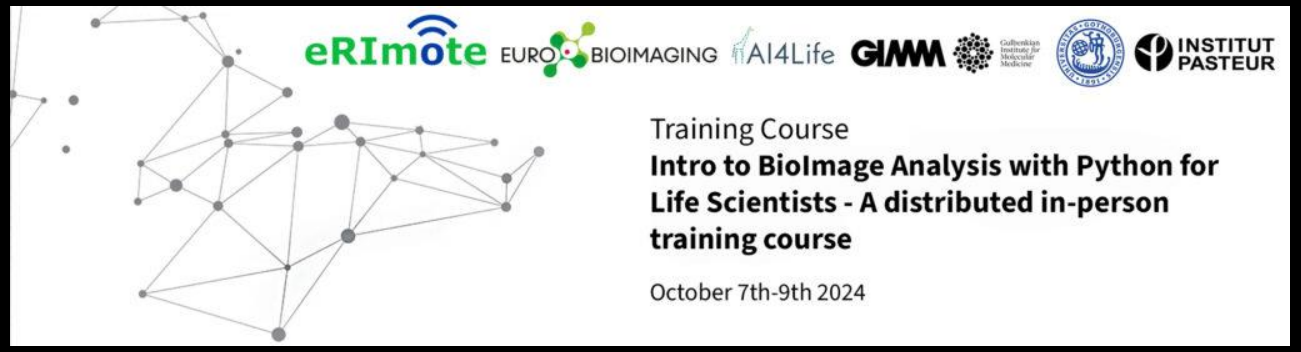


Day 3 - Introduction to Neural Networks and Deep Learning

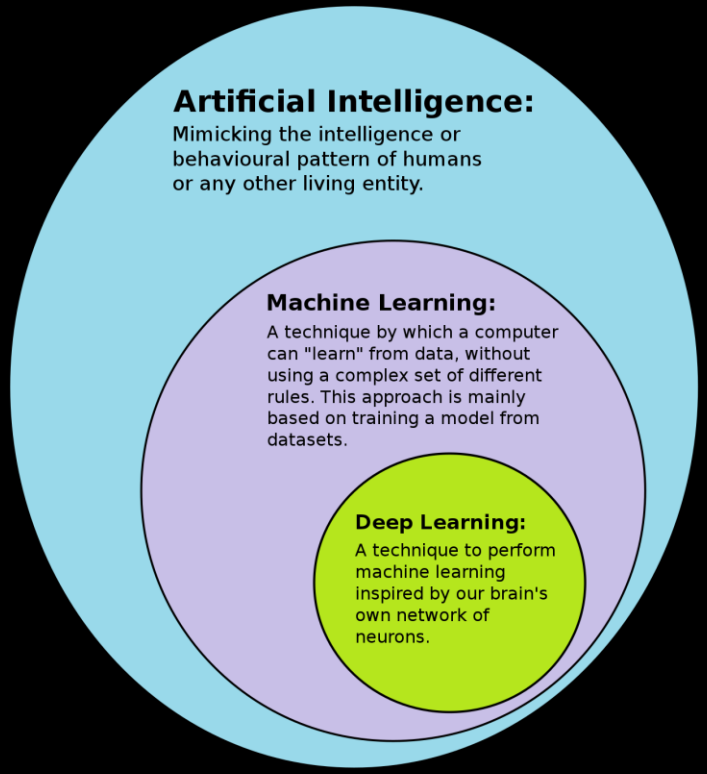
GIMM Gulbenkian Institute for Molecular Medicine

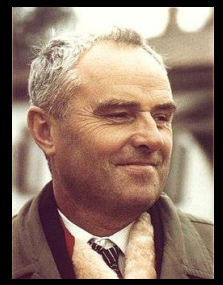
Mariana G Ferreira | Software Engineer
Ricardo Henriques' lab

9th October 2024

What is deep learning?

Deep learning is a subset of machine learning that's was first thought in the 1960s





1960s: Alexey Ivakhnenko works on deep neural networks

1986: Geoffrey Hinton proposes the backpropagation algorithm in its current form

For a long time, there were no more advancements, mostly because of lack of hardware capable of performing the massive calculations needed

The “boom” of Deep Learning

2000s: G. Hinton introduces “deep belief networks”

2012: AlexNet wins the ImageNet competition

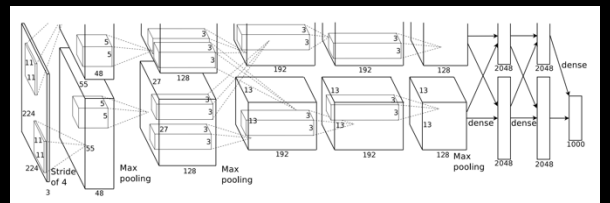
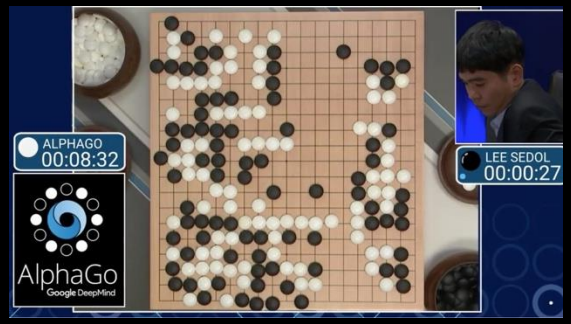
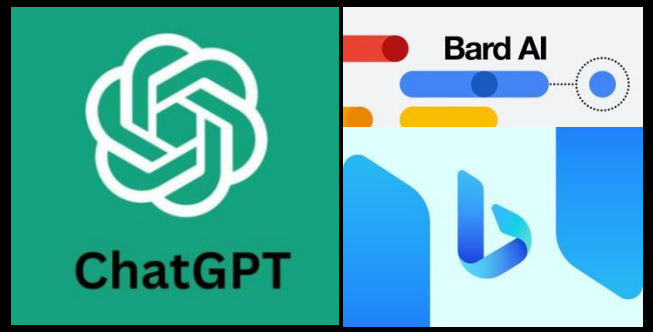


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network’s input is 150,528-dimensional, and the number of neurons in the network’s remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Recent years:



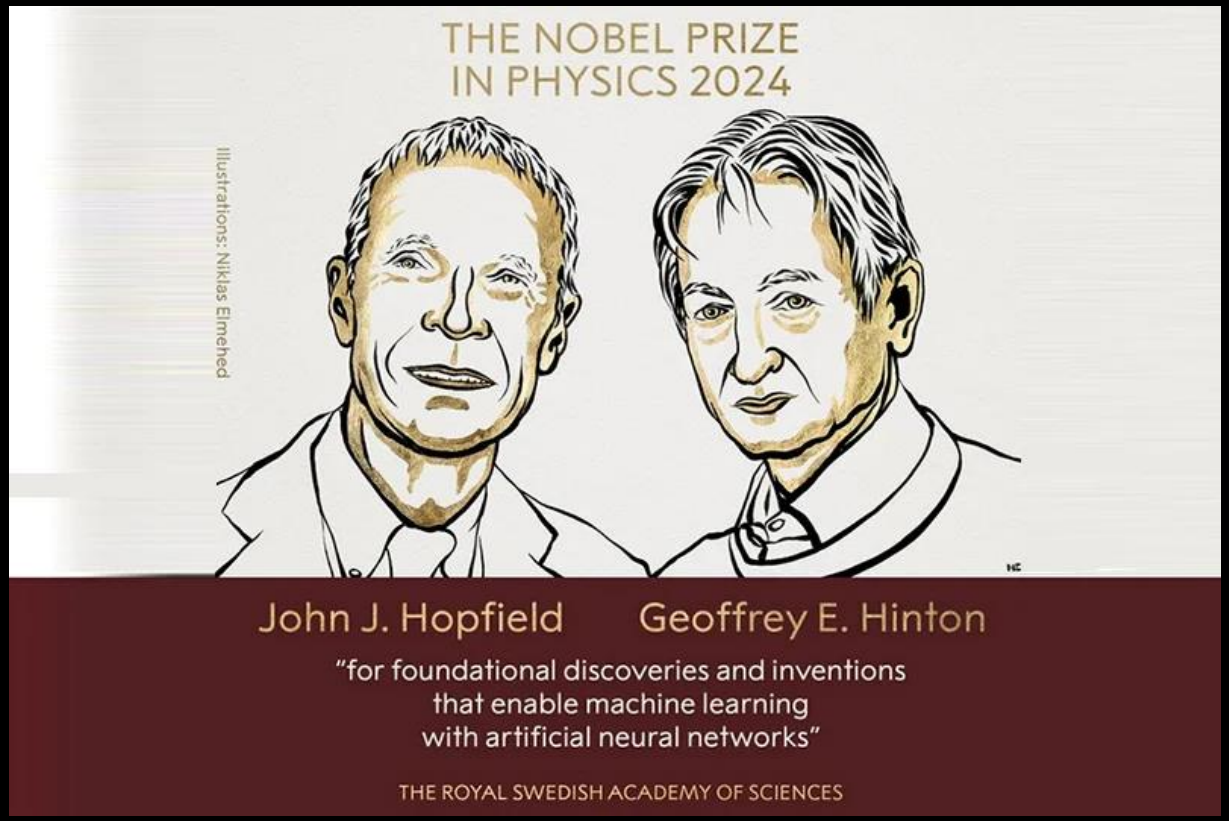
Google’s DeepMind capable of defeating world champion level players of Go



Large Language Models (LLM) capable of receiving questions from users and reply in a dialogue manner

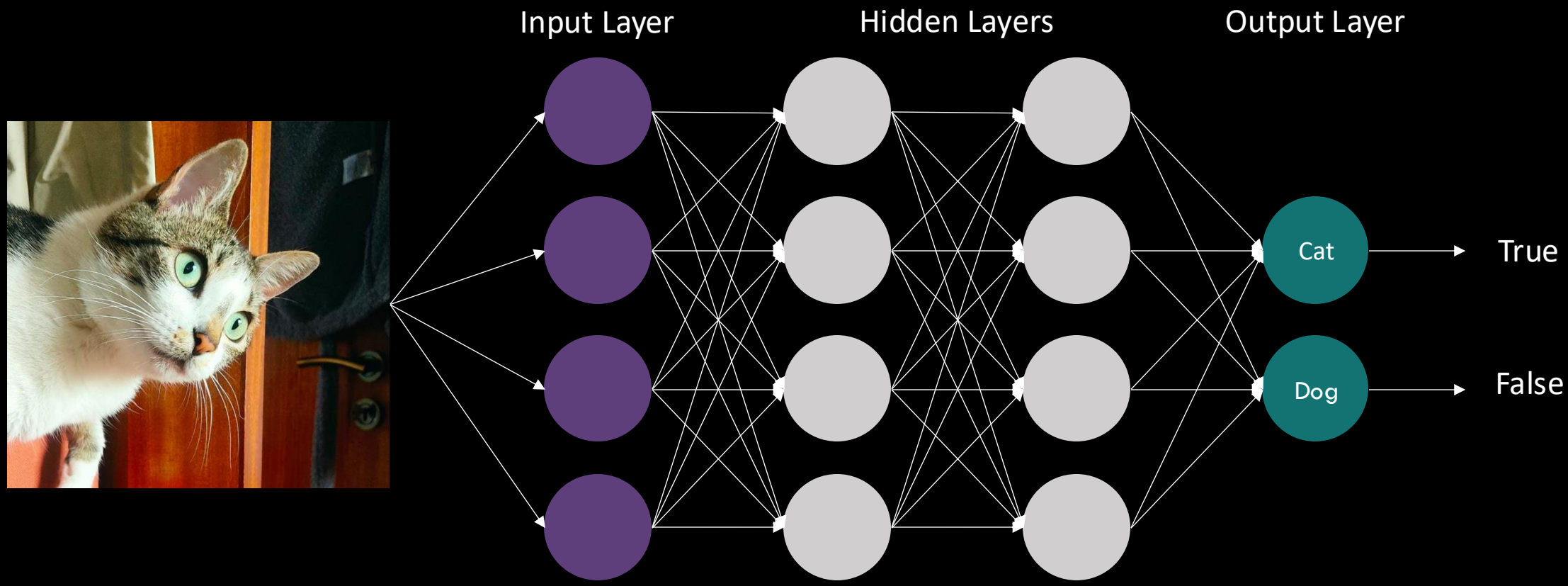
The “boom” of Deep Learning

In more recent years aka yesterday:



What is a neural network?

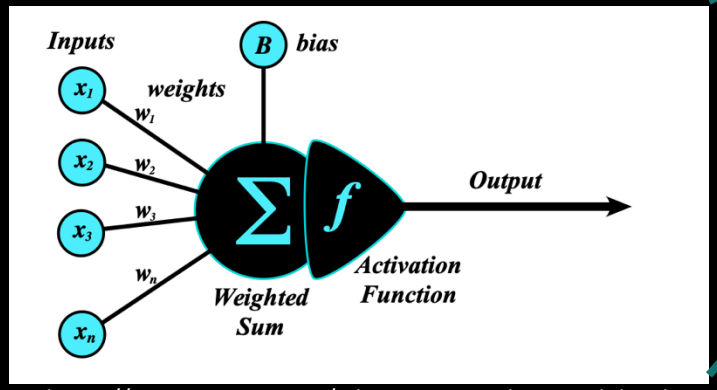
Example network for classifying pictures as cats or dogs



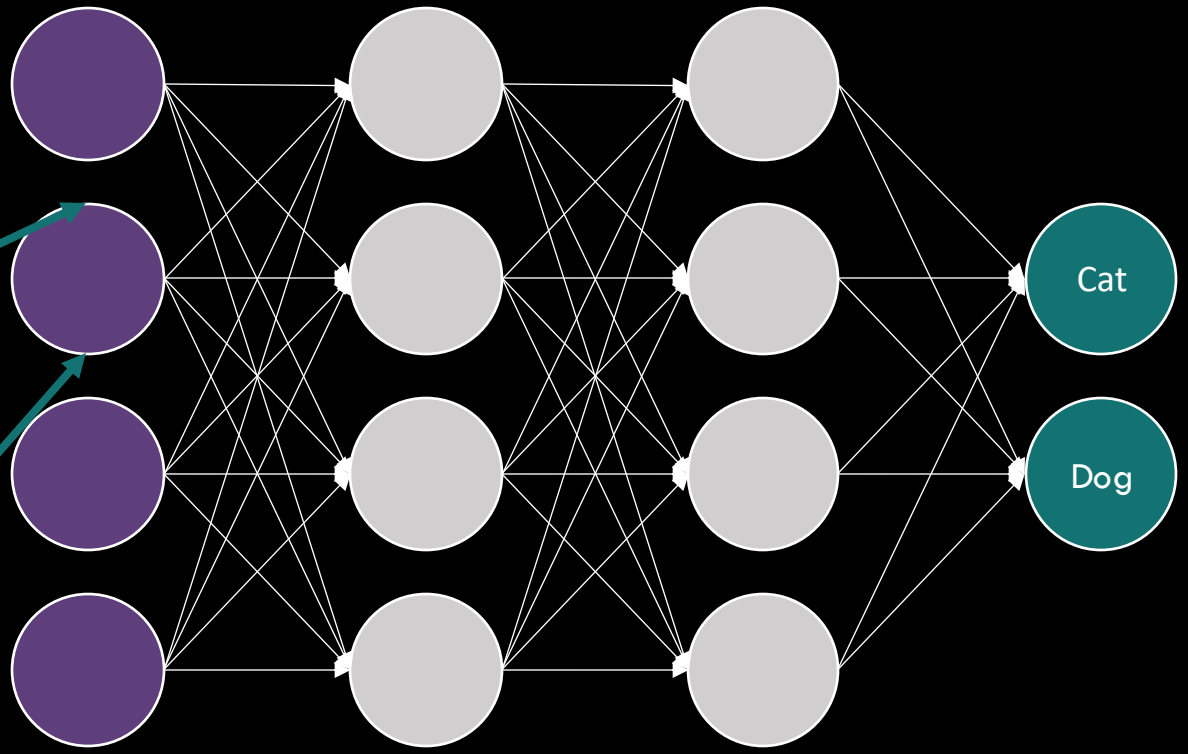
What is a neuron?

Neurons form the building blocks of neural networks.

A neuron (perceptron) will take in an input, multiply it by a weight and then apply a filter.
The result of this filter is then passed on to the other layers of the network

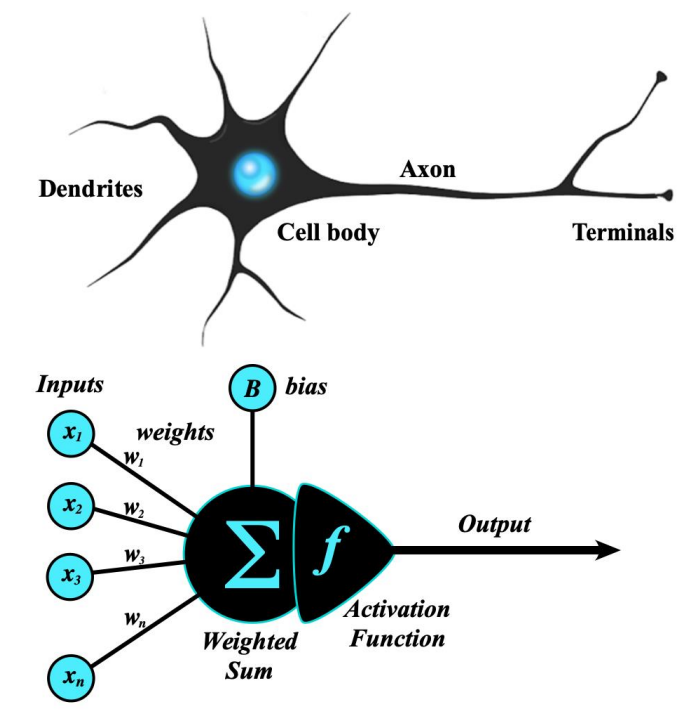


https://mriquestions.com/what-is-a-neural-network.html

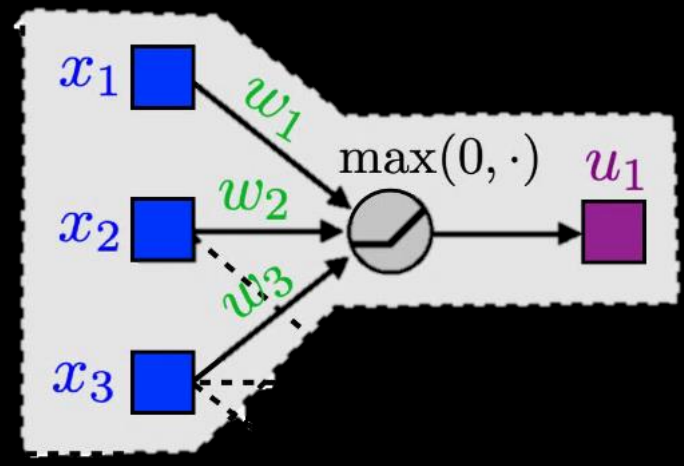


Neural networks

Deep learning neural networks are fairly like how neurons work in the human brain

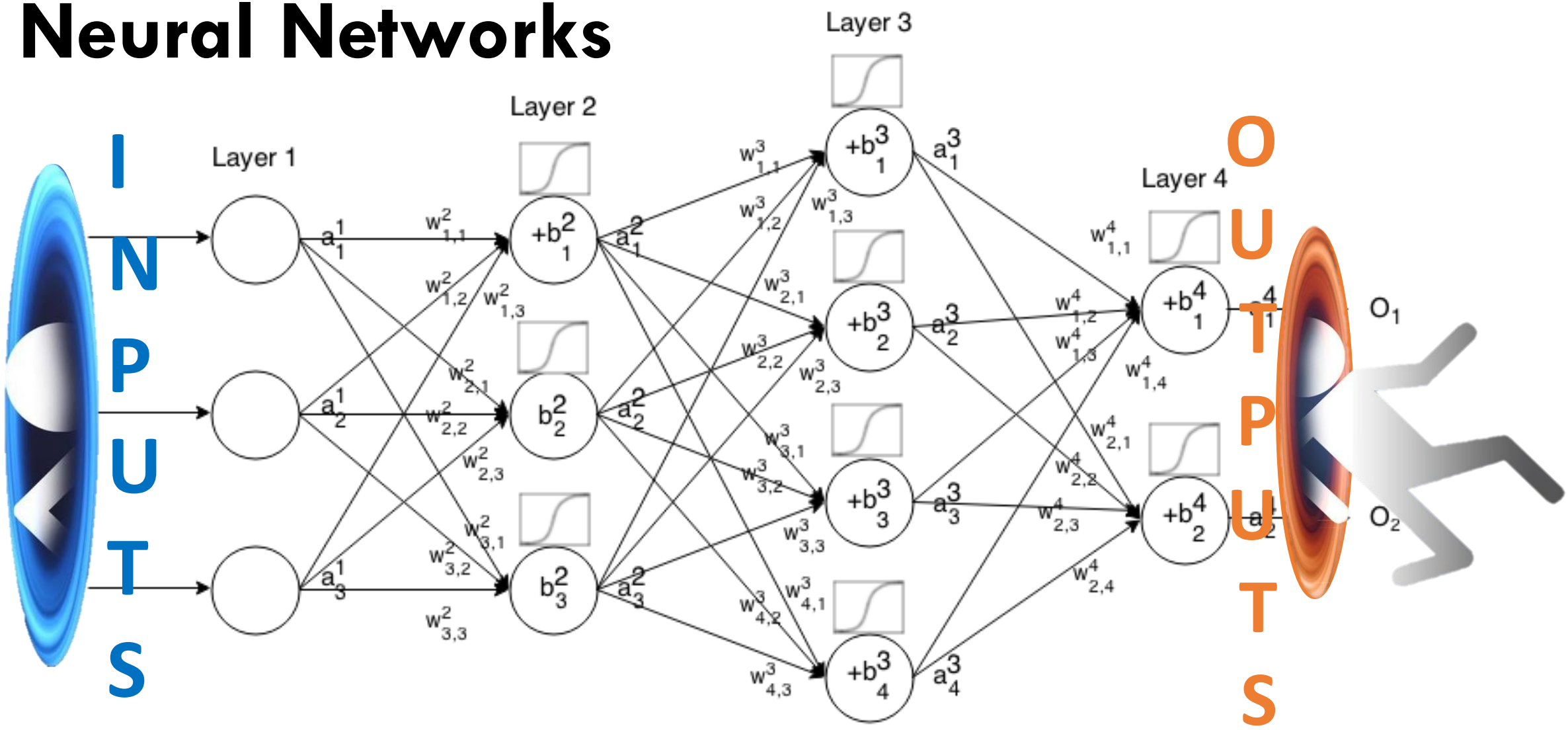


https://mriquestions.com/what-is-a-neural-network.html



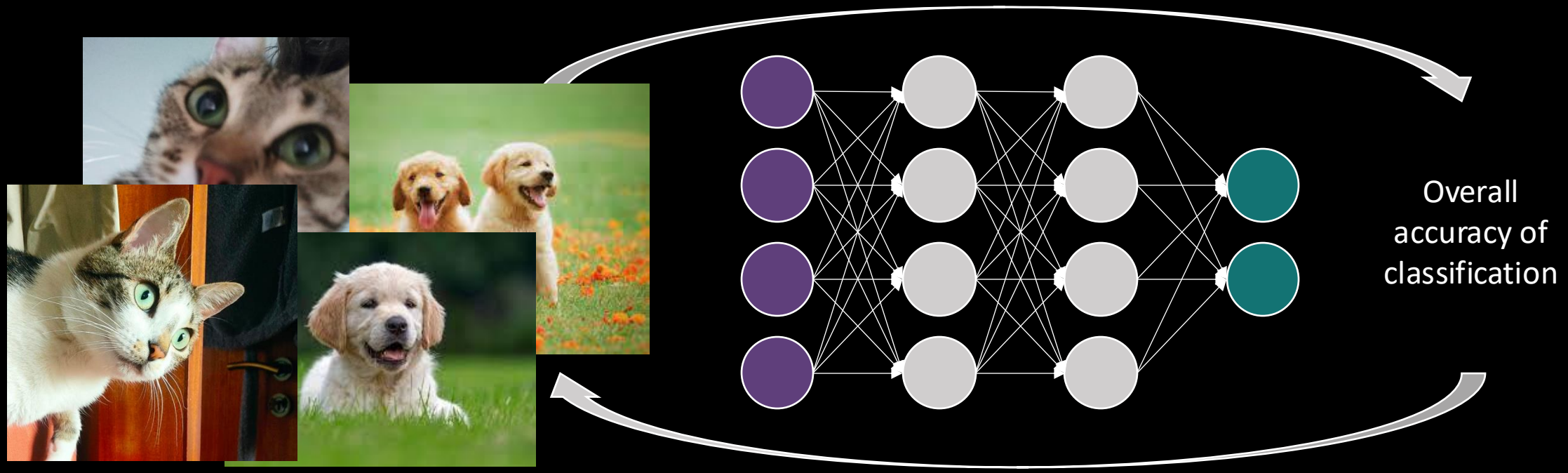
$$u_1 = \max(w_1 \times x_1 + w_2 \times x_2 + w_3 \times x_3 + w_4, 0).$$

Neural Networks



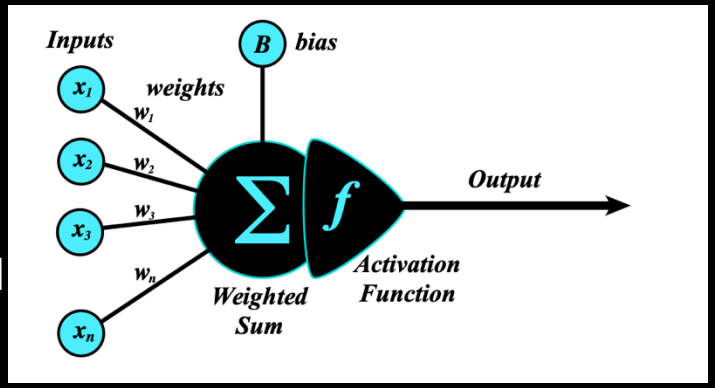
How a neural network learns a given task

The training dataset will go through the network and the output is measured, then the network will automatically adjust the filters and weights and repeat the process, trying to achieve the maximum accuracy possible



Key Concepts

- **Layer** – set of neurons in a network, can be Input, Output or Hidden
- **Feature** – input variable for a NN
- **Neuron** – distinct unit within a hidden layer
- **Activation Function** – enables non-linear relationships between features and label
- **Bias** – discrete offset from origin
- Forward propagation
- Backward propagation

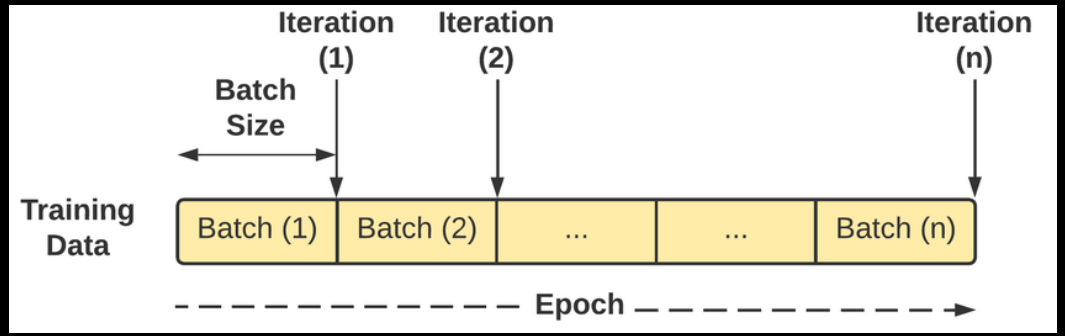


https://mriquestions.com/what-is-a-neural-network.html

https://developers.google.com/machine-learning/glossary

Key Concepts

- Training dataset
- Validation dataset
- Epoch – one pass over the whole training dataset
- Batch size - number of examples per iteration
- Iterations – updates of the weights and biases, one forward and one backwards pass
- Patch size - subset of the input image being fed to the NN

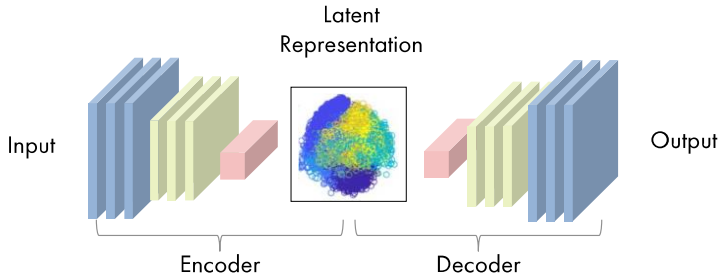


https://developers.google.com/machine-learning/glossary

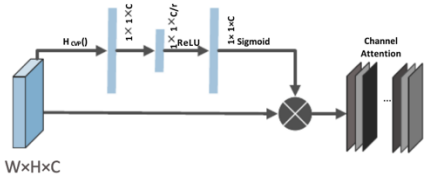
https://www.researchgate.net/figure/Illustration-of-batch-size-iteration-and-epoch_fig1_378880342

Common architectures

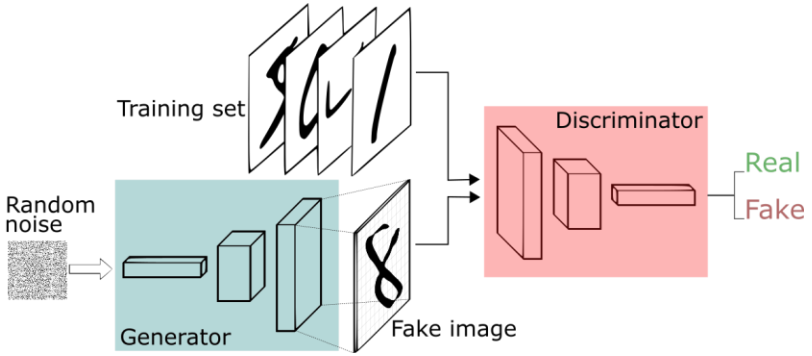
Autoencoders



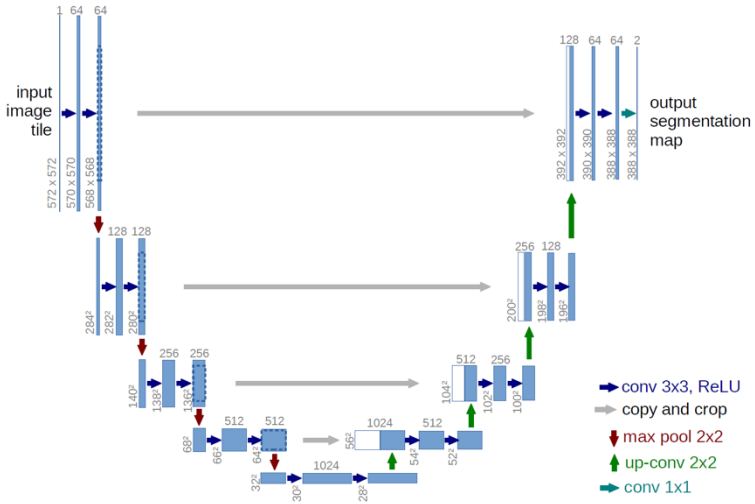
Channel attention



GAN

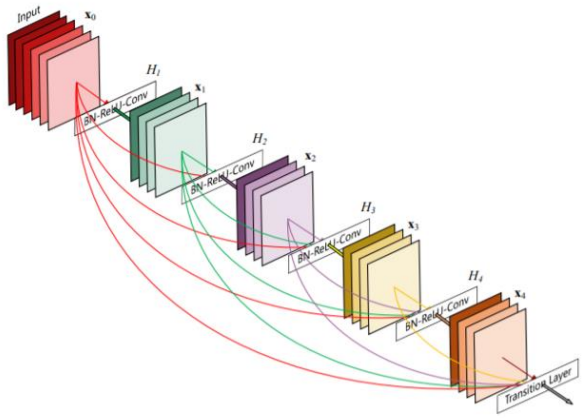


U-Net

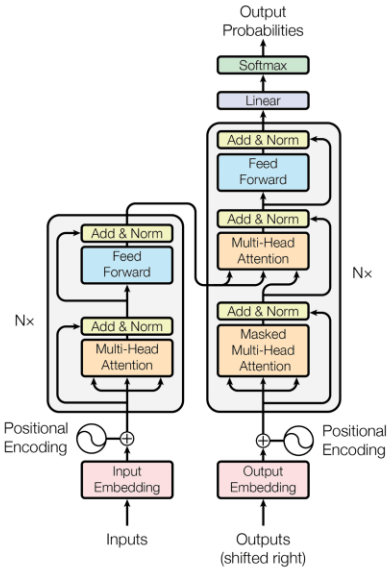


ResNet

(residual/skip connections)

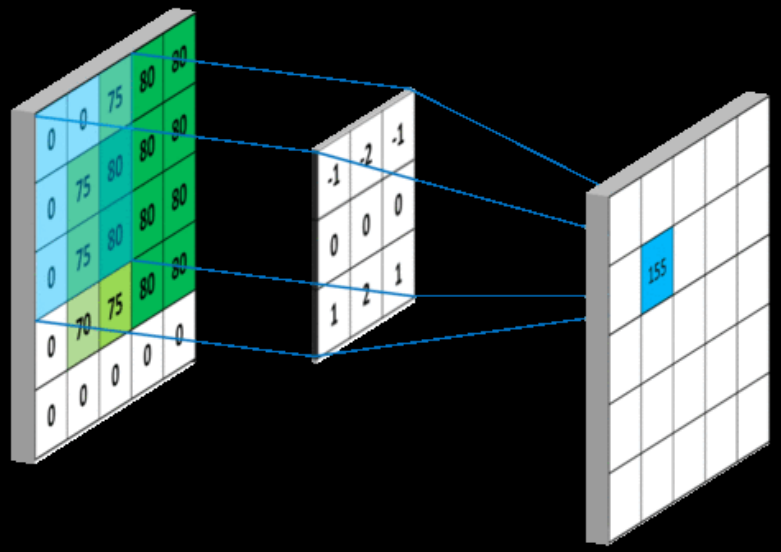


Transformer

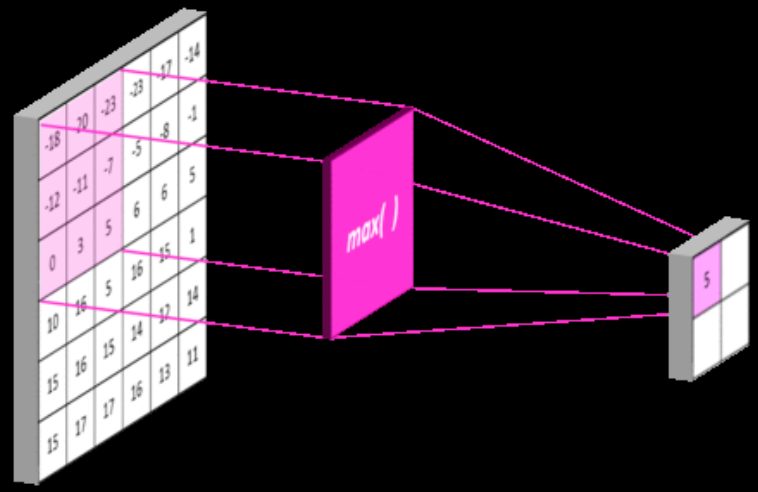


U-NET

Convolutional layers



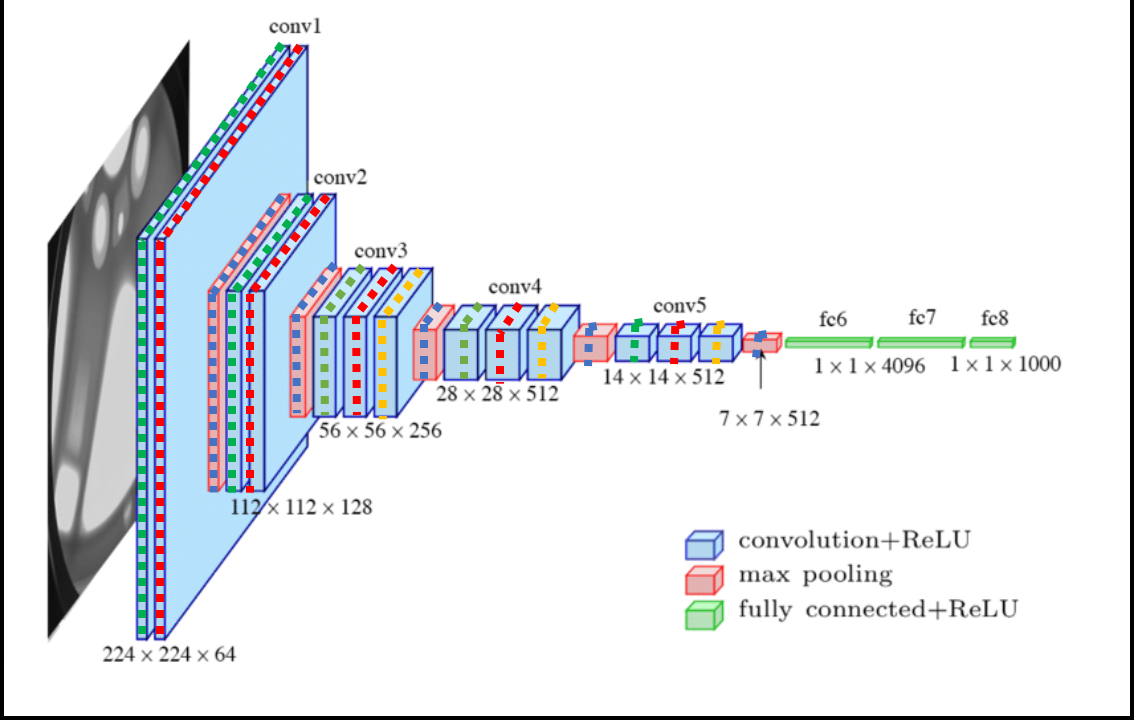
Pooling layers



https://adamharley.com/nn_vis/cnn/2d.html

Example code

VGG-16 [1]

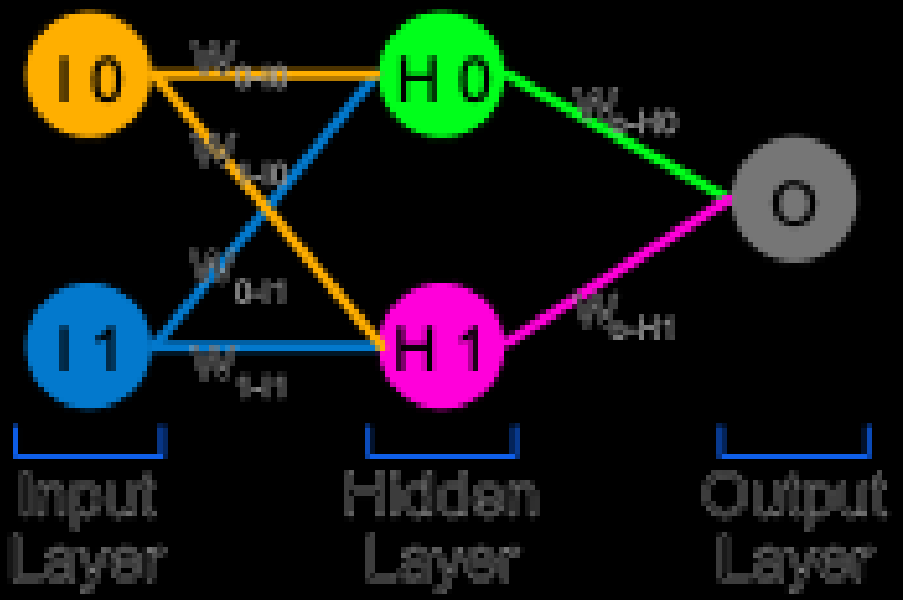



```
class VGG(nn.Module):
"""
Standard PyTorch implementation of VGG. Pretrained imagenet model is used.
"""
def __init__(self):
  super().__init__()
  self.features = nn.Sequential(
    # conv 1
    nn.Conv2d(3, 64, 3, padding=1),
    nn.ReLU(),
    nn.Conv2d(64, 64, 3, padding=1),
    nn.ReLU(),
    nn.MaxPool2d(2, stride=2, return_indices=True),

    # conv 2
    nn.Conv2d(64, 128, 3, padding=1),
    nn.ReLU(),
    nn.Conv2d(128, 128, 3, padding=1),
    nn.ReLU(),
    nn.MaxPool2d(2, stride=2, return_indices=True),

    # conv 3
    nn.Conv2d(128, 256, 3, padding=1),
    nn.ReLU(),
    nn.Conv2d(256, 256, 3, padding=1),
    nn.ReLU(),
    nn.Conv2d(256, 256, 3, padding=1),
    nn.ReLU(),
    nn.MaxPool2d(2, stride=2, return_indices=True),

    # conv 4
    nn.Conv2d(256, 512, 3, padding=1),
    nn.ReLU(),
    nn.Conv2d(512, 512, 3, padding=1),
    nn.ReLU(),
    nn.Conv2d(512, 512, 3, padding=1),
    nn.ReLU(),
    nn.MaxPool2d(2, stride=2, return_indices=True),

    # conv 5
    nn.Conv2d(512, 512, 3, padding=1),
    nn.ReLU(),
    nn.Conv2d(512, 512, 3, padding=1),
    nn.ReLU(),
    nn.Conv2d(512, 512, 3, padding=1),
    nn.ReLU(),
    nn.MaxPool2d(2, stride=2, return_indices=True)
  )

  self.classifier = nn.Sequential(
    nn.Linear(512 * 7 * 7, 4096),
    nn.ReLU(),
    nn.Dropout(),
    nn.Linear(4096, 4096),
    nn.ReLU(),
    nn.Dropout(),
    nn.Linear(4096, 1000)
  )

  # We need these for MaxUnpool operation
  self.conv_layer_indices = [0, 2, 5, 7, 10, 12, 14, 17, 19, 21, 24, 26, 28]
  Self.feature_maps = OrderedDict()
  self.pool_locs = OrderedDict()

def forward(self, x):
  for layer in self.features:
    if isinstance(layer, nn.MaxPool2d):
      x, location = layer(x)
    else:
      x = layer(x)
  x = x.view(x.size()[0], -1)
  x = self.classifier(x)
  return x
```

[1] Very Deep Convolutional Networks for Large-Scale Image Recognition, Karen Simonyan and Andrew Zisserman, 2015

Agenda for the afternoon

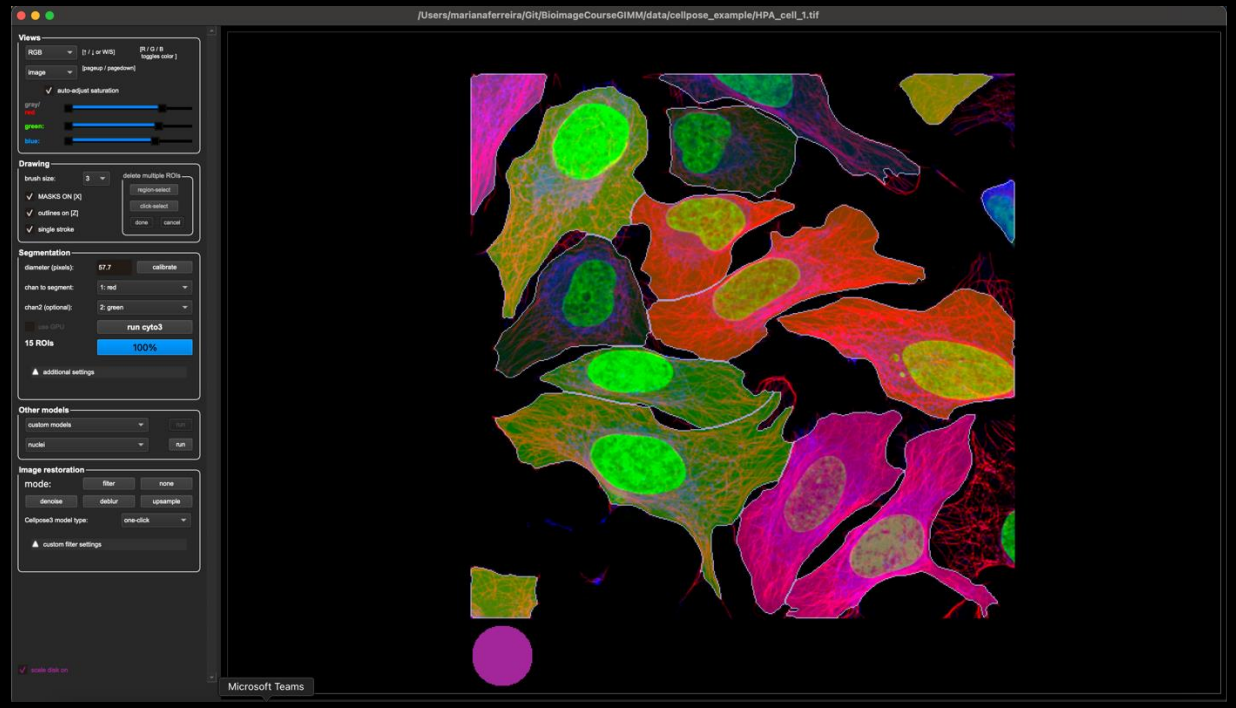
Notebook #4- Create your own Neural Network (Perceptron)



Agenda for the afternoon

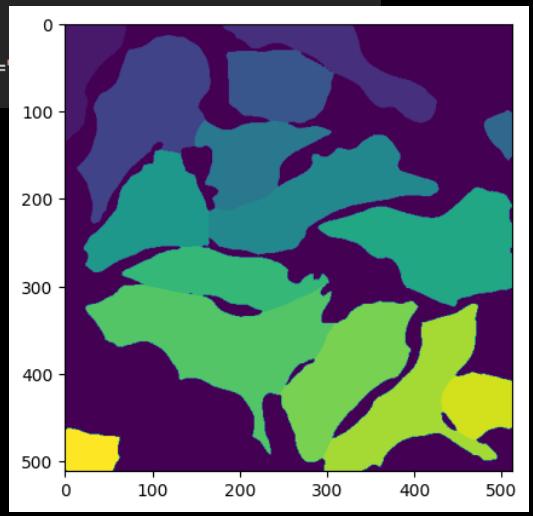
Notebook #5 – Deep Learning use case - Cellpose

Built-in Graphical User Interface (GUI)



Via Python code

```
from cellpose import io, models
import matplotlib.pyplot as plt

# import the image
img = io.imread('../../data/cellpose_example/HPA_cell_1.tif')

# Initialize the model
model = models.Cellpose(model_type='
```



Agenda for the afternoon

Notebook #6 - Accessible tools to use Deep Learning