averaging”. In: *Proceedings of the Twelfth Annual Conference on Computational Learning Theory, COLT 1999, Santa Cruz, CA, USA, July 7-9, 1999*. Ed. by Shai Ben-David and Philip M. Long. ACM, pp. 164–170. DOI: `10.1145/307400.307435`. URL: `https://doi.org/10.1145/307400.307435`.

McCullagh, Peter et al. (1986). “[Generalized Additive Models]: Comment”. In: *Statistical Science* 1.3, pp. 314–314.

Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig (June 2013b). “Linguistic Regularities in Continuous Space Word Representations”. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, pp. 746–751. URL: `https://www.aclweb.org/anthology/N13-1090`.

— (2013a). “Linguistic Regularities in Continuous Space Word Representations.” In: *Hlt-naacl*. Vol. 13, pp. 746–751.

Mikolov, Tomas et al. (2013a). “Distributed representations of words and phrases and their compositionality”. In: *Advances in neural information processing systems*, pp. 3111–3119.

Mikolov, Tomas et al. (2013b). “Efficient estimation of word representations in vector space”. In: *arXiv preprint arXiv:1301.3781*.

Miller, George A (1956). “The magical number seven, plus or minus two: some limits on our capacity for processing information.” In: *Psychological review* 63.2, p. 81.

Miller, Tim (2019). “Explanation in artificial intelligence: Insights from the social sciences”. In: *Artificial Intelligence* 267, pp. 1–38.

Milo, Ron et al. (2002). “Network motifs: simple building blocks of complex networks”. In: *Science* 298.5594, pp. 824–827.

Mnih, Volodymyr et al. (2015). “Human-level control through deep reinforcement learning”. In: *Nature* 518.7540, pp. 529–533.

Mojsilovic, Aleksandra and Aleksandra Mojsilovic (2020). *Introducing AI Explainability 360 — IBM Research Blog*. URL: `https://www.ibm.com/blogs/research/2019/08/ai-explainability-360/`.

Molnar, Christoph (2019). *Interpretable Machine Learning. A Guide for Making Black Box Models Explainable.* `https://christophm.github.io/interpretable-ml-book/`.

Molnar, Christoph et al. (2020). “Pitfalls to Avoid when Interpreting Machine Learning Models”. In: *CoRR* abs/2007.04131. arXiv: `2007.04131`. URL: `https://arxiv.org/abs/2007.04131`.

Montavon, Grégoire (2019). “Gradient-based vs. propagation-based explanations: an axiomatic comparison”. In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, pp. 253–265.

Montavon, Grégoire, Mikio L Braun, and Klaus-Robert Müller (2011). “Kernel Analysis of Deep Networks.” In: *Journal of Machine Learning Research* 12.9.

Montavon, Grégoire, Wojciech Samek, and Klaus-Robert Müller (2018). “Methods for interpreting and understanding deep neural networks”. In: *Digital Signal Processing* 73, pp. 1–15.

Montavon, Grégoire et al. (2017). “Explaining nonlinear classification decisions with deep taylor decomposition”. In: *Pattern Recognition* 65, pp. 211–222.

Montufar, Guido F et al. (2014). “On the number of linear regions of deep neural networks”. In: *Advances in neural information processing systems*, pp. 2924–2932.

Moosavi-Dezfooli, Seyed-Mohsen et al. (2019). “Robustness via curvature regularization, and vice versa”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9078–9086.

Mordvintsev, Alexander, Christopher Olah, and Mike Tyka (2015). “Inceptionism: Going deeper into neural networks”. In:

Muandet, Krikamol, David Balduzzi, and Bernhard Schölkopf (2013). "Domain Generalization via Invariant Feature Representation". In: *Proceedings of the 30th International Conference on Machine Learning.* Ed. by Sanjoy Dasgupta and David McAllester. Vol. 28. Proceedings of Machine Learning Research 1. Atlanta, Georgia, USA: PMLR, pp. 10–18. URL: http://proceedings.mlr.press/v28/muandet13.html.

Murdoch, W James et al. (2019). "Interpretable machine learning: definitions, methods, and applications". In: *arXiv preprint arXiv:1901.04592.*

Murphy, Kevin P. (2012). *Machine learning - a probabilistic perspective.* Adaptive computation and machine learning series. MIT Press. ISBN: 0262018020.

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814.

Narayanan, Hariharan and Sanjoy Mitter (2010). "Sample complexity of testing the manifold hypothesis". In: *Advances in neural information processing systems*, pp. 1786–1794.

Newell, Allen, Herbert Alexander Simon, et al. (1972). *Human problem solving.* Vol. 104. 9. Prentice-Hall Englewood Cliffs, NJ.

Nguyen, Anh, Jason Yosinski, and Jeff Clune (2015). "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436.

— (2016). "Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks". In: *arXiv preprint arXiv:1602.03616.*

Nguyen, Anh et al. (2016). "Synthesizing the preferred inputs for neurons in neural networks via deep generator networks". In: *Advances in Neural Information Processing Systems 29.* Ed. by D. D. Lee et al. Curran Associates, Inc., pp. 3387–3395. URL: http://papers.nips.cc/paper/6519-synthesizing-the-preferred-inputs-for-neurons-in-neural-networks-via-deep-generator-networks.pdf.

Nguyen, Anh et al. (2017). "Plug & play generative networks: Conditional iterative generation of images in latent space". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4467–4477.

Olah, Chris, Alexander Mordvintsev, and Ludwig Schubert (2017). "Feature Visualization". In: *Distill.* DOI: 10.23915/distill.00007. URL: https://distill.pub/2017/feature-visualization.

Olah, Chris et al. (2018). "The Building Blocks of Interpretability". In: *Distill.* DOI: 10.23915/distill.00010. URL: https://distill.pub/2018/building-blocks.

Olshausen, Bruno A and David J Field (1997). "Sparse coding with an overcomplete basis set: A strategy employed by V1?" In: *Vision research* 37.23, pp. 3311–3325.

Pan, Sinno Jialin and Qiang Yang (2010). “A Survey on Transfer Learning”. In: *IEEE Trans. Knowl. Data Eng.* 22.10, pp. 1345–1359. DOI: 10.1109/TKDE.2009.191. URL: https://doi.org/10.1109/TKDE.2009.191.

Pascanu, Razvan, Guido Montufar, and Yoshua Bengio (2013). “On the number of response regions of deep feed forward networks with piece-wise linear activations”. In: *arXiv preprint arXiv:1312.6098*.

Pascanu, Razvan, Guido Montúfar, and Yoshua Bengio (2014). “On the number of inference regions of deep feed forward networks with piece-wise linear activations”. In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1312.6098.

Paulus, Romain, Caiming Xiong, and Richard Socher (2017). “A Deep Reinforced Model for Abstractive Summarization”. In: *CoRR* abs/1705.04304. URL: http://arxiv.org/abs/1705.04304.

Pearl, Judea (2009). *Causality: Models, reasoning and inference*. 2nd ed. Cambridge, MA, USA: Cambridge University Press.

Pearl, Judea and Dana Mackenzie (2018). *The book of why: the new science of cause and effect*. Basic Books.

Pearson, Karl (1897). “Mathematical contributions to the theory of evolution.—on a form of spurious correlation which may arise when indices are used in the measurement of organs”. In: *Proceedings of the royal society of london* 60.359-367, pp. 489–498.

Pedregosa, F. et al. (2011). “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12, pp. 2825–2830.

Pérez, Guillermo Valle, Chico Q. Camargo, and Ard A. Louis (2019). “Deep learning generalizes because the parameter-function map is biased towards simple functions”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net. URL: https://openreview.net/forum?id=rye4g3AqFm.

Plate, Tony (2006). “Distributed representations”. In: *Encyclopedia of Cognitive Science*.

Pohl, Rüdiger and Rüdiger F Pohl (2004). *Cognitive illusions: A handbook on fallacies and biases in thinking, judgement and memory*. Psychology Press.

Poole, Ben et al. (2016). “Exponential expressivity in deep neural networks through transient chaos”. In: *Advances in neural information processing systems*, pp. 3360–3368.

Poursabzi-Sangdeh, Forough et al. (2018). “Manipulating and measuring model interpretability”. In: *arXiv preprint arXiv:1802.07810*.

Pruthi, Danish et al. (2019). *Learning to Deceive with Attention-Based Explanations*. arXiv: 1909.07913 [cs.CL].

Qu, Guangzhi, Salim Hariri, and Mazin Yousif (2005a). “A new dependency and correlation analysis for features”. In: *IEEE Transactions on Knowledge and Data Engineering* 17.9, pp. 1199–1207.

— (2005b). “A new dependency and correlation analysis for features”. In: *IEEE Transactions on Knowledge and Data Engineering* 17.9, pp. 1199–1207.

Quinlan, J. Ross (1986). “Induction of decision trees”. In: *Machine learning* 1.1, pp. 81–106.

Quinlan, J Ross (1999). “Some elements of machine learning”. In: *International Conference on Inductive Logic Programming*. Springer, pp. 15–18.

Quiroga, R Quian et al. (2005). “Invariant visual representation by single neurons in the human brain”. In: *Nature* 435.7045, pp. 1102–1107.

Radford, Alec, Luke Metz, and Soumith Chintala (2016). “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”. In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1511.06434.

Raghu, Maithra et al. (2017). “On the expressive power of deep neural networks”. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 2847–2854.

Rauber, Paulo E et al. (2017). “Visualizing the Hidden Activity of Artificial Neural Networks”. In: *IEEE Transactions on Visualization and Computer Graphics* 23.1, pp. 101–110.

Read, Stephen J and Amy Marcus-Newhall (1993). “Explanatory coherence in social explanations: A parallel distributed processing account.” In: *Journal of Personality and Social Psychology* 65.3, p. 429.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra (2014). “Stochastic backpropagation and variational inference in deep latent gaussian models”. In: *International Conference on Machine Learning*. Vol. 2.

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin (2016). “Why Should I Trust You?: Explaining the Predictions of Any Classifier”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 1135–1144.

— (2018). “Anchors: High-precision model-agnostic explanations”. In: *Thirty-Second AAAI Conference on Artificial Intelligence*.

Rifai, Salah et al. (2011a). “Contractive Auto-Encoders: Explicit Invariance During Feature Extraction”. In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*. Ed. by Lise Getoor and Tobias Scheffer. Omnipress, pp. 833–840. URL: https://icml.cc/2011/papers/455_icmlpaper.pdf.

Rifai, Salah et al. (2011b). “The Manifold Tangent Classifier”. In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor et al. Curran Associates, Inc., pp. 2294–2302. URL: http://papers.nips.cc/paper/4409-the-manifold-tangent-classifier.pdf.

Robnik-Šikonja, Marko and Marko Bohanec (2018). “Perturbation-based explanations of prediction models”. In: *Human and machine learning*. Springer, pp. 159–175.

Robnik-Šikonja, Marko and Igor Kononenko (2008). “Explaining classifications for individual instances”. In: *IEEE Transactions on Knowledge and Data Engineering* 20.5, pp. 589–600.

Rocktäschel, Tim et al. (2016). “Reasoning about Entailment with Neural Attention”. In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. URL: http://arxiv.org/abs/1509.06664.

Rosenfeld, Ronald and David S Touretzky (1987). *Four capacity models for coarse-coded symbol memories*. Tech. rep. CARNEGIE-MELLON UNIV PITTSBURGH PA ARTIFICIAL INTELLIGENCE and PSYCHOLOGY.

Rudin, Cynthia and Berk Ustun (2018). “Optimized scoring systems: Toward trust in machine learning for healthcare and criminal justice”. In: *Interfaces* 48.5, pp. 449–466.

Russakovsky, Olga et al. (2015). “ImageNet Large Scale Visual Recognition Challenge”. In: *International Journal of Computer Vision (IJCV)* 115.3, pp. 211–252. DOI: 10.1007/s11263-015-0816-y.

Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton (2017). “Dynamic Routing Between Capsules”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al., pp. 3856–3866. URL: http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.

Samek, Wojciech et al. (2017). “Evaluating the visualization of what a deep neural network has learned”. In: *IEEE transactions on neural networks and learning systems*.

Samek, Wojciech et al. (2019). *Explainable AI: interpreting, explaining and visualizing deep learning*. Vol. 11700. Springer Nature.

Sato, Makoto and Hiroshi Tsukimoto (2001). “Rule extraction from neural networks via decision tree induction”. In: *IJCNN’01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222)*. Vol. 3. IEEE, pp. 1870–1875.

Saul, Lawrence K and Sam T Roweis (2003). “Think globally, fit locally: unsupervised learning of low dimensional manifolds”. In: *Journal of machine learning research* 4.Jun, pp. 119–155.

Schank, RP (1986). *Explanation patterns: Understanding mechanically and creatively*. Lawrence Erlbaum Associates.

Schmidt, Ludwig et al. (2018). “Adversarially robust generalization requires more data”. In: *Advances in Neural Information Processing Systems*, pp. 5014–5026.

Schölkopf, Bernhard, Alexander Smola, and Klaus-Robert Müller (1998). “Nonlinear component analysis as a kernel eigenvalue problem”. In: *Neural computation* 10.5, pp. 1299–1319.

Schölkopf, Bernhard et al. (2012). “On causal and anticausal learning”. In: *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress. URL: http://icml.cc/2012/papers/625.pdf.

Sculley, D et al. (2015). “Hidden technical debt in machine learning systems”. In: *Advances in Neural Information Processing Systems*, pp. 2503–2511.

Selvaraju, Ramprasaath R. et al. (2016). “Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based Localization”. In: *CoRR* abs/1610.02391. arXiv: 1610.02391. URL: http://arxiv.org/abs/1610.02391.

Semenova, Lesia and Cynthia Rudin (2019). “A study in Rashomon curves and volumes: A new perspective on generalization and model simplicity in machine learning”. In: *arXiv preprint arXiv:1908.01755*.

Shams, Zohreh et al. (2021). “REM: An Integrative Rule Extraction Methodology for Explainable Data Analysis in Healthcare”. In: *bioRxiv*.

Shapley, Lloyd S (1953). “A value for n-person games”. In: *Contributions to the Theory of Games* 2.28, pp. 307–317.

Shimodaira, Hidetoshi (2000). “Improving predictive inference under covariate shift by weighting the log-likelihood function”. In: *Journal of statistical planning and inference* 90.2, pp. 227–244.

Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje (2017). “Learning important features through propagating activation differences”. In: *International Conference on Machine Learning (ICML)*.

Shrikumar, Avanti et al. (2016). “Not just a black box: Learning important features through propagating activation differences”. In: *arXiv preprint arXiv:1605.01713*.

Simard, Patrice et al. (1992). “Tangent prop-a formalism for specifying selected invariances in an adaptive network”. In: *Advances in neural information processing systems*, pp. 895–903.

Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2013). “Deep inside convolutional networks: Visualising image classification models and saliency maps”. In: *arXiv preprint arXiv:1312.6034*.

Simonyan, Karen and Andrew Zisserman (2014). “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *CoRR* abs/1409.1556. arXiv: 1409.1556. URL: http://arxiv.org/abs/1409.1556.

Sinha, Aman, Hongseok Namkoong, and John C. Duchi (2018). “Certifying Some Distributional Robustness with Principled Adversarial Training”. In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net. URL: https://openreview.net/forum?id=Hk6kPgZA-.

Sixt, Leon, Maximilian Granz, and Tim Landgraf. “When Explanations Lie: Why Many Modified BP Attributions Fail”. In: ().

Slack, Dylan et al. (2019). “How can we fool LIME and SHAP? Adversarial Attacks on Post hoc Explanation Methods”. In: *arXiv preprint arXiv:1911.02508*.

Smilkov, Daniel et al. (2017). “Smoothgrad: removing noise by adding noise”. In: *arXiv preprint arXiv:1706.03825*.

Smith, Leslie N. (2018). “A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay”. In: *CoRR* abs/1803.09820. arXiv: 1803.09820. URL: http://arxiv.org/abs/1803.09820.

Sokal, Robert R (1958). “A statistical method for evaluating systematic relationship”. In: *University of Kansas science bulletin* 28, pp. 1409–1438.

Sontag, Eduardo D (1998). “VC dimension of neural networks”. In: *NATO ASI Series F Computer and Systems Sciences* 168, pp. 69–96.

Sotelo, Jose et al. (2017). “Char2Wav: End-to-end speech synthesis”. In: *ICLR2017 workshop submission*.

Speicher, Till et al. (2018). “A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual &Group Unfairness via Inequality Indices”. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, pp. 2239–2248.

Springenberg, JT et al. (2015). “Striving for simplicity: The all convolutional neural net”. In: *Int. Conf. Learning Representations (ICLR)*.

Srivastava, Nitish et al. (2014). “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Steels, Luc (1985). “Second generation expert systems”. In: *Future generation computer systems* 1.4, pp. 213–221.

Storkey, Amos (2009). “When training and test sets are different: characterizing learning transfer”. In: *Dataset shift in machine learning*, pp. 3–28.

Strumbelj, Erik and Igor Kononenko (2010). “An efficient explanation of individual classifications using game theory”. In: *The Journal of Machine Learning Research* 11, pp. 1–18.

Štrumbelj, Erik, Igor Kononenko, and M Robnik Šikonja (2009). "Explaining instance classifications with interactions of subsets of feature values". In: *Data & Knowledge Engineering* 68.10, pp. 886–904.

Sugiyama, Masashi and Motoaki Kawanabe (2012). *Machine Learning in Non-Stationary Environments - Introduction to Covariate Shift Adaptation.* Adaptive computation and machine learning. MIT Press. ISBN: 978-0-262-01709-1. URL: `http://mitpress.mit.edu/books/machine-learning-non-stationary-environments`.

Sullins, John (1985). *Value cell encoding strategies.* Tech. rep. Computer Science Department, University of Rochester, Rochester NY.

Sundararajan, Mukund, Ankur Taly, and Qiqi Yan (2017). "Axiomatic attribution for deep networks". In: *International Conference on Machine Learning (ICML).*

Sutskever, Ilya, Oriol Vinyals, and Quoc V Le (2014). "Sequence to sequence learning with neural networks". In: *Advances in neural information processing systems*, pp. 3104–3112.

Szegedy, Christian et al. (2014). "Intriguing properties of neural networks". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings.* Ed. by Yoshua Bengio and Yann LeCun. URL: `http://arxiv.org/abs/1312.6199`.

Szegedy, Christian et al. (2015). "Going deeper with convolutions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9.

Szegedy, Christian et al. (2016). "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826.

Szegedy, Christian et al. (2017). "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.* Ed. by Satinder P. Singh and Shaul Markovitch. AAAI Press, pp. 4278–4284. URL: `http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14806`.

Tan, Sarah et al. (2019). *Learning Global Additive Explanations for Neural Nets Using Model Distillation.* URL: `https://openreview.net/forum?id=SJl8J30qFX`.

Tenenbaum, Joshua B, Vin De Silva, and John C Langford (2000). "A global geometric framework for nonlinear dimensionality reduction". In: *science* 290.5500, pp. 2319–2323.

Tenenbaum, Joshua B and William T Freeman (1997). "Separating style and content". In: *Advances in neural information processing systems*, pp. 662–668.

Thorburn, William M (1918). "The myth of Occam's razor". In: *Mind* 27.107, pp. 345–353.

Thorne, James et al. (2019). "Generating Token-Level Explanations for Natural Language Inference". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* *NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers).* Ed. by Jill Burstein, Christy Doran, and Thamar Solorio. Association for Computational Linguistics, pp. 963–969. DOI: `10.18653/v1/n19-1101`. URL: `https://doi.org/10.18653/v1/n19-1101`.

Tieleman, Tijmen and Geoffrey Hinton (2012). “Lecture 6.5-rmsprop, coursera: Neural networks for machine learning”. In: *University of Toronto, Technical Report.*

Tintarev, Nava and Judith Masthoff (2011). “Designing and evaluating explanations for recommender systems”. In: *Recommender Systems Handbook.* Springer, pp. 479–510.

Tomsett, Richard et al. (2018). “Interpretable to whom? A role-based model for analyzing interpretable machine learning systems”. In: *arXiv preprint arXiv:1806.07552.*

Tsipras, Dimitris et al. (2019). “Robustness May Be at Odds with Accuracy”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.* OpenReview.net. URL: `https://openreview.net/forum?id=SyxAb30cY7`.

Tukey, John W (1977). *Exploratory data analysis.* Reading, Mass.

Tyka, Mike (2016). *Class visualization with bilateral filters.*

Ustun, Berk and Cynthia Rudin (2016). “Supersparse linear integer models for optimized medical scoring systems”. In: *Machine Learning* 102.3, pp. 349–391.

Van Gelder, Tim (2013). “What is the” D” in” PDP”? A survey of the concept of distribution”. In: *Philosophy and connectionist theory.* Psychology Press, pp. 47–74.

Van Melle, William (1980). *A Domain-Independent System that Aids in Constructing Knowledge-Based Consultation Programs.* Tech. rep. DTIC Document.

Varshney, Kush R and Homa Alemzadeh (2016). “On the Safety of Machine Learning: Cyber-Physical Systems, Decision Sciences, and Data Products”. In: *arXiv preprint arXiv:1610.01256.*

Vaughan, Joel et al. (2018). “Explainable neural networks based on additive index models”. In: *arXiv preprint arXiv:1806.01933.*

Verma, Vikas et al. (2019). “Manifold Mixup: Better Representations by Interpolating Hidden States”. In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA.* Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 6438–6447. URL: `http://proceedings.mlr.press/v97/verma19a.html`.

Waa, Jasper van der et al. (2018). “Contrastive explanations with local foil trees”. In: *arXiv preprint arXiv:1806.07470.*

Wachter, Sandra, Brent Mittelstadt, and Chris Russell (2017). “Counterfactual explanations without opening the black box: Automated decisions and the GDPR”. In: *Harv. JL & Tech.* 31, p. 841.

Wah, Catherine et al. (2011). "The caltech-ucsd birds-200-2011 dataset". In:

Walker, Marilyn, Heng Ji, and Amanda Stent (2018). "Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers).*

Wang, Haohan, Zexue He, and Eric P. Xing (2019). "Learning Robust Representations by Projecting Superficial Statistics Out". In: *International Conference on Learning Representations*. URL: `https://openreview.net/forum?id=rJEjjoR9K7`.

Wang, Haohan et al. (2017). "Select-additive learning: Improving generalization in multimodal sentiment analysis". In: *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, pp. 949–954.

Wang, Tao et al. (2012). "End-to-end text recognition with convolutional neural networks". In: *Pattern Recognition (ICPR), 2012 21st International Conference on.* IEEE, pp. 3304–3308.

Wang, Tong et al. (2016). "Bayesian rule sets for interpretable classification". In: *Data Mining (ICDM), 2016 IEEE 16th International Conference on.* IEEE, pp. 1269–1274.

Weinberger, Kilian Q, Fei Sha, and Lawrence K Saul (2004). "Learning a kernel matrix for nonlinear dimensionality reduction". In: *Proceedings of the twenty-first international conference on Machine learning*, p. 106.

Weller, Adrian (2017). "Challenges for transparency". In: *ICML Workshop on Human Interpretability.*

— (2019). "Transparency: motivations and challenges". In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning.* Springer, pp. 23–40.

Wiegreffe, Sarah and Yuval Pinter (2019). "Attention is not not Explanation". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019.* Ed. by Kentaro Inui et al. Association for Computational Linguistics, pp. 11–20. DOI: `10.18653/v1/D19-1002`. URL: `https://doi.org/10.18653/v1/D19-1002`.

Wijaya, Maleakhi A et al. (2021). "Failing Conceptually: Concept-Based Explanations of Dataset Shift". In:

Williams, Ronald J (1986). "Inverting a connectionist network mapping by backpropagation of error". In: *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, pp. 859–865.

Wiskott, Laurenz and Terrence J Sejnowski (2002). "Slow feature analysis: Unsupervised learning of invariances". In: *Neural computation* 14.4, pp. 715–770.

Wolpert, David H (1996). “The lack of a priori distinctions between learning algorithms”. In: *Neural computation* 8.7, pp. 1341–1390.

Wong, Eric and J. Zico Kolter (2018). “Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope”. In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, pp. 5283–5292. URL: `http://proceedings.mlr.press/v80/wong18a.html`.

Woodward, James (2005). *Making things happen: A theory of causal explanation.* Oxford university press.

Xiao, Kai Y. et al. (2019). “Training for Faster Adversarial Robustness Verification via Inducing ReLU Stability”. In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.* OpenReview.net. URL: `https://openreview.net/forum?id=BJfIVjAcKm`.

Xu, Kelvin et al. (2015). “Show, attend and tell: Neural image caption generation with visual attention”. In: *International Conference on Machine Learning*, pp. 2048–2057.

Yao, Andrew Chi-Chih (1985). “Separating the polynomial-time hierarchy by oracles”. In: *26th Annual Symposium on Foundations of Computer Science (sfcs 1985).* IEEE, pp. 1–10.

Yeh, Chih-Kuan et al. (2018). “Representer point selection for explaining deep neural networks”. In: *Advances in neural information processing systems*, pp. 9291–9301.

Yeh, Chih-Kuan et al. (2019). “On Concept-Based Explanations in Deep Neural Networks”. In: *arXiv preprint arXiv:1910.07969.*

Yosinski, Jason et al. (2014). “How transferable are features in deep neural networks?” In: *Advances in neural information processing systems*, pp. 3320–3328.

Yosinski, Jason et al. (2015). “Understanding neural networks through deep visualization”. In: *arXiv preprint arXiv:1506.06579.*

Zahavy, Tom, Nir Ben-Zrihem, and Shie Mannor (2016). “Graying the black box: Understanding dqns”. In: *International Conference on Machine Learning*, pp. 1899–1908.

Zanker, Markus (2012). “The influence of knowledgeable explanations on users’ perception of a recommender system”. In: *Proceedings of the sixth ACM conference on Recommender systems.* ACM, pp. 269–272.

Zaslavsky, Thomas (1975). “Counting the faces of cut-up spaces”. In: *Bulletin of the American Mathematical Society* 81.5, pp. 916–918.

Zeiler, Matthew D and Rob Fergus (2014). “Visualizing and understanding convolutional networks”. In: *European conference on computer vision.* Springer, pp. 818–833.

Zemel, Richard S, Peter Dayan, and Alexandre Pouget (1998). “Probabilistic interpretation of population codes”. In: *Neural computation* 10.2, pp. 403–430.

Zhang, Chiyuan, Samy Bengio, and Yoram Singer (2019). “Are all layers created equal?” In: *ICML 2019 Workshop Deep Phenomena.*

Zhang, Hongyang et al. (2019). “Theoretically Principled Trade-off between Robustness and Accuracy”. In: *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, pp. 7472–7482. URL: `http://proceedings.mlr.press/v97/zhang19p.html`.

Zhang, Jianming et al. (2016). “Top-down neural attention by excitation backprop”. In: *European Conference on Computer Vision.* Springer, pp. 543–559.

Zhang, Kun et al. (2013). “Domain adaptation under target and conditional shift”. In: *International Conference on Machine Learning*, pp. 819–827.

Zhang, Quanshi and Song-Chun Zhu (2018). “Visual interpretability for deep learning: a survey”. In: *Frontiers Inf. Technol. Electron. Eng.* 19.1, pp. 27–39. DOI: `10.1631/FITEE.1700808`. URL: `https://doi.org/10.1631/FITEE.1700808`.

Zhou, Bolei et al. (2014). “Object detectors emerge in deep scene cnns”. In: *arXiv preprint arXiv:1412.6856.*

— (2016). “Learning deep features for discriminative localization”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929.

Zhou, Bolei et al. (2018). “Interpretable basis decomposition for visual explanation”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 119–134.

Zhou, Dengyong et al. (2004). “Learning with local and global consistency”. In: *Advances in Neural Information Processing Systems 16.*

Zhou, Jianlong and Fang Chen (2018a). “2D Transparency Space—Bring Domain Users and Machine Learning Experts Together”. In: *Human and Machine Learning.* Springer, pp. 3–19.

— (2018b). *Human and machine learning: Visible, explainable, trustworthy and transparent.* Springer.

Zhou, Yi-Tong and Rama Chellappa (1988). “Computation of optical flow using a neural network”. In: *IEEE International Conference on Neural Networks.* Vol. 1998, pp. 71–78.

Zilke, Jan Ruben, Eneldo Loza Mencía, and Frederik Janssen (2016). “Deepred–rule extraction from deep neural networks”. In: *International Conference on Discovery Science.* Springer, pp. 457–473.

Zintgraf, Luisa M. et al. (2017). “Visualizing Deep Neural Network Decisions: Prediction Difference Analysis”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings.* OpenReview.net. URL: `https://openreview.net/forum?id=BJ5UeU9xx`.

Zou, Hui, Trevor Hastie, and Robert Tibshirani (2006). “Sparse principal component analysis”. In: *Journal of computational and graphical statistics* 15.2, pp. 265–286.

Zurada, Jacek M, Aleksander Malinowski, and Ian Cloete (1994). “Sensitivity analysis for minimization of input data dimension for feedforward neural network”. In: *Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS’94*. Vol. 6. IEEE, pp. 447–450.

APPENDIX A

Machine Learning Fundamentals

Success is neither magical nor mysterious. Success is the natural consequence of consistently applying the basic fundamentals.

Jim Rohn

Causal vs Anticausal To see the difference between causal and anticausal predictions, let's consider the causal structure of two random variables – a cause c and effect e. The causal mechanism $p(\mathrm{e}|\mathrm{c})$ describes the transformation from cause c into effect e, while we denote variables $\mathbf{x}$ and $\mathbf{y}$ as the input and output. Notice that the cause and effect variables can each be an either an input or output of a prediction model.

The situation of a *causal* prediction occurs when the input $\mathbf{x}$ causes the output $\mathbf{y}$ as an effect. *Anticausal* predictions consider the opposite direction, in which the input is the effect of the cause that we are trying to predict. Although this might seem unnatural at first glance, it a frequently occurring phenomenon. Consider the popular handwritten digit recognition task MNIST (Deng, 2012). A human decides to write the digit 1, and this intention causes a particular pattern and in that way the output class label 1 (y_1) caused the input image ($\boldsymbol{x}$).

The exogeneity assumption states that the causal mechanism $p(\mathrm{e}|\mathrm{c})$ and cause c are independent, i.e, $p(\mathrm{e}|\mathrm{c})$ contains no information about $p(\mathrm{c})$ and vice versa (Pearl, 2009).

For example, let's look at the posterior

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})}.$$

In the causal case, if $p(\mathbf{x})$ changes than $p(\mathbf{y}|\mathbf{x})$ changes, but $p(\mathbf{y}|\mathbf{x})$ does not change. In what we can interpret this as the "laws of the universe" do not change, although the distribution of the cause changes.

Spurious correlations Spurious correlations are co-occurrence of frequently appearing artefacts that obscure the true cause and effect relationship (Pearson, 1897). These artefacts are present in the data; however, in reality, they do not correspond to meaningful information, making the signal **unstable**. That is, we would not expect the correlation to hold in the same way in the future, as it did in the past (Woodward, 2005). For example, if our dataset contained only images of phones, which are used by people, there is a spurious correlation that for something to be a phone, there has to be a human. Spurious correlation can cause extremely unreliable predictions such as a prediction that a person is speaking on the phone just a phone and a person are present in an image (Lopez-Paz, 2016; Woodward, 2005; Lake et al., 2017). Arjovsky et al. (2019) demonstrate that subtle changes to the background (landscapes and contexts), colouring or texture of images break powerful image classifiers. For example, a cow on a beach is classified as a camel, whereas a camel on a grassy meadow is classified as a cow.

Confounding factors A confounding factor is a causal concept, which influences two conditionally independent variables. The variables are independent if the confounding factor is observed; however, there is a spurious correlation between the two variables when the factor is unobserved or hidden.

In particular, confounding factors have a special causal relationship. For example, a *job occupation* "retired" is the effect of the causal feature *age*. If we see *education* feature "primary" we would not expect to see job occupation "retired" because the education gives us information about the age, which in turn gives us information about the job occupation. On the other hand, if the *age* is known (e.g., "above 65"), then the two variables become independent. The challenge of confounders is that they lead to spurious correlations.

Data distribution changes Data distribution changes can occur either at the stage of deployment, when the cause is the difference between the training and prediction distributions, or gradually over time because the world is dynamic and evolves. For example, the street numbers font, size and colour could change over the years, but it does not change instantaneously for all houses. This situation requires out-of-distribution (o.o.d) generalisation, which is the ability of a representation to generalise to unseen samples, samples of different nature or differently distributed samples.

Covariate shift is the situation in which the distributions of the training and test data differ in the sense that the distribution of the inputs $p(\boldsymbol{x})$ change, however, the conditional distribution ($p(\boldsymbol{y}|\boldsymbol{x})$) remains the same (Sugiyama and Kawanabe, 2012; Shimodaira, 2000).

Label shift or prior probability shift is a particular case of covariate shift, in which only the distribution of the outputs changes (Storkey, 2009; Schölkopf et al., 2012; Zhang et al., 2013; Lipton, Wang, and Smola, 2018). This should not be confused with **heteroscedastic** models, for which the variance of the labels changes depending on the inputs. This settings is the opposite of **homoskedastic** models, in which the variance of the labels is constant.

Concept drift is a related situation, in which contrary to the covaraite shift case, the prediction distribution does not change, but the causal mechanism changes, such that factor function that describes the transformation of the cause c into effect e changes (e.g. $\phi(c, N_e)$, where N_e is some random noise on the effect). That is to say concept drift is not related to the input or output distributions, but to the relationship between them (Storkey, 2009; Schölkopf et al., 2012; Zhang et al., 2013; Lipton, Wang, and Smola, 2018).

Domain adaptation is the goal of designing machine learning algorithms that generalise across more significant changes in the input data distributions that *change the nature of the input*. For example, we can train a sentiment analysis classifier to assign positive or negative sentiment to *news articles*. We would perform domain adaptation when we attempt to perform sentiment analysis on *movie reviews* (Gretton et al., 2009; Shimodaira, 2000).

Transfer learning There have been some debates in the research community, whether transfer learning is a related or a general case of domain adaption. Recent surveys (Pan and Yang, 2010; Kouw, 2018) suggest that transfer learning is a related case to domain adaptation, in which the *input is the same*, but the *target output* might be of a *different* kind. That is the task of learning an input-output mapping changes. For example, we can build a vision classifier to recognise images of animals. We may than want to learn about a new setting, in which we still recognise images, but this time of vehicles. This is a particularly powerful technique when there is a significantly larger amount of data available for the first setting, which would help the algorithm generalise faster with fewer samples to the second setting. Transfer learning assumes that the two settings share a vast number of low-level features such as edges, shapes, positions, etc.

Multi-task learning can be see as an extension of concept drift and transfer learning due to the fact that the input remains the same across tasks, while the output changes (Caruana, 1993). In contrast to the two previous approaches, here we aim to perform two or more different tasks at the same time. The rational behind this approach is that the factors that explain the variation in the observed data could be useful both for task A and another task B. That is to say that when we apply multi-task learning, we are relying on the underlying belief of shared factors (described in Section 2.2). For example, a representation that is useful to translate sentences from English to German, could learn useful factors of variations about English, so that it can be reused to translate

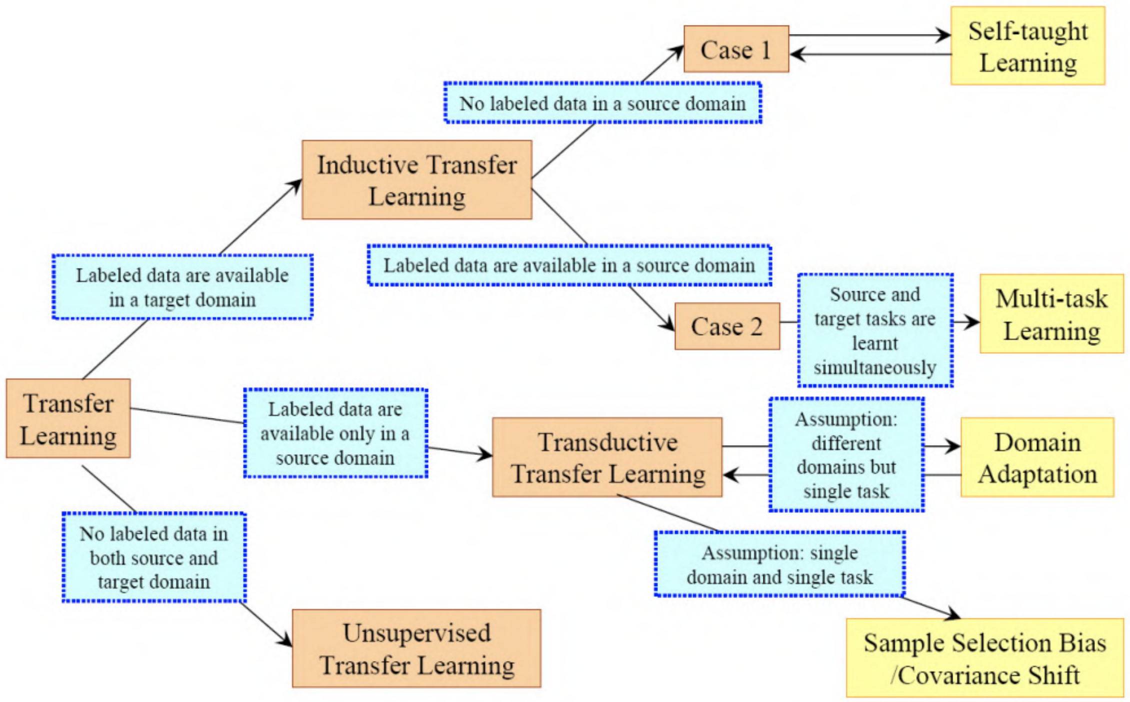

Case 1
Self-taught Learning
No labeled data in a source domain
Inductive Transfer Learning
Labeled data are available in a target domain
Labeled data are available in a source domain
Case 2
Source and target tasks are learnt simultaneously
Multi-task Learning
Transfer Learning
Labeled data are available only in a source domain
Transductive Transfer Learning
Assumption: different domains but single task
Domain Adaptation
No labeled data in both source and target domain
Assumption: single domain and single task
Unsupervised Transfer Learning
Sample Selection Bias /Covariance Shift

Figure A.1: Overview of different types of transfer learning. Image reproduced from (Pan and Yang, 2010).

from English to French. Hence, we can build a representation learning algorithm that can translate from English to many other languages.

Domain generalisation Domain generalisation (Muandet, Balduzzi, and Schölkopf, 2013) is an extension of domain adaptation and transfer learning, in which there are no available samples from the target distribution during training. This situation occurs because datasets often contain data from very heterogeneous sources, collected using different practises (Wang et al., 2017). Another example of this situation is doing predictive analytics on biological cells, in which each patient has a different distribution of cells.

Independently identically distributed and out-of-distribution Out-of-distribution predictions is a special case of distribution shift, in which selection, implicit, and over-generalisation bias [1] have lead to the situation in which our dataset does not contain all possible variations within the observations. Intuitively, imagine that the observed data is generated by a mixture model with n components. However, we have observed only $k \ll n$ components. That means there are regions of the input space, which do not behave as any of the regions that we have observed. Nevertheless, these regions are the result of a well-defined generative process. Despite that the structure of the unobserved regions

[1] See Section 3.5.2.3 for definitions of different types of cognitive bias.

is different from from all others, it can still be inferred using shared factors from the known regions. Let us imagine a European driver coming to the UK. He/She can infer that driving on the left side of the road requires the mirror actions of driving on the right side of the road without ever experiencing left road driving. For example, overtaking now necessitates moving from the left to the right line rather than vice versa. In that respect out-of-distribution predictions can include many different forms of distribution shift such as covariate shift, label shift or concept drift simultaneously.

APPENDIX B

Representation Learning

B.1 Representation Learning Assumptions

- **Multiple factors assumption**: Assumes that there are more than one factors of variation that explain the observed data. For example, if we take the 3D objects example, the lighting factor on its own would not be enough to explain the pixel intensities. This assumption allows us to easily solve any task provided we can capture and disentangle its key explanatory factors. Section 2.5 describes how this assumption motivates distributed representations with separate control over directions in representation space, such that each entry represents a factor of variation.

- **Causal factors assumption**: Assumes that the generative process is such that the observed data is an effect of the underlying factors of variation, and not vice versa. In this case, if the learned representation truly captures the factors of variation, then its elements represent the causes of the observed data (Schölkopf et al., 2012; Erhan et al., 2010). Hence, the 3D object lighting causes the pixel intensity increase rather than the pixel intensities causing the object to appear brighter. When this assumption holds, the learned model is more robust to changes in the input distribution because these changes are driven by shifts in the distribution of the underlying causal factors. For example, if we assume that $p(\mathbf{x})$ and $p(\mathbf{y}|\mathbf{x})$ are independent (i.e., the exogeneity assumption[1]), then changes in $p(\mathbf{x})$ do not interfere with our model of $p(\mathbf{y}|\mathbf{x})$ (Lasserre, Bishop, and Minka, 2006).

- **Shared factors assumption**: Assumes that different tasks share factors across a common pool of reusable latent factors of variation. Therefore, using one task to extract underlying factors of variation should be beneficial to discover factors relevant to other tasks. Transferring statistical strength of reusable features across tasks and

[1] See Appendix A.

domains motivates the successful application of representation learning algorithms to multi-task learning (Collobert et al., 2011), transfer learning (Goodfellow, Courville, and Bengio, 2012), and domain adaptation (Glorot, Bordes, and Bengio, 2011). As we will see in the next two assumptions and through this chapter, the ability to represent many examples with reusable features projects the input into a rich similarity space, where multiple examples are not constrained to be only *local* neighbours in input space. Therefore, this assumption results in exponential gain in the expressivity of the representation[2].

- **Hierarchical organisation assumption**: Assumes that the world is described by highly complex functions with a considerable degree of variation (ups and downs), but with an *underlying simple structure*, which is hierarchical. The rationale behind this assumption is that humans often describe concepts hierarchically with multiple levels of abstraction. For example, a software engineer prefers to represent information with a hierarchy of reusable components such as functions and modules rather than with one flat main program.

  While the shared factors assumption supposes the existence of reusable components, the hierarchical organisation assumption incorporates the belief that *a hierarchy of reusable components* can describe abstract ideas more easily. For example, we can describe the concept of cars through relationships about objects such as its parts (e.g., tires, windshields and doors). We can represent each of these objects with simpler shapes, such as rectangles, circles, and squares. The shapes can be represented through relationships between straight and curved lines. Naturally, concepts become more abstract as they become increasingly invariant to local input transformations, which are uninformative to the subsequent task.

  Assuming a hierarchical structure has a threefold benefit: (1) contributes to disentangling of factors of variation; (2) leads to exponential gains in representation power because it promotes the reuse of features; (3) induces a prior of building invariant features[3].

- **Manifolds assumption**: Assumes that the probability density of real-world high-dimensional data is highly concentrated along (often non-linear) connected regions of tiny volume (of much smaller dimensionality that the original space), called manifolds (Cayton, 2005; Narayanan and Mitter, 2010; Schölkopf, Smola, and Müller, 1998; Saul and Roweis, 2003; Tenenbaum, De Silva, and Langford, 2000; Brand, 2003; Belkin and Niyogi, 2003; Donoho and Grimes, 2003; Weinberger, Sha, and Saul, 2004). A manifold is a region consisting of connected data points,

[2] See Section 2.5.

[3] In Appendix B.4.3, we discuss these benefits in more detail.

such that one point is similar to its surrounding points. Movements along the manifold correspond to specific allowable transformations in input space. For example, Figure 2.1 demonstrates how transitions along the y-axis of the learned manifold correspond to up-down pose changes in the original space.

Low-dimensional manifolds, with dimensionality much smaller than that of the original space, can be learned to approximate the input space. The learned representation forms an intrinsic coordinate system such that each dimension of the low-dimensional manifold captures local variations of the input. The highest variance is observed along directions tangent to the manifold, while directions orthogonal to the manifold have minimal variance. Since infinitesimal perturbations along the tangent planes of the manifold define allowed data transformations in input space, interpolating between points along the tangent directions can yield new valid points, which were not part of the original dataset. However, most of the input space consists of invalid datapoints because there are very few directions tangent to a low-dimensional manifold. There are five important factors related to learning the structure of a manifold (Bengio and Monperrus, 2005; Rifai et al., 2011b; Verma et al., 2019): (1) noise (i.e., datapoints might lie slightly outside the manifold); (2) curvature (i.e., the degree to which the geometry of the manifold deviates from being a straight line), (3) dimensionality, (4) density (i.e., how sparsely populated is the manifold), (5) number of the manifolds, and (6) curvature of the high-entropy regions between the manifolds (i.e., transitions). In Chapters 5 & 6, we show that we can associate these manifold structures within DNNs to concepts or particular outputs; therefore, enhancing our ability to understand these algorithms.

- **Natural clustering assumption**: Assumes that the points of different classes, or with distinct characteristics, are likely to concentrate along separate manifolds, whereas similar points concentrate along connected manifolds, such that local variations within a manifold do not change the class identity (Rifai et al., 2011b).

  Low-density regions in input space separate the manifolds in a way that the distances between manifolds carry information regarding the difference between the points. Due to this fact, this assumption is sometimes referred to as the **"disconnected manifolds assumption"** because small input perturbations should not be able to transition between manifolds (Rifai et al., 2011b; Bengio and Delalleau, 2011; Bengio, Courville, and Vincent, 2013).

  This manifold geometry induces a **rich similarity space**, in which objects distant apart in input space, come together to form clusters. The rich similarity space yields potent generalisation properties because we can now transfer the knowledge about one point to exponentially many more points on the corresponding manifold[4]. Although originally it is assumed that a manifold corresponds to a single class (Bengio and Delalleau, 2011; Verma et al., 2019), meaning class manifolds do not overlap much, results in Chapters 5 & 6 suggest the presence of overlapping manifolds.

- **Simple factor relationships assumption**: Assumes that simple dependencies describe the relations between factors. For example, the simplest form of relationship is marginal independence. When the explanatory factors are independent of each other, the knowledge of the distribution of one factor generalises to various configurations of the others. We make this assumption when we use a linear classifier such as the softmax final layer in neural networks on top of a linear combination of a learned representation (Goodfellow, Bengio, and Courville, 2016a). Hence, we expect that the deeper layers of the networks have learned more abstract and linearly separable features. More sophisticated forms of dependence (e.g., polynomials of low order such as linear, quadratic, cubic, or even quartic) are also reasonable assumptions. Although the degree of the polynomials that usually describes physical properties ranges between two and four (Lin and Tegmark, 2016), currently these high order dependencies are rarely used in practice because of the computational and statistical challenges they introduce[5].

- **Sparsity assumption**: Assumes that the learned features have a high correlation with very few explanatory factors and are invariant to others; consequently, most of the time a feature will not be used to describe an input. For instance, a feature describing a steering wheel, will not be active for an image of a bird. That is to say, if the features describe a binary state – "present" or "absent", we assume that most of the features are absent most of the time. This assumption motivates sparse representations, the intuition for which is that the degree of sparsity controls the insensitivity of a model to small input changes[6].

- **Smoothness (local constancy) assumption**: Assumes that the function we learn (target function) should remain relatively constant within the neighbourhoods of its inputs (i.e., if $u \approx v$, then $f(u) \approx f(v)$). This assumption implies implicitly that the function should change slowly and rarely (Barron, 1993), which allows estimators to generalise to *nearby input points*, also known as **local generalisation** (Goodfellow, Bengio, and Courville, 2016a). Although this is one of the most generic and powerful machine learning assumptions, it makes it difficult to generalise to complicated high-dimensional functions with numerous peaks and troughs that span multiple

[4] See Appendix B.4.

[5] In fact, modern DNNs have been shown to exhibit a strong bias towards simple functions (Pérez, Camargo, and Louis, 2019).

[6] In Appendix C we develop the relationship between sparsity and invariance further.

regions. As we will see in Section 2.5, when a learner relies exclusively on the "smoothness prior" to generalise, it requires at least the same number of example as the number of distinct regions in input space, the number of which can grow exponentially.

- **Linearity assumption**: Assumes predominantly linear relationships between input, factor and output variables. This assumption is a subset of the simple factor relationships assumption. It allows the estimator to generalise to *very far unobserved input points*. For gradient-based methods, it also makes the computation of derivatives significantly easier, leading to faster optimisation. Notice there are two differences with the smoothness assumption – generalisation to distant data points rather than local neighbourhoods, and the lack of constancy within a region. The two assumptions together encode the belief that the learned function should be locally constant and globally linear. The limitation of the linearity assumption is that high-dimensional linear functions are vulnerable to the accumulation of small imperceptible change across multiple dimensions. This can lead to highly confident incorrect predictions, known as adversarial examples (Goodfellow, Shlens, and Szegedy, 2015).

- **Temporal and spatial coherence or invariance assumption**: Assumes that the most salient factors of variation change slowly, or remain invariant (Heinze-Deml, Peters, and Meinshausen, 2018), and are easier to predict (Becker and Hinton, 1992) over time, space or modality (vision, sound, and touch). The assumption is inspired by the slowness principle (Hinton, 1990; Földiák and Fdilr, 1989), which states that the critical aspects of a scene change more slowly than the individual scene measurements. For example, the movement of a horse in successive video frames will lead to a rapid shift in individual pixel values. However, the characteristic describing the horse or the position of its limbs will change more slowly. In its original form, the slowness principle imposes a strong prior that features should remain constant (invariant) across scenes, which leads to sub-optimal performance. For this reason, temporal and spatial coherence assumes that attributes should be easy to predict across scenes. More generally, we assume that different factors could change at different temporal or spatial scales, which is the current explanation of how V1 simple and complex brain cells behave (Hurri and Hyvärinen, 2003), motivating the Slow Feature Analysis algorithm (Wiskott and Sejnowski, 2002) and the pooling operations in CNNs (Zhou and Chellappa, 1988). There are three benefits to this assumption:

    1. consecutive moves in time or space can be contracted to represent minimal moves along manifolds, which makes generalisation easier;
    2. if we additionally assume that factors change at different scales (both time

and space), knowledge of the scale at which the factor varies can facilitate its disentangling from other factors;

3. explanatory factors can be disentangled into sub-components, which could vary together (e.g., representing position or colour as a 3D or RGB rather than a dictionary of all possible combinations).

B.2 The Ideal Data Representation Properties

B.2.1 Expressive

One simple way to measure the expressivity of a representation is to count the number of input regions (also known as configurations of the inputs) that the number of parameters available to the representation can encode. Alternatively, a neural network is essentially computing a linear function once we fix the activation pattern; thus, counting the number of possible activation patterns provides a concrete way of measuring the complexity and expressivity of a representation (Raghu et al., 2017).

This expressivity of a representation also known as representational power or representational capacity.

One of the main challenges for representation learning approaches is that often there is an extremely large number of underlying causal factors. Let us assume an ideal representation $\boldsymbol{h} \in \mathbb{R}^d$ such that encodes all causal factors and a subsequent classification task $\boldsymbol{y} \in \mathbb{R}^m$ such that $m \ll d$. We known that there exists a function f, which maps the underlying cause h_i to an outcome $y_k - f(\boldsymbol{h}) = \boldsymbol{y}$. An unsupervised representation learning approach will not know which h_i are relevant. Therefore, a brute force approach entails that the learner captures and disentangles all relevant factors hi. Unfortunately, it is challenging and often not feasible to capture all or most of the relevant factors that influence an observation. Should we always encode all small background objects in a scene? Or, should we encode all the features that do not change slowly over time in a video frame such as the background? We address this challenge next.

B.2.2 Abstract

As mentioned previously, when building representations, we are often forced to make a choice about which factors to keep (salience) and which factors to ignore (invariance). While building in layers of abstraction helps manage this trade-off by representing more specific factors in the lower layers and combining these into more general categories, it does not address the challenge of deciding which factors to keep.

Salience Currently, there are three strategies to address the "salience challenge". First, we can use supervised learning in conjunction with the unsupervised criterion to include an additional learning signal that will help capture the most salient factors of variation for a particular task. Second, we can use a huge representation with the hope that increased representational capacity will capture most of the relevant factors. Third, an emerging strategy is to modify the definition of salience. Usually, representation learning algorithms such as autoencoders or generative models optimise a fixed-criterion, which to a large degree determines the relevance of different causes. For example, autoencoders trained on images with mean squared error criterion have the implicit assumption that a cause is salient only when it is related to significant changes in the brightness of a considerable number of pixels. This assumption poses substantial problems in situations, which involve operations with small objects (e.g., generating ears or picking ping-pong balls in robotic tasks).

Another definition for salience is that any highly recognisable pattern should be considered salient. Generative adversarial networks (GANs) (Goodfellow et al., 2014b) have emerged as a popular technique to implement this strategy. The idea is to train a generative model to fool a discriminative model. The discriminator is trained to differentiate between samples from the training distribution and sample from the generative model. Lotter, Kreiman, and Cox (2015) demonstrates that mean squared error trained models often fail to generate ears in images of human heads, but GANs can successfully generate this highly recognisable pattern.

Invariance The goal of invariance is to reduce or remove the sensitivity of features to variations that are uninformative to the subsequent task. In fact, Heinze-Deml, Peters, and Meinshausen (2018) propose that there is an inherent link between invariance and causality. We saw that both the hierarchical organisation and the sparsity assumptions have the same aim of introducing invariance to local changes. Abstraction provides a simple framework to improve generalisation through invariant features. First, we build "low-level features that account for the observed variation. Second, combinations of low-level features are aggregated (e.g., pooled) to build more invariant higher-level features. Invariance inadvertently makes the target function highly non-linear in the input space because it ignores most local changes in the input. The high non-linearity facilitates the capture of more general categories that described more varied phenomenon (e.g., a plane can be on the ground, in the air, at a hanger, or could be a toy). The corresponding manifolds in input space of such general categories are larger and more wrinkled (more ups and downs) than the learned manifold, which makes generalisation easier because we can better model the observed variation (Bengio, Courville, and Vincent, 2013; Bengio et al., 2013).

B.2.3 Disentangling: Separate Directions

The explanatory factors of real-world inputs tend to change independently of each other, and only very few at a time. Consequently, the resulting features should be **sparse or independent** from each other, such that each feature or direction in representation space corresponds to a different explanatory cause and is insensitive to minor variations. We would expect that these factors group together to represent various forms of variation combinations. Such a construction implies that the distributions over the latent variables within the representation become factorised. That is, the latent variables contain multiple independencies, which makes them **easier to model**, especially for density estimation tasks. A factorised distribution over the latent variables generally results in more efficient computations and representations that are more comprehensible to humans. For example, Zhou et al. (2014) demonstrates that the hidden units within the top layers of deep convolutional neural networks (DCNNs) trained on ImageNet and Places datasets have interpretable features. That is, the features represent concepts which could be assigned naturally by a human. Further, Radford, Metz, and Chintala (2016) demonstrates that generative models can learn separate directions in representation space, which encode different underlying factors of variation. The factors can be subjected to mathematical operations to produce a new combination of semantically valid factors.

For instance, Figure B.1 illustrates that we can subtract from a vector representing a man with glasses, a vector that represents a man without glasses. Then we can add another vector representing a woman without glasses. The surprising result is a woman with glasses. Similar results can be seen in natural language processing, where different directions encode for gender and singularity vs plurality such that we can perform computations of the sort *king - man + woman = queen* and *queen + plural = queens* (Mikolov et al., 2013b; Mikolov et al., 2013a; Mikolov, Yih, and Zweig, 2013b).

Texture-bias In practise, interpretable features do not always emerge and in fact DCNNs have an unnatural bias towards textures rather than shapes (Geirhos et al., 2019). For example, an image can be constructed such that the shape of the image is one of a cat; however, the filling of the shape (colour, texture) is that of an elephant skin (see Figure B.2). Subject experiments demonstrate that the variety of humans classify the image based on the shape characteristic in stark contrast to DNNs. It is fascinating that both interpretable features, capturing the underlying variations, and texture-bias occur naturally without including particular regularisation terms. These findings suggest that interpretable features and texture-bias are somehow relevant to the optimisation task.

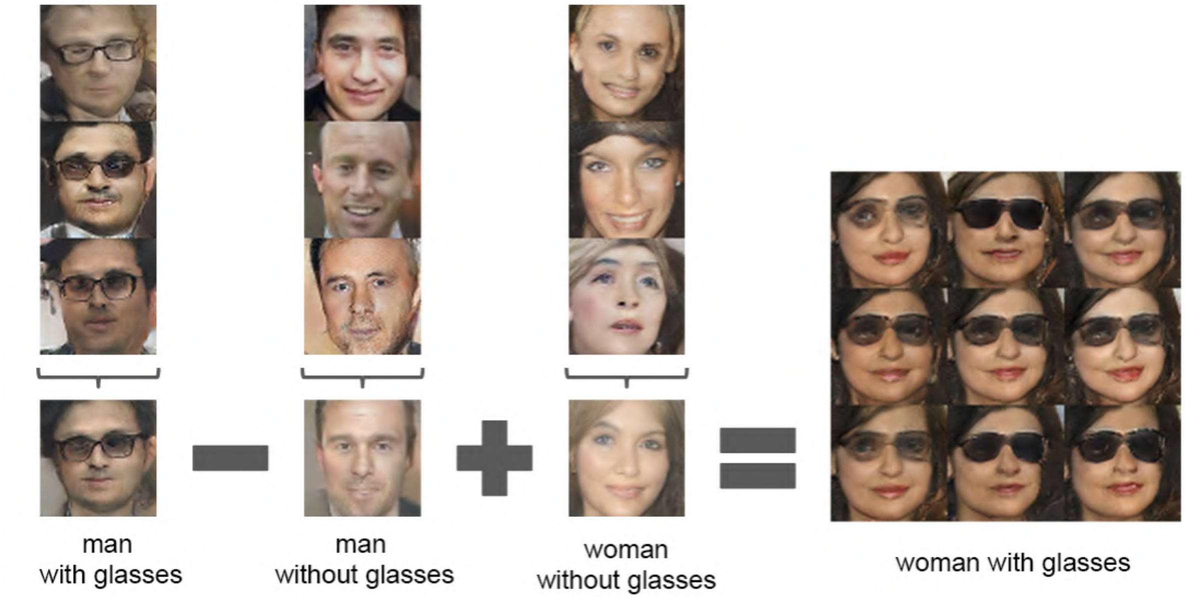


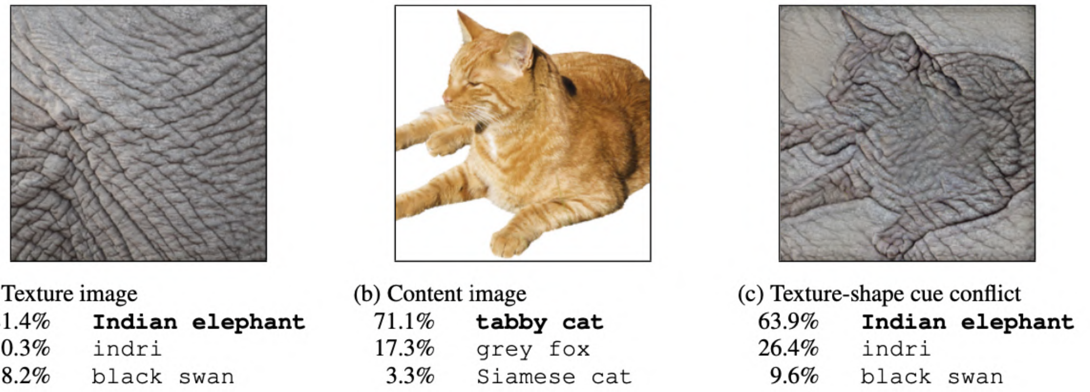
Figure B.1: The concepts of gender and wearing glasses can be manipulating separately, suggesting that the generative model has learned a distributed representation that disentangles the two concepts. Image reproduced from (Goodfellow, Bengio, and Courville, 2016b).

B.2.4 Easy to model

As we discussed above, the disentangling property requires that representations encode sparse and independent factors. A central theme of this thesis is the ability to interpret the representations learned by DNNs. When the distributions encoded within a representation do not involve all factors (i.e., they are sparse) and each factor can be observed without affecting the other factors, it is much easier for a human to comprehend the captured information (Miller, 1956). Next, we will look at compactness and robustness, which are less theoretical requirements, more concerned with the ability to use the learnt representation.

B.2.5 Compact

Two key considerations in any software are the space and time complexity of the algorithm. The naive way to build a representation of the world is just to have a table that encodes every possible value. According to the curse of dimensionality, this approach is bound to fail since it becomes exponentially more challenging to encounter every possible configuration as the dimensionality of the data increases. A much more practical approach, both computationally (fewer computations and less storage) and statistically (better generalisation), is to represent only the salient variations. Sometimes for computational reasons, we might even need to decrease the dimensionality of our data. In these cases, it is paramount that we prune out the directions with the least amount of variation. It is worth mentioning that there are two different schemes to measure the variation. The *local*



(a) Texture image
81.4% **Indian elephant**
10.3% indri
8.2% black swan

(b) Content image
71.1% **tabby cat**
17.3% grey fox
3.3% Siamese cat

(c) Texture-shape cue conflict
63.9% **Indian elephant**
26.4% indri
9.6% black swan

Figure B.2: Classification accuracy of ResNet-50 on a) texture centric image (elephant skin); b) normal image (both cat shape and texture); c) image with texture-shape conflict. Observe that when the DNN is force to chose it relies on texture rather than shape. Image reproduced from (Geirhos et al., 2019).

strategy measures the variation in local directions on the manifold around each sample, whereas the *global* strategy measure the variation across the entire dataset (e.g., PCA).

B.2.6 Robust

When we deploy machine learning applications to real-world problems, we want to make sure that our algorithms operate reliably in a wide variety of circumstances. This concern is particularly important in safety and security-focused applications. When we talk about robustness of a representation, we usually refer to the ability of a representation to resist changes in the data distribution, so that it generalises to worst-case or unseen inputs. To achieve generalisation, we want to make sure that a representation is robust against (not vulnerable to) two main types of changes:

1. corruptions or infinitesimally small perturbations to the inputs (e.g., adversarial examples, noise or missing inputs);
2. input, output, or conditional distribution shift[7] (out-of-distribution generalisation):
   (a) changes to the input distribution (e.g., domain adaptation and covariate shift);
   (b) changes to the output distribution (e.g., label shift, transfer learning and multi-task learning);
   (c) changes to the causal mechanism (e.g., concepts drift).

[7] See Appendix A.

Coarse Coding A curious paradox of neural representations is that we can learn a target function more accurately when a set of neurons has a coarse-grained rather than fine-grained response function of the input (Plate, 2006). That is, the accuracy of the representation will increase when the precision of the individual neurons decreases. Decreasing the precision of a neuron implies that we increase its *"receptive field"*, which is the range of inputs it responds to. To illustrate, let's imagine a continuous function describing a particle's position in 2-D or higher-dimensional space. If we represent this space with a set of neurons such that each neuron responds to a circular region with radius r (receptive field) within a k-dimensional input space, then the inaccuracy of the representation is proportional to $\frac{1}{r^{k-1}}$ (Hinton, McClelland, and Rumelhart, 1986). Hence, in a 3-dimensional space, doubling the radius to make the neuron more coarse, yields a 4 times better representation.

The reason for this strange phenomenon is related to information theory. A neuron with a small radius activates for a tiny fraction of the total inputs, resembling a nearly deterministic probability distribution over the data, thus carrying a negligible amount of information. On the other hand, a neuron with a larger radius activates for a greater fraction of inputs, which means it has a higher uncertainty over the input and a much higher amount of information. Since a representation of an entity is formed by intersecting all active neurons and coarse coding leads to an increased number of active neurons, coarse coding improves the accuracy of the representation. Hence, we can conclude that the resolution of a representation depends on the density and the overlap between the unit receptive fields (Sullins, 1985).

Notice that the receptive field of units within a local representation is constrained to specific concepts. In contrast, the field of distributed units is the set of all patterns a neuron participates in (Rosenfeld and Touretzky, 1987). Intuitively, we can think of coarse coding as implicitly encoding the slowness prior since we need to change the values of the neuron stimulus drastically to produce a change in activation. For example, a representation encoding the size of an animal can respond to discretised values of small, medium or large sizes rather than to the exact height, width and length of an animal.

This concludes the discussion on representation characteristics that describe the form of representations that meets the ideal data representation requirements the most – partially-distributed representations. Let us now look at the source of partially-distributed representations' representational power.

Conclusions

An ideal representation is *expressive* and captures all the salient underlying causes of the observed data. It uses multiple levels of abstraction to balance the trade-off between sensitivity to informative and invariance towards non-informative directions (*abstract*). The representation *disentangles* the salient causes in a way that a separate feature or direction in feature space represents each of the causes in a maximally *easy to model* and *compact* way so that it is easier to interpret the representation and perform subsequent tasks. Finally, we require the representation to be *robust* to minor corruptions or perturbations and to have powerful generalisation properties to unseen or differently distributed samples.

B.3 Limitations

Here we expand the three shortcomings of distributed representations:

- **interpretability**: the ability to be understandable to a human (Doshi-Velez and Kim, 2017).
- **robustness**: the ability to resist minor corruptions and generalise to unseen data distributions.
- **generalisation**: in particular the **binding problem**, which is the inability to maintain associations between multiple concepts (Plate, 2006).

B.3.1 Interpretability

Distributed representations have not been designed with interpretability in mind. On the contrary, they were designed to make subsequent processing tasks easier, more efficient and more robust to noise. Plate (2006) proposed two ways to address this challenge: (1) elicit the concepts that are represented through the superposition of activation patterns, provided the concepts of the basic patterns are known; and (2) elicit an intuitive space of learned features that describe a concept in a human-understandable way. The first approach maps probability distribution over the activation patterns and probability distributions over the concepts (Zemel, Dayan, and Pouget, 1998), while the second approach leverages clustering and dimensionality reduction techniques to cast the hidden space into a more intuitively understandable format (Elman, 1990; Elman, 1991). In Chapter 5, we propose two frameworks that take a step forward in both of these directions. Notice that there is a subtle difference between interpreting a complete DNN model and its internal representations. The latter can contribute to the former. A comprehensive review of both model and representation interpretation approaches can be found in Chapter 3.

B.3.2 Robustness

In Section B.2.6 we described the two desiderata for building robust representation: (a) robust against infinitesimally small perturbations or corruptions; and (b) robust against distribution shifts. We can think of the robustness property as a type of invariance. Essentially, we want the representation to remain the same for various non-informative changes that affect the input distributions. In that respect, we can cast the two types of robustness as explicitly reducing the difference between representations of datapoints in the vicinity of each other – *intra*-domain differences; and reducing the differences between representations from different domains or distributions – *inter*-domain differences.

One of the main challenges for building robust representations seems to be the long-standing framework for learning Empirical Risk Minimisation (ERM). ERM optimises and evaluates the performance of a learning algorithm on the empirical distribution due to lack of knowledge of the true distribution. ERM has been extremely successful in finding classifier with low population risk, that is with small error on the corresponding task. However, distribution shifts (Geirhos et al., 2019; Hendrycks and Dietterich, 2019) violate the independent identically distributed (i.i.d.) assumption breaking the foundation of existing generalisation theory (Bartlett and Mendelson, 2002; McAllester, 1999). Empirically, this means that ERM learning results in models with non-robust representations (Szegedy et al., 2014; Biggio et al., 2013; Arjovsky et al., 2019).

We can think of intra-domain robustness as violating the i.i.d. assumption at the micro level. The sampling granularity of the training data is different to the sampling granularity of the perturbed data. Fortunately, in the intra-domain robustness case, we can rely on the local constancy prior to generalise since perturbed points in the vicinity of a training point should have the same output or share characteristics.

On the other hand, inter-domain robustness explicitly breaks the i.i.d. assumption on a macro level because it requires the representation to generalise to unseen or different distributions. The robustness goal to ensure o.o.d. generalisation aims to relax the pivotal assumption of independently identically distributed to independently non-identically distributed (i.n.d).

Here we list the main directions of modifying ERM to increase the robustness of representations:

1. reducing the intra-domain differences of representations (c.f. interpolation)

   (a) **robust training:** explicitly enforcing resistance to infinitesimally small perturbations or corruptions (Madry et al., 2018; Tsipras et al., 2019; Wang, He, and Xing, 2019; Wong and Kolter, 2018; Sinha, Namkoong, and Duchi, 2018; Xiao et al., 2019)

2. reducing the inter-domain differences of representations (c.f. extrapolation)

   (a) **transfer learning**: captures anticausal factors that disentangle the representation of $p(x)$ and $p(y|x)$, which would enable inter-domain re-usability of representations

(b) **domain generalisation**: looks for stable representations of anticausal factors that are optimal across all (including unseen) domains (Arjovsky et al., 2019; Arpit, Xiong, and Socher, 2019; Muandet, Balduzzi, and Schölkopf, 2013).

The most important aspect of the model that determines robustness is the distance of the closes point to the decision boundary (Madry et al., 2018; Tsipras et al., 2019). Points further away will resist a wider array of perturbations before representing a crossing across the decision boundary. See Figure B.3 for an example.

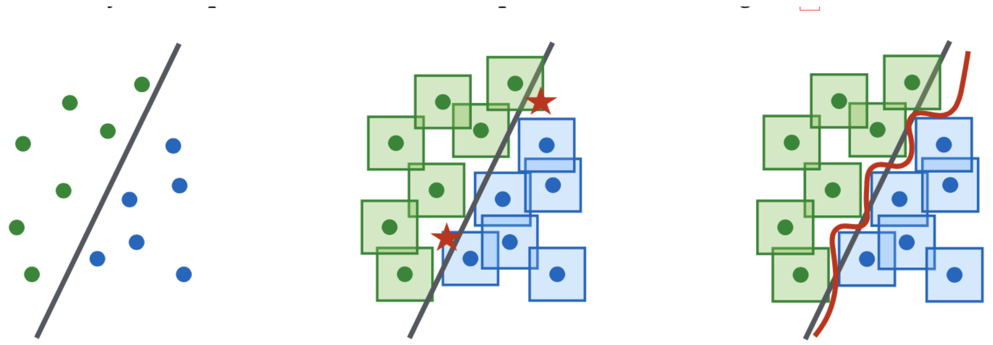


Figure B.3: A hypothetical illustration of the difference between a robust (right) and non-robust decisions (middle)boundaries for a set of linearly separable points (left). Middle: Observe that the simple decision boundary cannot separate the L^p-bounded perturbations around datapoints (here squares). This leads to adversarial example (red stars). Right: A more complex decision boundary is required to separate the point neighbourhoods; hence, a model with a higher capacity is more likely to attain better robustness. Image reproduced from (Madry et al., 2018).

Small curvature in the vicinity of datapoints Interestingly Moosavi-Dezfooli et al. (2019) confirms the hypothesis that robust training induces increased distance between datapoints and the decision boundary. The result is a locally-linear boundary in the vicinity of the datapoints. Additionally, Moosavi-Dezfooli et al. (2019) challenge the hypothesis of the highly non-linear decision boundary. The eigenvalue spectral analysis of the Hessian[8] of the loss function with respect to the inputs[9] suggests that the decision boundary becomes significantly flatter in all directions. The implication of this result is a strong relationship between high robustness and *small local curvature*. We further confirm this phenomenon in Appendix C.2.3. Notice that this does not contradict, but supports Madry's conjecture (Madry et al., 2018; Tsipras et al., 2019), which says nothing about the global shape of the decision boundary. The boundary can still be a piece-wise

[8] Note that the maximum / minimum eigenvalues determine the maximum / minimum second derivatives, thereby determining the maximum curvature / flatness respectively.

[9] Note that the authors compute an approximation of the Hessian, which measures large variations in the gradient in the neighbourhood of datapoints. The approximation makes it difficult to draw any inferences about the global curvature of the decision boundary. Further, notice that the authors measure the Hessian of the loss landscape and use it as a proxy for the curvature of the decision boundary.

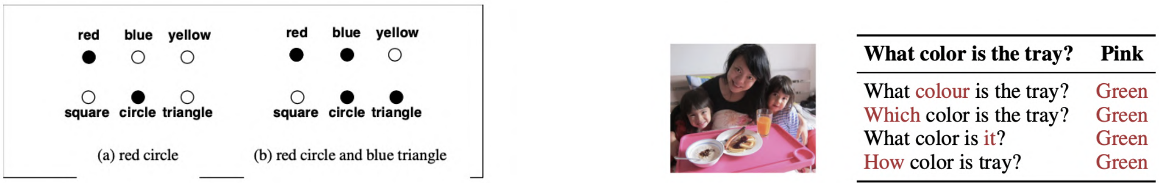
linear function: highly-rugged (non-smooth) and locally linear, which implies small local curvature.

Flat decision boundaries, separated by broader high entropy regions Under classical regularisation techniques (e.g., weight decay, dropout, batch-norm) the decision boundary is often sharp (rather than smooth) and close to datapoints. Hence, we end up with configurations that project the datapoints into congested regions in hidden space. Narrow representation space regions cause highly confident, but not necessarily accurate, prediction because there is not enough room to encode uncertainty. In other words, the representation is sharply jumping from one region of low entropy (high confidence) to another (Verma et al., 2019).

On the contrary, flatter decision boundaries, separated by broader high entropy regions, in both input and hidden space, give rise to two phenomena. First, the change in any one single direction must be much more significant to change the prediction into a highly confident region. Second, the representation is much sparser, so a change across many more directions and mostly highly contributing directions is necessary to cause a significant variation in the output (Verma et al., 2019). These results confirm the hypothesis that smoothness and margin (distance between the closest datapoint and the decision boundary is paramount for generalisation to noisy environments (robustness) (Bartlett and Shawe-Taylor, 1999; Lee, Bartlett, and Williamson, 1995).

Drawbacks Although adversarial robustness is a useful property, it comes at a cost. The most obvious drawback of robust training is **increased training time** since we are computing new worst-case perturbations at each update step. Additional statistical costs accompany these computational costs. Schmidt et al. (2018) and Alayrac et al. (2019) demonstrate that robust training requires **significantly more data** and that the increased data requirement is irrespective of the training algorithm or the model family. Alayrac et al. (2019) show that unlabelled data can be leveraged effectively to increase the robustness, at least partially mitigating the cost of expensive labelling.

Additionally, there could be an inherent **trade-off** between **robustness and accuracy** if no assumptions about the data distributions are made (Tsipras et al., 2019; Zhang et al., 2019). The features learned by optimal standard and optimal robust classifier can be substantially different, which suggests the need for specialised techniques tailored for finding robust representations. Interestingly, robust training can be beneficial to the classification performance in the regime of limited training data (Tsipras et al., 2019).



What color is the tray?	Pink
What colour is the tray?	Green
Which color is the tray?	Green
What color is it?	Green
How color is tray?	Green

(i) The binding problem: representing multiple objects using independent representations over their features loses the information about the association between the feature and the object. Image reproduced from (Plate, 2006).

(ii) Paraphrasing a question complete changes the prediction of a Question Answering system. Notice that the first 3 corrections are very natural, yet still completely confuse the system. Image reproduced from (Ebrahimi et al., 2018).

Figure B.4: Examples of the binding problem.

B.3.3 Generalisation: The Binding Problem

One of the main remaining representational challenges is the *binding problem* (Plate, 2006). The binding problem describes the difficulty of representing the associations between multiple variables. The challenge is that information is inevitably lost when the features of multiple objects are encoded within the same distributed representation. Let's imagine the task of representing different figures, with two features (colour and shape), represented independently as shown in Figure B.4.i. In the case of a single object, Figure B.4.i(a), the association between the colour and shape is preserved, whereas, in the case of multiple objects, Figure B.4.i(b), the association is not maintained. The representation for a red circle and a blue triangle is identical as the representation for blue circle and red triangle; hence, the association (binding) between different features is lost without an additional data structures that could describe the association explicitly.

This challenge emerges in both NLP and vision tasks. For example, let us consider the following sentence: "John watched Sam cook the eggs". There are two types of binding problems. The first one is the difficulty of associating the correct entities within the different representations – representing that John is watching and Sam is cooking and not vice versa. The second one is the subject-object dependence within the relationship – "Sam cooks the eggs"; "John watched Sam". Observe the recursive nature of the latter association, which demonstrates the hierarchy assumption in action. We first need to represent the relation "Sam cooks the eggs" and then we want to bind that as the object of the association "John watched". The recursive nature is one of the reasons for the hierarchical structure. The binding problem, together with the linearity assumption, could be one possible explanation behind the recently discovered fragility of Image Question Answering System (Ebrahimi et al., 2018). As illustrated in Figure B.4.ii, a natural rephrasing of a question regarding an image leads to entirely different answers.

The challenges emerge in vision problems as well. A nonsensical reordering of semantic components such as eyes, mouth, and nose, does not lead to significant changes in the model's predictions. One possible solution could be Capsule Networks (Sabour, Frosst, and Hinton, 2017), which contain additional data structures to learn the associations.

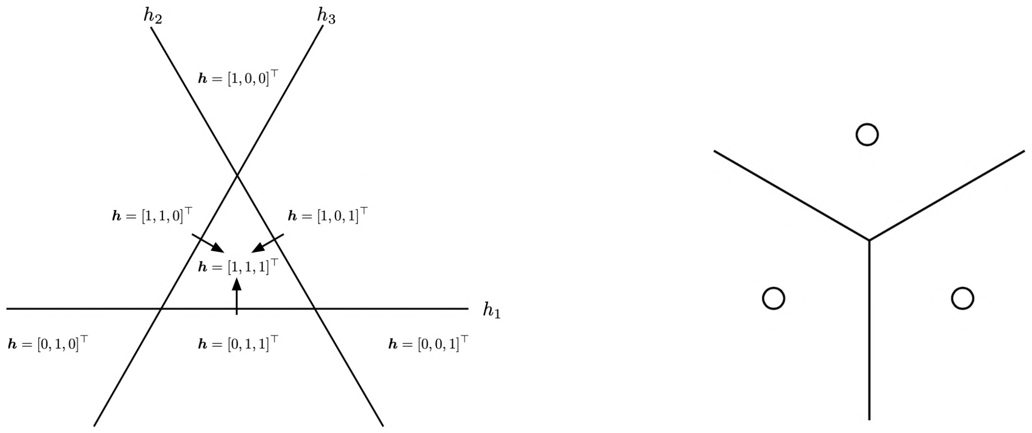
B.4 The Case for Distributed Representations

Disentangle Independent, Invariant, and Linearly-separable factors Here we demonstrate that distributed representation have been designed to incorporate multiple of the assumptions described in Section 2.2 and the heuristics described in Section 2.4 to fulfil the requirements set out in Section 2.3. In particular, distributed representations are designed to: (1) disentangle independent, invariant and linearly separable factors (disentangling); (2) form a natural clustering in a rich similarity space of reusable factors connected in a hierarchical structure of simple relationships (abstract). These two design considerations give distributed representations (3) exponential gains in representation power over non-distributed representations (expressive & compact).

The disentangling characteristics of distributed representations result in separate control over the underlying factors of variation. Better disentangling between the factors leads to features with strong mutual information with one or very few of the underlying factors of variation and high **invariance** to all other factors or non-informative variations. That means that each of the features would become specialised and highly predictive of its corresponding factor or small set of factors independently of other variations.

Expressivity To illustrate the power of distributed representations, let us compare them to a type of non-distributed representations – **symbolic representations**. Symbolic representations associate the input with a single element or category of the representation. For example, the **one-hot encoding** representation is a binary basis vector with n bits, which means that the bits are all mutually exclusive. Only one vector element can be active at a time (e.g., a vocabulary of n words, in which a basis vector $\boldsymbol{e}^{(i)}$ represents each word i).

Non-distributed Representations K-nearest neighbours, decision trees, kernel machines with local kernels, clustering methods, such as k-means or Gaussian mixture models, all rely on non-distributed representations. The challenge with these approaches is that although multiple parameters or template examples produce the output, these parameters *cannot be controlled separately.* That is, although changing one template or support vector modifies the carved out region in input space, it does not define a new region. The parameters cannot be combined in new ways to shatter the input space additionally. For example, a point cannot be assigned to two clusters simultaneously by creating a new region between two clusters (see Figure B.5b).


h_2
h_3
$\boldsymbol{h} = [1, 0, 0]^\top$
$\boldsymbol{h} = [1, 1, 0]^\top$
$\boldsymbol{h} = [1, 0, 1]^\top$
$\boldsymbol{h} = [1, 1, 1]^\top$
h_1
$\boldsymbol{h} = [0, 1, 0]^\top$
$\boldsymbol{h} = [0, 1, 1]^\top$
$\boldsymbol{h} = [0, 0, 1]^\top$

(a) Distributed representation with binary features splitting the input space. Each feature splits the space into two half-planes with different configuration at every intersection of the half-planes. Notice that not all configurations are possible such as $\boldsymbol{h} = \mathbf{0}$. Nevertheless, the number of unique regions in $\mathbb{R}^d$ space that n binary features can distinguish is equal to $\sum_{i=0}^{d} \binom{n}{j} = O(n^d)$ (Pascanu, Montúfar, and Bengio, 2014; Zaslavsky, 1975). Hence, the growth of distinguishable regions is exponential in the input dimension, but polynomial in the representation size.

(b) Non-distributed representation such as k-nearest neighbours splitting the input space. Each region is defined by a different set of parameters in this case template examples (represented by circles). Each parameter defines the boundaries of the region (represented by lines) and the output of the algorithm. Therefore, we need n examples to distinguish at most n regions.

Figure B.5: Comparison between the ability of distributed and non-distributed representations to break up the input space. Observe that distributed representations can separate exponentially large number of regions. Images reproduced from Goodfellow, Bengio, and Courville, 2016a.

Similarly, decision-trees associate a given input with a one-hot representation over the leaves because they partition the input space in sub-regions, where each region has separate parameters. The path of a leaf's ancestors to the root defines the parameters of each leaf and a decision tree with n leaves requires 2^{n-1} parameters (Bengio, 2009; Bengio, Delalleau, and Simard, 2010).

This discussion illustrates a main point: *for all non-distributed representations, the* ***number*** *of different* ***regions*** *that the representation can partition* ***scales linearly*** *with the number of* ***parameters*** *or the size of the representations.* Hence, good generalisation requires the **same number of examples** as the number of distinct **input space regions**.

Non-distributed Generalisation Not only do decisions trees need the same number of examples as different variations in the target function, but also they capture only the variation in the training data, without any sophisticated mechanism to generalise to unseen variations (Bengio, Delalleau, and Simard, 2010). For some of these "non-distributed" algorithms, the output is not constant for each region, but interpolates between neighbouring regions. Still, they generalise only locally due to the smoothness prior. Local constancy or interpolation within regions and between regions fails to describe functions with many variations, even if these functions have short functional descriptions (i.e., low Kologmorov complexity[10]) (Bengio, Delalleau, and Roux, 2006)[11].

In other cases, more fine-grained control is possible. For example, in mixture models, each mixture component can be controlled by different parameters giving rise to non-discrete membership. We will now explore that the representational capacity difference is still exponential since the parameters between mixture components cannot be shared (Bengio, 2009; Bengio and Delalleau, 2011).

Distributed Generalisation On the contrary, distributed representations can control each parameter separately and combine parameters to achieve multi-clustering properties. A binary distributed representation with n features can have 2^n configurations and carve out 2^n number of regions in input space because each combination of directions (features) can correspond to a different configuration value (Pascanu, Montúfar, and Bengio, 2014). To illustrate, consider the examples in Figure B.5. Figure B.5a depicts how a distributed representation can split the input space into exponentially more regions with the same representation size as a non-distributed representation (Figure B.5b).

Generally, the **argument in favour of distributed representations** is that *a distributed learning algorithm can* ***represent*** *$O(r)$* ***regions*** *with $O(\log r)$* ***parameters*** *compared to $O(r)$ parameters in the non-distributed setting.* Therefore, the distributed algorithm has **fewer parameters to learn** and thus requires much **less training data** to **generalise** well (Goodfellow, Bengio, and Courville, 2016a; Bengio, 2009).

B.4.1 Linearly Separable

Although distributed representations can encode an exponential number of regions, the capacity of deep learning models is constrained because we cannot use the entire code space. This observation comes from an interesting result from complexity theory. The VC dimension of binary output neural networks with linear threshold activation functions is only $O(w \log w)$, where w is the number of hidden units in the layer (Sontag, 1998)[12]. Consequently, we can interpret any two layers of the network as a linear predictor on top of a distributed representation. The combination of distributed representations with linear predictor induces a prior belief that learned concepts should be linearly separable as a function of the features (i.e., bias against XOR logic). For example, the model will be

[10] The length of the shortest computer program that can describe the function.

[11] A more detailed discussion on the local nature of these and other algorithms can be found in Section 3 of Bengio (2009).

[12] Similar results can be derived for networks with binary outputs and piece-wise linear activation functions (Bartlett and Maass, 2003)

biased towards learning concepts such as all pink objects or all elephants rather than pink elephants and green giraffes. Recent results (Pérez, Camargo, and Louis, 2019) support this hypothesis, suggesting that DNNs are inherently exponentially biased towards simple functions.

B.4.2 Natural Clustering in a Rich Similarity Space

When a distinct set of parameters can be controlled separately, as in distributed representations, different concepts can share the **attributes** of the representation. For example, there are many similarities between cars and trucks, which common features could describe: "number_of_wheels", "has_door", "has_windshield". These features generalise across concepts, which means information about one concept can supplement information about another and thus decrease the amount of data necessary to learn both concepts.

Rich similarity space Due to the fact that semantically similar objects **share reusable features** through similar activation patterns, distributed representations induce a rich similarity space (Elman, 1990). This is one of the most powerful properties of distributed representations because it allows for **complex operations** in **representation space**. The simplest possible operation is interpolating between datapoints in hidden space. It contributes to the generalisation power of distributed representations because unknown points can be easily labelled. More complex operation are also possible, such as non-exact matching for information retrieval tasks or vector addition and subtraction in word embeddings (Mikolov et al., 2013a).

Word Embeddings Word embeddings are the most notable illustration of the rich similarity space. A one-hot encoding of a word does not say anything about the relationship with other words. In fact, in a basis-vector space, any word is at an equal distance to all other words. On the other hand, neural language models, based on distributed representations, learn representations that share attributes between words[13], which frequently appear in the similar contexts. A sharing of attributes often gives rise to a **natural clustering**, where semantically similar words tend to be neighbours in the representation space (Mikolov, Yih, and Zweig, 2013a). This clustering is a particularly powerful way to **counteract the curse of dimensionality**. A large number of shared factors leads to the transfer of information from one setting to another (e.g., from one training sentence to an exponential number of semantically similar sentences) (Bengio, Ducharme, and Vincent, 2000).

[13] Notice that the rich similarity space property is closely related to one of shared reusable features.

B.4.3 Hierarchical compositional structure

Distributed representations can be stacked together to form deep distributed representations (deep learning). As we noted previously, a fundamental assumption of deep learning is that the algorithm should learn a hierarchical representation such that high-level concepts are defined using simpler ones.

Deep distributed representations assume a hierarchy is more likely to disentangle independent high-level factors. The abstract concept these factors represent are related to the input in complex extremely non-linear ways, but simpler (lower degree polynomial) ways. Therefore, we make the general assumption that the function describing these factors is composed of multiple simpler non-linear functions of reusable low-level features. The simpler functions recursively describe the different ways, in which the high-level concepts relate to the input.

The hierarchy assumption has three main benefits: (1) contributes to disentangling of factors of variation; (2) induces a prior of building invariant features; and (3) leads to exponential gains in representation power because it promotes the reuse of features. Together these benefits form one of the main motivations behind distributed representations since a deep hierarchy requires powerful intermediate representations of concepts to perform a series of processing stages.

Disentanglement First, we can think of feature composition as the generative equivalent of feature representation's goal of disentangling factors of variations. In that sense, the hierarchical organisation is the inverse function of factor disentanglement. Empirical results support the hypothesis that deep representations help to disentangle the factors of variation. For example, Bengio and Delalleau (2011) demonstrate that empirically the marginal distributions of the deeper layers representations lead to better and more interpretable separation of inputs in deeper layers. More concretely, the marginal distributions of the hidden units deeper layers become smoother, more spread out, and more unimodal. Such distributions lead to the unfolding and expanding of high-dimensional manifolds representation compared to their corresponding manifolds in input space. Smoother and more spread-out manifolds make interpolation between high-probability samples easier, thus **improving** the **generalisation**. Intuitively, this means that DNNs are capable of disentangling highly curved input manifolds into flat hidden space manifolds (Poole et al., 2016). At the same time, the unimodal property **separates** the **factors** on different manifolds.

Invariance Second, empirical results suggest that deeper layers of representations learn features that are more invariant to the less informative variations in the observed data (Yosinski et al., 2015; Bengio, 2009; Bengio and Delalleau, 2011). Convolutional deep belief networks learn features that are significantly invariant to input transformations. Deeper stacks of autoencoders learn moderately invariant features as the depth increases (Goodfellow et al., 2009).

Invariance is increasingly becoming a more important characteristic. Some researchers (Heinze-Deml, Peters, and Meinshausen, 2018; Arjovsky et al., 2019; Ahuja et al., 2020) see invariance as the path to causality because strong invariance to non-informative variations could entail high specificity to causal factors.

Expressivity Third, the hierarchy assumption leads to exponential gains in representation power because a neural network of depth l can represent exponentially more regions than a network of depth $l-1$. Theoretically, the number of regions that a piece-wise linear network (e.g., DNN with a ReLU activation function) is bounded by:

$$O\bigg(\binom{n}{d}^{d(l-1)} n^d\bigg),$$

where the network's parameters are d inputs, l depth, n hidden units per layer. Empirically, larger depth does seem to be correlated with better performance Montufar et al. (2014), Pascanu, Montufar, and Bengio (2013), and Goodfellow et al. (2014a)[14].

The number of ways we can reuse a feature grows exponentially with depth. Therefore, the power of building a hierarchy over reusable features through the composition of non-linearities can give an exponential increase in representation capacity in addition to the exponential growth resulting from representing these features in a distributed fashion.

Not all layers are created equal If the representational power grows exponentially with depth, then small changes to parameters in the lower layers have larger effects on the output than changes in higher layers. For this reason, optimising the weights in lower layers is especially important, although depth increases the representation power (Raghu et al., 2017). The importance of lower layers has substantial implications for interpretability. In Section 3.5.2 we discuss that lower layers have been completely overlooked, although recent results (Adebayo et al., 2018) seem to suggest that they play a crucial role in the fidelity of explanations.

[14] These results generalise to representing joint probability distributions with more variables than hidden units. For example, shallow binary neural networks cannot differentiate between r-independent distributions and r-independent uniform distributions (i.e., independent random noise). Order r-polynomials over the real numbers cannot capture r-independent distributions (Braverman, 2011; Bengio and Delalleau, 2011).

APPENDIX C

Model and Data Properties that Affect Interpretability

C.1 Relationship between Data & Model Properties and Model Characteristics

We define a set of dataset and model properties that determine the representational capacity and the dataset complexity (i.e., the complexity of the classification problem), and we look into a subset of their relationships with the model characteristics of accuracy, interpretability, and robustness. The following properties should be considered when conducting model explanation:

1. Model and representation properties:
   (a) **curvature** of the decision boundary:
      i. **local curvature** around a training point or for a particular feature;
      ii. **global curvature** of hidden representation manifolds and the regions of low probability separating these manifolds, which describe the relationships between features;

   (b) **sparsity** of hidden representations (proxy for dimensionality of global manifolds, thus, global curvature);

   (c) **invariance**, or robustness, of hidden representations to noise and unstable signals (proxy for curvature and generalisation due to causal relationships).

2. Dataset complexity:
   (a) **size**:

   i. number of datapoints
   ii. number of features

(b) **intra-feature properties** (properties related to the marginal distribution of a feature):
   i. location (e.g., mean, mode, median)
   ii. variability (e.g., range, standard deviation, variance)
   iii. lack of symmetry (skewness)
   iv. heavy-tailed or light-tailed (kurtosis)

(c) **inter-feature properties** (properties related to the join distribution of features):
   i. **confounding factors**
   ii. **individual feature contributions**

Figure C.1 summarises the implicit relationships between these proprieties and the model characteristics. Here, we present the theoretical reasons for the importance of these properties and relationships, while Section C.2 gives empirical results supporting our claims. Let us now examine each of these properties in more detail.

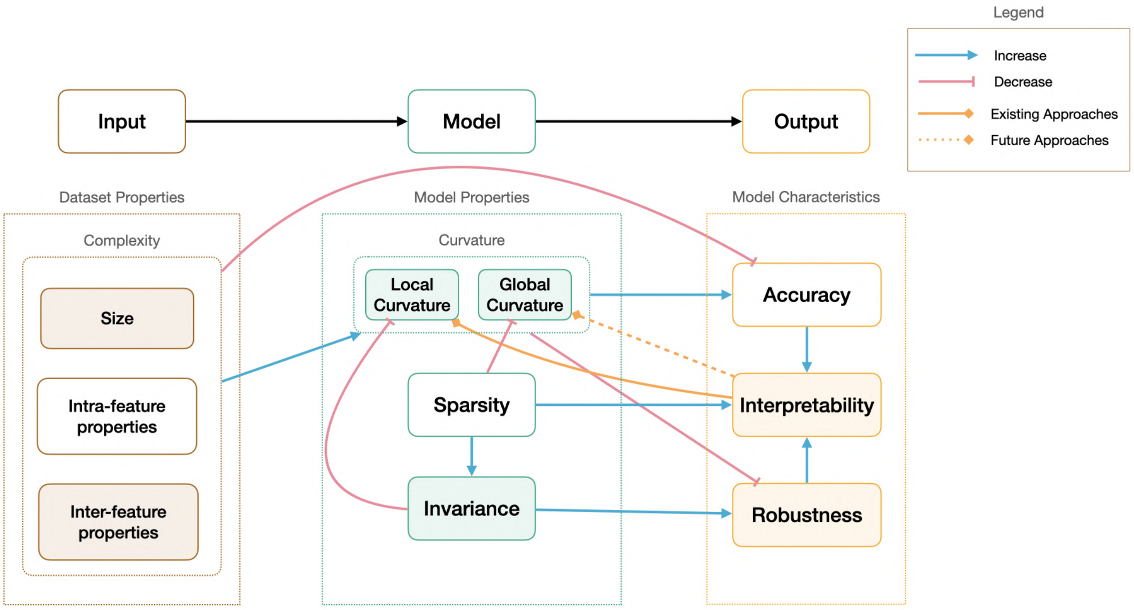


Figure C.1: The interplay between properties influencing interpretability and model characteristics. The shaded boxes indicate subjects that we investigate empirically in this appendix.

C.1.1 Dataset Properties

The complexity of the dataset describes the difficulty of the task at hand. More difficult tasks can be solved using models with higher capacity, which makes the models more difficult to interpret. This difficulty is the result not only of more sophisticated behaviours to be explained, but also of the fact that current ways to explain models overlook important properties of the dataset. The most common descriptors of dataset complexity are related to the size and feature-based properties of the data (Lorena et al., 2019). The size of the dataset is usually determined by the ratio between the number of datapoints and features. On the other hand, we propose to divide feature-based properties on the basis of whether they are concerned with a single feature, **intra-feature properties**, or multiple features, **inter-feature properties**. The inter-feature properties look at the relationship of multiple feature with the output. We propose that two such properties are confounding factors and the ranking of the individual feature contributions to the output.

The effects of dataset complexity in terms of size and intra-feature have been widely studied in statistics literature (Heckert et al., 2002; Ho and Basu, 2002). Here we focus on the inter-feature properties. To the best of our knowledge we are the first to explicitly study the effect of inter-feature properties on interpretability. Molnar et al. (2020) have recently investigated similar ideas in parallel with us; however, they study global functional description explanation, such as partial dependence plots, accumulated local effects, and individual conditional expectations[1], whereas we focus on feature importance explanations.

We make two important observations that highlight the significance of explicitly incorporating inter-feature information when interpreting models. First, Section C.2.1 illustrates that the i.i.d assumption[2] does not hold even for commonly used dataset, leading to misleading conclusions that the algorithm does not depend on a particular feature, when the information about that feature is easily inferable from confounding factors. Second, Section C.2.3.1 investigates the effect of dataset size on model robustness. portrays that small and large dataset can have radically different effects to the training accuracy, robustness, and interpretability because of their effect on the decision boundary.

C.1.2 Model Properties

Figures C.1 & C.2 depict that the curvature of the decision boundary is tightly linked with the sparsity and invariance of hidden representations. At the same time, Figure C.2 portrays how these three properties influence each other to yield specific model characteristics across the spectra of possible values. Manipulating any of these properties inadvertently affects the others, so they jointly determine the model characteristics. For example, increasing

[1] See Section 3.4.5.3 for more details.

[2] See Appendix A.

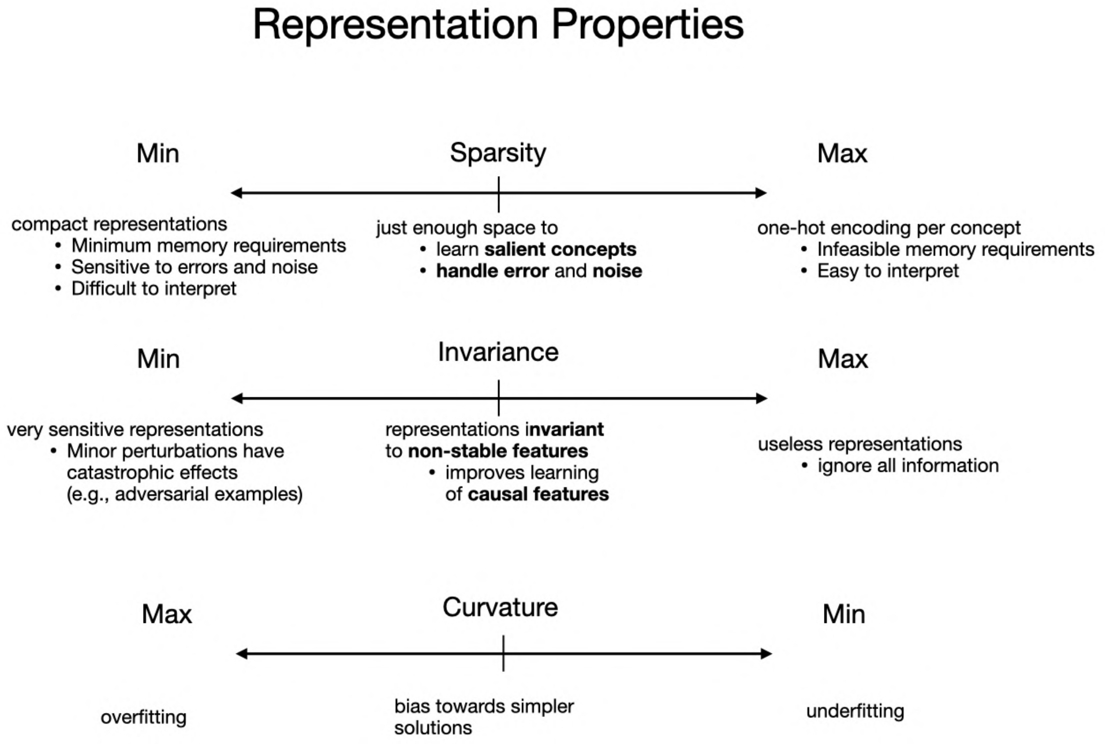

Representation Properties
Min
Sparsity
Max
compact representations
• Minimum memory requirements
• Sensitive to errors and noise
• Difficult to interpret
just enough space to
• learn **salient concepts**
• **handle error** and **noise**
one-hot encoding per concept
• Infeasible memory requirements
• Easy to interpret
Min
Invariance
Max
very sensitive representations
• Minor perturbations have catastrophic effects (e.g., adversarial examples)
representations **invariant** to **non-stable features**
• improves learning of **causal features**
useless representations
• ignore all information
Max
Curvature
Min
overfitting
bias towards simpler solutions
underfitting

Figure C.2: The relation between the model properties of sparsity, invariance and curvature. The representation sparsity and invariance affect the curvature of the decision boundary. Minimum levels of sparsity and invariance lead to overfitting, while maximum levels lead to underfitting. Hence, we need to balance the level of sparsity and invariance to encode different assumptions, such as Occam's razor.

the sparsity, leads to more invariant representations, which could result in a flat decision boundary and underfitting models. Let us now explore these interactions in more details.

Curvature The curvature of the decision boundary possibly contains the most exhaustive information regarding the behaviour of the model (Ho and Basu, 2002). It is also the most important property that determines the training accuracy of a model. Figures C.1 & C.2 illustrate that high curvature leads to more accurate models because more complex relationships can be described; however, an extremely high curvature leads to overfitting due to overparameterised models. Figure C.2 depicts that this overparameterisation can be controlled using the sparsity and invariance of the internal representations to balance the model's representational capacity, thus decreasing the probability of overly high-curvature.

The sparsity and invariance of the representations influence the curvature in distinct ways. As we shall see in the following paragraphs, the invariance increases the smoothness of the decision boundary, whereas sparsity increases the margin between learned concepts (e.g., class identities or manifolds in hidden space). Smoothness and margin have long been established as factors of generalisations and high-performing models (Bartlett and Shawe-Taylor, 1999; Lee, Bartlett, and Williamson, 1995). These ideas can be extended to describe more fine-grained properties of the curvature. The smoothness of the boundary around a particular training point or feature (one dimension of the input space) determines the **local curvature**. On the other hand, the **global curvature** elucidates how the internal representation manifolds and the regions of low-probability between them (i.e., the margins) describe the relationships between multiple features and outputs.

The global curvature describes the overall complexity of the decision boundary. It characterises how the decision boundary folds to produce a number of peaks and troughs, or critical points (points for which the gradient is zero). On the other hand, the local curvature specifies the shape of the boundary in the neighbourhood of a critical point (i.e., how curved the peak or through is). Ideally, we want locally flat boundaries, which are insensitive to minor perturbations, and globally smooth boundaries, such that the transition between manifolds is gradual, but pronounced, so that it reflects the decrease in confidence of the prediction.

A globally smooth and locally flat decision boundary is also preferable for feature importance techniques. The smoothness would make it easier to detect meaningful contributions of the features because there would exist regions where the gradient will be defined and non-zero. On the other hand, the local constancy would mean that an explanation would not attribute unnecessary importance to minor fluctuations in the decision boundary.

Two challenges for feature importance explanations remain: (i) incorporating information that describes the global curvature; (ii) establishing the optimal perturbation size, such that a perturbation is significant to describe variations along the global curvature, but not excessively large to marginalise out a wide range of the model behaviour.

Sparsity The sparsity of hidden representations determines the dimensionality of the manifolds in hidden space. Increasing the sparsity decreases the representational capacity (and by proxy the effective capacity) because the representation can encode a smaller number of concepts ("representational real estate"). The presence of fewer concepts increases the invariance because there is just enough space to learn only the most salient concepts, while discarding the rest (as portrayed in Figure C.2)[3].

As long as the representational capacity is high enough [4], sparsity directly increases robustness because there is more representational real-estate to encode information, which shrinks the probability of overriding (ghosting) or overlapping (interference) [5] concepts representations. In Section B.3.2, we noted that the most important property for robustness

[3] Notice that this statement does not imply that the representation becomes explicitly invariant to confounded concepts. That is, if the concepts of a cow and grass are confounded, the decreased representational capacity might make it more likely that the two concepts remain confounded rather than encoding an explicit invariance towards the spurious signal of grass.

[4] Remember that sufficiently high model capacity is crucial for robustness (Madry et al., 2018).

[5] See Section 2.4 for definitions of ghosting and interference.

is the distance of the nearest point to the decision boundary (Madry et al., 2018; Tsipras et al., 2019). In a sparse representation a more significant change would be required to shift between representations. It might be the case therefore that sparsity is a way to control the differences between representations. The effect of sparsity is that a different low-dimensional manifold (in hidden space) represents each concept, so the distance between manifolds is larger and smoother (i.e., the manifolds are separated by wider and less sharp regions of high-entropy).

We can interpret sparsity as the opposite of compactness. Figure C.2 depicts that a maximally sparse representation encodes each variation in the data in a separate one-hot encoding. This encoding is useful because it is more robust and easier to interpret due to the lower number of superimposed activity patterns of interference or ghosting. However, maximally sparse representation became infeasible to represent since every variation requires additional dimensions to be added to the representation.. Hence, there is a direct trade-off between compact (efficient) and robust representations. A compactly compressed encoding of the same amount of information has less redundancy, which gives more room for error and decreases robustness.

Invariance While sparsity decreases the compression of hidden space, it induces lossless and lossy compression of the input space information in the form of invariance. As a form of **lossless** compression, the invariance eliminates any statistical redundancy or noise signals without affecting the training accuracy. As a **lossy** compression, the invariance eliminates signals that are less relevant or discriminative for the task at hand. However, Figure C.2 illustrates that excessively high invariance could lead to constant classifiers that completely ignore the input. On the other hand, the optimal level of invariance flattens the local curvature around non-discriminative features, thereby improving the model generalisation and robustness (Verma et al., 2019). In fact, the adversarial explanation attack (described in Chapter 4) explicitly induces invariance to particular features to manipulate explanations. The fact that six explanation methods indicate a decrease in feature importance, but we register little change in accuracy also suggests that current explanation methods are over-reliant on local curvature.

Another important benefit of invariance to statistically unstable signals (e.g., image background), is that it increases the likelihood of capturing causal features (Heinze-Deml, Peters, and Meinshausen, 2018)[6].

The identification of features with stable relationships has the potential to move the field of interpretability, and possibly representation learning, higher on the ladder of causal

[6] In Appendices A & B.3.2 we discuss the link between invariance, causality, and generalisation. Specifically, causal factors are invariant to unstable signals across domains; hence, the property of invariance is useful for out-of-distribution generalisation because it reduces the difference between representations of different domains.

queries. That is, moving the field from the level of association (detecting correlations between variables) to a higher form of reasoning – intervention (acting with the world to establish causal relationships) (Pearl, 2009). In Section 3.3.2, we argued that causal explanations are the highest form of interpretability, hence increasing the invariance to spurious correlations will make models more comprehensible and trustworthy.

C.1.3 Model Characteristics

Accuracy & Interpretability Chapter 4 provided additional evidence in support of the Rashomon set hypothesis, demonstrating that we could alter the decision boundary of a pre-trained model and affect the interpretability and the apparent fairness of a model, with little change in accuracy. In Chapter 3, we presented one of the reasons for this phenomenon – the limitation of the majority of explainability methods to describe only very local model behaviour (Jiang et al., 2018). However, both the local and global curvature of the decision boundary play an important part in defining the model characteristics since they determine the effective capacity[7] of a model (Ho and Basu, 2002). An extremely high curvature, coupled with a low dataset complexity, increases the likelihood of low quality models that are overfitting because of small invariance and little robustness to redundant or spurious signals. The model quality directly affects interpretability because *explanations of low-quality models are difficult to validate.* One reason for this is that humans are subject to confirmation bias[8] and will accept an explanation as long as it makes sense to them (Adebayo et al., 2018). Hence, accuracy is not a variable to trade-off with trustworthiness. On the contrary, it contributes to the increased trust in the model. Therefore, future interpretability research should focus not on finding a compromise between accurate and interpretable models, but on describing both the local and global curvature of models.

Robustness & Interpretability In Section B.3.2, we discussed that robustness can be controlled with two properties: (1) closeness of similar concepts; and (2) distance between different concepts; and that the majority of current robustness approaches focus primarily on the former technique.

While invariance of representations is used to control the concept similarity using the local constancy prior[9], the sparsity controls concept dissimilarity by elongating the paths between manifolds, which describe the different concepts. Hence, the optimal conditions

[7] While the representational capacity defines the maximum complexity of the model behaviour, the effective capacity describes the actual capacity of the model after training. Due to limitations of the learning algorithm or idiosyncrasies of the dataset, the effective capacity might be, and often is, smaller than the representational capacity.

[8] See Section 3.5.2.3, “Cognitive fragility”.

[9] See Section 2.2 for more details.

for robust representations are locally flat curvature in input space, and globally distant manifolds in hidden space. A robust representation makes the decision boundary much smoother, and a smoother decision boundary yields significantly better explanations (Dombrowski et al., 2019). It may be the case therefore that explainability and robustness are "two sides of the same coin". For example, a gradient-based explanation is only locally faithful (i.e., within an infinitesimally small region around the decision boundary) (Jiang et al., 2018). However, if the decision boundary has many peaks and valleys in close proximity, minor perturbations have a significant impact on the explanation.

An exciting avenue of future research would be to develop fine-tuned control of the trade-off between maintaining the representational compactness constant in order to improve memory requirements, while modifying the distance and flatness (i.e., the opposite of curvature) between manifolds to achieve robustness and improve explanation quality.

C.2 Supplementary Experimental Results

Here we conduct three different investigations using the experimental set-up defined in Section 4.3.1 (unless stated otherwise) to support the hypothesis that the dataset properties, the curvature, and invariance affect interpretability.

C.2.1 Dataset Features

Here we propose that the analysis of interpretability techniques needs to be grounded in a thorough understanding of the dataset. We support this argument with a study of the datasets properties that influence the effect of our attack. In particular, we investigate the effects of interrelated features (confounding factors) and the individual importance of separate features on the adversarial explanation attack. The aims of this study are to:

1. explore the possibility that the model is using confounding factors to infer the signal of the target feature;

2. investigate whether the attack is a property of the dataset. That is, features that are non-essential to the task are easy to conceal or ignore, in contrast to highly important features.

Confounding Factors A straightforward way to decrease the target feature importance without a significant detriment to the accuracy of the model is to infer the value of the target feature from the set of remaining features. Inferring the feature could be possible if there are confounding factors[10].

[10] See Appendix A.

There are two simple ways to measure the dependence between features: decision-independent and decision-dependent (Qu, Hariri, and Yousif, 2005a; Qu, Hariri, and Yousif, 2005b). The difference between these dependence measures is that the later evaluates the **feature-class** correlations, while the former evaluates feature-feature (**inter-feature**) correlations. In that respect, the decision-dependent analysis assumes that the decision is the confounding factor. On the other hand, the decision-independent analysis measures the upper bound (maximum) degree of dependence between features.

While we investigated decision dependent correlation in Section 4.4.3.2, here we analyse the decision-independent impact of the intrinsic signal that is contained in the features.

A simple well-established metric for measuring both the decision dependent and decision independent (Al-Ani and Deriche, 2002; Qu, Hariri, and Yousif, 2005a) correlation is the mutual information (MI) between the features (Al-Ani and Deriche, 2002), that is, the similarity between the joint $p(x, y)$ and factored marginal $p(x)p(y)$ distributions. In other words, this is the reduction in uncertainty in one random variable x after observing another y: $H(p(x)) - H(p(x|y))$, where $H(x) = \mathbb{E}_{k \sim p(x)}[\log \frac{1}{k}]$ is the uncertainty, or entropy. The MI is zero iff the variables are independent $p(x|y) = p(x)$ (MacKay and Mac Kay, 2003).

The decision independent correlation between features is defined as:

$$I(\mathbf{x}; \mathbf{y}) \triangleq KL(p(\mathbf{x}, \mathbf{y}), p(\mathbf{x})p(\mathbf{y})) = \sum_{x} \sum_{y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \tag{C.1}$$

where KL is the Kullback-Leibler divergence or relative entropy.

We use scikit-learn's implementation (Pedregosa et al., 2011) of mutual _info _classif and mutual _info _regression to estimate the mutual information (I) for discrete or continuous target variables respectively.

Tables C.1, C.2, C.3 demonstrate the risks of confounding-factor interference to our evaluation. The explicitly confounded features can be found in Table C.1. Table C.3 summarises the results from Table C.2 to illustrate that on average there is a considerable number of confounders for each feature across the datasets. In fact, Table C.3 depicts that across all datasets there is on average at least 1 confounded feature and a significant amount of information about the feature can be extracted from other signals. Closer inspection of Table C.2 shows that for three out of the four datasets, the age feature can be almost if not completely inferred from other features. Table C.1 is quite revealing in this way, portraying that some datasets (e.g., COMPAS) are even defined with redundant features (e.g., age and categorical age).

Taken together, the results demonstrate that the i.i.d. assumption does not hold even for many popularly used datasets. Only two out of the ten examined features (compas-gender,compas-race) do not have significantly confounded variables, demonstrating that the majority of the sensitive features are not independent. Even in the case of the features

		Confounders ($X_{:,j}$)
Dataset	Feature ($X_{:,i}$)	
german	gender	response, duration
	age	other-debtor, present-emp
adult	age	workclass, occupation, education, education-num, hours-per-week, relationship, marital-status
	race	native-country
	gender	occupation, marital-status, relationship
bank	age	education, emp.var.rate, nr.employed, cons.price.idx, cons.conf.idx, marital, euribor3m, job
	marital	job, age
compas	gender	
	age	priors-count, age-cat=Greater than 45, age-cat=Less than 25, age-cat=25 - 45
	race	

Table C.1: The weak ($I(X_{:,j}, X_{:,i}) > 0.05$), medium ($I(X_{:,j}, X_{:,i}) > 0.1$), and strong ($I(X_{:,j}, X_{:,i}) > 0.2$) confounding factors ($x_j$) for each target feature (x_i) and dataset in the training data. The mutual information between all the features and the target feature is used to ascertain the confounding factors, while the threshold values were determined after manual observations of the mutual information distribution across non-target features. Colour signifies: weak, medium, and strong confounders.

compas-gender and compas-race the full set of non-target features still contains some information about these target features. The mutual information between the target feature and the full-set of the remaining features is 0.09 and 0.14, for compas-gender and compas-race respectively[11].

The fact that the i.i.d. assumptions does not hold implies that a reasonable explanation technique needs to ground its insights both in the model's operations and the data. At this point, it is not clear whether the fragility of interpretation follows from the unreliable nature of the models or the unreliability of the interpretation techniques. The lack of well-develop techniques to isolate the effects of confounding factors makes both the learning algorithm and the explanation methods extremely susceptible to latent data variations and dependencies. For this reason, we argue that the future of explainability research necessitates well-controlled experimental settings.

Effect of Feature Importance Ranking Another naive way to "fool" all explanation methods is to take an already non-informative feature and decrease its importance. Here

[11] Section 4.4.3.2 presents strong evidence against the possibility that the modified model is ignoring the target feature, while maintaining performance using only information from confounding factors. However, we cannot completely rule out the possibility that our attack somehow forces the model to pay more attention to the information from the confounding factors.

Dataset	Feature ($X_{;,i}$)	$\sum_{j \neq i} I(X_{:,j}, X_{;,i})$	# Weak	# Medium	# Strong
german	gender	0.50	2	2	0
	age	0.19	2	0	0
adult	age	0.95	7	3	1
	race	0.21	1	0	0
	gender	0.61	3	3	1
bank	age	0.90	8	4	0
	marital	0.26	2	1	0
compas	gender	0.09	0	0	0
	age	2.74	4	4	3
	race	0.14	0	0	0

Table C.2: Summary of Table C.1. The number of weak, medium, and strong confounding factors and the average total information contained in all non-target features ($X_{:,-i}$) per target feature ($X_{:,i}$) across the 4 datasets. Notice that there are only two features (compas, gender and race) that do not have significant single confounding factors. Nevertheless, the full set of non-target features still contains some information about the target feature.

Feature ($X_{;,i}$)	$\sum_{j \neq i} I(X_{:,j}, X_{;,i})$	# Weak	# Medium	# Strong
age	1.2	5.2	2.8	1.0
gender	0.4	1.7	1.7	0.3
marital	0.3	2.0	1.0	0.0
race	0.2	0.5	0.0	0.0

Table C.3: The mean values of the number of weak, medium, and strong confounding factors and the average total information contained in all features for the target feature across the 4 datasets. Notice that across all datasets there is on average at least 1 strong confounder and the information about the feature can be extracted from other signals.

we present a case study that investigates how the importance of a feature correlates with the susceptibility of the feature to the attack. Lower values of the explanation loss indicate that the attack was more successful because it induced lower target feature attribution and the particular feature was more susceptible to the attack. The importance of a feature is determined based on ablation experiments, which measure the drop in accuracy when the feature is kept constant. This is a common way to estimate the individual feature importance, sometimes referred to as permutation feature importance (PFI) (Breiman, 2001; Fisher, Rudin, and Dominici, 2019).

Figure C.3 demonstrates that most of the features cluster together given their relative importance and the resulting target feature attribution (i.e., the ability of the attack to affect the curvature of the model w.r.t. each target feature). This observation implies that our attack performs consistently for most features and the German, Adult and Bank datasets[12]. However, the attack susceptibility is lowest (the explanation loss is high) for the most important features. This finding suggests that unless we are attacking the most critical features, features importance does not play a significant role, which means that a variety of minor ethical nuances can be hidden away. The slightly greater difficulty of concealing the most important features is an expected result since the curvature or the slope of the model with respect to the most important features should be the highest. Hence, unsurprisingly, for Adult and Bank, the feature importance is negatively correlated with attack susceptibility (-0.83 and -0.49 Pearson correlation coefficients, respectively). One unanticipated result is that for German, the two metrics seem to be slightly positively correlated 0.34. These differences can be explained in part by a few outlier features that influence the trend strongly or for which in some of the random initialisations the modified model turns into a constant deterministic classifier. There are, however, other possible explanations. One possibility could be related to the ratio between representational capacity and dataset complexity. When a model's capacity is much higher than the

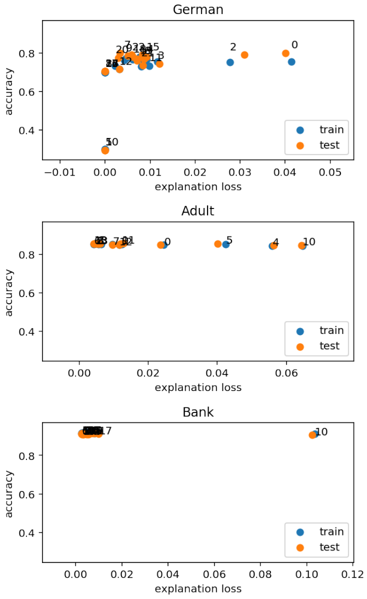


Figure C.3: A scatter plot across all features between the inverse importance of a feature (y-axis) (measured with an ablation study, i.e., the drop of accuracy when the feature is kept constant) and the attack susceptibility (x-axis) (measured as the explanation loss of the modified model after an explanation attack with respect to that feature). Lower accuracy means greater drop due to the feature; hence, the more important a feature is, the lower it is on the y-axis. The clustering effect of the points is on purpose since scale of the y-axis is shared across the three subplots to demonstrate the differences of feature importance across each of the datasets. The colours represent the train (blue) and test (orange) datasets, depicting the the features maintain their relative importance for both datasets. What is important in this plot is that most of the features cluster together, suggesting that for the majority of features importance does not play a significant role.

[12] We do not investigate COMPAS due to the computational implications of conducting the experiments across 400 features.

dataset complexity (as is the case for German), there is more “room” to wiggle, and it is easier to conceal a feature. Another possibility is that the attack has a regularisation effect in the regime of limited training data, yielding smoother decision boundaries and better performance. We discuss further evidence and the implications of each of the two hypotheses in Appendix C.2.3.1.

C.2.2 Curvature

The aim of this section is to support the hypothesis that the adversarial explanation attack can be used as a preference articulation technique because it affects the curvature of the decision boundary. As such it influences feature importance explanation techniques and hence the explainability model characteristics.

$\nabla_{\boldsymbol{X}_{:,j}}\mathcal{L}$ vs $\nabla_{\boldsymbol{X}_{:,j}} f(\mathbf{x})$ Both the adversarial explanations attack (described in Chapter 4) and the method in Heo, Joo, and Moon (2019) penalise the gradient with respect to the loss function($\nabla_{\boldsymbol{X}_{:,j}}\mathcal{L}$) rather than the gradient with respect to the element of the output vector corresponding to the correct label ($\nabla_{\boldsymbol{X}_{:,j}} f(\mathbf{x})$). Here we study the implications of using either approach to the success of the attack and the resulting curvature of the model in a similar fashion to the eigenvalue spectral analysis of the Hessian in Moosavi-Dezfooli et al. (2019).

Remark C.2.1

Here we briefly review the idea of eigenvalue spectrum. The eigenvalue spectrum of a matrix is the set of all eigenvalues. A key element of this spectrum is the absolute maximum eigenvalue, which is known as the spectral radius, or spectral norm, of a matrix. The spectral radius helps us gain some perspective about the local curvature of the decision boundary in the neighbourhood of training points. In Section B.3.2 Paragraph “Small curvature in the vicinity of datapoints” we briefly mentioned that the maximum / minimum **eigenvalues** of the Hessian determine the maximum / minimum second derivatives, thereby **determining** the degree of **curvature**. Since the second derivative is a measure of curvature, when the second derivative is positive, the function curves upwards, whereas when second derivative is negative, the function curves downwards. When the second derivative is zero, the function is flat. Notice that flat does not imply constant. Only when the first derivative is also zero, then there is no slope and the function is locally constant.

Theoretically, the gradient w.r.t the loss is:

$$\nabla_{\boldsymbol{X}_{:,j}}\mathcal{L} = \nabla_{\boldsymbol{X}_{:,j}} - \mathbb{E}_{\hat{p}_{\text{data}}}\left[\mathbf{z}_i - \log \sum_k \exp(\mathbf{z}_k)\right], \tag{C.2}$$

whereas the gradient w.r.t the element of the function output corresponding to the correct label is:

$$\nabla_{\boldsymbol{X}_{:,j}} f(\mathbf{x}) = \nabla_{\boldsymbol{X}_{:,j}} \exp(\mathbf{z}_i) - \sum_k \exp(\mathbf{z}_k), \tag{C.3}$$

where j is the target feature, $\mathbf{z}_i$ is the pre-softmax logit $f(\mathbf{x})_i = \text{softmax}(\mathbf{z})_i = \frac{\exp(\mathbf{z}_i)}{\sum_j \exp(\mathbf{z}_j)}$, and i is the element of one-hot encoding output vector corresponding to the correct class $\mathrm{y}_i = 1. y_i \in \mathbf{y}$.

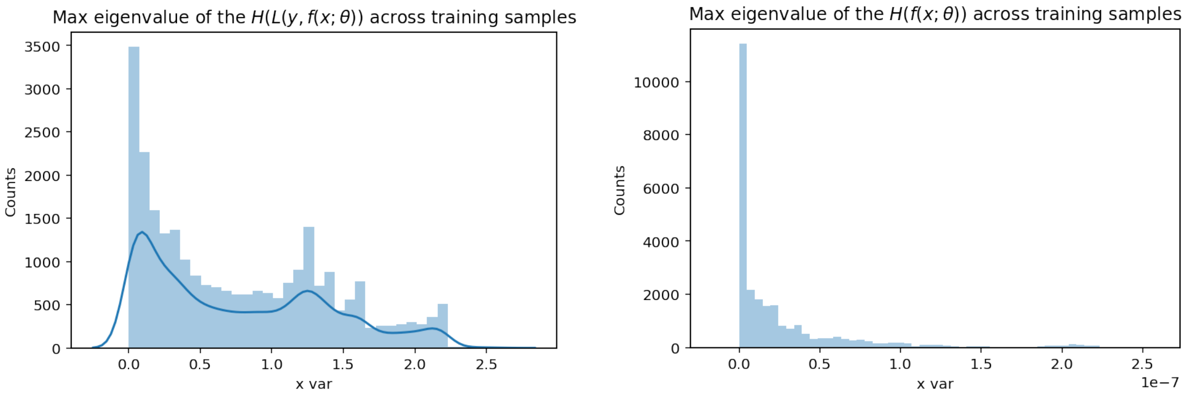
Equations C.2 & C.3 taken together suggest that theoretically, the main difference between the approaches is whether we undo the exponent term with the log function or not. Both Equations C.2 & C.3 include the contributions of each neuron within the entire output vector due to the denominator of the softmax function. We observed that taking the activation of the pre-softmax logit made the training process extremely unstable, causing violent oscillations of the loss function. The reasons for this strange finding can be explored in future work.

Experimental Set-up Next we conduct two eigenvalue spectrum analysis experiments on the 5-hidden layer model architectures to explore how the curvature of the model is affected by (1) the different loss functions stemming from Equations C.2 & C.3; and (2) the adversarial explanation attack regularisation in comparison to the original and constant models defined Chapter 4.

In the first experiment, we compare the distribution of the maximum eigenvalue of the Hessian (H) with respect to the different loss functions: (a) $\mathcal{L}$ and (b) $f(\mathbf{x})$ evaluated across every training sample. This experiment gives us an understanding of the overall curvature of the decision boundary. On the other hand, in the second experiment, we look at the second partial derivatives w.r.t a feature and dataset for the modified, constant, and original 5-hidden layer models. This experiment gives an intuition about the curvature around the particular feature. Therefore, the former experiment examines global curvature, whereas the latter examines local curvature.

Spectral analysis across loss functions Figure C.4 demonstrates that there is a significant difference in the eigenvalue spectrum between $\nabla_{\boldsymbol{X}_{:,j}} \mathcal{L}(\mathbf{y}, f(\mathbf{x}; \boldsymbol{\theta}))$ and $\nabla_{\boldsymbol{X}_{:,j}} f(\mathbf{x}; \boldsymbol{\theta})$. A comparison between Figures C.4a & C.4b suggests that models attacked directly on the logits seem to have much flatter models (with most eigenvalues being zero or less than 10^-7, which is close to numerical error). Higher flatness increases the likelihood of ignoring rather than concealing the target feature, making the use of $\nabla_{\boldsymbol{X}_{:,j}} f(\mathbf{x}; \boldsymbol{\theta})$ less suitable.

Figure C.4 illustrates that there is a large number of zero maximum eigenvalues of the Hessian (H) for both $\mathcal{L}$ and $f(\mathbf{x})$. However, the zero maximum eigenvalues for $f(\mathbf{x})$ are at least three times more than those for $\mathcal{L}$ ($> 10,000$ vs 3,500) and the highest values are no



(a) Hessian with respect to $\mathcal{L}$.

(b) Hessian with respect to $f(\mathbf{x})$.

Figure C.4: Distribution of the maximum eigenvalue of the Hessian (H) with respect to (a) $\mathcal{L}$ and (b) $f(\mathbf{x})$ evaluated across every training sample for Adult-gender. Notice that the distribution in the case of $\mathcal{L}$ is much more multimodal and spread out, whereas the distribution w.r.t $f(\mathbf{x})$ is Laplace distributed with 0 mean. Further, observe the different scales of the distribution. While (a) is on the scale of $[0, 2.5]$, (b) is on an *exponentially smaller scale* – $[0, 10^{-7}]$, which is approximately zero. When most of the maximum eigenvalues are 0, the decision boundary is flat in the vicinity of the training points.

larger than 2.5^{-7}, which is approximately zero. The fact that most maximum eigenvalues for $f(\mathbf{x})$ tend to zero implies that most of the decision boundary geometry consists of degenerate locations of wide, flat regions of constant value, where both the gradient and the Hessian could be zero. Therefore, the exponentiation of softmax (as in Equation C.3) leads to a model that is completely ignoring its input data. Additionally, the exponentiation makes it more likely for the optimisation algorithm to encounter computational instability due to the extremely tiny values of the gradient. This insight is one example of using an understanding of the curvature property to guide the development process, helping us to design a better adversarial explanation attack. Specifically, we choose to differentiate with respect to the loss to increase the likelihood of concealing rather than ignoring the feature, and to maintain a numerically stable computation.

Spectral analysis across models Figure C.5 illustrates the effect of the adversarial explanation attack on the curvature of the modified, constant, and original models (defined in Chapter 4). The most interesting aspect of this figure is that the modified model has second partial derivatives with respect to the target feature that are: (1) orders of magnitude smaller than those of the original model, but (2) consistently larger than those of the constant model.

These findings have two implications. First, the adversarial explanation attack significantly affects the curvature of the model. At the same time, in Chapter 4 we demonstrated that the attack influences the results of multiple explanation techniques. These observations may support the hypothesis that current explanation methods are highly dependent on the

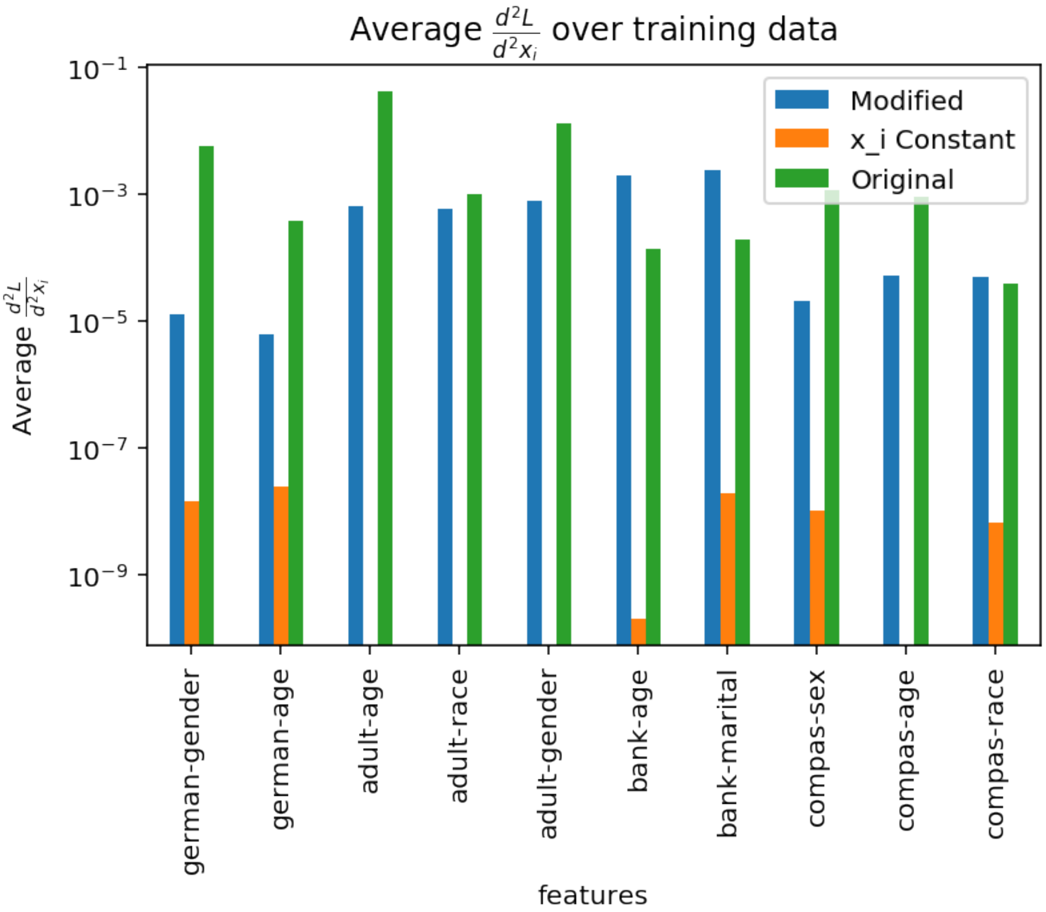


Figure C.5: Log-scale plot of the average second partial derivatives w.r.t the corresponding feature and dataset on the x-axis averaged over all training points for the modified, constant, and original 5-hidden layer models across 10 initialisations. Notice that although the modified model on average has exponentially smaller eigenvalues than the original model, it has exponentially larger values than the constant model. Hence, the degree of curvature of the model is somewhere in between the original and constant models, which is a strong indication that the signal from the feature is preserved globally. Additionally, the curvature of the **modified models** for German (the smaller dataset) is a degree lower than the other datasets suggesting the presence of overly flat models in the regime of limited training data.

local curvature, which can be manipulated with the adversarial explanation attack. Second, the comparison of partial derivatives between the three types of models depicts that it is less likely for the modified model to be ignoring the feature. Instead, the particular form of the resulting decision boundary conceals the signal from current explanation methods. For example, the decision boundary has a particular shape, which is flat in the infinitesimally small neighbourhoods around training points, but curves outside these neighbourhoods. That is, the decision boundary is locally flat, but globally curved.

C.2.3 Invariance and Robustness

Recently it has been demonstrated that robust training[13] leads to interesting properties, including smoother and more semantically meaningful classification boundaries (Tsipras et al., 2019). Here we investigate the relationship between robustness and the quality of explanations through the shape of the decision boundary. In particular, we investigate the relationship of the explanation attack to robustness. We demonstrate that robust training converges to significantly different parameters, which affect both the model curvature and fidelity of explanations.

We conduct experiments on three datasets – German, Adult, Bank. We do not include COMPAS in the investigation since it contains 400 one-hot encoded features, which makes robust training awkward because a value of one feature can be moved simultaneously in multiple mutually exclusive directions (e.g., both male and female).

We examine the effect of robust training on the model accuracy and attack susceptibility (which is a proxy for the curvature of the model with respect to the target feature, measured as the explanation loss convergence) in three different settings: (1) vanilla training with vanilla attack (vanilla); (2) vanilla training with robust attack (robust attack); (3) robust training with robust attack (robust init & attack).

In the setting of robust attack we continue the robust training, while conducting the attack (i.e., preserve the robust training term in the loss function). Therefore, the training objective is now as in Equation 4.1 where $\mathcal{L}$ uses the loss term from Tsipras et al. (2019):

$$\mathcal{L} = \max_{\delta \in \Delta} \ell(\boldsymbol{x} + \boldsymbol{\delta}, \boldsymbol{y}; \Theta) \tag{C.4}$$

where ℓ is the categorical cross entropy, Θ is the vector of model parameters $f(\mathbf{x}; \Theta)$, and $\Delta = \{\boldsymbol{\delta} \in \mathbb{R}^d \; \big| \, ||\boldsymbol{\delta}||_p < \epsilon\}$ is the set of allowed perturbations (Madry et al., 2018; Tsipras et al., 2019).

These data must be interpreted with caution because the experiments are performed for a particular model complexity of 5 hidden-layer MLP due to substantial computational requirements.

C.2.3.1 Findings

Figure C.6 summarises the results for the most important feature of each of the three datasets. It reveals that both robust training and robust attacking influence differently the attack susceptibility (measured as the explanation loss) and model performance (measured as accuracy). We find seven notable results:

1. The parameter setting prior to the attack (initialisation) converge to considerably

[13] See Section B.3.2 for definitions and details.

different optima, which can have serious implications for the accuracy and fidelity of explanations.

2. One unanticipated finding is that robust training might be a useful defence mechanism.

3. The adversarial explanation attack and robust training might be affecting the curvature at different scales.

4. The discrepancy of the effect on curvature between robust training and our attack could be due to feature interactions, the effective capacity, or the dataset complexity.

5. In the setting of our attack, the relationship between robustness and accuracy conditioned on the size of the dataset inverses. That is robust training is detrimental to the accuracy for smaller datasets, but it is beneficial for larger datasets.

6. High uncertainty over the model parameters (suggested by violent performance oscillations) might be one possible explanation for the inconsistent results on scarce data.

7. The instability of convergence for smaller datasets might raise intriguing questions regarding the role of datasets in understanding the model performance.

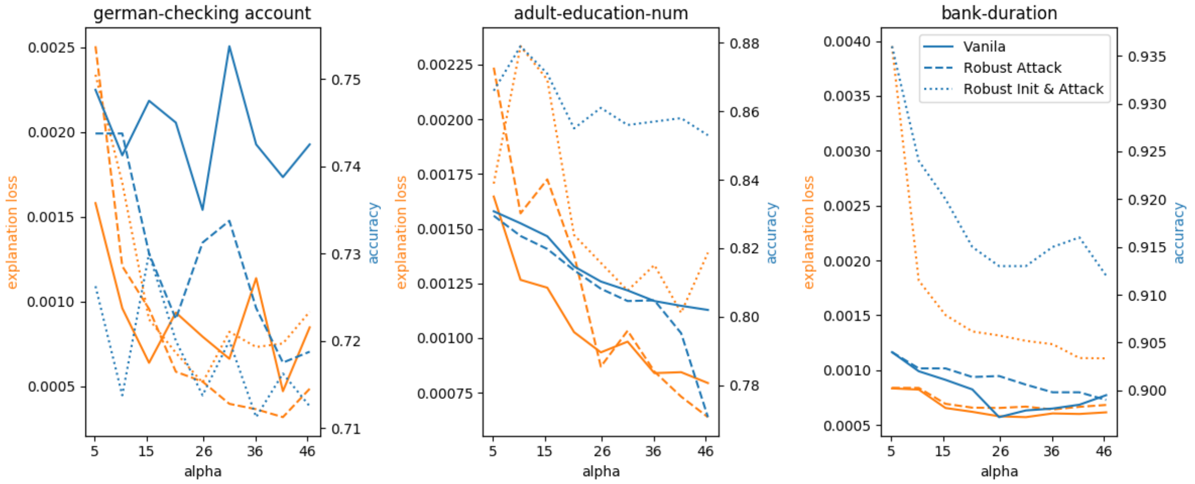


Figure C.6: Illustration of the effect of robust training on the susceptibility across different values of alpha for the most important feature, "checking account","education-num",and "duration", of respectively German (left), Adult (center), Bank (right). The *solid line* indicates indicates vanilla explanation attack, the *dashed line* indicates vanilla training and robust attack, and the *dotted lines* indicates robust training with robust attack. Orange lines show explanation loss, while blue lines show accuracy.

Conclusions

The aforementioned findings demonstrate that regularisation techniques such as the adversarial explanation attack and robust training could help us transition along the Rashomon curve of models. However, the adversarial explanation attack and robust training could be controlling different aspects of the decision boundary curvature in terms of local and global effects. Additionally, our findings illustrate that the resulting shape of the decision boundary, and by proxy, the accuracy and fidelity of explanations of models, is highly dependent on the model parameters (i.e., the representation properties) and the dataset properties. Our findings give arguments in favour of the hypothesis postulated that robustness and interpretability are very likely related through the curvature of the decision boundary and the stability of the solution in parameter space. Given that such subtle differences in the parameter configurations have substantial implications for the results of explainability method, we propose that future interpretability research should rely on well-understood and possibly manually defined **models**. This configuration would ensure that the model becomes a **controlled variable** when conducting scientific experiments with explainability techniques.

APPENDIX D

Regularisation for Preference Articulation

Regularisation is a standard way to control model characteristics. Here we argue that the adversarial explanation attack is a subset of regularisation methods, which can be used to control the model characteristics with more precision. A surprising result in Chapter 4[1] is that as we increase the representational capacity, the modified models achieve higher performance than both the original and the constant models. As expected, deeper models trained for the same number of epochs as more shallow models converge to optima of lower accuracy. Training deeper models for the same number of epochs and same hyper-parameters of early stopping imposes a strong prior that the weights do not change significantly from their initial values. Naturally, it is more difficult to fit an over-parameterised model (Goodfellow, Bengio, and Courville, 2016a). These data must be interpreted with caution because we train all models for the same number of 1000 epochs with early stopping and patience of 100 epochs. However, it seems possible that these results are due to a regularising effect of the explanation loss term. Hence, the adversarial explanation attack is an instance of such model fine-tuning approaches. Here, we explore this conjecture in more depth.

The adversarial explanation attack is indeed similar to two regularisation techniques: (1) tangent propagation (Simard et al., 1992) and (2) double backpropagation (Drucker and Le Cun, 1992). Similarly to our attack, tangent propagation includes an additional penalty term, which makes the output of the black-box classifier invariant to pre-defined factors of variation. The class of methods that append a term containing the derivatives of the output with respect to the input to their loss can be unified under the family of double backpropagation methods (Etmann, 2019).

Double backpropagation forces the Jacobian of the output function with respect to the

[1] See Figure 4.10.

input to be small. A quintessential example is the Contractive autoencoder (CAE) (Rifai et al., 2011a). The CAE introduces an explicit regulariser to the reconstruction loss in the form of double backpropagation term:

$$\left\| \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right\|_F^2 . \tag{D.1}$$

Double backpropagation is comparable in principle to adversarial training, which synthesises new input points in the vicinity of each training point and optimises the model to assign them the same output as the original point. Both approaches use a contractive mapping to encode the local constancy and sparsity priors[2] (Rifai et al., 2011b). A contractive mapping warps the space to associate neighbourhoods in input space to smaller neighbourhoods in output space (i.e., the model exhibits smooth output within local neighbourhoods). As a result the model becomes *invariant* to changes in *all directions* in input space, as long as these changes are small, and highly *sensitive* to very *few salient directions*. In contrast, our attack and tangent propagation require the model to become invariant to particular *user-specified directions*. Hence, our attack can be seen as a special case of tangent propagation.

Local smoothness, global curvature Naturally, the invariance constraints of CAEs induce the ideal robust representation characteristics of locally similar representations, which are invariant to noise, and globally different representations, which are sensitive to changes in salient directions. Rifai et al. (2011a) demonstrate that this type of invariance results in sparser representations and lower-dimensional manifolds when compared to other autoencoders.

The model becomes locally invariant, but globally two different points $\mathbf{x}^i$ and $\mathbf{x}^j$ may or may not have similar output values – $f(\mathbf{x}^i) \nsim f(\mathbf{x}^j)$. We conjecture that the adversarial explanation attack has a similar regularising effect as autoencoders. Therefore, the attack might not lead to learning features that are constant with the input, but might instead learn features that are locally constant and globally varying. If this hypothesis holds, we would expect to see local smoothness, but global curvature in the predictor function with respect to the target feature. We already observed some evidence of this hypothesis in Section 4.4.3.1, and we investigated this conjecture further in Section C.2.2.

Infinitesimal vs fixed-sized perturbations Tangent propagation is comparable to another regularisation concept – dataset augmentation. In both cases, the user includes prior knowledge by encoding the types of transformations, to which the model should become invariant. However, tangent propagation regularises the model to resist infinitesi-

[2] See Section 2.2.

mally small perturbations to the input. On the other hand, dataset augmentation makes the model resist larger fixed-sized transformations. Consequently, we can think of tangent backprop as the infinitesimal version of dataset augmentation. Just as tangent propagation is the infinitesimal version of dataset augmentation, adversarial (robust) training is the infinitesimal version of double backpropagation (Alain and Bengio, 2014).

The contractive and denoising autoencoders are related in a similar fashion. Essentially, the difference is that contractive autoencoders encourage the encoder function to resist infinitesimally small changes in the input. In contrast, the denoising autoencoders encourage the encoder function to resist slightly larger finite-sized perturbations of the input (Alain and Bengio, 2014).

So far we have seen that tangent propagation, double backpropagation, CAEs are all the infinitesimal version of their fixed-size perturbation counterparts – dataset augmentation, adversarial training, and denoising autoencoders. Notice that the key difference is whether the technique enforces infinitesimal or fixed-sized changes. While in the former case we modify the parameters directly through optimisation, in the latter we modify the parameters indirectly through introducing changes in the input. The adversarial explanation attack bears a resemblance to tangent propagation and double backpropagation. By extension, our attack is similar to contractive autoencoders because of its method of modifying parameters.

The three correspondences between building invariance to infinitesimal changes or fixed-sized perturbations suggest that we might be able to downgrade a target feature without modifying the model. Indeed, we could introduce small fixed-size perturbation in the input to modifying the network parameters indirectly and downgrade a target feature (Ghorbani, Abid, and Zou, 2019). This leads us to a theory unifying all these approaches. Our method and the method proposed in Ghorbani, Abid, and Zou (2019) are related in the same fashion as contractive to denoising autoencoders, tangent propagation to dataset augmentation, and double backpropagation to adversarial training, and the relation is the former is the infinitesimal version of the latter.

appendix E

Depeendency Graphs Additional Results

Here we provide supplementary figures to the arguments made in Chapter 5. Figure E.1 illustrates the performance of all 10 class-specific dependency graphs across 10 thresholds. Figure E.2 & E.3 demonstrate that DGINN is not affected by sparse-parameter (L1) or weight decay (L2) regularisation. Figures E.4 & E.5 demonstrate the heatmap pixel contribution visualisation of each of the most relevant neurons for both the Hammerhead shark and Egyptian cat classes, respectively.

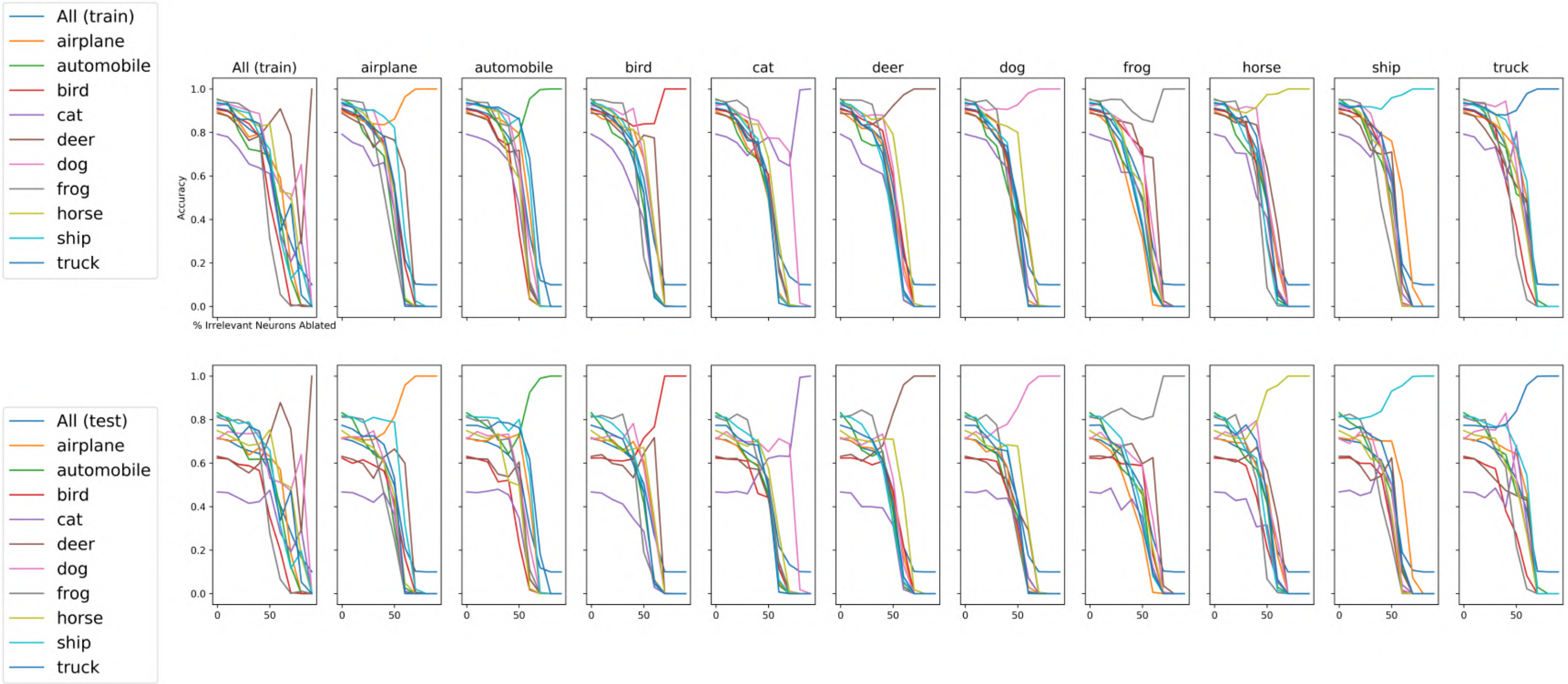


Figure E.1: A comparison of the true positive rates (TPRs) between all 10 class-specific dependency graphs extracted with the gradients relevance function. The lines indicate the accuracy of network once all neurons outside the dependency graph have been masked. Each column indicates the data-set, for which the dependency graph has been specialised. The row indicates whether the datasets belong the training (row 1) or test set (row 2).

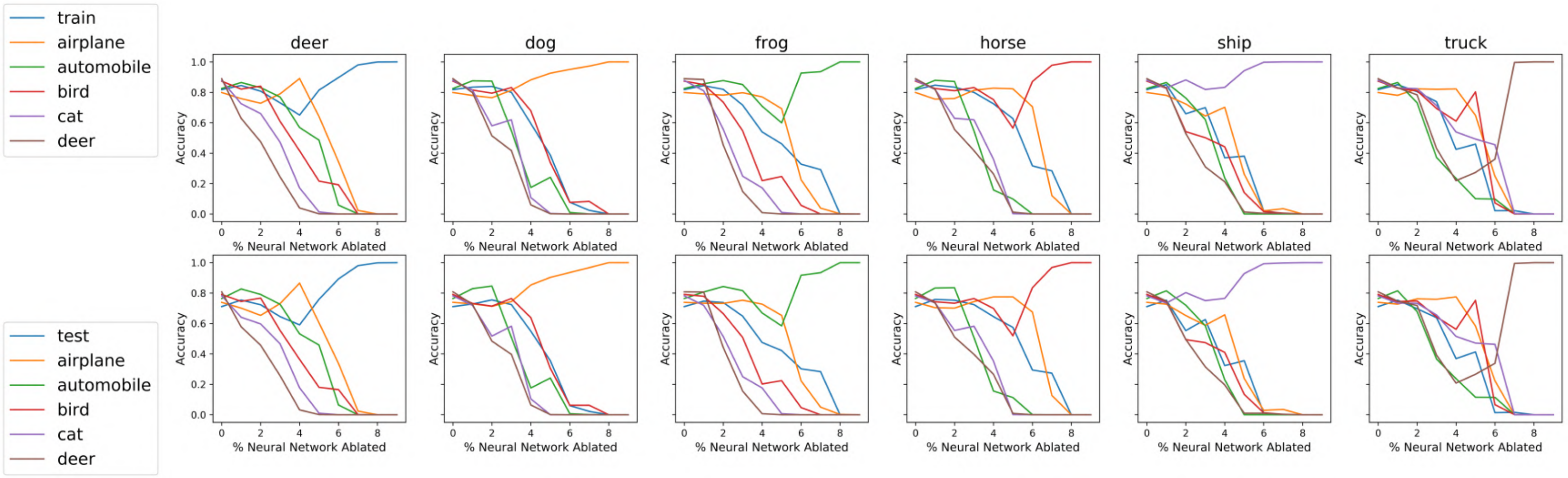


Figure E.2: A comparison of the true positive rates (TPRs) between 5 class-specific dependency graphs extracted with the gradients relevance function on a Conv-Net with **L1**-regularisation on all weight layers with norm penalty parameter $\alpha = 0.001$. The lines indicate the accuracy of network once all neurons outside the dependency graph have been masked. Each column indicates the data-set, for which the dependency graph has been specialised. The row indicates whether the datasets belong the training (row 1) or test set (row 2).

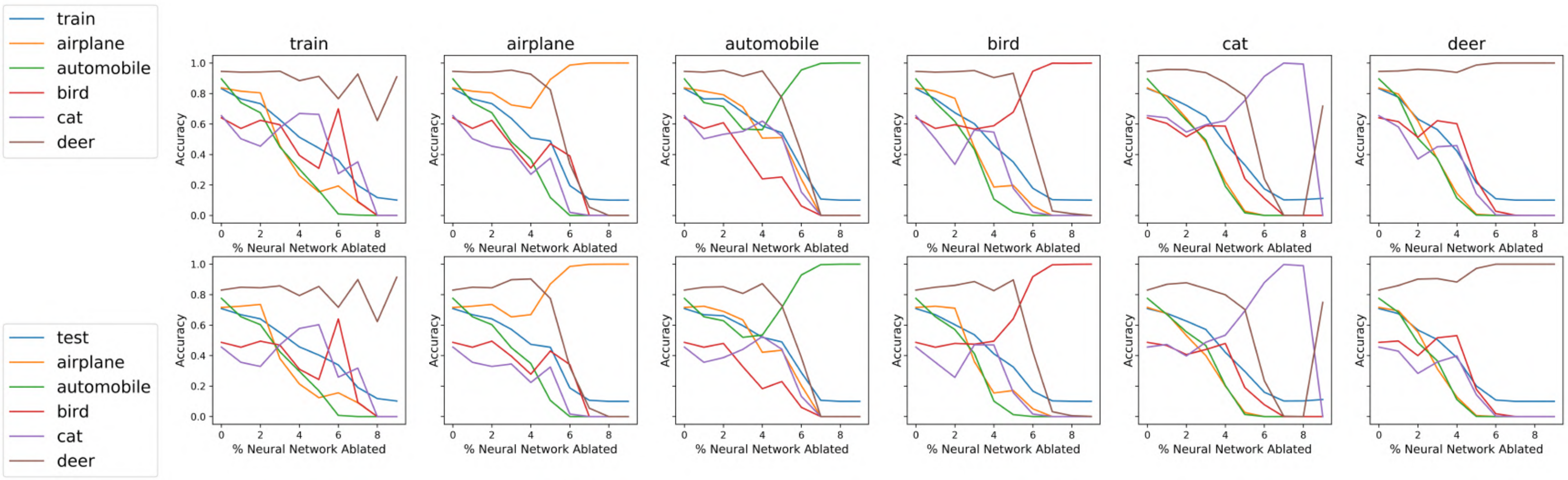


Figure E.3: A comparison of the true positive rates (TPRs) between 5 class-specific dependency graphs extracted with the gradients relevance function on a Conv-Net with **L2**-regularisation on all weight layers with norm penalty parameter $\alpha = 0.001$. The lines indicate the accuracy of network once all neurons outside the dependency graph have been masked. Each column indicates the data-set, for which the dependency graph has been specialised. The row indicates whether the datasets belong the training (row 1) or test set (row 2).

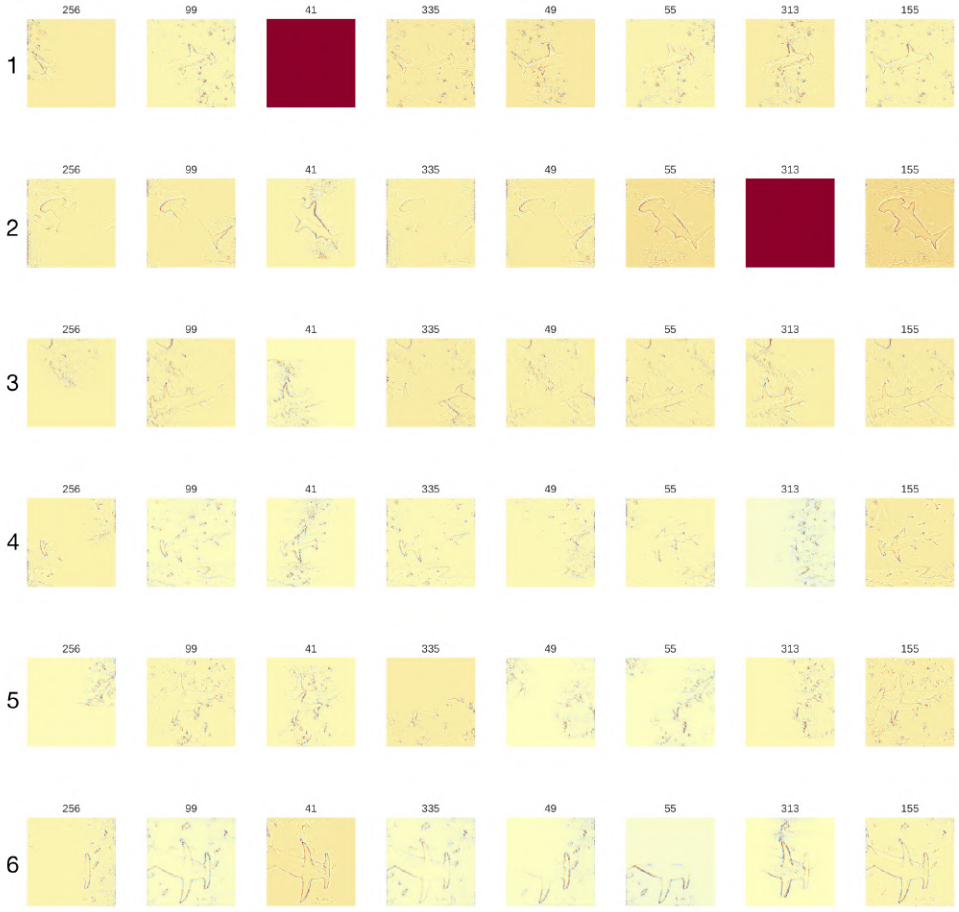


Figure E.4: Heatmaps of all activation maps at layer f^{b5c3}, relevant to neuron f^{fc2}_{1820} for Class 4: 'Hammerhead shark'. The red heatmaps indicate absence of relevant pixels to a particular activation map (best viewed in digital).

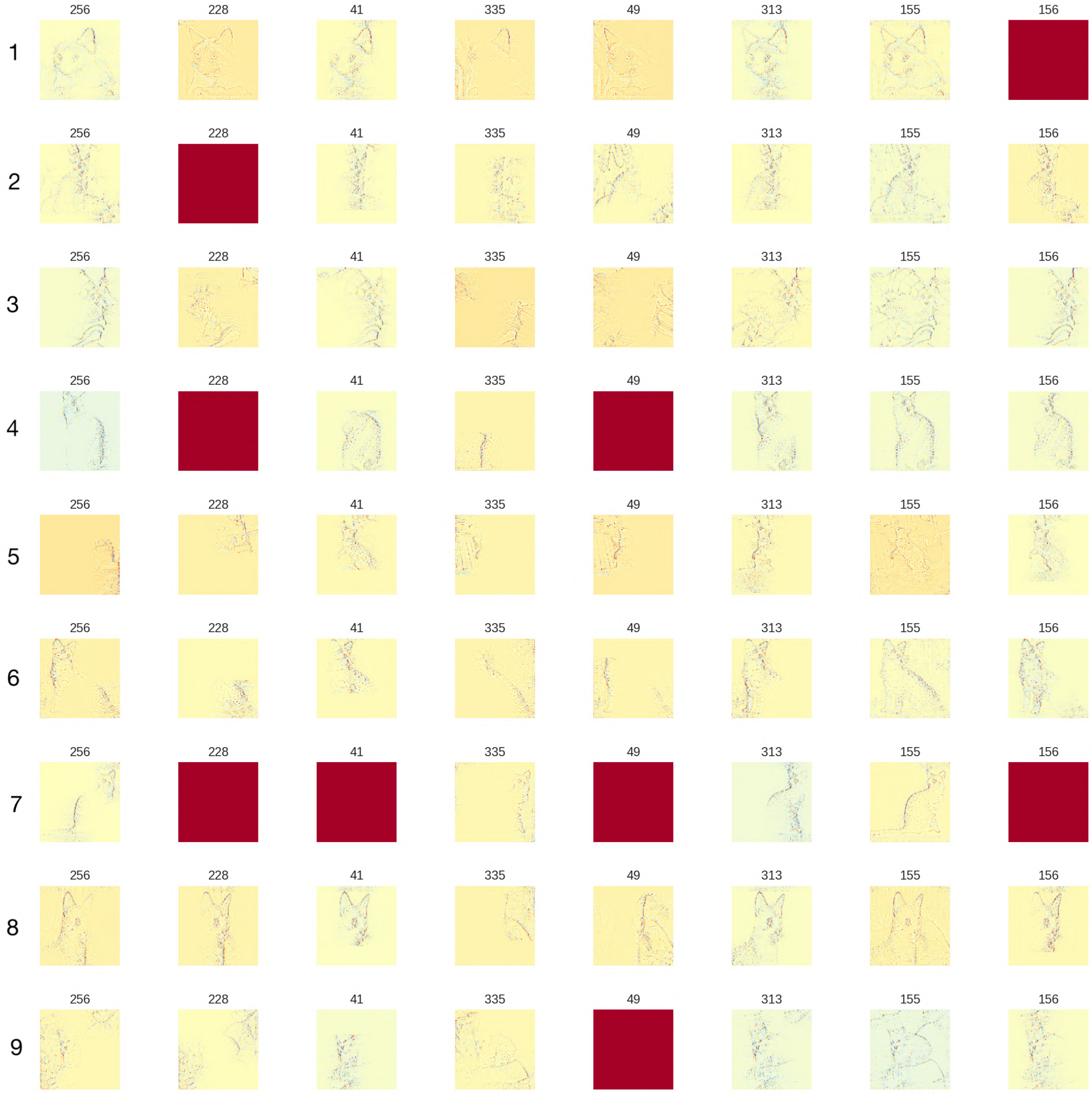


Figure E.5: Heatmaps of all activation maps at layer f^{b5c3}, relevant to neuron f^{fc2}_{1820} for Class 285: 'Egyptian cat'. The red heatmaps indicate absence of relevant pixels to a particular activation map (best viewed in digital).

APPENDIX F

CME Additional Results

The CUB model has a considerably larger number of layers, and a considerably larger number of task concepts. Hence, for the sake of space, we demonstrate an example here using only 6 different model layers of the CUB model, and showing only the top 5 important concepts identified using the magnitude of the parameters of a linear regressor trained to predict the outputs given concept labels. In Figure F.1, the concepts are named using their indices, and the layers are named following the naming convention used in Koh et al. (2020). Further details regarding layer naming and/or concept naming can be found in the official repository[1]. For all concepts, concept values become significantly better-separated after the `Mixed_7c` layer. However, the figure shows that concept values are still quite mixed together for some of the points, even for later layers. This low separability indicates that concept values will still be mis-predicted for some of the points, and that concept extraction for the CUB task will likely perform suboptimally.

[1] https://github.com/yewsiang/ConceptBottleneck/tree/master/CUB

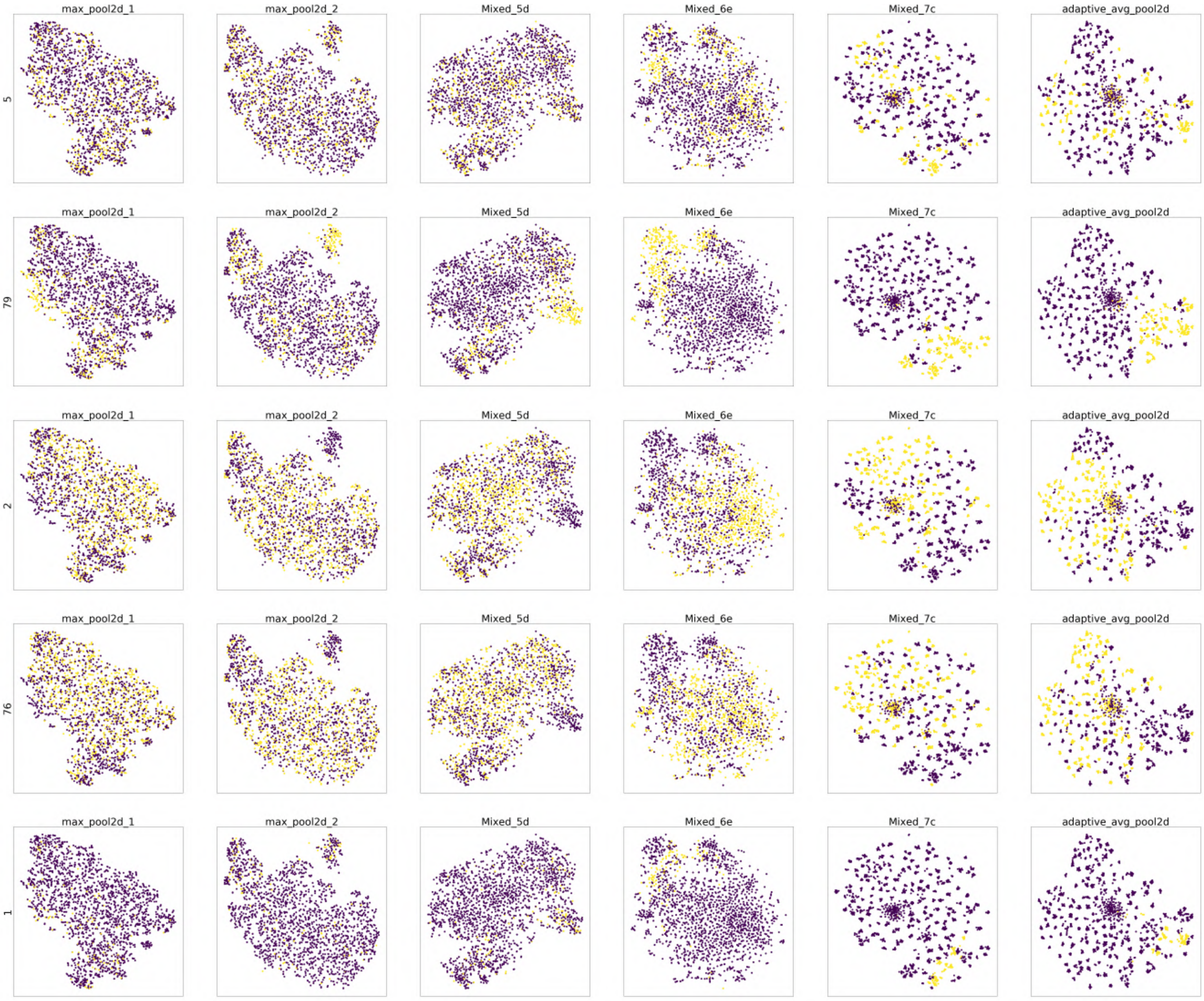


Figure F.1: t-SNE plots for the top 5 CUB concepts. Each column corresponds to a different layer of the CUB model. Each plot is coloured with respect to the concept's values.

Index