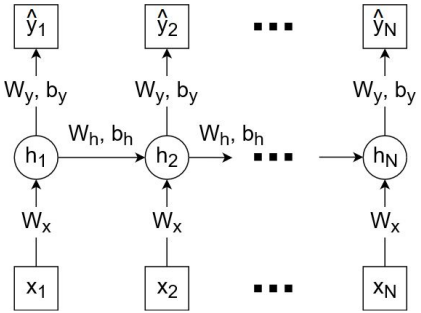
Deep Network Development

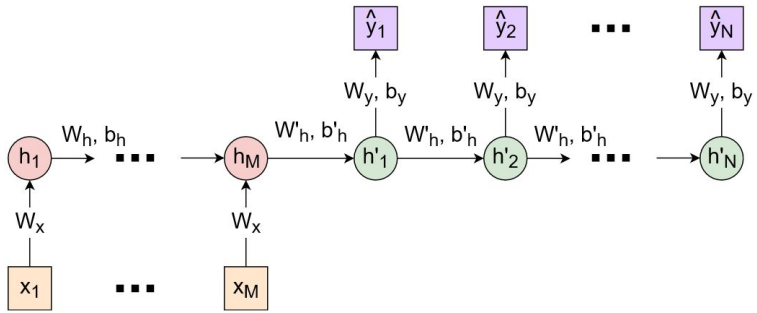
Lecture #12

Viktor Varga
Department of Artificial Intelligence, ELTE IK

Previously - Problems with RNNs

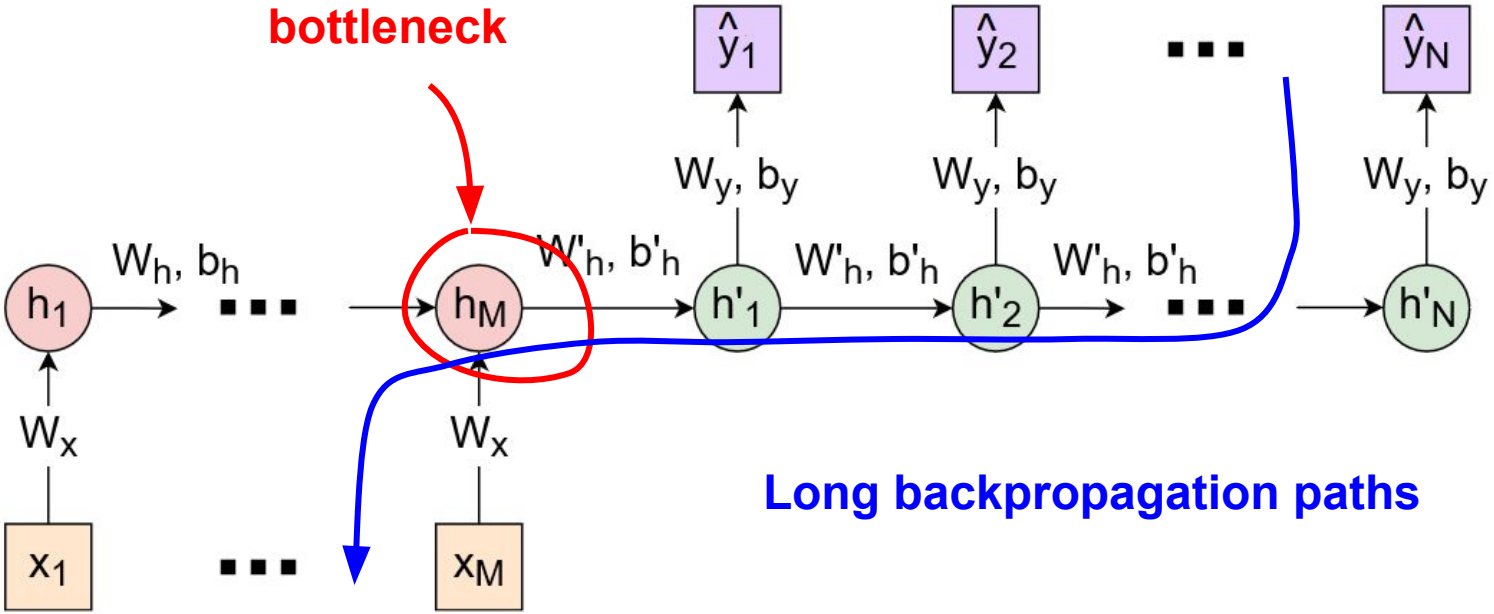
1. The **implementation is inefficient** when the sequence length is variable or too long.
2. **Sequential data processing is not ideal** for learning / identifying complex relationships.
3. Excessively **long backpropagation paths** → Unstable gradient problem.





Previously - Problems with RNNs

“M → N” RNN variant for machine translation (Seq2seq, 2014)



Previously - Attention-based models

We **estimate how important each element of the input is** for estimating the next element of the output.

We **pay attention** to the elements **deemed important** when generating the next output.

$$\forall_{t,t'} : a_{t,t'} := f([h'_{t-1}, h_{t'}])$$

$$\alpha_{t,t'} := \text{softmax}_{t'}(a_{t,t'})$$

$$h'_t = g([h'_{t-1}, \sum_{t'} \alpha_{t,t'} \cdot h_{t'}])$$

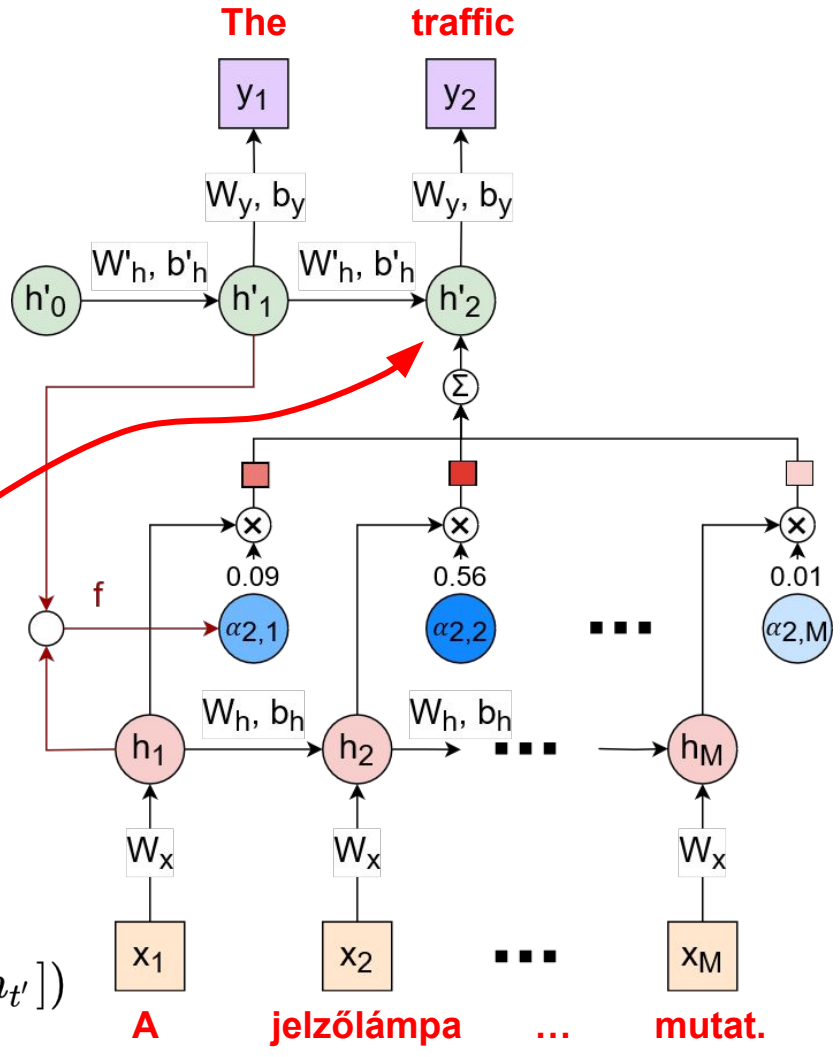
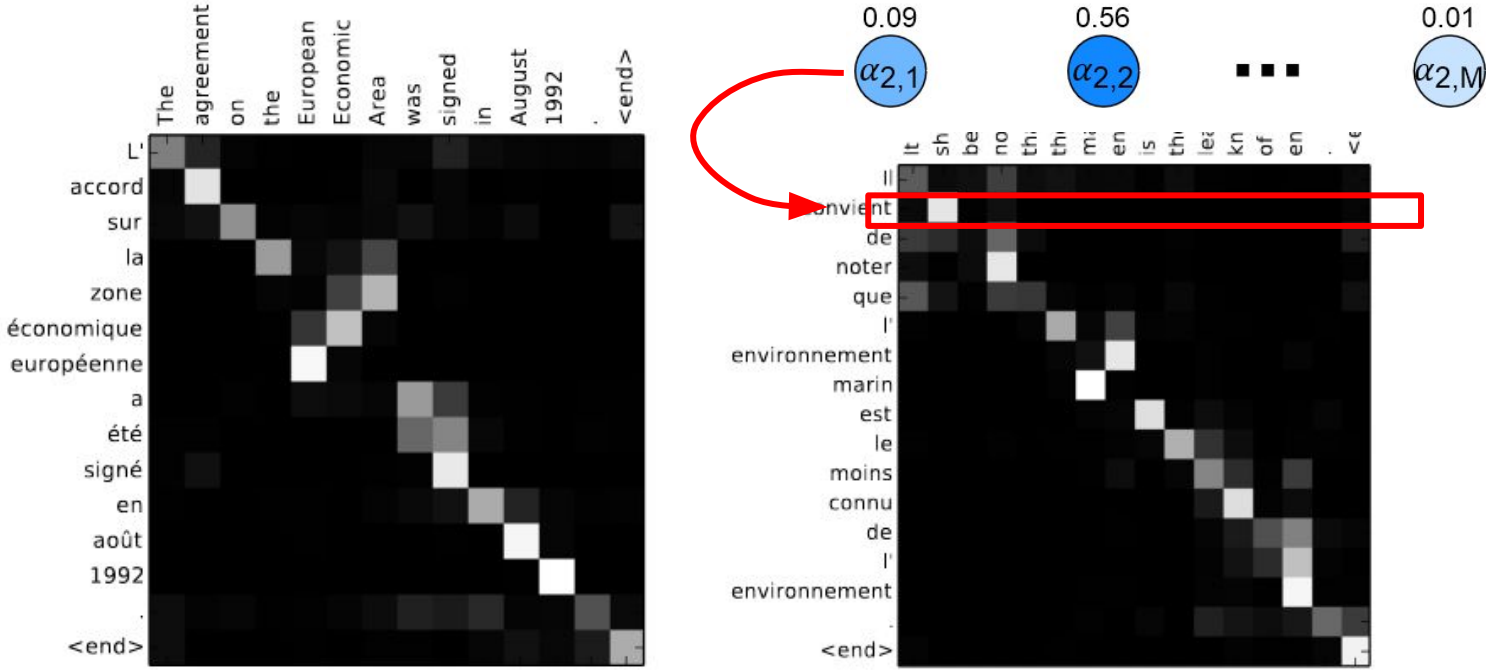


Image source: Bahdanau et al., 2015

Previously - Attention-based models, example

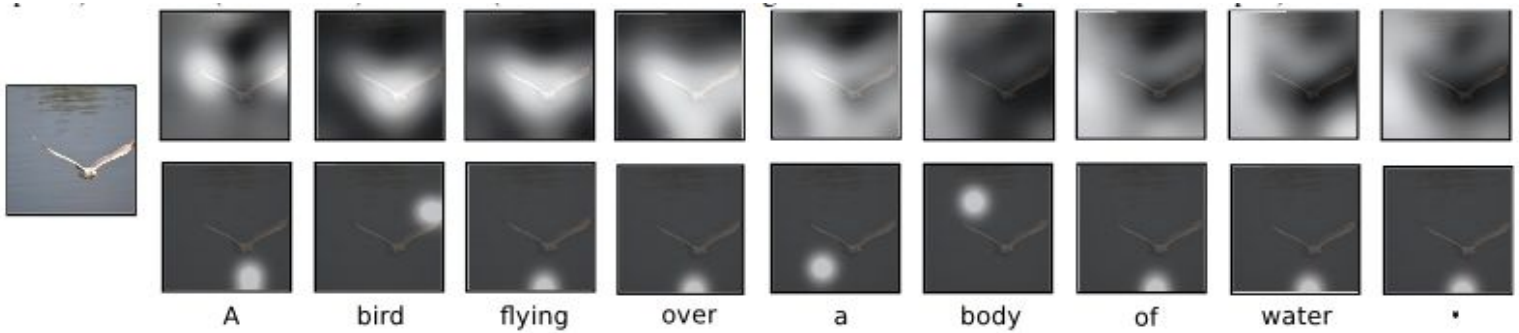


Element (i, j) of the matrix: The attention score of h'_{i-1} and h_j.
I.e., "how strongly are they related?"

Image source: Xu et al., 2015

Previously - Attention-based models

An example application: Image captioning using an attention-based model placed on top of a convolutional neural network.

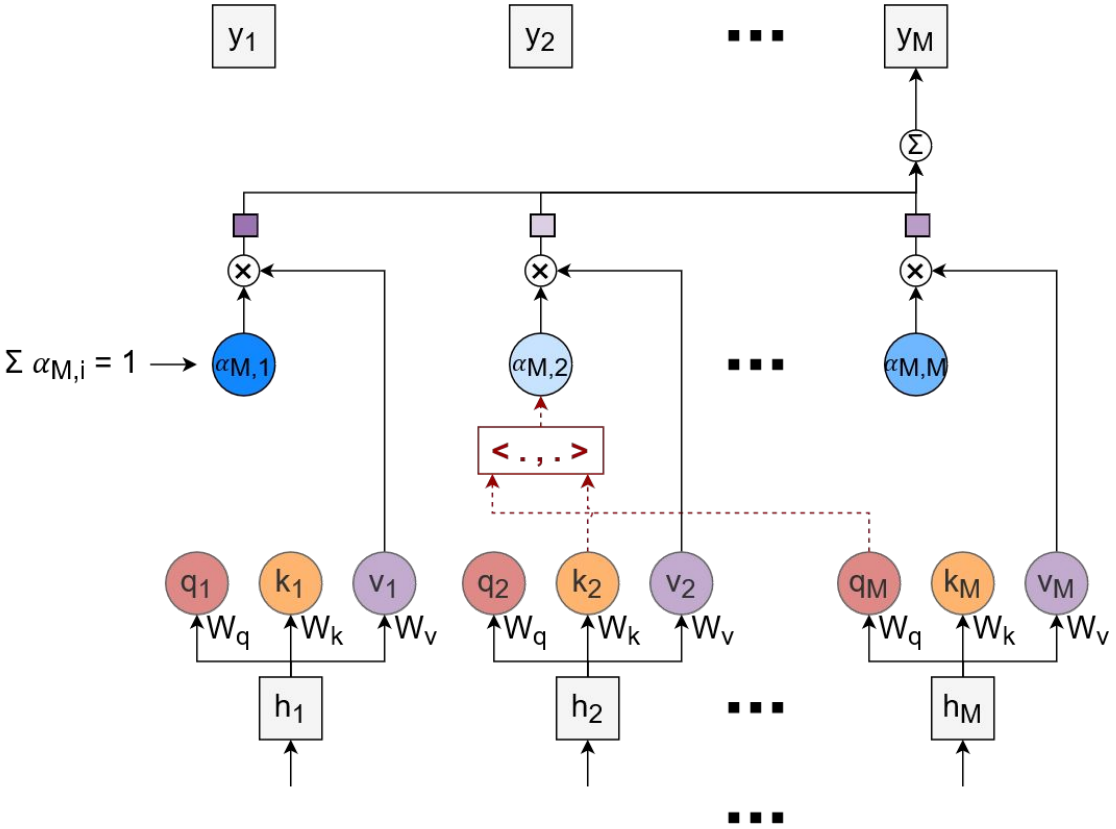


Previously - Self-attention layer

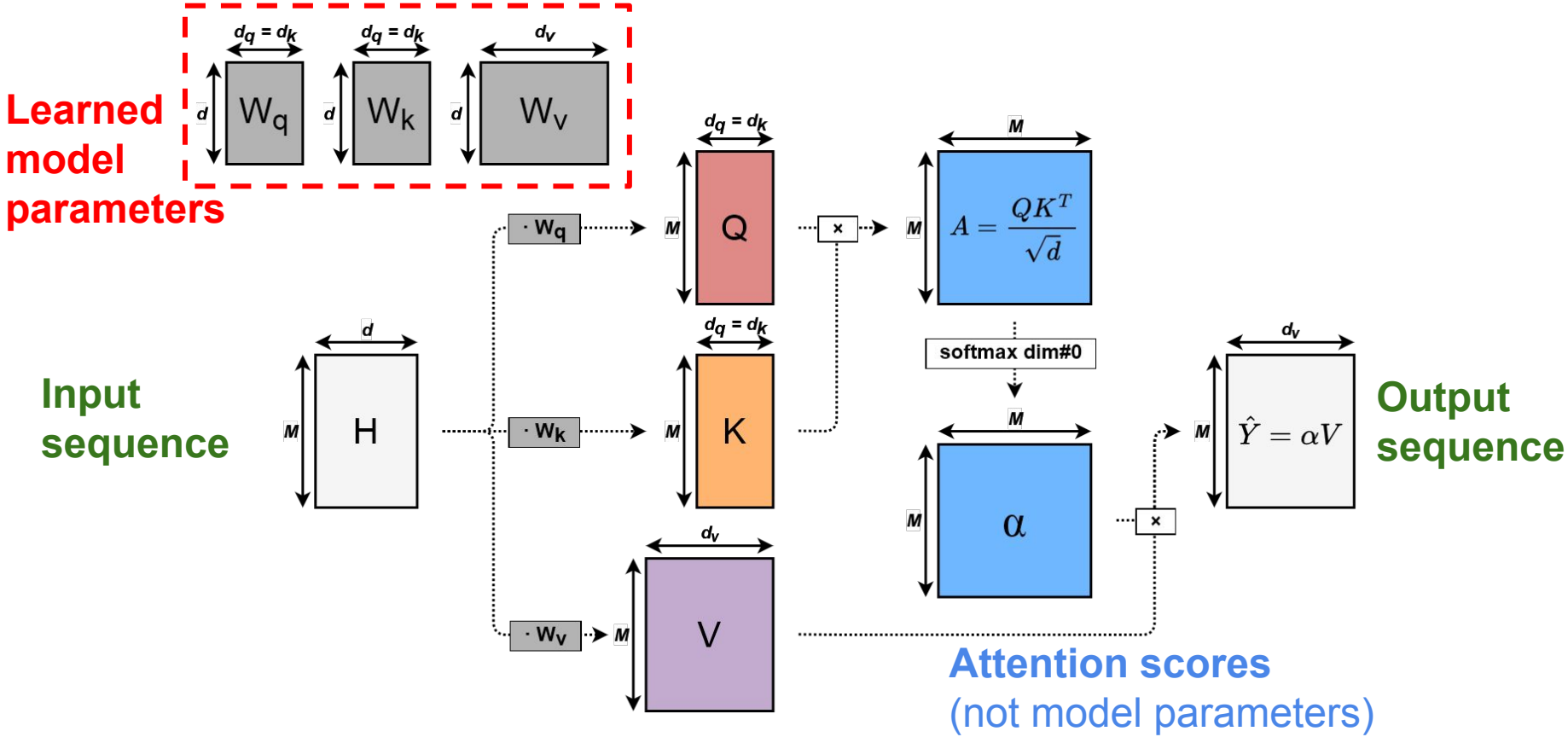
Self-attention layer

A stackable, attention-based layer that implements sequence processing.

$$q_i = W_q h_i \quad \text{Query}$$
$$k_i = W_k h_i \quad \text{Key}$$
$$v_i = W_v h_i \quad \text{Value}$$
$$\alpha_{i,j} = \text{softmax}_i\left(\frac{\langle q_i, k_j \rangle}{\sqrt{d_q}}\right)$$
$$\hat{y}_i = \sum_{j=1}^{M} \alpha_{i,j} \cdot v_j$$



Previously - Self-attention layer, vectorized form



Previously - Self-attention layer, vectorized form

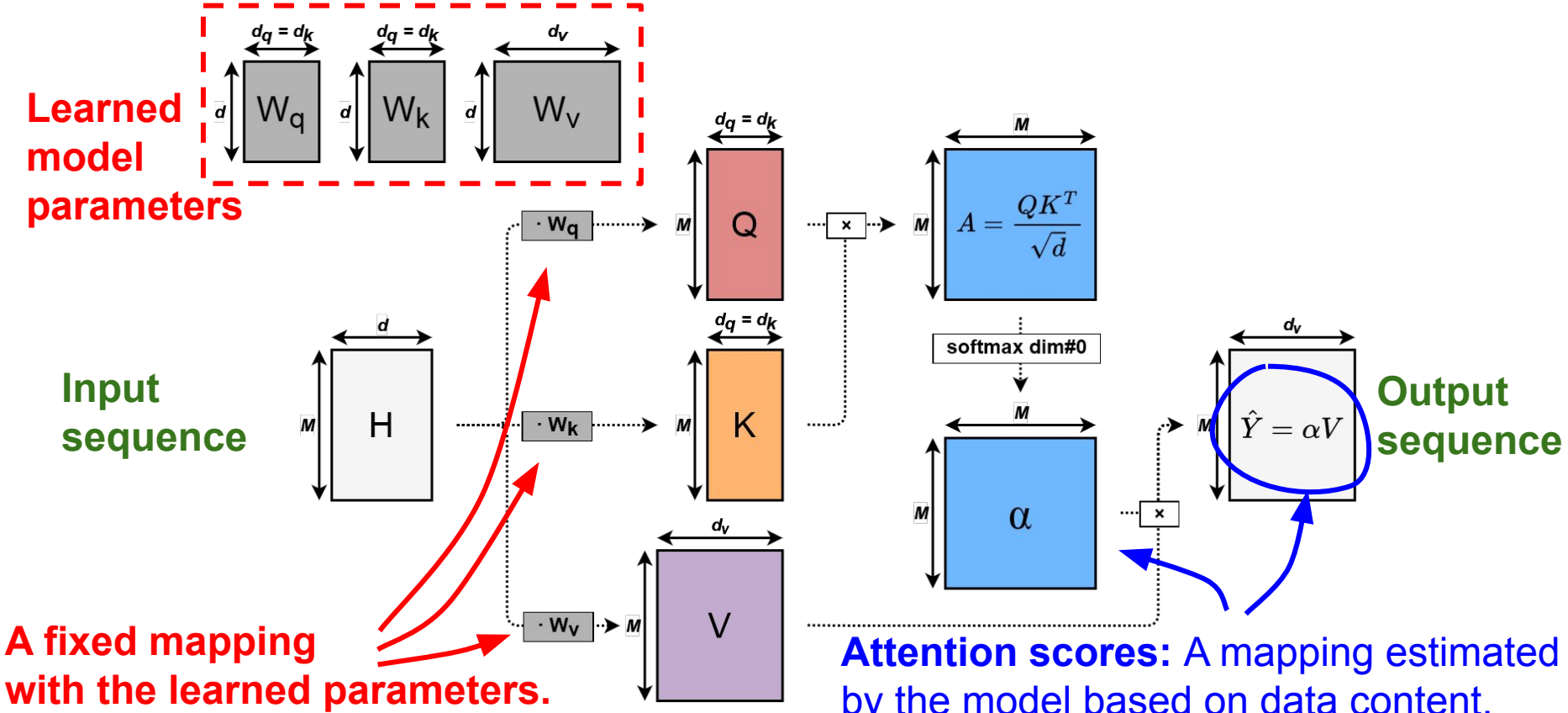


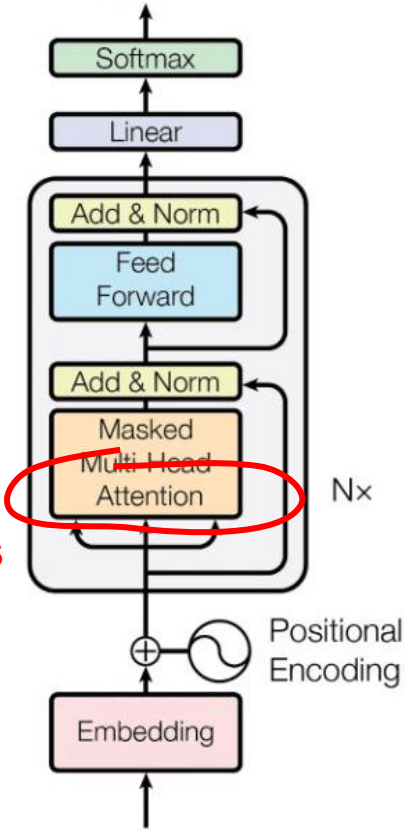
Image source: Vaswani et al., 2017

[_le, _feu, _est _rouge]

Previously - Attention-based models

Transformer Network (2017)

The foundations of (Chat)GPT…



Self-attention layers

[<START>, _le, _feu, _est]

Unsupervised learning

Creating large labeled databases can be expensive.

A huge amount of data is available to us in the world but most of it is unlabeled.

Is unlabeled data useless?

Unsupervised learning

Creating large labeled databases can be expensive.

A huge amount of data is available to us in the world but most of it is unlabeled.

Is unlabeled data useless?

No. → **Unsupervised learning**

Previously - Supervised learning

Given: The training sample, a set of (input, label) pairs

$$\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$$

$$x \in X \subset \mathbb{R}^n,\ y \in Y \subset \mathbb{R}^k$$

Task: The estimation of the label (the expected output) from the input

I.e., we search for a (hypothesis-)function h_θ, for which:

$$h_\theta(x) = \hat{y} \approx y$$

Unsupervised learning

The **training set**:

$$\{x^{(0)}, x^{(1)}, \ \dots \ , x^{(m)}\}$$

$$x \in X \subset \mathbb{R}^n$$

Objective: ???

Unsupervised learning

The **training set**:

$$\{x^{(0)}, x^{(1)}, \ldots, x^{(m)}\}$$

$$x \in X \subset \mathbb{R}^n$$

Notation remains: $x_i^{(j)}$

j: index of data point

i: index of variable

Objective: ???

Unsupervised learning

Without labels, the task is unclear...

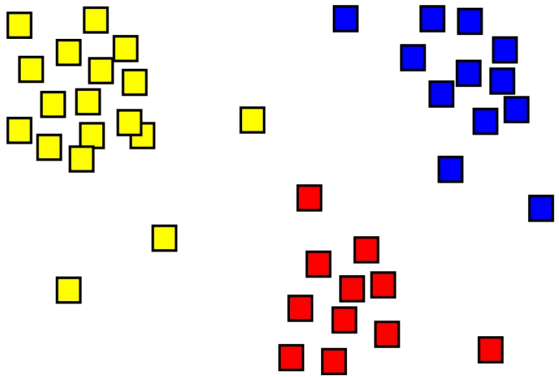
Several specific tasks are possible:

- **Clustering**
- **Compression / dimension reduction**
- **Noise removal / denoising**
- **Sample generation**
- **...**

Clustering

Clustering - Grouping of data points

Task: Assign labels to the $x^{(j)}$ data points so that similar data points (based on some sort of similarity) receive the same label.



Clustering

Clustering - Grouping of data points

Task: Assign labels to the $x^{(j)}$ data points so that similar data points (based on some sort of similarity) receive the same label.

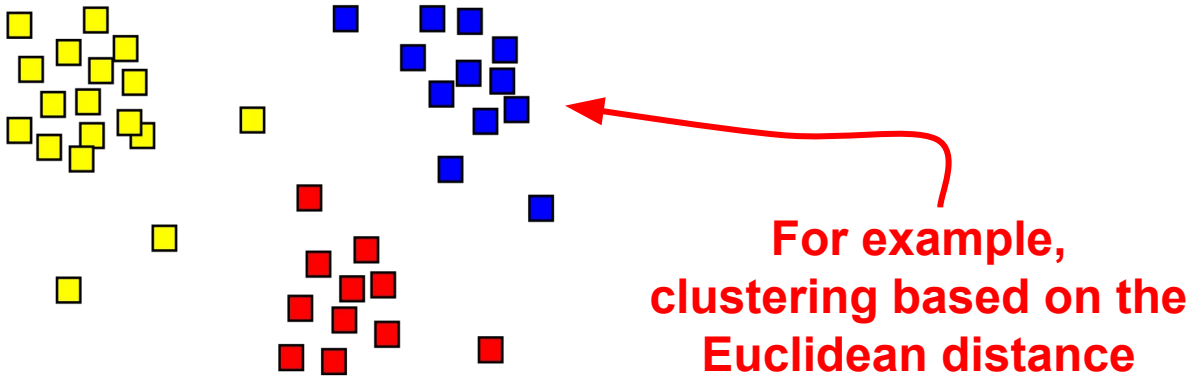


Image source: Saglam et al., 2016

Clustering

Clustering - Grouping of data points

- **Marketing, targeted advertising, recommendation systems:** Grouping similar customers for targeted offers
- **Classic computer vision**, e.g., image segmentation
- **Outlier / anomaly detection**

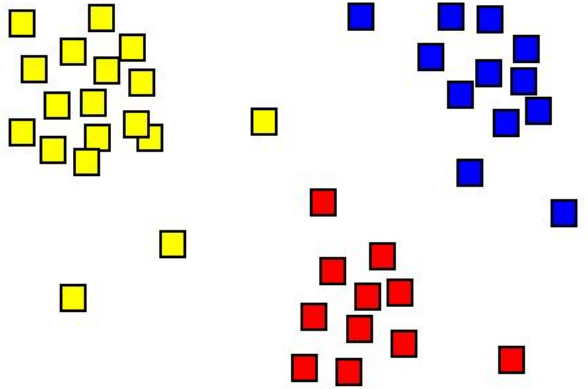


Image source: https://scikit-learn.org/

Clustering

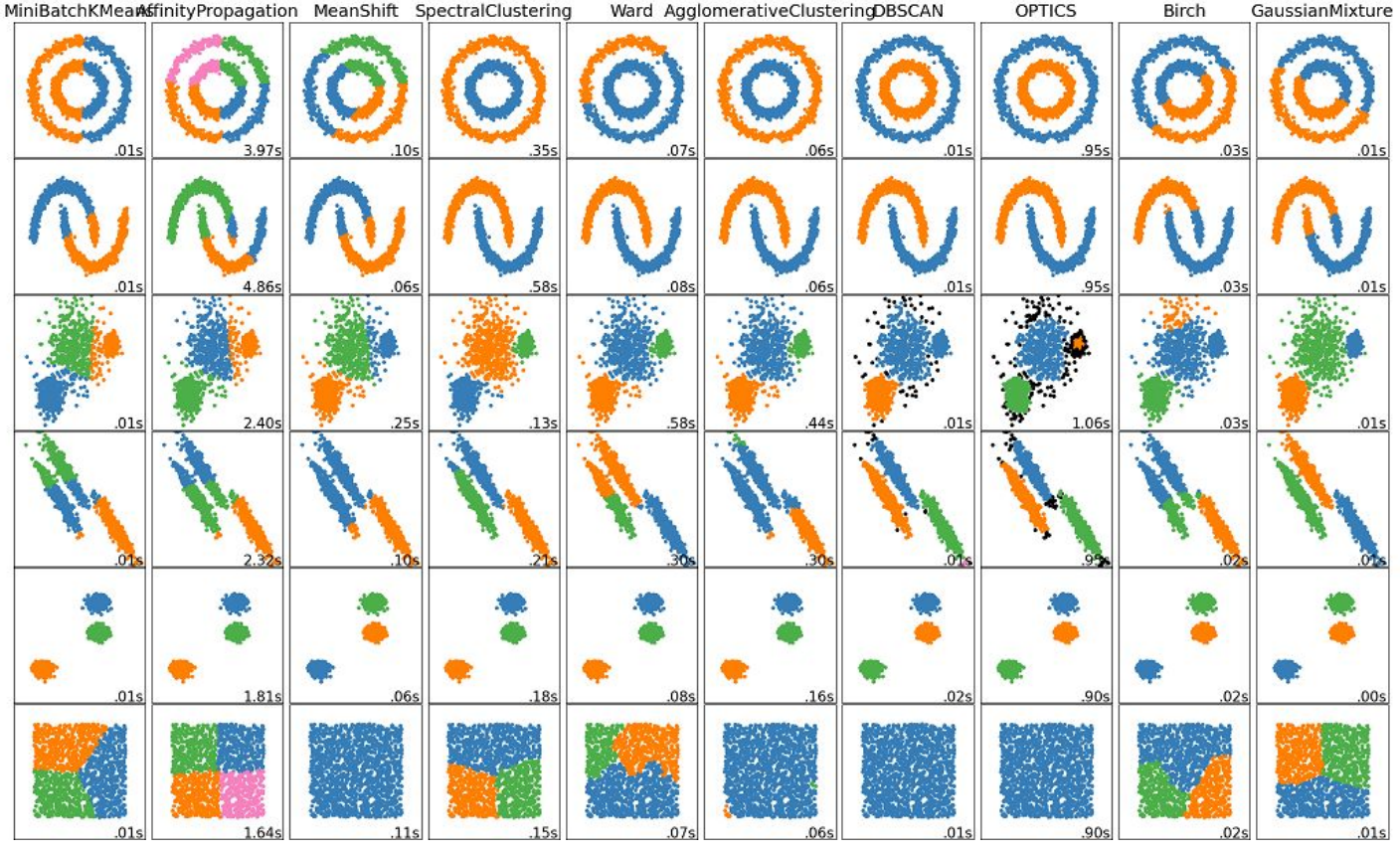
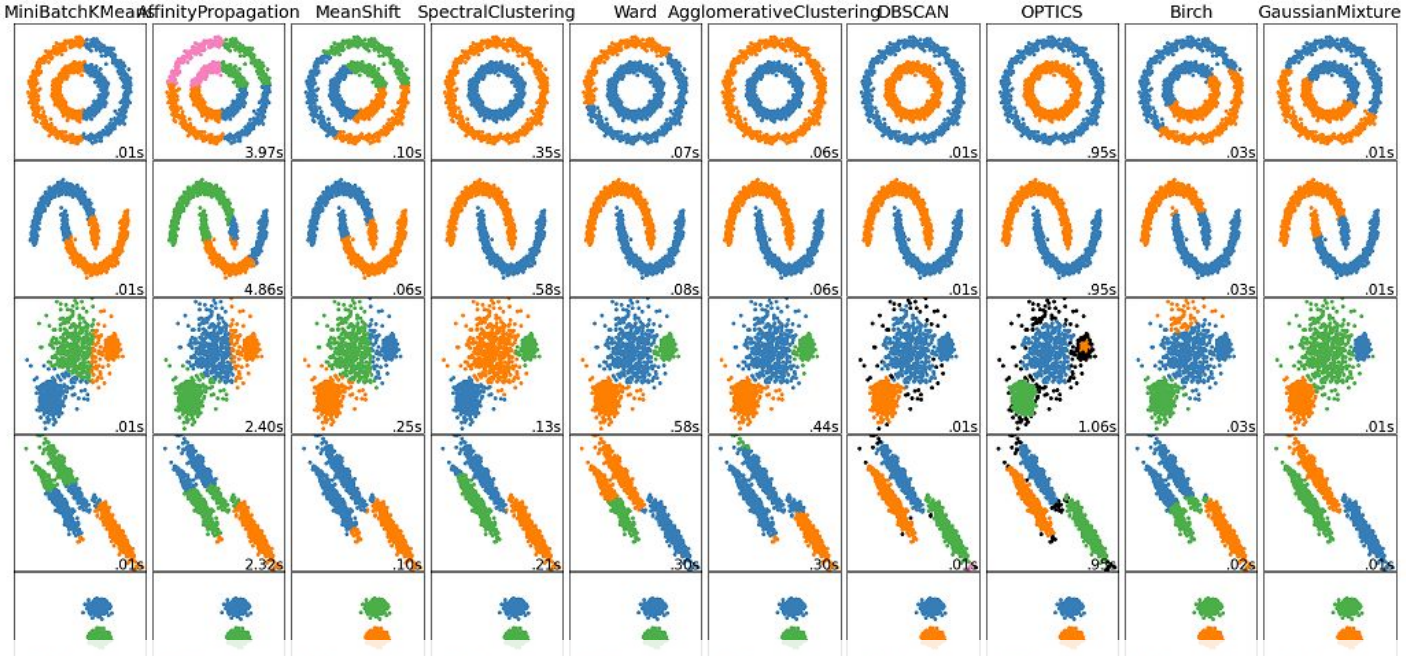


Image source: https://scikit-learn.org/

Clustering



Clustering (without true labels) is not a well-defined task…

The outcome / usefulness of clustering depends largely on the data and the way the clustering algorithm works…

Unsupervised learning

Without labels, the task is unclear...

Several specific tasks are possible:

- **Clustering**
- **Compression / dimension reduction**
- **Noise removal / denoising**
- **Sample generation**
- **...**

Compression / dimension reduction

Task: Represent multivariate data points using fewer variables with the smallest possible reconstruction error!

Applications:

- (Lossy) **data compression**

Compression / dimension reduction

Task: Represent multivariate data points using fewer variables with the smallest possible reconstruction error!

Applications:

- (Lossy) **data compression**
- Learning **more compact representations**
  - When we choose to use a simpler model with less parameters for further data processing.

Compression / dimension reduction

Task: Represent multivariate data points using fewer variables with the smallest possible reconstruction error!

Applications:

- (Lossy) **data compression**
- Learning **more compact representations**
  - When we choose to use a simpler model with less parameters for further data processing.
- **Visualization of high-dimensional data** for manual data examination (compression into 2 or 3 dimensions)

Compression / dimension reduction

Task: Represent multivariate data points using fewer variables with the smallest possible reconstruction error!

Is this a well defined task?

Compression / dimension reduction

Constraint

Task: Represent multivariate data points using fewer variables with the smallest possible reconstruction error!

Objective

Is this a well defined task?
Yes!

Compression / dimension reduction

Constraint

Task: Represent multivariate data points using fewer variables with the smallest possible reconstruction error!

Objective

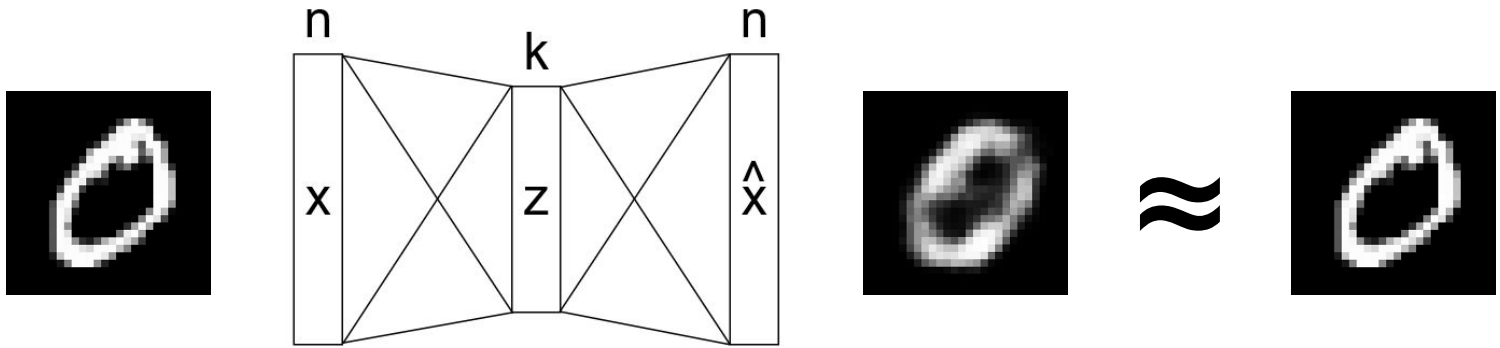
Can we learn it with a neural network?

Autoencoder - Compression

Undercomplete autoencoder

Lossy compression:

- **Objective:** Reconstruct the input
  on the output with the least possible error!



Autoencoder - Compression

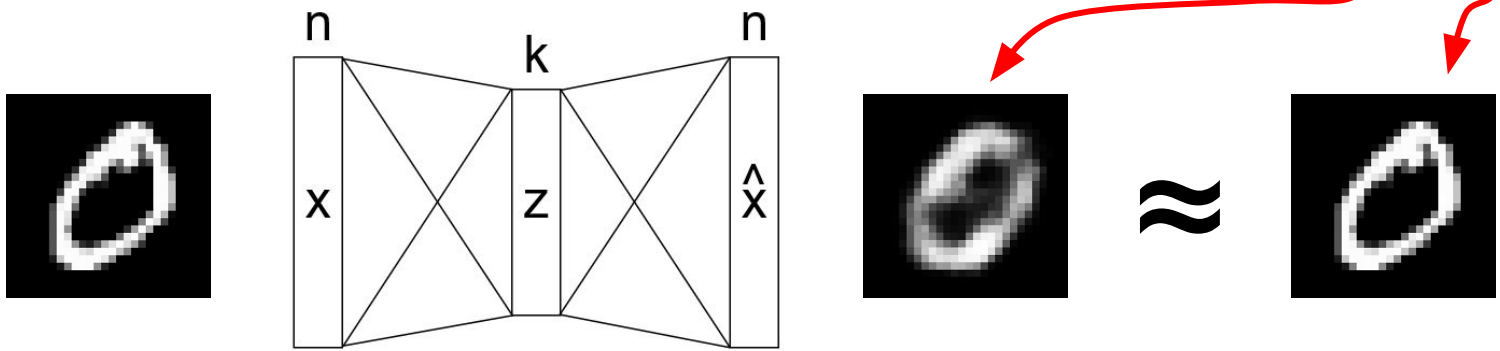
Undercomplete autoencoder

Lossy compression:

- **Objective:** Reconstruct the input on the output with the least possible error!

The loss function is, for example, a simple MSE

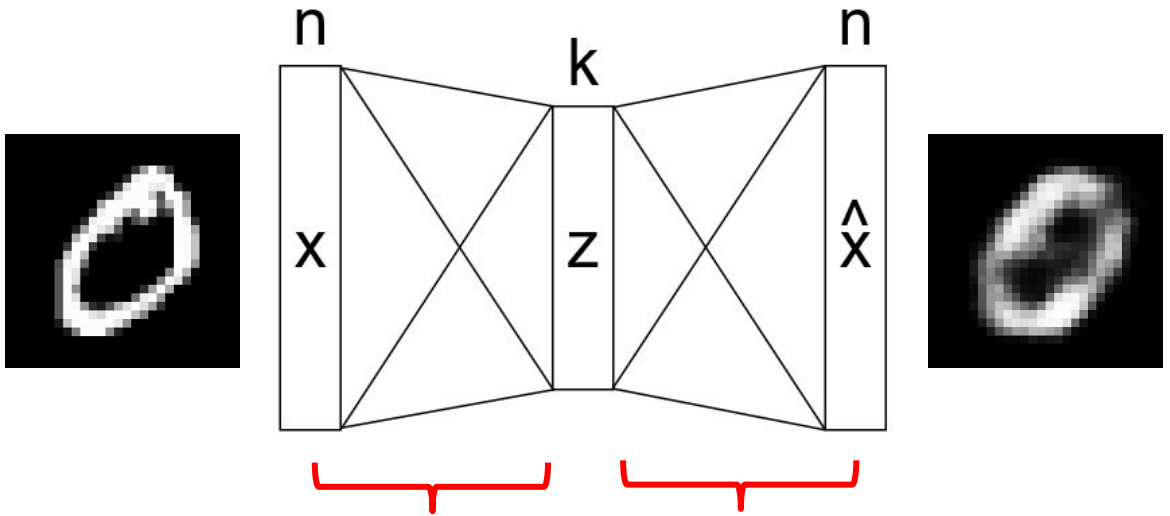
$$J(\Theta) = \frac{1}{n}||\hat{x} - x||_2^2$$



Autoencoder

$$h(x) = \hat{x} = W_{dec}(W_{enc}x + b_{enc}) + b_{dec}$$

Undercomplete autoencoder



$W_{enc} \in \mathbb{R}^{k \times n}$ $W_{dec} \in \mathbb{R}^{n \times k}$

$b_{enc} \in \mathbb{R}^{k}$ $b_{dec} \in \mathbb{R}^{n}$

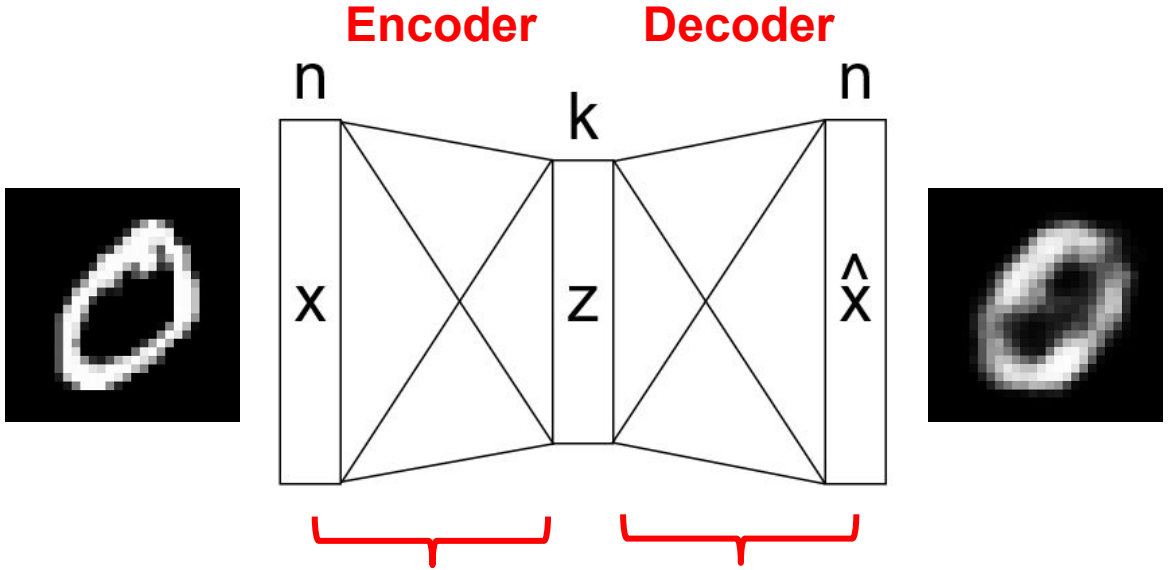
Autoencoder

$$h(x) = \hat{x} = W_{dec}(W_{enc}x + b_{enc}) + b_{dec}$$

h(x) corresponds to a simple linear autoencoder.

One fully-connected layer for the encoder and the decoder each.

Undercomplete autoencoder



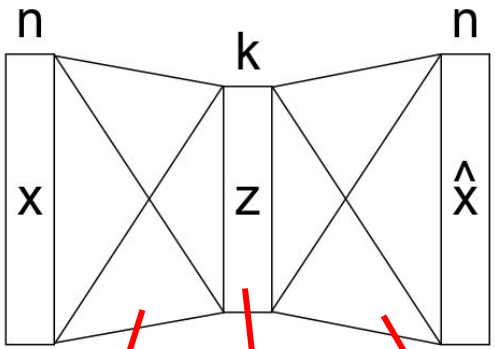
Autoencoder - Compression

Lossy compression

Example: Learning to compress 64 by 64 pixel grayscale portraits.

Autoencoder - Compression

Lossy compression - Example



Analogy:

x: uncompressed data

Encoder: zip.exe

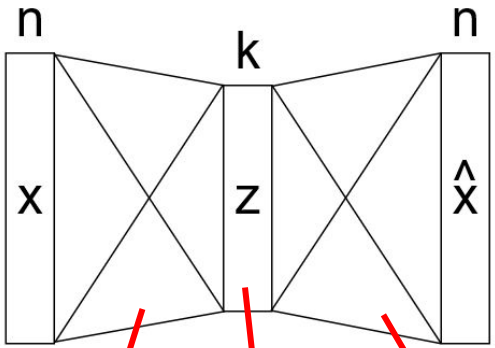
Decoder: unzip.exe

$\hat{x}$: reconstructed data

Autoencoder - Compression

Lossy compression - Example

Keep in mind: The ZIP format supports lossless compression algorithms. ZIP is mentioned here only as an analogy. The autoencoder is more comparable to (lossy compression) formats such as JPEG, MP3, etc.



Analogy:

x: uncompressed data

Encoder: zip.exe

Decoder: unzip.exe

$\hat{x}$: reconstructed data

Image source: Sandipan Dey, sandipanweb.wordpress.com/

Autoencoder - Compression

$\hat{x}$: output (the input reconstructed from the compressed representation)

Lossy compression - Example

x: original input

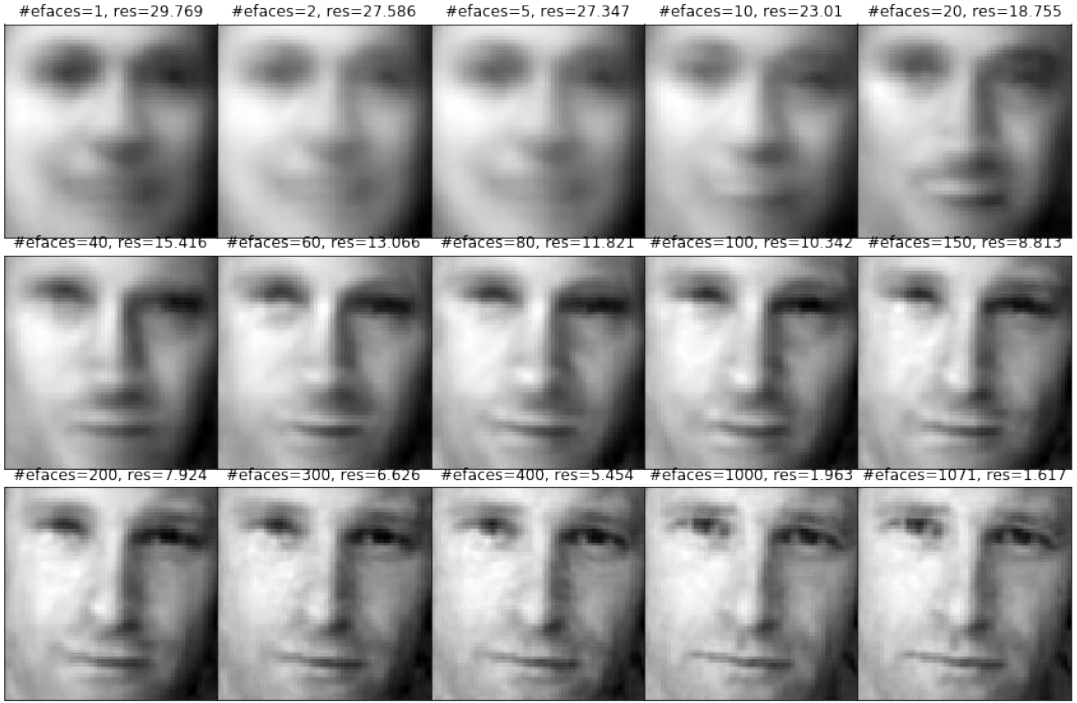


Image source: Sandipan Dey, sandipanweb.wordpress.com/

Autoencoder - Compression

$\hat{x}$: output (the input reconstructed from the compressed representation)

Lossy compression - Example

|z| = 1

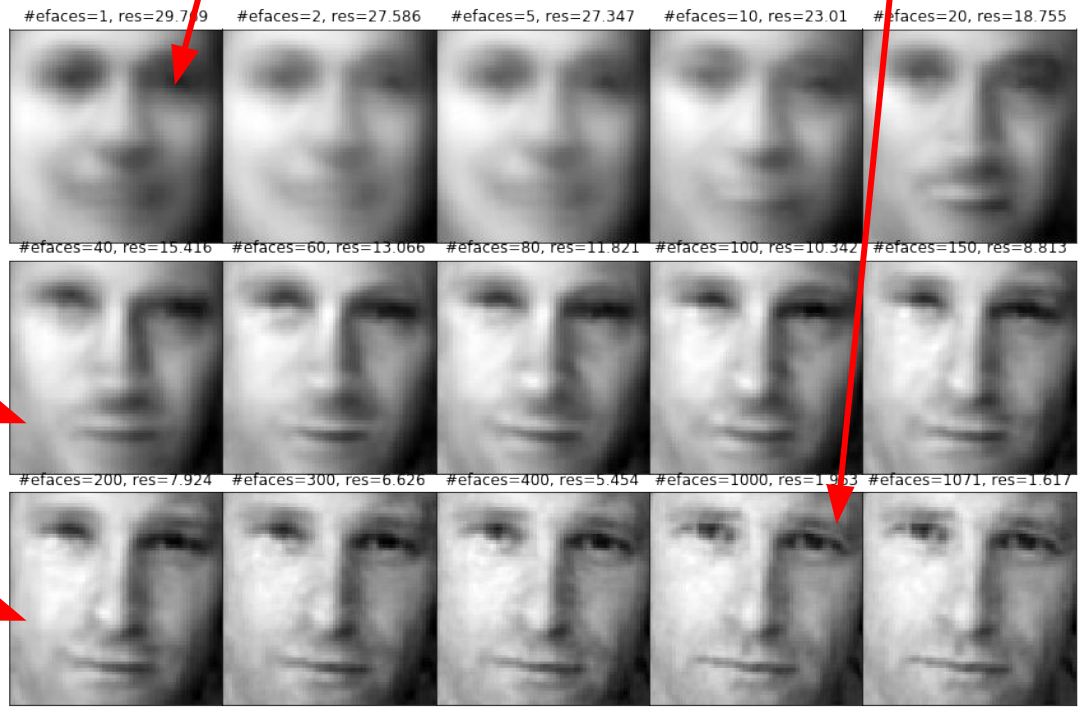
|z| = 1000

|x| = 4096

x: original input

|z| = 40

|z| = 200



Autoencoder - Compression

Lossy compression - Example

The JPEG and MP3 lossy compression algorithms have been optimized for photos and music by manual deisgn.

Autoencoder - Compression

Lossy compression - Example

The JPEG and MP3 lossy compression algorithms have been optimized for photos and music by manual deisgn.

The JPEG compression standard saves space by heavily compressing high-frequency components. Since photographs typically contain few sharp color transitions, good image quality can be expected. However, for images containing text, JPEG compression is not ideal for this very reason.

Image source: Sandipan Dey, sandipanweb.wordpress.com/

$\hat{x}$: output (the input reconstructed from the compressed representation)

The network was trained on human faces

Autoencoder - Compression

Lossy compression - Example

Similarly, machine learning-based compression will only work well on data similar to what it was trained on.

x: original input

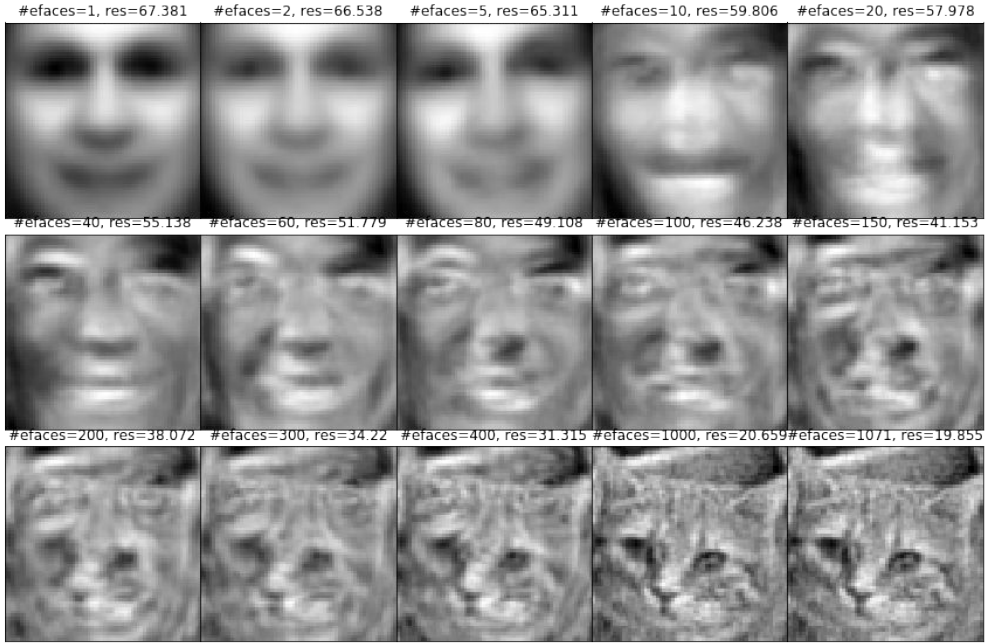


Image source: Sandipan Dey, sandipanweb.wordpress.com/

Autoencoder - Compression

$\hat{x}$: output (the input reconstructed from the compressed representation)

The network was trained on human faces

Lossy compression - Example

Similarly, machine learning-based compression will only work well on data similar to what it was trained on.

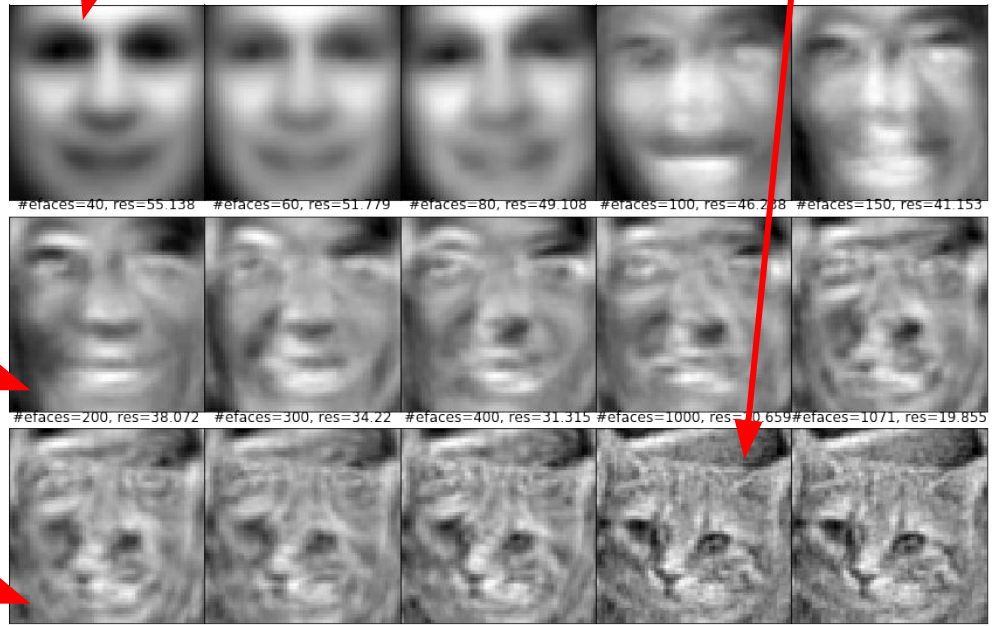
x: original input

|z| = 1

|z| = 1000

|z| = 40

|z| = 200



Unsupervised learning

Without labels, the task is unclear...

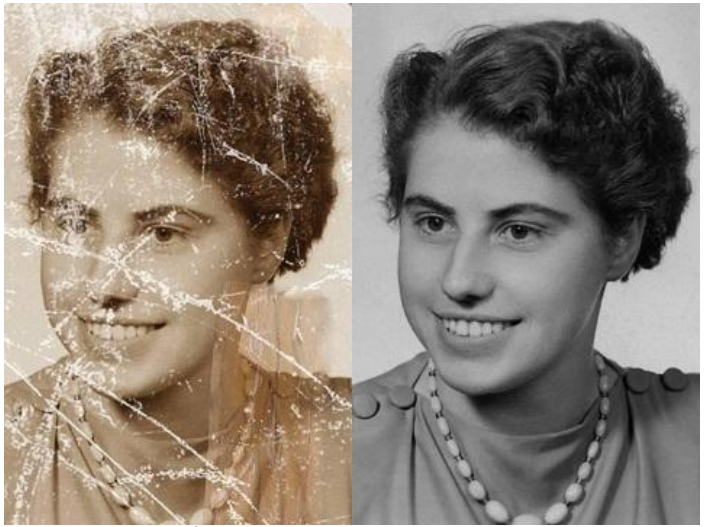
Several specific tasks are possible:

- **Clustering**
- **Compression / dimension reduction**
- **Noise removal / denoising**
- **Sample generation**
- **...**

Image source: https://github.com/WenchenLi/PhotoRestoration, https://medium.com/@sorenlind

Denoising

Improving the quality of photos - Denoising



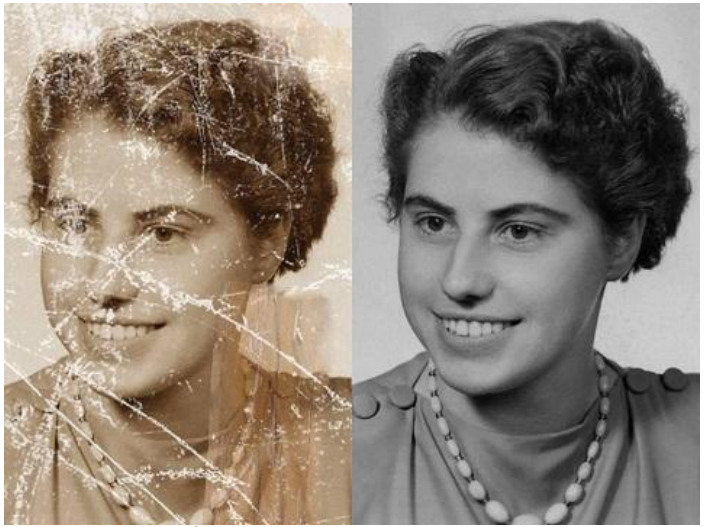
x' $\hat{x}$

x' $\hat{x}$

Image source: https://github.com/WenchenLi/PhotoRestoration, https://medium.com/@sorenlind

Denoising

Improving the quality of photos - Denoising



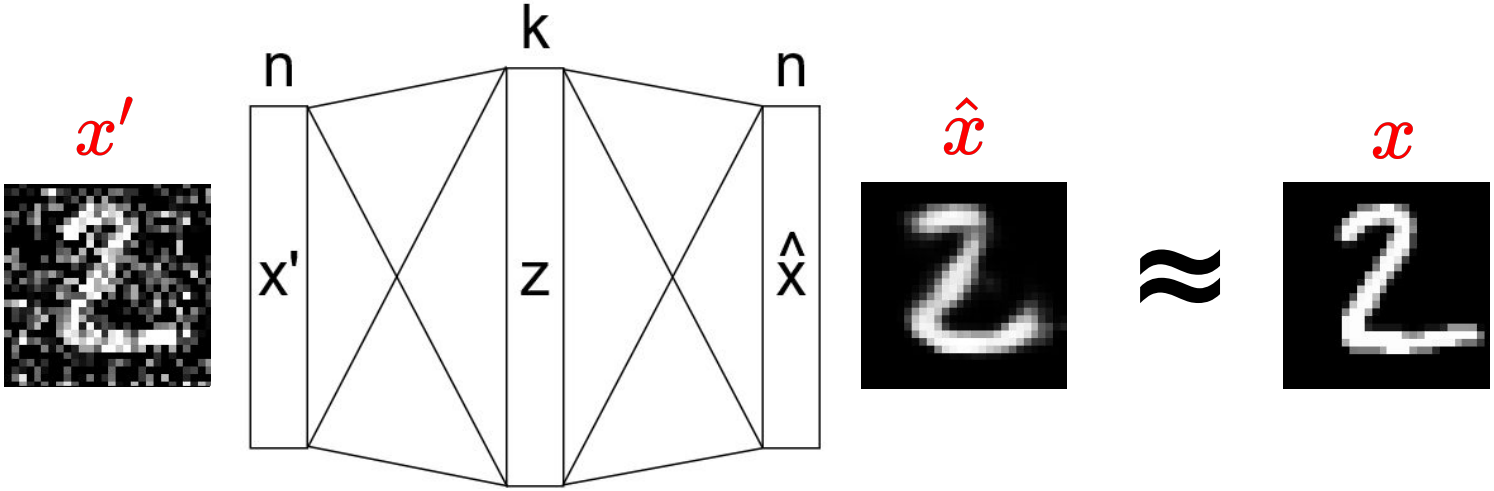
x' $\hat{x}$

x' $\hat{x}$

Can we learn it with a neural network?

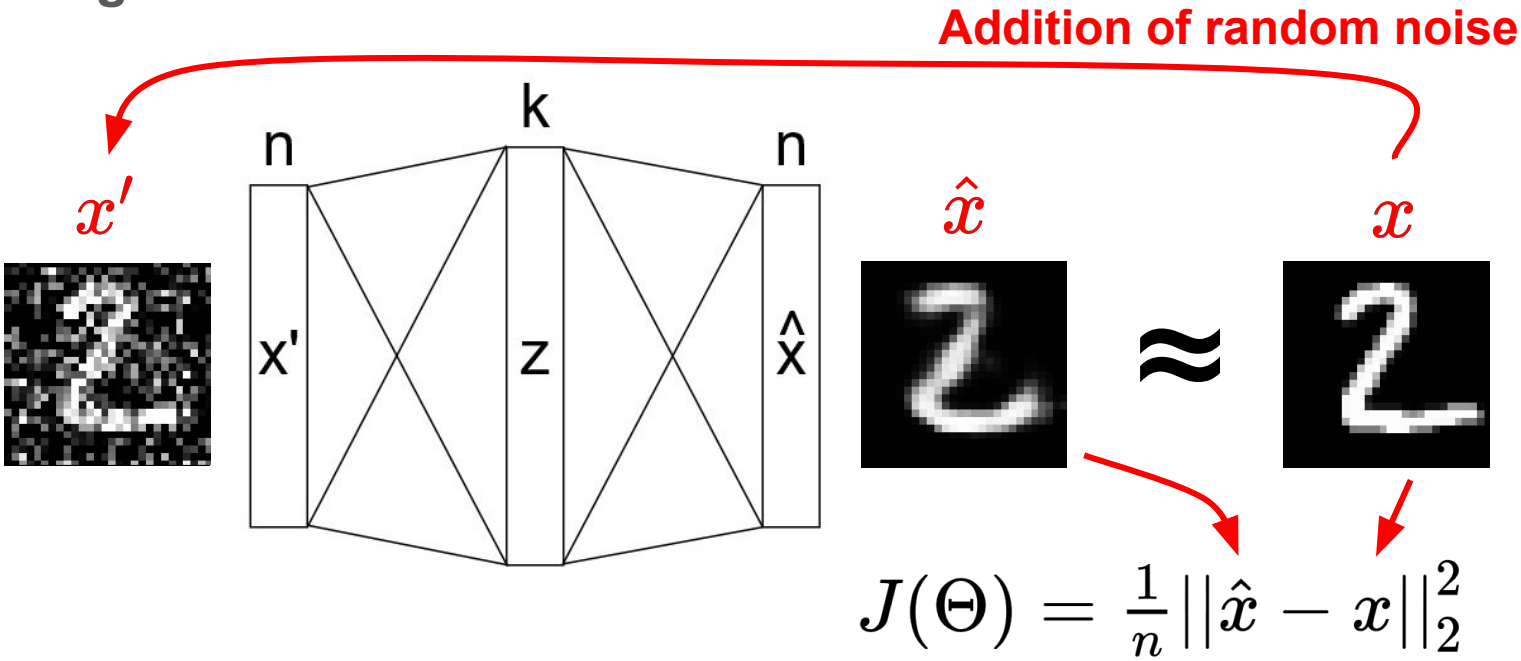
Autoencoder

Denoising autoencoder



Autoencoder

Denoising autoencoder

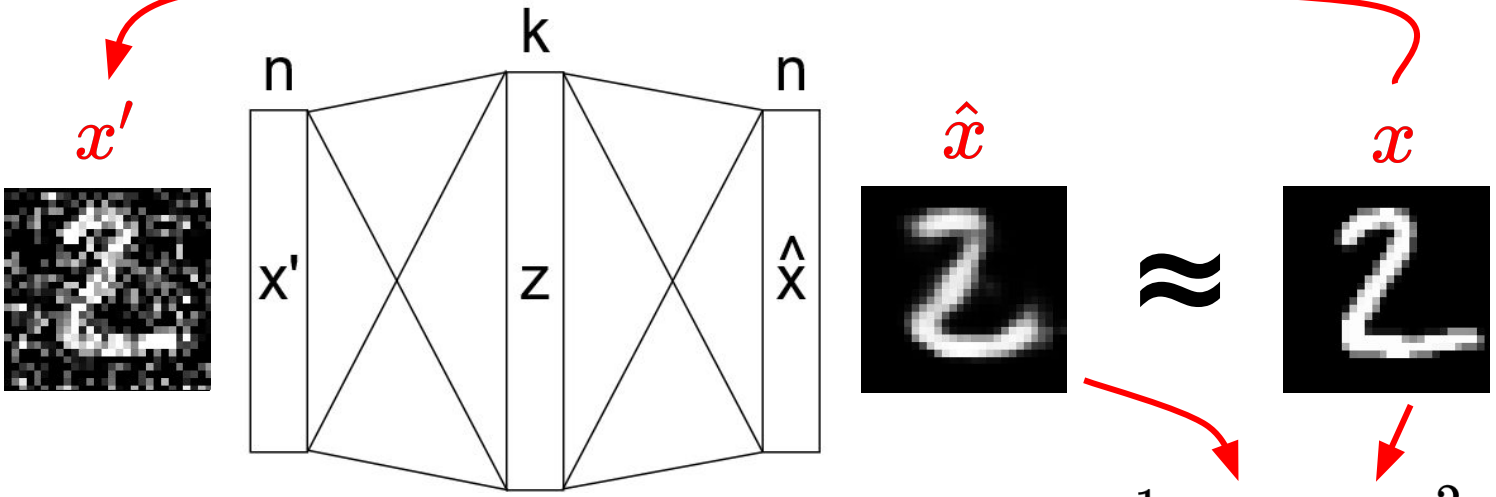


Autoencoder

Training requires the knowledge of the original, noise-free sample…

Denoising autoencoder

Addition of random noise



$$J(\Theta) = \frac{1}{n}||\hat{x} - x||_2^2$$

The loss fn. is simple MSE

Denoising autoencoder - applications

Inpainting - Filling in missing parts

x' $\hat{x}$ x

Image source: Larsson et al., 2017

Denoising autoencoder - applications

Colorization

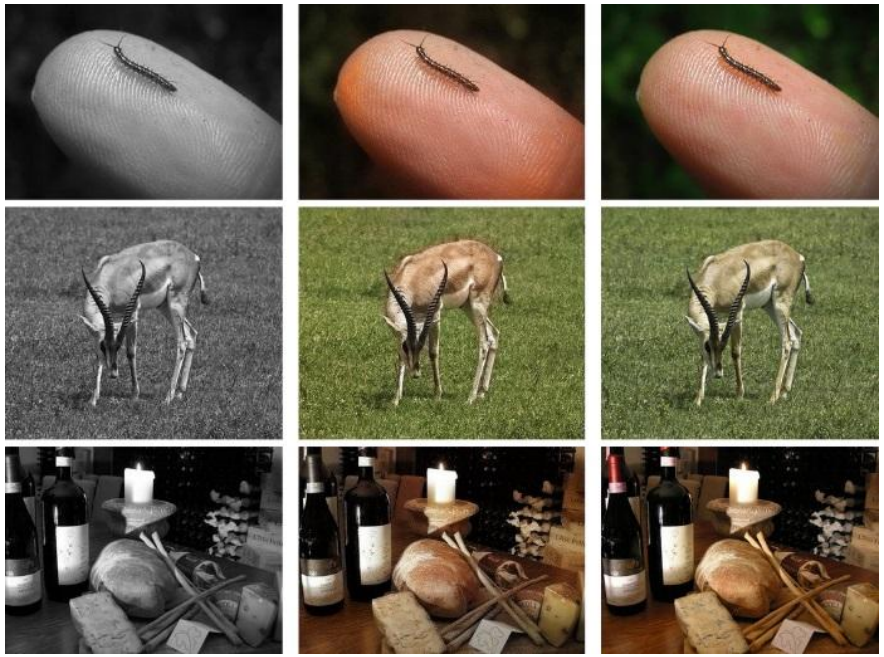
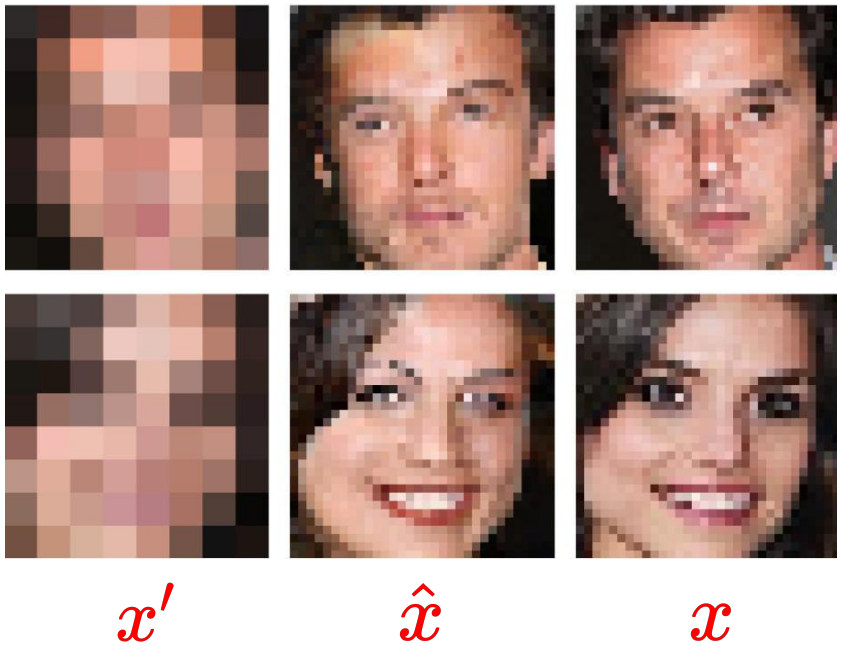


Image source: Dahl et al., 2017

Denoising autoencoder - applications

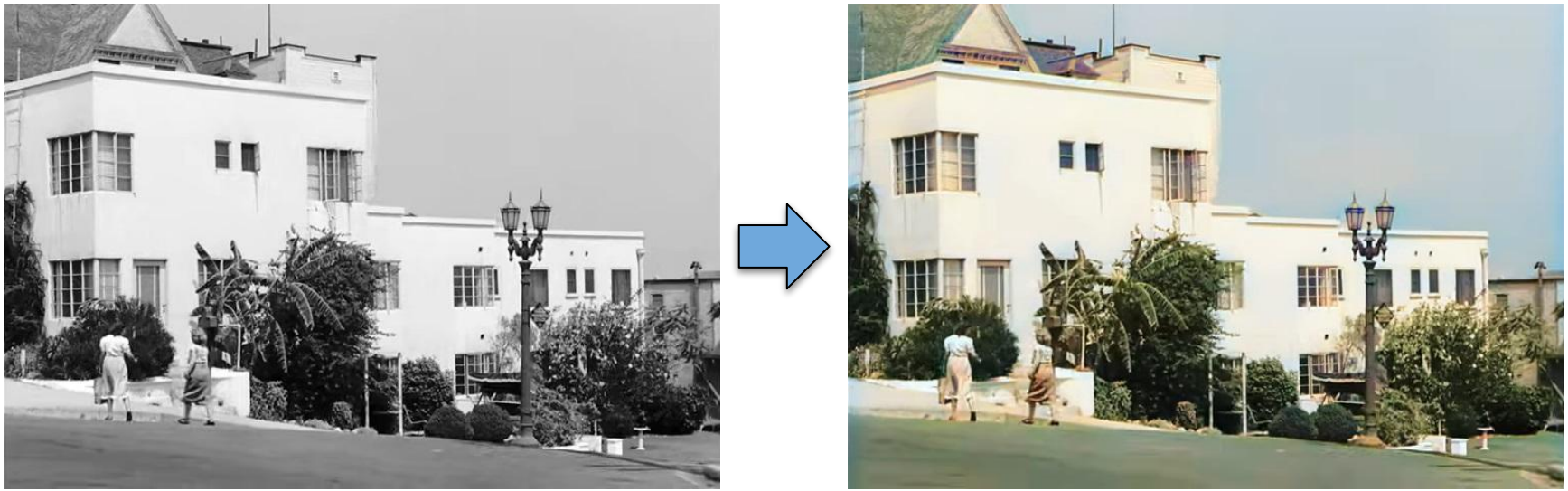
Super-resolution - Smart upscaling



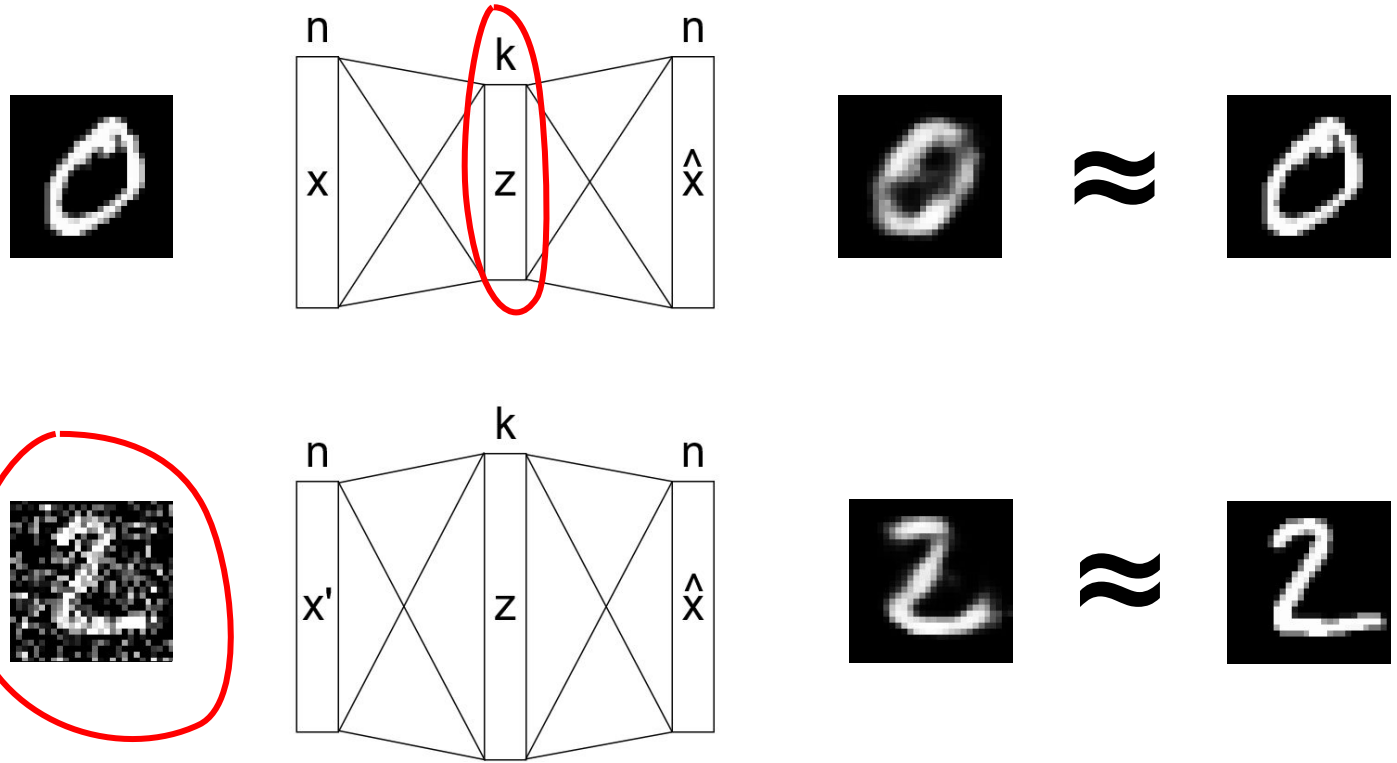
Denoising autoencoder - applications

“DeOldify”

https://www.youtube.com/watch?v=u9Z3lCmIOlg



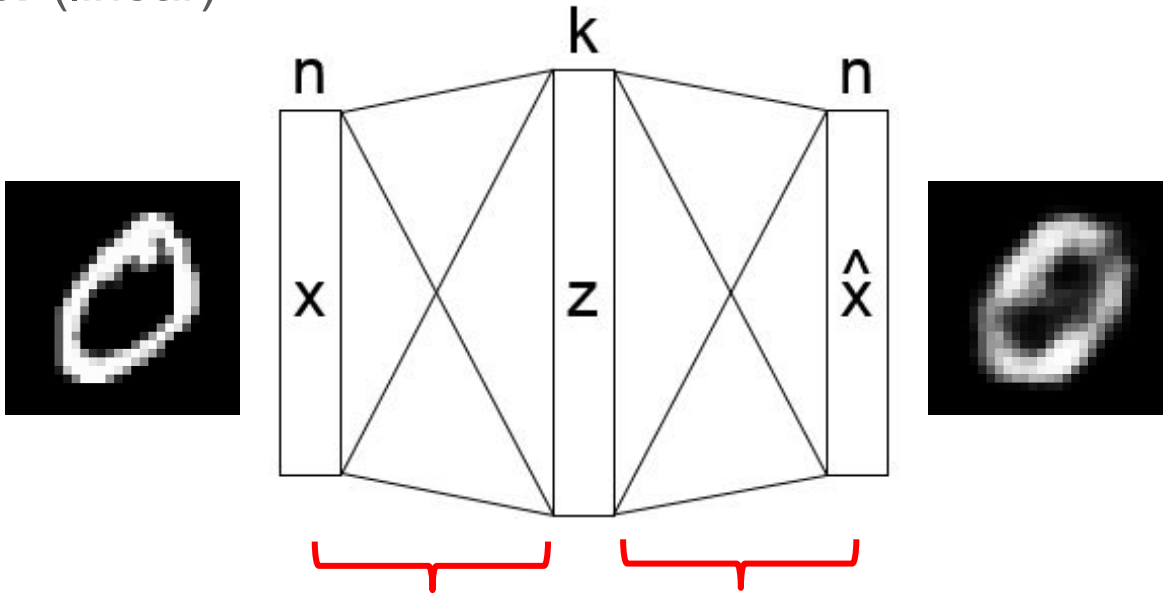
Autoencoder - The importance of constraints



Autoencoder

$$h(x) = \hat{x} = W_{dec}(W_{enc}x + b_{enc}) + b_{dec}$$

Autoencoder (linear)

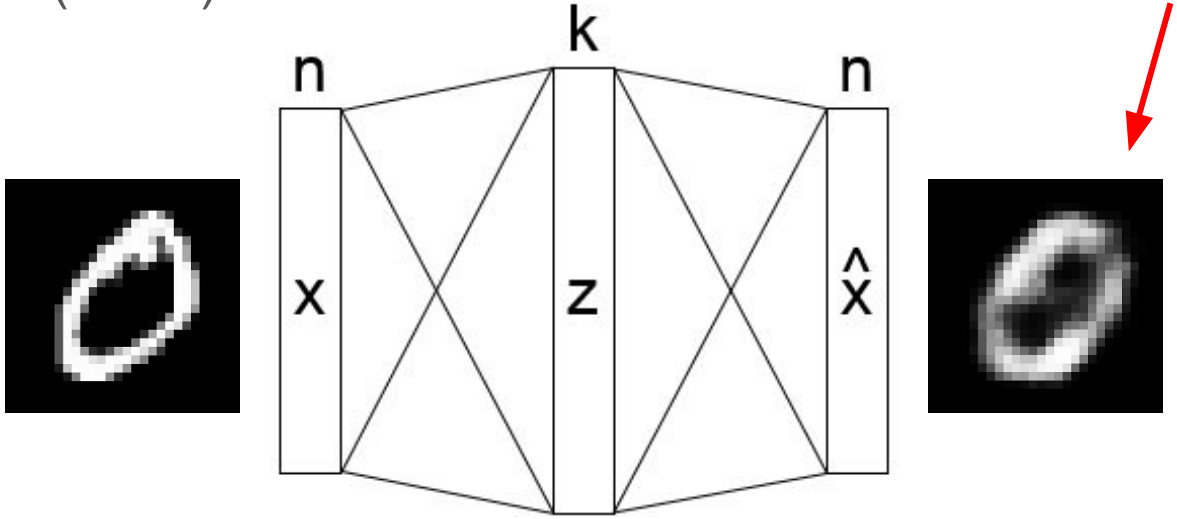


Autoencoder

$$h(x) = \hat{x} = W_{dec}(W_{enc}x + b_{enc}) + b_{dec}$$

Autoencoder (linear)

Let's learn to reconstruct the input on the output!



In this case, this is a two-layer MLP with no intermediate activation functions

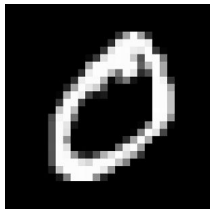
$W_{enc} \in \mathbb{R}^{k \times n}$ $W_{dec} \in \mathbb{R}^{n \times k}$

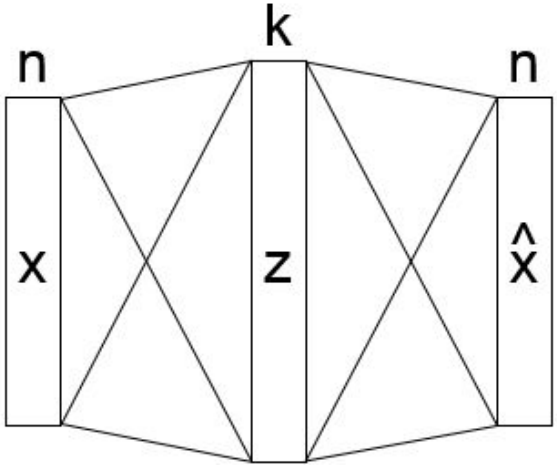
$b_{enc} \in \mathbb{R}^{k}$ $b_{dec} \in \mathbb{R}^{n}$

Autoencoder

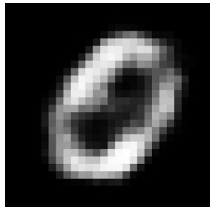
$$h(x) = \hat{x} = W_{dec}(W_{enc}x + b_{enc}) + b_{dec}$$

Autoencoder (linear)

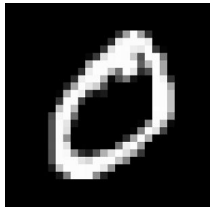




Let's learn to reconstruct the input on the output!



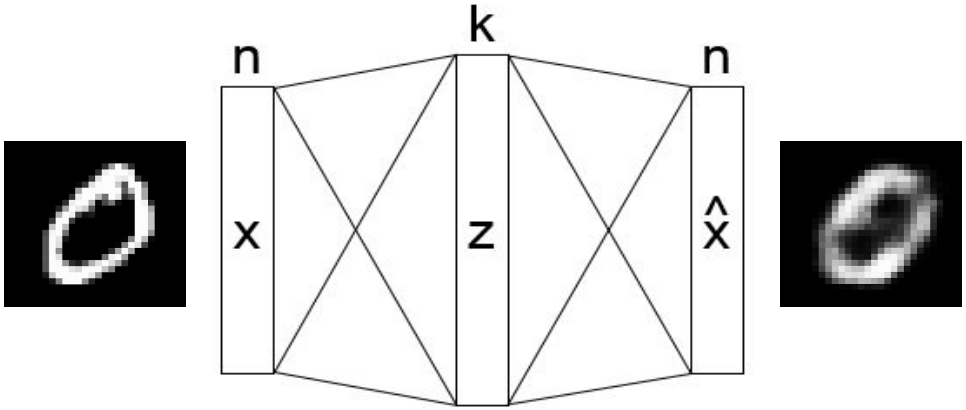
≈



$$J(\Theta) = \frac{1}{n}||\hat{x} - x||_2^2$$

The loss fn. is simple MSE

Autoencoder



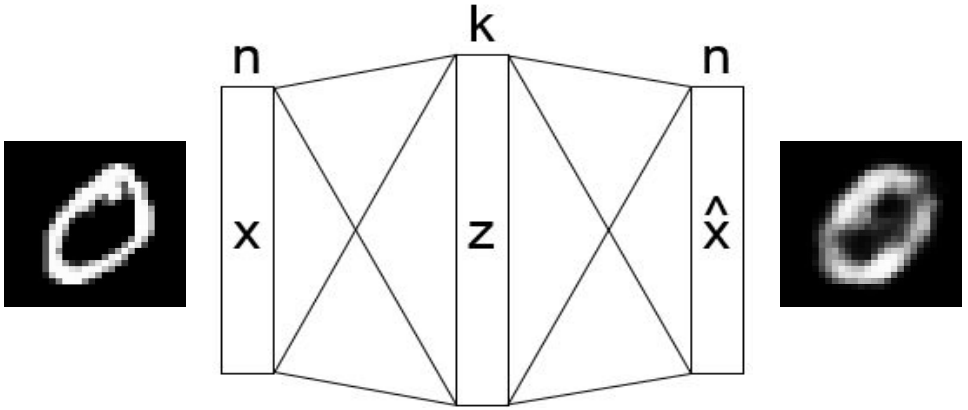
The hypothesis function

$$h(x) = \hat{x} = W_{dec}(W_{enc}x + b_{enc}) + b_{dec}$$

can be rearranged to:

$$h(x) = \hat{x} = \underbrace{(W_{dec}W_{enc})}_{\in \mathbb{R}^{n \times n}}x + \underbrace{(W_{dec}b_{enc} + b_{dec})}_{\in \mathbb{R}^{n}}$$

Autoencoder



The hypothesis function

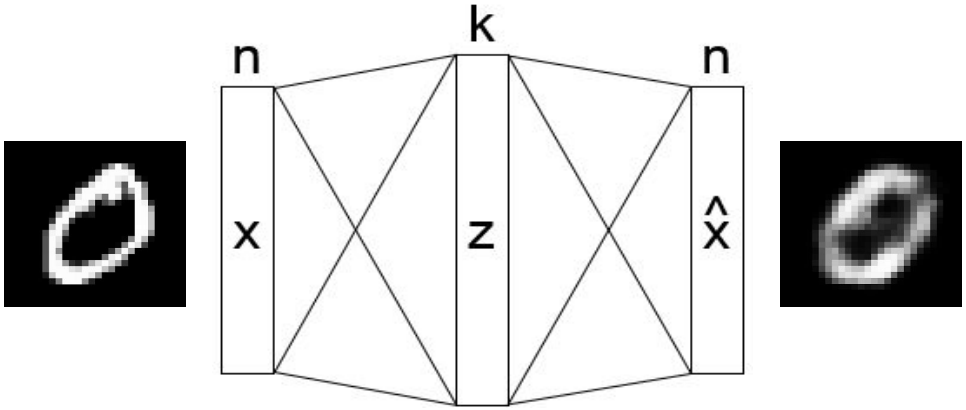
$$h(x) = \hat{x} = W_{dec}(W_{enc}x + b_{enc}) + b_{dec}$$

can be rearranged to:

$$h(x) = \hat{x} = \underbrace{(W_{dec}W_{enc})}_{\in \mathbb{R}^{n \times n}}x + \underbrace{(W_{dec}b_{enc} + b_{dec})}_{\in \mathbb{R}^{n}}$$

The composition of two linear layers can be described using a single layer!

Autoencoder



The hypothesis function

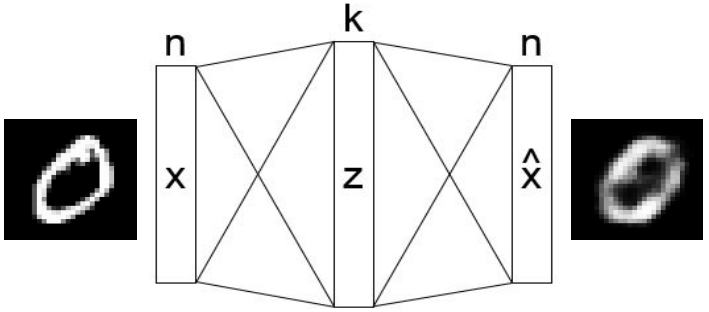
$$h(x) = \hat{x} = W_{dec}(W_{enc}x + b_{enc}) + b_{dec}$$

can be rearranged to:

$$h(x) = \hat{x} = \underbrace{(W_{dec}W_{enc})}_{\in \mathbb{R}^{n \times n}}x + \underbrace{(W_{dec}b_{enc} + b_{dec})}_{\in \mathbb{R}^{n}}$$

In fact, there's nothing stopping the network from learning the identity matrix!

Autoencoder

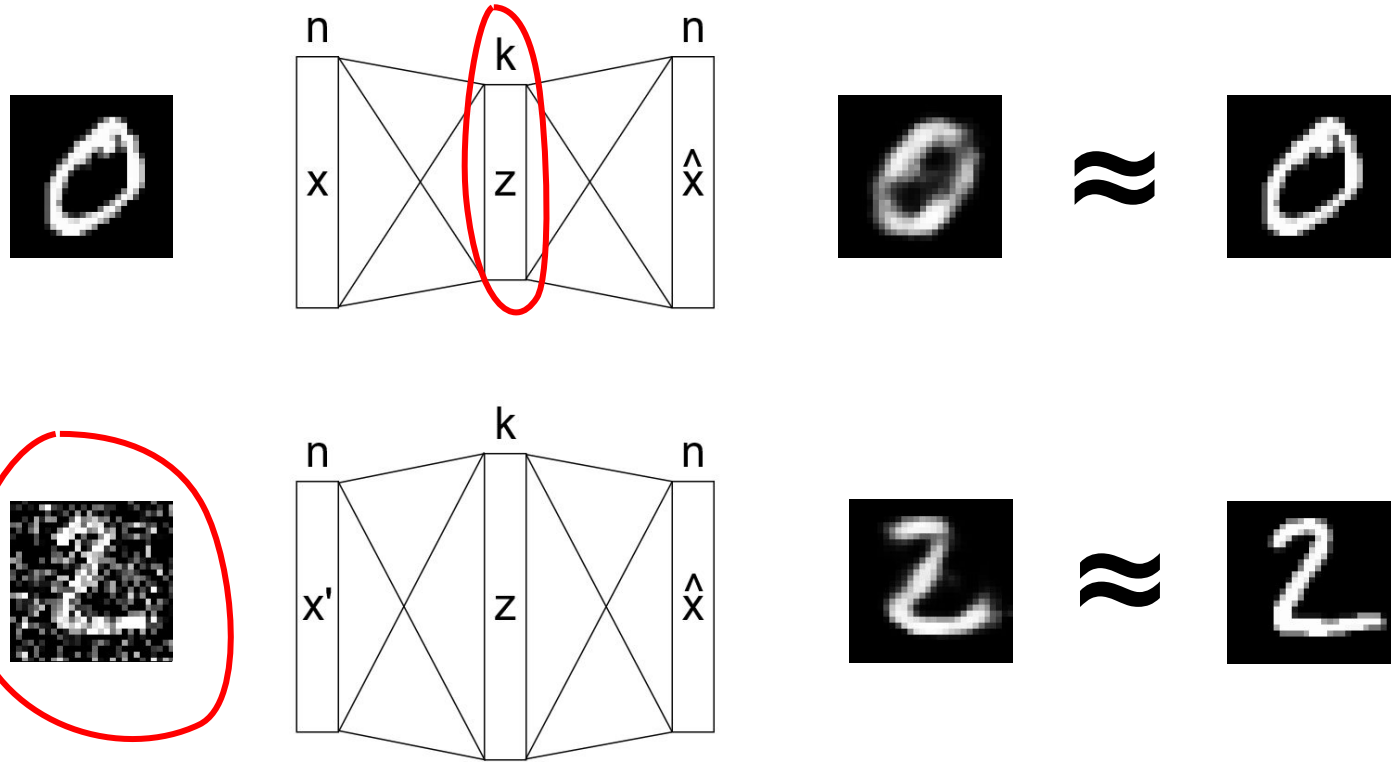


Problem:

The goal of an autoencoder is to reconstruct the input from the output. In this form, **without constraints,** nothing prevents the network from learning trivial weights (an identity matrix).

→ **It achieves its goal perfectly** (loss = 0), **but it is of no use to us**

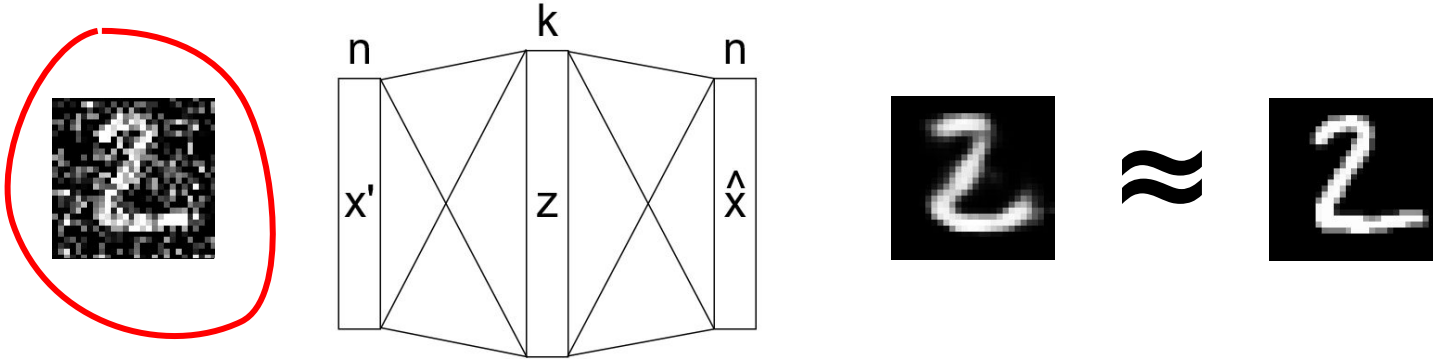
Autoencoder - The importance of constraints



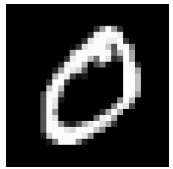
Autoencoder - The importance of constraints

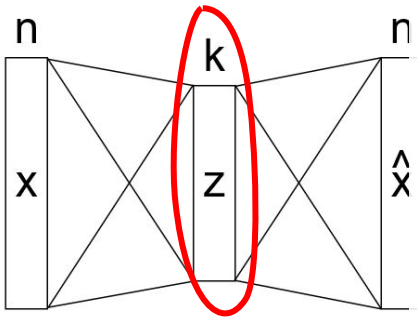
Autoencoder (loosely defined): A neural network of any architecture or type that attempts to learn an identity mapping or a slightly distorted version of it.

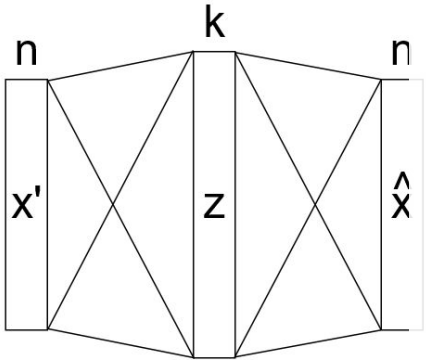
By adding constraints to the learning task or the architecture, we can ensure that the task is not easily solvable, and that **the neural network is only able to solve it by exploring the structure of the data and learning useful knowledge.** The knowledge gained in this way can later be used to solve other tasks involving the data.



Autoencoder - The importance of constraints







When training autoencoders and other neural networks using unsupervised learning, **it is important to prevent them from learning the trivial solution to the identity mapping problem.**

Examples for constraints:

- Undercomplete AE: **Smaller internal representation. $W_{enc} \cdot W_{dec}$ cannot be a full-rank matrix, and therefore cannot be an identity matrix either.**
- Denoising **AE: The goal is not to learn the identity mapping, but rather denoising.**
- **…**

Unsupervised learning

Without labels, the task is unclear...

Several specific tasks are possible:

- **Clustering**
- **Compression / dimension reduction**
- **Noise removal / denoising**
- **Sample generation**
- **...**

Sample generation

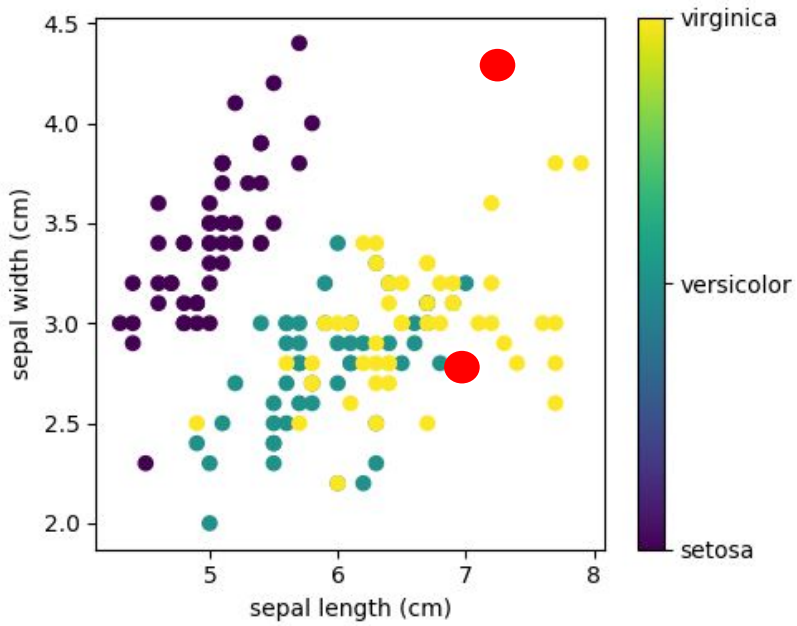
Sample generation (learning the distribution of the data)

Generating new data points similar to those found in the training set.

Sample generation

Sample generation (learning the distribution of the data)

Generating new data points
similar to those found
in the training set.



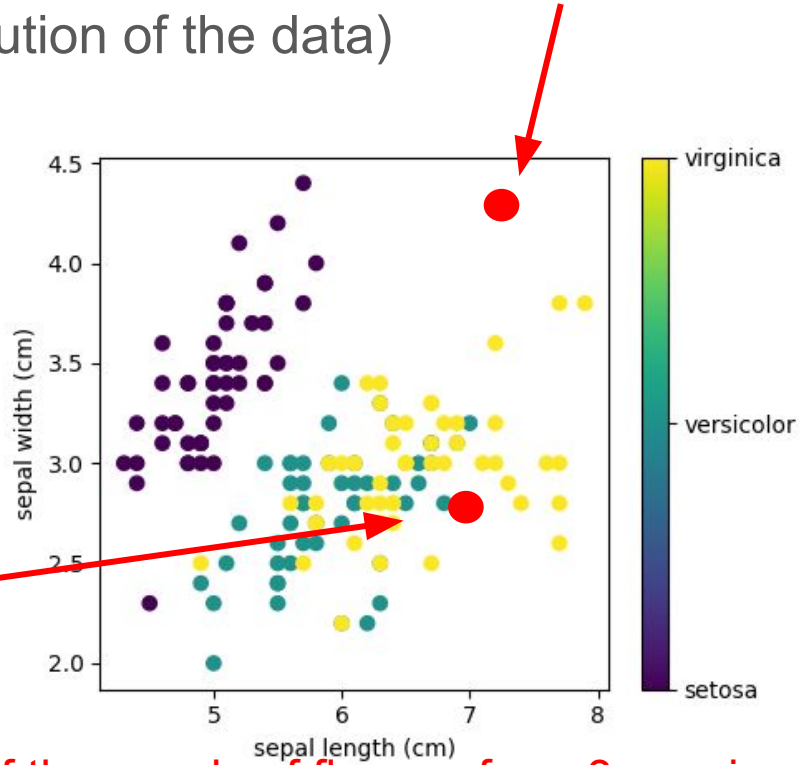
Sample generation

This new data point is unlikely:
The values of both variables
are too large…

Sample generation (learning the distribution of the data)

Generating new data points
similar to those found
in the training set.

This new data point is likely:
The values of the variables are similar
to the real data points.



IRIS dataset: The length & width (2 variables) of the sepals of flowers from 3 species

Sample generation

Sample generation (learning the distribution of the data)
https://www.youtube.com/watch?v=36lE9tV9vm0

Image source: https://blog.segmind.com/sdxl-samplers-2/

State-of-the-art methods in unsupervised learning

Stable Diffusion - Sample generation solved by iterative denoising

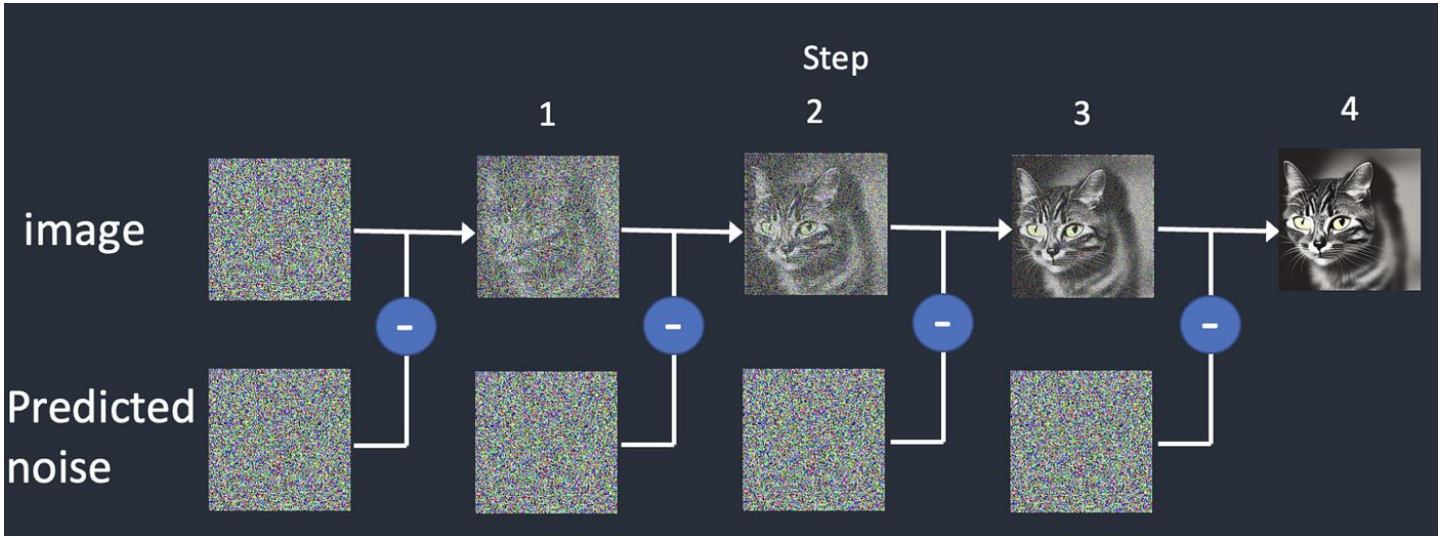
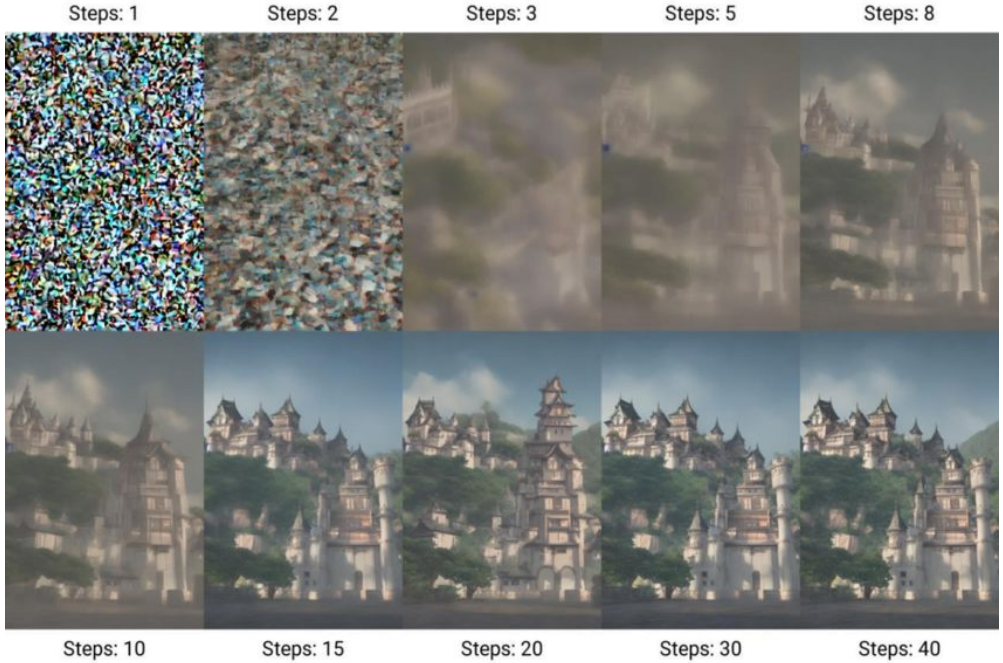


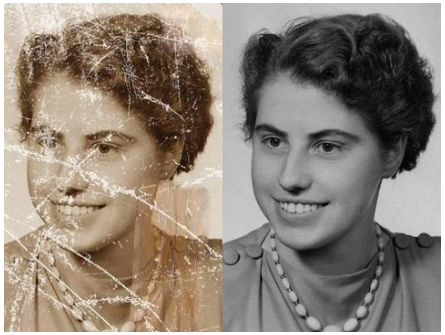
Image source: Fahmy et al., 2024

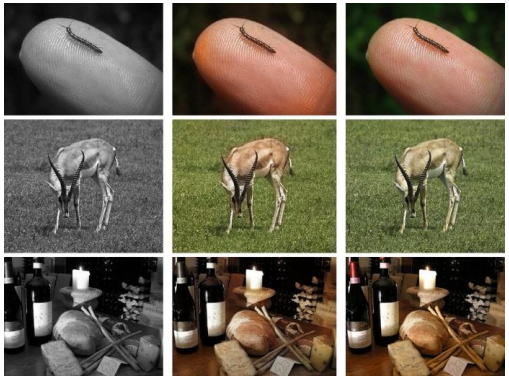
State-of-the-art methods in unsupervised learning

Stable Diffusion - Sample generation solved by iterative denoising



Unsupervised learning





Very nice applications…

However, unsupervised learning has taken on a much more important role in recent years!

Image source: businessolution.org

Unsupervised learning

OpenAI GPT models:

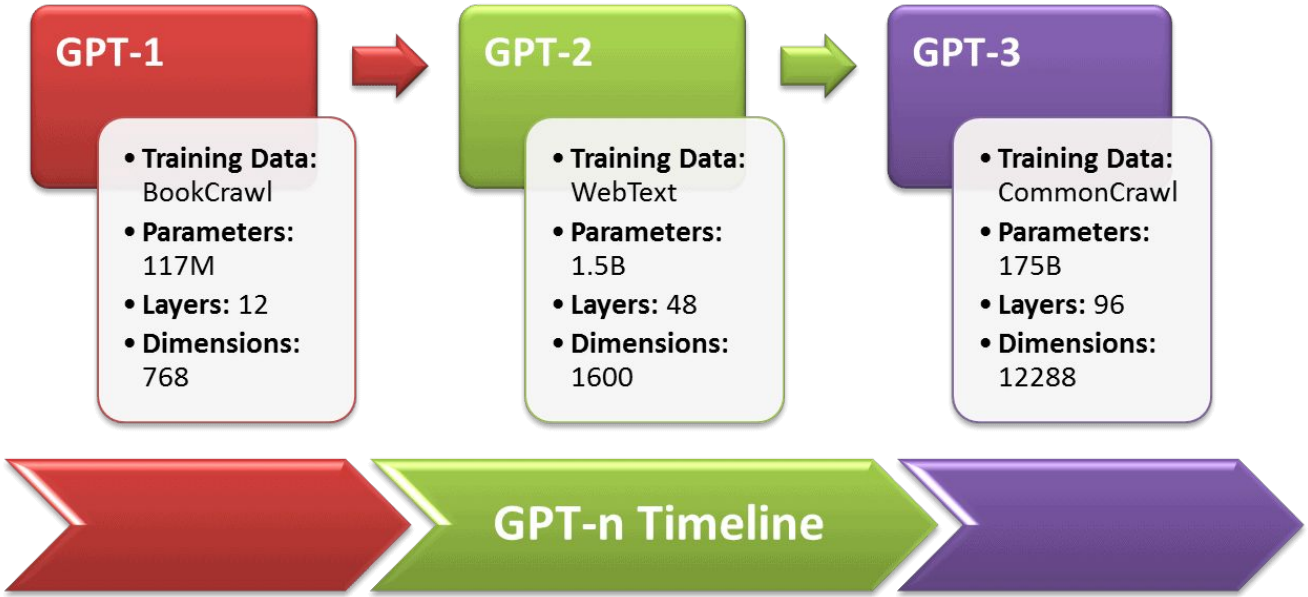


Image source: businessolution.org

Unsupervised learning

OpenAI GPT models:

GPT-3 (2021)**:** 175 billion parameters.

It is impossible to create datasets of that size through manual labeling. A neural network of this scale needs huge amounts of data to not overfit.

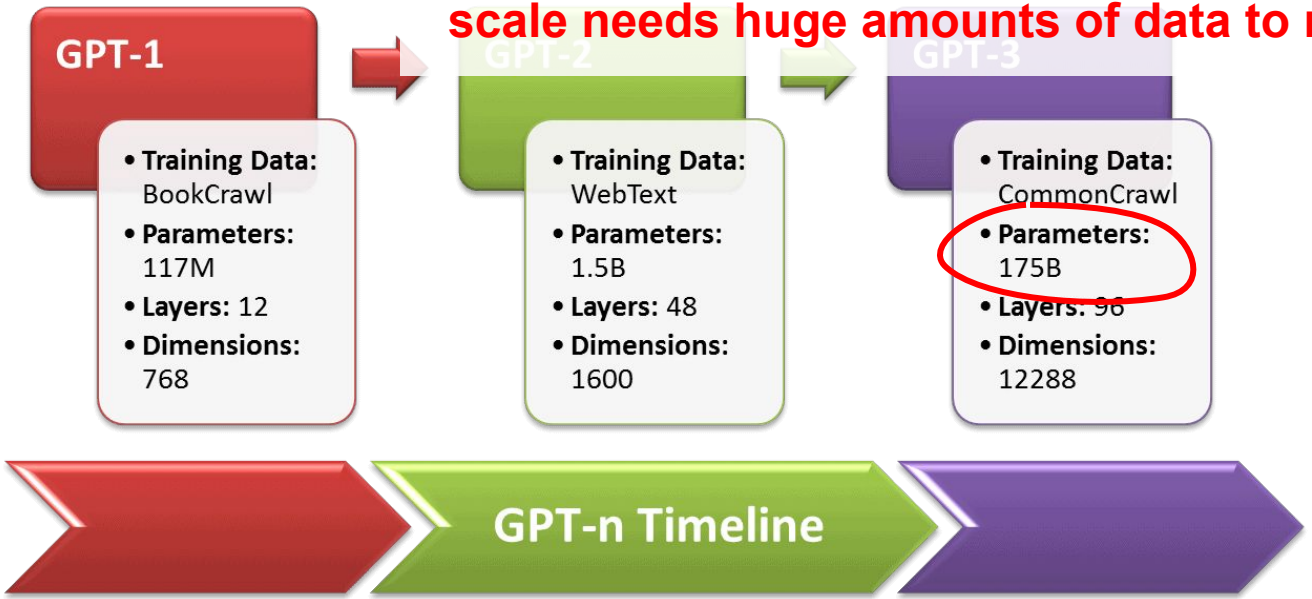


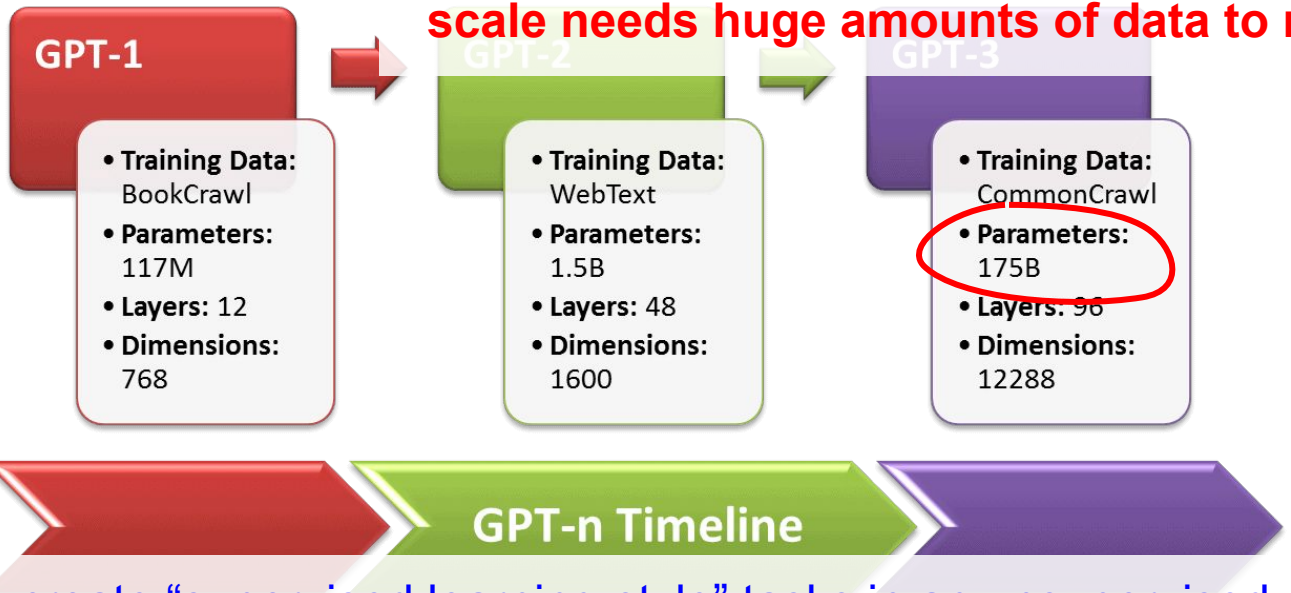
Image source: businessolution.org

Unsupervised learning

OpenAI GPT models:

GPT-3 (2021)**:** 175 billion parameters.

It is impossible to create datasets of that size through manual labeling. A neural network of this scale needs huge amounts of data to not overfit.

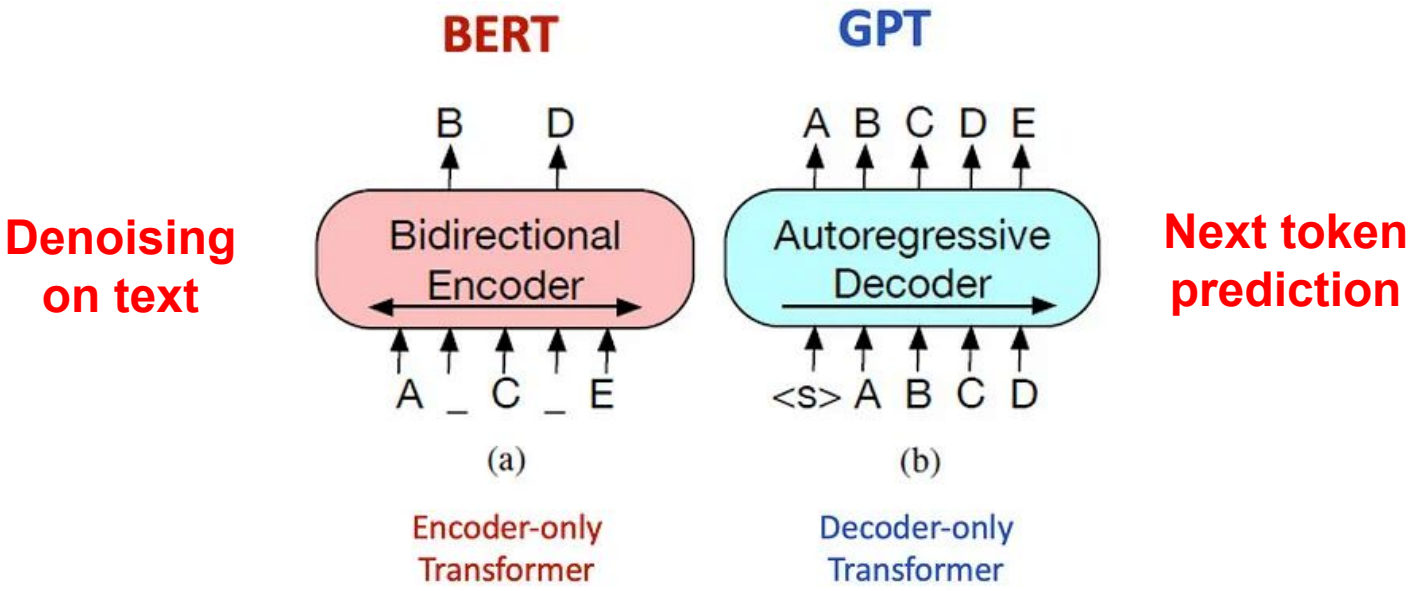


Idea: Let's create "supervised learning-style" tasks in an unsupervised manner, so that we can train large neural networks with even more data!

Image source: LM Po @ medium.com

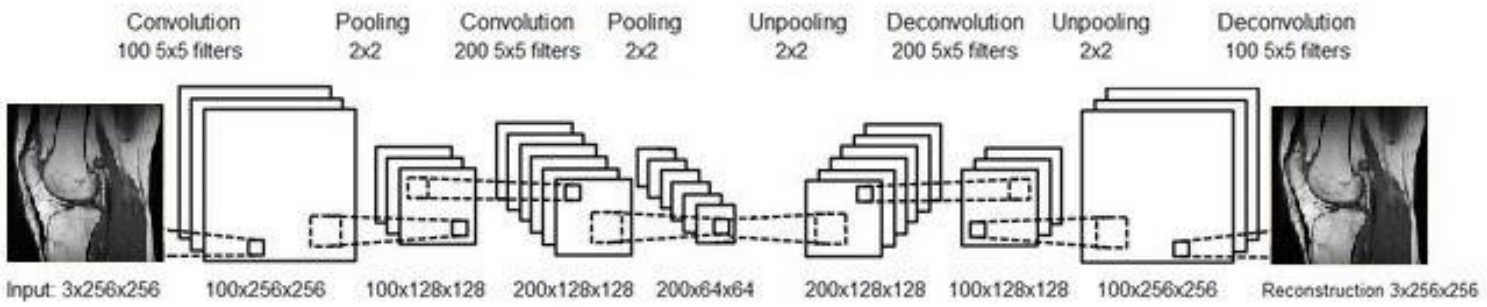
State-of-the-art methods in unsupervised learning

Natural Language Processing - **Noise Filtering / Next Token Prediction** as an unsupervised learning task.



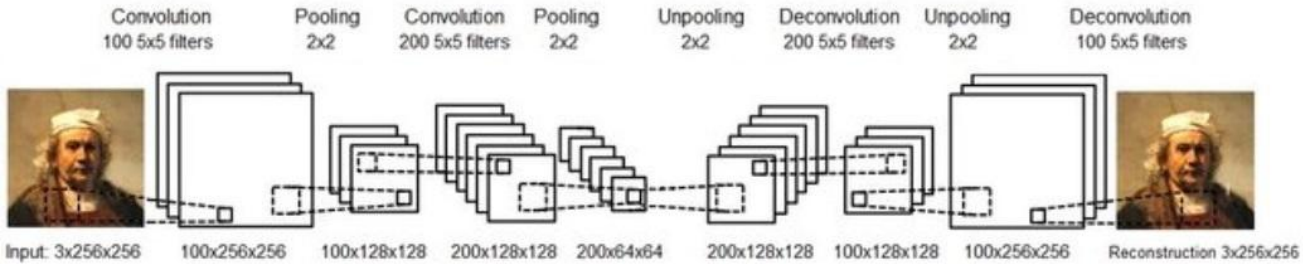
Transfer learning

With the help of **unsupervised pre-training**, our very large neural networks can be trained on more data than ever before!

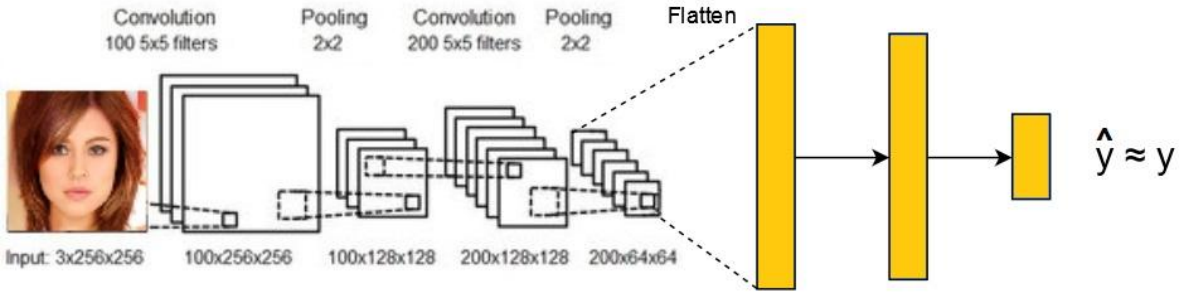


Transfer learning

1) **Unsupervised pre-training** on large (possibly unlabeled) datasets



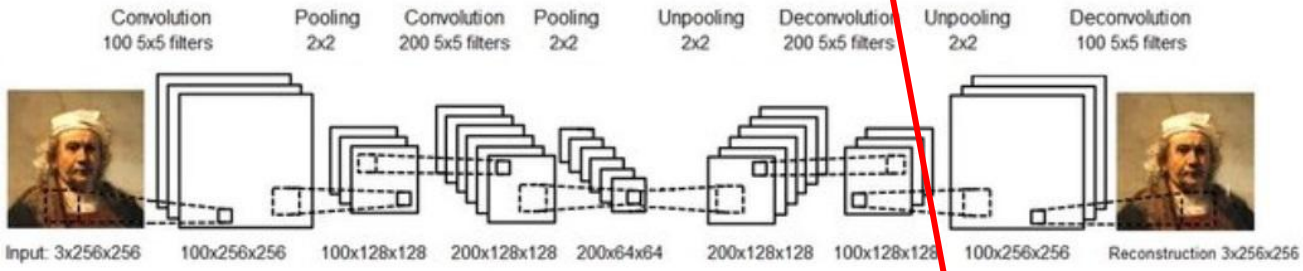
2) **Supervised fine-tuning** - a smaller dataset can be sufficient.



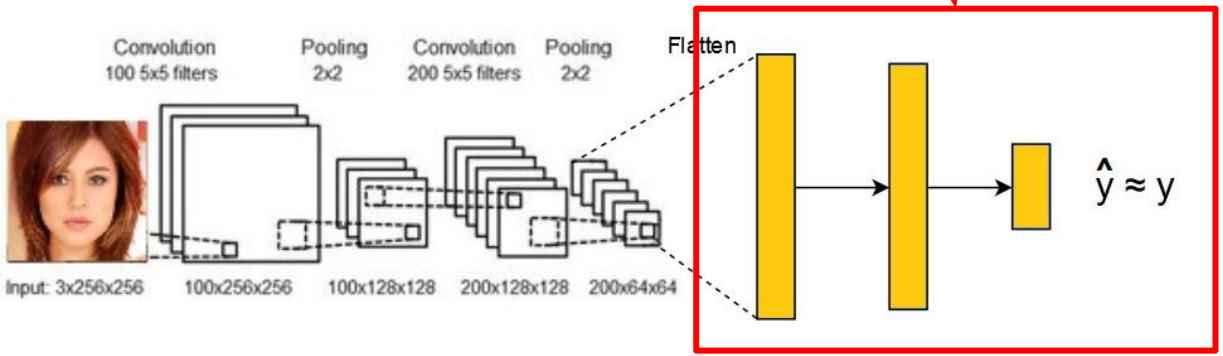
Transfer learning

Replacing part of the network with the layers required for the target task

1) **Unsupervised pre-training** on large (possibly unlabeled) datasets



2) **Supervised fine-tuning** - a smaller dataset can be sufficient.

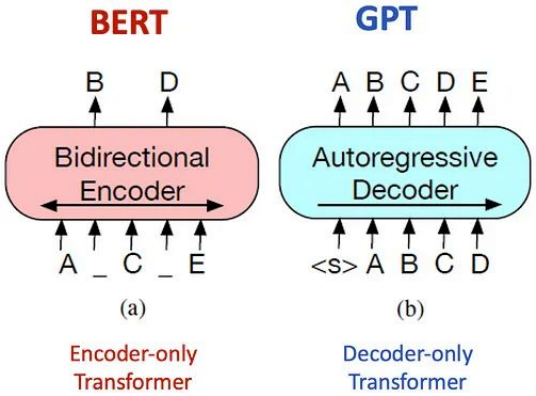


freezing (some of) the pre-trained weights is optional…

Image source: LM Po @ medium.com

State-of-the-art methods in unsupervised learning

Transfer learning in Natural Language Processing



- **Author identification**
- **Finding similar documents**
- **Text sentiment analysis**
- **Text summarization**
- **Machine translation**
- **Text generation**
- **Question answering**
- **Chatbot**

Pre-training (unsupervised)

Fine-tuning on the target (downstream) **task** (partly/fully supervised)

State-of-the-art methods in unsupervised learning

ChatGPT 3.5 (2022)

Fine-tuning Large Language Models – Example

Step 0: Unsupervised pre-training

~ 400 billion tokens

Step 1: Supervised fine-tuning

~ 10k question-answer pairs

Step 2: Training a critic model with human feedback

~ 30-40k questions
~ 100k evaluated answers

Step 3: Fine-tuning with RL using the critic model

State-of-the-art methods in unsupervised learning

ChatGPT 3.5 (2022)

Fine-tuning Large Language Models – Example

Step 0: Unsupervised pre-training

~ 400 billion tokens

The accumulation of the core knowledge

Step 1: Supervised fine-tuning

~ 10k question-answer pairs

Minimal adjustments to align model answers to downstream task requirements

Step 2: Training a critic model with human feedback

~ 30-40k questions
~ 100k evaluated answers

Step 3: Fine-tuning with RL using the critic model

Summary of the semester - Theory (1)

- Main tasks of supervised learning: regression, classification
- Hypothesis function, parameters, loss function, gradient descent
- Linear and logistic regression (artificial neuron)
- Fully connected neural layers, MLP neural network architecture
- Overfitting and its management
- Efficient implementation of the gradient method on complicated functions (neural networks): computational graphs, backpropagation algorithm

Summary of the semester - Theory (2)

- Convolution and pooling layers, convolutional neural networks
- Transfer learning, pre-training, fine-tuning
- Training deep neural networks: the gradient instability problem and its solutions (BatchNorm, residual networks)
- Sequence processing (recurrent and attention-based networks)
- Transformer networks and their components
- The goal of unsupervised learning, autoencoder neural networks

Summary of the semester - Practice

- Array programming, NumPy/PyTorch
- Software for implementing neural networks: PyTorch