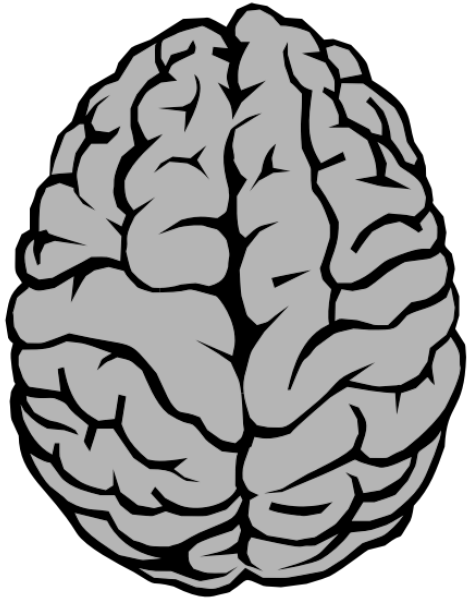
# Supervised Classification using Neural Networks methods in PyTorch

# Why are they called Neural Networks?



Inspired by the way the brain learns - pattern matching, frequency and recency.

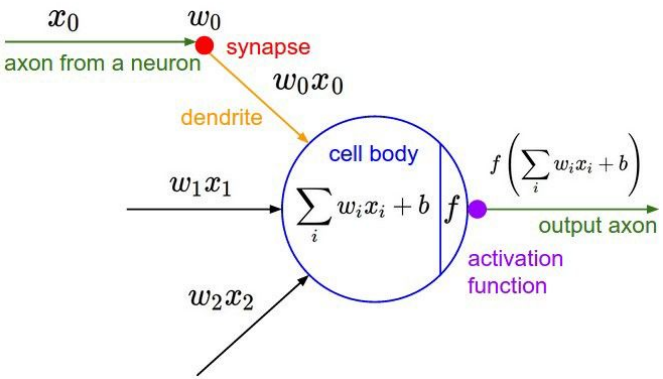
Connections are strengthened when we get things right and we tend to lose connections that aren't useful (neural pruning)

Tend to learn more from our mistakes. Fail hard and fail fast!

All models are wrong, but some are useful - George Box

Task generalisation and context awareness are missing from this model of learning

# Why are they called Neural Networks?




More explicitly inspired by the model of a neuron/perceptron

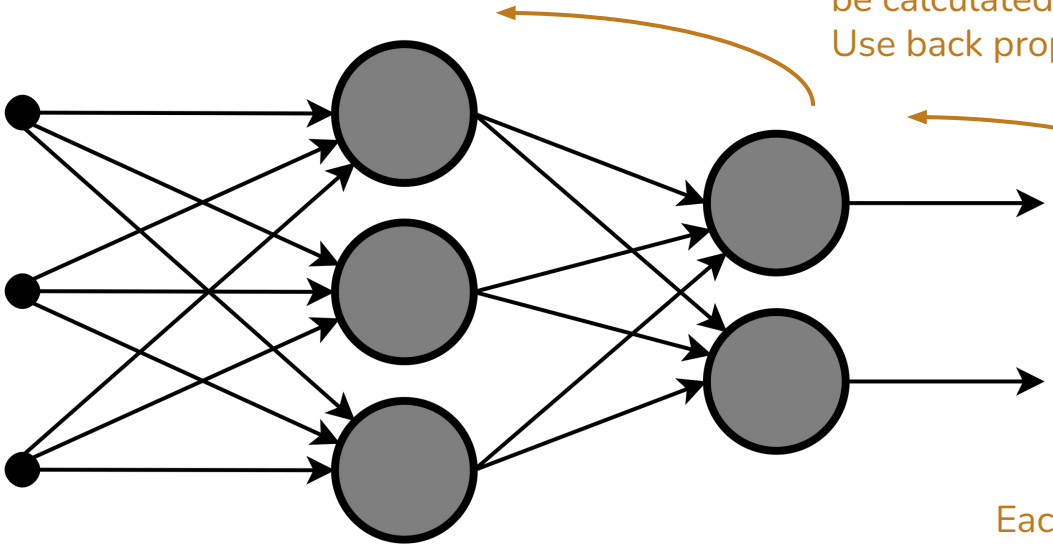
- Neuron receives inputs from dendrites
- If total input exceeds excitation threshold then neuron becomes active

We can adjust or learn the weights of each “dendrite” to get the right output, i.e. strengthen the connections that give the right answer and weaken the connections that give the wrong answer

Why do they work so well? Ability to construct complex and nonlinear functions by the composition of many layers

# The basic Neural Network for classification

Error attributable to each weight in the network can be calculated using the chain rule of differentiation. Use back propagation to update the network



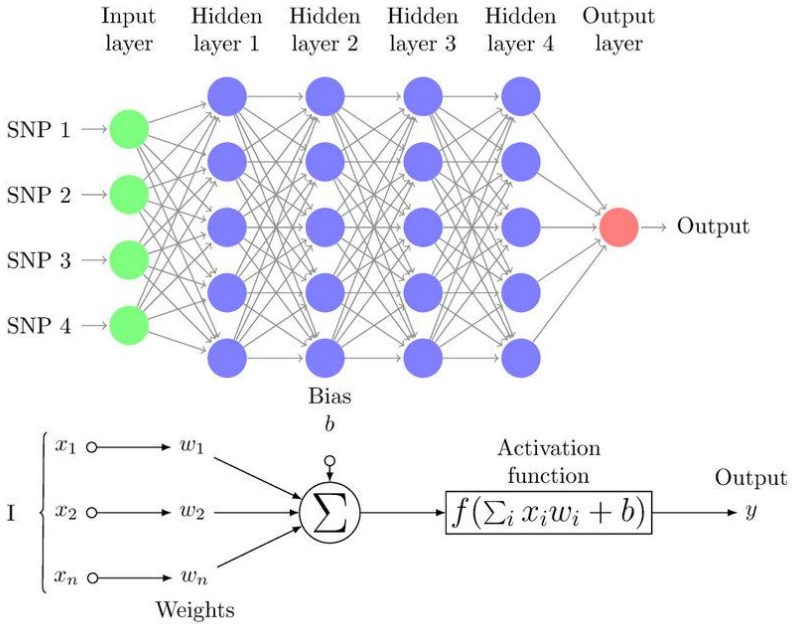
Calculate error w.r.t to true labels

hidden layer

output layer

Each hidden layer is characterised by its weight, bias and activation function. We can connect multiple hidden layers of varying sizes to build a network

# Training a neural network




Pass input data through the network to guess the label.

Use the ground truth to calculate the error. Adjust each weight by its contribution to the error.

Iterate through feedback loop until an acceptable loss is achieved

# Neural Network Building Blocks

## Featurisation

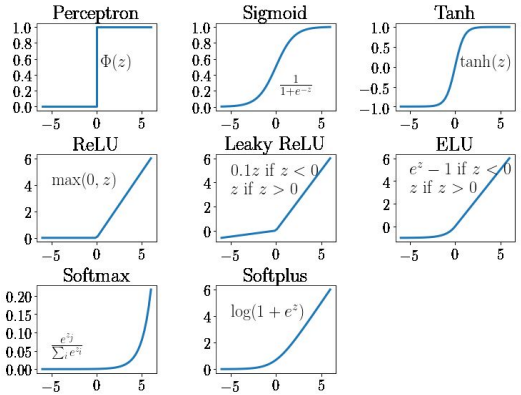
Linear layers

Convolutional layers

Pooling layers

Dropout

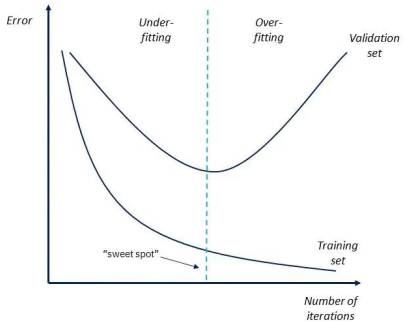
## Activation functions



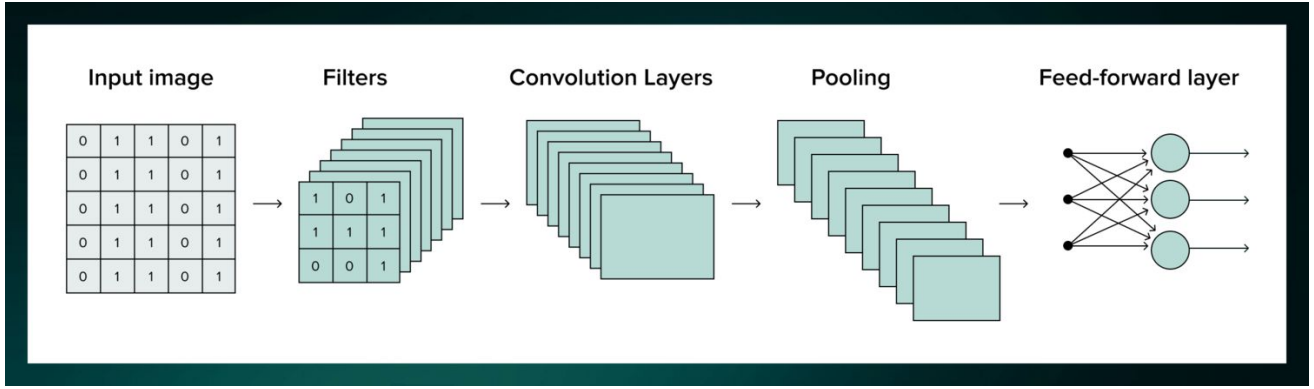

## Loss functions

Mean Squared Error (regression)

Binary Cross Entropy (classification)



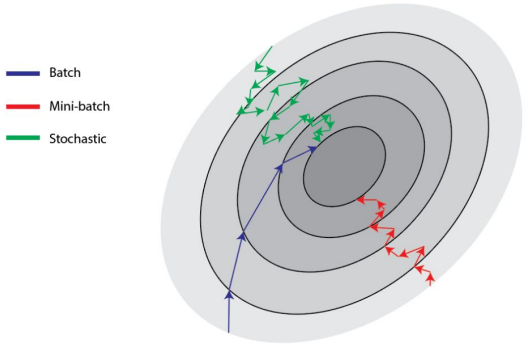
# An example image classification network might look like….



# Other aspects

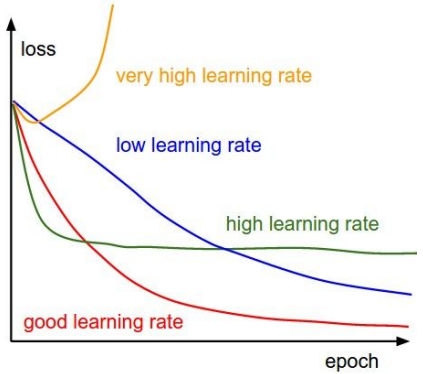
Input is typically passed in batches before updates rather than either individually or all at once

- Batch size can have an impact on your results. Size of data that can be loaded in at run time may be a constraint on your batch size
- An Epoch is when all batches have been passed through




The learning rate is a hyperparameter that controls how much to change the model in response to the error at each iteration

- Large learning rates can create unstable training and lead to non-convergence. Too small will either take too long to converge or not make any learning progress
- Learning rate can be scheduled to vary through the training process



# Implementation