1 Prerequisites on Learning Disentanglement

1.1 Learning from Data

Learning from data is commonly understood as the ability of algorithms to improve their performance on a task with experience accumulated from the observation of data [1]. The source of data is usually a dataset - set of data points $X = \{x_i | i \in \{1 \dots n\}\}$, which are sampled from a probability distribution $x_i \sim p(x)$. In general, these data points are multi-dimensional. In computer vision in particular, data are images $\mathbf{x}$ with height h and width w, so that the data points are $\mathbf{x} \in \mathbb{R}^{h \times w}$.

1.1.1 Supervised

The term supervised learning denotes the task to learn a mapping from data points x_i to target labels y_i. A supervised algorithm has access to data-label pairs $(y_i, x_i) \sim p(y, x)$, in order to estimate the connection between data points and labels, either in form of a conditional probability $p(y|x)$, or in form of a deterministic function $y = f(x)$. The label y can be either discrete (*e.g.* information about an object class) or continuous (*e.g.* the location of an object part in an image). Recent advances, in particular the effectiveness of neural network models (cf. Sec. 1.1.3) on big datasets, have led to huge progress on problems that can be formulated as regression or classification. That is why on many traditional computer vision problems, such as object recognition, image classification or human pose estimation, machines are now performing on a superhuman level; hence, these problems are now considered to be essentially solved.
The Achilles' heel of supervised learning lies in the need for a viable supervision signal. To get labels, it is usually required to manually annotate the data. The human effort in this is costly, error-prone and not scalable to the ever-growing vast amounts of raw data.

1.1.2 Unsupervised

Unsupervised learning is the endeavour to learn about structures and patterns in unlabelled data. In this paradigm, the learning algorithm has access to the samples of the data distribution $x \sim p(x)$. The task is usually framed as a form of density estimation, *i.e.* to model the entire distribution in a probabilistic generative model (cf. Sec. 1.2). Unsupervised learning is considered much harder than supervised learning [2]. There are several complications in the design of unsupervised algorithms:

- Naturally, without supervision, *the goal of learning is not specified*, hence surrogate objectives have to be formulated. The lack of specification renders the evaluation oftentimes arbitrary and subjective [3].
- It is a priori not clear, *how much prior knowledge* should be embedded. To introduce no artificial bias, some argue for a purely data-driven approach. Others argue for the importance of certain inductive priors to guide learning [4]. A related modeling choice is, whether the algorithm should be model-free or model-based. In this work we argue for using more prior knowledge and modelling assumptions to obtain strong constraints.
- Lastly, the *definition of the term unsupervised* itself is subject to discussion. What entitles an algorithm to be called unsupervised? While the definition itself has no practical importance, unclear and imprecise terminology unnecessarily confuses. Here, unsupervised learning shall mean to use no label information for the dataset samples, but assuming a model or inductive priors shall be fine. This is indeed necessary to enable unsupervised learning of disentanglement at all, as we will see in Sec. 1.4.

1.1.3 Artificial Neural Networks

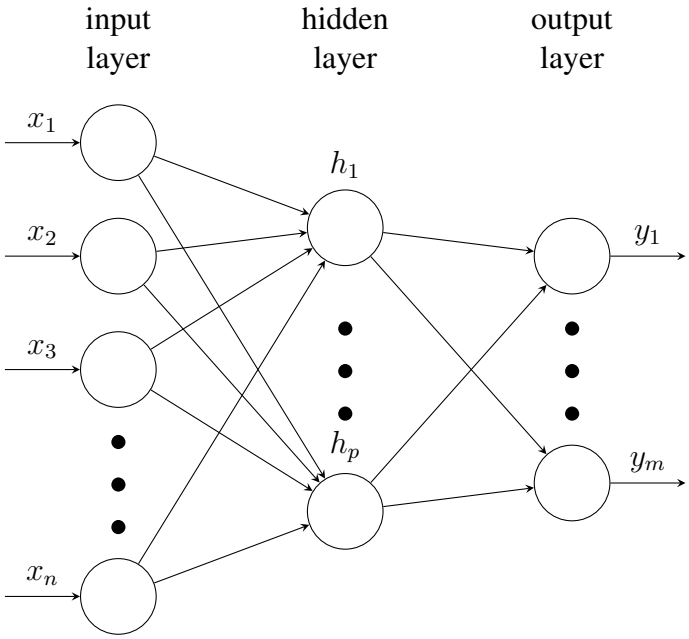

Figure 1.1: Sketch of a one-hidden-layer artificial neural network model: input $x = \{x_i | i = 1 \dots n\}$ and output $y = \{y_j | j = 1 \dots m\}$ are connected through a hidden layer $h = \{h_k | j = 1 \dots p\}$

Artificial neural networks are a powerful and flexible tool for function approximation. Inspired by biological neurons, there have been numerous questionable claims w.r.t. their

biological plausibility. Here, we will treat an artificial network solely as a parametric non-linear function approximator. They can approximate a function $y = f(x)$ with vector input $x = \{x_i | i = 1 \dots n\}$ and vector output $y = \{y_j | j = 1 \dots m\}$, also see Fig. 1.1. At minimum they connect the input and the output through one hidden layer $h = \{h_k | j = 1 \dots p\}$ by:

$$\begin{aligned} h_j &= a(\sum_i w_{ji} x_i + w_{0i}) \\ y_j &= a'(\sum_i w'_{ji} h_i + w'_{0i}), \end{aligned} \tag{1.1}$$

with weight matrices w, w', non-linear so-called activation functions a, a' and bias vectors w_0, w'_0. Neural networks can also comprise multiple hidden layers connect by $h_j = a(\sum_i w_{ji} h_i + w_{0i})$. It can be shown, that in the limit of infinite hidden units h_j a one-hidden-layer network is enough to approximate any (continuous) function arbitrarily close [5, 6]. In practice, however, networks with more that one layer, referred to as deep neural networks, seem to work better. This may be due to the possibility of building a hierarchical feature representation [7], that reflects the hierarchical nature of the physical reality. Typical activation functions are for example the sigmoid function (a_{sigm} or the rectified linear unit (a_{relu}):

$$a_{\text{sigm}}(x) = \frac{1}{1 + e^{-x}} \tag{1.2}$$

$$a_{\text{relu}}(x) = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases} \tag{1.3}$$

The activation function needs to be non-linear, otherwise the neural network is just a linear classifier (matrix multiplies are again matrices). For processing image data, the weight matrices can be constrained to be only locally connected and to share weights across locations to enforce translation invariance, resulting in *convolutional* neural networks.

Deep neural networks have highly non-convex likelihood functions, hence for optimization iterative numerical methods are used: The weights w are initialized to some initial value $w^0$and then updated at time step t with an update rule $w^{t+1} \rightarrow w^t$. A simple yet successful rule is given by gradient descent,

$$w^{t+1} = w^t + \lambda \nabla_{w^t} \mathcal{L}(w^t), \tag{1.4}$$

where λ is the learning rate, parametrizing the step size. In practice, calculating derivatives of the likelihood w.r.t. the weights can be done efficiently via error backpropagation. For big datasets it becomes cumbersome to calculate the gradient w.r.t the whole dataset. Taking only a random subset of the data for an approximation of the gradient, renders the optimization stochastic; the procedure is then called stochastic gradient descent.

1.2 Generative Models

What I cannot create, I do not understand. - R. Feynman

Learning and understanding structure in data by being able to generate, is the rationale behind generative modelling. Generative models are mostly applied for unsupervised learning and can be contrasted to discriminative models. While discriminative models are used to model posterior conditionals $p(y|x)$ (*e.g.* for supervised learning (cf. Sec. 1.1.1), generative models capture the complete data distribution $p(x)$ in an estimate $\hat{p}(x)$ [2]. Thus, after estimation, one can generate samples from this model $\hat{p}$. Hence the name generative model. The currently predominant generative models are built on either autoencoding or adversarial formulations:

1.2.1 Autoencoding Formulations

An autoencoding model is learning by reconstructing samples of data, $\hat{x} = f(x)$. To enforce data compression (otherwise the identity function is a trivial solution of autoencoding) the function has an information bottleneck, namely an inferred latent code z of reduced dimension. The autoencoder is then the chain of an encoding function $z = e(x)$ and a decoding function $\hat{x} = d(z) = d(e(x))$.

Whereas the conventional autoencoder consists of deterministic mappings e, d, the variational autoencoder [8] models the probability distribution $p(x)$. More specifically, it maximizes a lower bound to the logarithmic likelihood $\log p(x)$ of data x. This so-called variational lower bound $\mathcal{L}$ is given by:

$$\mathcal{L} = \mathbb{E}_{z\sim q(z|x)} \log p(x|z) - \mathbb{E}_{z\sim q(z|x)} \log \frac{q(z|x)}{p(z)} \tag{1.5}$$

Where z introduces latent variables, with a prior distribution $p(z)$, with an approximation to the posterior $q(z|x)$ of the latent variables, and the posterior of the data given the latent variables $p(x|z)$. If one wants to model the distributions with neural networks, one typically uses Gaussian distributions and lets the networks predict the parameters (mean μ and variance Σ) based on the image. In the current machine learning contexts, all functions (e, d) and moments (μ, Σ) are modelled with neural networks.

1.2.2 Adversarial Formulations

Generative adversarial networks (GAN) [9] consist of two neural networks competing in a zero-sum game. A generator network G is generating images based on a latent code z sampled from a distribution $p(z)$. The discriminator network D is a binary classifier with the task to classify an image as originating from the data distribution p_{data} or from the distribution produced by G. The loss function of G is the negative of the loss of D, such that one can formulate the optimization in a minmax form:

$$\min_D \max_G -\frac{1}{2}\mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] - \frac{1}{2}\mathbb{E}_{z\sim p(z)}[\log(1 - D(G(z)))] \tag{1.6}$$

The generator is then optimized to make the output indiscriminable from the data distribution. The discriminator can be interpreted as a learned similarity metric, to measure the closeness of an image to the data distribution [10]. There are many variants and extensions to this basic principle of learning with an adversarial task. For example, one can learn a discriminator for a set of image patches [11].

1.3 Disentangling Representations

In supervised learning, a performance measure is naturally induced by the metric, that is being optimized. In the unsupervised setting, judging the performance of a model is less straightforward. How to rate the quality of the latent representation?

1.3.1 Learning Representations

> Disentangle as many factors as possible, discarding as little information about the data as is practical. - Bengio *et al*. [12]

According to Bengio *et al*. [12], a representation is useful, if it can be applied to many - in advance unknown - different tasks, while being trained on only one particular task. As the downstream tasks can be multifarious, the essential *information* should be contained in the representation. For some tasks only a subset of aspects of the data will be necessary, that is why *disentangled factors* make a representation particularly practical.

The latent representation z learned by generative models captures the essential *information* of the data distribution. That is made sure by requiring the ability to generate samples from the original data distribution from it. How then to reach the second goal, the *disentanglement* of generative factors?

1.3.2 Disentangling defined by Equivariance and Invariance

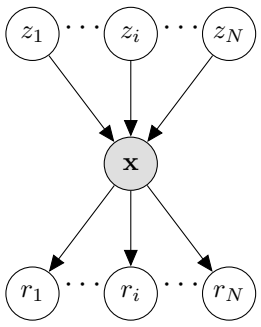

Figure 1.2: Disentangling causal factors means to infer an estimate - *i.e*. a representation - from an image

What is a factor? As outlined in the introduction (cf. Sec. **??**), factors in a representation should correspond to causal elements of the world. In general, these factors can

interact in complicated ways to finally result in an image. Here, we only consider the case where multiple independent factors each have an influence (cf. Fig. 1.2):

$$p(z_1 \dots z_N) = \prod_i p(z_i) \tag{1.7}$$

A change in an element, should then lead to: *i)* a corresponding change in the representational factor and *ii)* leave other factors, that represent other elements, unchanged. Formally, this can be seen as inference: a number of latent variables $z_1 \dots z_N$ interacted to cause the existence of the observed image $\mathbf{x}$. The task is now to infer estimates for these latent variables $r(\mathbf{x})_i := r_i$. A graphical model of the process is shown in Fig. 1.2. A disentangled representation should simultaneously fulfill equivariance and invariance: A change in z_i should: *i) equivariantly* change in the abstract representational factor r_i, *ii)* while leaving the other factors $r_j, j \neq i$, that represent other causes, *invariant*.

1.4 Theoretical Impediments from Causality

Generative factors represent causal elements. Learning a disentangled representation of generative factors is then understood as causal inference. In accordance with the causal literature [13], we can make statements about the type of knowledge, that can be gained by the type of data provided. It turns out that from "raw" image data, it is actually impossible to learn a disentangled representation z - raw data referring to images x sampled from $p(x)$, without further assumptions. To elucidate this fact, we start with a primer for causal learning (Sec. 1.4.1), outline which inductive biases are needed for disentanglement (Sec. 1.4.2) and assess how one can instantiate such biases for disentangling the factors of shape and appearance in images (Sec. **??**, Sec. **??**)).

1.4.1 Causal Learning

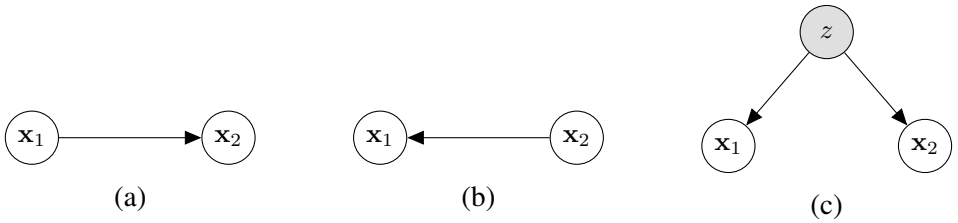

Figure 1.3: Correlation implies causation - if x_1 and x_2 correlate, a) x_1 may cause x_2, b) x_1 may be caused by x_2 or c) both are contingent on a latent cause z

Learning to infer causality is harder than statistical learning. We outline the basic problem for the case of two variables x_1, x_2: statistical learning aims at estimating probabilistic properties such as $p(x_1, x_2)$ or $p(x_2|x_1)$ from data. It is a well-known theme in statistics is that correlation does not imply causation. Less well-known is Reichenbachs principle [14, 15], that states: if two random variables are statistically dependent, then

there exists a third variable that influences both or a direct causal link between them (Fig. 1.3). In addition to estimating the probability distribution, also the causal structure has to be inferred [14].

To show the limitations of raw data, we sketch an intuitive example problem (adapted from [16]): How to learn the causal connection between a barometer and the weather? If the barometer is working well, there exists a clear correlation between the weather condition and the needle position. Given a dataset showing both barometer and corresponding weather condition, a capable machine learning algorithm will be able to capture this correlation. However, it will fail to understand the causal direction, since this is not possible from the data. Imagine how a human would go about solving this problem: Having a mechanistic model of the world he could reason about the precise causal mechanism relating weather to air pressure to needle position. A simple model could be: weather influences air pressure, pressure influences barometer needle position. What if one has no prior knowledge? A solution of child-level simplicity is, to force the needle to move with a finger. Without the power of magic, the weather will not change. Hence causality has to go other way or via a third latent variable influencing both *i.e.* air pressure. To conclude, the strength of association (correlation) can be estimated with observational data alone, this can answer the question: how likely will it rain, if the barometer needle sinks? But not: how would the weather change if I force the barometer needle to sink?

Pearl [16] distinguishes between three types of questions, that can be answered by different types of knowledge:

Table 1.1: Ladder of causation [13]. Questions at level i of the ladder are only accessible with information from level i or higher.

Level	Symbol	Typical Activity	Typical Questions
1. Association	$P(y\|x)$	Seeing	What if I see?
2. Intervention	$P(y\|\mathrm{do}(x), z)$	Doing	What if I do?
3. Counterfactual	$P(y_x\|x', y')$	Imagining	What if had done?

The levels of this *ladder of causation* [16, 13] are separate not only conceptually, but in the type of data or assumptions that have to be made in order to access them. In particular, by unsupervised learning from observational data only the first level is accessible. The second level requires interactional data or model assumptions, while the third is inaccessible without an explicit model. The answers to these hypothetical questions (counterfactuals) lie by definition not in the data (facts).

1.4.2 Disentangling requires Interventions or Model Assumptions

The results from the study of causal inference also entail that "purely" unsupervised disentangling, *i.e.* estimating $\hat{\mathbf{z}}_\mathbf{i}$ from samples $x \sim p(x)$, is impossible. A proof for this can be found in [17]. Current machine learning operates mostly on the level of association, estimating (complex) correlations from raw data. As we have seen, this purely

data-driven approach can only go so far. In contrast, humans seem to have the ability to interact with their environment and have innate assumptions on coherence, causality, physics etc., which introduce inductive priors [4]. To bring *i)* interventions and *ii)* model assumptions to our problem of disentangling shape and appearance, we *i)* apply changes to an image, which are assumed to change only one factor and *ii)* model the causal process of the image generation in the theme of analysis-by-synthesis.

2 Bibliography

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. 1

[2] Christopher M Bishop. Pattern recognition and machine learning (information science and statistics). 2006. 1, 4

[3] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv*, 2015. 2

[4] Josh Tenenbaum. Building machines that learn and think like people. In *AAMAS*, 2018. 2, 8

[5] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 1989. 3

[6] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 1991. 3

[7] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 3

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2013. 4

[9] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014. 4

[10] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv*, 2015. 5

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *arXiv*, 2017. 5

[12] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *TPAMI*, 2013. 5

[13] Judea Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution. In *WSDM*, 2018. 6, 7

[14] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017. 6, 7

[15] H. Reichenbach. *The Direction of Time*. University of California Press, 1956. 6

[16] Judea Pearl and Dana Mackenzie. *The Book of Why*. Hachette Book Group, 2018. 7

[17] Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. *arXiv*, 2018. 7