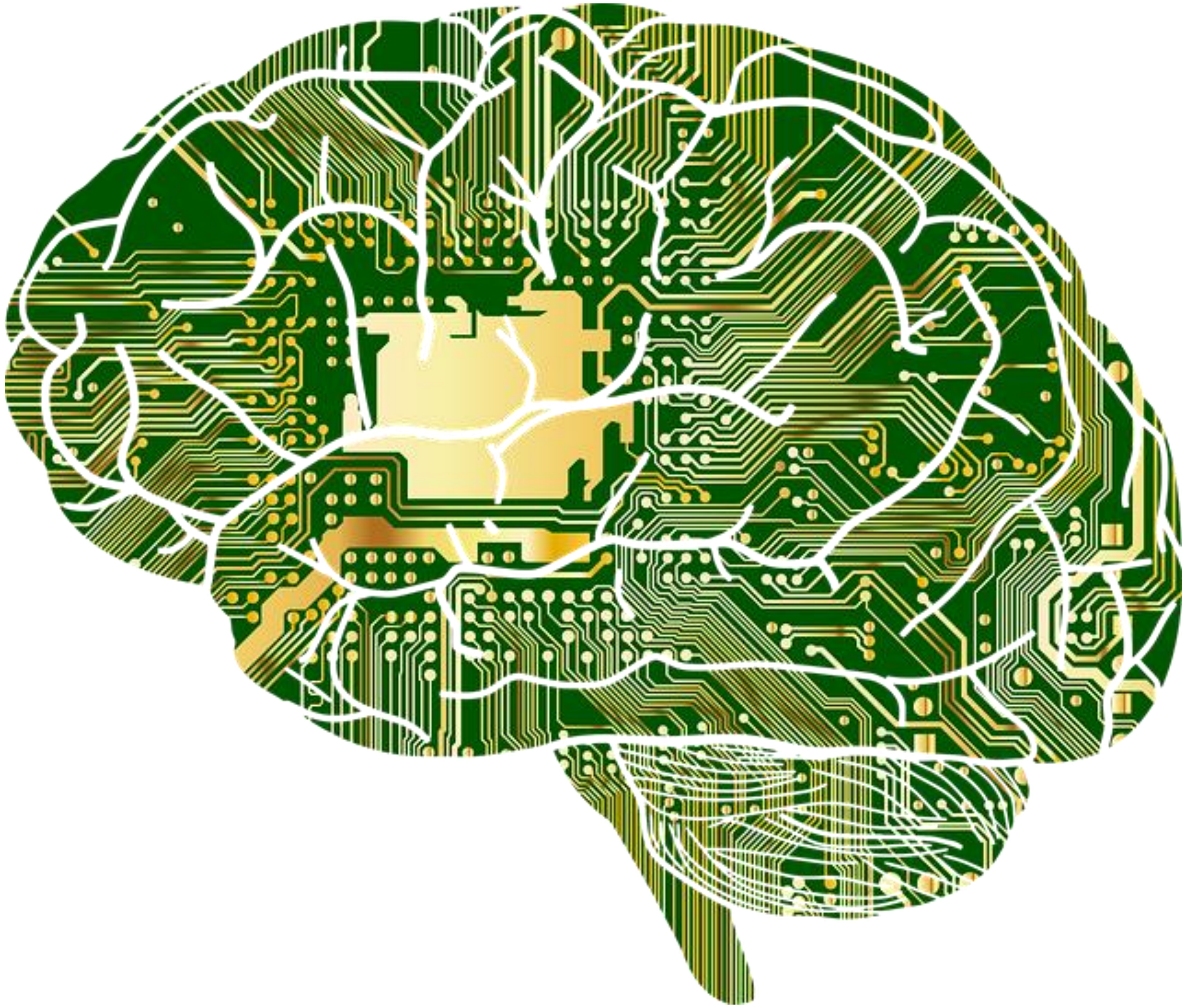
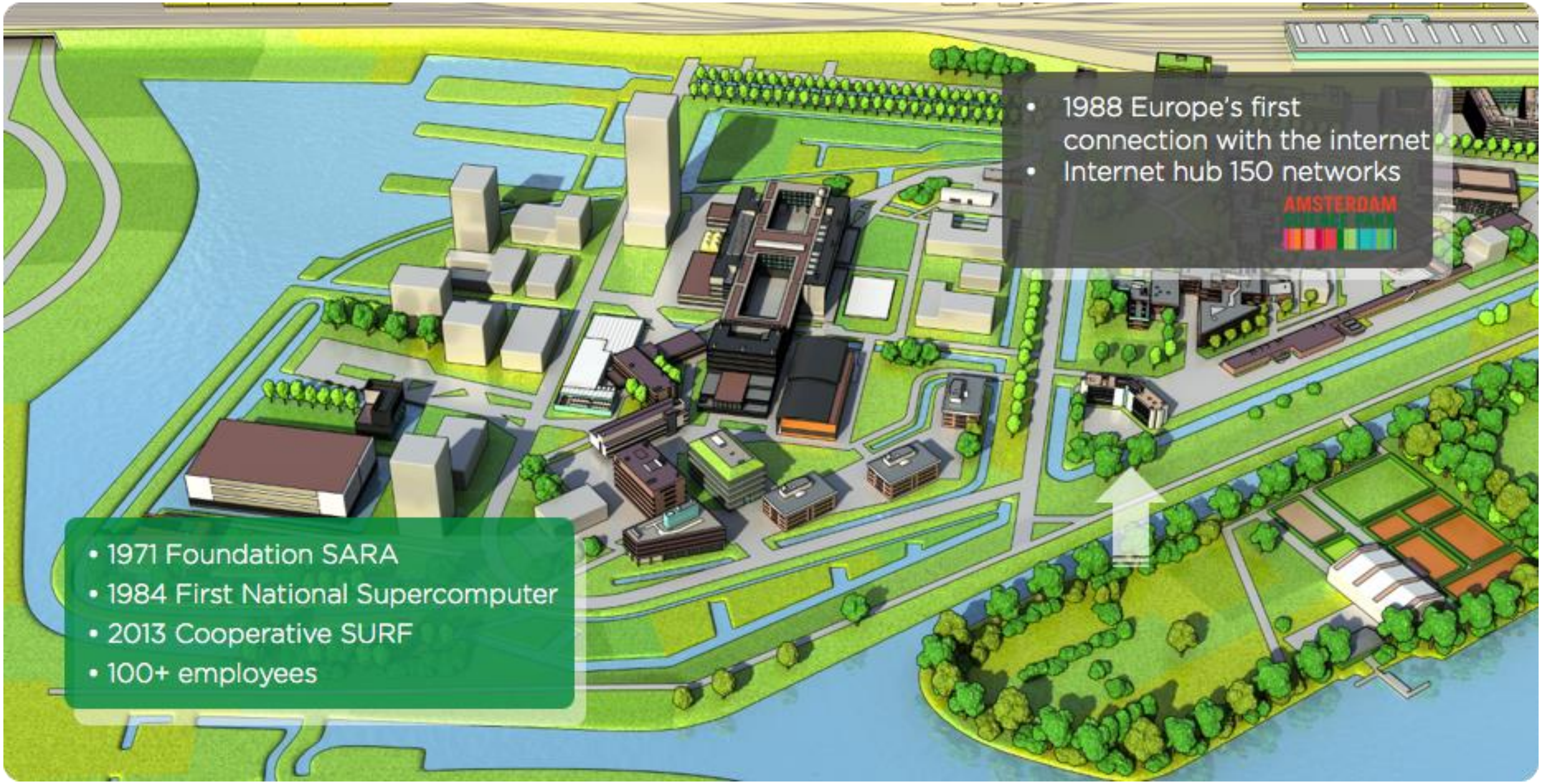


Deep learning

Introduction

Caspar van Leeuwen, David Ruhe

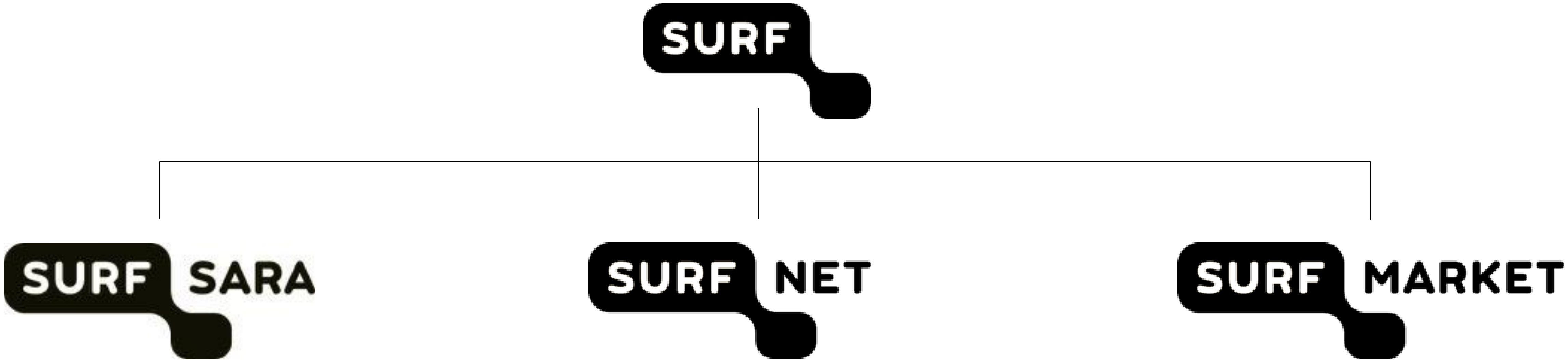
SURFsara



- 1971 Foundation SARA
- 1984 First National Supercomputer
- 2013 Cooperative SURF
- 100+ employees

- 1988 Europe's first connection with the internet
- Internet hub 150 networks

SURF

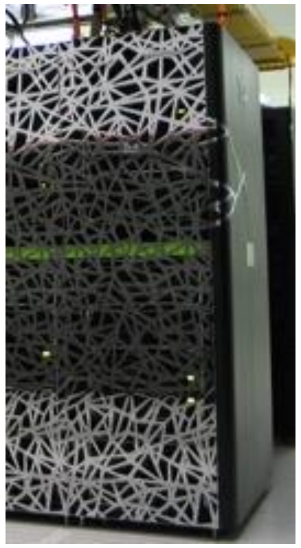


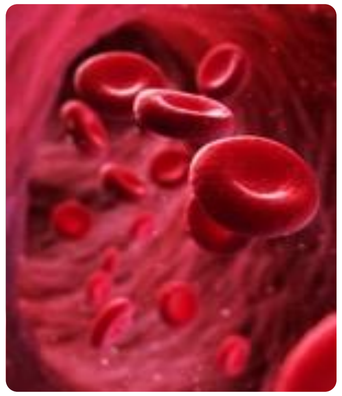
SURF

SURF SARA

SURF NET

SURF MARKET





SURFsara

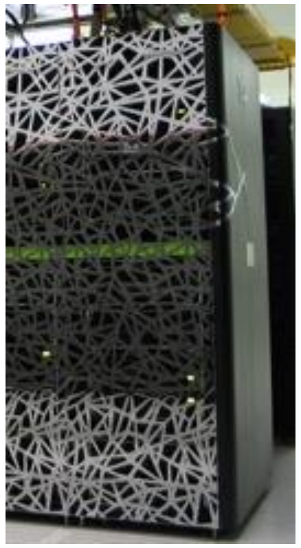
High performance computing

Supercomputing

Clustercomputing

Machine learning

HPC cloud



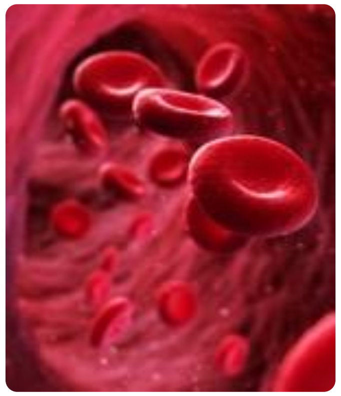
Data processing

Data analytics

Grid services

Visualization

HPC cloud



Data services

Mass online storage

PID service

Data management

Data preservation

Who are we?

Caspar van Leeuwen

- Advisor Cluster/Supercomputing
- Support users in using Cartesius / Lisa
- Strong interest in Deep Learning
- Other activities, e.g. Training

David Ruhe

- Advisor Machine Learning
- Support ML workloads on HPC infrastructure
- Other activities, e.g. joint (ML) research projects

Course information

- The course will be “hands-on".
- We will use **R**, the programming language.
  - Disclaimer: R is *not* a common HPC language, but was thought to fit best with your backgrounds. We are sure we can manage, but are not fluent in R syntax!
- We will use **Keras**, a popular deep learning framework.
- Online Jupyter notebook environment - later today
- Syllabus, announcements & discussion board @ Canvas

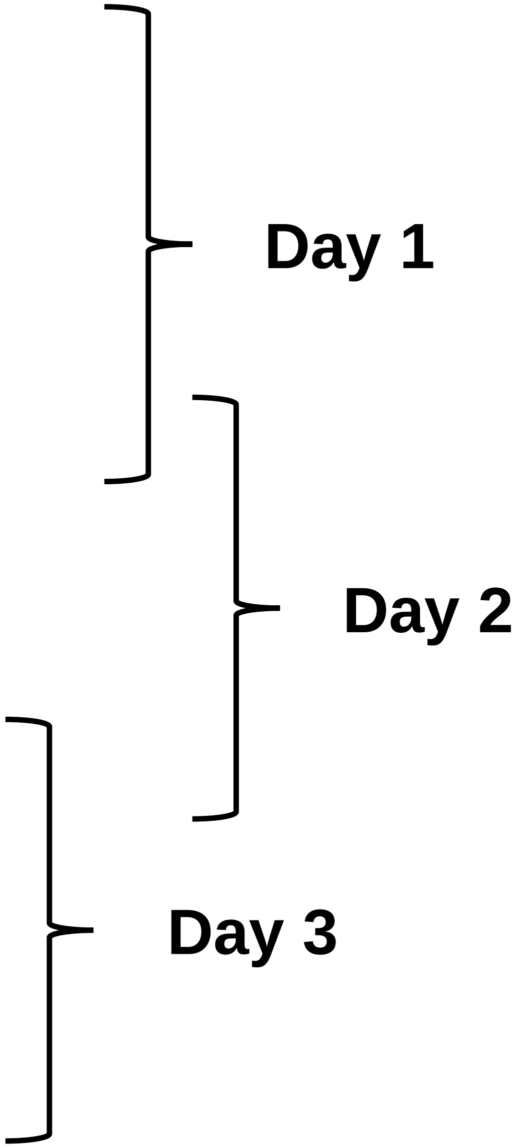
Your backgrounds

Any experience in

- Programming
- R
- Statistics
- Machine learning

Course plan

1. **Introduction and ML & DL fundamentals**
2. **Optimisation & network training**
3. ML tasks (regression vs classification) & regularisation
4. Sequential data analysis with recurrent neural networks
5. Image processing with convolutional neural networks
6. Unsupervised learning
7. Complex architectures and models



Day 1

Day 2

Day 3

Today’s program

14:00-14:30 What is ML / DL? What is a neuron?

14:30-15:30 Hands on: building a neuron from scratch

15:30-15:45 How do neural networks work?

15:45-16:15 Break

16:15-16:30 How do neural networks work? (continued)

16:30-17:15 Hands on: building a network from scratch

17:15-18:00 Loss function & updating weights

18:00-19:00 Diner

19:00-19:45 Hands on: the XOR problem

19:45-20:00 Dataset splitting & Performance evaluation

20:00-21:00 Hands on: Keras on fashion Mnist

If we finish early: Machine learning tasks (classification vs regression)

Overview

Today we will cover

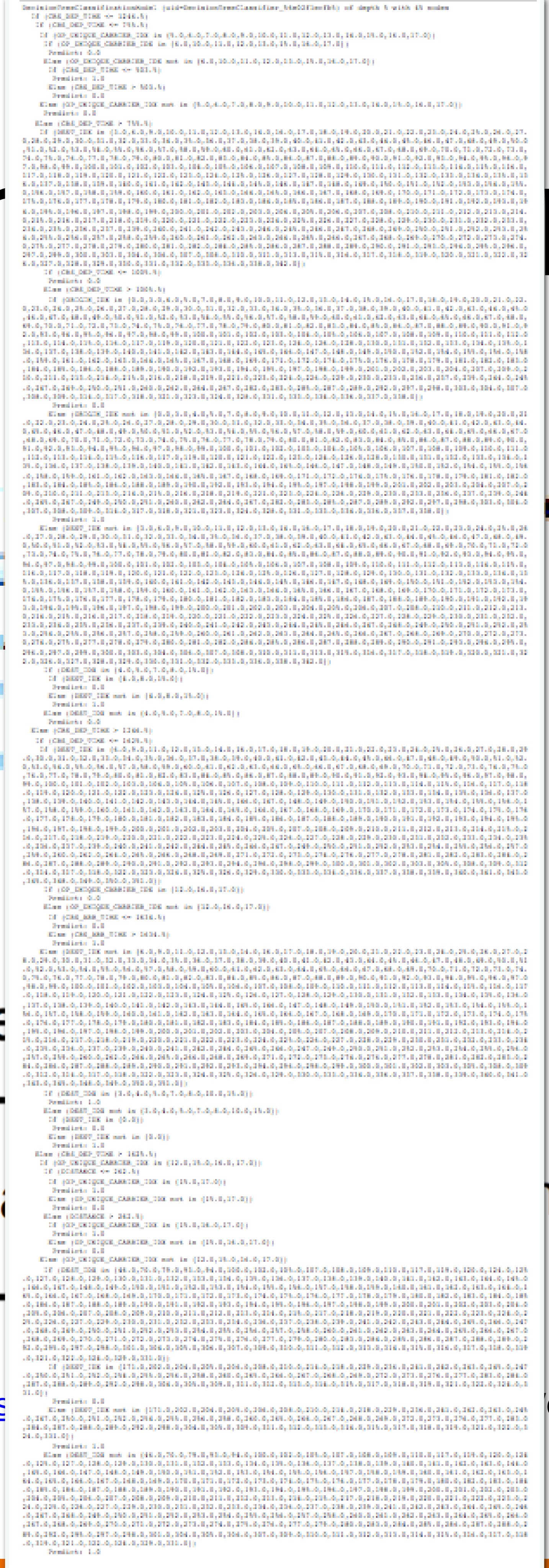
- Where deep learning has made an impact.
- The basic components of a neural network.
- Build a basic neural network from scratch
- Perform binary classification using a neural network
- Get to know the online Jupyter notebook environment.

Mach rning

```
DecisionTreeClassif                    epth 1 with 3 nodes
  If (DEPARTURE_TIM
   Predict: NOT_DEL
  Else (DEPARTURE_T
   Predict: DELAYED
```

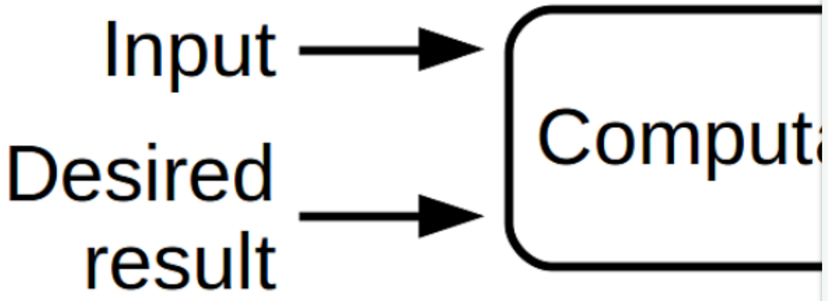
P

s



```
DecisionTreeClassificationModel (uid=DecisionTreeClassifier_[illegible]) of depth 5 with 15 nodes
 If (CRS_DEP_TIME <= 1246.5)
  If (CRS_DEP_TIME <= 755.5)
   If (OP_UNIQUE_CARRIER_IDX in {[illegible]})
    If (OP_UNIQUE_CARRIER_IDX in {[illegible]})
     Predict: 0.0
    Else (OP_UNIQUE_CARRIER_IDX not in {[illegible]})
     If (CRS_DEP_TIME <= [illegible])
      Predict: 1.0
     Else (CRS_DEP_TIME > [illegible])
      Predict: 0.0
   Else (OP_UNIQUE_CARRIER_IDX not in {[illegible]})
    Predict: 0.0
  Else (CRS_DEP_TIME > 755.5)
   If (DEST_IDX in {[illegible]})
[illegible]
```

Machine le



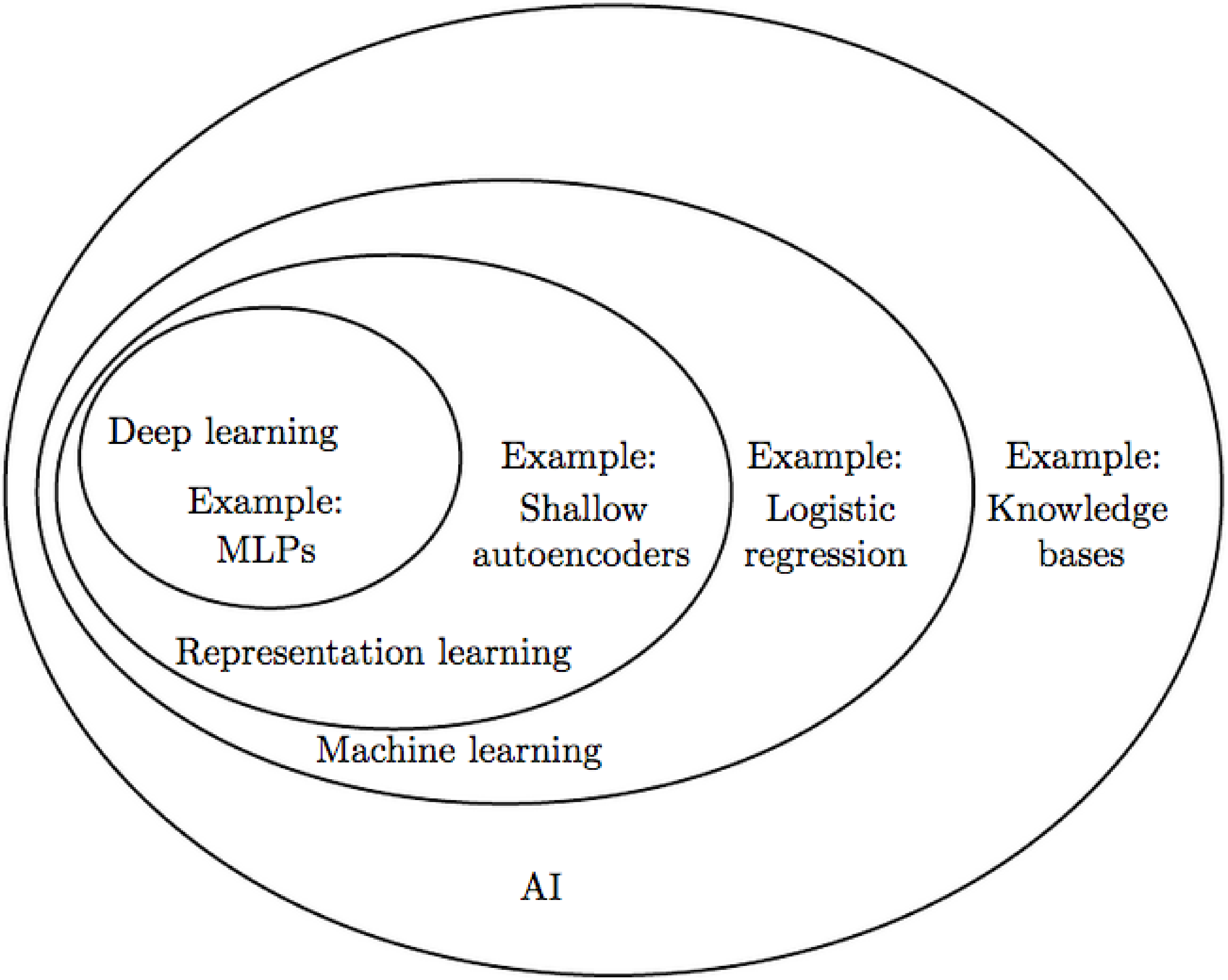
Features and result
'Decision' system
Limited human input

(delivery, full-term).

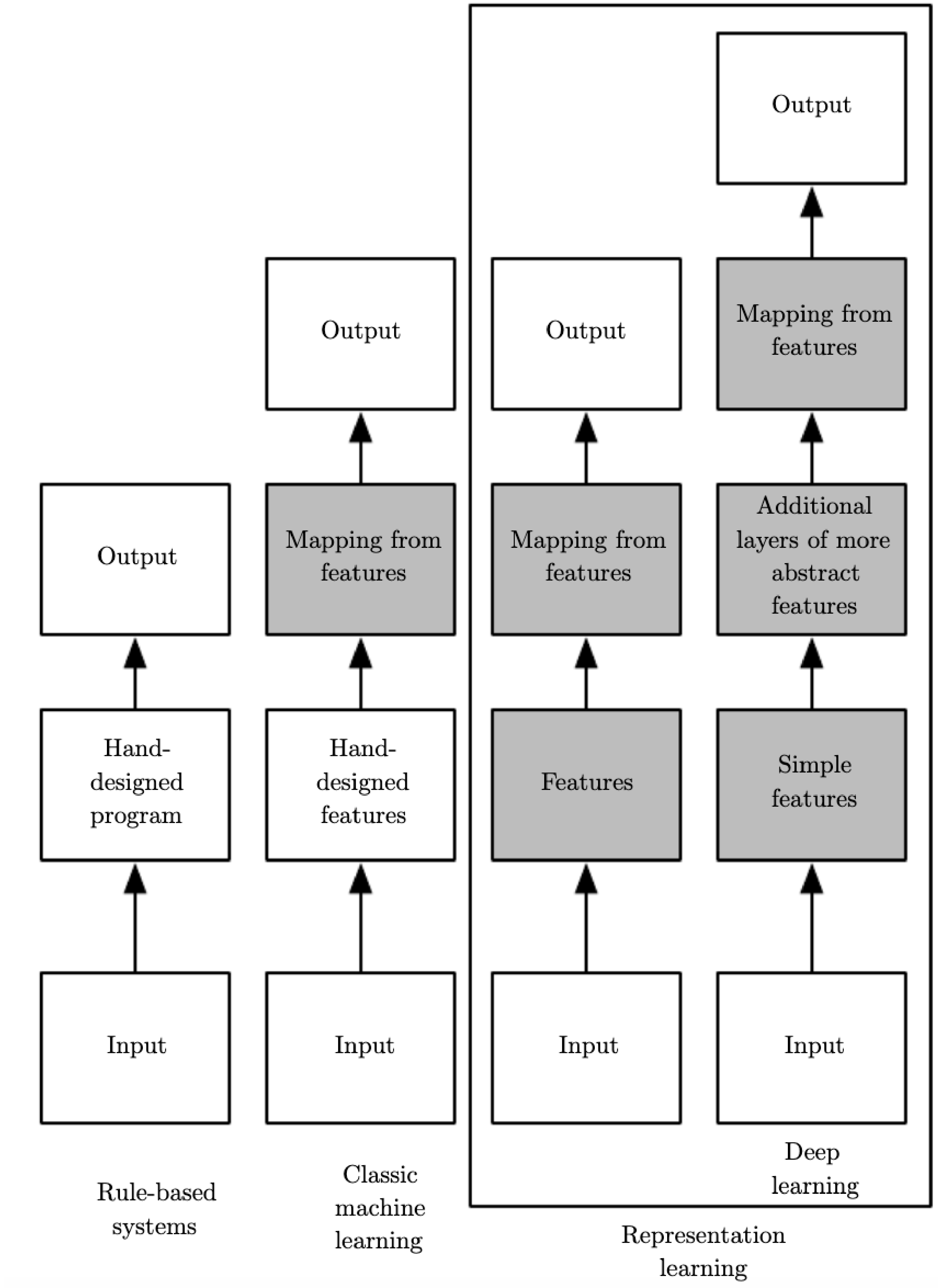
Source: Machine Learning for an Expert Sys oolery et al. (1994)

Machine learning

- Learning tasks are often broken into two broad categories, **supervised** or **unsupervised**.
- In **supervised** learning we train using the expected output.
  - **Regression**
  - **Classification**.
- In **unsupervised** learning the expected output is not provided (or does not exist) during training. Instead we rely on the *similarity* of data points.
  - **Clustering**
  - **Dimensionality reduction**
- Machine learning has many good models but most require some manual work (feature engineering) to perform.

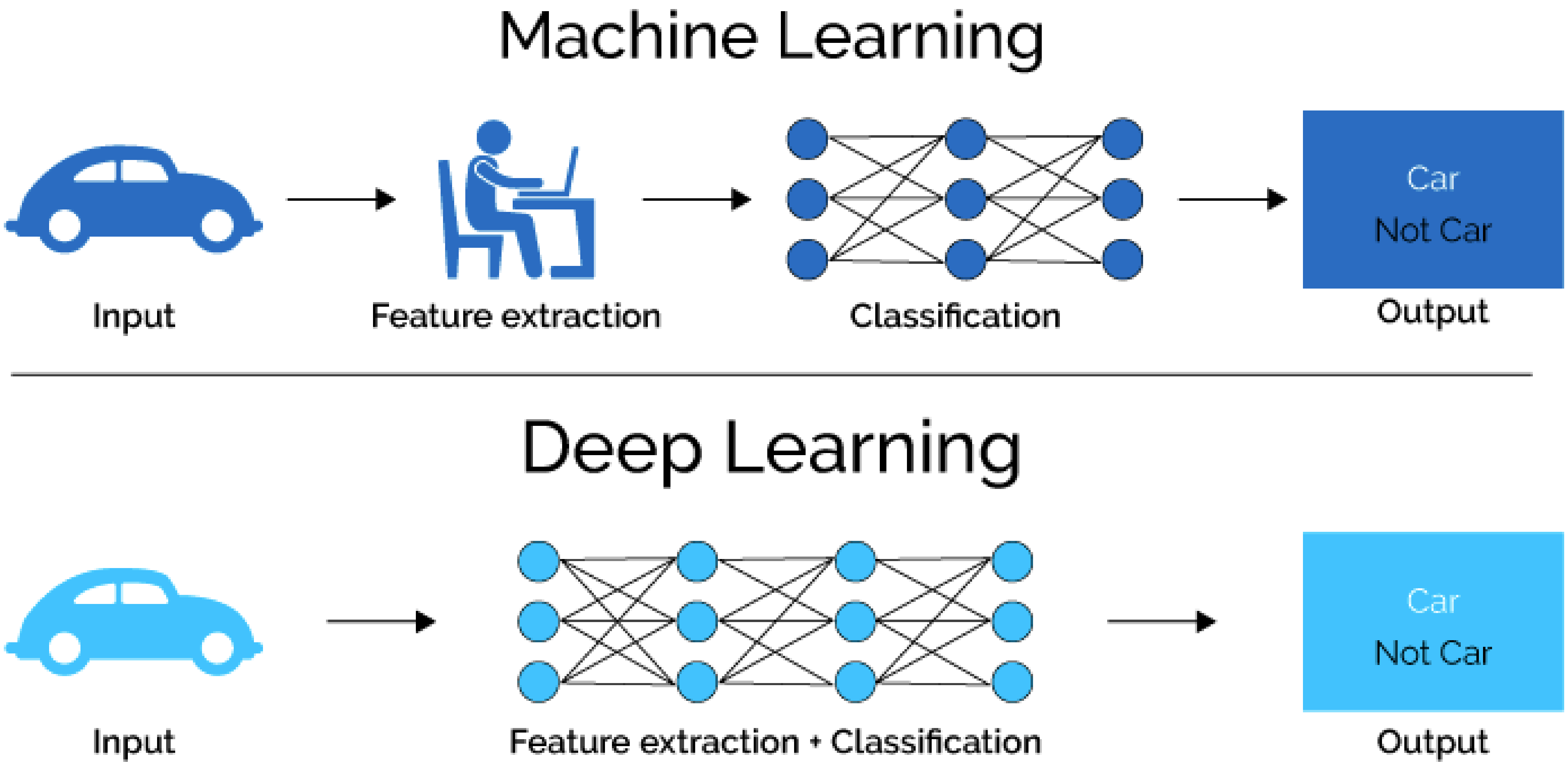


Source: Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville



Source: Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville

What is deep learning?

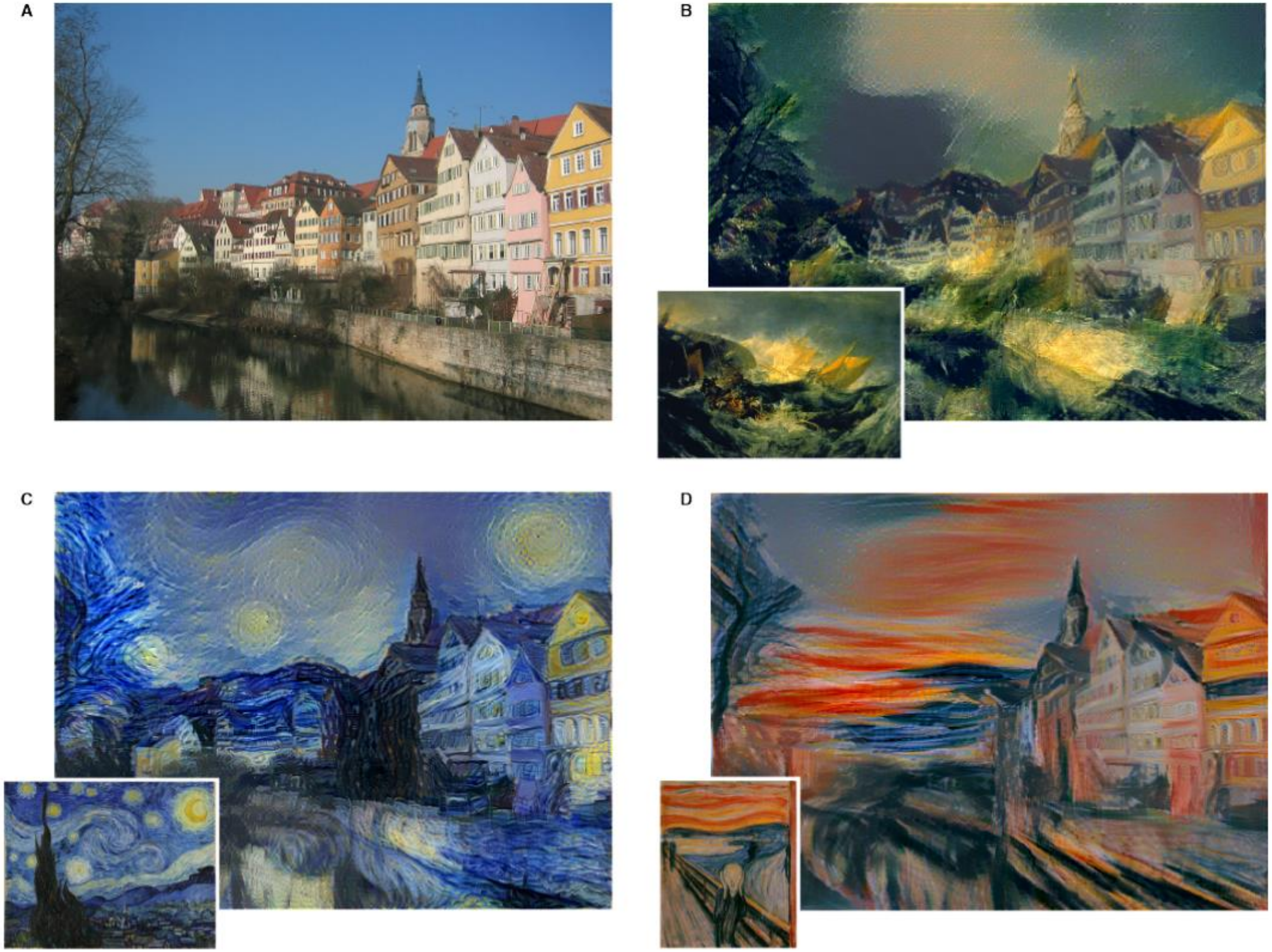


Source: Alex, D., 2017: Difference between Machine Learning and Deep Learning.

Success of deep learning

- Machine translation
- Speech recognition
- Language recognition based on characters
- Image classification
- Image screen caption generation
- Recommender systems
- Drug design
- ...

Success of deep learning



Style transfer
Source: https://ai.googleblog.com/2016/10/supercharging-style-transfer.html

Success of deep learning

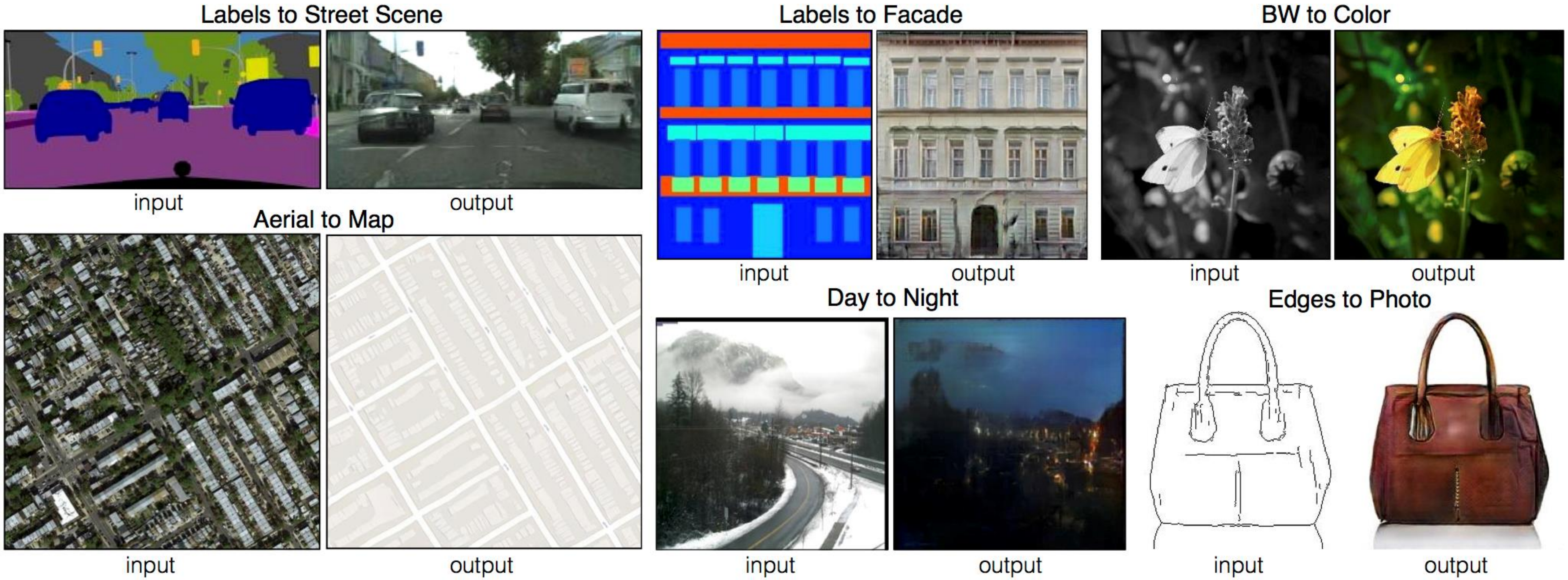
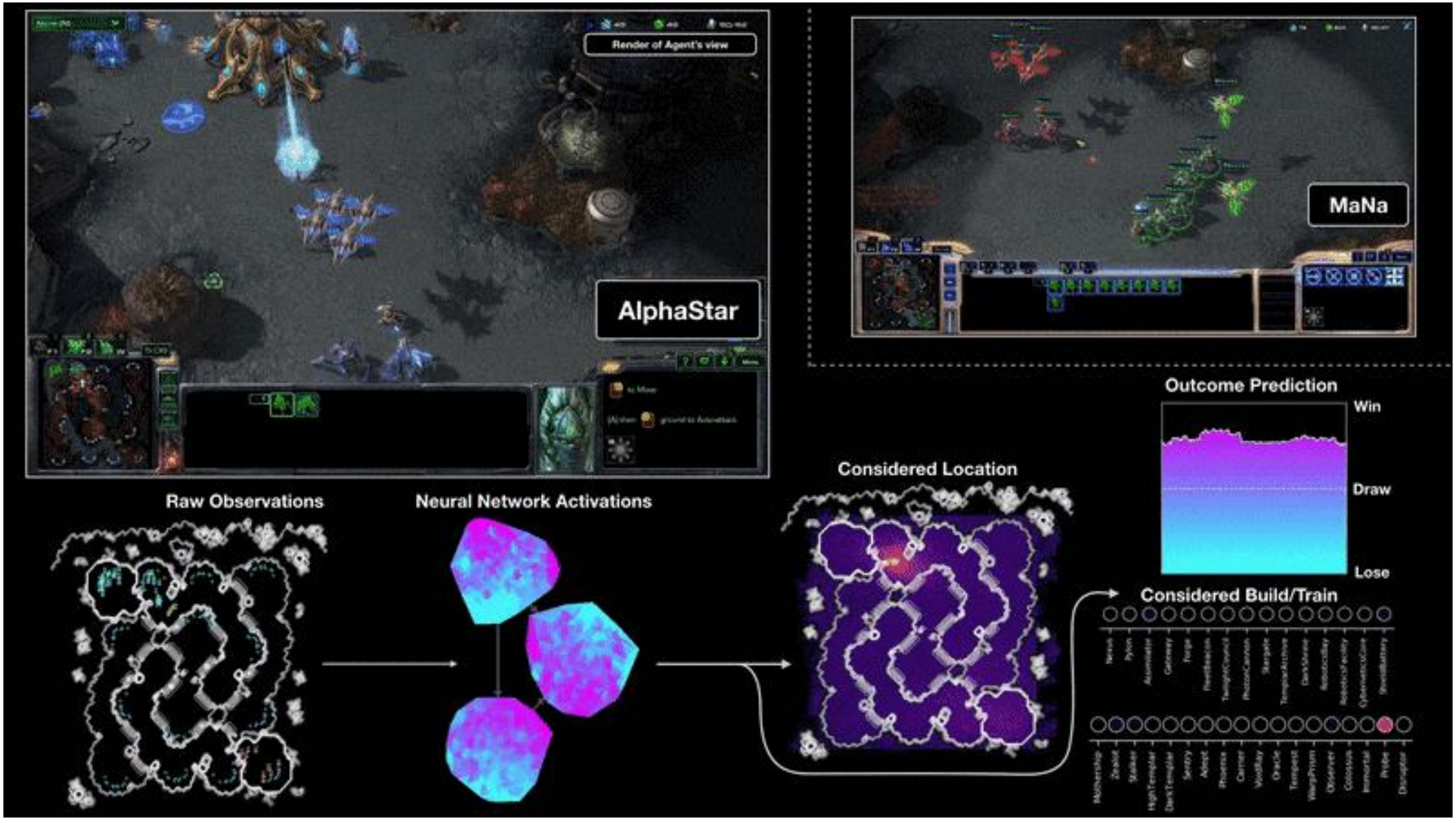


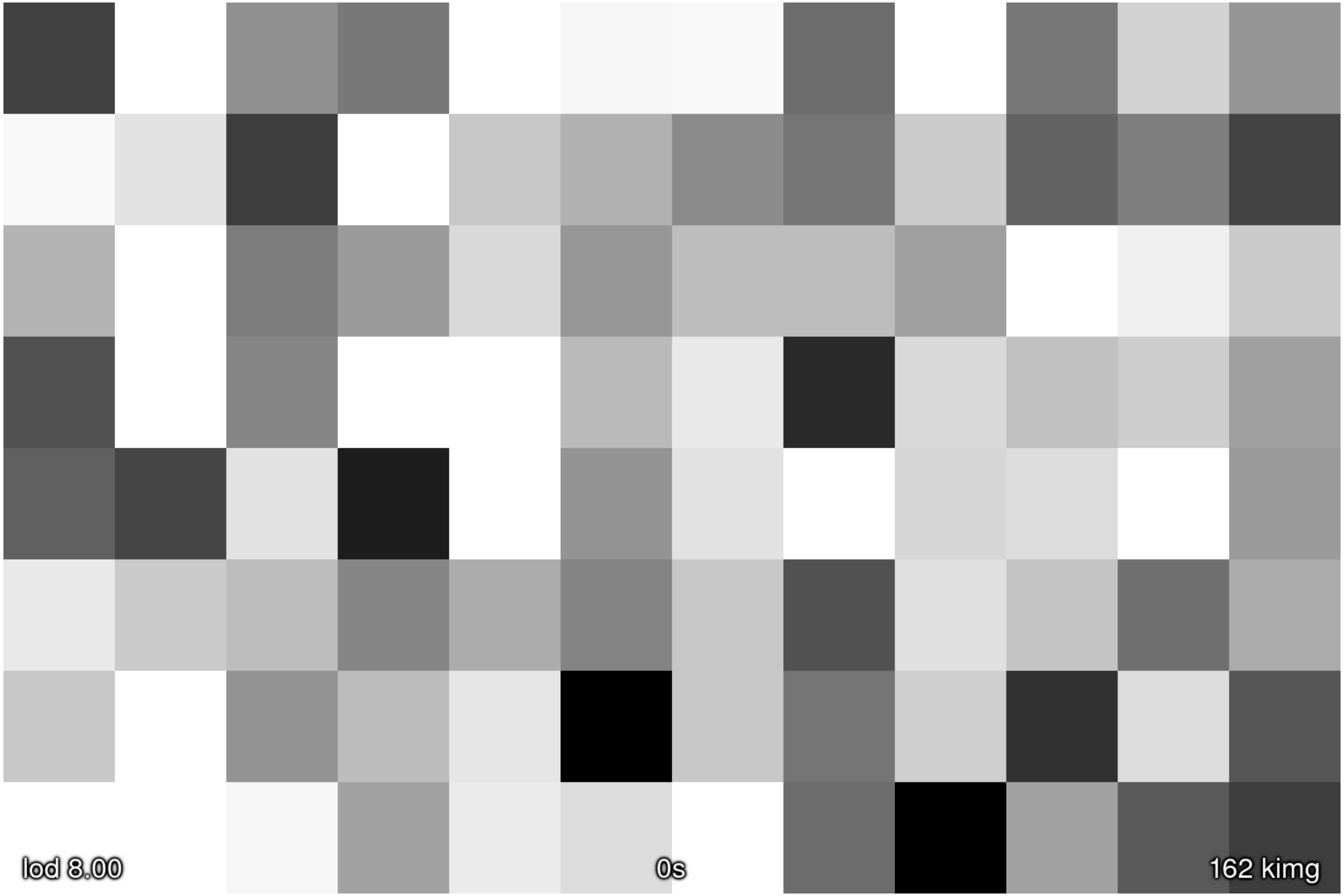
Image to image
Source: https://phillipi.github.io/pix2pix/

Success of deep learning



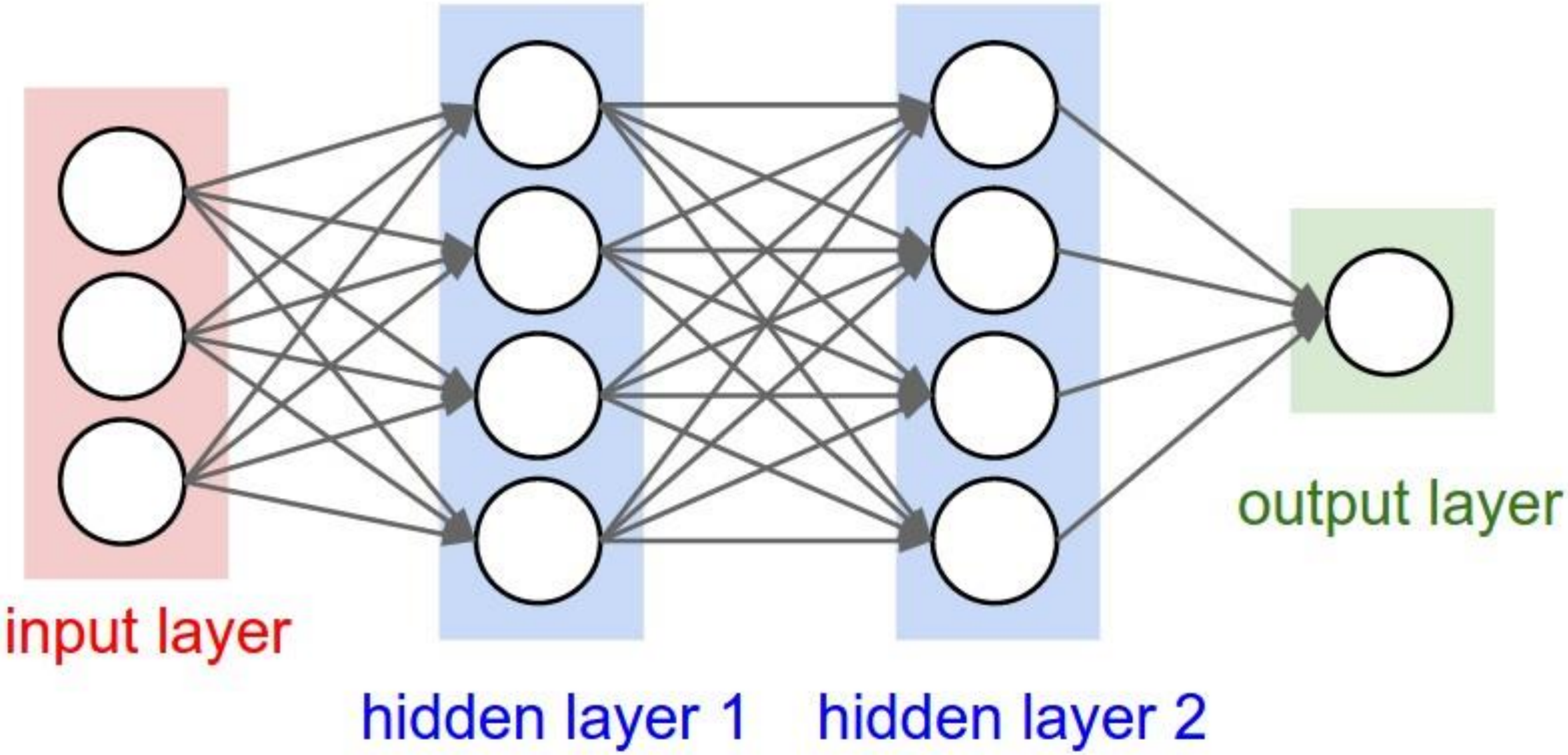
Deep reinforcement learning
Source: https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/#gif-264

Success of deep learning

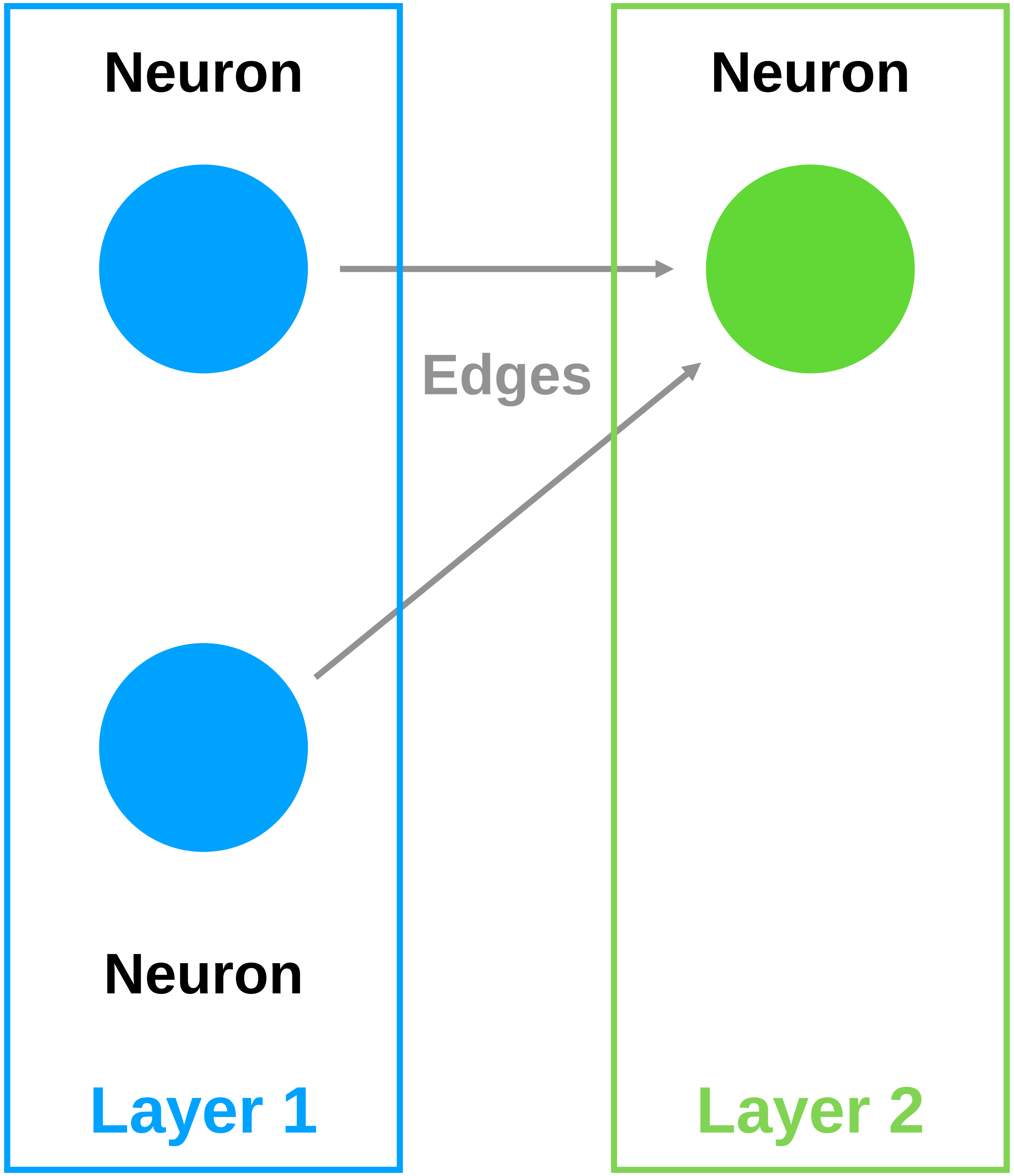


Deep learning

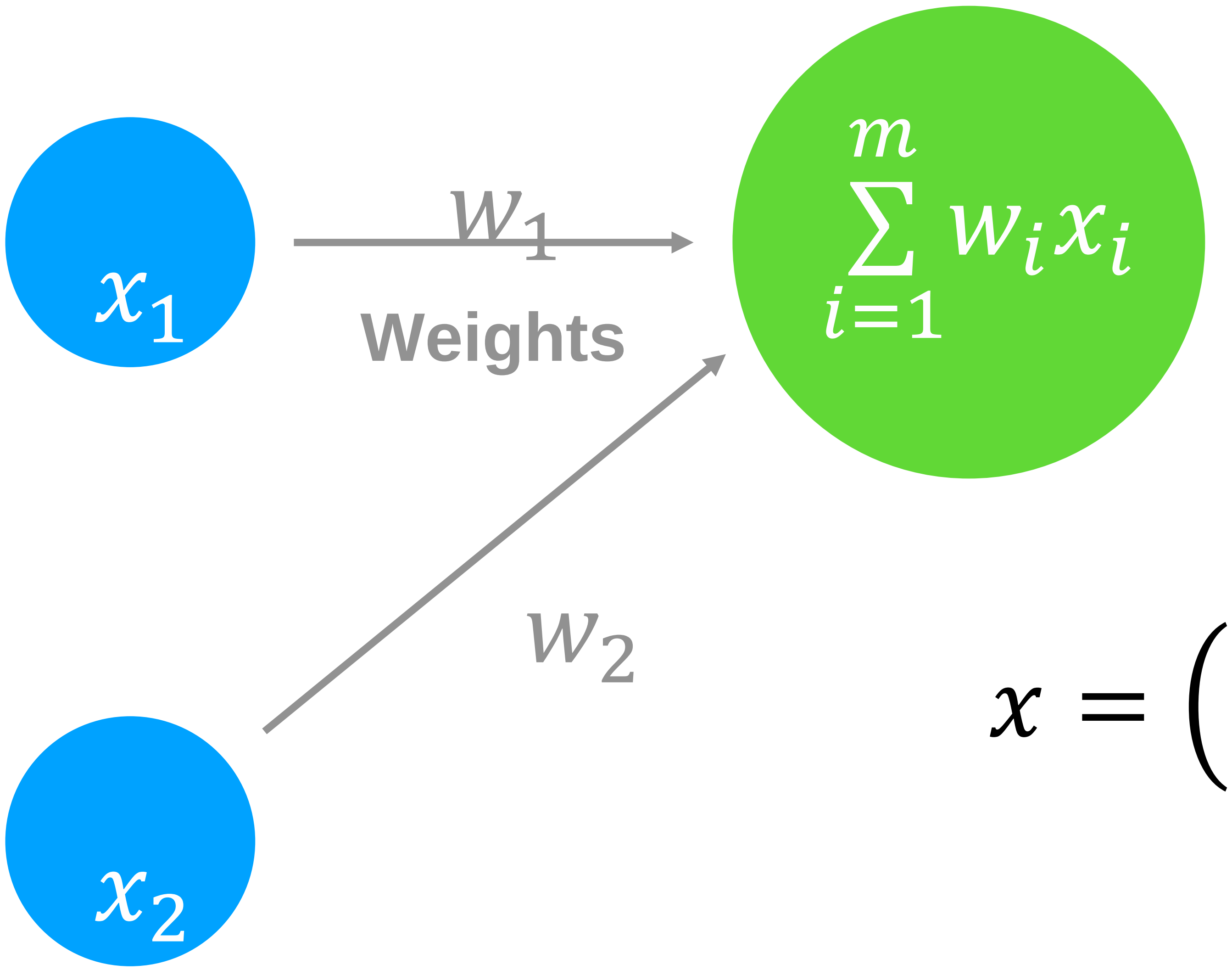
Deep learning is based on neural networks



What is a neural network?



A single neuron

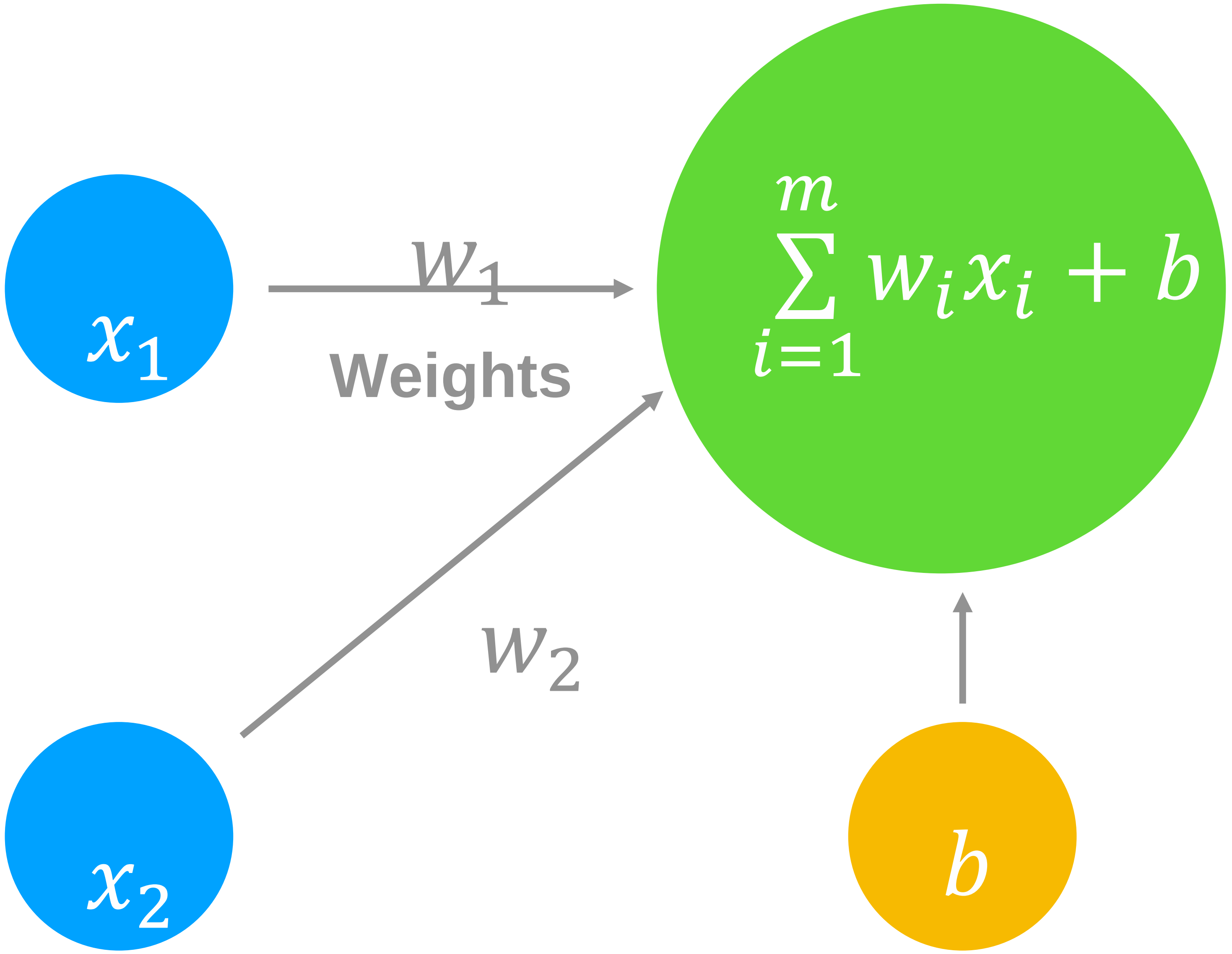


$$w_1 x_1 + w_2 x_2$$

$$x = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad w = \begin{pmatrix} 0.5 \\ 1 \end{pmatrix}$$

$$\mathbf{w}^{\mathsf{T}}\mathbf{x} = \begin{pmatrix} 0.5 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \end{pmatrix} = 2.5$$

A single neuron

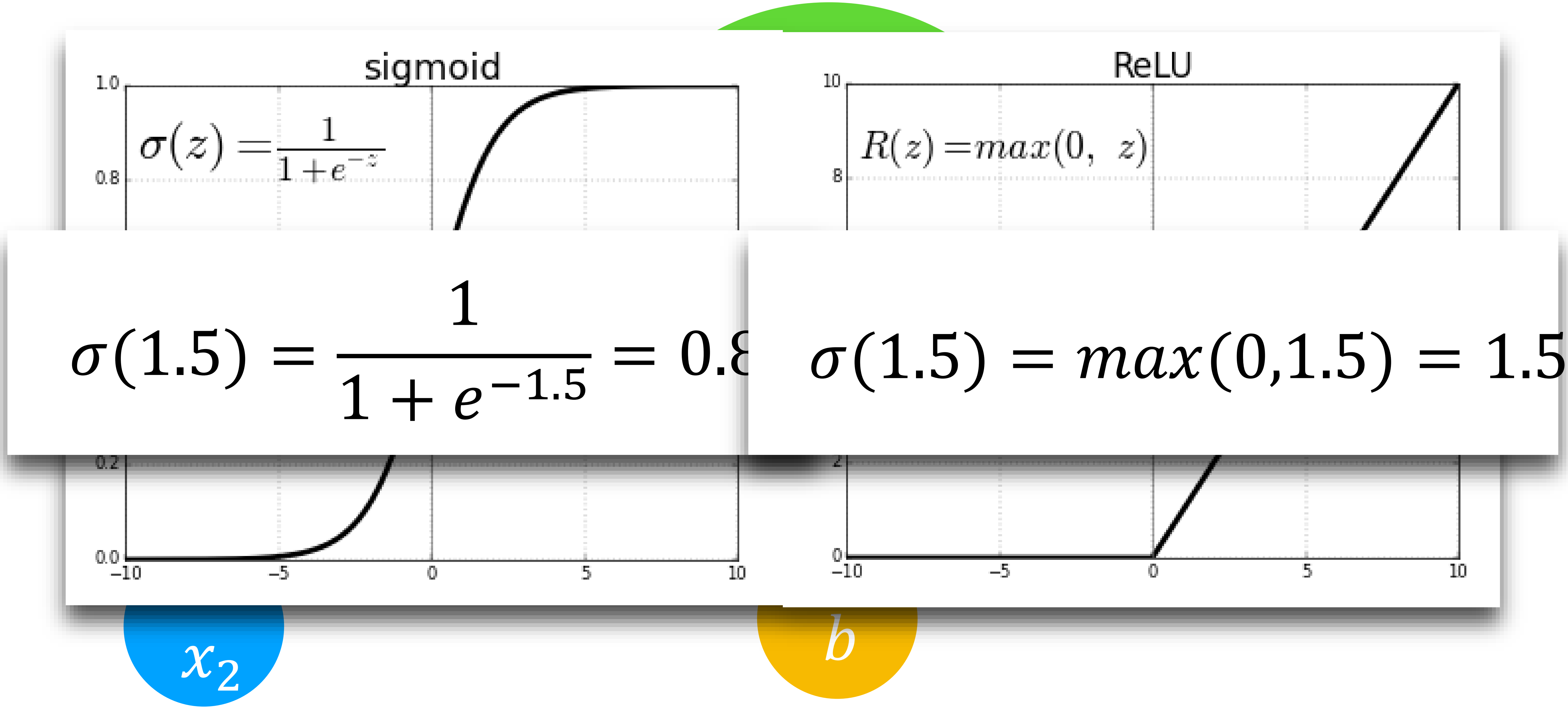


$$w_1 x_1 + w_2 x_2 + b$$

$$b = -1$$

$$\mathbf{w}^\intercal \mathbf{x} + b = (0.5 \quad 1)\binom{1}{2} - 1 = 1.5$$

A single neuron

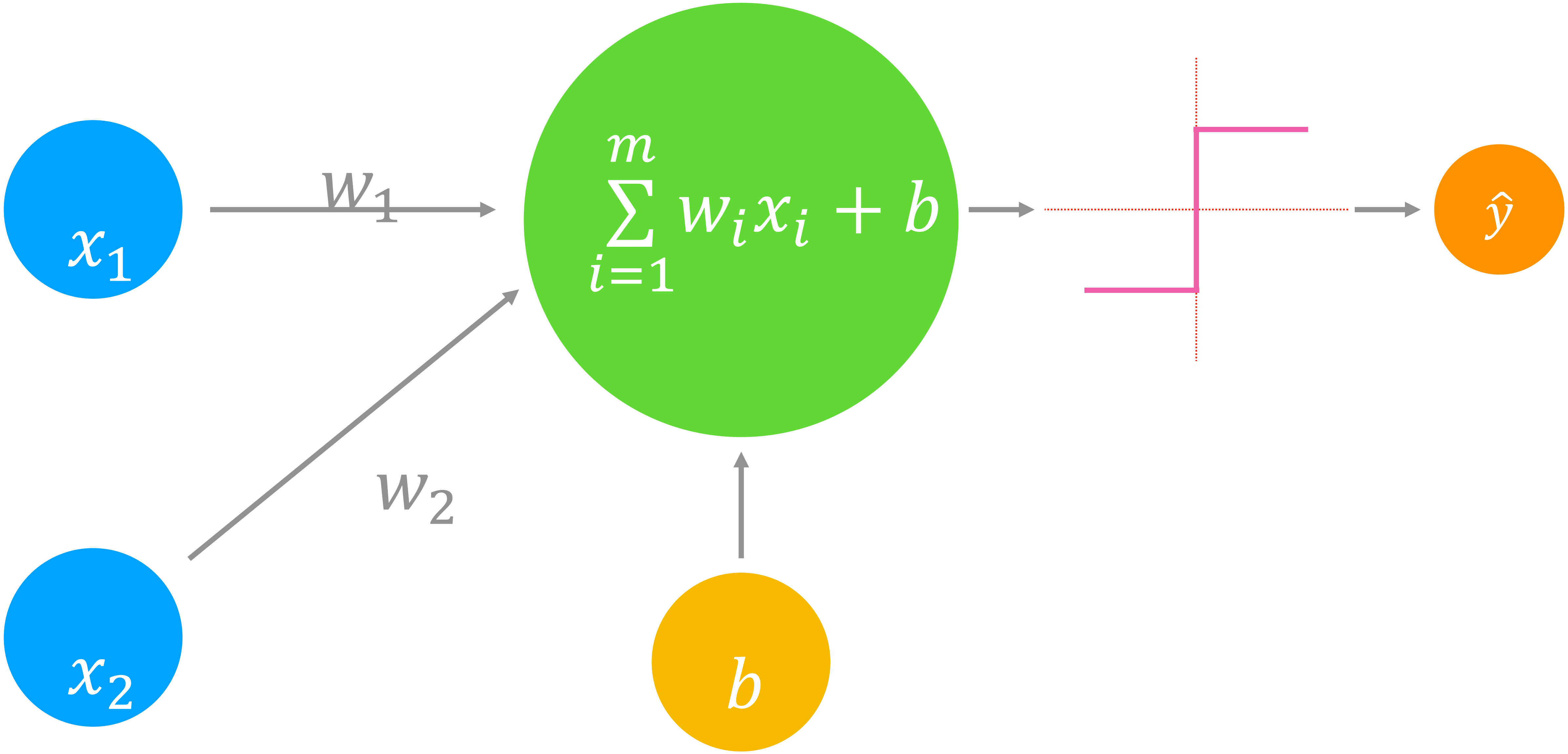


$$\sigma(1.5) = \frac{1}{1+e^{-1.5}} = 0.8$$

$$\sigma(1.5) = max(0{,}1.5) = 1.5$$

Source: Activation Functions in Neural Networks

Prediction



inputs weights summation + bias activation output

Before we begin

- Refresh some R
- Tensor manipulation
- Implement a neuron from scratch

Hands-on

- Exercises in Jupyter notebooks
- Notebooks run on Lisa, the national compute cluster
    - Lisa has approximately 300 nodes, 4800 cores
    - During course hours: 16 full nodes reserved
    - Outside course hours: 1 node / capacity for 8 notebooks
- First users of Jupyter @ Lisa!

Hands-on

- Notebook session will start when you login (may take up to a minute or so).
- Closing the browser does *not* shut down the notebook session.
- To manually start / shut down a notebook session:
  - ‘Home’ (top left) => Start / Stop My Server.
  - Please shut down sessions when working outside of course hours, since capacity is limited!

Hands-on

First time: login & change password @ https://portal.surfsara.nl

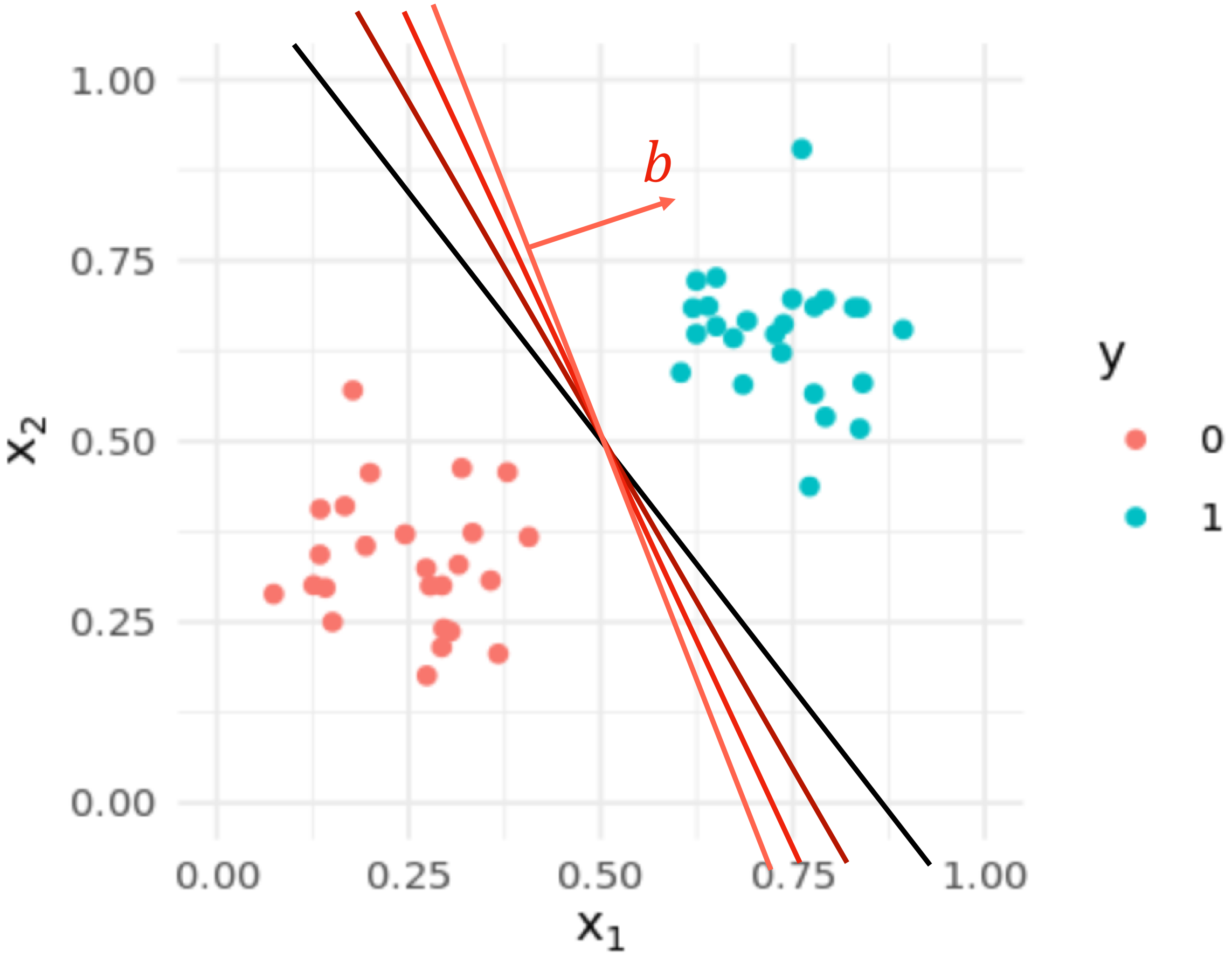
Go to https://https://jupyter.lisa.surfsara.nl:8000/

Notebook: 01a-neuron-from-scratch.ipynb

14:30-15:30

Notebook recap

- Data preparation / restructuring is necessary
- Weights change the 'slope'
- Bias shifts decision boundary



Today's program

14:00-14:30 What is ML / DL? What is a neuron?

14:30-15:30 Hands on: building a neuron from scratch

15:30-15:45 How do neural networks work?

15:45-16:15 Break

16:15-16:30 How do neural networks work? (continued)

16:30-17:15 Hands on: building a network from scratch

17:15-18:00 Loss function & updating weights

18:00-19:00 Diner

19:00-19:45 Hands on: the XOR problem

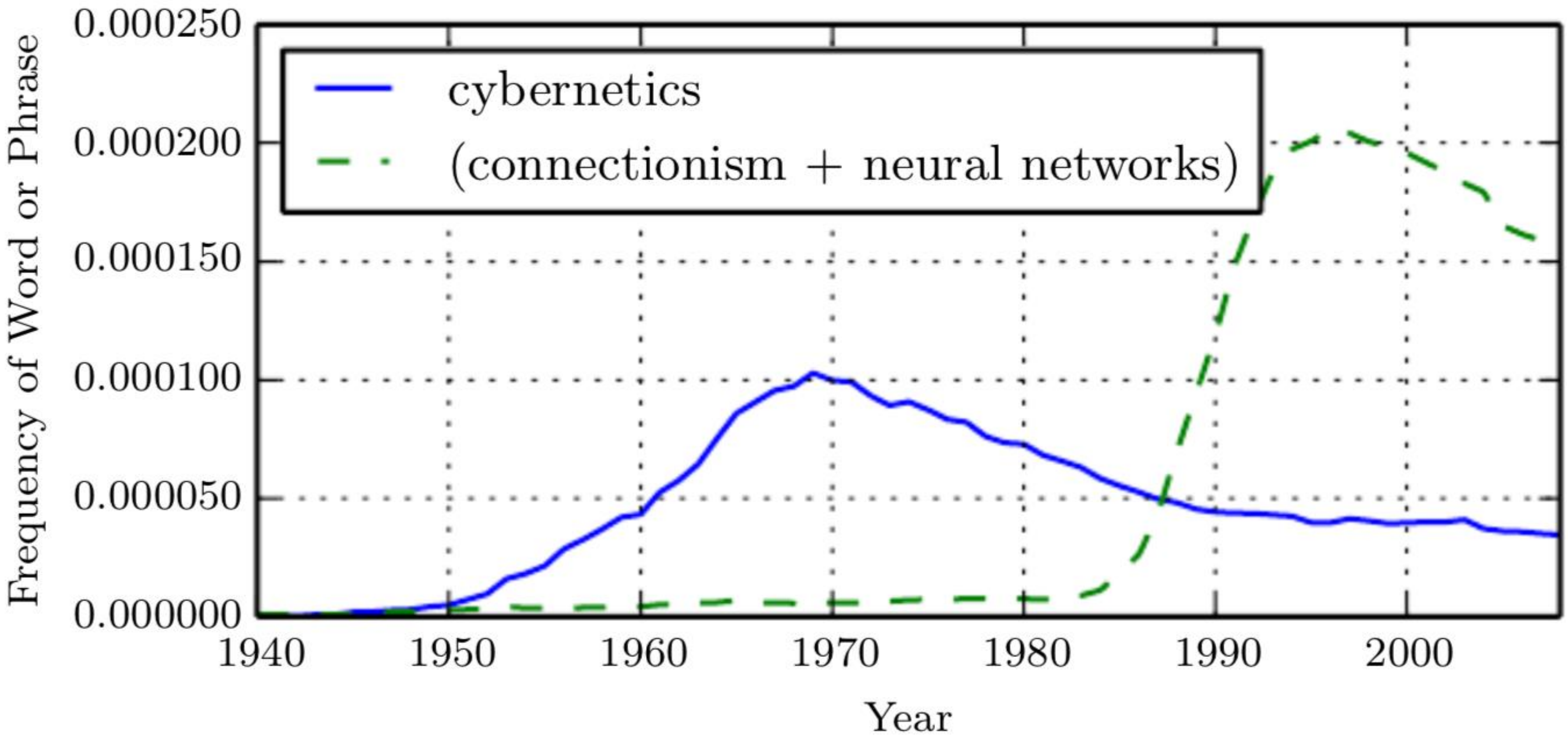
19:45-20:00 Dataset splitting & Performance evaluation

20:00-21:00 Hands on: Keras on fashion Mnist

If we finish early: Machine learning tasks (classification vs regression)

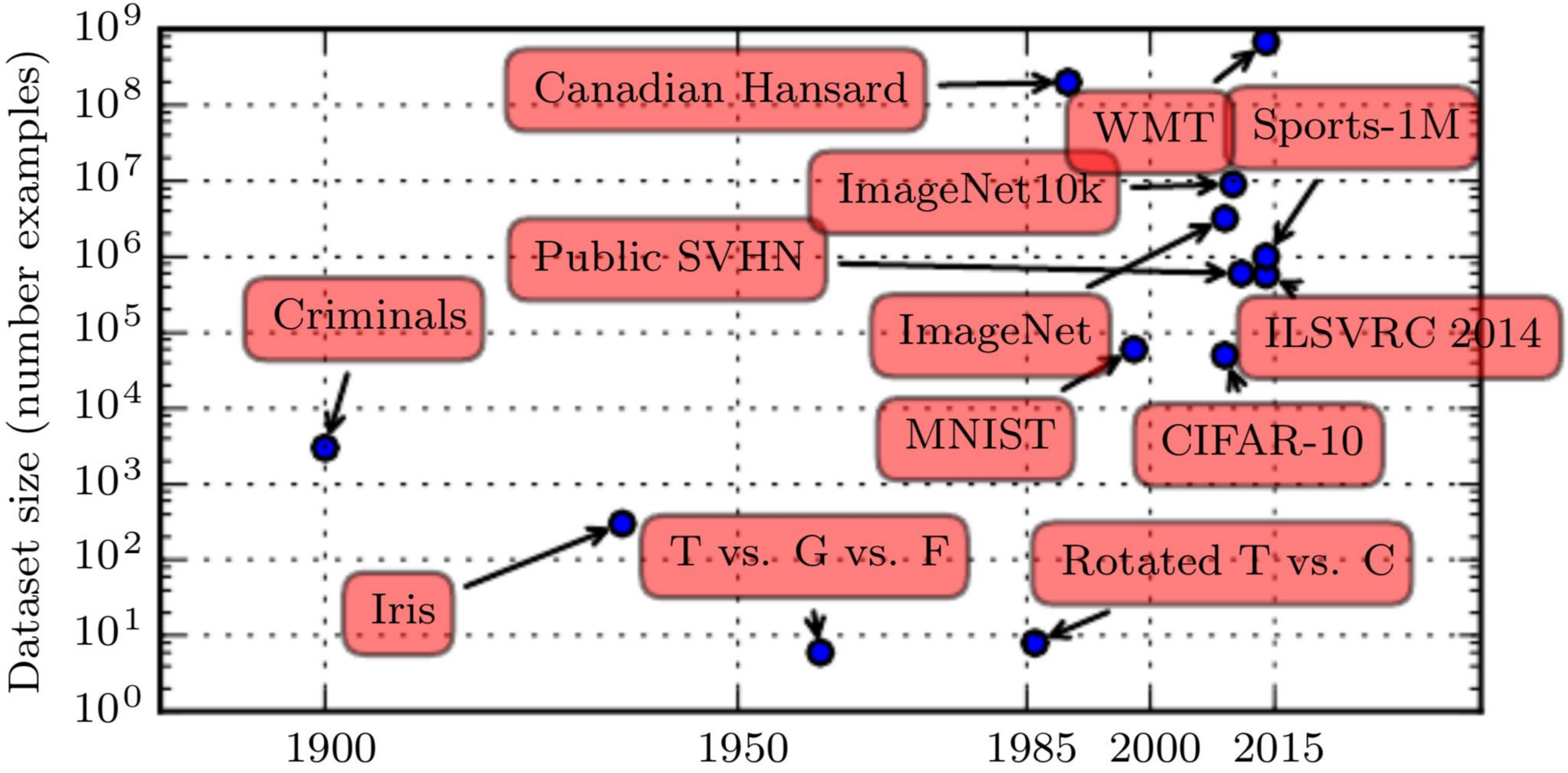
Why the success?

No single reason, much of the theory is old



Why the success?

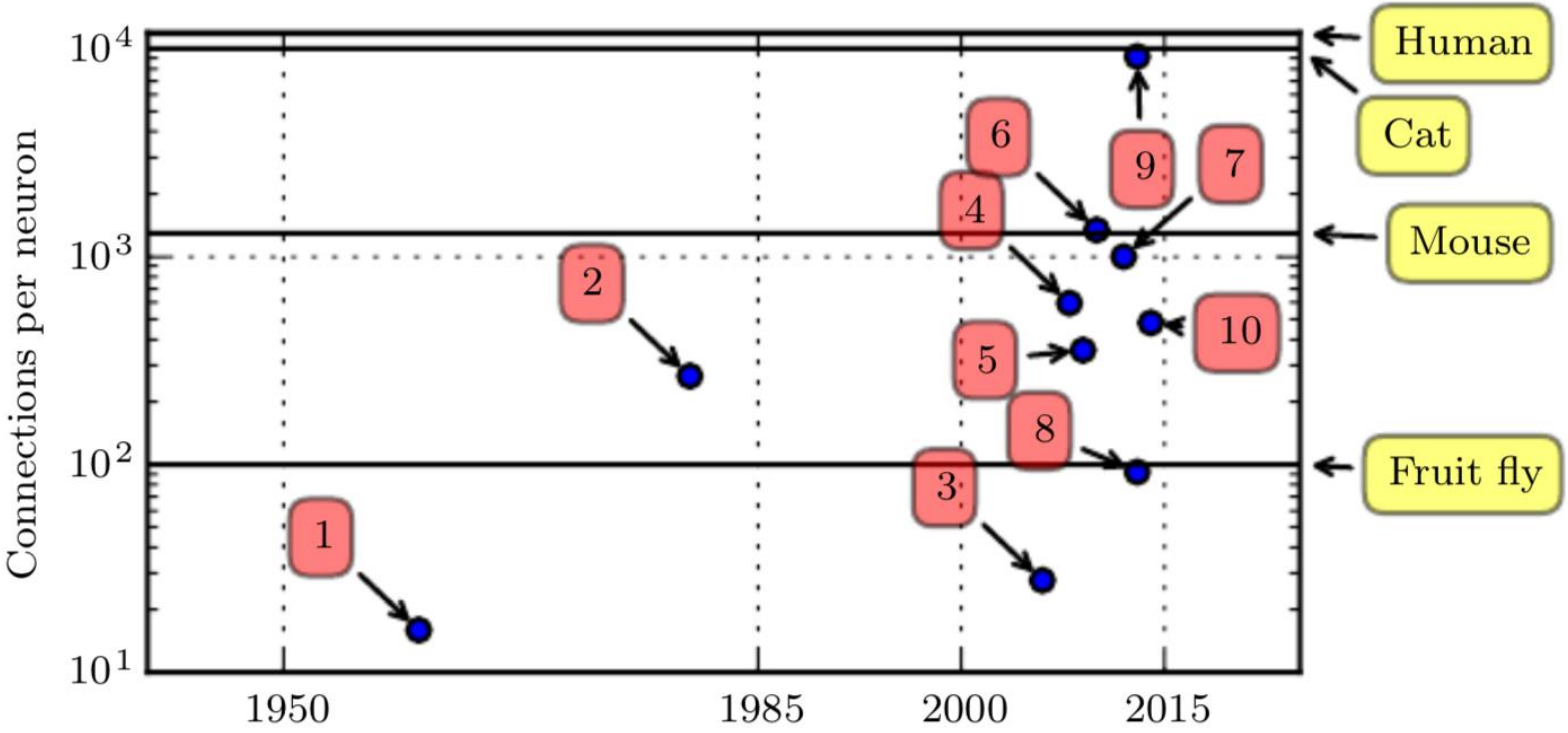
More data available



Source: Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville

Why the success?

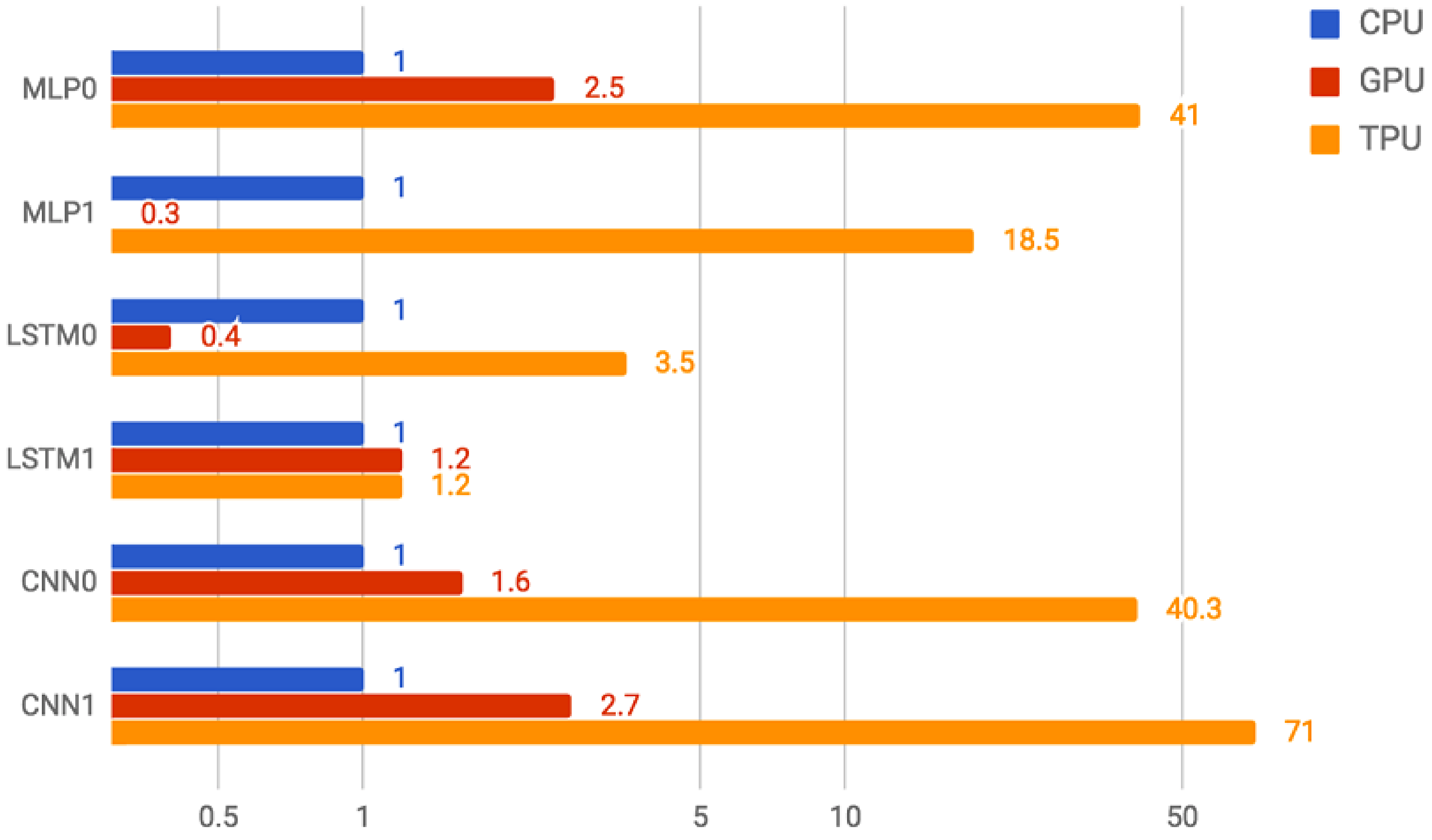
Bigger models



Source: Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville

Why the success?

More compute power

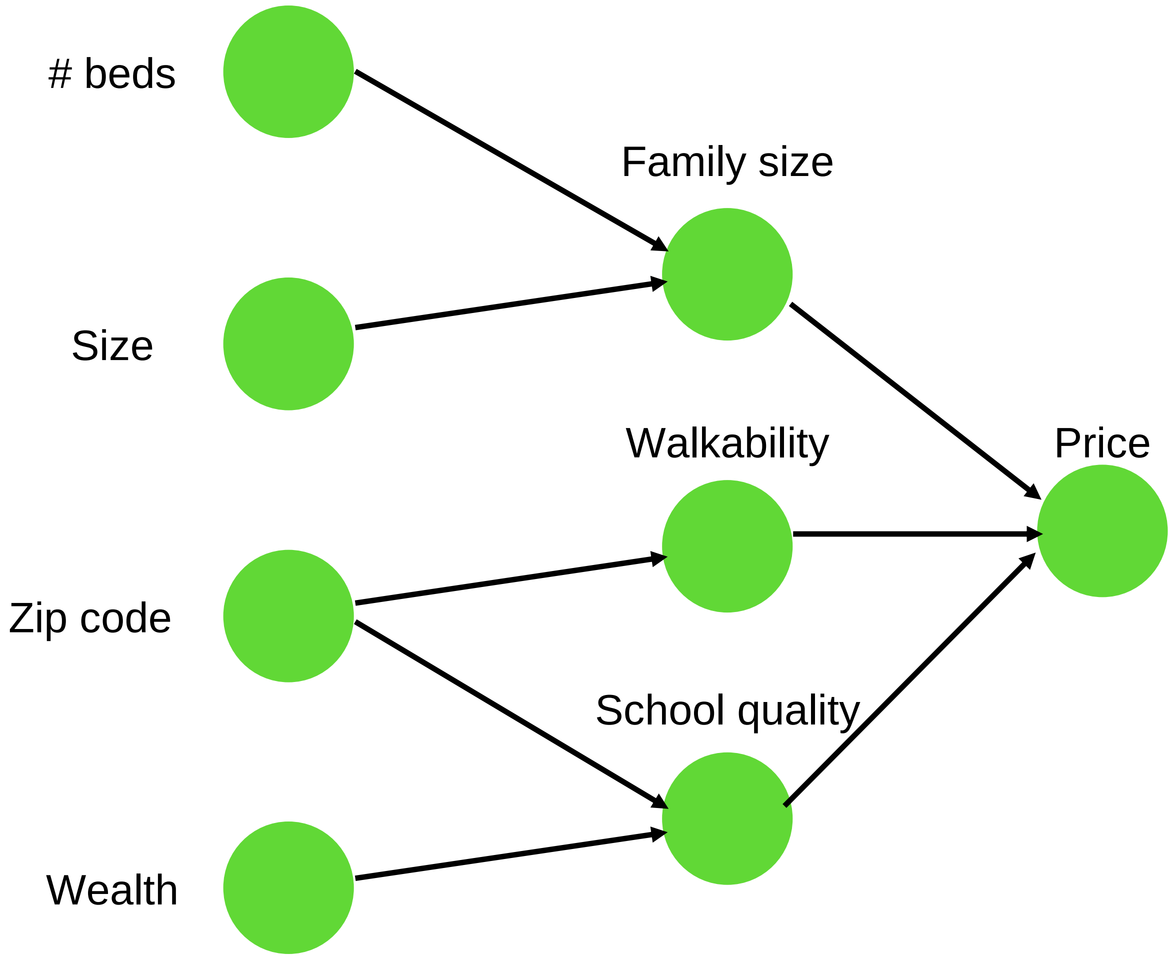


CPU vs GPU vs TPU performance comparison on six reference workloads log scale)

Source: https://www.androidauthority.com/google-powering-ai-816146/

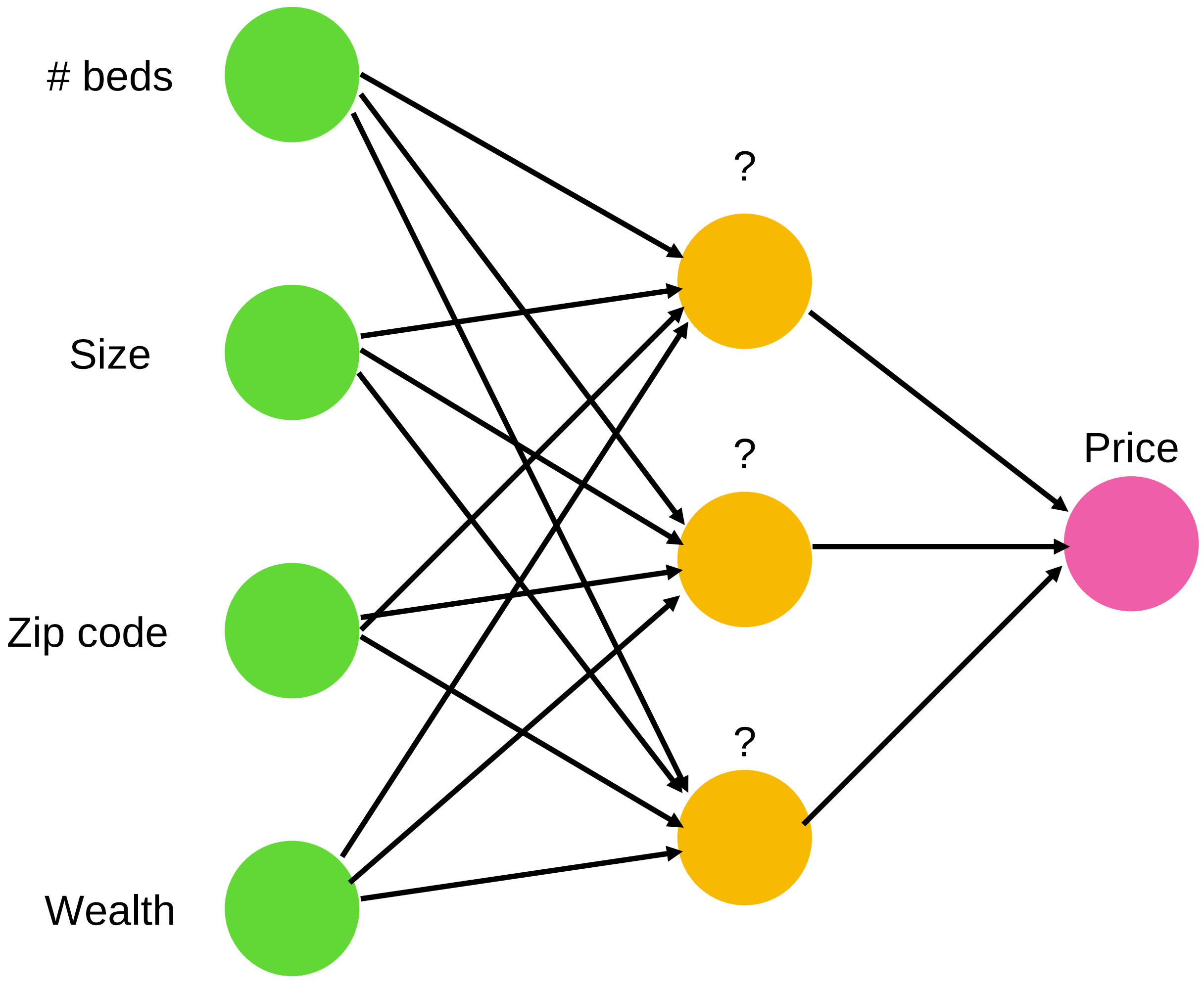
How do NNs work?

- We want to make a model to predict housing prices

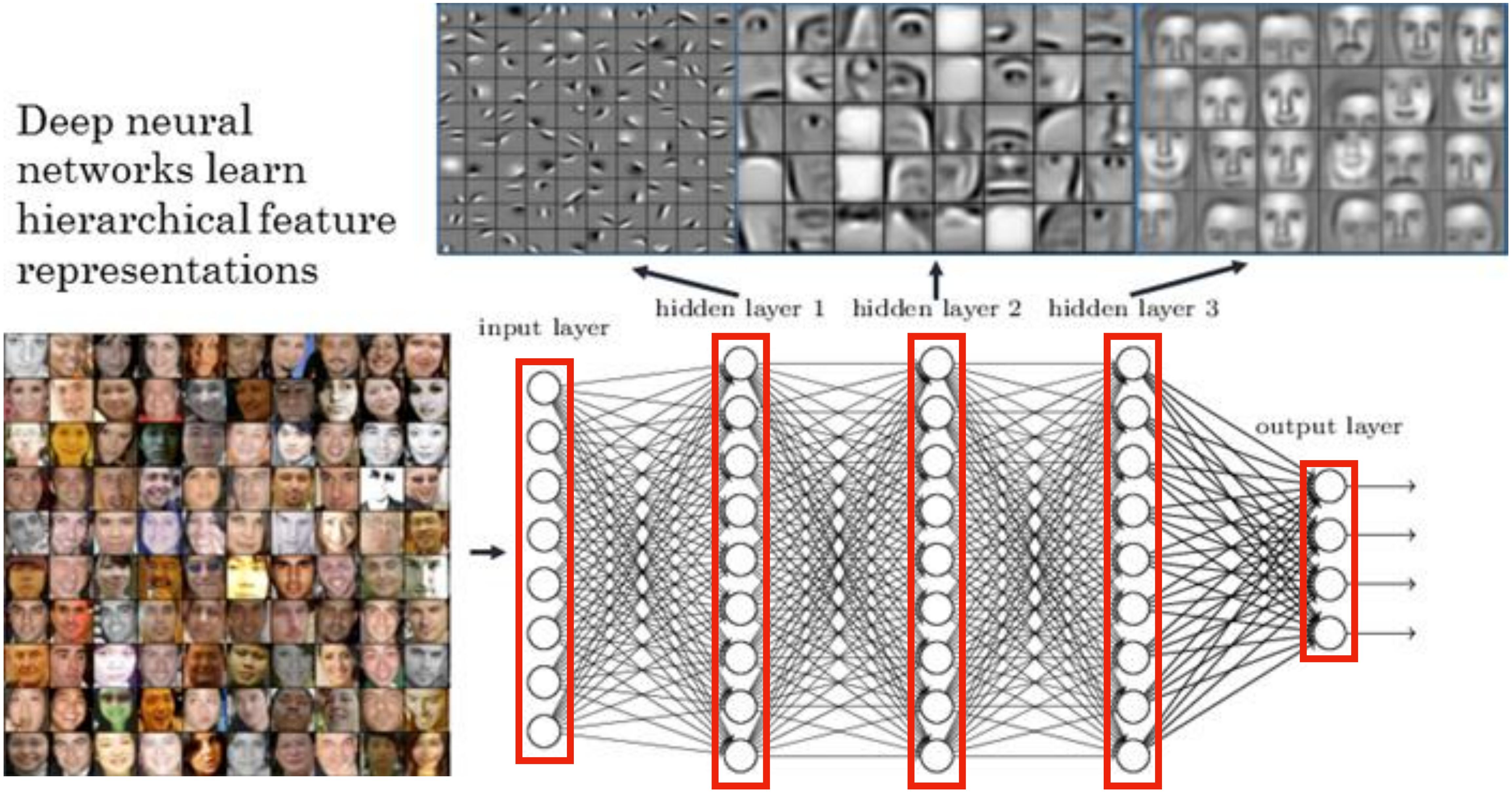


Hidden layers

- Or just hit it with a hammer and let the model learn the features



How does it work?



Deep neural networks learn hierarchical feature representations

Layers

Source: A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning

Hidden layers

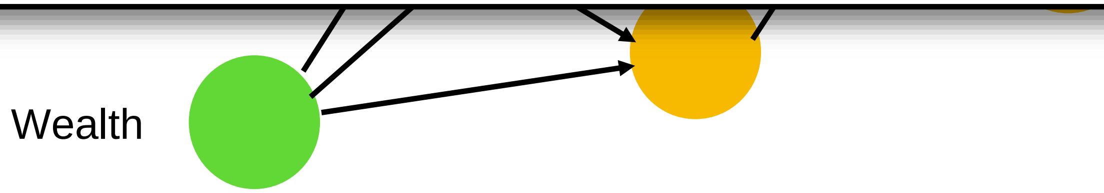
- We can
- Interpret

Universal approximation theorem

"Given a neural network with a single hidden layer of sufficient size, the network can model can approximate any function in $\mathbb{R}^p$ "

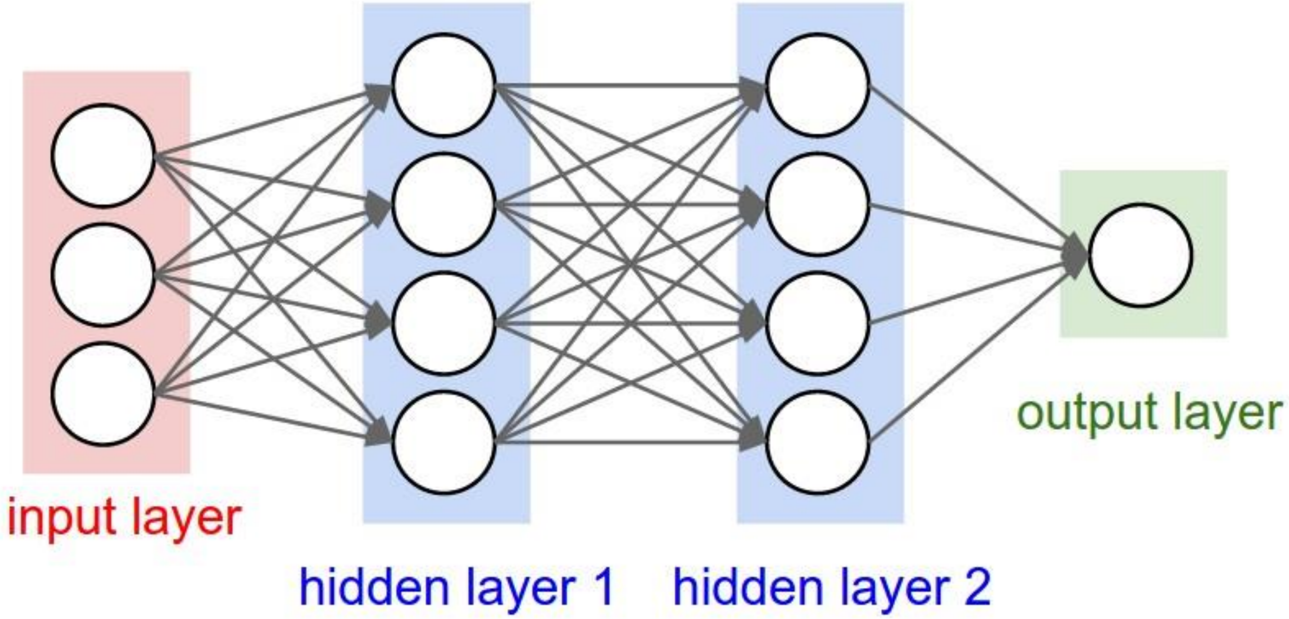
In other words:

- There is a 'true' function f relating the inputs X to outputs Y
- NN can approximate this function to arbitrary precision given sufficient layer size
- Required layer size can be extremely large and grow rapidly (e.g. exponential) with the dimensionality of the problem (p)



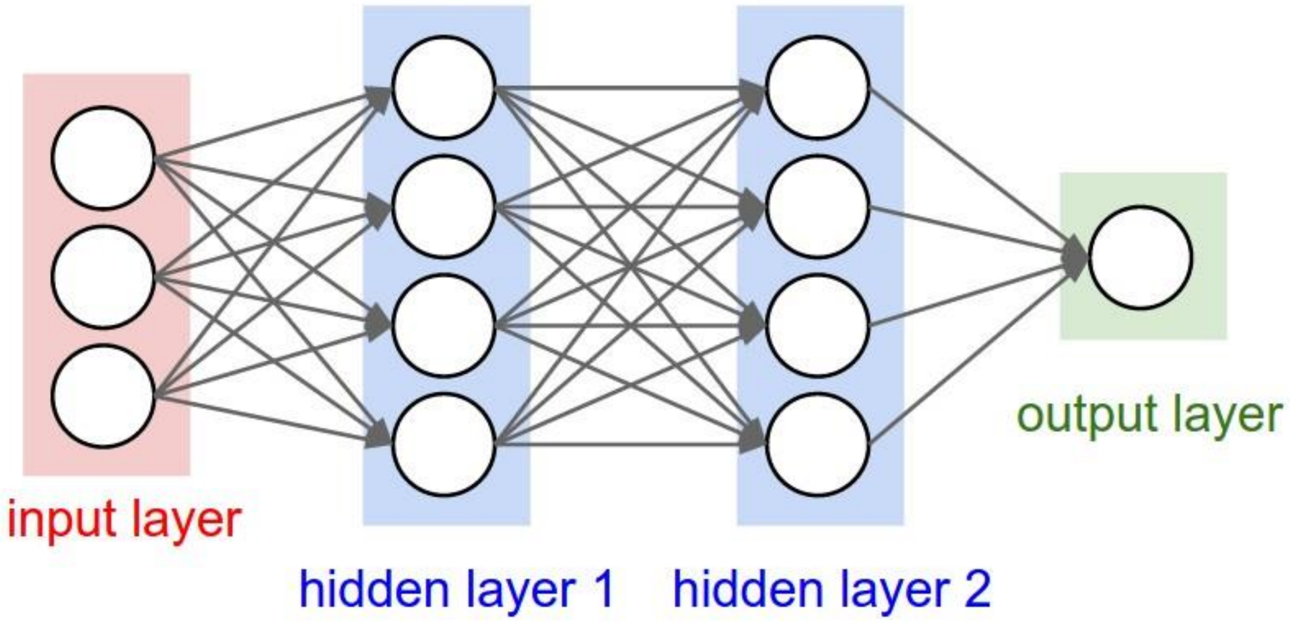
The components

- Neural networks are made from **neurons** and **edges**.
- A collection of neurons is a **layer**.
- The output of a previous layer is used as an input to the next layer.
- The input layer is x, the raw data, and the output layer is $\hat{y}$, the prediction.
- Anything in between is "**hidden**".
- Layers are usually represented as **vectors**.
- Edges are usually represented as **matrices - the weights**.
- The **edges** are **parameters** to be trained.

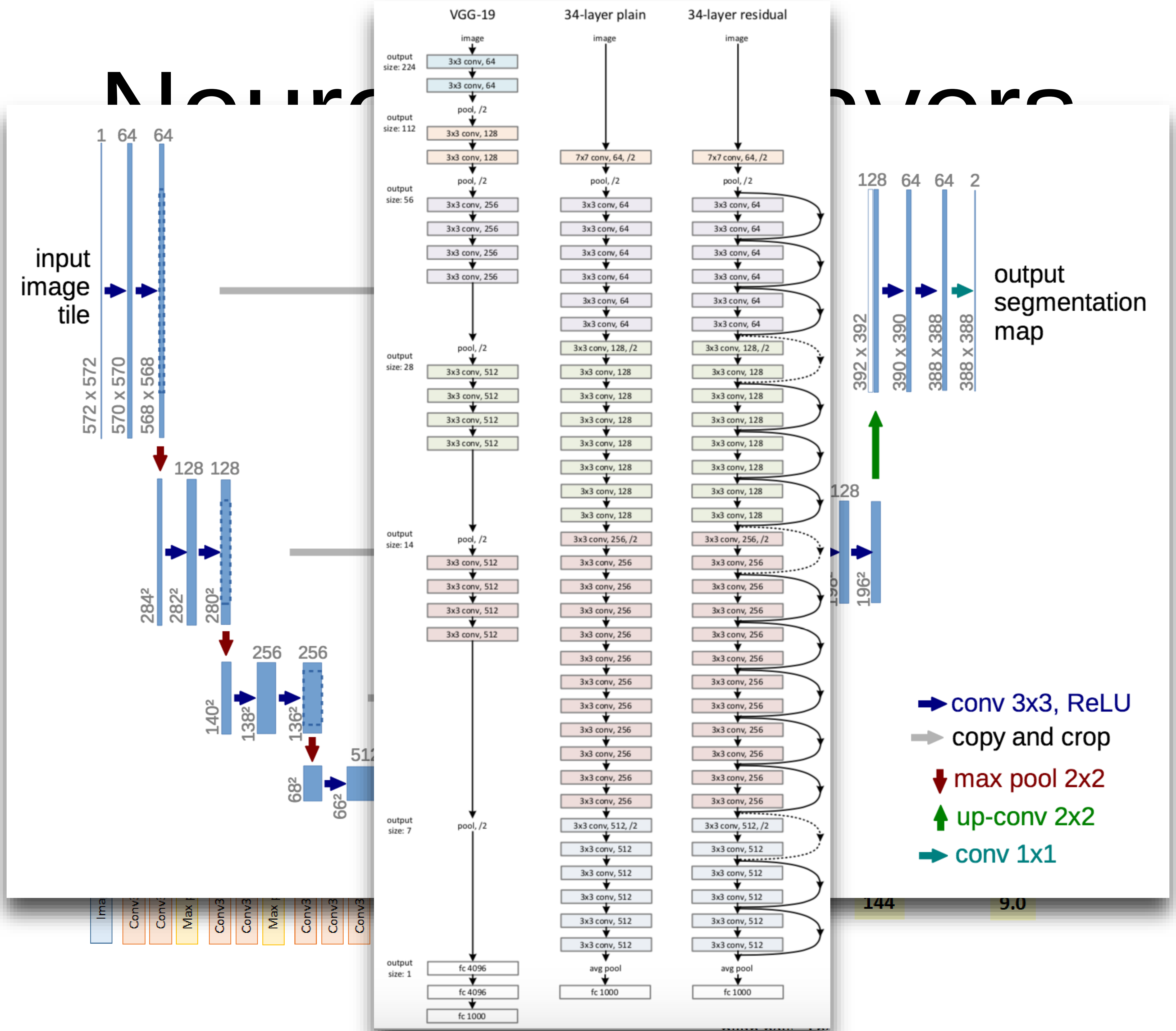


The components

- There are many **types** of layers, the previous example is a **dense** layer: all neurons in one layer are connected to all neurons in the next layer.
- Piling many dense layers, creates a Multi Layer Perceptron (MLP) and is very powerful, but expensive.

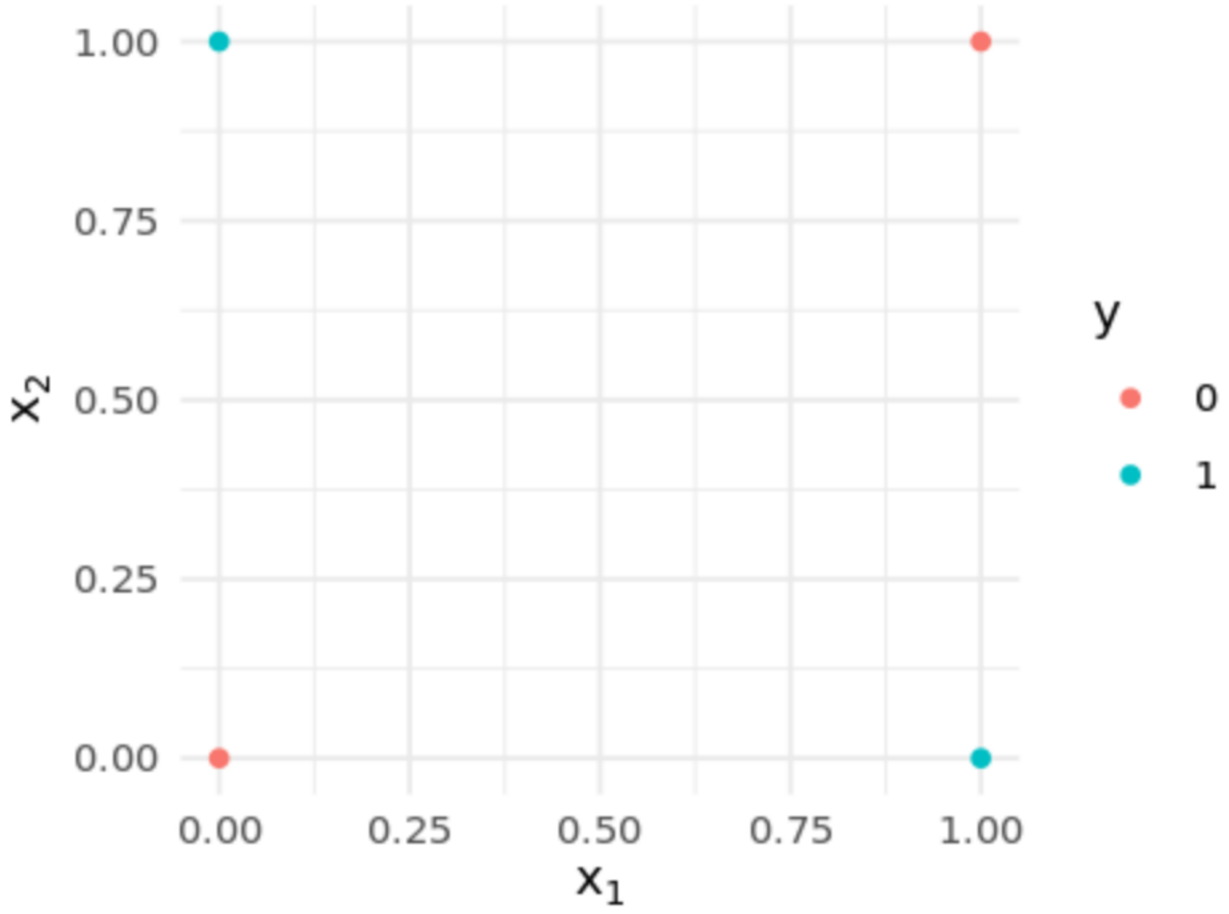


Neur ayers



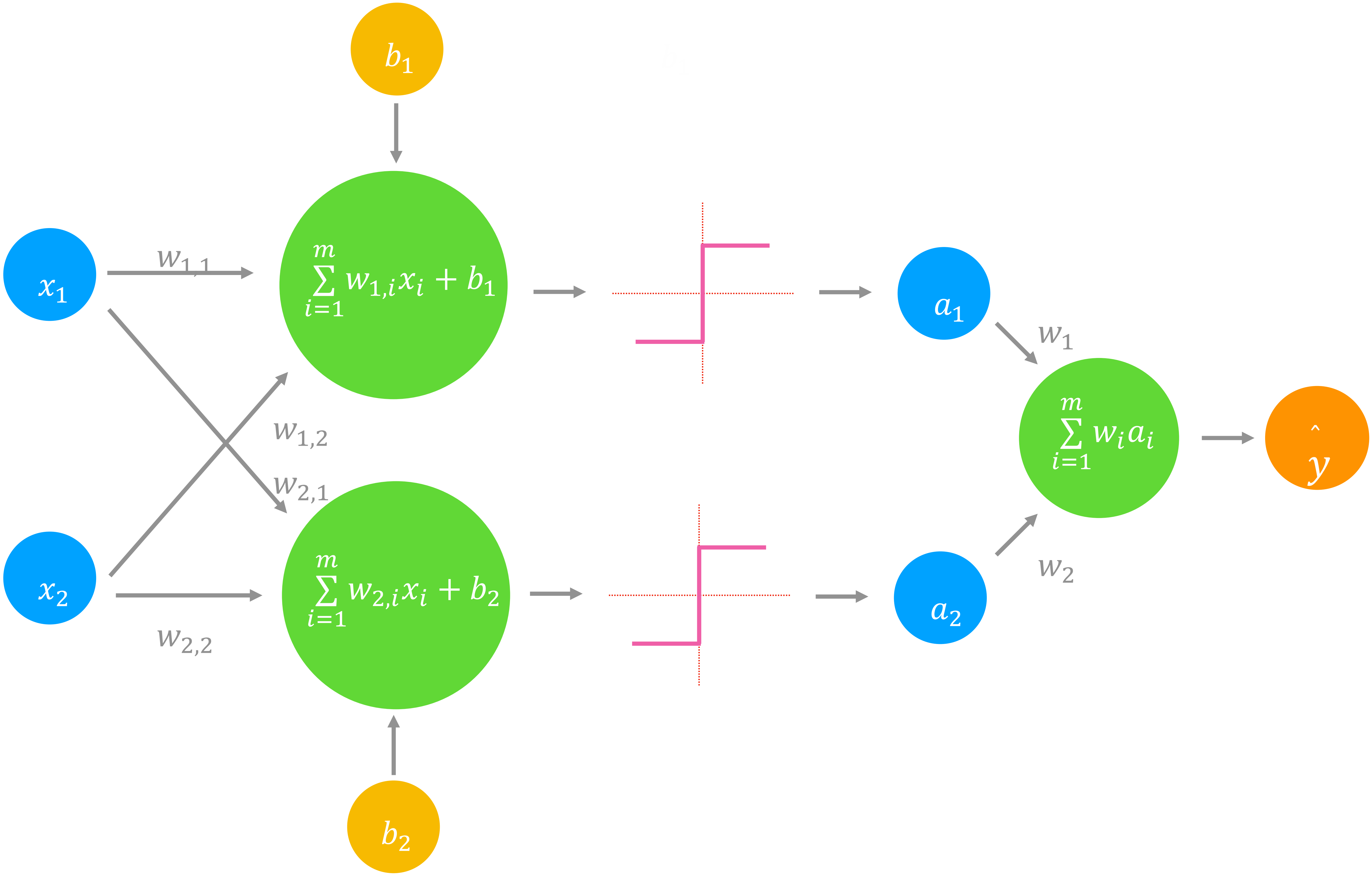
VGG-19
34-layer plain
34-layer residual
image
output size: 224
3x3 conv, 64
3x3 conv, 64
pool, /2
output size: 112
3x3 conv, 128
3x3 conv, 128
pool, /2
output size: 56
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool, /2
output size: 28
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
output size: 14
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
output size: 7
fc 4096
fc 4096
fc 1000
output size: 1
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
avg pool
fc 1000
1
64
64
input image tile
572 x 572
570 x 570
568 x 568
128
128
284²
282²
280²
256
256
140²
138²
136²
512
68²
66²
128
64
64
2
output segmentation map
392 x 392
390 x 390
388 x 388
388 x 388
128
196²
196²
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1
Ima
Conv
Conv
Max
Conv3
Conv3
Max
Conv3
Conv3
Conv3
144
9.0

The XOR problem



- The XOR problem is a famous problem (1960s) in deep learning.
- Early networks only used a single layer, and cannot solve XOR.
- By adding a single hidden layer we can implement XOR.

Prediction with a hidden layer



Hands-on

Go to https://https://jupyter.lisa.surfsara.nl:8000/

Notebook: 01b-network-from-scratch.ipynb

16:30-17:15

Summary

- Deep learning is a subfield of machine learning in which the model learns to represent the data using many levels of concept hierarchies.

- Deep learning has greatly improved performance in many tasks involving complex data.

- Neural networks are made from connected **neurons** which combine into **layers**. The network maps the **input x** to an **output y**.

- The environment @ https://jupyter.lisa.surfsara.nl:8000/

- If you have any questions, post them @ Canvas