DEMYSTIFYING DEEP LEARNING: TUTORIAL SERIES

CHAPTER 3: BUILDING BLOCKS OF NEURAL NETWORK-2

Vaibhav Verdhan

May 18 2020

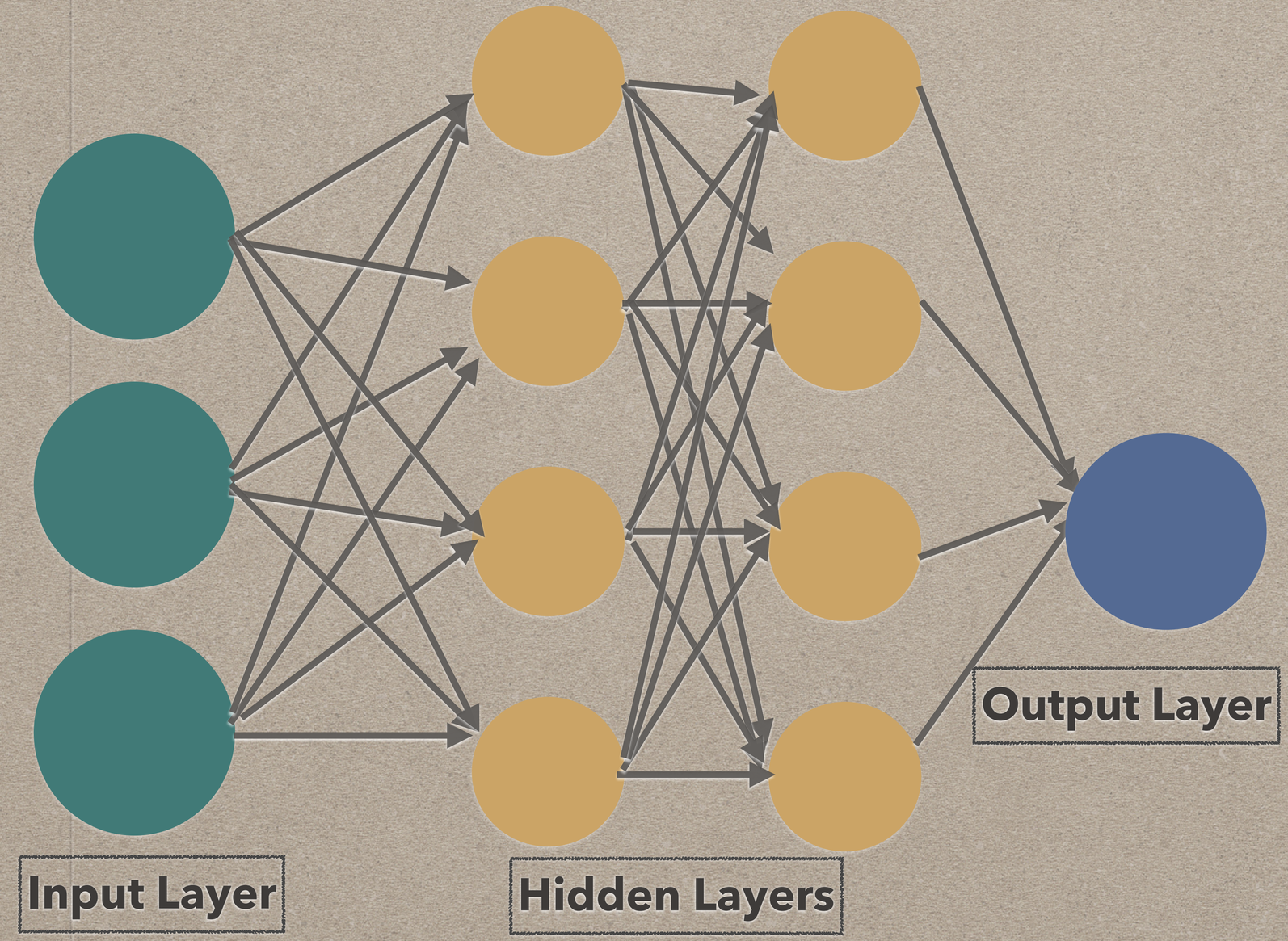
AGENDA FOR THE SERIES

Session	Topic
Session 1	Introduction to Deep Learning
Session 2	Building blocks of Neural Network - 1
Session 3	Building blocks of Neural Network - 2
Session 4	Convolutional Neural Network (May 22)
Session 5	CNN for Image Classification (May 29)
Session 6	CNN for Object Detection
Session 7	Architectures like AlexNet, Inception etc.
Session 8	Recurrent Neural Network
Session 9	NLP Applications of RNN

AGENDA FOR SESSION 3

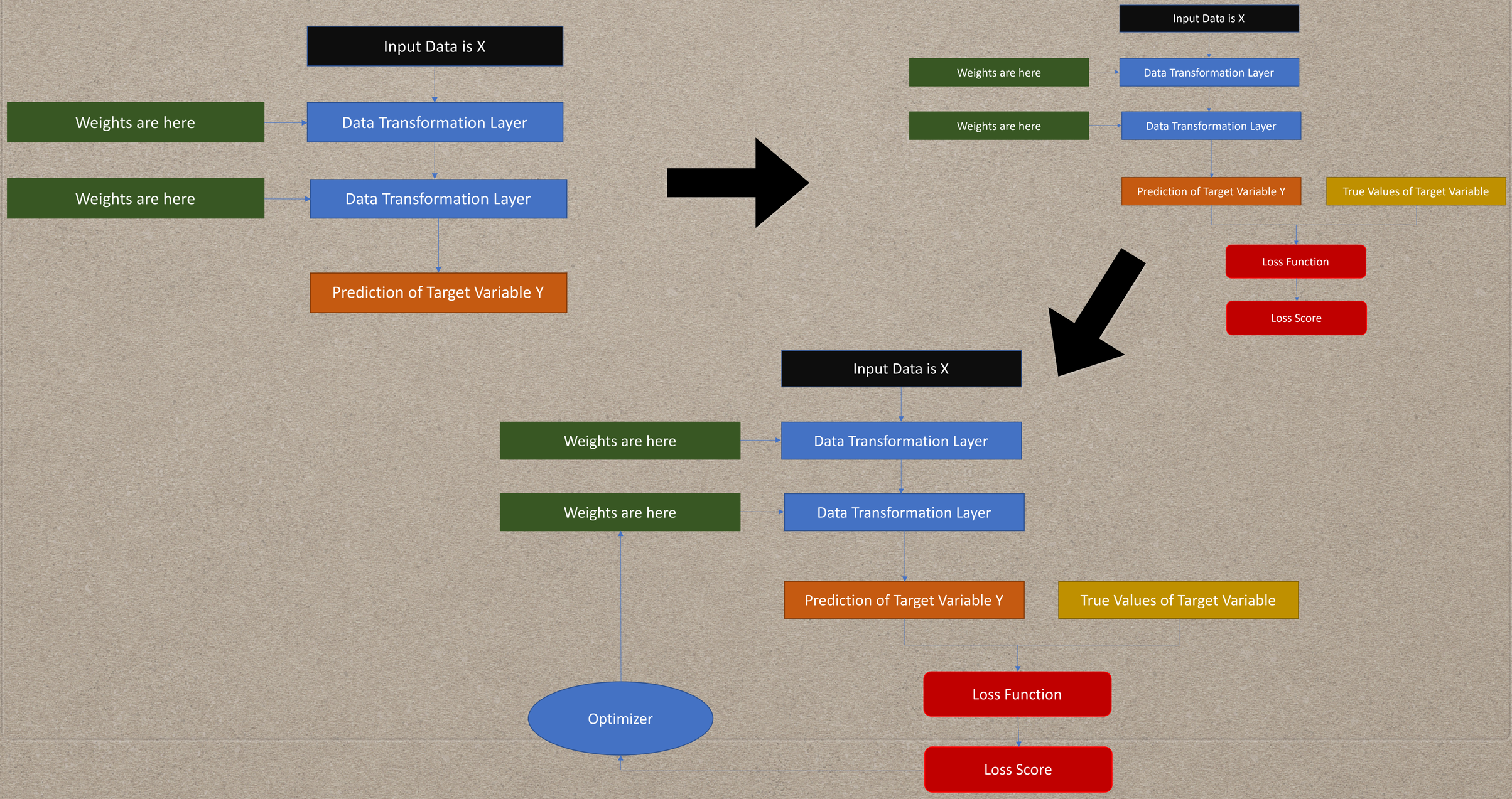
- *Components of Neural Network*
- *Refresh of the concepts*
- *Accuracy measurement*
- *Normalisation and dropout*
- *Common errors, debugging ways and best principals in Neural Networks*

A TYPICAL NEURAL NETWORK

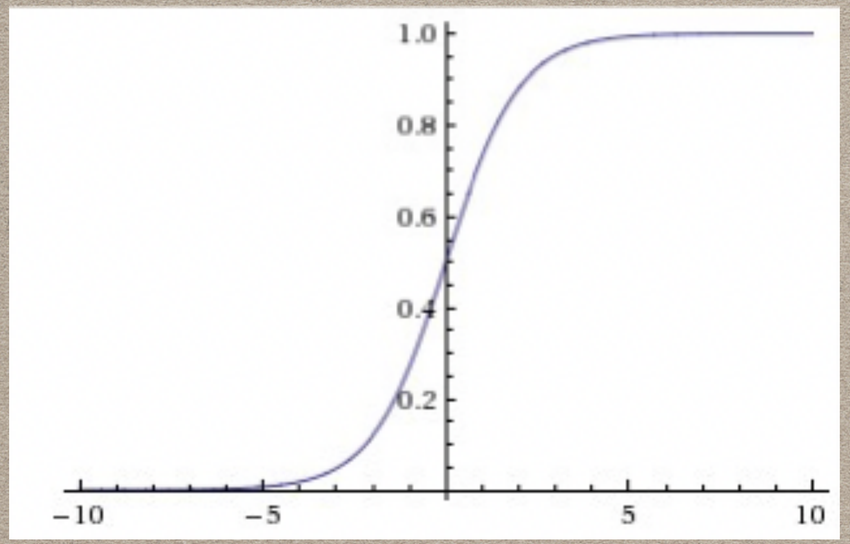


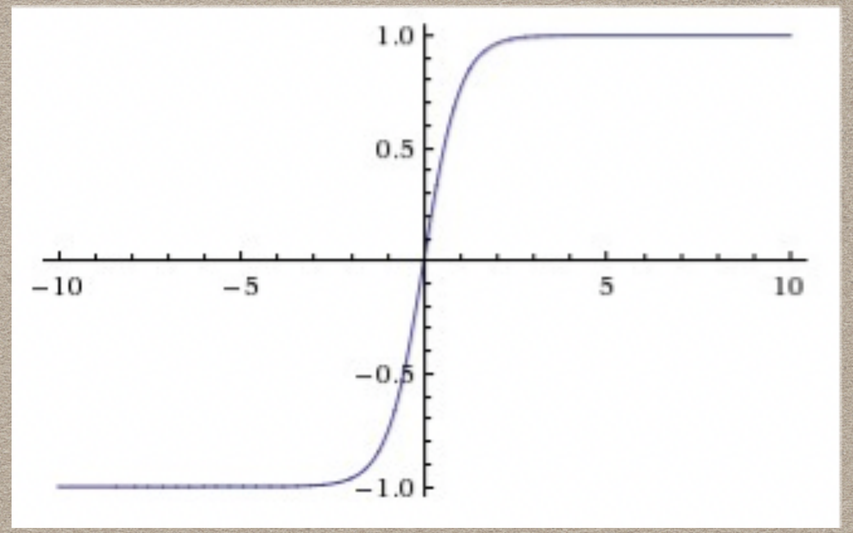
- *Deep refers to the depth in the Neural Network which means the number of hidden layers*
- *Number of neurons in a layer are also important*
- *But if the network is deep, it does not necessarily mean that the accuracy will be higher*

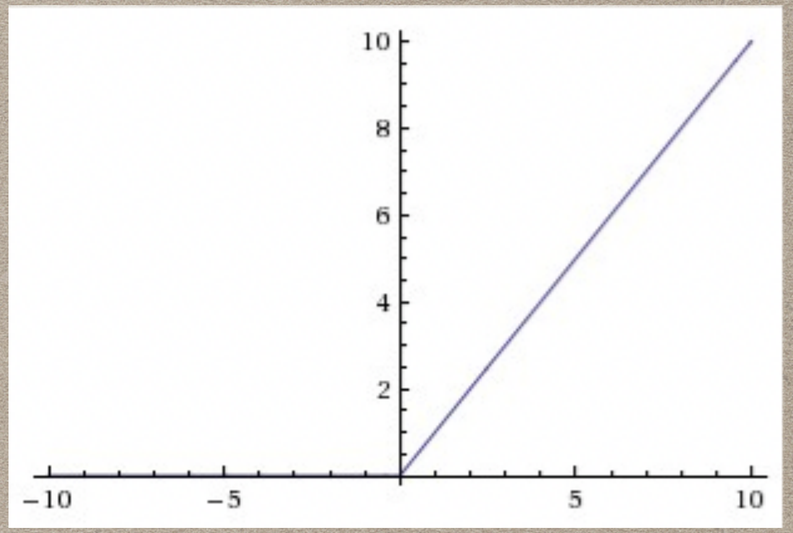
TRAINING PROCESS



ACTIVATION FUNCTIONS







Sigmoid Function

tanh Function

ReLU Function

- Why do we need non-linear functions?
- Every activation function (or non-linearity) takes a single number and performs certain fixed mathematical operation on it.

SOFTMAX FUNCTIONS

- Softmax function is a multinomial logistic classifier, i.e. it can handle multiple classes

- Softmax typically the last layer of a neural network based classifier

- Softmax function is itself an activation function, so doesn’t need to be combined with an activation function

LOSS FUNCTIONS

- Error = difference between the actual value and the predicted output.
- The function that is used to compute this error is known as Loss Function.

- Different loss functions will give different errors for the same prediction, and thus have a considerable effect on the performance of the model. And hence it is advisable to have different loss functions for regression vs classification
- One of the most widely used loss function is mean square error, which calculates the square of difference between actual value and predicted value.

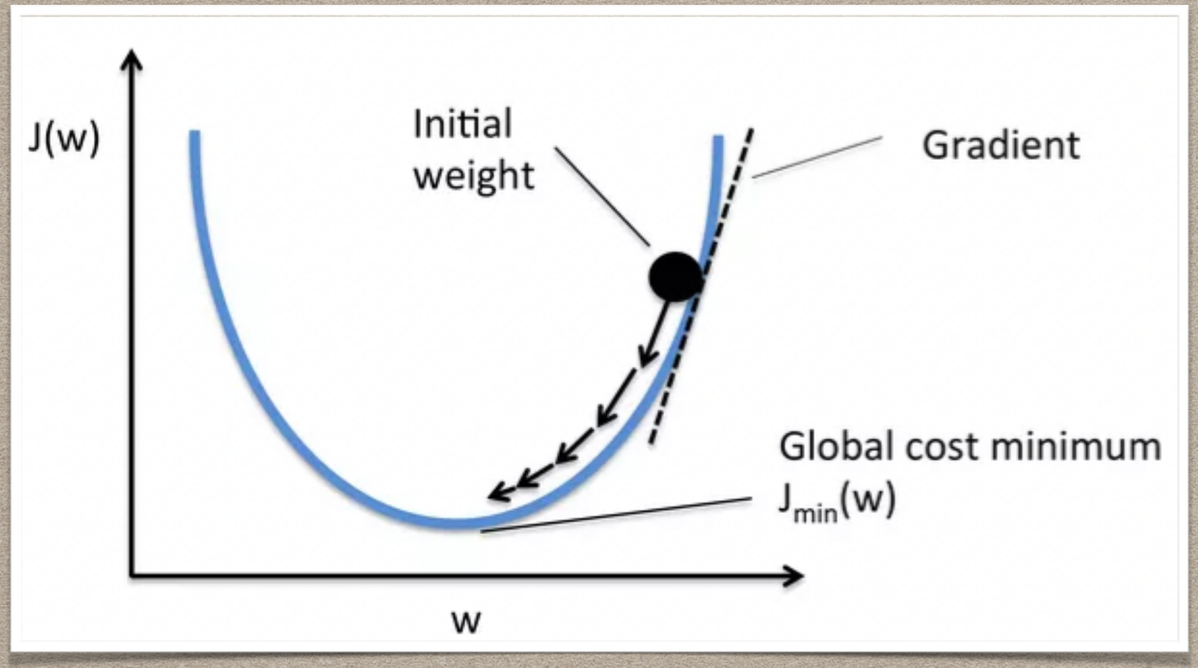
LOSS FUNCTIONS

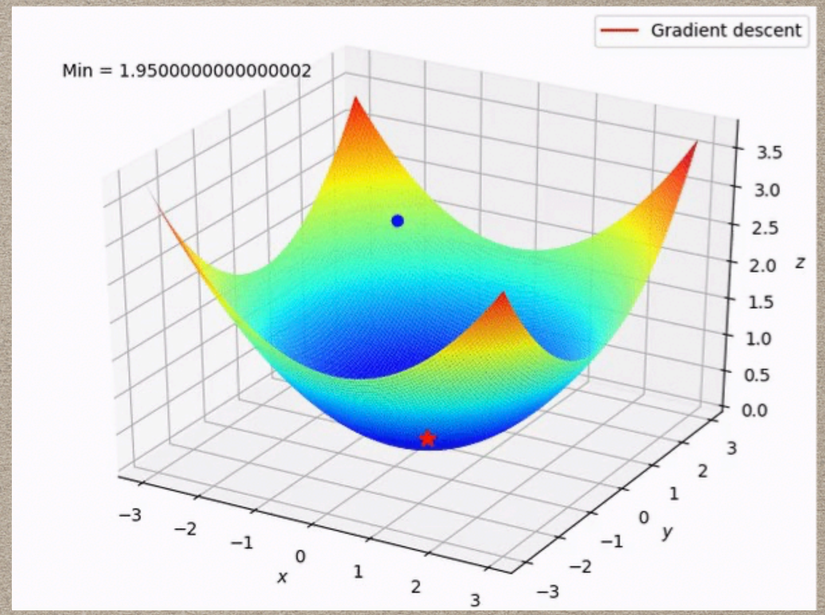
- Cross-entropy loss (often called Log loss) quantifies our unhappiness for the predicted output based on its deviation from the desired output

- Perfect prediction would have a loss of 0

- With gradient descent, we try to reduce this (cross-entropy) loss for a classification problem

OPTIMISATION: FIND WEIGHTS TO MINIMISE LOSS



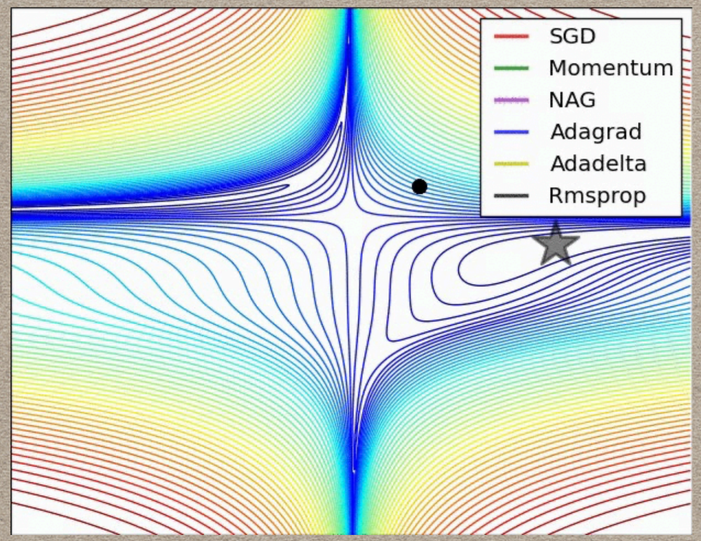


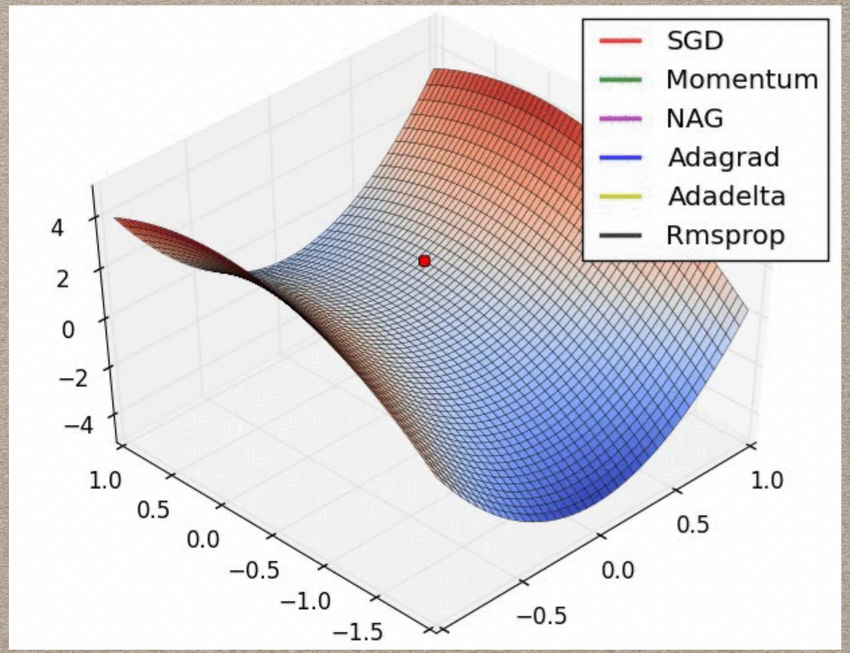
Optimisation functions usually calculate the gradient i.e. the partial derivative of loss function with respect to weights, and the weights are modified in the opposite direction of the calculated gradient. This cycle is repeated until we reach the minima of loss function.

The procedure of repeatedly evaluating the gradient and then performing a parameter update is called Gradient Descent.

IMPORTANCE OF THE LEARNING RATE

- Learning rate will directly impact the training
- Higher the rate faster is the decay of the algorithm
- If you keep low learning rates, linear improvements will be seen
- Advisable to have higher learning rates and then decrease it





CONFUSION MATRIX TO MEASURE CLASSIFICATION ACCURACY

	Actual Condition		Accuracy	
	Condition +ve	Condition - ve	**78%**	
Predicted Condition +ve	**465**	**60**	Precision **89%**	False discovery rate **11%**
Predicted Condition - ve	**104**	**121**	False omission rate **46%**	Negative predictive value **54%**
Prevalence **76%**	Sensitivity, Recall TPR **82%**	Fallout FPR **33%**	+ve likelihood ratio **2.47**	- ve likelihood ratio **0.27**
	Miss Rate FNR **18%**	Specificity TNR **67%**	F1 Score **165**	

Precision - % of true predictions are actually true?

FDR - % of true predictions are actually false?

FOR - % false predicitons are actually true?

NPV - % false predicitons are actually false?

LR+ ratio of TPE to FPR

LR- ratio of FNR to TNR

Sensitivity - % of actual positive recalled correctly?

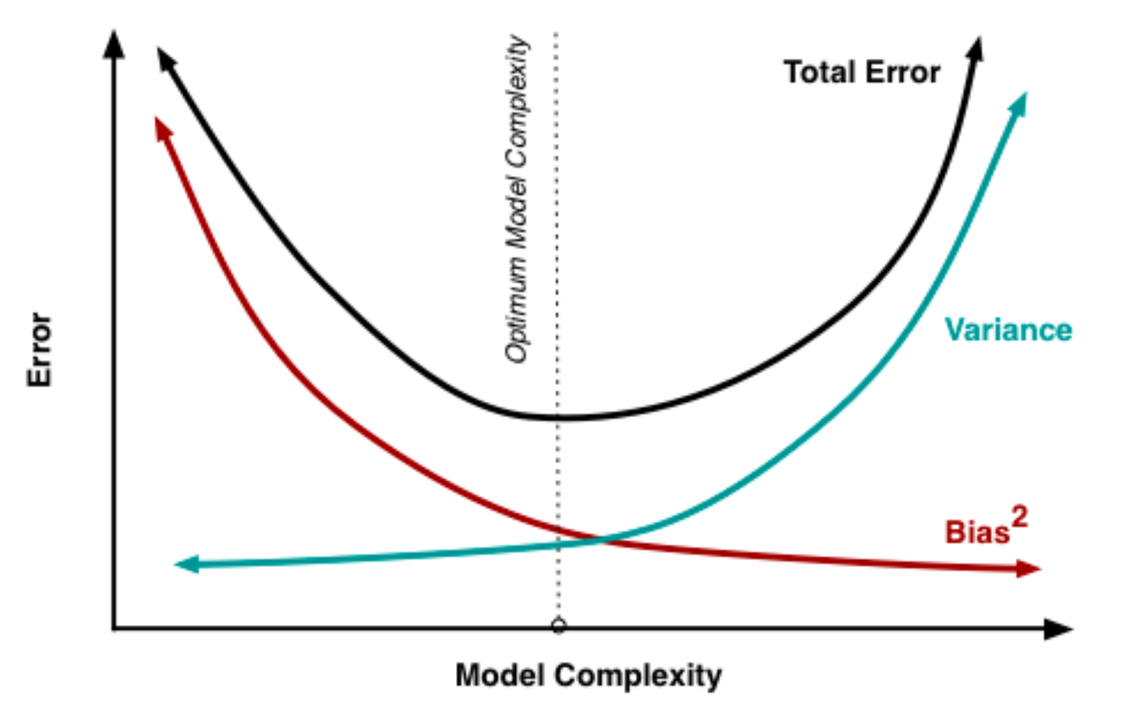
Miss Rate - % of actual positive predicted wrongly?

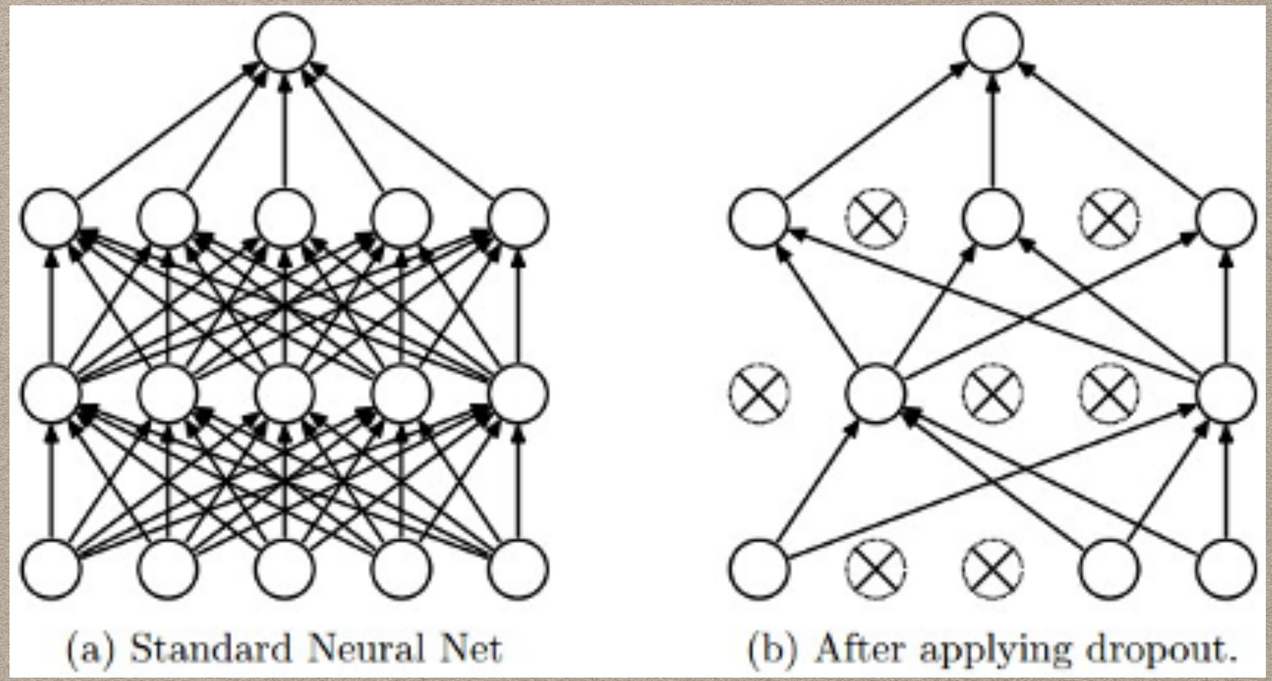
Fallout - % of actual negatives predicted wrongly?

Specificity -% of actual negatives recalled correctly?

F1 Score mean of precision & recall

BIAS VARIANCE TRADEOFF





(a) Standard Neural Net

(b) After applying dropout.

BATCH NORMALISATION IS GENRALLY PREFERRED

- Improves gradient flow through the network
- Allows higher learning rates
- Reduces the strong dependence on initialization
- Acts as a form of regularization in a funny way, and slightly reduces the need for dropout, maybe

ERRORS AND BEST PRACTICES IN DL

TIME TO HIT THE LAB

QUESTIONS PLEASE!

github.com/vverdhan

Thanks