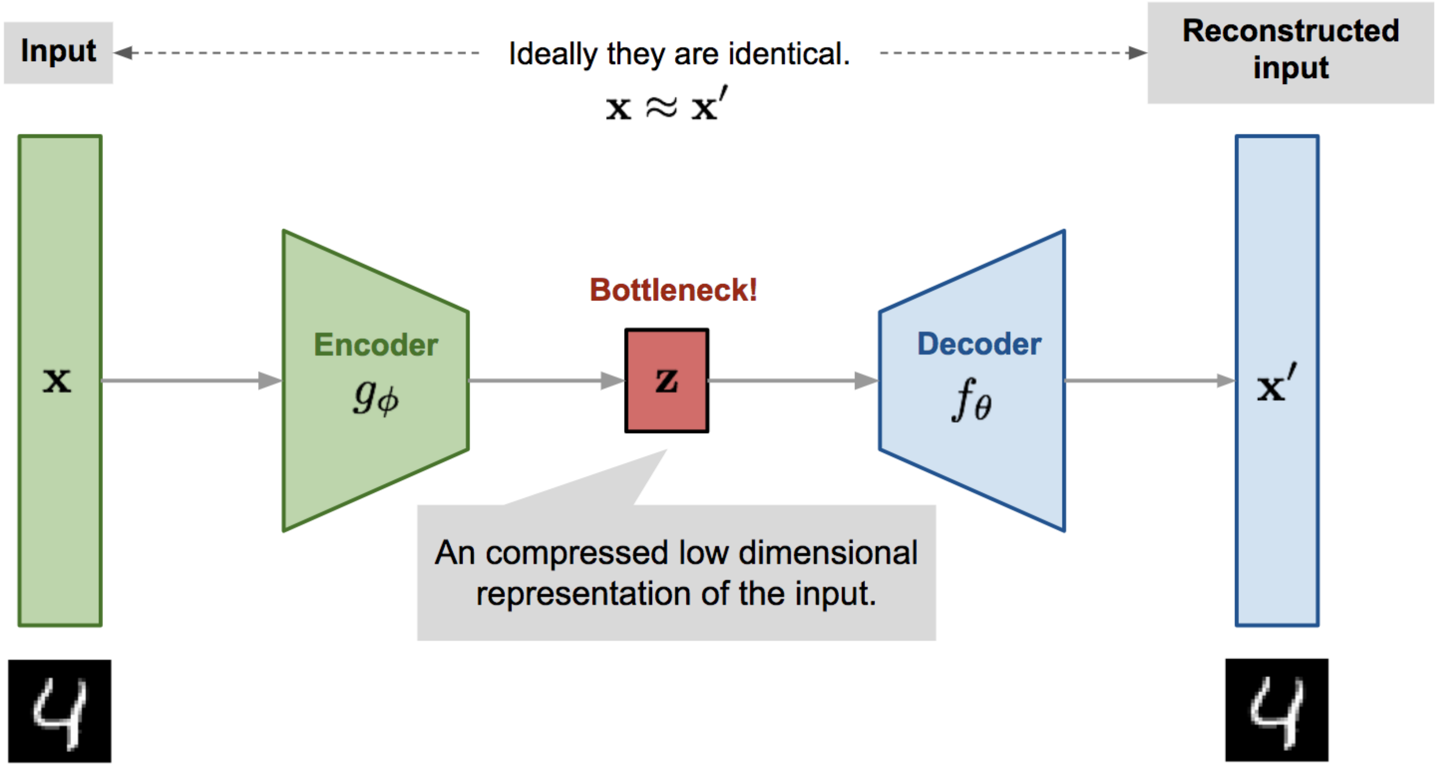
Autoencoders

WORKSHOP ON MACHINE LEARNING TECHNIQUES, AUTUMN 2022

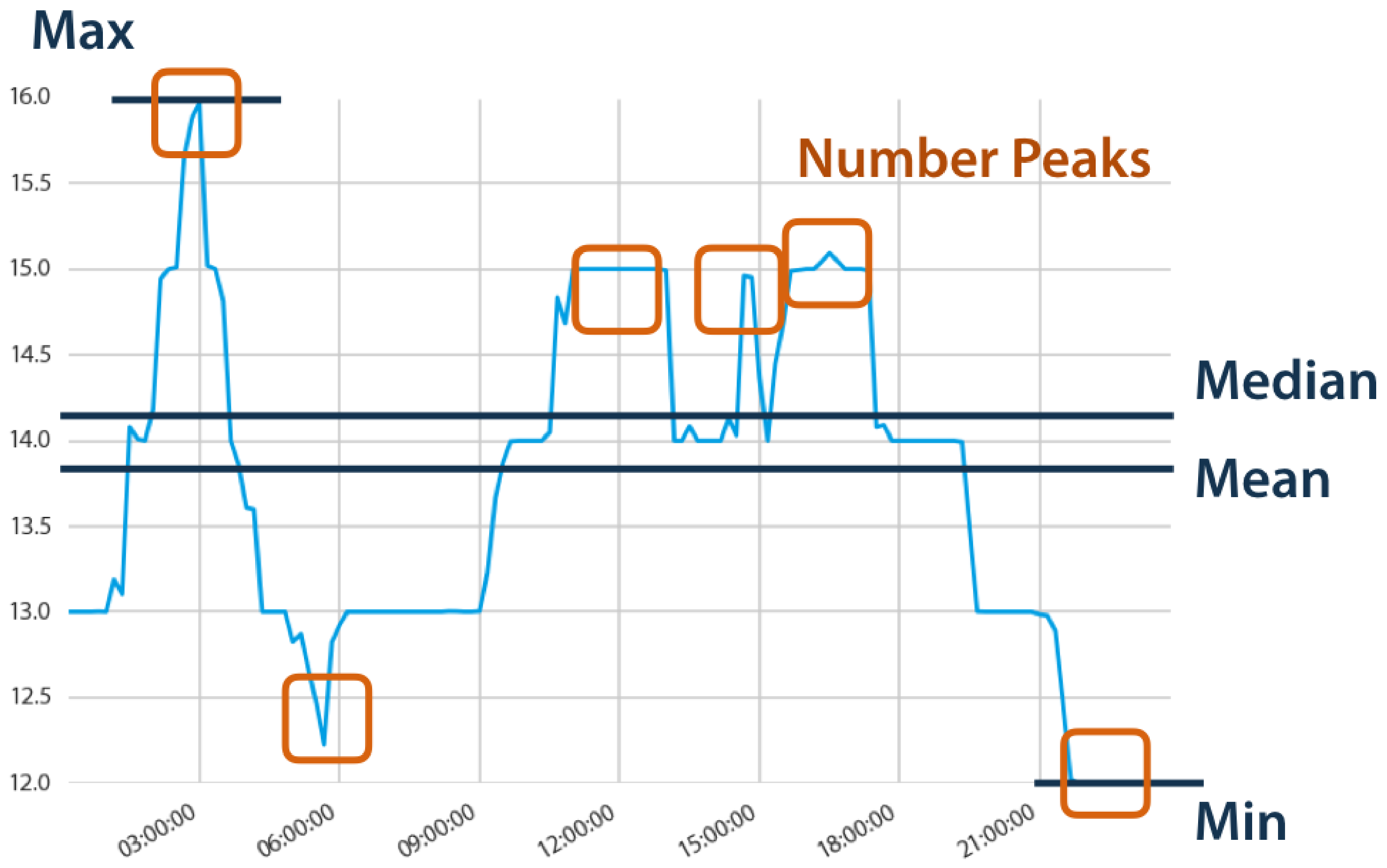
Learning representations

in self-supervised manner

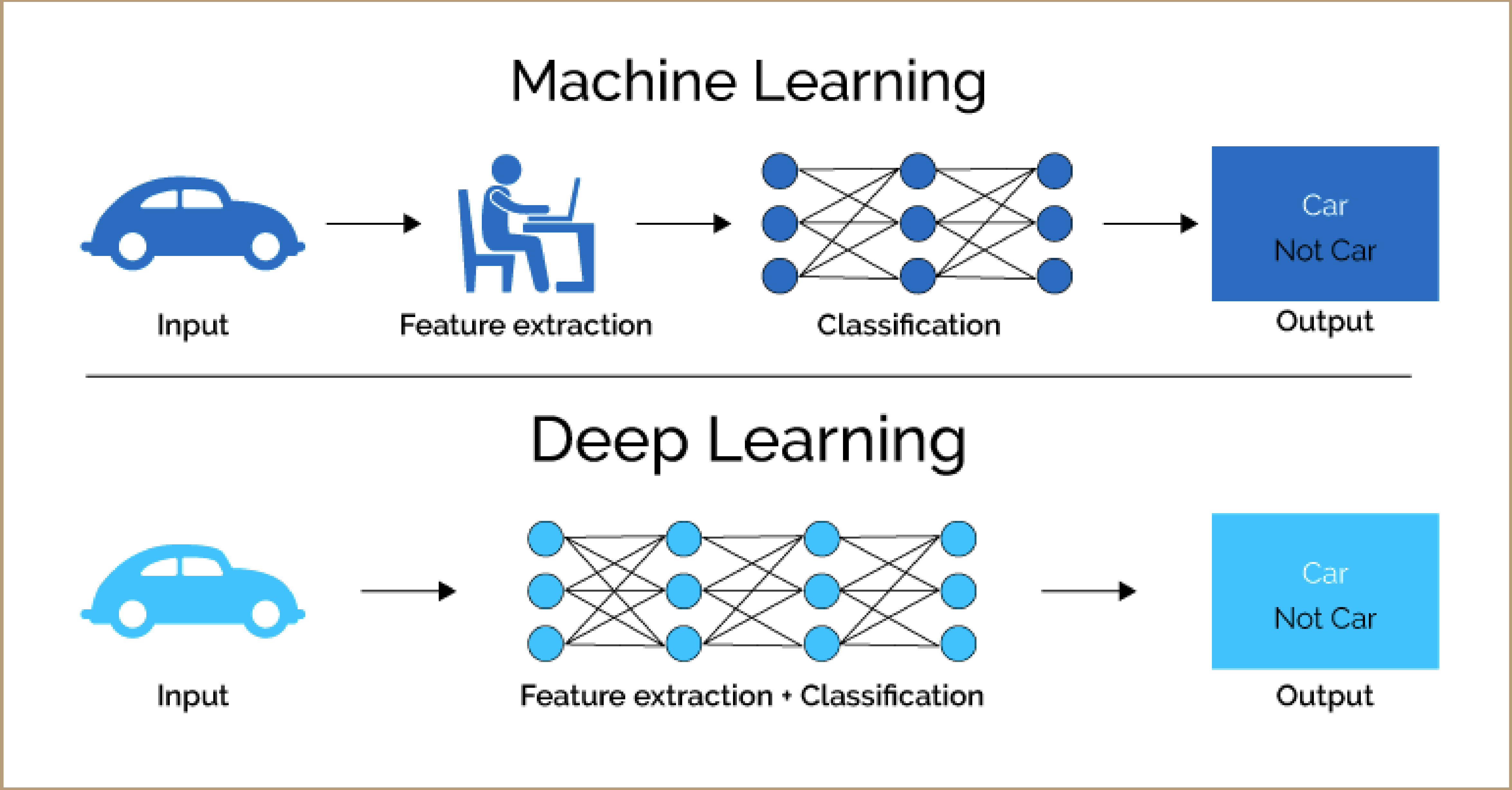


Representations

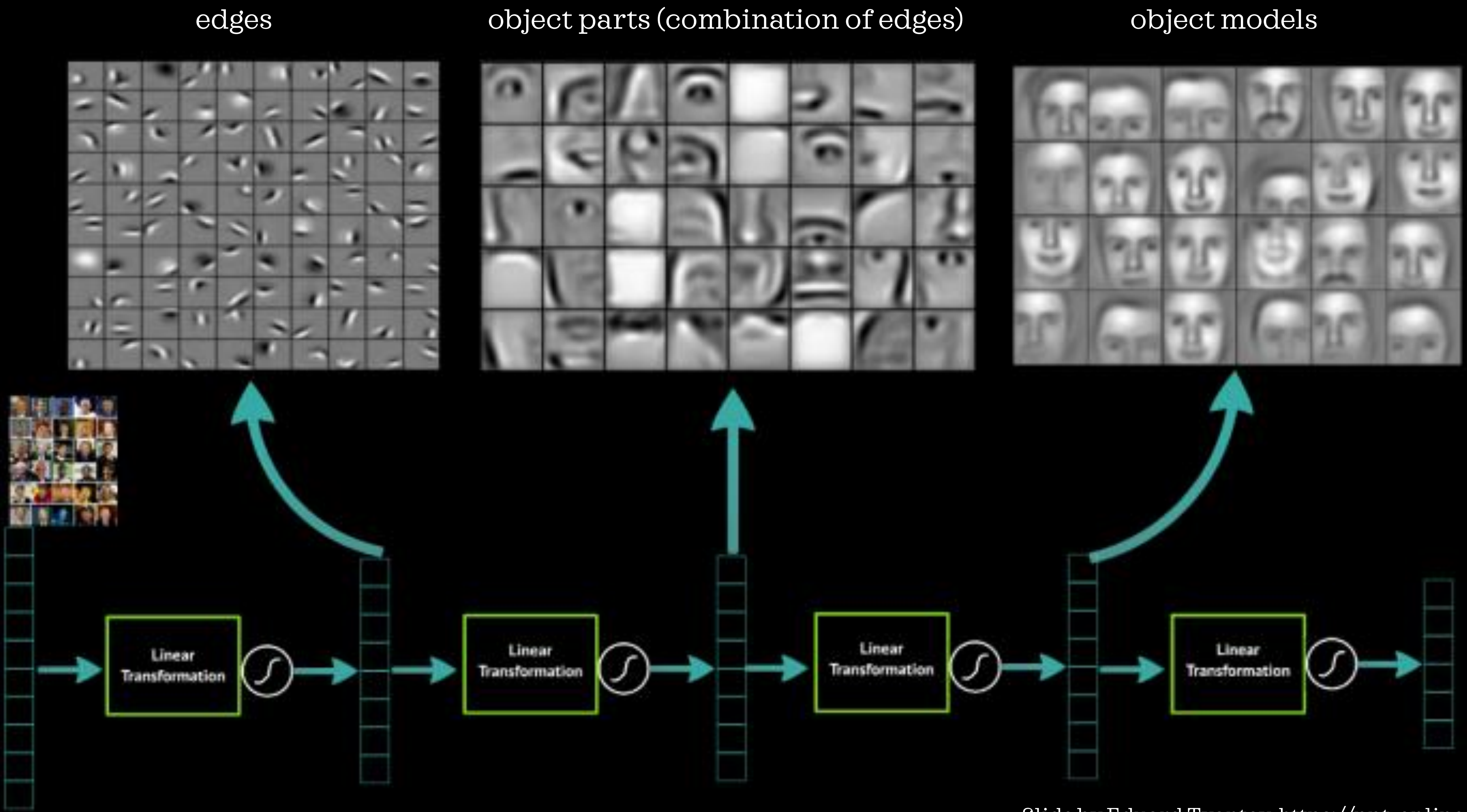
„Classical” feature extraction



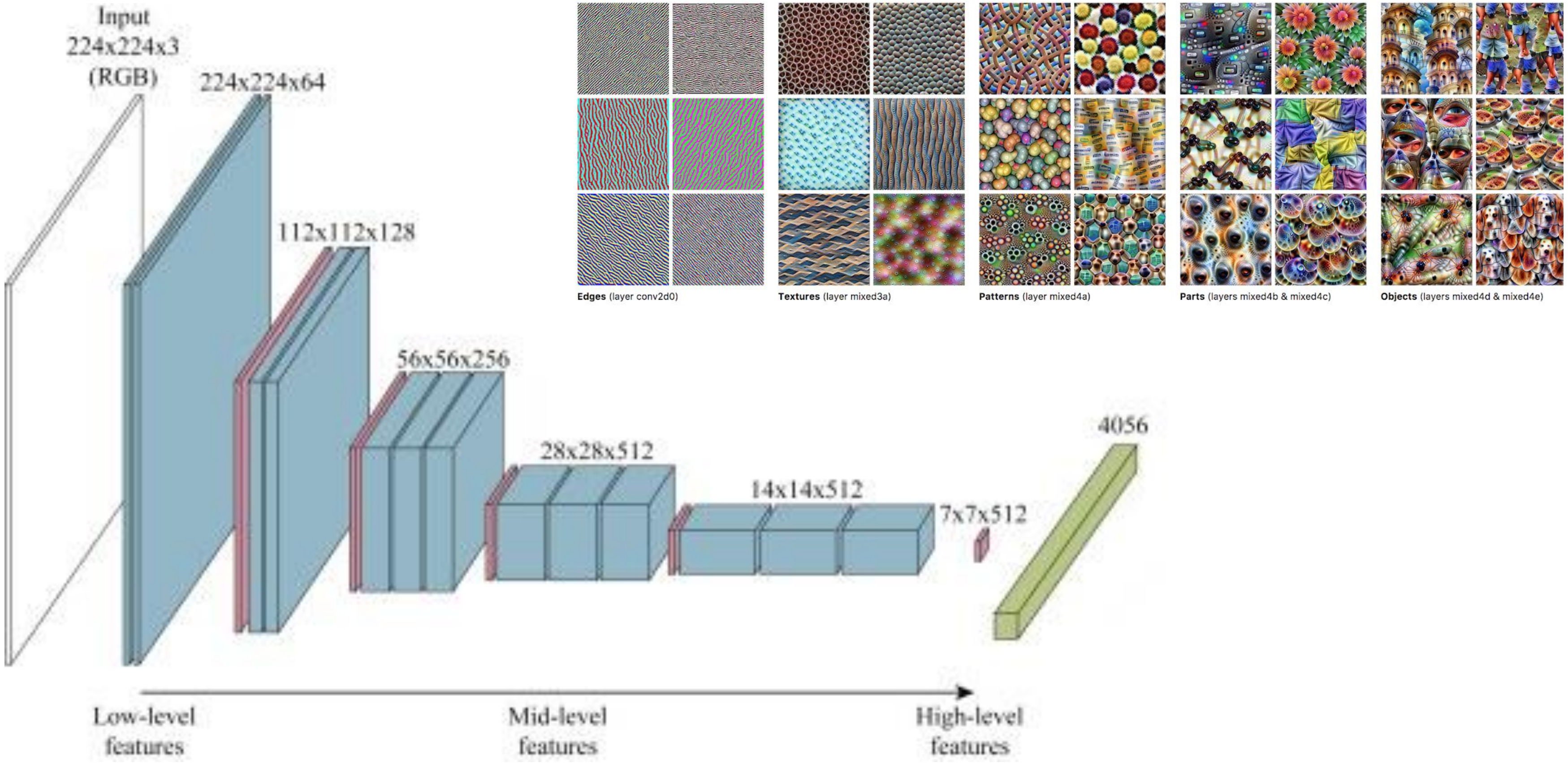
Time series can be represented by its features:
min, max, median, number of peaks etc.


Machine Learning
Car
Not Car
Input
Feature extraction
Classification
Output
Deep Learning
Car
Not Car
Input
Feature extraction + Classification
Output

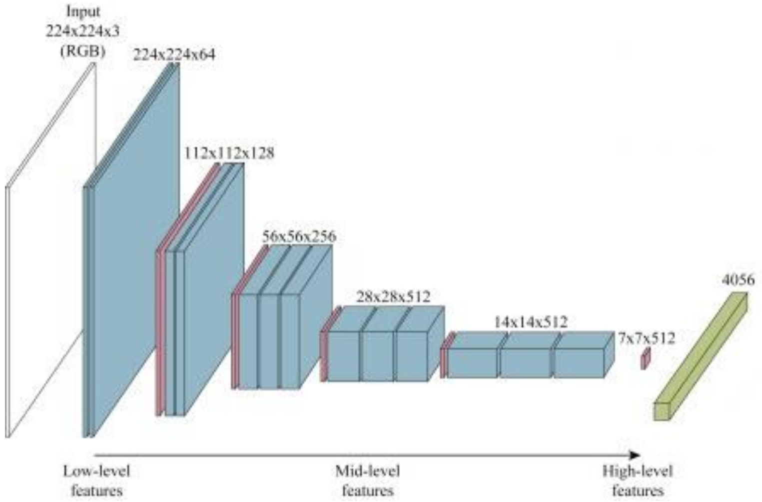
Deep Learning learns layers of features



Slide by Eduard Tyantov, https://ppt-online.org/354650



Deeper layers -> more complex features



Feature vector $x_1, x_2, \ldots, x_n$

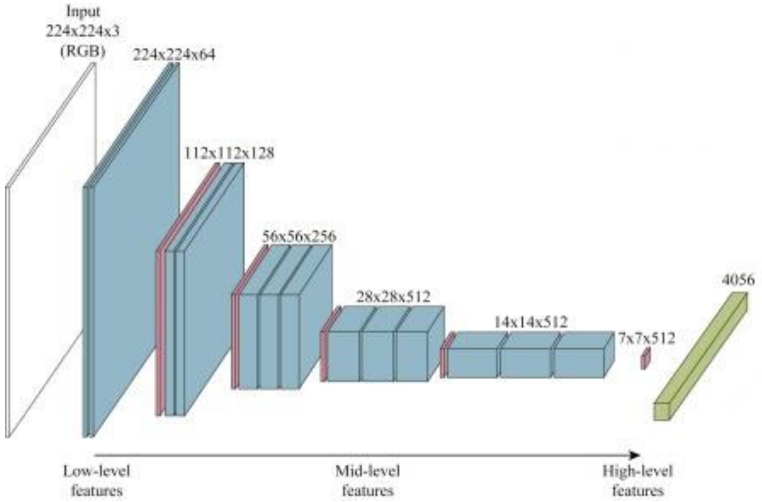
CLASSIFIER

Categories weights $w_0^{cat}, w_1^{cat}, \ldots, w_n^{cat}$

$w_0^{dog}, w_1^{dog}, \ldots, w_n^{dog}$

Categories scores $y_{cat} = w_0^{cat} + x_1 w_1^{cat} + \ldots + x_n w_n^{cat}$

$y_{dog} = w_0^{dog} + x_1 w_1^{dog} + \ldots + x_n w_n^{dog}$



Feature vector $x_1, x_2, \dots, x_n$

CLASSIFIER

Categories weights $w_0^{cat}, w_1^{cat}, \dots, w_n^{cat}$

$w_0^{dog}, w_1^{dog}, \dots, w_n^{dog}$

Categories scores $y_{cat} = w_0^{cat} + x_1 w_1^{cat} + \dots + x_n w_n^{cat}$

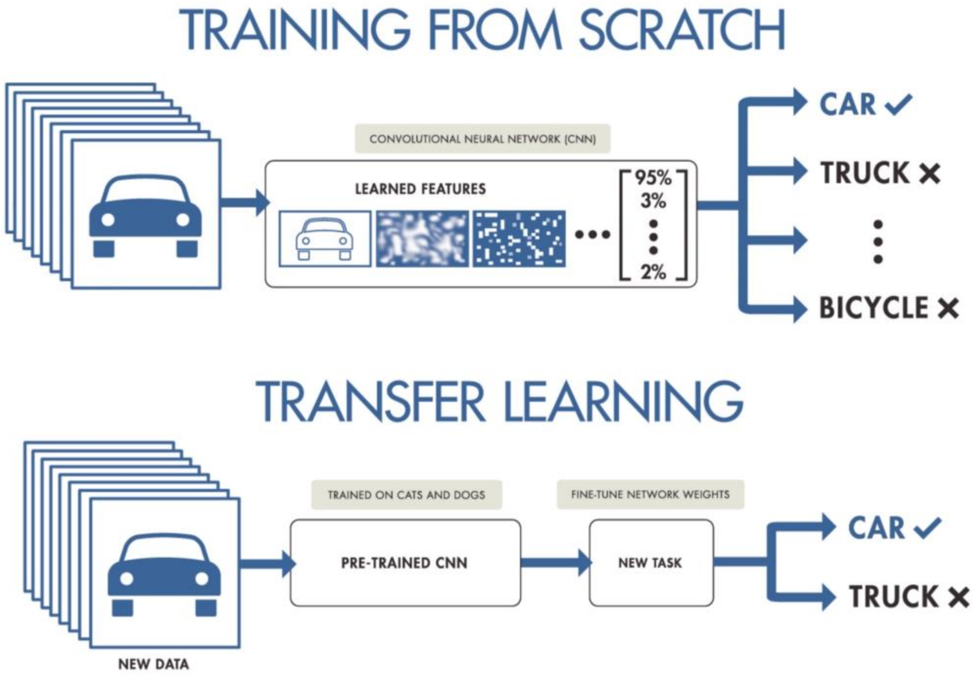
$y_{dog} = w_0^{dog} + x_1 w_1^{dog} + \dots + x_n w_n^{dog}$

construct representation
(„features extraction”)
often hard and universal part

Work on representations

often simpler and more specific part

Feature extractor „backbone” in transfer learning



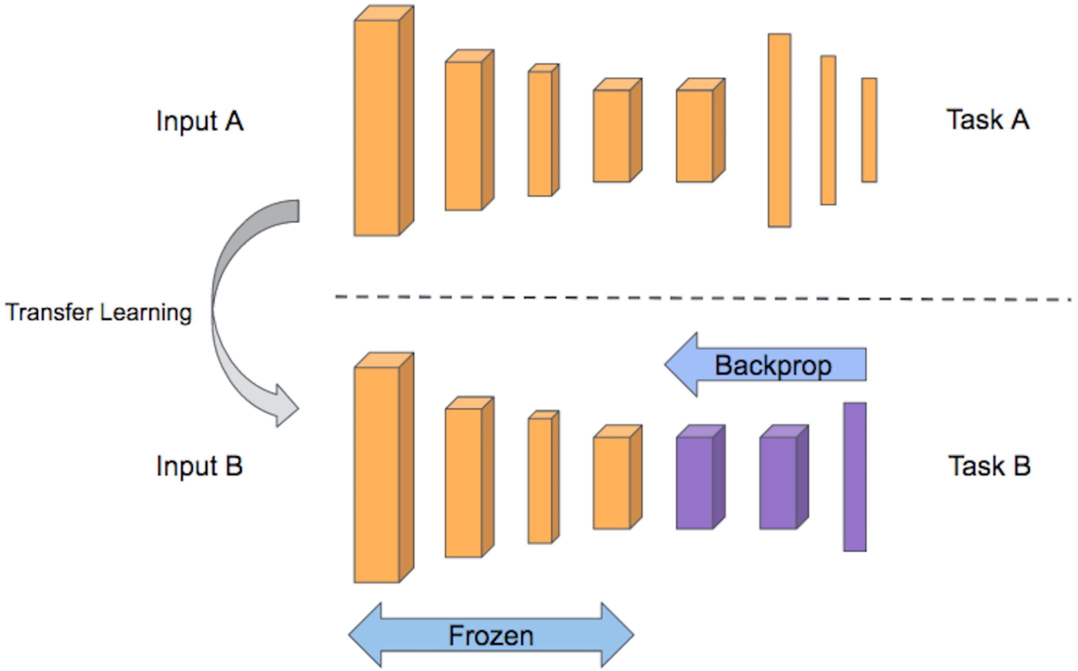
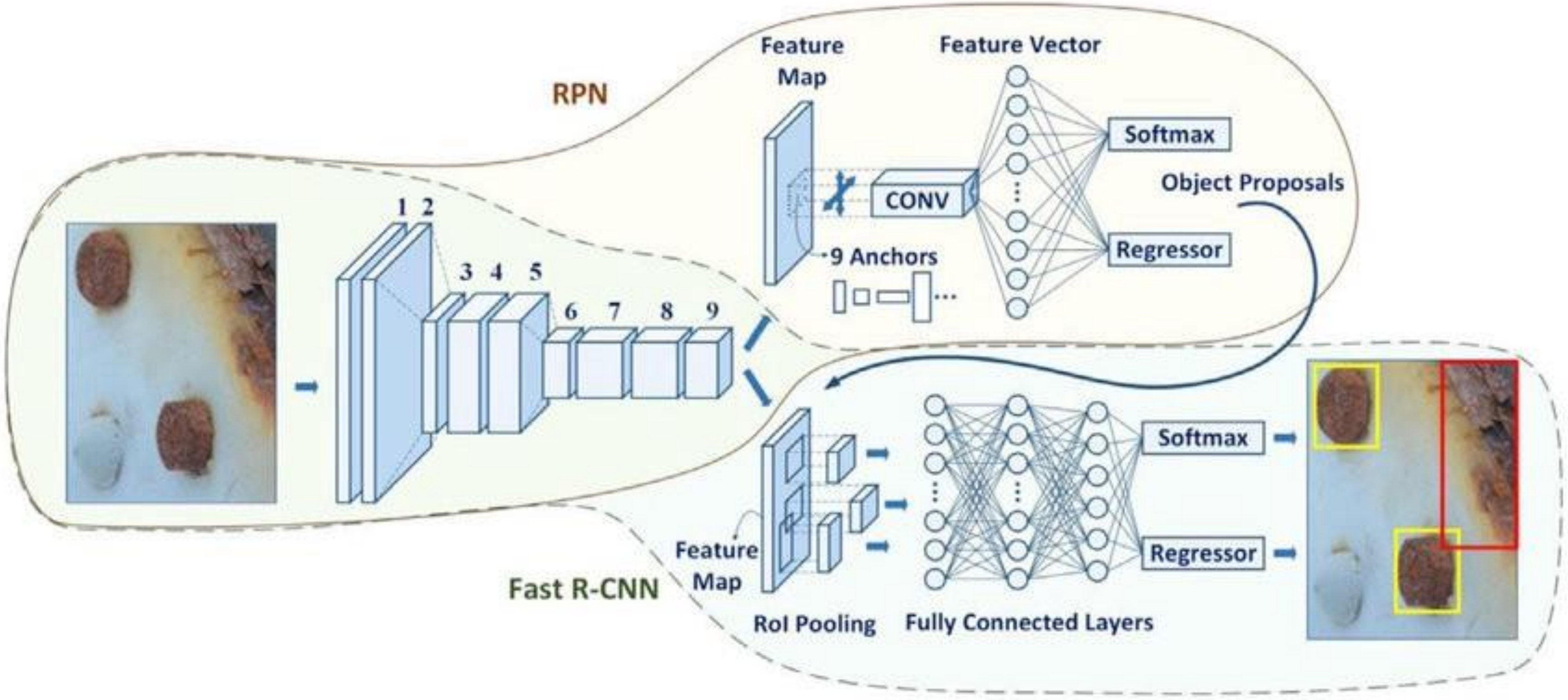


Image: https://medium.datadriveninvestor.com/introducing-transfer-learning-as-your-next-engine-to-drive-future-innovations-5e81a15bb567

Image: https://medium.com/@subodh.malgonde/transfer-learning-using-tensorflow-52a4f6bcde3e

Feature extractor „backbone” in detection



Cha, Y. J., Choi, W., Suh, G., Mahmoudkhani, S., & Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Computer-Aided Civil and Infrastructure Engineering, 33(9), 731-747.

Face recognition

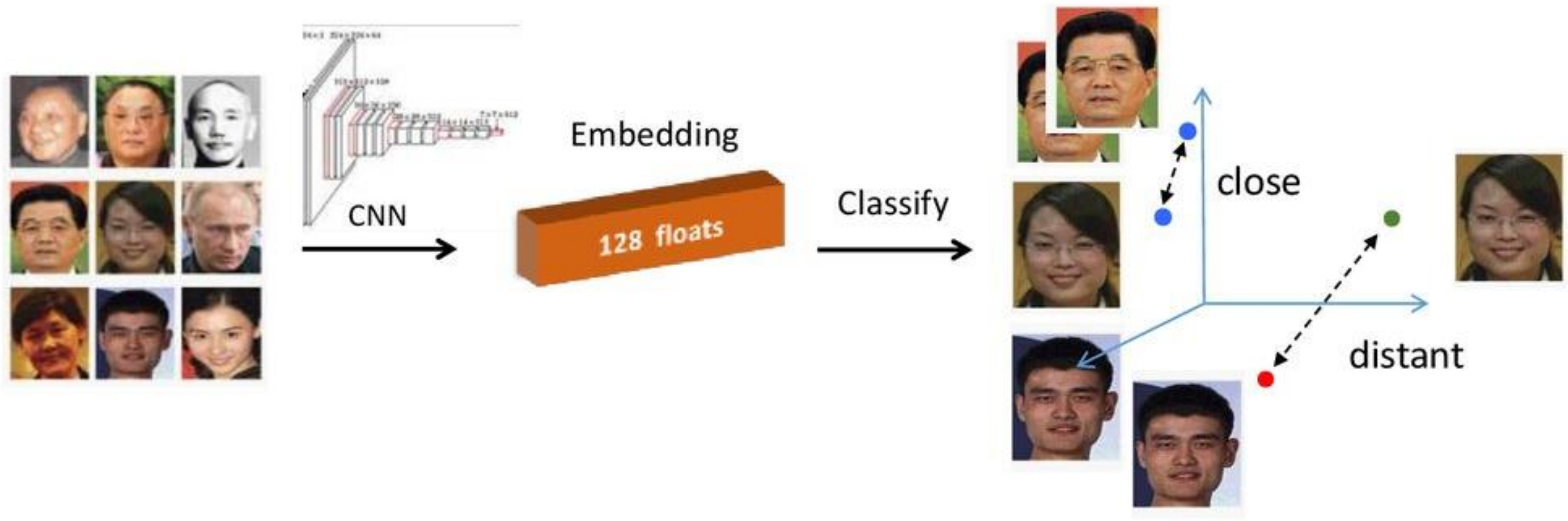
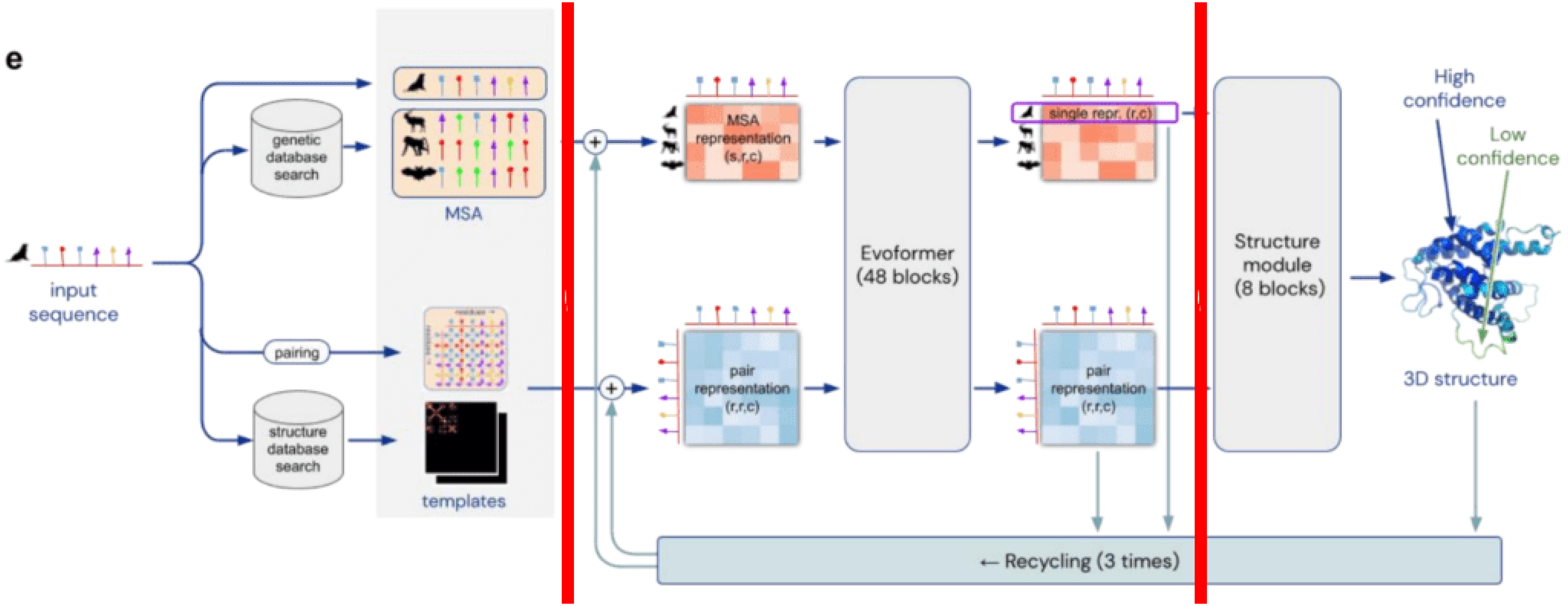


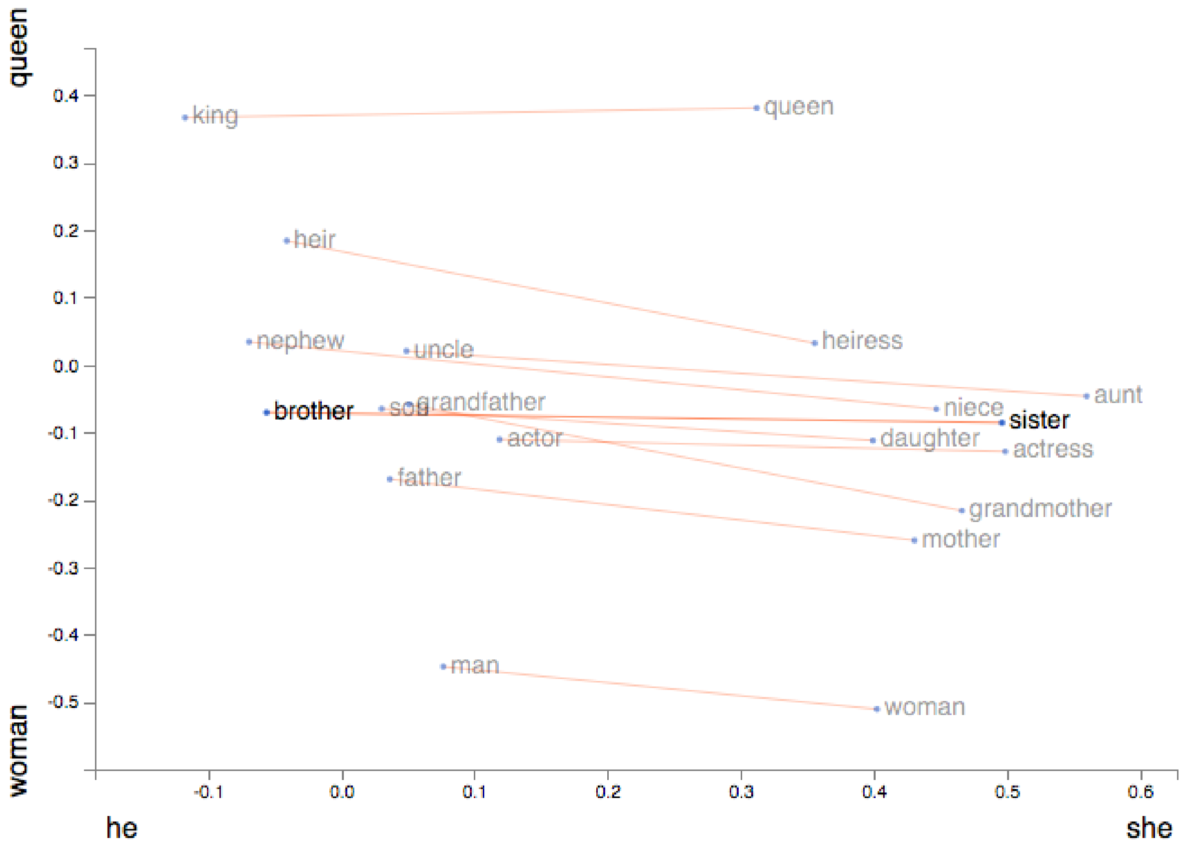
Image by Eduard Tyantov, https://ppt-online.org/354650

Alphafold



Improve representation leyer by layer

Language embeddings



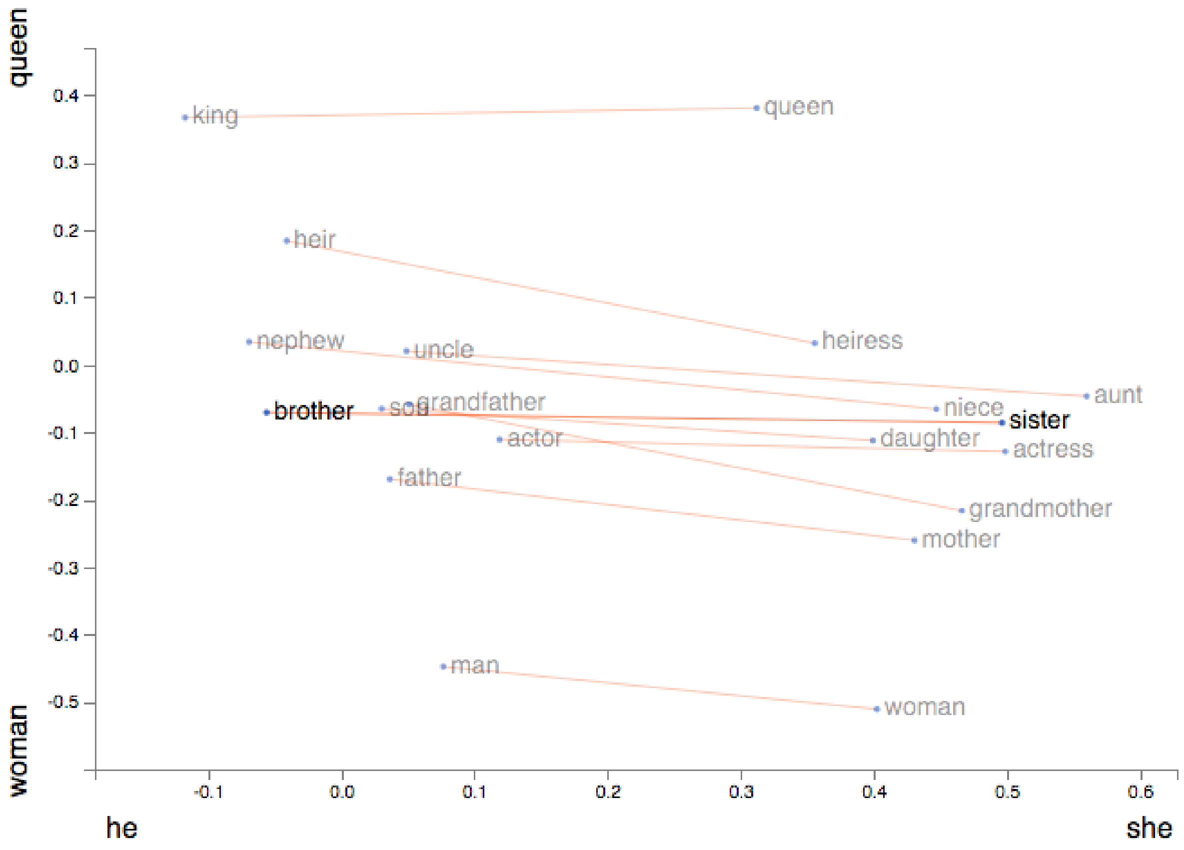
king - man + woman ≈ queen

biking - today + yesterday ≈ biked

Paris - France + Poland ≈ Warsaw

https://wiki.pathmind.com/word2vec

Language embeddings



king - man + woman ≈ queen

biking - today + yesterday ≈ biked

Paris - France + Poland ≈ Warsaw

Iraq - Violence ≈ Jordan

Human - Animal ≈ Ethics

President - Power ≈ Prime Minister

Library - Books ≈ Hall

https://wiki.pathmind.com/word2vec

Representations

Protein structure

Language

Time series

Face recognition

Images

...

Representations

Much smaller than input space

Contain information relevant for the task

Unreadable – black box

We may work in latent space:

- Similar input maps to similar representation (e.g., different viewpoints)
- Similar representations give similar output (VAE)
- Distribution in latent space
- Sometimes directly interpretable directions

Representations

Much smaller than input space

Contain information relevant for the task

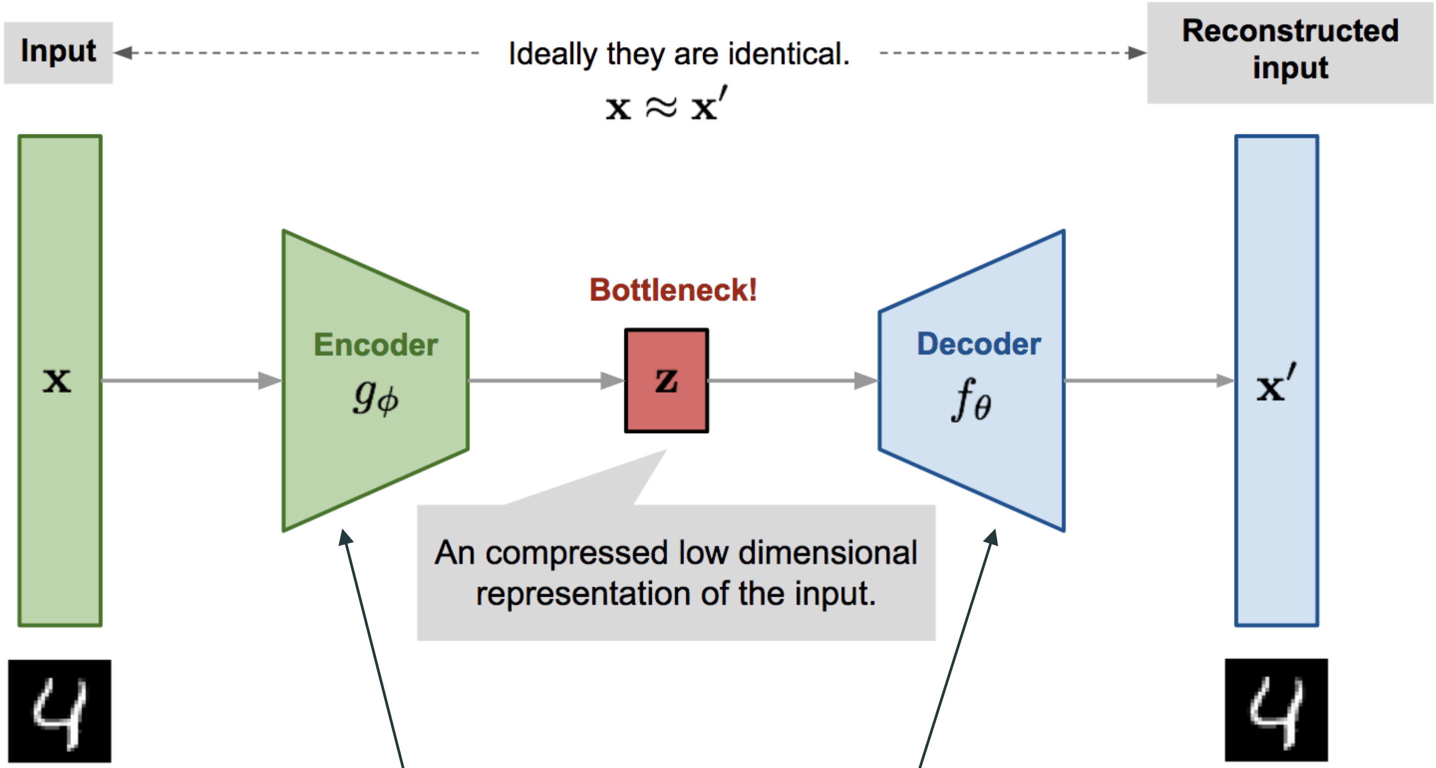
Unreadable – black box

We may work in latent space:

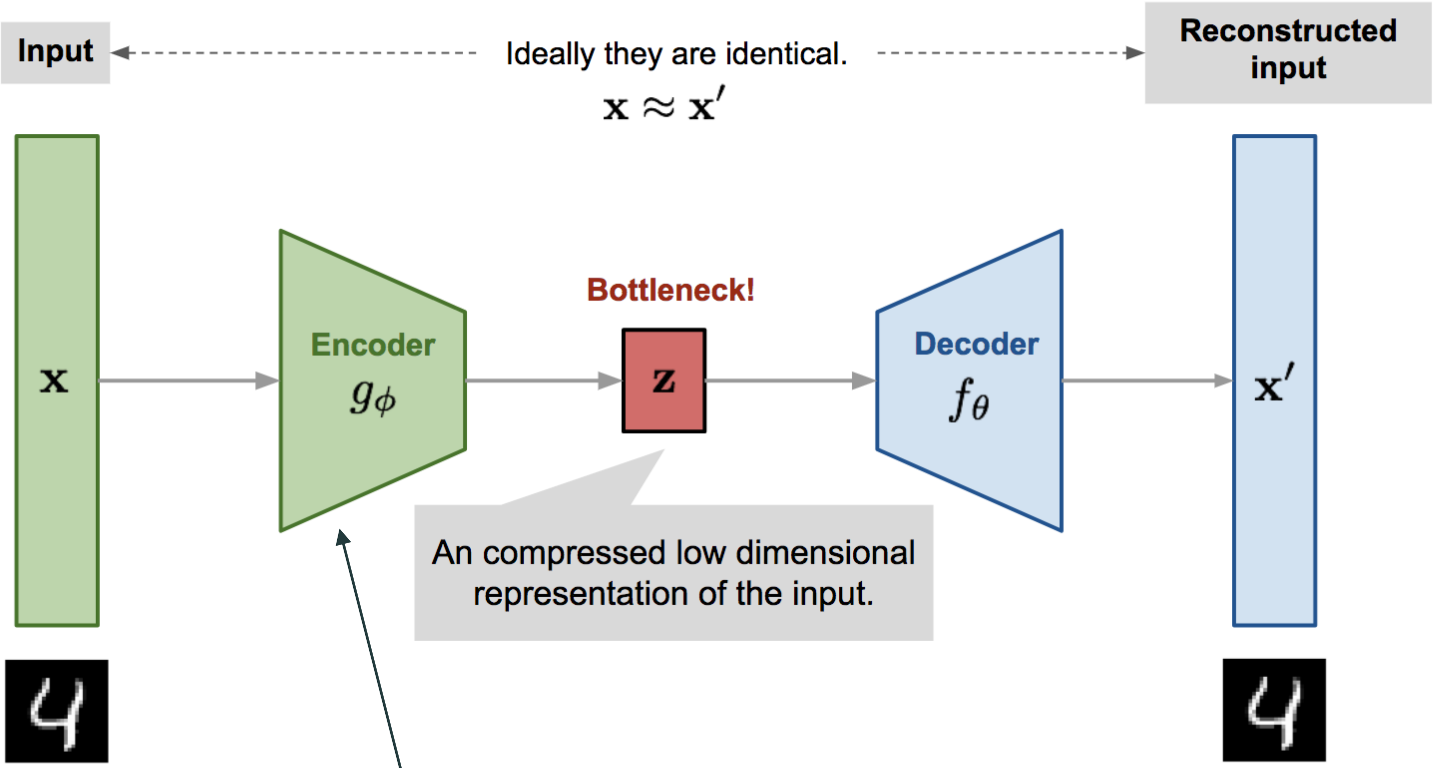
USUALLY BY ADDING PENALTY (extra term in loss function)

- Similar input maps to similar representation (e.g., different viewpoints)
- Similar representations give similar output (VAE)
- Distribution in latent space
- Sometimes directly interpretable directions

Autoencoders

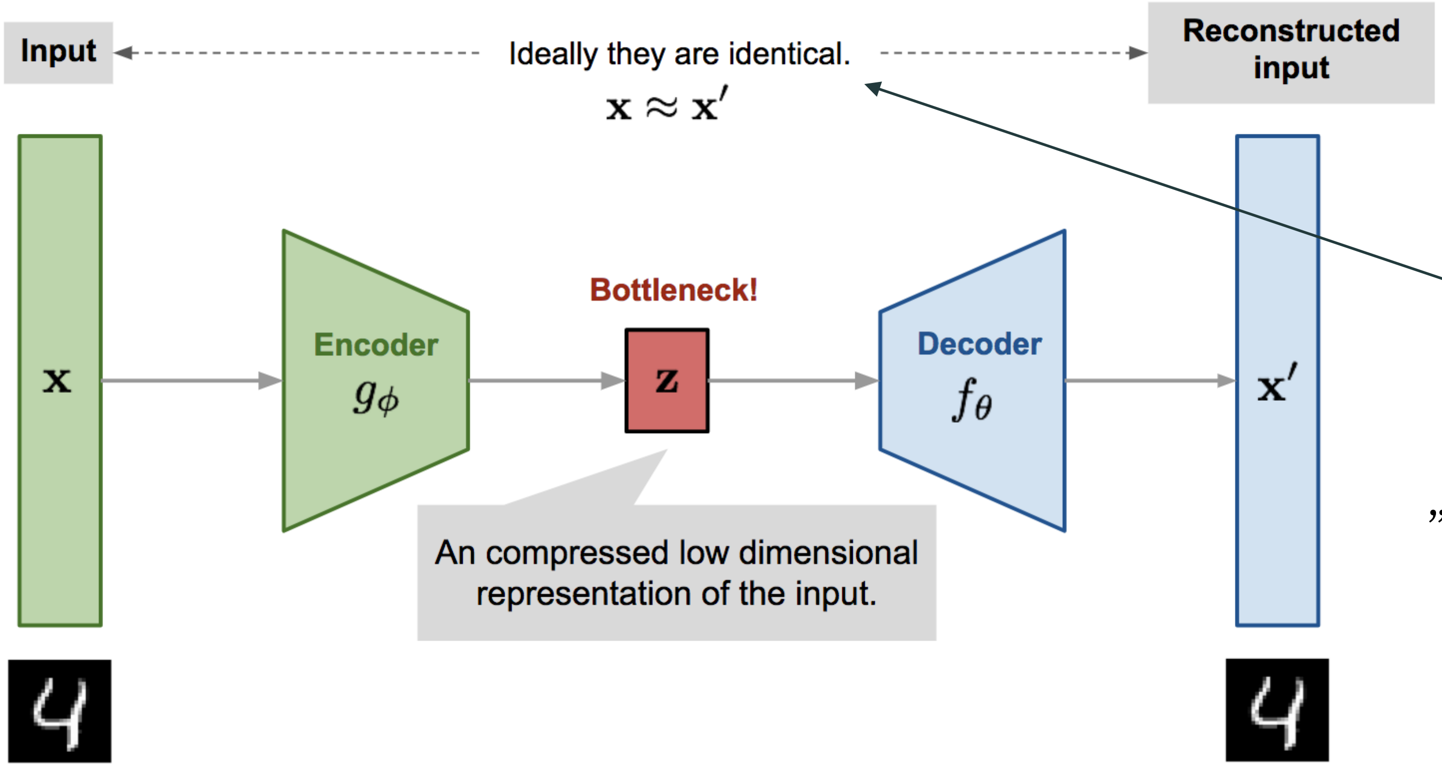


Two distinct neural networks
(together: autoencoder)



Model type adequate for input

Input:
- image
- tabular data
- time series
- …

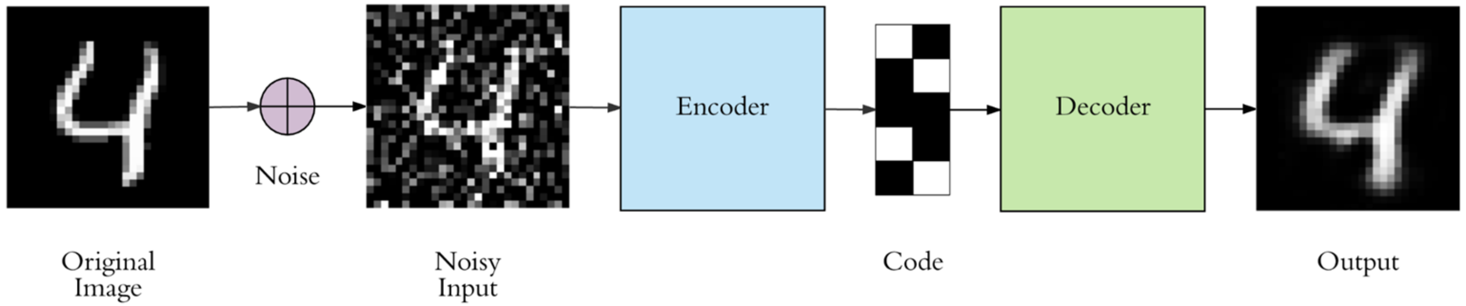


Does not require labeled data (self-supervision)

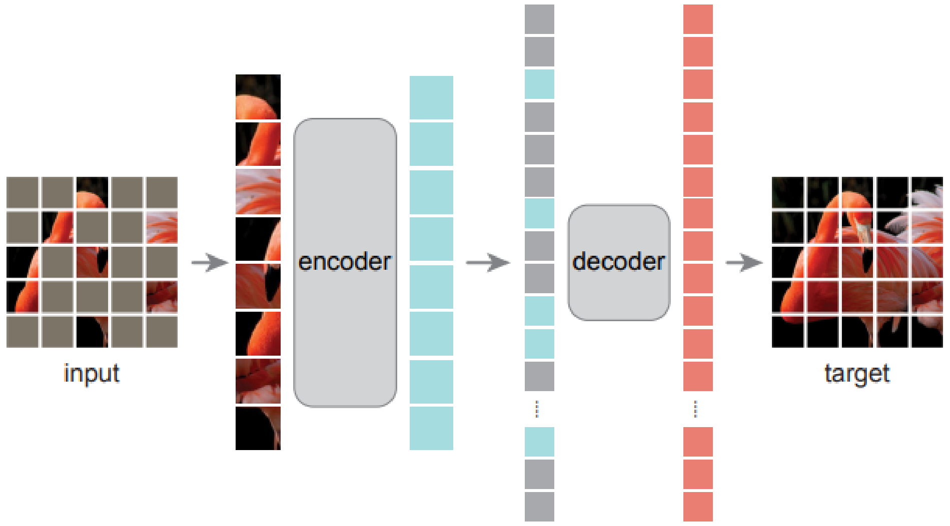
„Similarity" measure may be tricky

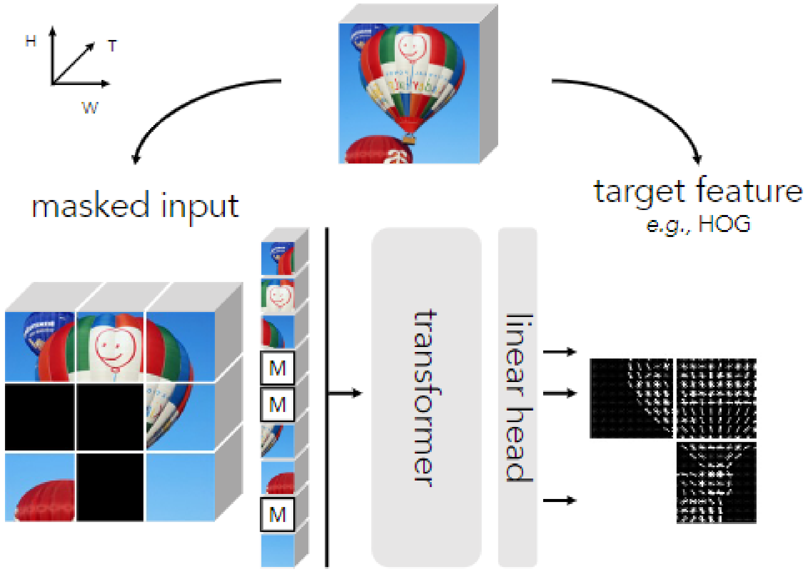
Target different from the input

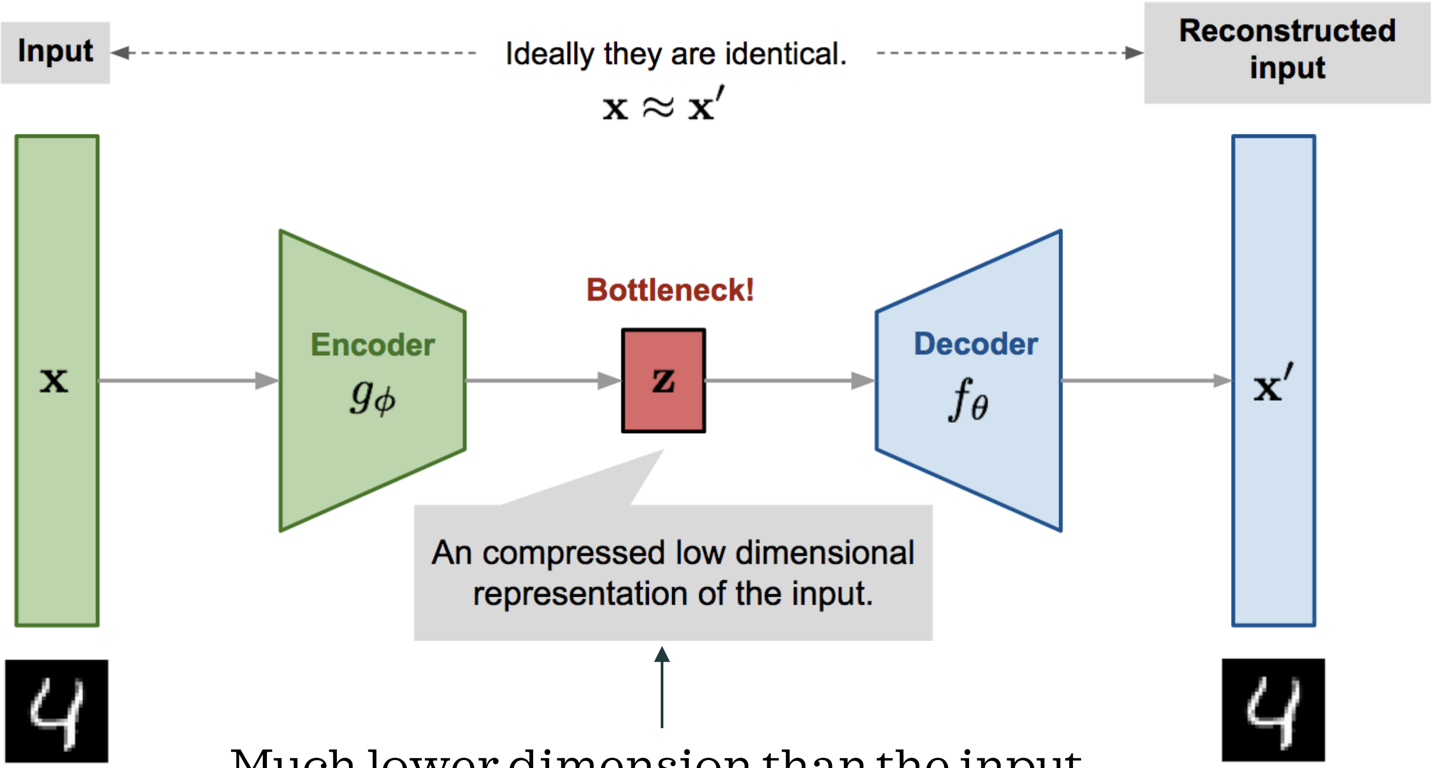
denoising



pretraining on masked images

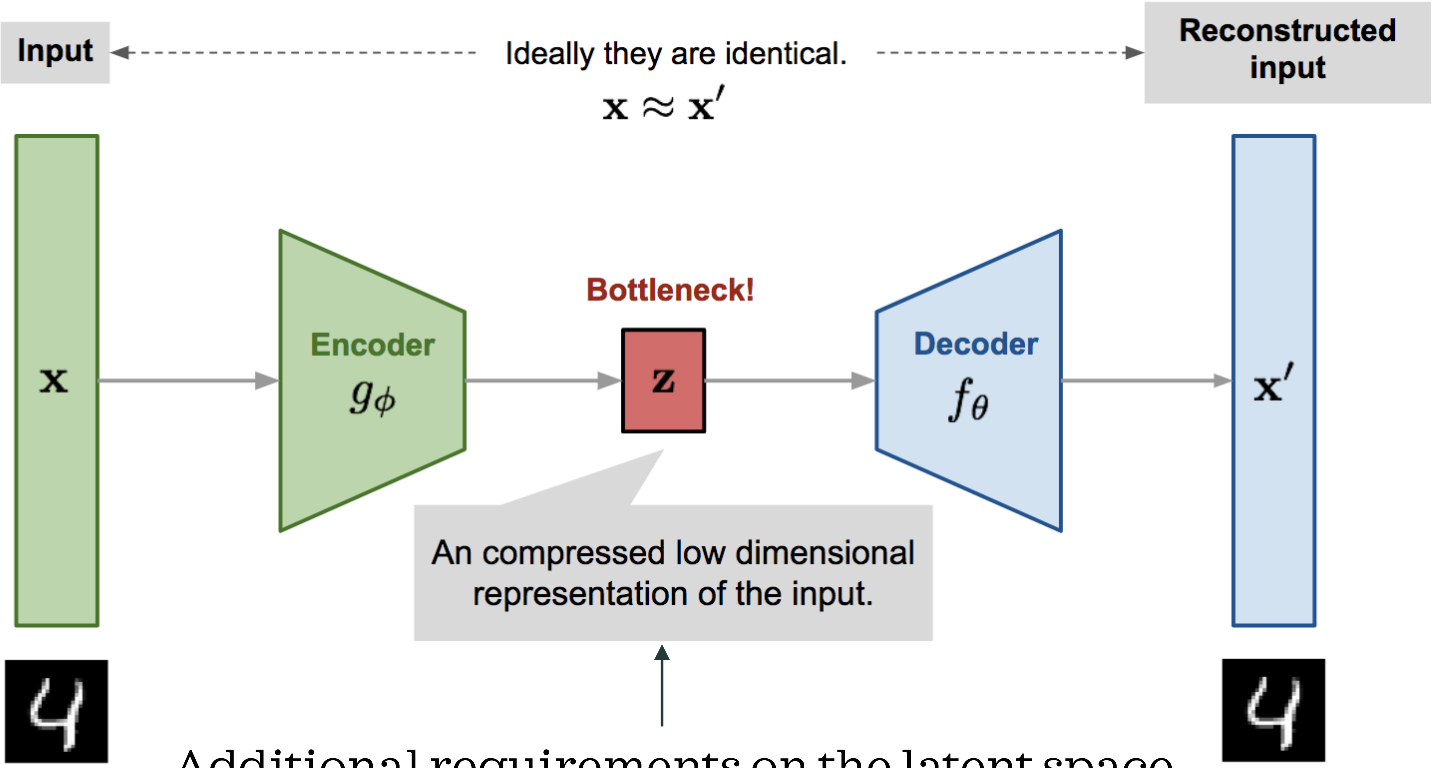






Much lower dimension than the input,
yet most of relevant information is present

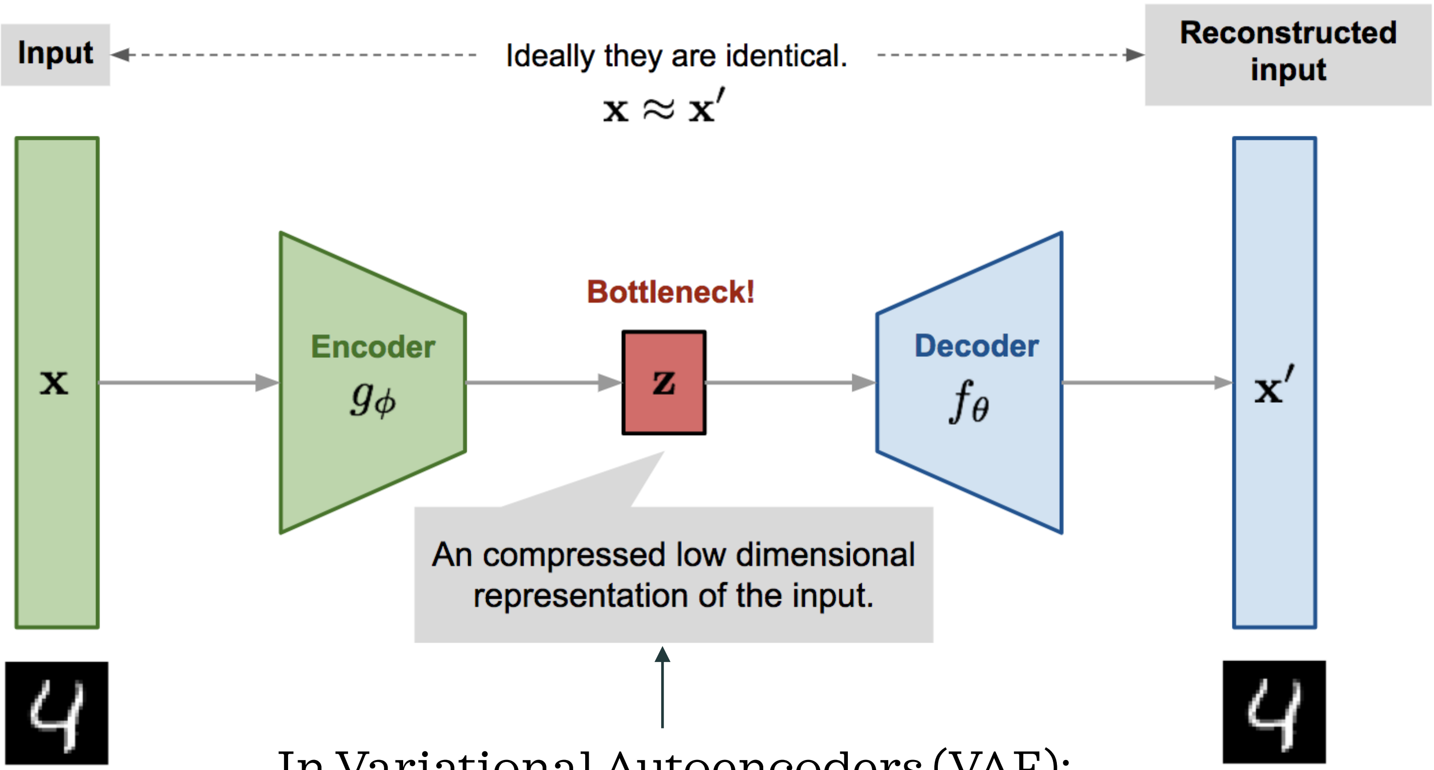
Model learns the effective coding (compression)
for given data



Additional requirements on the latent space may be given, such as:

- Distribution of latent space representations
- similar latent (z) -> similar reconstruction (x')

Requirements are usually imposed by adding relevant loss terms



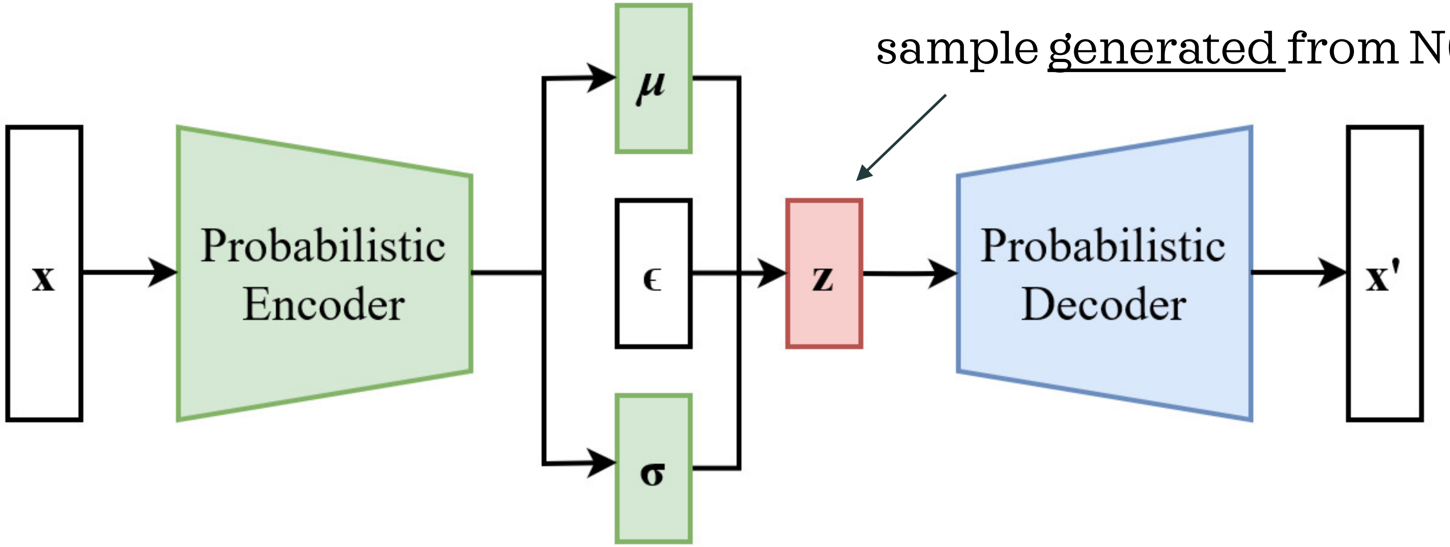
In Variational Autoencoders (VAE):

- Distribution in the latent space representations is preferred to be Normal(0,1)
- Latent space vector for reconstruction (decoding) is <u>sampled</u> from vicinity of encoded vector z

VAE

mean of distribution

sample generated from N($\boldsymbol{\mu}$, $\boldsymbol{\sigma}$)



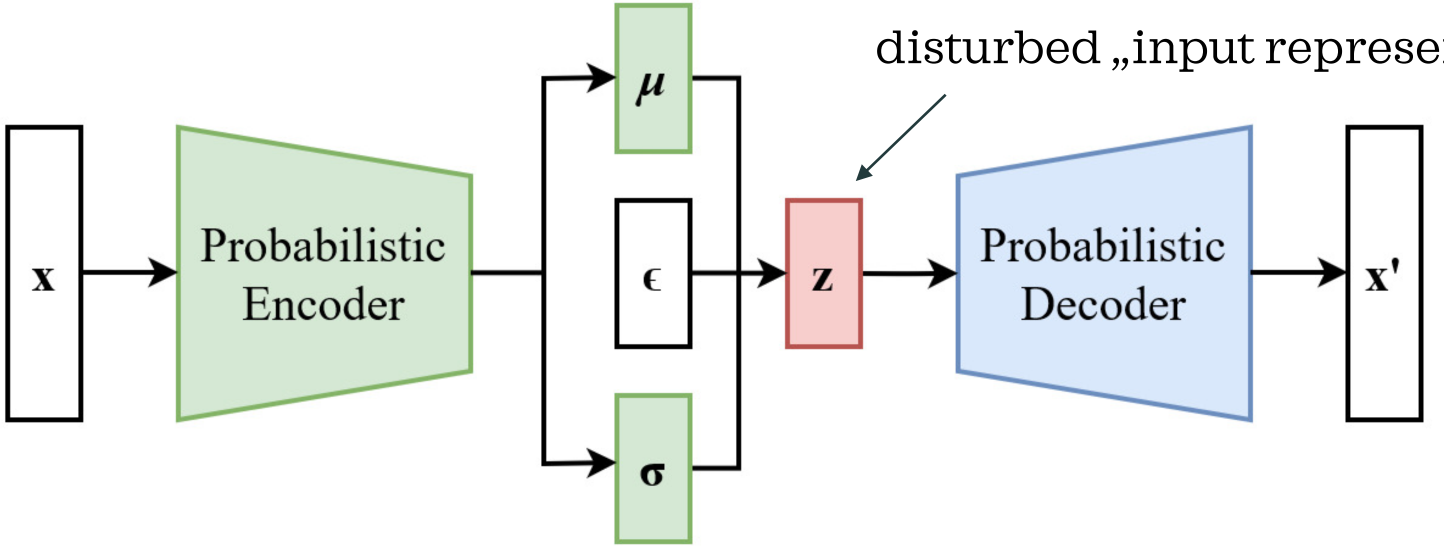
standard deviation

Loss = reconstruction loss + penalty for $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ deviation from N(0,1)

VAE

„input representation”

disturbed „input representation”



„noise strength”

Loss = reconstruction loss + penalty for $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ deviation from N(0,1)

Applications

Anomaly detection

Pretraining

Denoising

Downstream analysis (dimensionality reduction), VAE preferred :

- Visualization
- Clustering
- Any model with reduced number of features

Generative model (VAE)

Today workshop

Workshop contents

PyTorch model recapitulation

Introduction to GPU computing

Autoencoder step by step

Anomaly detection

KNN on latent space

Denoising autoencoder

Imposing latent space distribution

Workshop aims

Feel confident with building and training models in PyTorch

Hands-on experience autoencoders

Be able to use AE for anomaly detection

Get feeling of latent space (representations)

Good luck!