Q4] Describe learning by gradient descent in general

Ans ⇒ Gradient descent is by far the most popular optimization strategy used by machine learning & deep learning.

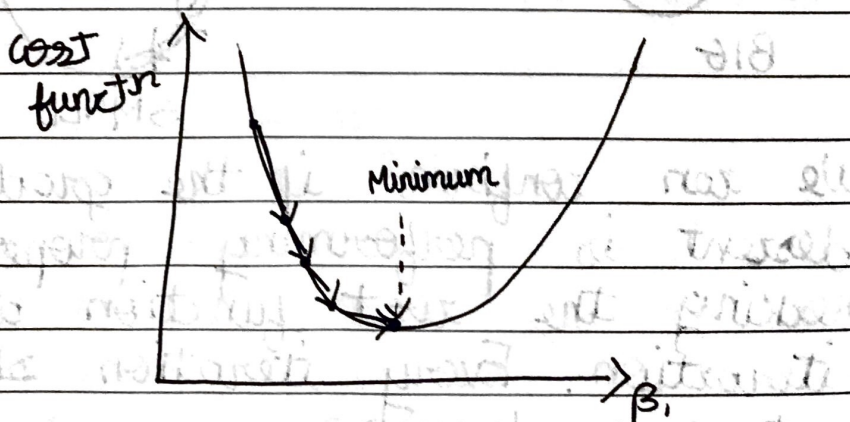
⇒ Gradient descent is used while training a machine learning model. It is based on a convex function, that tweaks the parameters iteratively to minimise a given function to its local minimum.

⇒ A gradient is used to measure how the output of a function changes if you change the inputs a little bit.

⇒ Gradient descent is a 2 step process:-

1] Given the gradient, calculate the change in parameter with respect to the size of step taken, with

2] The new value of parameter, calculate new gradient. Go back to step 1 until we hit the convergence.

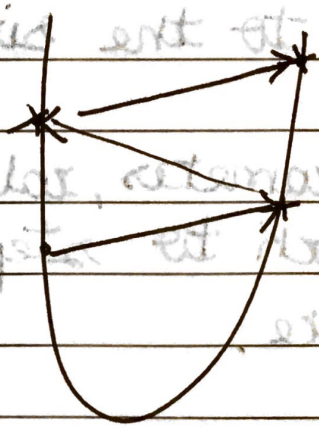


⇒ ~~If the~~ In order to reach local minimum. A parameter called learning rate plays a crucial role.

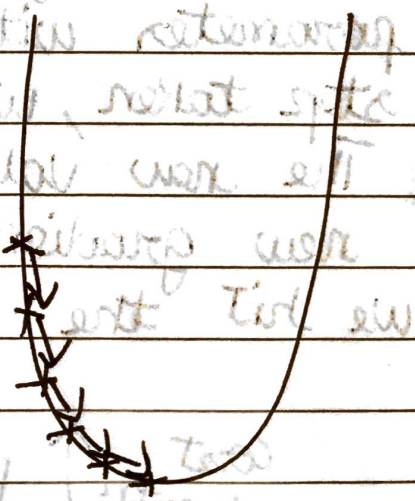
⇒ As it decides the steps that gradient descent takes into the direction of local minimum.

⇒ The learning rate should be set at a appropriate value, that should neither be too low or big.

⇒ If the steps taken are big, it may never reach local minimum because it will just bounce back & forth between the convex function. While if we set the learning rate to a very small value, then the gradient descent will reach local minimum but the time taken will be increased drastically.



BIG



SMALL

⇒ We can confirm if the gradient descent is performing properly by checking the cost function at every iteration. Every iteration should capture a decrease

MLP = Feed foorword ANN = ANN

Q] Describe backpropogation algorithm for MLP/ANN and explain how it relates to gradient descent

Ans ⇒ Backpropogation is considered to be one of the simplest and most general method used for supervised training of multi-layered neural networks.

⇒ Back propogation works by approximating the non-linear relationship between the i/p and output by adjusting the weights value internally.

⇒ The backpropogation can be divided into 2 phases
- training
- testing

⇒ During the training phase the n/w is shown sample inputs & the correct classification

⇒ The operations of the back propogation neural network can be divided into 2 steps :-
- ⇒ Feed foorword
- ⇒ Back propogation.

⇒ In the first step, an i/p pattern is applied to the input layer and its effect propogates layer by layer, through the network, unit an output is produced.

⇒ The n/w actual value is then compared to the expected o/p.

DELTA RULE :-

$$\Delta w_{ji}^{(m)} = -\eta \delta_j^{(m)} x_i^{(m)}$$

⇒ An error signal is computed for each of the output nodes. Since all the hidden nodes have to some degree contributed to the errors evident in the output layer, the o/p error signals are transmitted backwards from the o/p layer to each node in hidden layer that immediately contributed to the output layer.

⇒ Once the error signal for each node has been determined, the errors are then used by the nodes to update the values for each connection weights until the n/w converges to a state that allows all the training patterns to be encoded.

⇒ The backpropagation algorithm looks for the minimum value of the error function in weight space using a technique called as the delta rule or gradient descent.

⇒ The weights that minimize the error function is then considered to be a solution to the learning problem.

⇒ To conclude, Backpropagation is the algorithm that is used to calculate the gradient of the loss function with respect to parameters. While gradient descent is the optimisation algorithm that is used to find parameters that minimise the loss function

Q] Discuss the scope of the backpropogation algorithm, including the type of connection's models it may be applied to, its complexity and sample of problems it may encounter.

Ans ⇒ Backpropogation works with any direct cyclic graph having continous units & no binary thresholds.

⇒ It accepts loops with time delays.

⇒ The performance for back-propogation is very efficient $O(|w|)$.

⇒ It is also capable of handling large networks i.e $10^4 - 10^6$ weights have been reported in many real world applications

⇒ Also, if 1 hidden layer is enough to represent any function that doesn't mean that it is the most efficient configuration

⇒ It is possible that with multiple hidden layers more complex n/ws can be represented using fewer weights.

⇒ The loading problem for backpropogation is NP complete i.e the problem can be solved by using brute force search algorithm

⇒ But in protical, the n/w can be trained in a reasonable amount of time.

⇒ Often local minima is not a big issue. But some techniques can be applied to speed up gradient descent.

Q] Drawbacks of gradient descent & 2 alternatives / techniques to overcome gradient descent problems

Ans.

Following are the disadvantages of GD:-

⇒ Converging to a local minimum can be quite slow

⇒ If there are multiple local minima, then there is no guarantee that the procedure will find the global minimum.

⇒ Requires careful selection of the parameter learning rate

⇒ In some cases, it might get close to the optimum but never converges

⇒ It cannot be reliably applied to non convex problems.

Two alternative techniques to overcome problem of gradient are

⇒ Stochastic GD :- When compared to BGD, SGD converges much faster compared to GD but error function is not as well minimised when compared to BGD. But often the close approximation by SGD are enough to reach optimal value.

* It is faster than BGD because we use only one training sample rather than using all samples
* SGD starts improving itself from the first example.

⇒ In this variation, the gradient descent procedure is run but the update to the w-efficients is performed for each training instance, rather than at the end of the batch.

⇒ The update procedure for the w-efficients is the same as that above, except the cost is not summed over all training patterns but calculated for one training pattern.

⇒ Thus learning can be much faster with stochastic gradient for very large training datasets. & often we require only small number of passes through the dataset to reach a good or good enough set of w-efficients.

Batch Gradient :- The concept of carrying out GD is the same as stochastic gradient descent but only the w-efficients are updated in the batch.

Advantage :- ⇒ Less oscillation and noisy steps taken towards the global minima of the loss function

⇒ Can benefit from vectorization which increases the speed of processing

⇒ It produces a more stable gradient descent convergence and stable errors than Stochastic GD.

⇒ Computationally efficient.

Disadvantage :-

⇒ Sometimes a stable errors gradient can lead to local minima

⇒ The entire training set can be too large to process in the memory due to which additions might be required.