Artificial Neural Networks (ANNs) Basic Theory

Max Cotton

1 Structure of an ANN:

- Input layer of multiple inputs/neurons (in an array)
- Hidden layers of calculations (the more layers, the more accurate the prediction), consisting of the following:
  - An Activation function:
    - Calculate the dot product of the input array with a hidden weight array, then sum the result with a bias
  - A Transfer function to get an output, Eg: Sigmoid function to transform the result of the Activation function to a number between 0 and 1, then the result can be classified as closer to 0 or 1 (known as logistic regression), with 0 being one state and 1 being another
- Output layer to output the final result of the calculations from the hidden layers, consisting of the following:
  - An Activation function:
    - Caclulate the dot product of the hidden layer output with an output weight array, then sum the result with a bias
  - A Transfer function to get an output (round output to one end of range with a meaning)
- Afterwards, the weights and the bias are all initialised to zero/s

2 How ANNs are trained:

- Forward Propagation, the process of feeding inputs into the neural network and getting a result/prediction
- Back Propagation, the process of calculating the error in the prediction then adjusting the weights and biases accordingly, consisting of the following:
  - A Cost/Loss function (used for graphs and deriving formula for gradient descent):
    - Finds the average of the difference between the actual and the predicted value for each input
  - Gradient descent:

* Update the hidden/output weight arrays and bias, by subtracting the rate of change of Loss with respect to Weight, multiplied with a learning rate
* This repetitive process will continue to reduce the loss to a minimum, if the learning rate is set to an appropriate value

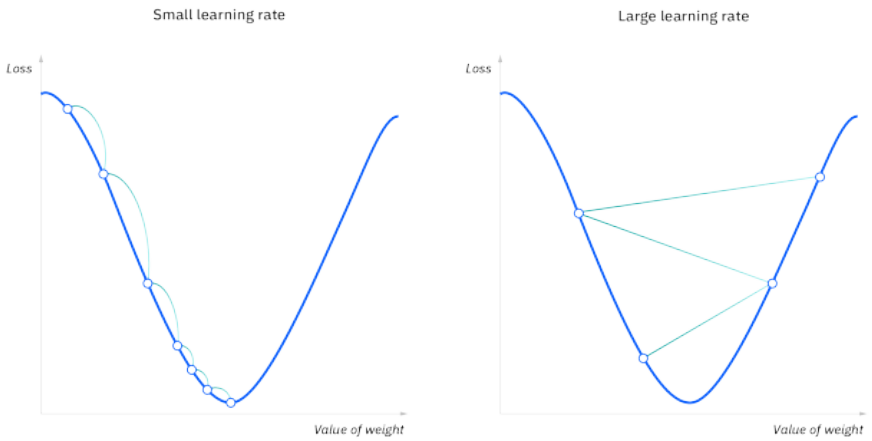

Figure 1: Gradient Descent
sourced from https://www.ibm.com/topics/gradient-descent