Introduction AI

Raoul Grouls, 28-02-2024

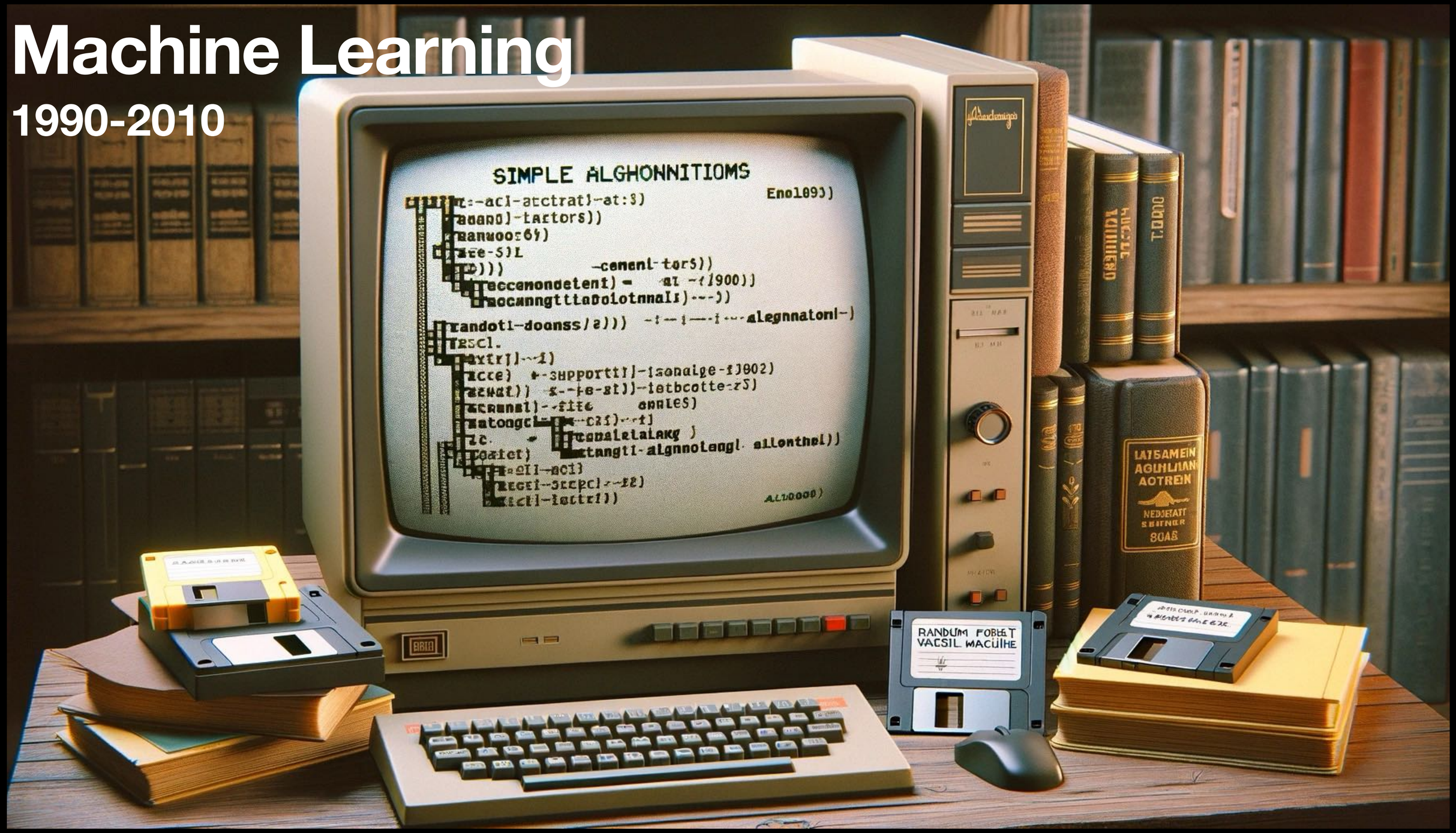
Recap

What would you use and why?

- You want to do traffic flow simulations
- There is not really a solution for your problem in literature; it is highly specific and fast changing; you have some vague ideas about different solutions.
- Your optimization problem would need insane amounts of memory, or time, to compute.

Machine Learning

1990-2010



Deep Learning

The Machine Learning Canvas (v0.4)

Designed for: | Designed by: | Date: | Iteration:

Decisions

How are predictions used to make decisions that provide the proposed value to the end-user?

ML task

Input, output to predict, type of problem.

Value Propositions

What are we trying to do for the end-user(s) of the predictive system? What objectives are we serving?

Data Sources

Which raw data sources can we use (internal and external)?

Collecting Data

How do we get new data to learn from (inputs and outputs)?

how

how

PREDICT

LEARN

GOAL

(what, why, who)

Making Predictions

When do we make predictions on new inputs? How long do we have to featurize a new input and make a prediction?

Offline Evaluation

Methods and metrics to evaluate the system before deployment.

Features

Input representations extracted from raw data sources.

Building Models

When do we create/update models with new training data? How long do we have to featurize training inputs and create a model?

how well

Live Evaluation and Monitoring

Methods and metrics to evaluate the system after deployment, and to quantify value creation.

EVALUATE

Licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

Predict

- Decisions: descriptive, predictive, prescriptive
- Task: classification, regression, clustering?
- Evaluation: what are good metrics?

Learn

- Data sources: what is good data?
- Collecting: how do we get new data?
- Features: what kind of information do you need?

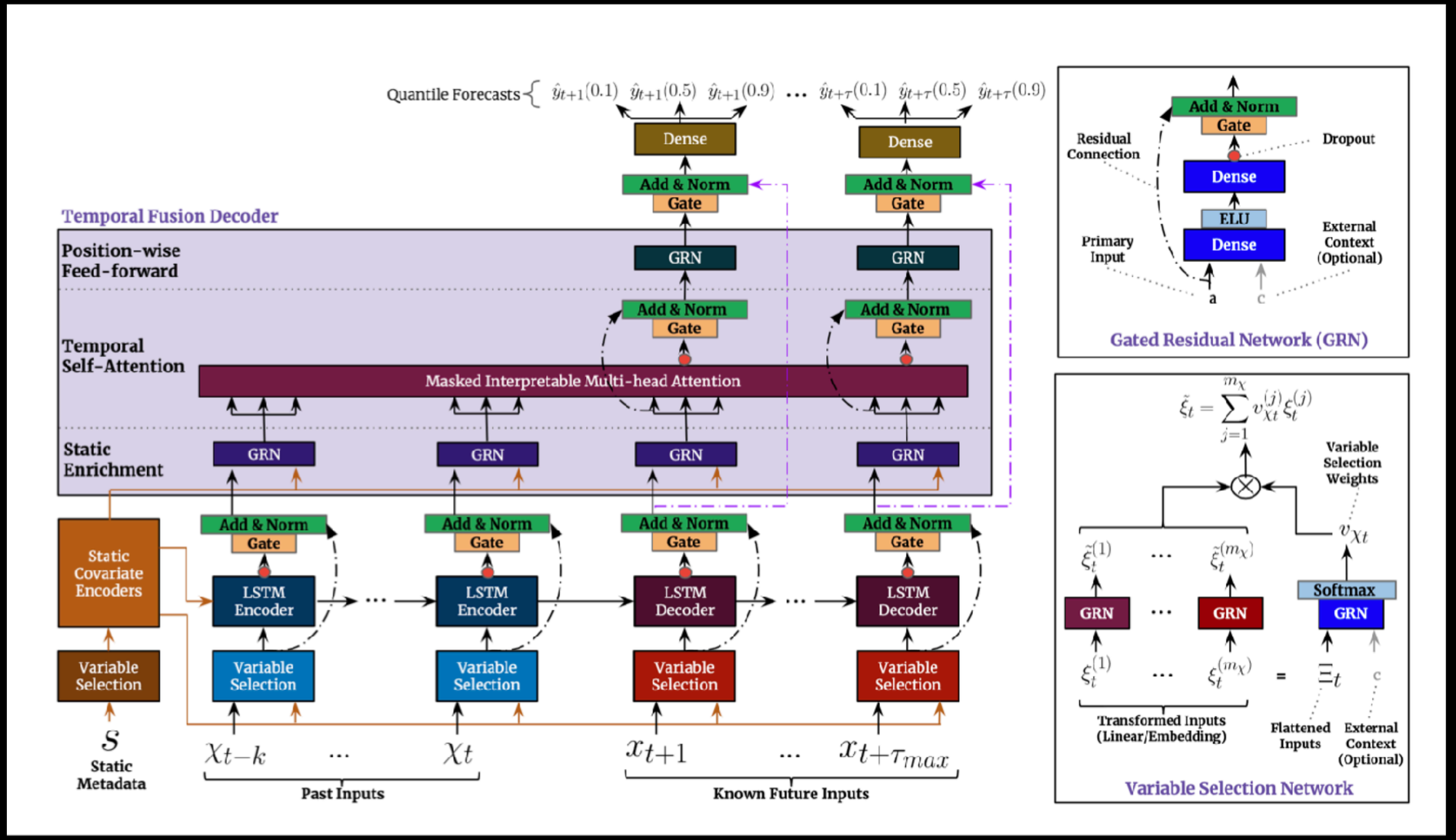
Neural Networks

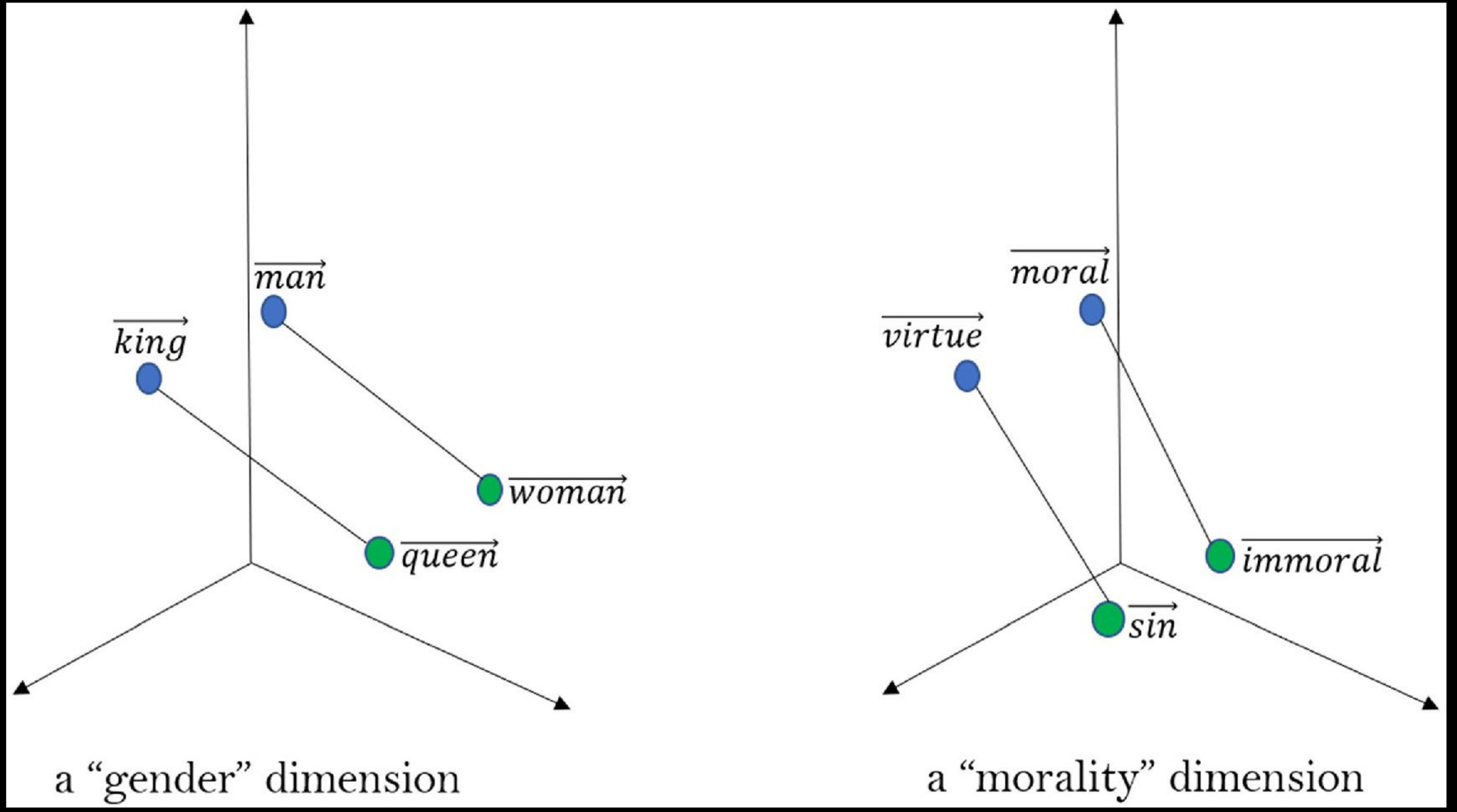
Machine learning

The summary

$$f\colon X \to y$$

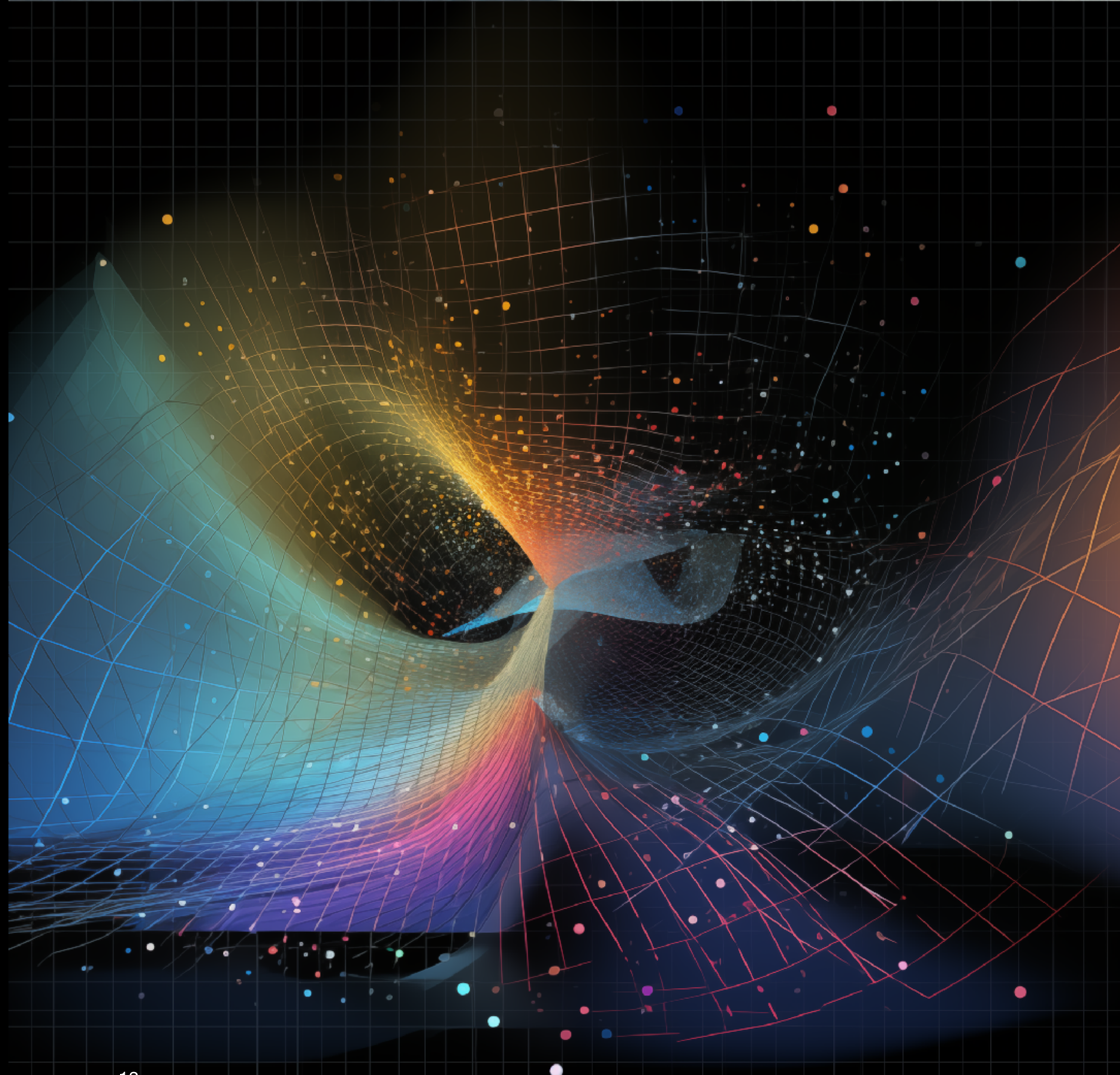
Complexity 2021 - TFT




$\overrightarrow{man}$
$\overrightarrow{king}$
$\overrightarrow{woman}$
$\overrightarrow{queen}$
a “gender” dimension
$\overrightarrow{moral}$
$\overrightarrow{virtue}$
$\overrightarrow{immoral}$
$\overrightarrow{sin}$
a “morality” dimension

LLMs use a semantic vectorspace with 768 dimensions : $\mathbb{R}^{768}$

Atoms in the visible universe : 10^{80}



Linear models

The basic mathematical shape of a linear model is:

$$Y = WX + b$$

- $Y = \{y_1, \ldots, y_m\}$ are labels, so we have m labels.
- Every label Y_i has corresponding features $X_i = \{x_{i,1}, \ldots, x_{i,n}\}$, so we have n features.
- We can store the observations in an (m, n) matrix X
- We can store n weights for every feature in a matrix W
- b is an additional bias weight

The matrix notation is concise. We could write this out in full as:

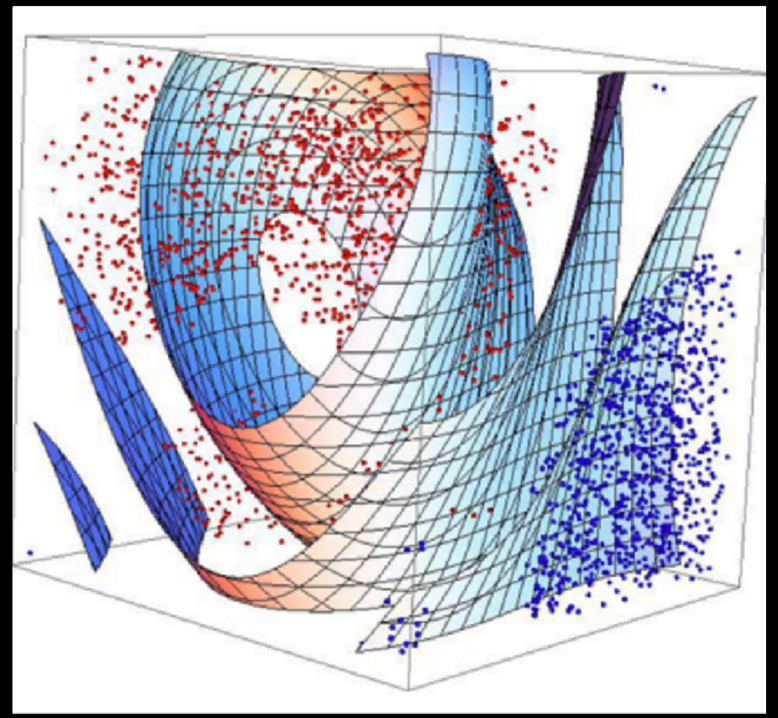
$$y_i = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n + b$$

Non-linear models

A lot of data is non-linear. This means we would need a curved hyperplane.

One trick to do this is the kernel-trick, which is commonly used with Support Vector Machines, which we touched upon in the short history.

Deep learning has found another trick



Deep Learning

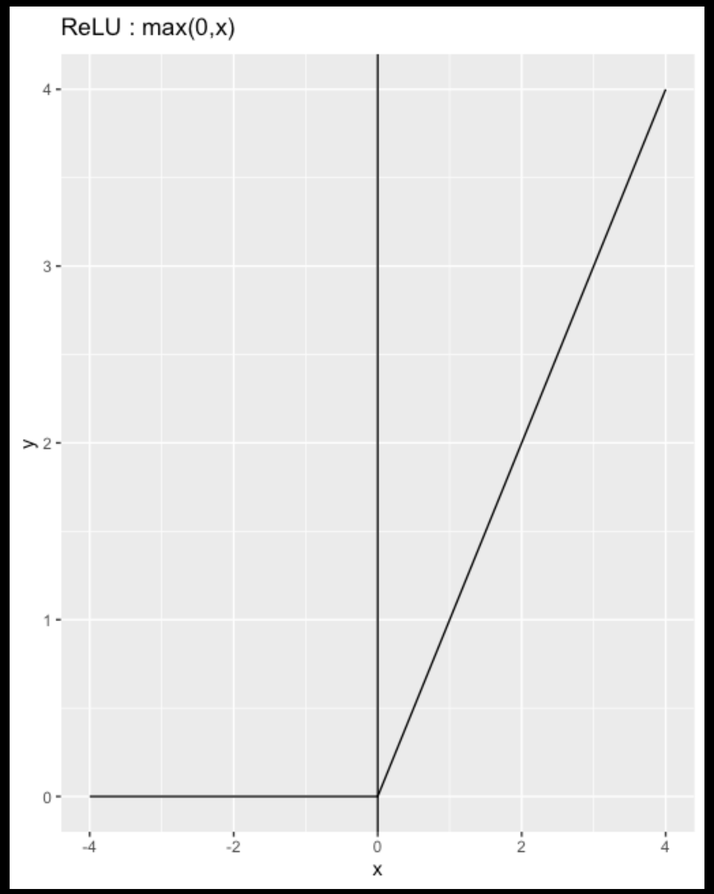
Linear function $f(X) = WX + b$

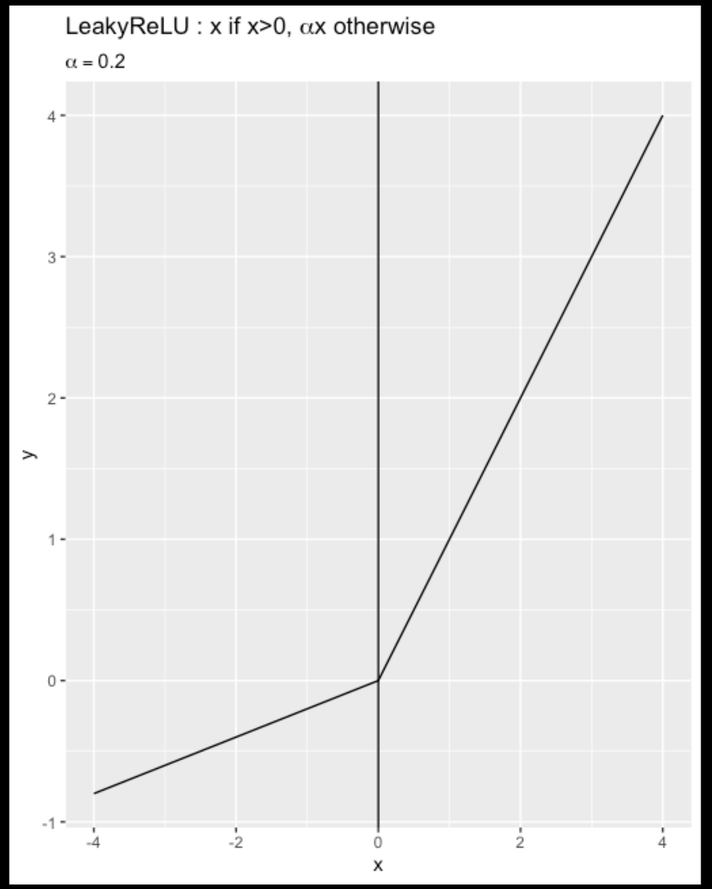
Nonlinear function $\sigma(X) = max(0,X)$

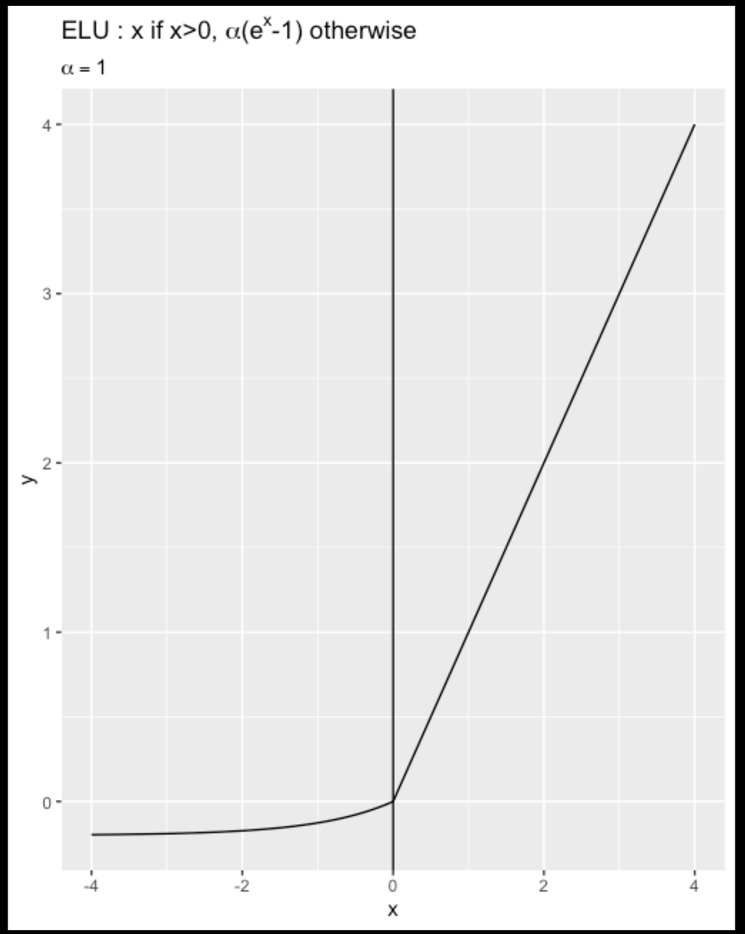
Neural network $\hat{y} = f_n \circ \sigma \circ f_{n-1} \circ \ \ldots \ \circ \sigma \circ f_1$

Activations

non-linearities



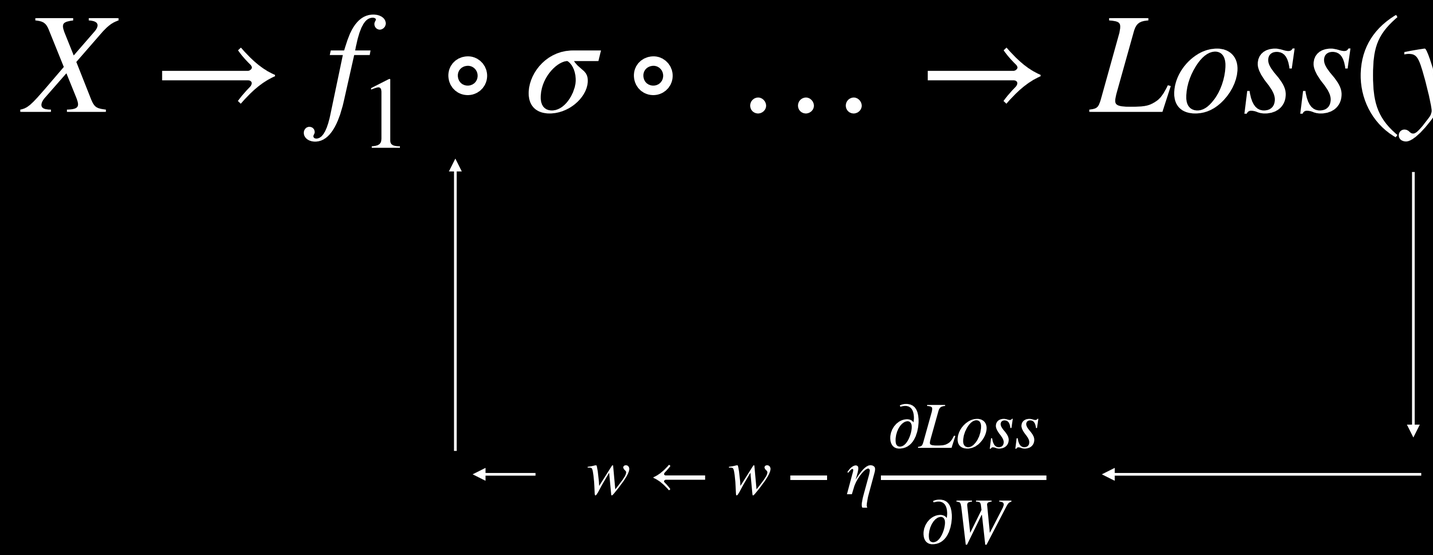




Deep Learning

Data	$X = \{\overrightarrow{x_1}, \ldots, \overrightarrow{x_n} \mid \vec{x} \in \mathbb{R}^d\}$
	$y = \{y_1, \ldots, y_n \mid y \in \{0,1\}\}$
Trainable Weights	W, b
(Non)linearity	$f(X) = WX + b \quad \sigma(X) = max(0,X)$
Predict	$\hat{y} = f_n \circ \sigma \circ f_{n-1} \circ \ \ldots \ \circ \sigma \circ f_1$
Loss	$Loss(y, \hat{y})$
Optimize	$w \leftarrow w - \eta \frac{\partial Loss}{\partial W}$

Training


$X \rightarrow f_1 \circ \sigma \circ \ \ldots \ \rightarrow Loss(y, \hat{y})$
$w \leftarrow w - \eta \frac{\partial Loss}{\partial W}$

Convolutional Neural Networks

Convolutions - the motivation

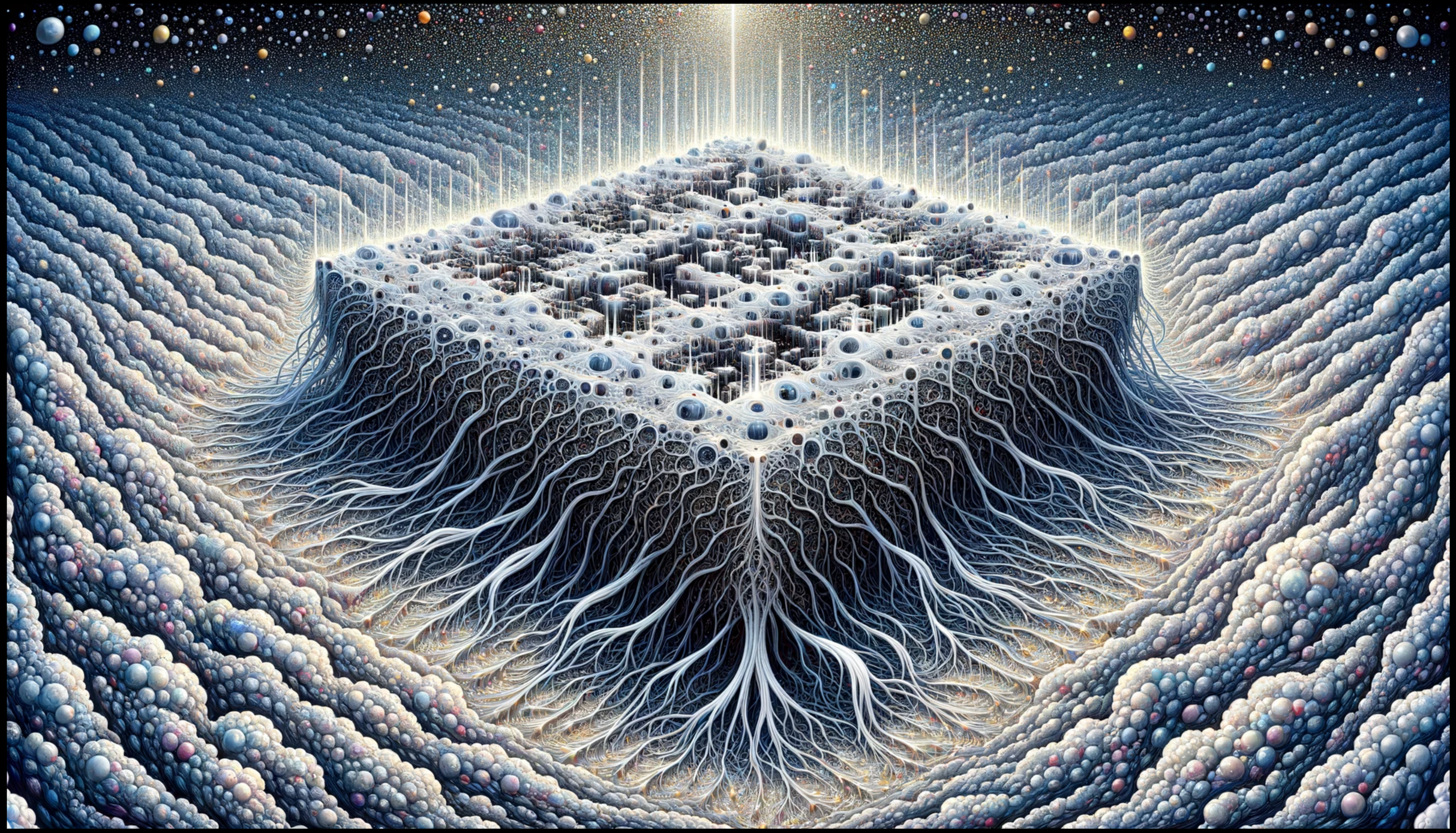
An image of size 28x28 has 784 pixels

With 256x256 you are already at 65536 pixels

Treating every pixels as a feature will blow up the amount of parameters for your model:

Assuming you halve the dimension, a batch with dimensions (32, 65536) will need a (65536, 32000) weight matrix.

That are over 2×10^9 parameters, just for the first layer…

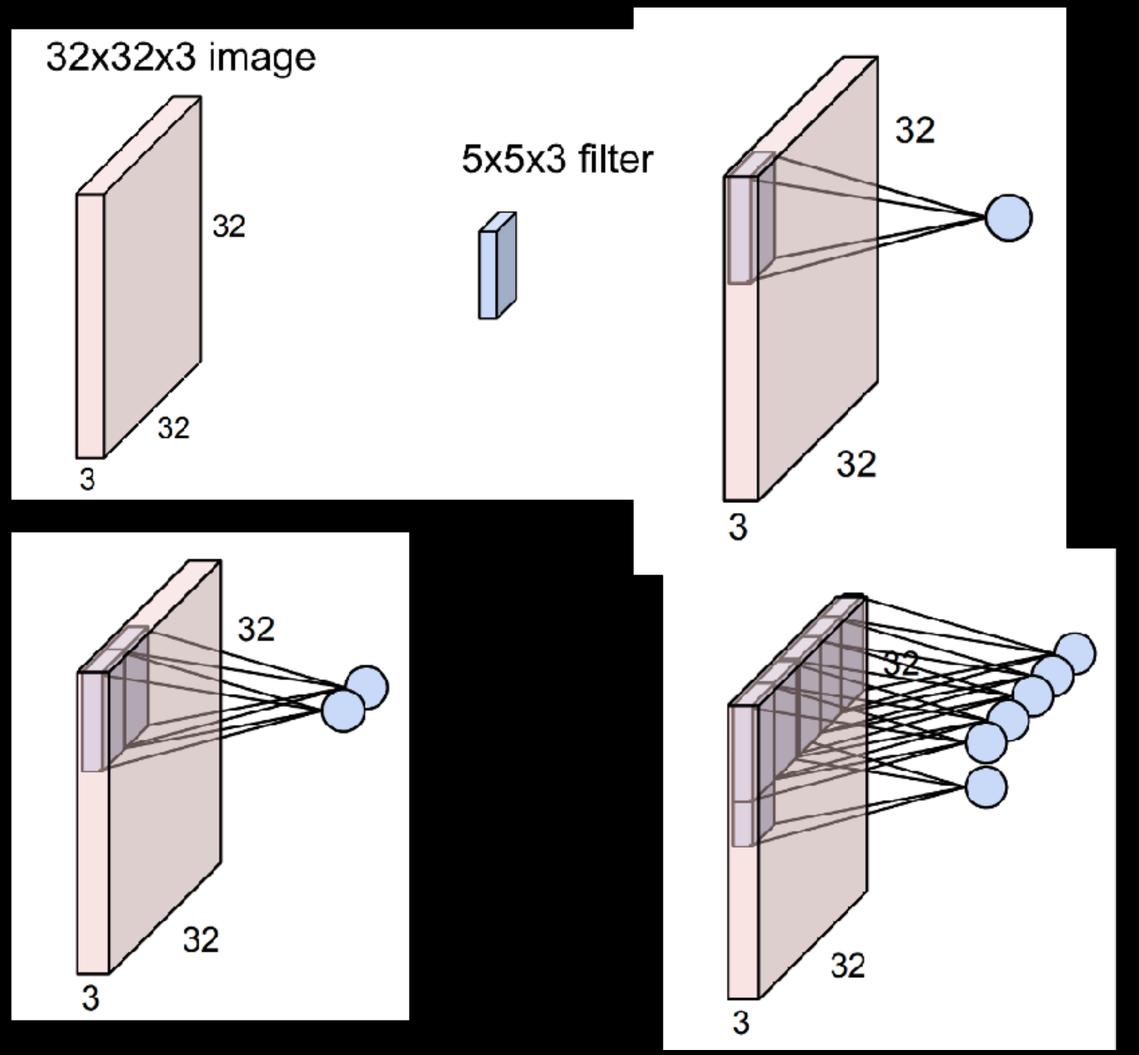


Convolutions

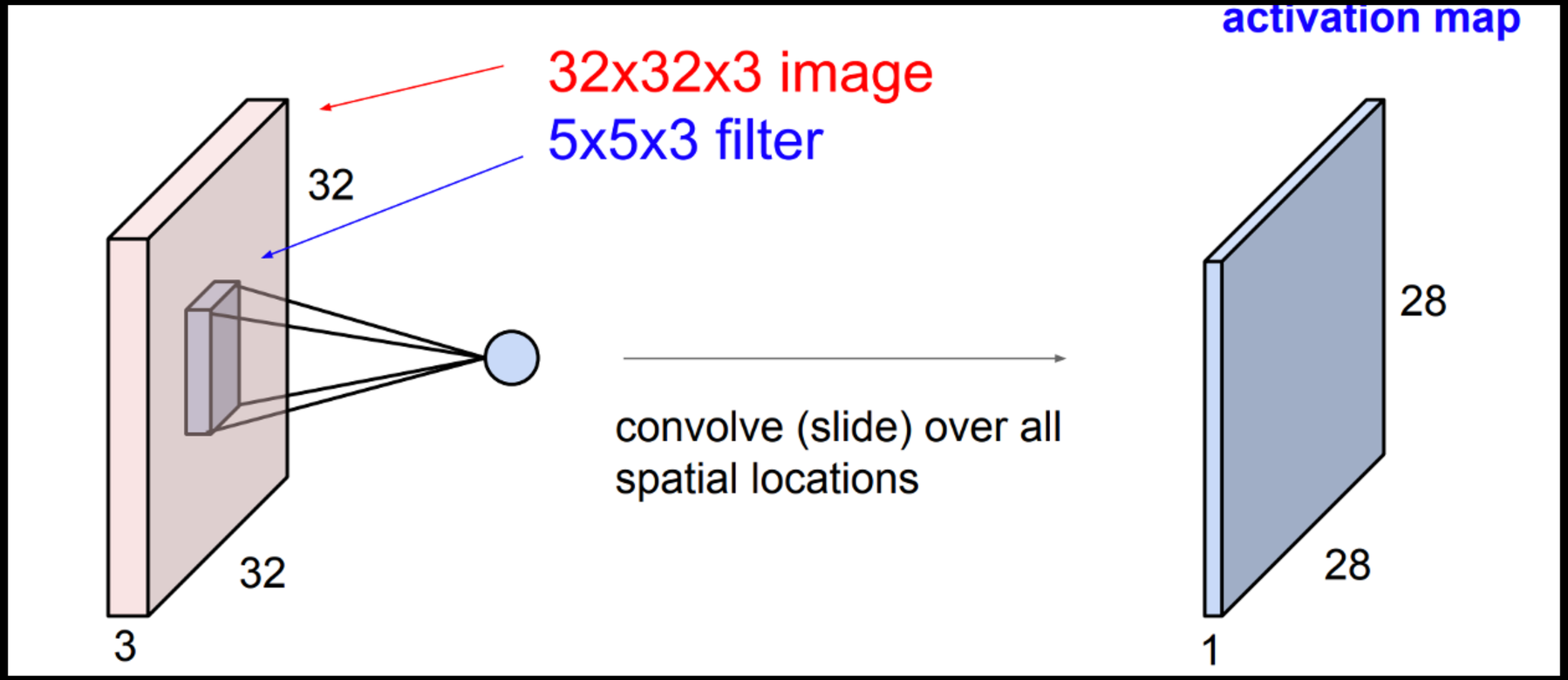
- Calculate the dot product between the filter and the image
- E.g.,

$$\begin{bmatrix} 1 & 10 \\ 2 & 20 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix} =$$

$$(1 * 1) + (10 * 2) + (2 * 2) + 0 = 25$$



Convolutions

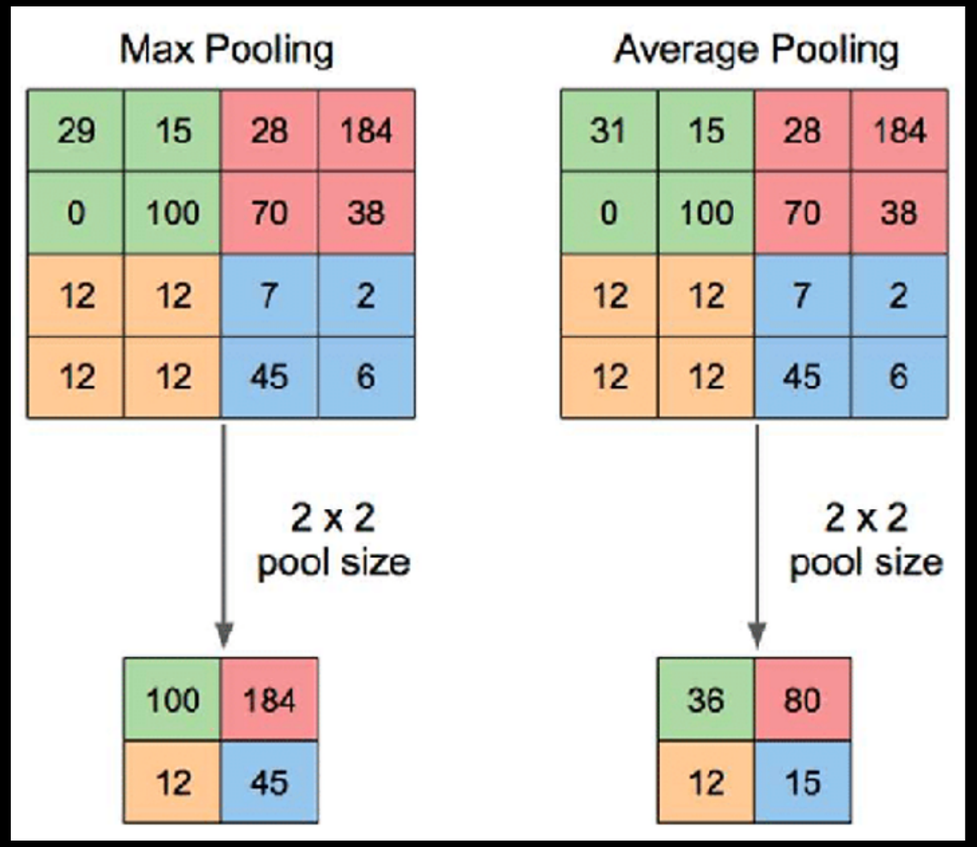


Convolutions

- In deep learning, we won’t construct the filters for ourselves!
- We only define the parameters (eg size of the kernel, how many) and let the model learn the convolutions, based on the data and labels.

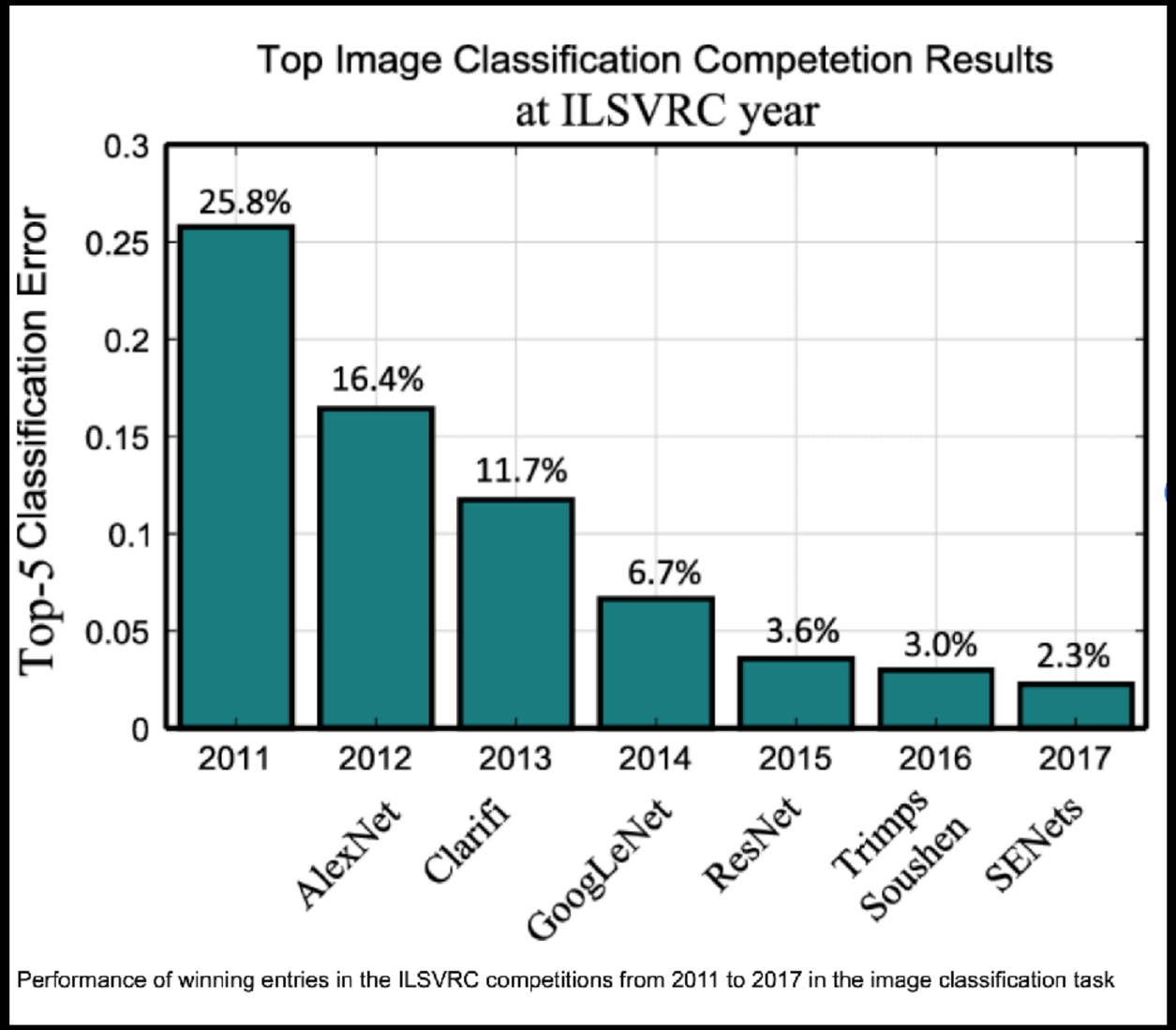
Pooling

- A way to downsample the input
- max pooling: focus on the highest value
- average pooling: smooth the pixels



Imagenet Challenge

- Human performance is about 5%



Performance of winning entries in the ILSVRC competitions from 2011 to 2017 in the image classification task

AlexNet

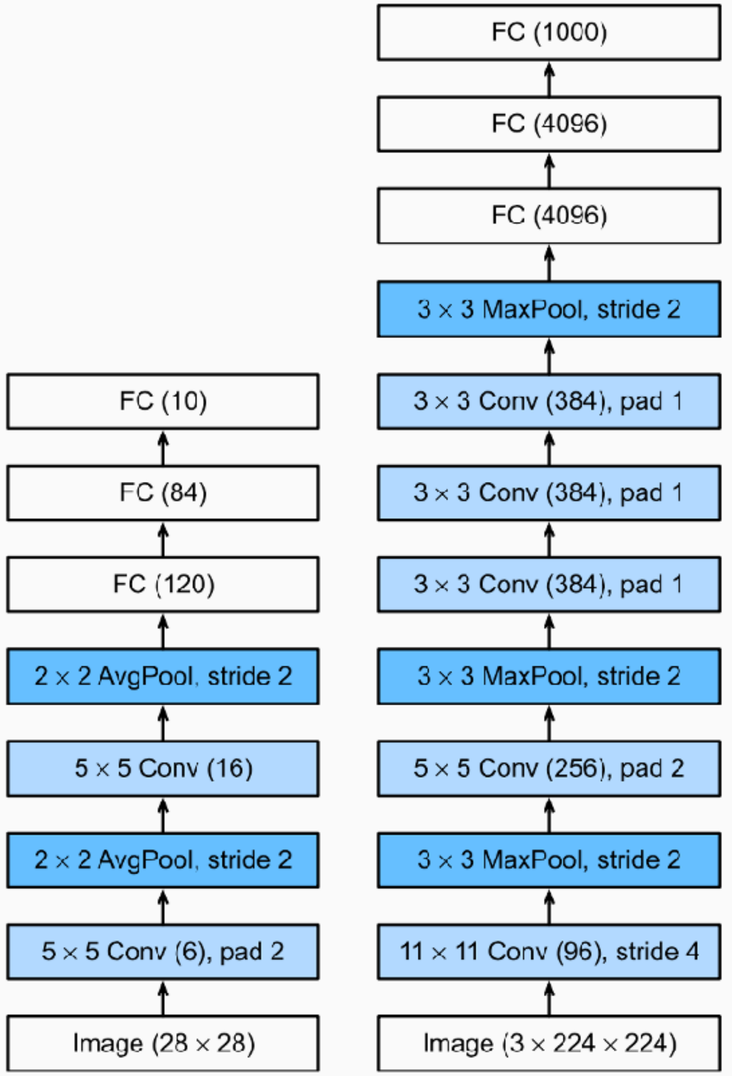


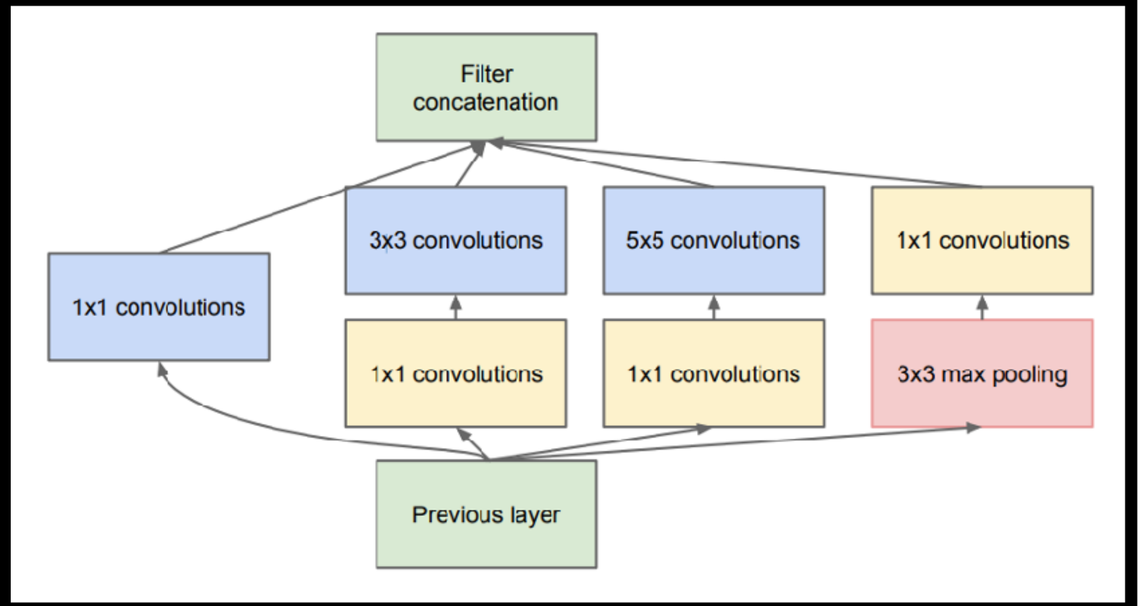
Fig. 7.1.2 From LeNet (left) to AlexNet (right).

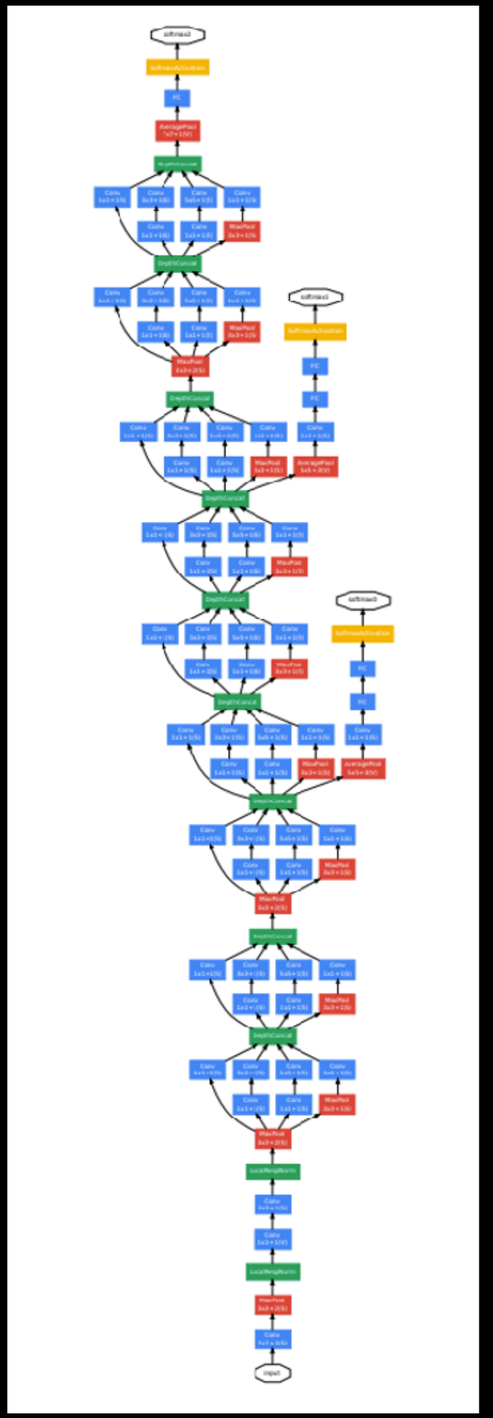
VGG

Changes in:
Filter size
channel numbers
depth
138 million parameters

GoogleNet

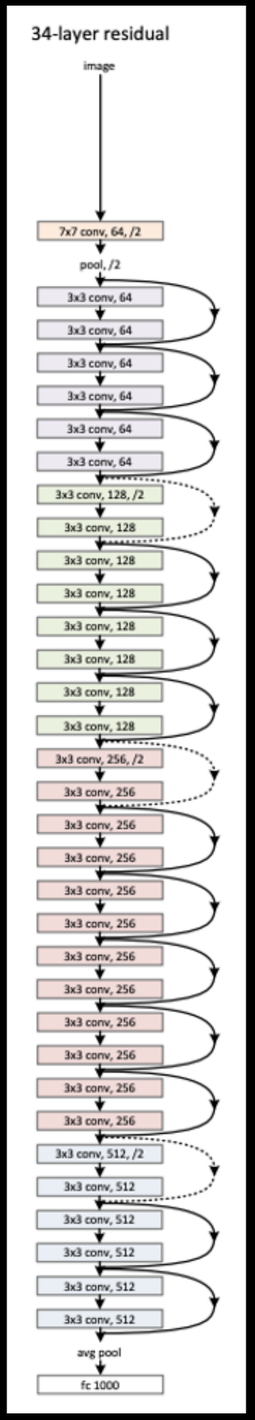
Parallel filters
Bottleneck layers
6.7 million parameters





ResNet

batch normalisation
Residual blocks
11 million parameters for ResNet18

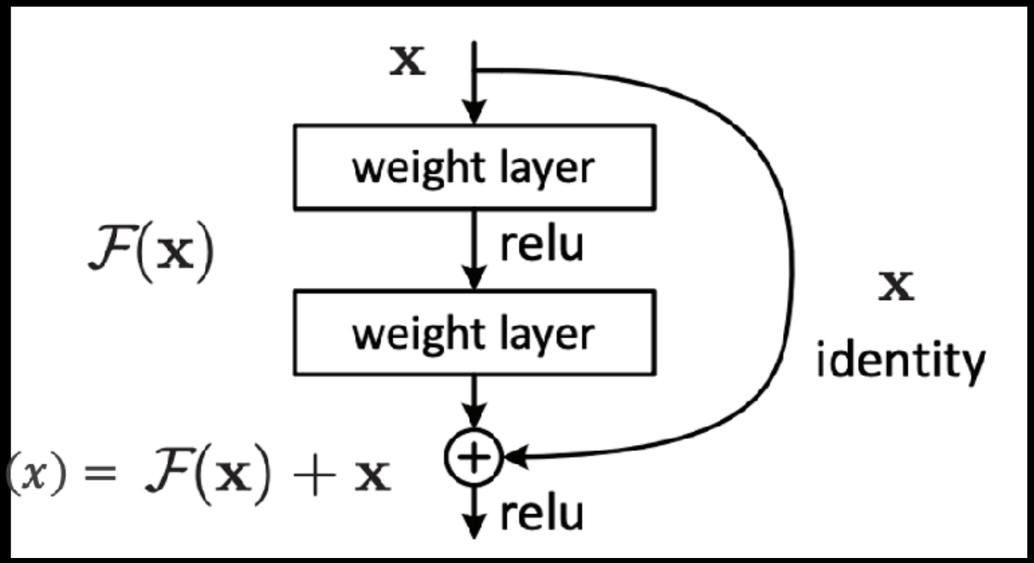


ResNet

Learn the residual, instead of the full mapping

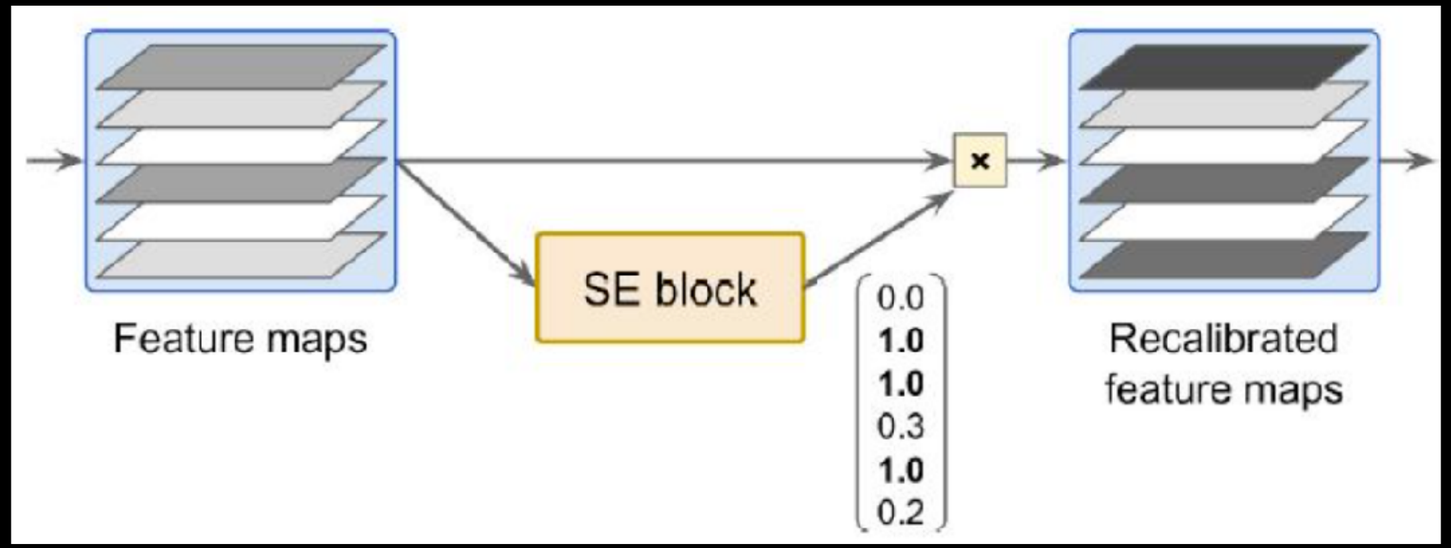
E.g.,“use the same settings, but just change x” is easier than starting from scratch.

It is easier for the gradient to flow back



SEnet

- Which features are most likely to be activated together?
- E.g., if you see a left eye, you expect a right eye. Even if it is hidden in the image.

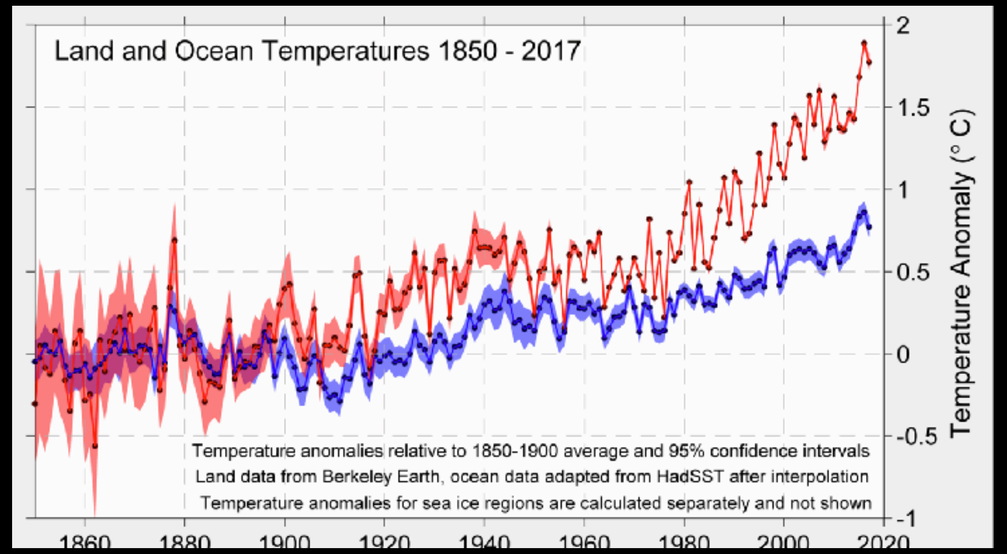


Motivation for RNNs

With sequences, the past offers context:

- Ik krijg geld van de bank
- Ik wil een nieuwe bank aanschaffen

We need the past to make sense of the future.



Simple RNN

To incorporate the hidden state, we simply add it:

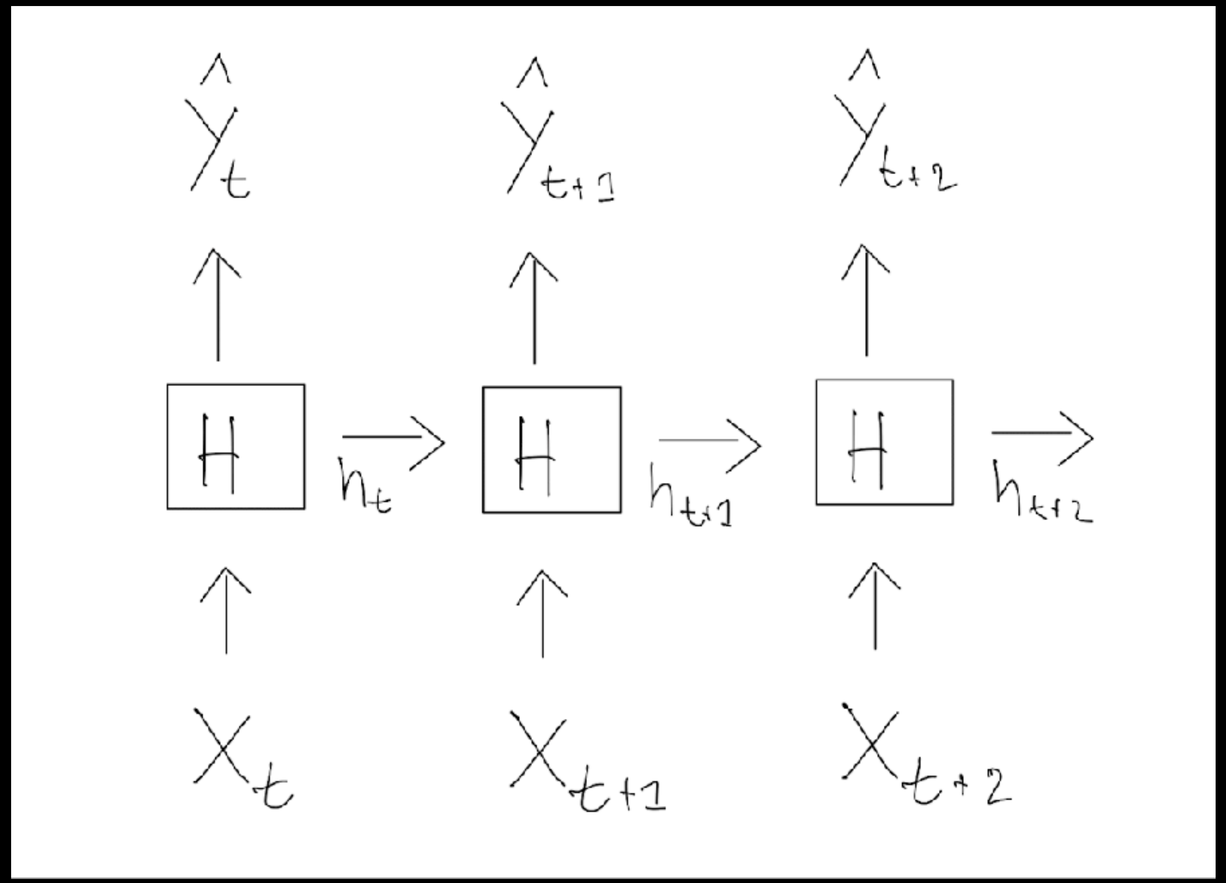
$$h_t = \sigma(W_x X_t + W_h h_{t-1} + b)$$

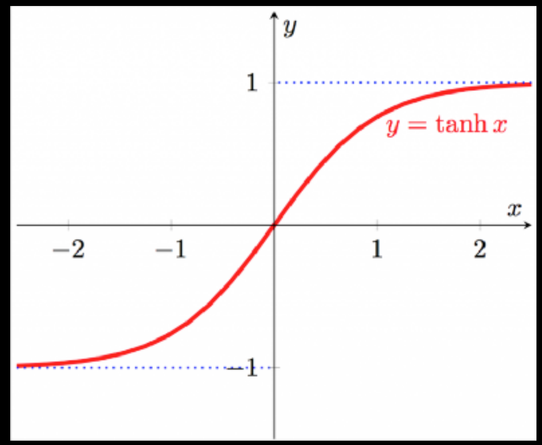
This is equivalent to

$$h_t = \sigma(W[X_t, h_{t-1}] + b)$$

where $[X, h]$ means concatenate

σ is an activation function, typically $tanh$



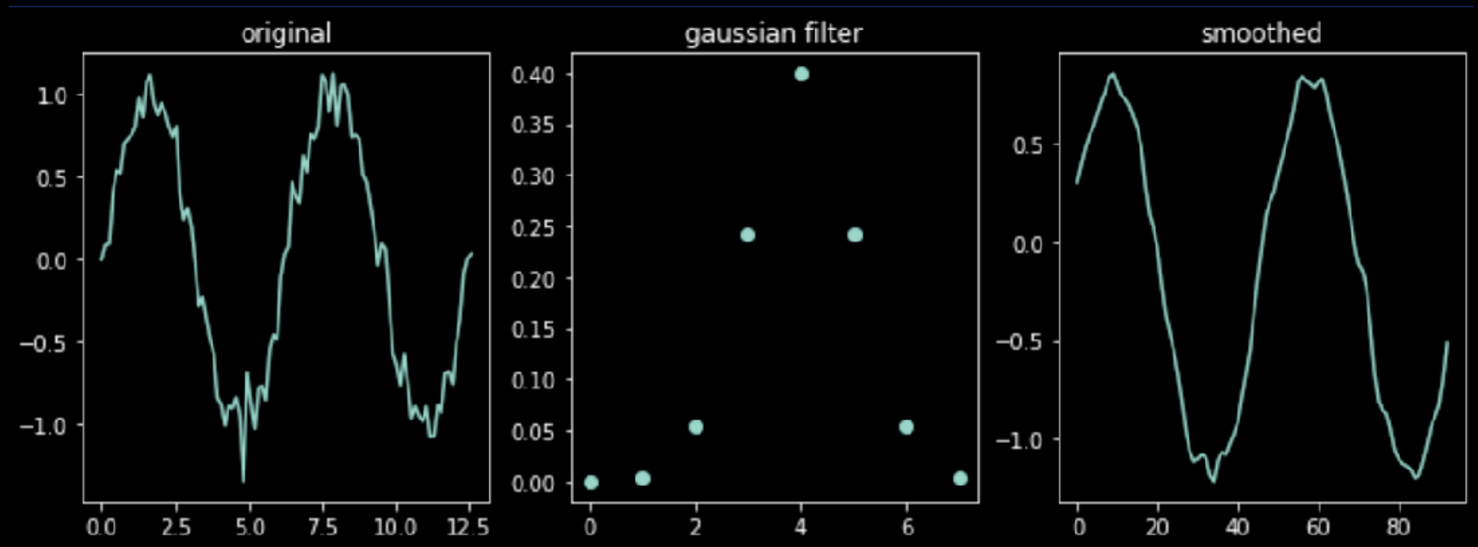


Transformers

Motivation Transformers

local weighing is not enough

- We have been using convolutions
- These look at local context
- But what about more distant relations?



Transformers: mixing in context

Let ξ_i be the vector that describes the semantic meaning of word i.

We can imagine the we want to “mix in” other words, some more than others. A simple way would be with multiplication.

This way, we obtain a vector of numbers, one for every word in the sequence.

	ξ_1	ξ_2	ξ_3
ξ_1	$\xi_1\xi_1$	$\xi_1\xi_2$	$\xi_1\xi_3$

Reweighing relevance

For the meaning of ξ_1, let ξ_2 be irrelevant, and ξ_3 be highly relevant.

We would want $\xi_1\xi_2$ to be closer to zero, and $\xi_1\xi_3$ to be closer to one.

If we normalize all weights with a softmax, we make sure everything adds to 1

We can now reweigh the original ξ_1 to obtain $\tilde{\xi_1}$

And it will contain a little bit of ξ_2 context but much more of ξ_3 context

$$W = softmax(\begin{bmatrix}\xi_1\xi_1 & \xi_1\xi_2 & \xi_1\xi_3\end{bmatrix})$$

$$W = [0.4 \quad 0.1 \quad 0.4]$$

$$\tilde{\xi_1} = w_1\xi_1 + w_2\xi_2 + w_3\xi_3$$