Self Explaining Neural Networks: A Review

Omar Elbaghdadi
12660256
omarelb@gmail.com

Aman Hussain
12667447
aman.hussain@student.uva.nl

Christoph Hoenes
12861944
christoph.hoenes@gmail.com

Ivan Bardarov
12579572
ivan.bardarov@student.uva.nl

ABSTRACT

For many applications, understanding *why* a predictive model makes a certain prediction can be of crucial importance. In the paper "Towards Robust Interpretability with Self-Explaining Neural Networks", [Alvarez Melis and Jaakkola 2018] propose a model that takes interpretability into account by design. We study the reproducibilty and validity of the proposed framework. Several weaknesses of the approach are identified. Most notably, we find that the model rarely generates good explanations, and that performance is compromised more than reported by the authors when enforcing explanations to be stable. We put forward improvements to the framework that address these weaknesses in a principled way, and show that they enhance the interpretability of generated explanations.

1 INTRODUCTION

Increasingly, decisions that have a critical impact on peoples' lives are taken by machine learning (ML) algorithms. It is therefore imperative that they should satisfy some important criteria [Doshi-Velez and Kim 2017] such as safety [Amodei et al. 2016; Otte 2013; Varshney and Alemzadeh 2017], not discriminating against certain groups [Bostrom and Yudkowsky 2014; Hardt et al. 2016; Ruggieri et al. 2010], and being able to provide the right to explanation of algorithmic decisions [Goodman and Flaxman 2017].

These criteria are often hard to quantify completely. Instead, a proxy notion is regularly made use of: *transparency* [Ribeiro et al. 2016a]. The idea is that if we can explain the inner workings of a model i.e. *why it makes the predictions it does*, then we can check whether that reasoning is *reliable*. Currently, there is no complete consensus on the definition of transparency or how to evaluate it.

One approach taken towards attaining model transparency is to formulate a framework of models that are transparent *by design*. The models derived from this framework are then inherently transparent. This approach is also taken by [Alvarez Melis and Jaakkola 2018], henceforth "the authors", who propose a self-explaining neural network (SENN) that optimizes for transparency **during** the learning process. They also propose three desiderata for explanations in general – explicitness, faithfulness, and stability.

The authors propose to achieve transparency by learning interpretable feature representations called concepts. Each feature is then given a relevance score which is enforced to behave similarly for relatively small changes in concepts. Their experiments yield good explanations and minor accuracy losses. *Our contributions* are as follows: We study the reproducibilty and validity of the proposed framework, and we provide an extension that clearly increase the interpretability of generated explanations.

2 METHOD

Much recent work has focused on post-hoc interpretability methods, which try to understand a model's inner workings *after* it has been trained [Lundberg and Lee 2017; Ribeiro et al. 2016a]. Most of these methods make no assumptions about the model to be explained, and instead treat them like a black box. In contrast, SENNs try to take interpretability into account *by design*, without sacrificing too much modelling power. The difference between these methods will be elaborated on in Section 6.

The three desiderata proposed for explanations are:

(1) **Explicitness**: *Are the explanations immediate and understandable?*
(2) **Faithfulness**: *Are relevance scores indicative of "true" importance?*
(3) **Stability**: *How consistent are the explanations for similar examples?*

2.1 Self Explaining Neural Networks

The authors begin their treatment of SENNs by positing that, for input features $x_1, \ldots, x_n \in \mathbb{R}$ and parameters $\theta_0, \ldots, \theta_n \in \mathbb{R}$, the linear model $f(x) = \sum_i^n \theta_i x_i + \theta_0$ is an interpretable model. We discuss the validity of this assumption in Section 6.

Then, they generalize the linear model by allowing it to be more complex, while retaining the interpretable properties of a linear model. A *self explaining neural network* f is then defined by:

$$f(x) = g(\theta(x)_1 h(x)_1, \ldots, \theta(x)_k h(x)_k),$$

where

- θ is a neural network mapping input features to *relevance scores*, or *parameters*. We call this a **parameterizer**, and elaborate on it more deeply in Section 2.2.
- $h : \mathcal{X} \to \mathbb{R}^k$ computes k interpretable feature representations of the input x, where k is small. These feature representations are referred to as *basis concepts*. We call this a **conceptizer**, and discuss properties basis concepts should have in Section 2.3.
- g is a monotonically increasing, completely additively separable *aggregation function*.

The *explanation* of $f(x)$ is then defined to be the set $\mathcal{E}_{f(x)} := \{(h(x)_i, \theta(x)_i)\}_{i=1}^k$ of the basis concepts and their relevance scores.

2.2 Parameterizer

An important property for the interpretability of linear models is that parameters stay constant as feature values vary. This property is lost when the parameters $\theta(x)$ are highly complex functions of input features. For $\theta(x)$ to act as coefficients of a linear model in the basis concepts $h(x)$, the authors propose that the parameterizer θ should be **locally-difference bounded** by the conceptizer h. Intuitively, this means that for a small region around some input value x_0, a small change in h should lead to a small change in θ, i.e. θ is robust to small changes in the concept values.

Locally-difference boundedness is enforced by minimizing the *robustness loss*:

$$\mathcal{L}_\theta := \|\nabla_x f(x) - \theta(x)^{\mathrm{T}} J_x^h(x)\|, \tag{1}$$

where $J_x^h(x)$ is the Jacobian of h with respect to x.

2.3 Basis Concepts

Concepts could be generated by domain experts, but this is expensive and in many cases infeasible. An alternative approach is to learn the concepts directly [Kim et al. 2018]. For the SENN explanations to be useful, the basis concepts need to be directly interpretable by humans. While interpretability is still not well defined, the authors propose three desiderata for interpretable concepts:

(1) **Fidelity**: the representation of x in terms of concepts should preserve relevant information,
(2) **Diversity**: inputs should be representable with few non-overlapping concepts, and
(3) **Grounding**: concepts should have an immediate human-understandable interpretation

These conditions are enforced on the concepts by: (1) learning h as the latent encoding of an autoencoder (2) making said autoencoder sparse and (3) providing interpretations for concepts by prototyping [Li et al. 2017], e.g. by. finding a set of observations that maximally activate a certain concept. An autoencoder learns to map an input x to a lower dimensional information preserving latent representation, or encoding, h. A *sparse* autoencoder is one in which only a relatively small subset of the latent dimensions activate for any given input.

2.4 Implementation

Although a public implementation of SENN *is* available [1], the authors have not officially released that code with the paper. There also seems to be a major bug in this code: the concept loss $\mathcal{L}_h$ is not used. Therefore, the framework is re-implemented with the original paper as ground truth.

On a high level, the SENN model consists of three main building blocks: a parameterizer, a conceptizer, and an aggregator. The **parameterizer** θ is actualized by a neural network, and the **conceptizer** is actualized by an autoencoder. The specific implementations of these networks may vary. For tabular data, we use fully connected networks. For image data, we use convolutional networks [LeCun et al. 1998]. Like the authors, we use a sum operator for the **aggregator** g for all experiments discussed here. An overview of the model can be found in Figure 1.

[1] https://github.com/dmelis/SENN

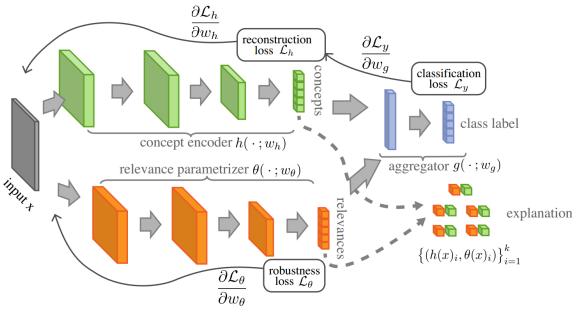


Figure 1: SENN architecture [Alvarez Melis and Jaakkola 2018].

To train the model, the authors minimize the loss function

$$\mathcal{L} := \mathcal{L}_y(f(x), y) + \lambda \mathcal{L}_\theta(f(x)) + \xi \mathcal{L}_h(x), \tag{2}$$

where

- $\mathcal{L}_y(f(x), y)$ is the **classification loss**, i.e. how well the model predicts the ground truth label.
- $\mathcal{L}_\theta(f(x))$ is the **robustness loss** given by Equation (1). λ is a regularization parameter controlling how heavily robustness is enforced.
- $\mathcal{L}_h(x)$ is the **concept loss**. The concept loss is a sum of 2 different losses: **reconstruction loss** and **sparsity loss**. ξ is a regularization parameter on the concept loss.

The authors don't specify which classification loss, reconstruction loss, and sparsity loss they specifically use. We implement the classification loss with the cross entropy loss, the reconstruction loss with a mean squared error loss, and the ℓ_1-norm of the concept activations $h(x)$ as the sparsity loss. For the robustness loss, the authors don't specify which matrix norm should be used. We use the Frobenius norm, which returns the sum of squared elements of a matrix. Finally, the authors confusingly refer to the concept regularization loss hyperparameter ξ, which is present in Equation (2), as the sparsity strength parameter. We choose to interpret it as the latter.

2.5 Extensions

Besides studying the reproducibility of the SENN framework, we also propose several extensions. The first extension deals with *explicitness*: we improve the interpretability of explanations by creating more interpretable concepts. If the learned concepts cannot be interpreted, the SENN explanations are nearly meaningless. For this reason, we aim to improve the *diversity* and *grounding* of the concepts.

The authors represent concepts with prototypes. While multiple prototyping methods are proposed, the authors delegate some to future work and only explore the following method: *A single concept is represented by a set of data samples that maximizes that concept's activation.* The authors reason that we can read off what the concept represents by examining the feature shared by these maximizing samples. Although this approach might seem reasonable at first glance, it has some major shortcomings.

Firstly, only showcasing the samples that maximize an encoder's activation is quite arbitrary. One could just as well showcase samples that *minimize* the activation instead, or use any other method. These different approaches lead to different interpretations even if the learned concepts are the same. The authors also hypothesize that a sparse autoencoder will lead to *diverse* concepts. However, enforcing only a subset of dimensions to activate for an input does not explicitly enforce that these concepts should be non-overlapping. Additionally, each single concept may represent a mix of features that are entangled in some complex way. This greatly increases the complexity of interpreting any concept on its own.

Disentangling SENN (DiSENN). To enhance concept interpretability, we therefore propose to explicitly enforce *disentangling the factors of variation* in the data and using these as concepts instead. Matching a single generative factor to a single latent dimension allows for easier human interpretation [Bengio et al. 2014], while additionally enforcing concepts to be non-overlapping. More abstractly, a disentangled representation may be viewed as a concise representation of the variation in data we care about most – the generative factors.

We enforce disentanglement by using a β-VAE [Higgins et al. 2016], a variant of the Variational Autoencoder (VAE) [Kingma and Welling 2014], as conceptizer. β-VAE introduces a hyperparameter β that weights the KL-divergence term in the VAE objectve. For larger β, the latent space will be encouraged to look like a unit Gaussian, so that the dimensions are encouraged to be independent.

We call a disentangled SENN model with β-VAE as the conceptizer a **DiSENN** model. Let an input x produce the Gaussian encoding distribution for a single concept $h(x)_i = \mathcal{N}(\mu_i, \sigma_i)$. The concept's activation for this input is then given by μ_i. We then vary a single latent dimension's values around μ_i while keeping the others fixed, call it μ_c. If the concepts are disentangled, a single concept should encode only a single generative factor of the data. The changes in the reconstructions $h_{\text{dec}}(\mu_c)$ will show which generative factor that latent dimension represents. We plot these changes in the reconstructed input space to visualize this. μ_c is sampled linearly in the interval $[\mu_i - q, \mu_i + q]$, where q is some quantile of $h(x)_i$.

Robustness Study. The authors enforce the *stability* desideratum of explanations by minimizing the robustness loss. They then test stability by adding Gaussian noise to an input and examining the perturbation in the obtained explanations. However, adding Gaussian noise does not correlate highly with a change in human interpretation of similarity. We are actually only interested in how robust explanations are with respect to images that are perceived to be similar *by humans*. Our contribution is to study exactly that behavior. We find qualitatively similar MNIST images and compare the robustness of their explanations.

Concept Visualization. We implement a prototyping method that the authors delegate to future work. A concept is represented by samples that maximize a concept's activation *and* minimize the other concepts' activations. We qualitatively analyze the obtained prototypes.

3 EXPERIMENTAL SETUP

In our experiments, we use the MNIST digit recognition image dataset and Propublica's COMPAS Recidivism Risk Score dataset[2] (Compas) tabular dataset. The authors' description of their preprocessing of the Compas dataset is very incomplete. Full details about this can be found in Appendix A.1). The specific SENN architectures used for these datasets can be found in Table A.3.

3.1 Reproducibility Experiments

An important requirement for the SENN framework is that it should be capable of achieving **high performance**, which is what the authors claim. This is therefore the first thing we try to reproduce. We train SENN models on the previously mentioned datasets and compare our performance to the authors' reported performance.

We evaluate the **quality of explanations** generated by the framework on random images from the MNIST test set. Evaluation is done by trying to reason about the interpretability of the learned concepts and their relevance scores. For evaluation, concepts are represented by maximally activated prototypes, like the authors do.

We also investigate the **trade-off between accuracy and robustness**. We train models with different values of the robustness loss regularization parameter λ and report the achieved accuracies. All the experiments mentioned in this section use the same hyperparameters and architecture that were reported by the authors. The specific hyperparameters can be found in Table 1 in the appendix.

3.2 Extension Experiments

Our most important extension is the **disentangling of the learned concepts**. As discussed in Section 2, DiSENN's concept submodel is actualized by a β-VAE (refer to A.3 for the architecture details). During training, we set $\beta = 4$, since this is reported to generally work well [Higgins et al. 2016]. There is a trade-off in how well the latent space is disentangled and how well the input is reconstructed [Burgess et al. 2018]. Therefore, we pre-train the β-VAE with $\beta = 1$, which reduces it to a standard VAE. This preserves fidelity by having the network first learn how to reconstruct the input well, before encouraging disentanglement during the training process.

In the **robustness study**, we need to find examples that are semantically similar. We do not do this by hand, but cluster images based on the Euclidean distance in the latent space of an autoencoder trained on MNIST. Clustering is done using K-nearest-neighbors.

4 RESULTS

4.1 Reproducibility Experiments

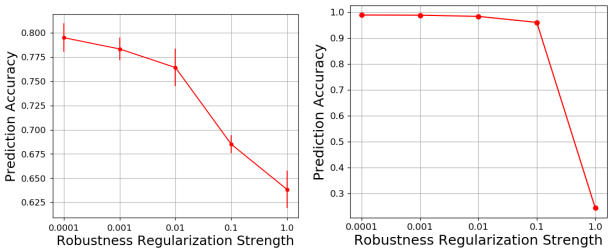
We first report the **performance** our models attain. On MNIST, the highest test set accuracy we reach is 98.9%, while the authors report 99.1% to 98.7% for different robustness regularization settings. On the Compas dataset, our model reaches a final accuracy of 80.9% compared to the 82% reported in the original paper. These results seem to match up reasonably well. We deem classification performance to be reproducible.

To evaluate the **quality of explanations**, we plot MNIST model explanations as in Figure 2, which displays 2 of these. The upper

[2] http://github.com/propublica/compas-analysis/

explanation, for digit eight, is arguably interpretable if the authors' logic is adopted. It can be reasoned about in the following way. Concepts 2 and 4 have high relevance scores. This makes sense as concept 2 seems to encode "digit 8" and a diagonal stroke, which is present in the image. Concept 4 seems to encode "digit 3", which is visually similar to the right half the input image. All other concepts have negative relevances, which matches the authors' interpretation that their prototypes are dissimilar from the test example. However, this example was not randomly sampled but *selected by hand* for the purpose of this demonstration. In general, the produced explanations look more like the second example, for which no reasonable explanation can be found. Similar explanations for the Compas dataset are shown in the Appendix (Figure 5).

We now examine the **trade-off between accuracy and robustness**. In Figure 3, we show the change in test accuracy as robustness regularization increases. We see that, as expected, accuracy drops for increasing regularization. We compare these results to the authors' in Section 5.



(a) Compas (b) MNIST

Figure 3: Change in test accuracy for increasing robustness regularization (λ). Error bars represent the standard deviation over three different seeds.

4.2 Extensions

DiSENN. We now examine the DiSENN explanations by analyzing a generated DiSENN explanation for the digit 7 in Figure 4. The contribution of concept i to the prediction of a class c is given by the product of the corresponding relevance and concept activation $\theta_{ic} \cdot h_i$.

First, we look at how the concept prototypes are interpreted. To see what a concept encodes, we observe the changes in the prototypes from left to right. Taking the second row as an example, we see a circular blob slowly disconnect at the left corner to form a 7, and then morph into a diagonal stroke. This explains the characteristic diagonal stroke of a 7 connected with the horizontal stroke at the right top corner but disconnected otherwise. As expected, this concept has a positive contribution to the prediction for the real class "digit 7" and a negative contribution to that of another incorrect class, "digit 5".

Robustness Study. If the robustness regularization achieves its intended effect, perceptually similar inputs should lead to similar explanations. We generate explanations for semantically similar and find that the explanations are indeed robust. For detailed results, we refer to Figure 6 in the appendix.

5 DISCUSSION

Reproducibility. The results presented in Section 4.1 show that the authors' results are partly reproducible. Specifically, we are able to achieve similar test accuracies, which seems to validate the authors' claim that SENN models have high modelling capacity. Even though this is the case, the MNIST and Compas datasets are of low complexity. The relatively high performance on these datasets is therefore not sufficient to show that SENN models are on par with non-explaining state-of-the-art models.

We now look at the reproducibility of the experiment that compares the *impact of the robustness regularization on accuracy*. Figure 3 shows that regularizing more decreases classification accuracy. This behavior matches up with the authors' findings. However, we find the accuracy drop on the Compas dataset for increasing regularization to be **significantly larger** than reported by the authors. On the MNIST dataset, they report an accuracy of 98.7% for regularization strength $\lambda = 1$, whereas accuracy drops critically to 40% in our experiments. We conclude that this experiment is **not reproducible**.

Assessing the **quality of explanations** is inherently subjective. It is therefore difficult to link the quality of explanations to reproducibility. This difficulty is exacerbated when it is not clear what generative factor a concept represents. However, we can still partially judge reproducibility by qualitative analysis. In Figure 2, we showcase one example that can be explained quite well and one that cannot. In general, we see that finding an example whose explanation makes sense is difficult and that such an example is **not representative of the generated examples**. Therefore, we conclude that obtaining good explanations is, in that sense, not reproducible.

Section 2.4 outlines **other factors that impede reproducibility**: classification, reconstruction, and sparsity losses are not specified; the matrix norm used in Equation (1) is left unspecified; and the value of the concept loss hyperparameter ξ in Equation (2) is ambiguous.

Beyond reproducibility, we also find that the framework is **difficult to extend to domains beyond those mentioned in the paper**. With current deep learning frameworks, the implementation of the Jacobian computation, and hence the robustness loss, does not generalize to arbitrary data formats. One would have to manually adapt its implementation any time they want to use a SENN for a new data format.

DiSENN. DiSENN provides a principled way to generate prototypes, since interpolation in the latent space of a VAE can be done meaningfully. Another advantage is that prototypes are not constrained to the input domain, since the VAE is a generative model. The prototypes generated by the DiSENN are more *complete* than highest activation prototypes, since they showcase a much larger portion of a concept dimension's latent space. Seeing the transitions in concept space provides a more intuitive idea of what the concept means.

However, despite the hope that disentanglement encourages diversity, we observe that concepts still demonstrate overlap. This can be seen from concepts 1 and 2 in Figure 4. This means that the concepts are still not disentangled enough, and the problem of interpretability, although alleviated, remains. The progress of

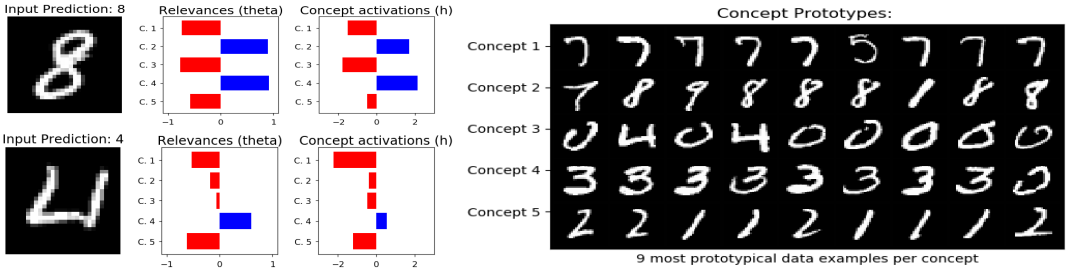


Figure 2: Two MNIST test observations (left); their explanation, consisting of relevance scores and concept activations (middle); and the corresponding concept prototypes (right).

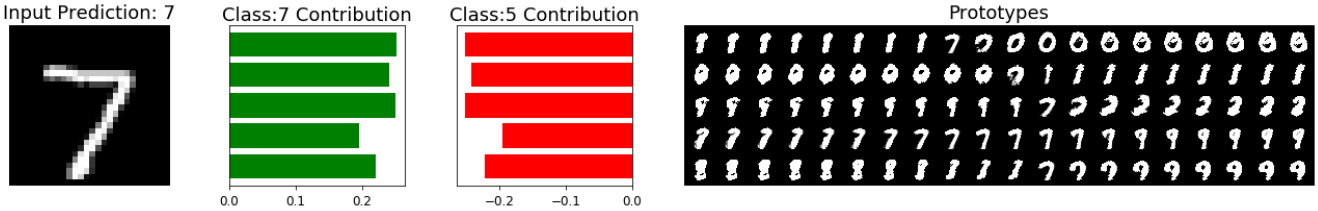


Figure 4: DiSENN test sample (left). The concept contributions for the real class (7) and the concept contributions for another class (5) for the same input (middle). The concept contributions are defined as the product $\theta_c \cdot h$ where c is the class. The plot on the right depicts the concept prototypes.

good explanations therefore depends on the progress of research in disentanglement.

6 BROADER IMPLICATIONS

As we have seen, SENNs are *designed* and trained *during learning* to give explanations for particular predictions. This approach differs from much work done previously, which focus on finding explanations *after* a model has been trained. Since the explanations are given for single predictions and not for the model as a whole, they are called local explanations.

A well-known method for producing local explanations is the use of *surrogate models*. In this paradigm a simpler "interpretable" model is fit to the predictions of the black box model we attempt to explain. The simpler surrogate model is then used as a proxy to draw conclusions about the black box model. This can be done "globally" for the whole model, or "locally" on individual predictions. [Ribeiro et al. 2016a,b], the first to propose local surrogate models, use a sparse linear model as their interpretable model. The coefficients of the linear model are then used as feature importances. They call this method LIME.

[Lipovetsky and Conklin 2001] propose a game theoretic approach to determine how much each feature contributes to the overall prediction, which can be measured by the *Shapley value*. [Lundberg and Lee 2017] propose ways to efficiently estimate the Shapley value for linear models, tree-based models, and deep models.

[Sundararajan et al. 2017] describe axioms that any explanation method should satisfy, and derive the *Integrated Gradients* explanation method based on these axioms.

In contrast to SENN, the aforementioned methods all focus on post-hoc explanations. Another important distinction between previously mentioned approaches and SENN is the usage of concepts. The aforementioned methods use the raw input features along with their importances as explanations. The authors argue that raw features (such as individual pixels in images) tend to be hard to analyze coherently for high-dimensional inputs, often leading to unstable explanations. Concept learning can therefore be seen as an advantage of SENN, but only on the presupposition that the concepts themselves are immediately interpretable. As we have seen in Section 5, this is not necessarily the case.

A big assumption of the SENN framework is that linear models are interpretable. This claim's validity, however, depends on the notion of interpretability that is used [Lipton 2017].

7 CONCLUSION

In this work, we examine the reproducibility of [Alvarez Melis and Jaakkola 2018], who propose SENN, a framework for models that are both highly performant and intrinsically capable of explaining their predictions. We find that the accuracies reported by the authors can be reached. However, for a higher robustness regularization parameter, our accuracies drop significantly more than reported by the authors. We also see that finding an example whose explanation makes sense is difficult, and that the framework is difficult to extend to domains beyond those mentioned in the paper. We propose an extension that alleviates the problem of finding good explanations slightly by providing a principled approach for concept prototyping. Due to the limited reproducibility we do not award the paper any ACM badge.

REFERENCES

David Alvarez Melis and Tommi Jaakkola. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. *Advances in Neural Information Processing Systems 31* (2018), 7775–7784. http://papers.nips.cc/paper/8003-towards-robust-interpretability-with-self-explaining-neural-networks.pdf

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete Problems in AI Safety. *arXiv:1606.06565 [cs]* (July 2016). http://arxiv.org/abs/1606.06565 arXiv: 1606.06565.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2014. Representation Learning: A Review and New Perspectives. *arXiv:1206.5538 [cs]* (April 2014). http://arxiv.org/abs/1206.5538 arXiv: 1206.5538.

Nick Bostrom and Eliezer Yudkowsky. 2014. The ethics of artificial intelligence. In *The Cambridge Handbook of Artificial Intelligence*. Cambridge University Press, Cambridge, 316–334. https://doi.org/10.1017/CBO9781139046855.020

Christopher P. Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. 2018. Understanding disentangling in β-VAE. *arXiv:1804.03599 [cs, stat]* (April 2018). http://arxiv.org/abs/1804.03599 arXiv: 1804.03599.

Finale Doshi-Velez and Been Kim. 2017. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv:1702.08608 [cs, stat]* (March 2017). http://arxiv.org/abs/1702.08608 arXiv: 1702.08608.

Bryce Goodman and Seth Flaxman. 2017. European Union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine* 38, 3 (Oct. 2017), 50–57. https://doi.org/10.1609/aimag.v38i3.2741 arXiv: 1606.08813.

Moritz Hardt, Eric Price, Eric Price, and Nati Srebro. 2016. Equality of Opportunity in Supervised Learning. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 3315–3323. http://papers.nips.cc/paper/6374-equality-of-opportunity-in-supervised-learning.pdf

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. (Nov. 2016). https://openreview.net/forum?id=Sy2fzU9gl

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. 2018. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). *International Conference on Machine Learning (ICML)* (June 2018). http://arxiv.org/abs/1711.11279 arXiv: 1711.11279.

Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]* (May 2014). http://arxiv.org/abs/1312.6114 arXiv: 1312.6114.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. 2017. Deep Learning for Case-Based Reasoning through Prototypes: A Neural Network that Explains Its Predictions. *arXiv:1710.04806 [cs, stat]* (Nov. 2017). http://arxiv.org/abs/1710.04806 arXiv: 1710.04806.

Stan Lipovetsky and Michael Conklin. 2001. Analysis of Regression in Game Theory Approach. *Applied Stochastic Models in Business and Industry* 17 (Oct. 2001), 319–330. https://doi.org/10.1002/asmb.446

Zachary C. Lipton. 2017. The Mythos of Model Interpretability. *arXiv:1606.03490 [cs, stat]* (March 2017). http://arxiv.org/abs/1606.03490 arXiv: 1606.03490.

Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems 30* (2017), 4765–4774. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

Clemens Otte. 2013. Safe and Interpretable Machine Learning: A Methodological Review. In *Computational Intelligence in Intelligent Data Analysis (Studies in Computational Intelligence)*, Christian Moewes and Andreas Nürnberger (Eds.). Springer, Berlin, Heidelberg, 111–122. https://doi.org/10.1007/978-3-642-32378-2_8

Marco Ribeiro, Sameer Singh, and Carlos Guestrin. 2016a. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. 97–101. https://doi.org/10.18653/v1/N16-3020

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016b. Model-Agnostic Interpretability of Machine Learning. *arXiv:1606.05386 [cs, stat]* (June 2016). http://arxiv.org/abs/1606.05386 arXiv: 1606.05386.

Salvatore Ruggieri, Dino Pedreschi, and Franco Turini. 2010. Data mining for discrimination discovery. *ACM Transactions on Knowledge Discovery from Data* 4, 2 (May 2010), 1–40. https://doi.org/10.1145/1754428.1754432

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. *arXiv:1703.01365 [cs]* (June 2017). http://arxiv.org/abs/1703.01365 arXiv: 1703.01365.

Kush R. Varshney and Homa Alemzadeh. 2017. On the Safety of Machine Learning: Cyber-Physical Systems, Decision Sciences, and Data Products. *arXiv:1610.01256 [cs, stat]* (Aug. 2017). http://arxiv.org/abs/1610.01256 arXiv: 1610.01256.

A APPENDIX

A.1 Data Preprocessing

We use the original MNIST with standard mean and variance normalization. Additionally, we separate 10% of the training data for validation.

The Compas dataset resides in a GitHub repository and needs to be handled more carefully. Unfortunately, multiple datasets exist in the repository and the name of the file, used for the experiments in the original paper, is not explicitly specified. For this reason, we use the version, used by the public implementation[3], where the preprocessing is already done. Furthermore, the authors suggest that as part of the preprocessing task they removed inconsistent examples, *whose label differs from a strong (80%) majority of other identical examples*. However, they do not elaborate any further on the exact approach so we used the same preprocessing code found in the public implementation. We use splits of 80%, 10%, 10% for training/validation/testing.

A.2 Hyperparameters

Table 1: Hyperparameters SENN

	Compas	MNIST
Number of epochs trained	100	5
Number of concepts	11 (input dimension)	5
Learning rate	2×10^{-4}	2×10^{-4}
Concept Regularization	0	1
Sparsity Regularization (ξ)	2×10^{-5}	2×10^{-5}
Robustness Regularization (λ)	1×10^{-4}	1×10^{-4}
Dropout Rate	0.0	0.5

Table 2: Hyperparameters DiSENN

	MNIST
Number of concepts	5
Learning rate	2×10^{-4}
Concept Regularization	1
Sparsity Regularization (ξ)	2×10^{-5}
Robustness Regularization (λ)	1×10^{-4}
Dropout Rate	0.5
β	4

A.3 Architectures

In Table 3. the layers are defined as follows: FC $\rightarrow$ Fully-connected + ReLU + Dropout, CL $\rightarrow$ Convolutional + ReLU, UP $\rightarrow$ Transposed Convolution + ReLU. Note that the last layer of the conceptizer‘s encoder does not have an activation while the last layer of the conceptizer‘s decoder and the parameterizer, the ReLU activation is replaced with a Tanh.

[3] https://github.com/adebayoj/fairml/raw/master/doc/example_notebooks/propublica_data_for_fairml.csv

Table 3: SENN Architectures (A.3 for explanations)

	Compas	MNIST
$h(x)_{encoder}$	$h(x)_{encoder}$ = x	CL((5x5, 10))
$h(x)_{decoder}$	$h(x)_{decoder}$ = x	UP((5x5,16),(5x5,8),(2x2,1))
$\theta(\cdot)$	FC(10,5,5,10)	Conv((5x5,10),(5x5,20)) $\rightarrow$ FC(320,50)
$g(\cdot)$	$\theta(x)^T h(x)$	$\theta(x)^T h(x)$

Table 4: DiSENN Architecture (A.3 for explanations)

	MNIST
$h(x)_{encoder}$	FC(512, 256, 100, 5)
$h(x)_{decoder}$	FC(5, 100, 256, 512, 784)
$\theta(\cdot)$	Conv((5x5,10),(5x5,20)) $\rightarrow$ FC(320,50)
$g(\cdot)$	$\theta(x)^T h(x)$

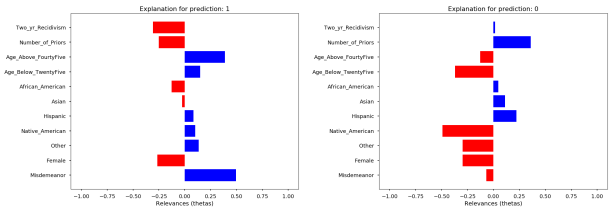


Figure 5: One negative and one positive example of an explanation produced for the Compas dataset.

A.4 Robustness Study

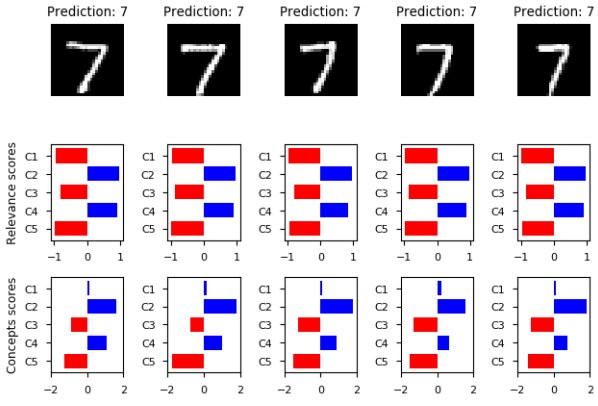
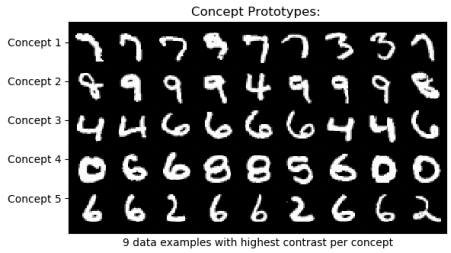


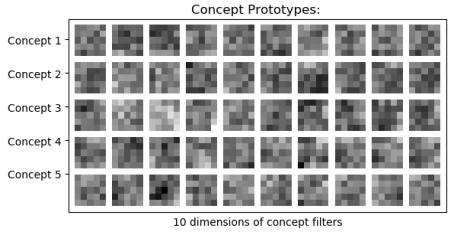
Figure 6: Perceptual similar test examples (top), their relevances (middle), and concept activations (bottom).

A.5 Concept Representations

The results of the reproducibility experiments revealed that there is potential for improvement in the concept representation. The extent to which the explanations are human interpretable given the concept representation is very limited. The authors propose two alternative methods to be explored in future work. One of them is to select the prototypes of the concept not only by the highest activation but also by optimizing for little activation of all remaining concepts at the same time. We call this method *highest contrast*. The other approach visualizes the filters of the last layer of the convolutional encoder that correspond to each concept. This is a reasonable choice as deeper layers represent higher layer features. Figure 7 shows the obtained concept representations with these methods applied to the same SENN model used for the analysis of explanations (Figure 2). The prototypes selected by highest contrast seem



(a) Highest Contrast



(b) Filter

Figure 7: Alternative concept representations. Highest contrast and filter visualisation.

to have a higher entropy of digit classes within one concept compared to the prototypes by highest activation. The interpretation of concepts is changed when using this method for concept representation instead of the standard approach. For example, concept 4 was previously most activated by digit three prototypes. However, with highest contrast, the concept is not represented by prototypes of a certain digit class. This inconsistency in interpretability of the concepts when using different visualization methods raises doubts that the way of representing the concepts by prototypes of the dataset is meaningful at all. The second method of visualizing the filters associated with one concept does not show human interpretable results on MNIST. The patterns in the filters do not uncover any clear features.

A.6 Who did what?

We collaborated effectively to create the model training infrastructure, and everyone contributed equally. Additionally, Ivan worked on reproducing the robustness study, Christoph worked on alternative visualization techniques, Aman worked on the β-VAE implementation while Omar focused on putting the whole report together and discussing our findings.