ANN
COURSE 2 - WEAK 1 SUMMARY

By: Mohamed Ahmed Nassar

* In Machine learning The model initially fit on training dataset, and successively the fitted model is used to predict the response of the observation on validation dataset. But this make validation set somehow is biased, so we used **Test dataset** to test the general behavior of the model on completely unseen data after finishing model training .

* Bias means the model can't learn from the data a that happened for many reason,
    1- Network needs to be larger (increase number of parameters to learn much more features)
    2- Data is imbalanced (on class appears much more then other)
    3- Data is noisy and model can't extract useful information from it

* We can identify Bias by tracking the training & testing accuracy if both are low, that's mean the network is has a high bias

* Variance means model is over-fitting the data, so the model tracks every single input with expected output which means the model isn't generalizing, it just save the distribution of the training.

* We can identify high variance by tracking training and testing accuracy, if the accuracy increases on training while deceasing on testing that's mean the network has a high variance.

*- Regularization is used to decrease over fitting and improve network generalization

* There are many types of regularization like L1, L2 and dropout

* L1 and L2 Manly used to force the network to keep weights as small as possible by adding the average weight value to the cost function so, as Weights increases >> cost function increases >> GD will try to minimize cost by minimizing the weight values.

* minimizing the weight value deceases the chance of over fitting because it's prevent some weights from getting larger while ignoring the reset.

* Large weights means the contribute in the decision by large value, which make its decision most important so by keeping all weights as small as possible they all will have the same contribution in output which improve generalization,

* Decreasing weights is equivalent to reducing the order of the network, a higher order network means a very complex function mapping which that can track every single point even noisy ones so network will not generalize

* Dropout regularization is another approach also helps in increasing generalization and decreasing over-fitting

* Drop-out rely on a simple concept which is " Let each neuron try hard to learn the whole problem it self without being dependent on another neuron output" so by switching off random neurons every iteration the reset of neurons will try to reach to the output and learn as much as possible to produce a correct output which increases the overall performance

* the best way to improve generalization in decrease the over fitting is by increasing the data, which make network trying to learn the most common feature between all training example which is the most useful features to generalize
* You can increase the data by using augmentation or combine different dataset with different distribution.

* normalizing inputs have a large effect on increasing training speed as it make the data circular around the origin, which means we don't need a very small learning rate to avoid oscillation in one dimension

* normalizing inputs also git us rid of a very large input values which may saturate the non-linear activation which kill the gradient over time and decrease the learning speed

* Vanishing gradient occurs usually due to non-linear function saturation, for that reason the ReLu activation is preferable with normalizing the input and regularize the weights.

* Exploding Gradients occurs due to multiplying gradient and accumulate them through deep-network which increase the value of gradient which in turn increase the weight update value and drive network to instability.

* initialize network with a very large random values drive network gradients to explode or vanish dependent on activation function type

* To solve exploding gradient we may redesign a new network with less number of layers, or initialize weights with a more careful technique or using gradient clipping technique which clip the gradient value if it exceeded a certain limit.

* The most usable rule of thumb to initialize the weights is "He" weight initialization method, which states that weights should be initialized by random number that has a mean of 0 and
std = sqrt(2/L[-1]) where L[-1] = number of neurons in previous layer