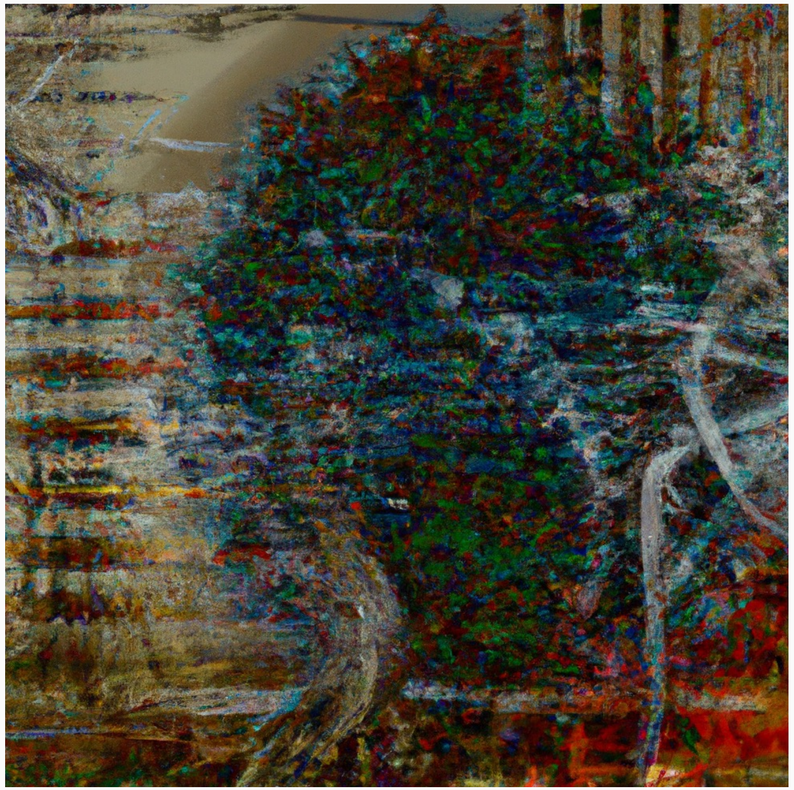
Imperial College London

Deep Learning

Lecture 6 Representation Learning

Krystian Mikolajczyk & Chen Qin & Seyed Moosavi



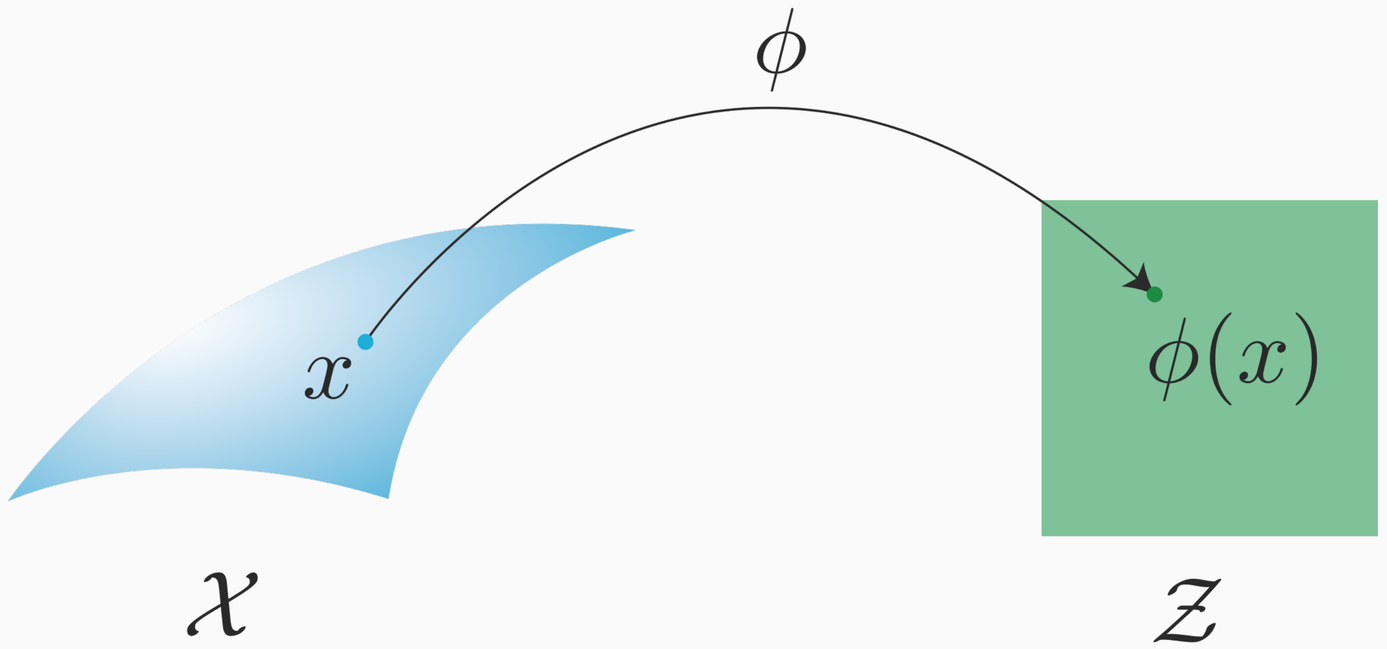
Hidden patterns emerge
In data's vast complexity
Representation born

— *ChatGPT*

What is a *representation*?

Representation: a mapping of data

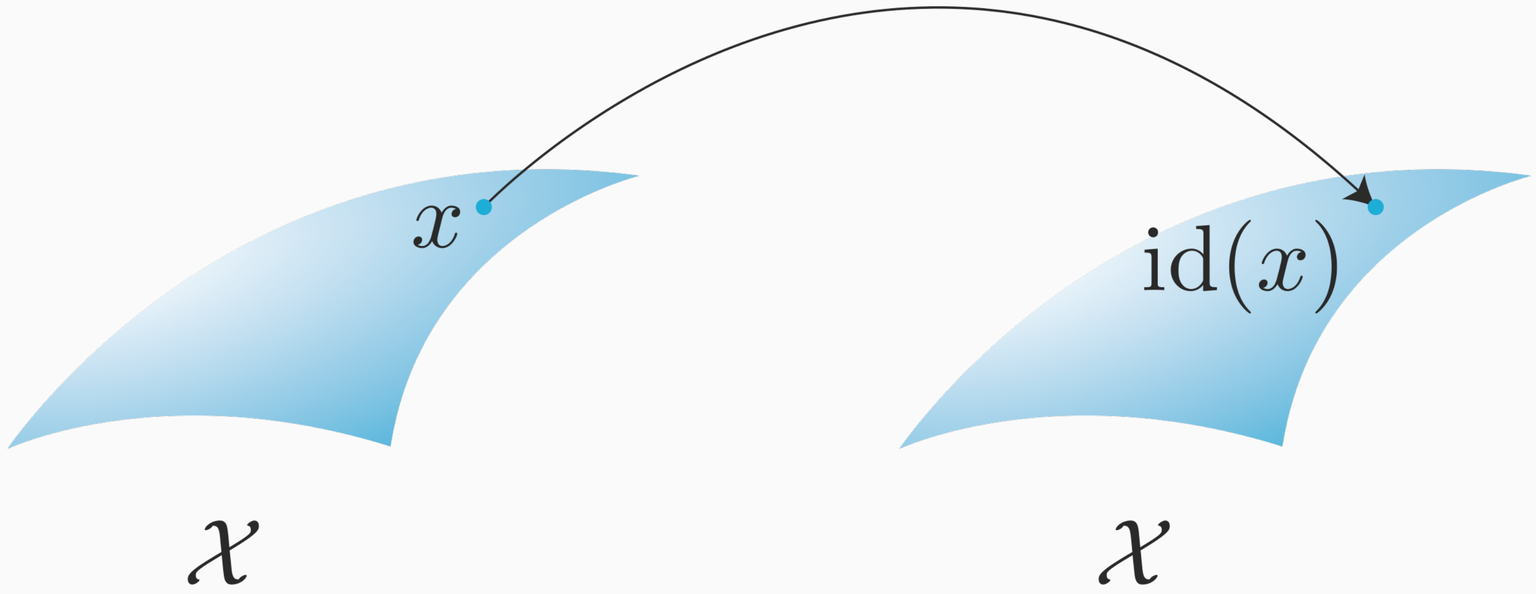
Representation is a feature of data that can entangle and hide more or less the different explanatory factors or variation behind the data. - *From Bengio, Courvillie and Vincent. "Representation learning: A review and new perspective."*



Data space

Latent/feature/representation space

Identity map

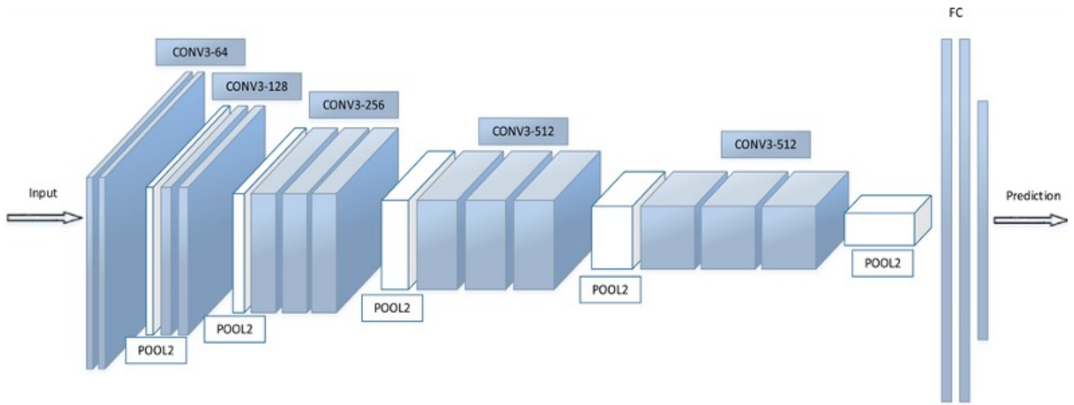


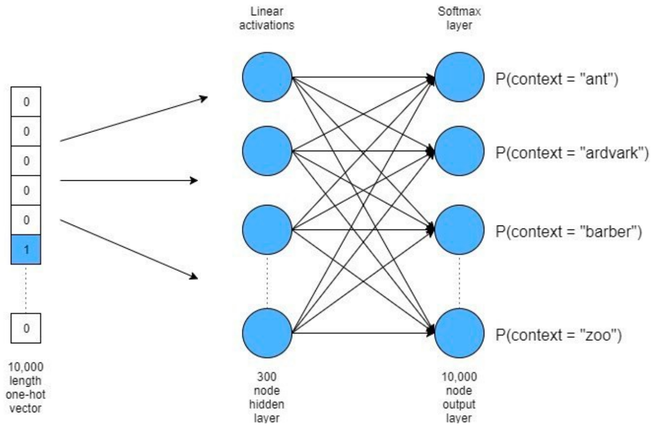
Data space

Latent/feature/representation space

What is a representation?

- Image Embeddings in Image Classification/Recognition
- Word/Token Embedding in NLP
- Multi-modal Representation
- …





Why are representations useful?

Invariance and abstraction

- Anomaly detection
- Information retrieval

Photo by Rupert Britton on Unsplash

Feature re-use

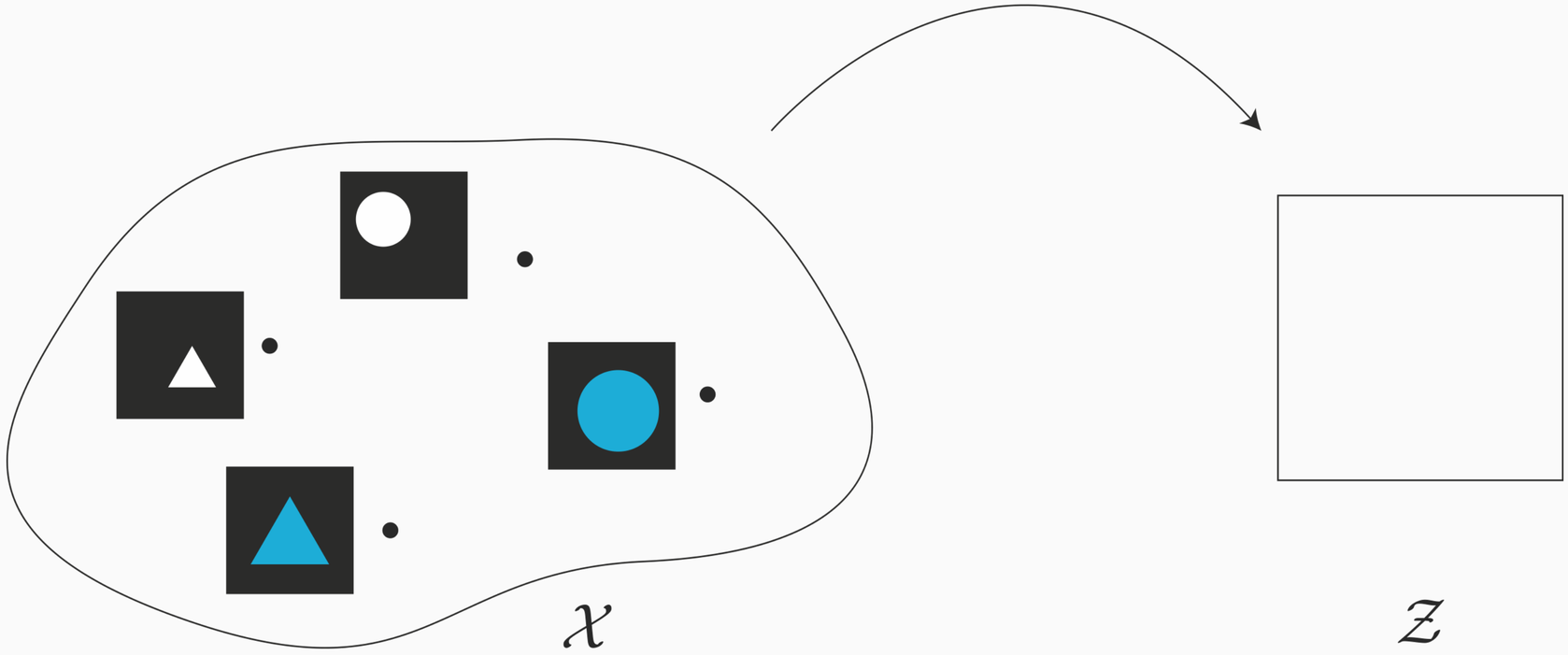
- Transfer learning
- Data generation

Representation learning is useful as it helps to automatically discover meaningful representations of data, which can then be used for various tasks such as classification, clustering, and dimensionality reduction. It enables machines to understand the underlying structure of data, leading to improved performance in tasks such as natural language processing and computer vision.

ChatGPT

What constitutes a *good* representation?

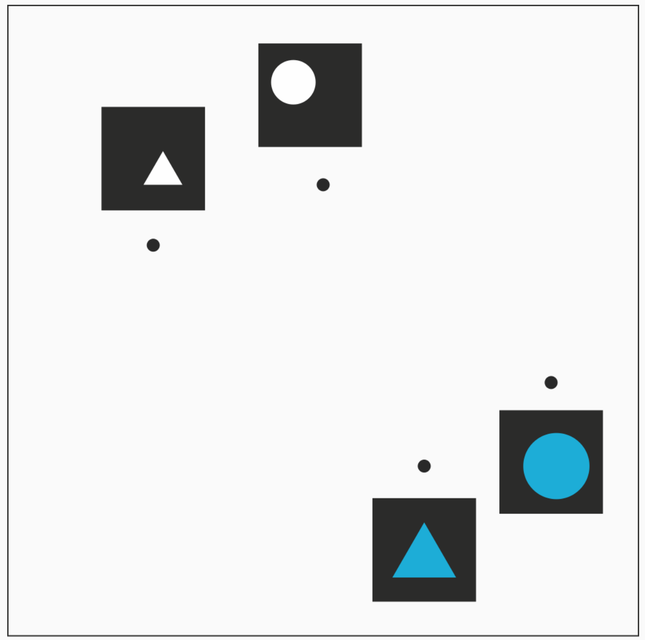
A toy example



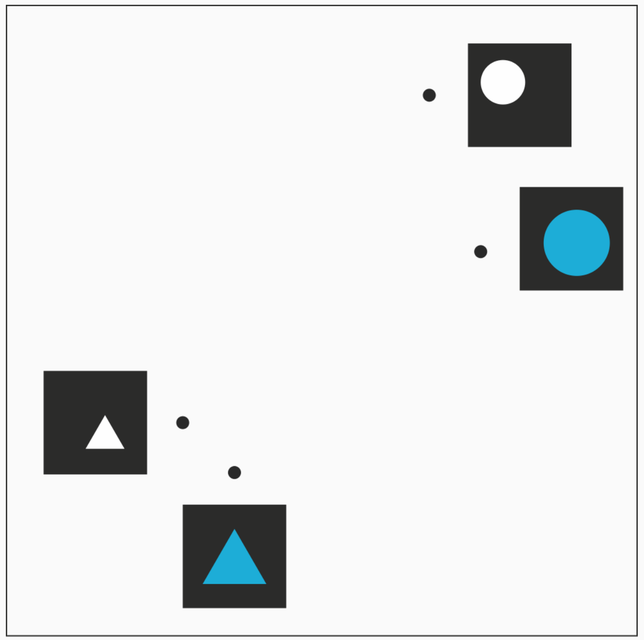
Data space

Representation/lat
ent/feature space

Which representation is better?

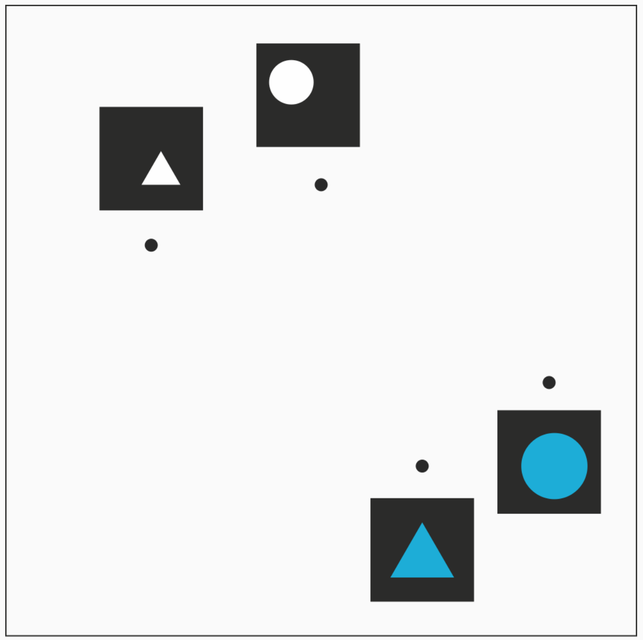


$\mathcal{Z}$

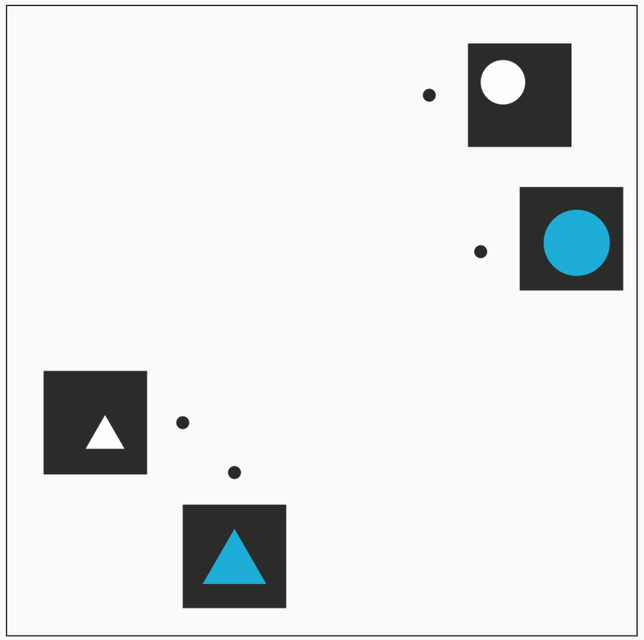


$\mathcal{Z}$

Which representation is better?



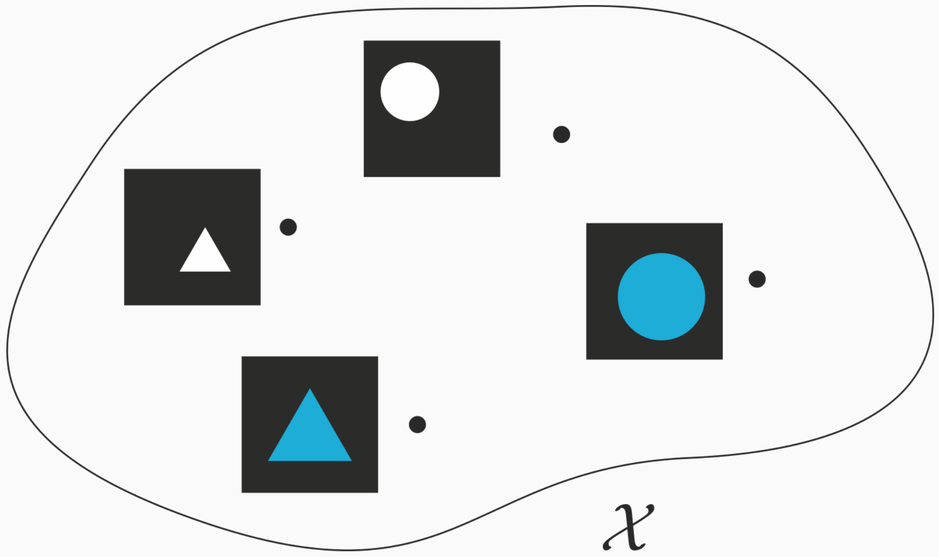
$\mathcal{Z}$



$\mathcal{Z}$

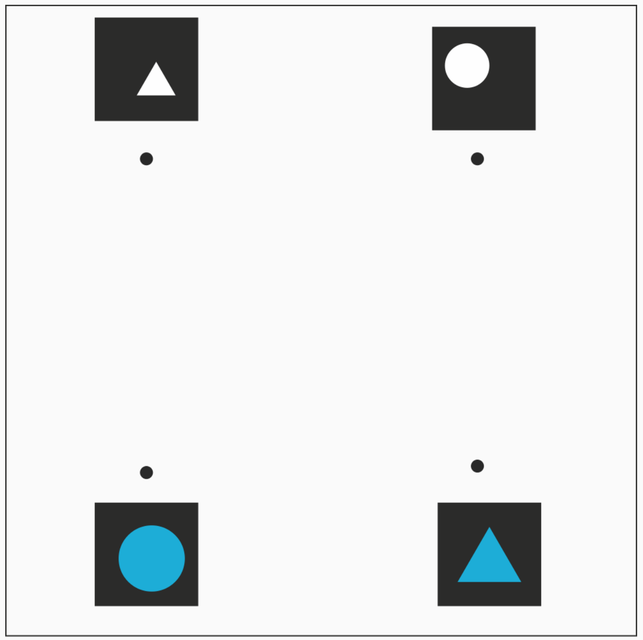
No Free Lunch! There is not a single feature representation that can solve all problems.

Good representations are *expressive.*

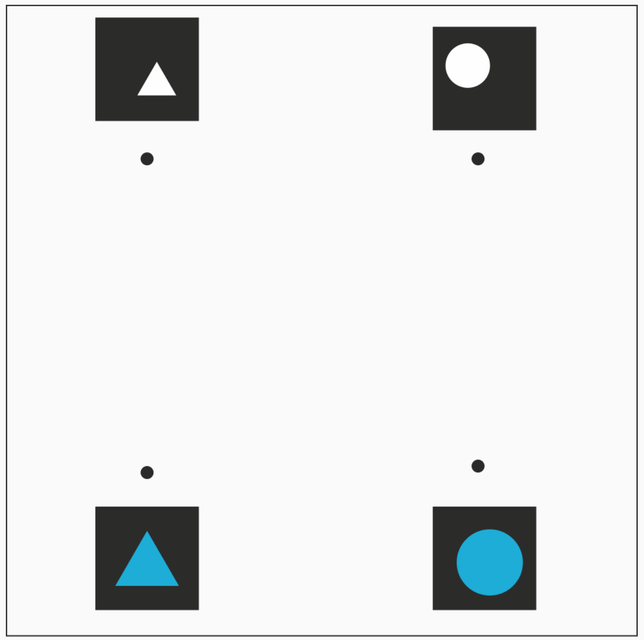


Colour
Position
Scale
Shape

Good representations are *disentangled.*

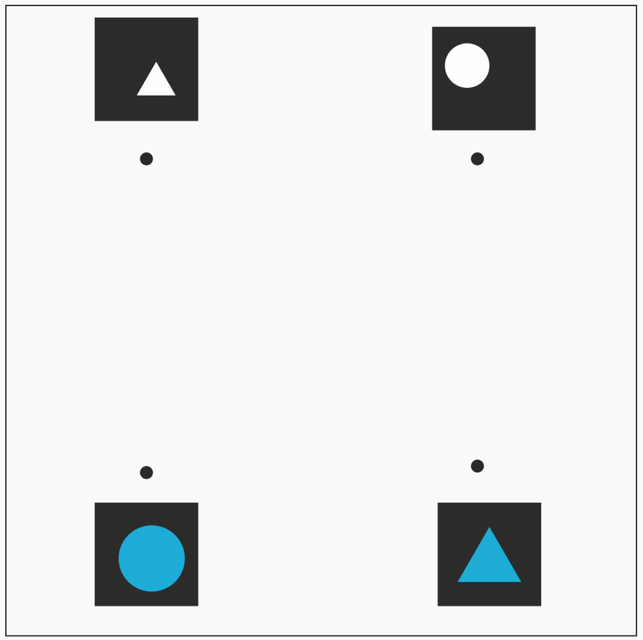


$\mathcal{Z}$

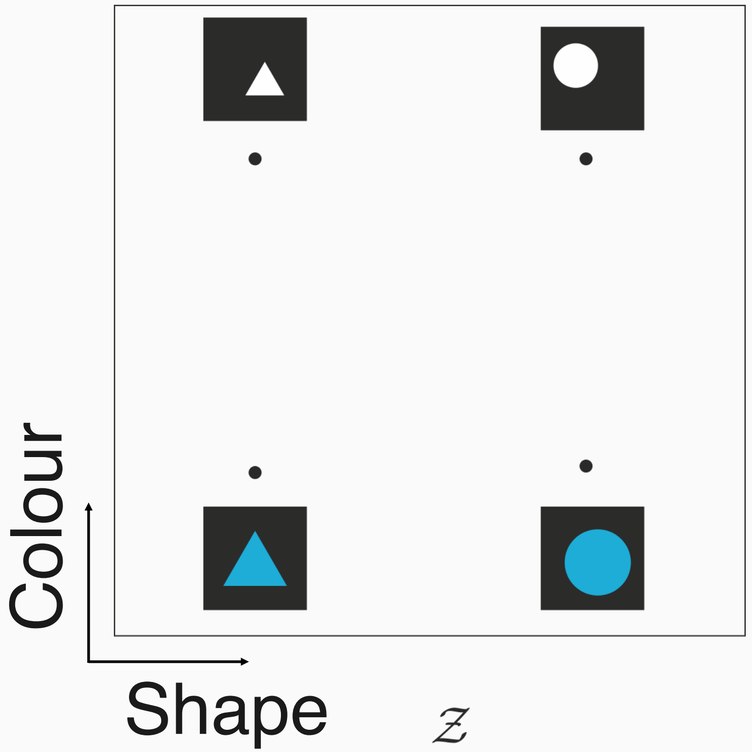


$\mathcal{Z}$

Good representations are *disentangled.*

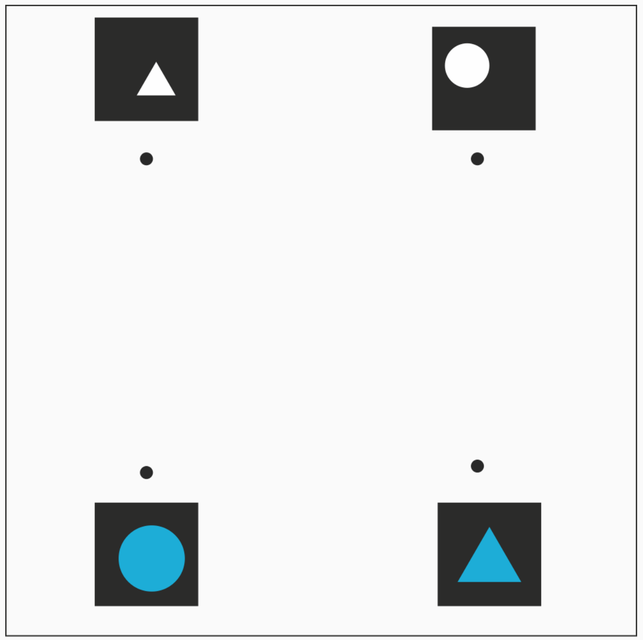


$\mathcal{Z}$

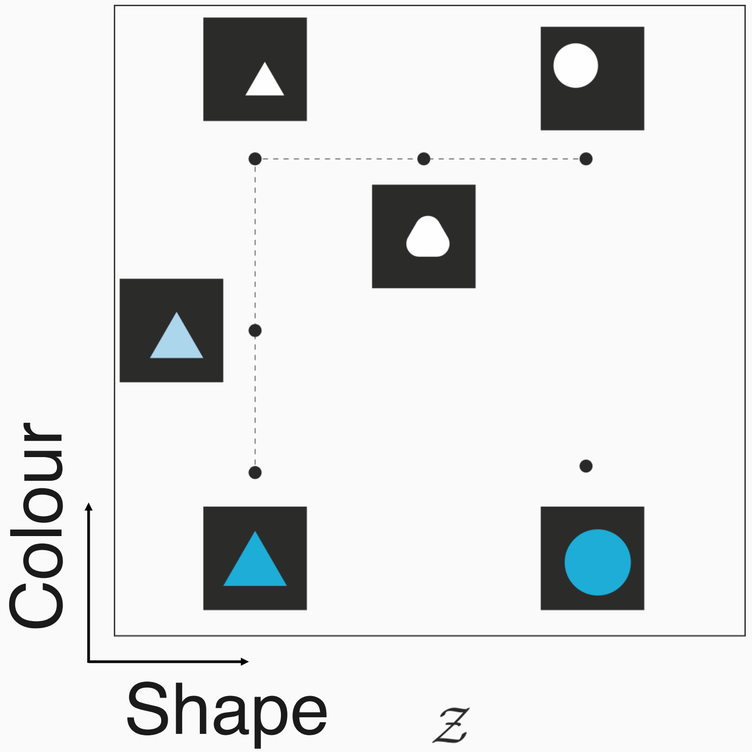


$\mathcal{Z}$

Good representations are *disentangled.*



$\mathcal{Z}$



$\mathcal{Z}$

A selective *history* of “Representation Learning”.

A *selective* history

Ronald Fisher

Karl Pearson

Early dimensionality reduction methods, such as PCA and LDA.

Early 1900s

A *selective* history

David Rumelhart

Paul Werbos

Representation learning via backpropagation

Early 1900s

Late 1900s

A *selective* history

Geoffrey Hinton

Yoshua Bengio

Layer-wise
representation learning

Early 1900s

Late 1900s

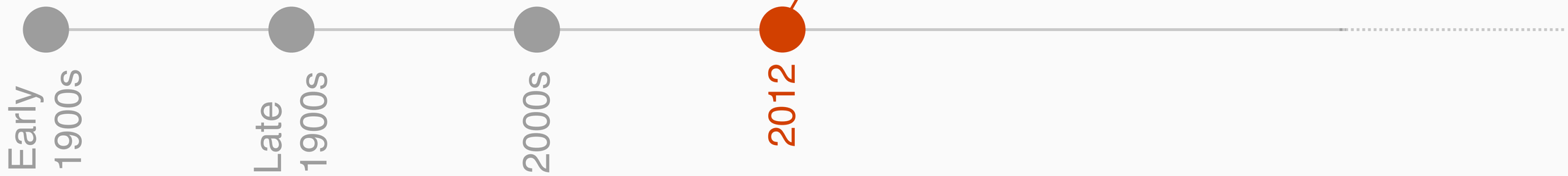
2000s

A *selective* history

Alex Krizhevsky

Yann LeCun

AlexNet: deep supervised representation learning



A *selective* history

Max Welling

Durk Kingma

Variational
AutoEncoders (VAEs)

Early 1900s

Late 1900s

2000s

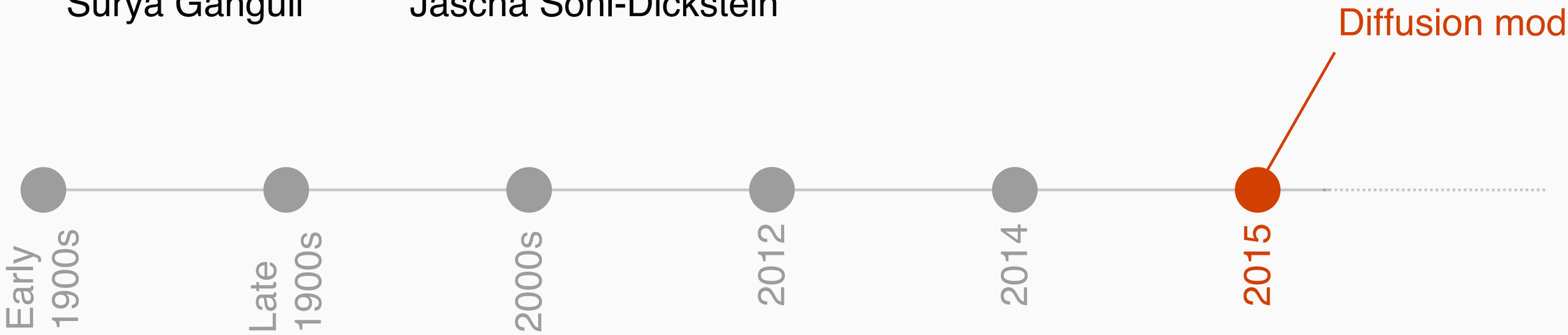
2012

2014

A *selective* history

Surya Ganguli

Jascha Sohl-Dickstein

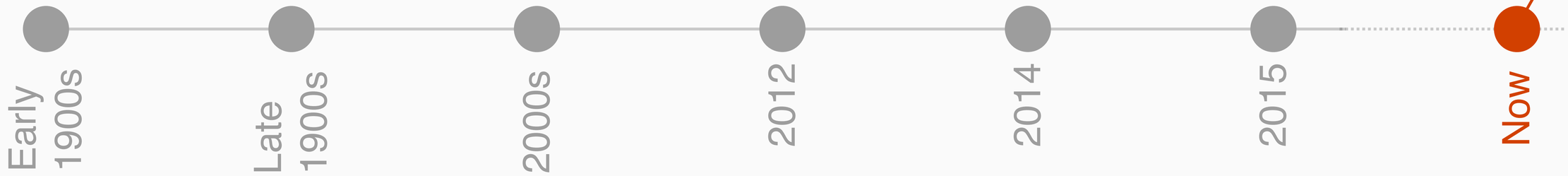


Now…

OpenAI

Stability AI

Latent diffusion models and multi-modal representation learning, such as StableDiffusion.

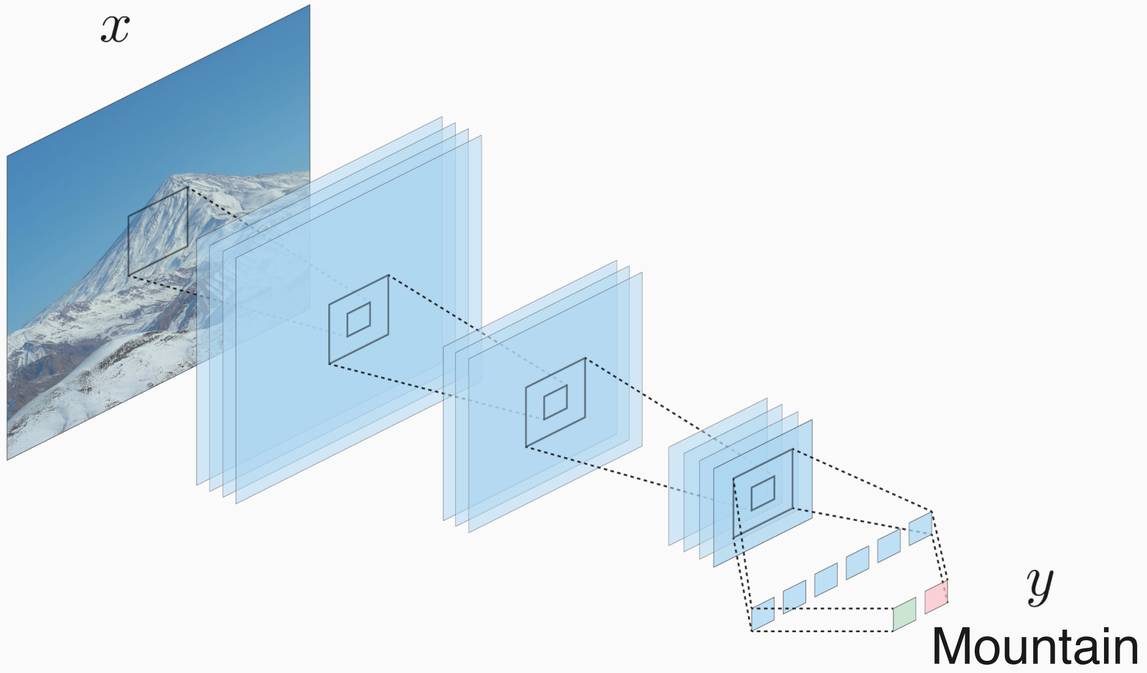


Unsupervised representation learning.

Supervised representation learning

- What we have seen so far: task-dependent representations, which are good for capturing $p(y|x)$.

- Goal: learn a relation between inputs and outputs f: X -> y



Unsupervised representation learning

What if we don’t have enough (labelled) data?

Unsupervised representation learning

Idea: Learning representations using unsupervised data.

Two main purposes:

- **Simplification:** reduce the (sample) complexity of a supervised learning problem by identifying only useful features in the data and discarding distracting elements

- **Interpretability:** identify common qualities (features) between individual elements or groups from huge amounts of data leading to a better understanding of the problem.

Benefit: usually inexpensive to collect large training sets (no need for labeling).

Unsupervised representation learning

Learning representations using unsupervised data.

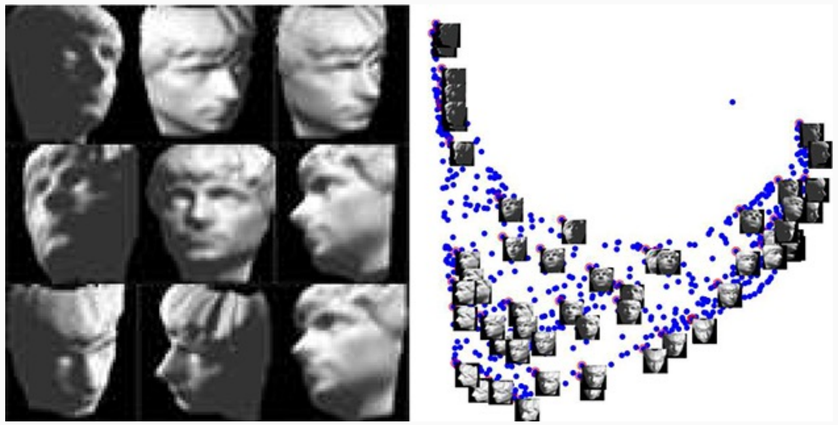
- Semi-supervised learning hypothesis: input data needs to encode useful properties about input-output relation, i.e., $p(x)$ and $p(y|x)$ share “something”.

Unsupervised representation learning

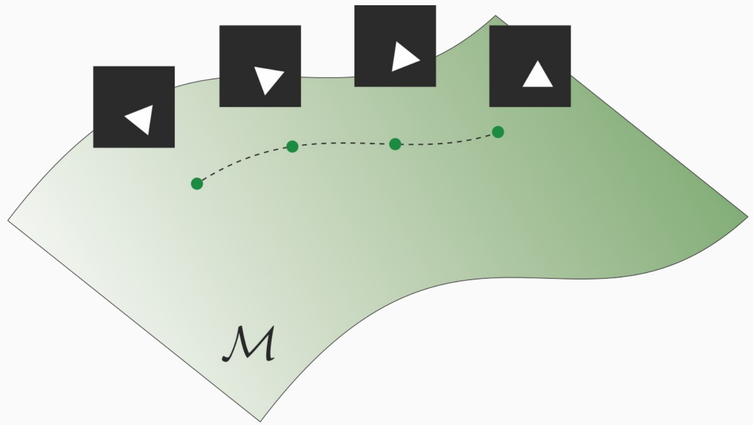
Learning representations using unsupervised data.

- Semi-supervised learning hypothesis: input data needs to encode useful properties about input-output relation, i.e., $p(x)$ and $p(y|x)$ share “something”.

- Manifold hypothesis: Data is likely to concentrate around a “low-dimensional manifold”.

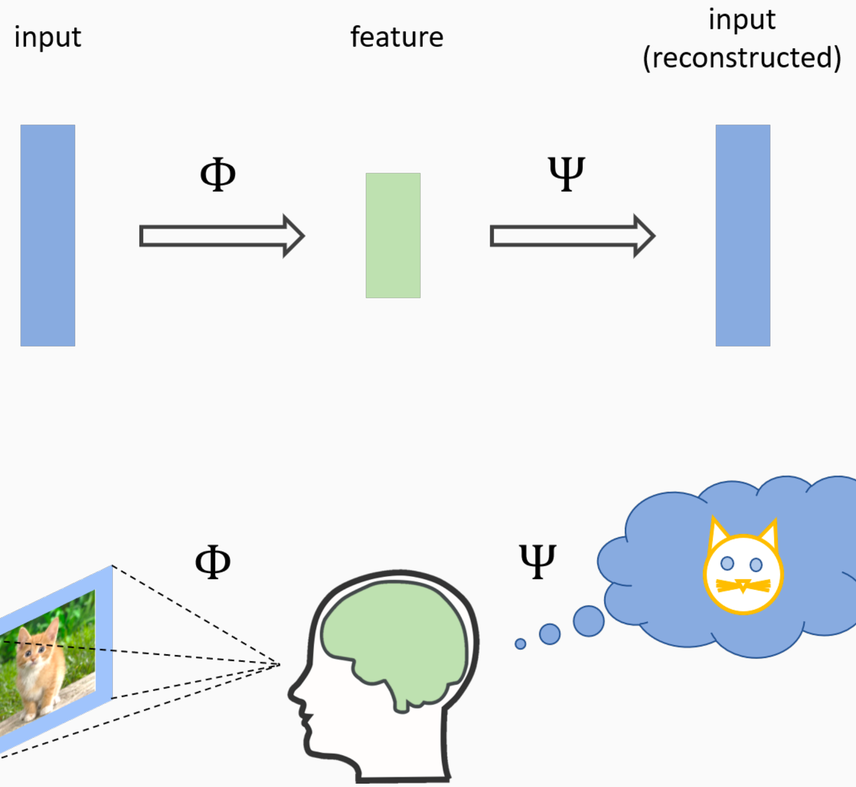


Pictures of human faces span a very low-dimensional subspace with respect to all possible images.



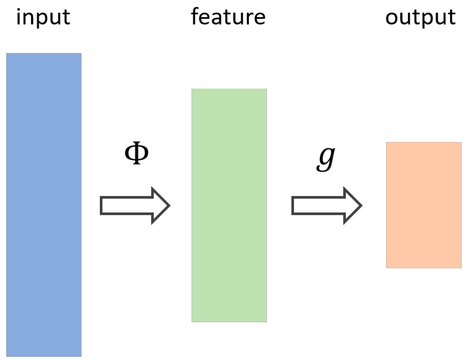
Efficient Coding Hypothesis

Efficient Coding Hypothesis *(Barlow '61)*
"… the Efficient Coding Hypothesis holds that the purpose of early visual processing is to produce an efficient representation of the incoming visual signal." *(Simoncelli '03)*



Representation Learning

We want to learn $f(x) = g(\phi(x))$



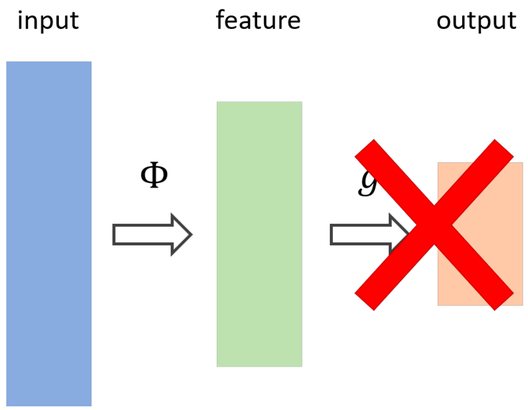
- ϕ feature representation
- g label predictor

End-to-end learning: learn ϕ and g jointly.

- Standard approach in supervised learning
- Works well if we have lots of labelled data

Representation Learning

We want to learn $f(x) = g(\phi(x))$



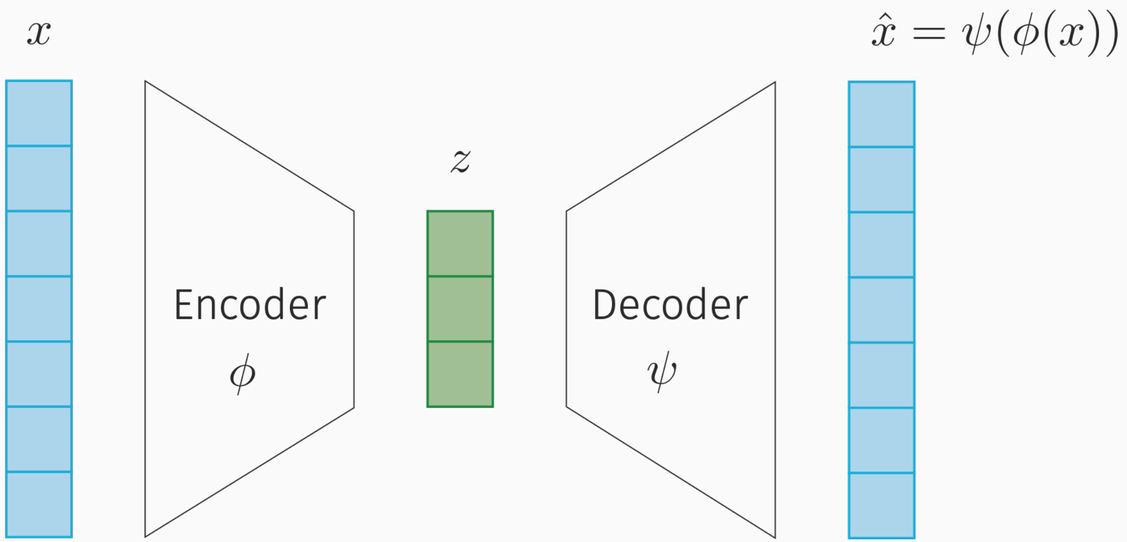
- ϕ feature representation
- g label predictor

Note: g needs output labels to be learned (which are expensive). But what about ϕ ?

Can we learn ϕ **without** using output labels?

Representation Learning: Autoencoder

Idea: use the reconstruction error on input data as a form of self-supervision

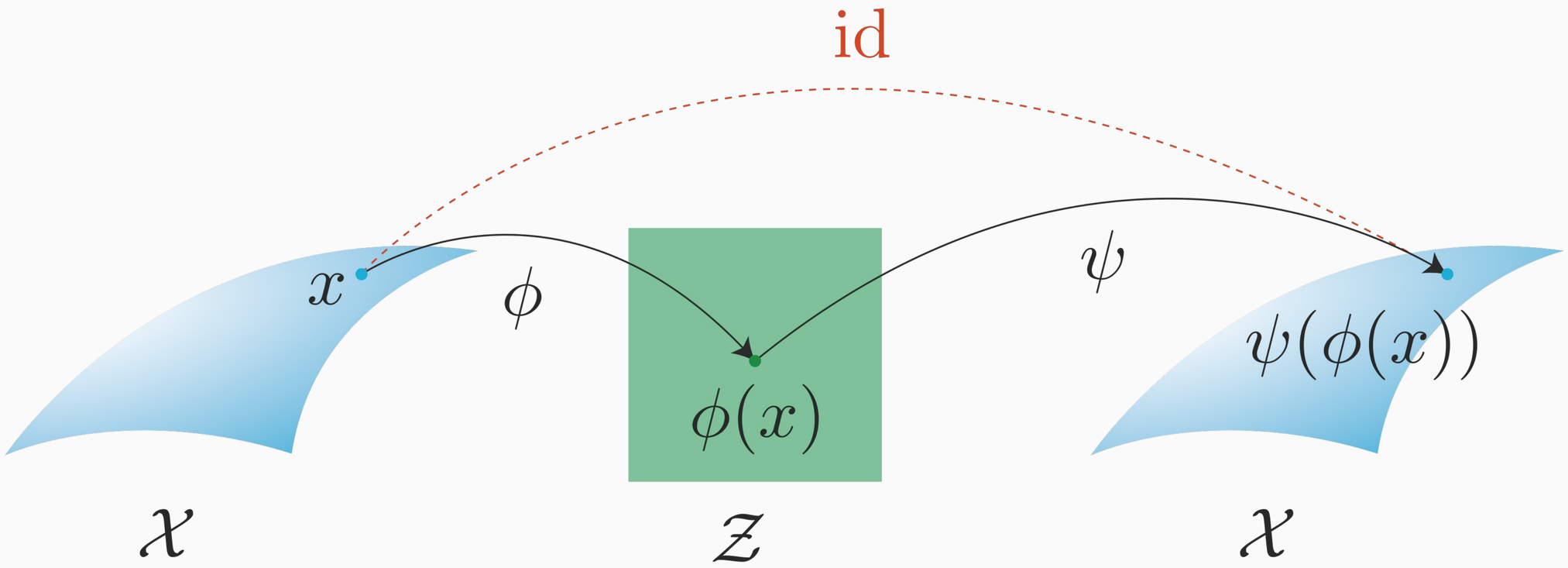


$$\mathcal{L}_{\text{AE}}(\phi, \psi) = \sum_i \|x_i - \hat{x}_i\|_2^2$$

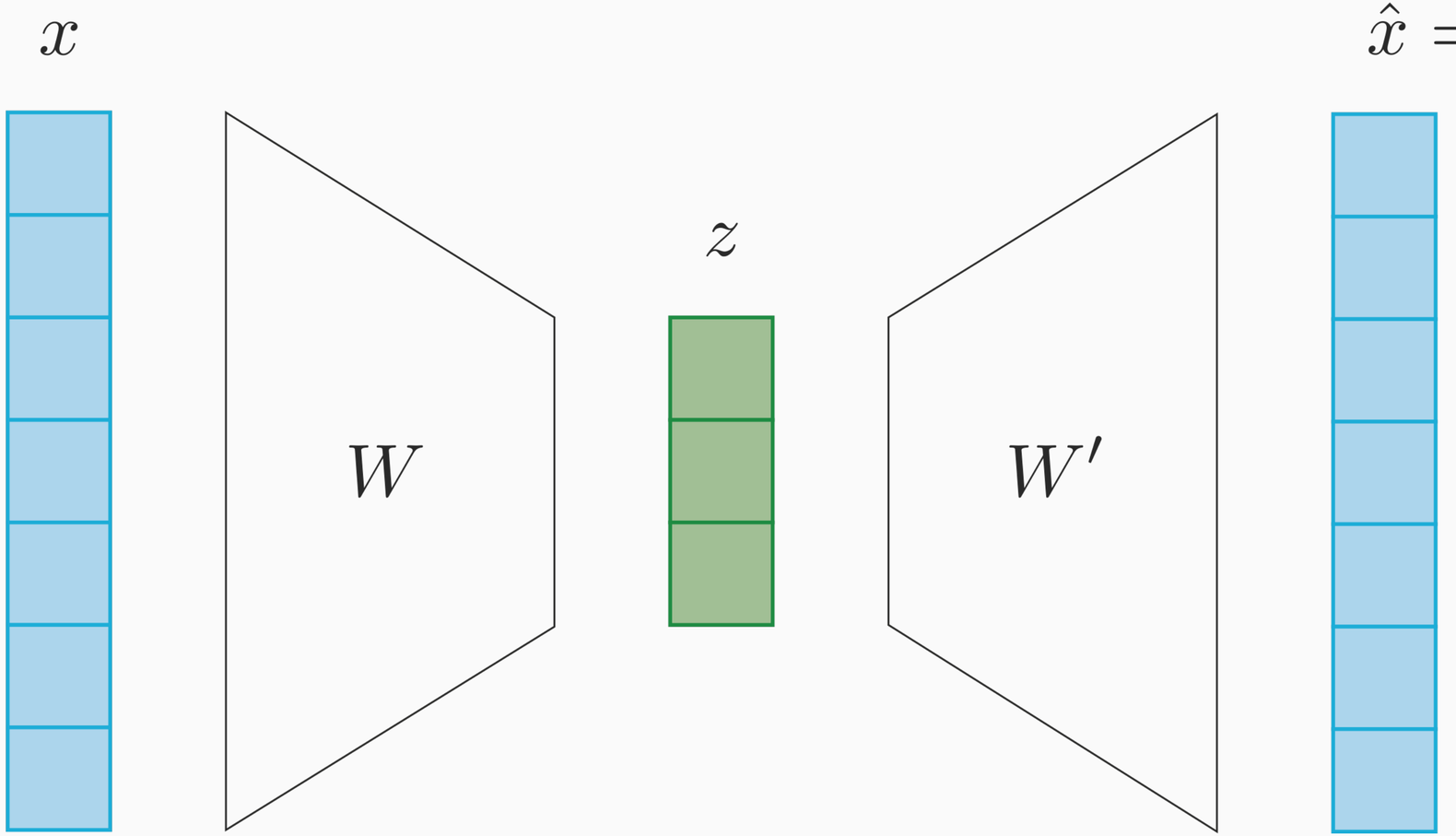
Benefits:

- Unlabelled input data is (often) inexpensive: we can get many self-labelled examples
- Learned ϕ can capture more general properties of the data: can be used for different tasks

“Self”-supervision



Linear Autoencoder



$$\mathcal{L}_{\mathrm{AE}}(W, W') = \sum_i \|x_i - W'Wx_i\|_2^2$$

Optimality of the linear autoencoder

Reconstruction loss

$$\mathcal{L}_{\text{AE}}(W, W') = \|X - W'WX\|_F^2.$$

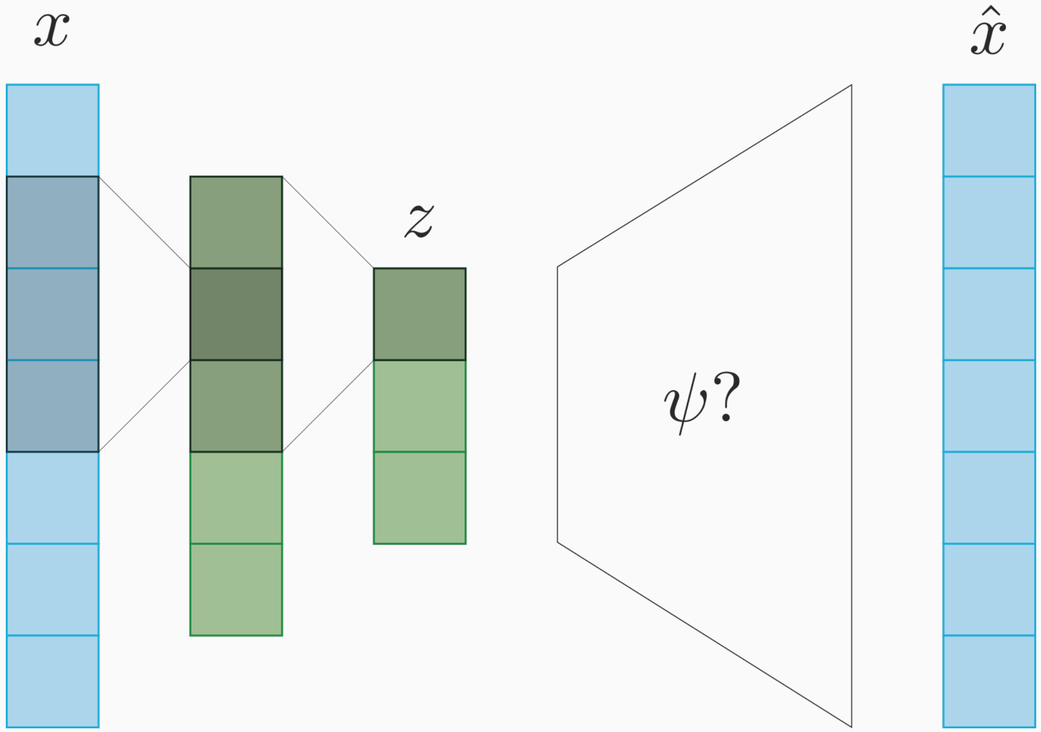
$X_{d\times n}$ is the data matrix, and $W_{p\times d}$ and $W'_{d\times p}$ are the encoder's and decoder's weights resepctively.

Let X have the following *singular value decomposition*: $X = U\Sigma V^\top$. Then the following W and W' are a minimizer of $\mathcal{L}_{\text{AE}}$:

$$W = \Sigma^{-1}_{1:p,1:p} U^\top_{\cdot,1:p},$$
$$W' = U_{\cdot,1:p}\Sigma_{1:p,1:p}.$$

Note that it is indeed connected to **Principal Component Analysis (PCA)**!

Convolutional Autoencoder

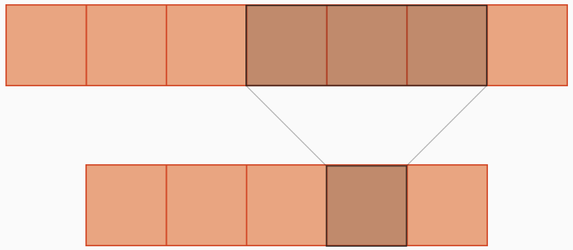


Transposed convolution

Standard convolution

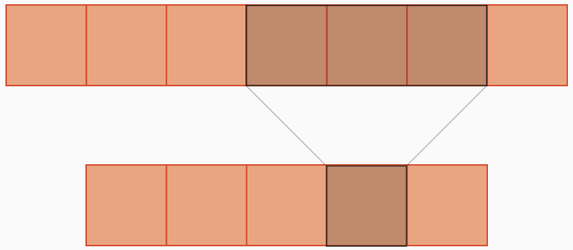
Transposed convolution

Standard convolution



Transposed convolution

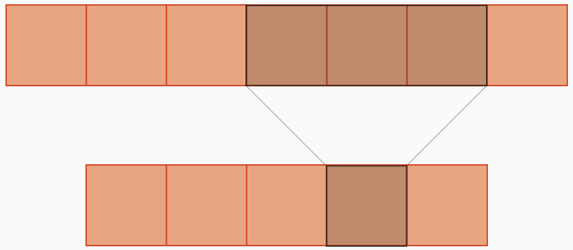
Standard convolution



Transposed convolution

Transposed convolution

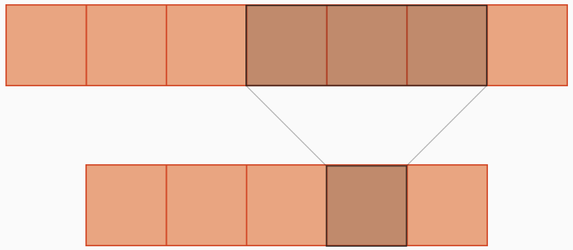
Standard convolution



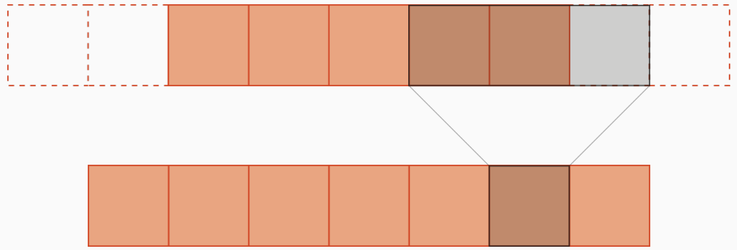
Transposed convolution

Transposed convolution

Standard convolution

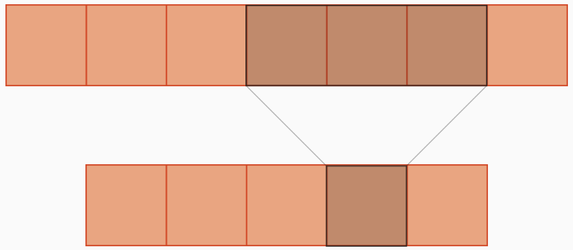


Transposed convolution



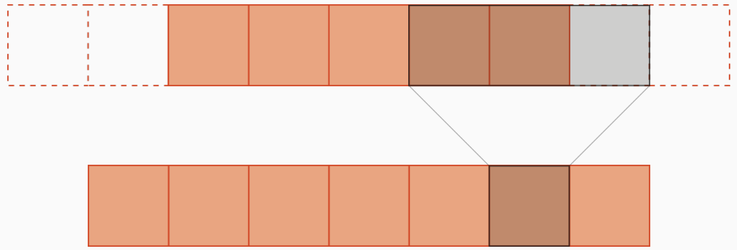
Transposed convolution

Standard convolution



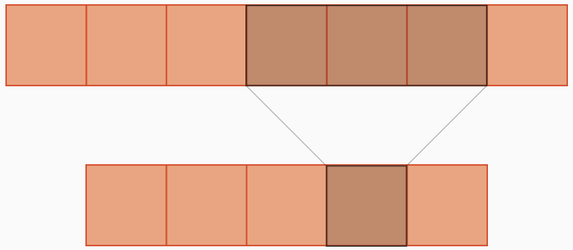
$$\begin{bmatrix} w_1 & w_2 & w_3 & 0 & 0 & 0 & 0 \\ 0 & w_1 & w_2 & w_3 & 0 & 0 & 0 \\ 0 & 0 & w_1 & w_2 & w_3 & 0 & 0 \\ 0 & 0 & 0 & w_1 & w_2 & w_3 & 0 \\ 0 & 0 & 0 & 0 & w_1 & w_2 & w_3 \end{bmatrix}$$

Transposed convolution



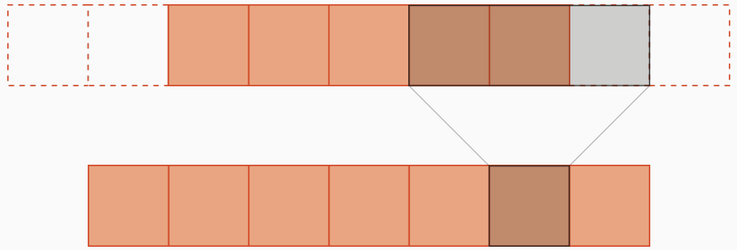
Transposed convolution

Standard convolution



$$\begin{bmatrix} w_1 & w_2 & w_3 & 0 & 0 & 0 & 0 \\ 0 & w_1 & w_2 & w_3 & 0 & 0 & 0 \\ 0 & 0 & w_1 & w_2 & w_3 & 0 & 0 \\ 0 & 0 & 0 & w_1 & w_2 & w_3 & 0 \\ 0 & 0 & 0 & 0 & w_1 & w_2 & w_3 \end{bmatrix}$$

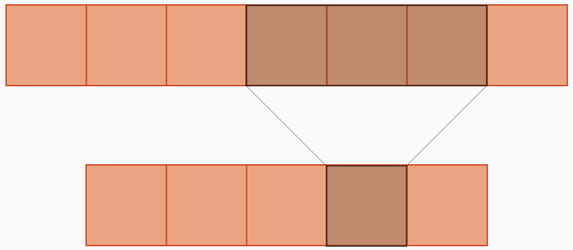
Transposed convolution



$$\begin{bmatrix} w_1 & 0 & 0 & 0 & 0 \\ w_2 & w_1 & 0 & 0 & 0 \\ w_3 & w_2 & w_1 & 0 & 0 \\ 0 & w_3 & w_2 & w_1 & 0 \\ 0 & 0 & w_3 & w_2 & w_1 \\ 0 & 0 & 0 & w_3 & w_2 \\ 0 & 0 & 0 & 0 & w_3 \end{bmatrix}$$

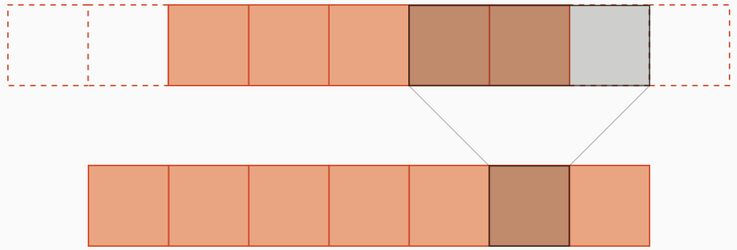
Transposed convolution

Standard convolution



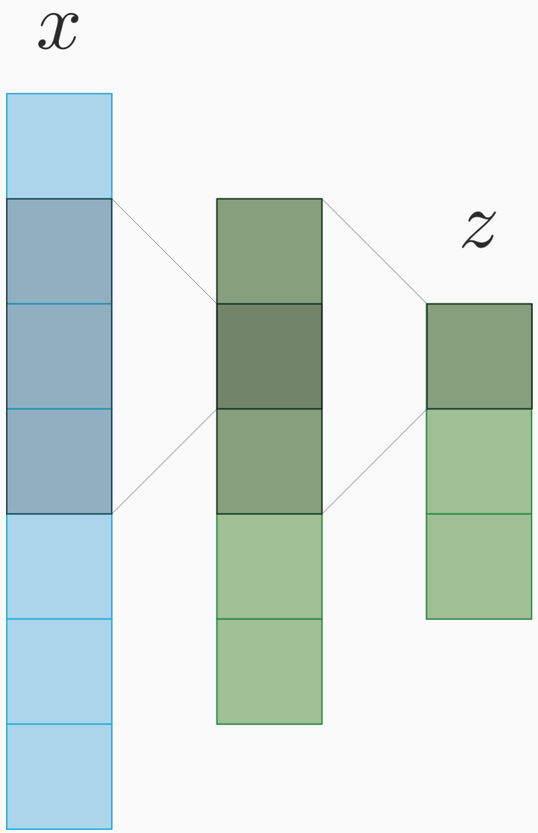
$$\begin{bmatrix} w_1 & w_2 & w_3 & 0 & 0 & 0 & 0 \\ 0 & w_1 & w_2 & w_3 & 0 & 0 & 0 \\ 0 & 0 & w_1 & w_2 & w_3 & 0 & 0 \\ 0 & 0 & 0 & w_1 & w_2 & w_3 & 0 \\ 0 & 0 & 0 & 0 & w_1 & w_2 & w_3 \end{bmatrix}$$

Transposed convolution

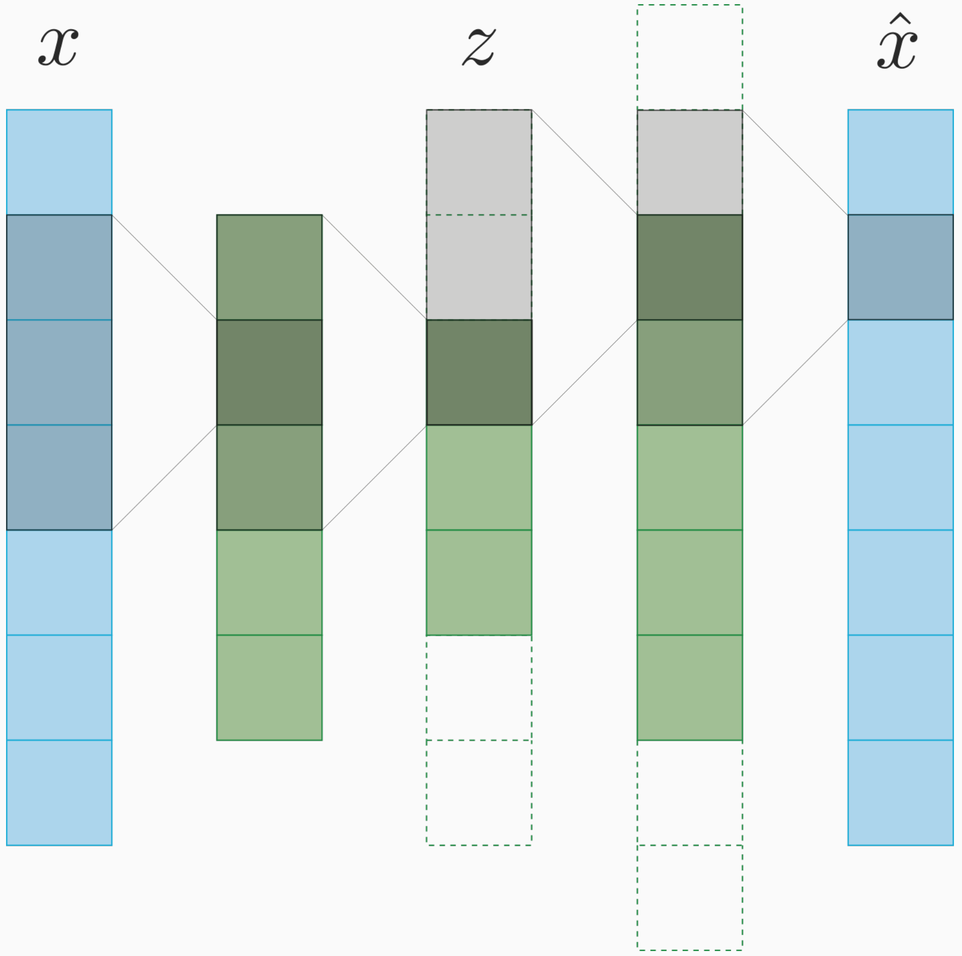


$$\begin{bmatrix} 0 & 0 & w_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_2 & w_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & w_3 & w_2 & w_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & w_3 & w_2 & w_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_3 & w_2 & w_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & w_3 & w_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w_3 & 0 & 0 \end{bmatrix}$$

Convolutional Autoencoder



Convolutional Autoencoder



Quiz

d the dimension of input/output space.

p the dimension of latent space.

Can we pick $p \geq d$?

Autoencoders: Models

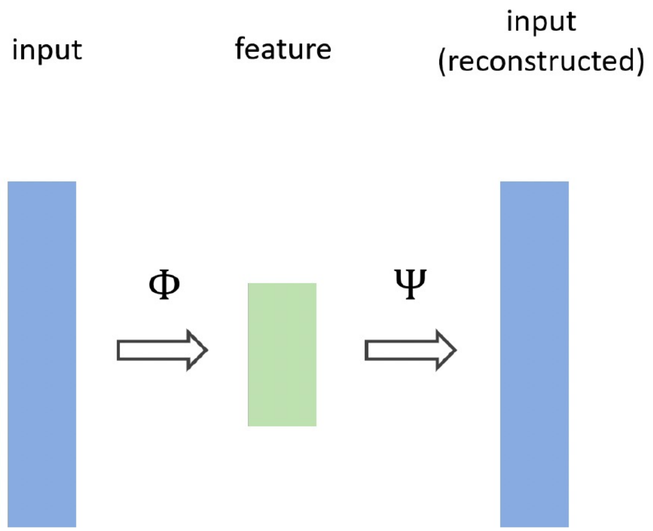
Autoencoders: Models

Many different options for Autoencoders:

- Undercomplete
- Overcomplete + Regularised
- Denoising
- Contractive
- …

Undercomplete Autoencoders

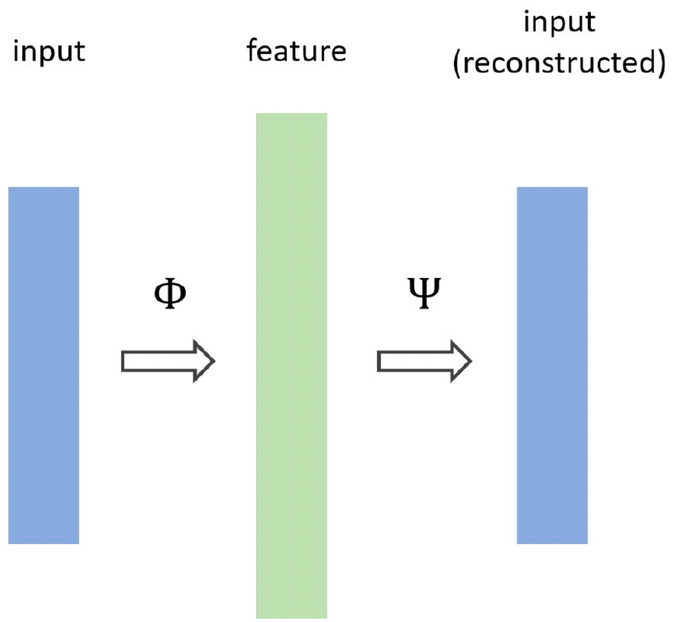
Feature representation layer dimension significantly smaller than input layer.



- We obtain a concise description of the input features (a sort of "semantic" compression).
- Also useful for: dimensionality reduction, data visualization, clustering, ...

Overcomplete Autoencoders

Feature representation layer dimension significantly larger than input layer.



- Allows for a rich representation (e.g. neurons learn a "dictionary" of patterns)...
- ...but needs regularization!

Regularised Autoencoders

Add a penalty $\Omega(\psi, \phi)$ (often just $\Omega(\phi)$) to the learning problem

$$\min_{\psi,\phi} \frac{1}{n} \sum_{i=1}^{n} \|x_i - \psi(\phi(x_i))\|^2 + \Omega(\psi, \phi)$$

Intuition: Ω to encourages the weights of the networks ϕ and ψ to be:

- small,
- sparse,
- symmetric,
- ...

Sparse Autoencoders

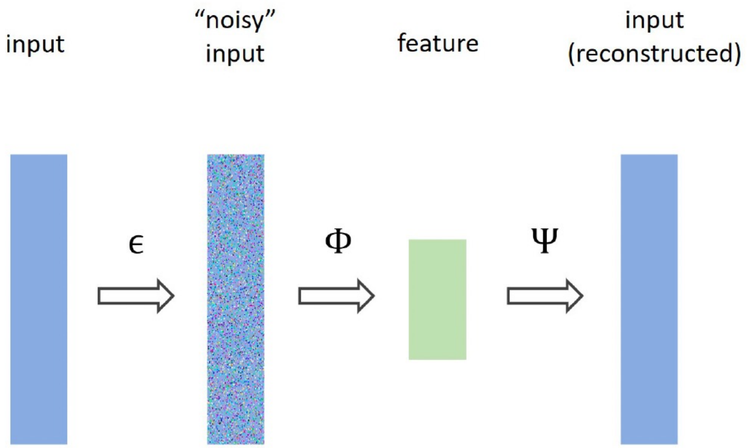
Idea: Learn an overcomplete representation of neurons that are tuned only to specific patterns in the data $\longrightarrow$ we would like to observe sparse activation patterns!

Connection with the behavior of simple cells in the mammalian primary visual cortex.

Denoising Autoencoders

Setting: noise can affect data. An efficient coding should automatically "ignore" it.

Approach: given a data point x, "corrupt" it with some noise ε, obtanining for instance $\hat{x} = x + \varepsilon$ (or $\hat{x} = x\varepsilon$). Then learn how to "denoise" $\hat{x}$ and reconstruct x.



$$\min_{\phi,\psi} \frac{1}{n} \sum_{i=1}^{n} \|x_i - \psi(\phi(x_i + \varepsilon_i))\|^2$$

Denoising Autoencoders aim to be robust with respect to non-smooth perturbation of input data (such as **noise**).

Denoising Autoencoders

Denoising Autoencoder: a Data Augmentation Perspective

- For each x_i, we can sample many perturbations $\varepsilon_i^{(1)}, \ldots, \varepsilon_i^{(m)}$ "generate" even more training points!

- Noise is meant in a very abstract sense. For instance if we want to be **invariant** to specific input transformations (e.g. rotations, translations, etc.), we can perturb x_i accordingly.

Robust Autoencoders

A different approach to build robust representations is to replace the quadratic error with a loss function that is more robust to outliers.

$$\min_{\psi,\phi} \frac{1}{n} \sum_{i=1}^{n} \|x_i - \psi(\phi(x_i))\|^2 \qquad \longrightarrow \qquad \min_{\psi,\phi} \frac{1}{n} \sum_{i=1}^{n} \ell(x_i, \psi(\phi(x_i)))$$

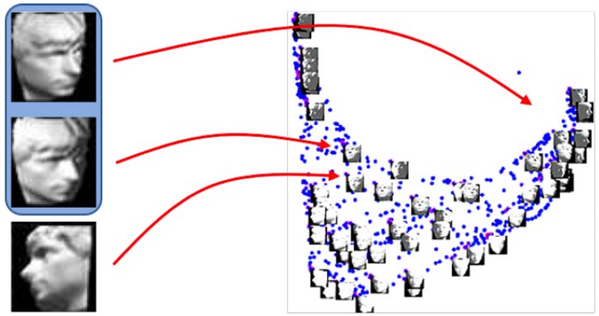
Examples:

- Absolute value loss $\ell(x, \psi(\phi(x))) = \sum_{k=1}^{d} |x^{(k)} - \psi(\phi(x))^{(k)}|$.
- "Correntropy" $\ell(x, \psi(\phi(x))) = \sum_{k=1}^{d} \exp(-(x^{(k)} - \psi(\phi(x))^{(k)})^2/\sigma)$. With $\sigma > 0$.
- Huber, Cauchy, German-McLure, Fair, $L2 - L1$...

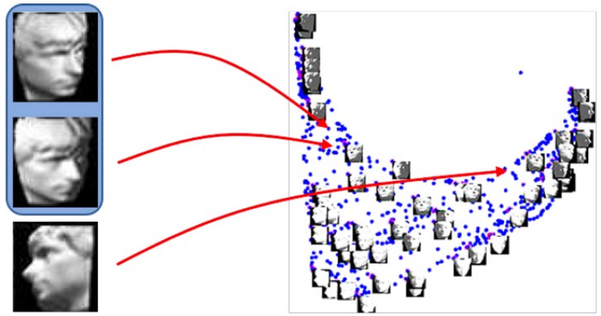
with $x^{(k)}$ denoting the k-th entry of $x \in \mathbb{R}^d$.

Contractive Autoencoders

Idea: We would like the feature map ϕ to be "stable" to small perturbations in the input.



Unstable
(similar input -> different representation)



Stable
(similar input -> similar representation)

Wishlist. We would like a representation that is:

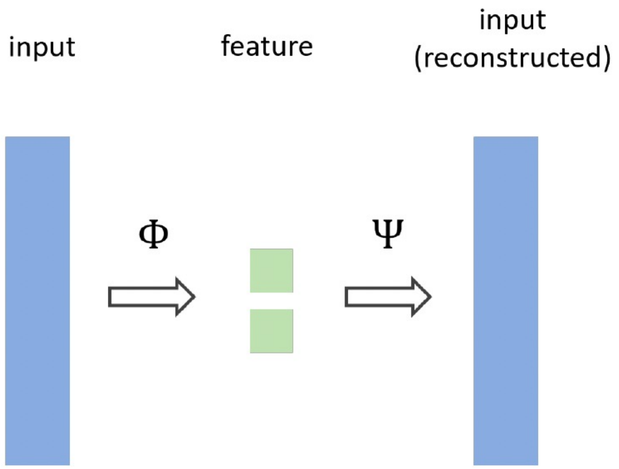
- Locally invariant to small changes around training points,
- Able to dicover the low-dimensional manifold structure in the data (if any).

Autoencoders: Applications

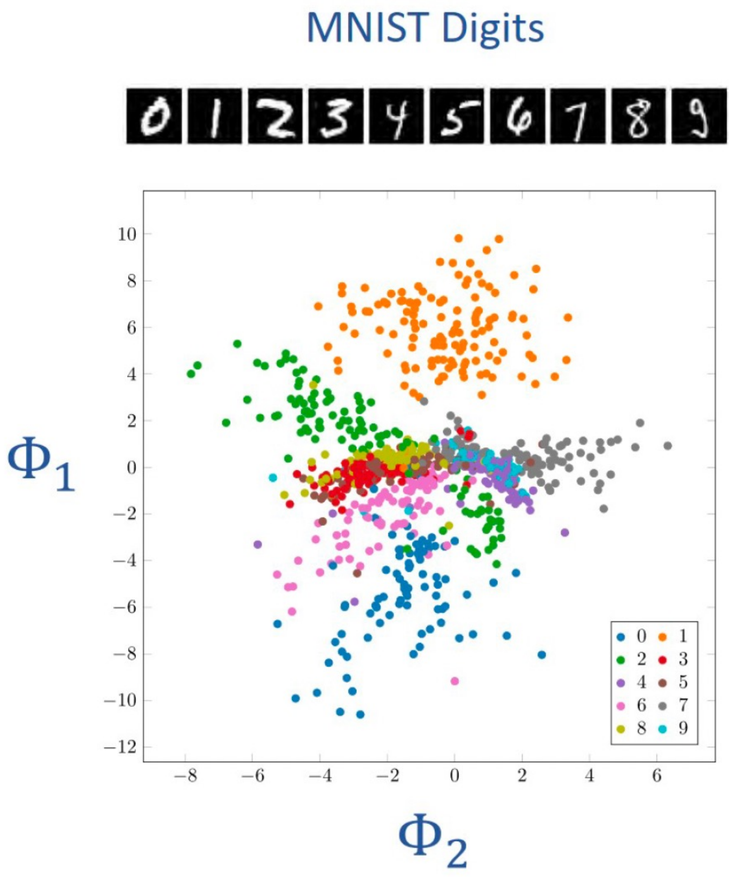
Applications: Data Visualisation

Idea: set the feature representation layer to have only 2 or 3 dimensions.

Then we can visualize each data point in this new representation!



Useful for: interpretation.



Application: Anomaly Detection

Task: given a set of points $x_1, \ldots, x_n$ sampled from a distribution ρ, we want to determine whether a new point x is sampled from the same distribution or not ($=$ an anomaly).

Idea: since Autoencoders implicitly learn the manifold on which ρ is "supported"

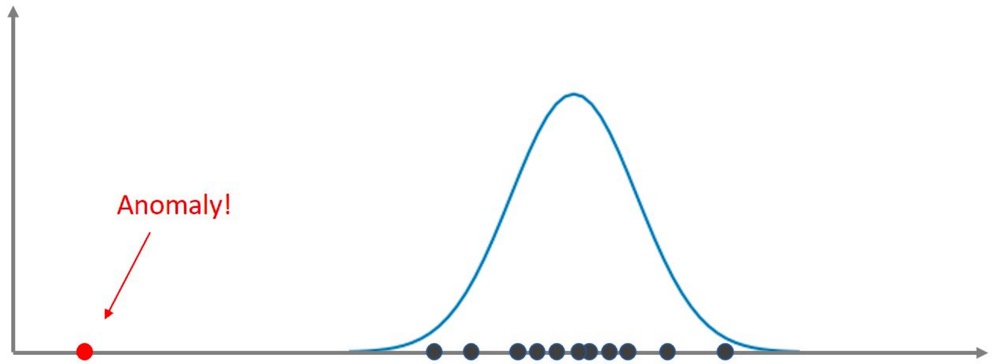
$\Longrightarrow$ whenever a point x is not sampled from ρ it will have a high reconstruction error.

Approach:

1. Train an Autoencoder on $x_1, \ldots, x_n$.
2. Given a new point x, if its reconstruction error

$$\|x - \psi(\phi(x))\| > \tau$$

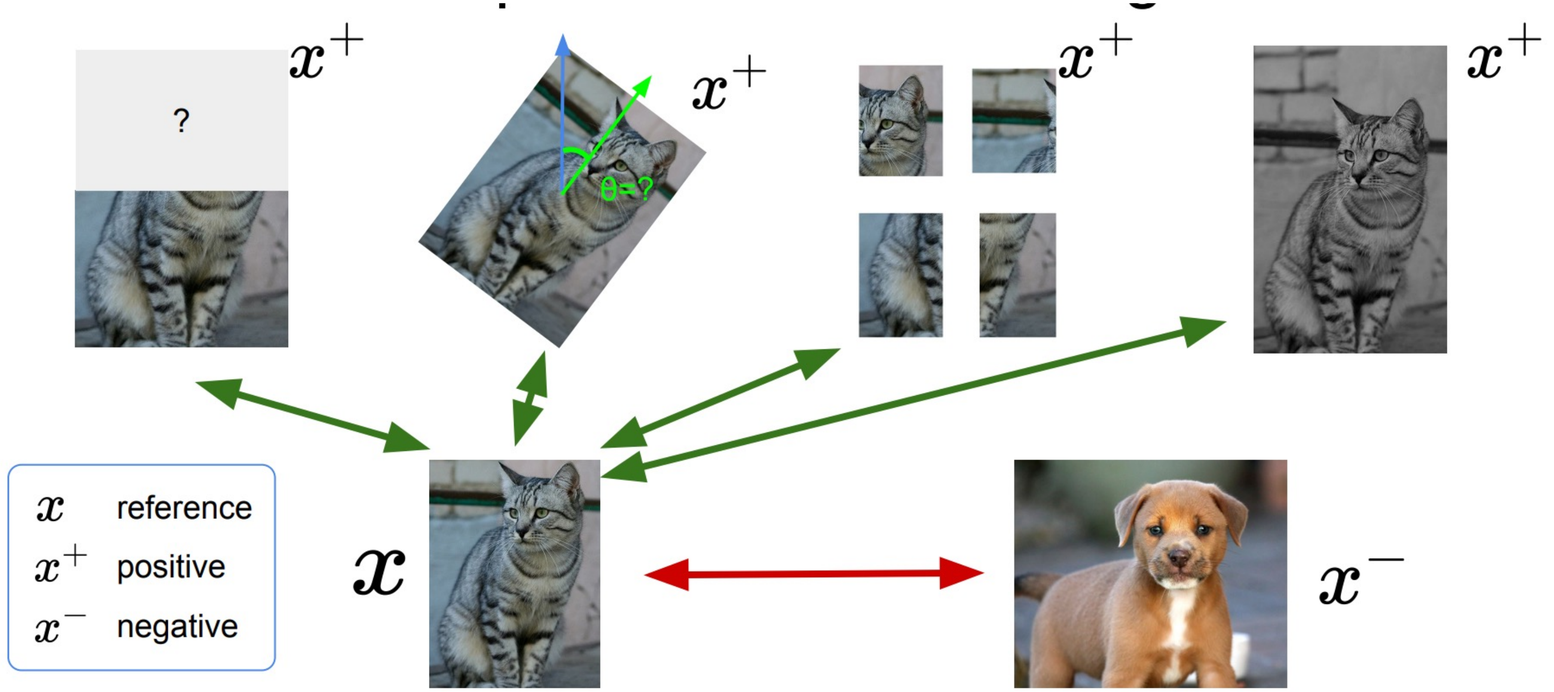
is greater than a threshold $\tau > 0$, mark it as an anomaly.



Contrastive Representation Learning

Contrastive Representation Learning

Contrastive learning is an approach to learning that focuses on extracting meaningful representations by contrasting positive and negative pairs of instances.



$$\text{score}(f(x), f(x^+)) >> \text{score}(f(x), f(x^-)) \quad \Rightarrow \quad L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

score for the positive pair

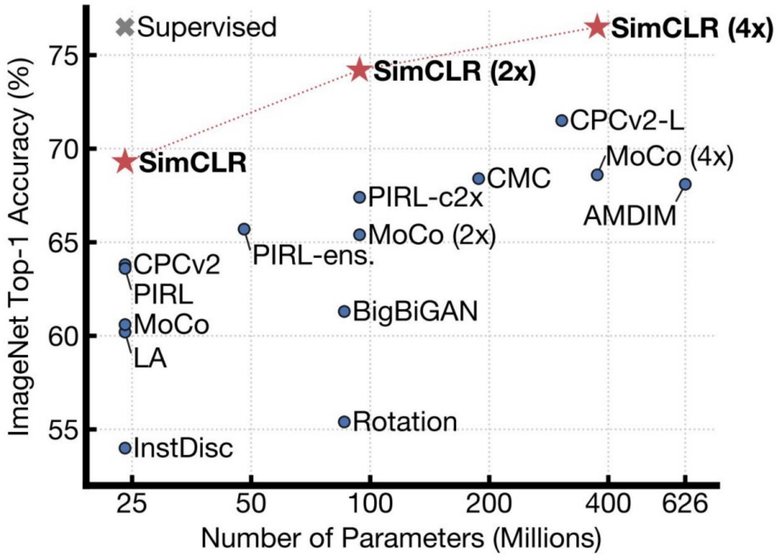
score for the N-1 negative pairs

Image credit: CS231n

Contrastive Representation Learning

Simple Contrastive Learning of Representations (SimCLR)

Idea: maximise the agreement between augmented views of the same instance while minimising the agreement between views from different instances



Summary

- **Representation**
- **Representation Learning:** Autoencoders
- **Autoencoder Models**
  - Undercomplete/Overcomplete Autoencoders
  - Denoising/Robust Autoencoders
  - Contractive Autoencoder
  - …
- **Autoencoder Applications**
  - Data Visualisation/Interpretation
  - Anomaly Detection
  - Transfer Learning
  - …
- **Other Representation Learning Approaches:** Contrastive Learning, Contrastive Predictive Coding, Masked Autoencoder, Joint Embedding Architecture…