1) explain activation function in your own

Language.

a.sigmoid

b.tanh

C.ReLU

D.ELU

E.leakyReLU

F.swish.

Ans :

The activation function defines the output of a neuron node given an input or set of input (output of multiple neurons). It's the mimic of the stimulation of a biological neuron.

A.sigmoid : A sigmoid unit in a neural network. When the activation function for a neuron is a sigmoid function it is a guarantee that the output of this unit will always be between 0 and 1. Also, as the sigmoid is a non-linear function, the output of this unit would be a non-linear function of the weighted sum of inputs

B.tanh : Tanh Activation is an activation function used for neural networks: f ( x ) = e x − e − x e x + e − x. Historically, the tanh function became preferred over the sigmoid function as it gave better performance for multi-layer neural networks.

C.ReLU & ELU : ELU is very similar to ReLU except negative inputs. They are both in identity function form for non-negative inputs. On the other hand, ELU becomes smooth slowly until its output equal to −α whereas RELU sharply smoothes. Pros. ELU becomes smooth slowly until its output equal to −α whereas RELU sharply smoothes.

D.leakyReLU :Leaky Rectified Linear Unit, or Leaky ReLU, is a type of activation function based on a ReLU, but it has a small slope for negative values instead of a flat slope. The slope coefficient is determined before training.

E.swish : Swish is an activation function, f ( x ) = x ⋅ sigmoid ( β x ) , where a learnable parameter. Nearly all implementations do not use the learnable parameter , in which case the activation function is x σ ( x ) ("Swish-1").

2)what happen when you increase or decrease the Optimizer learning rate.?

Ans : The optimization problem addressed by stochastic gradient descent for neural networks is challenging and the space of solutions (sets of weights) may be comprised of many good solutions (called global optima) as well as easy to find, but low in skill solutions (called local optima).

The amount of change to the model during each step of this search process, or the step size, is called the “learning rate” and provides perhaps the most important hyperparameter to tune for your neural network in order to achieve good performance on your problem.

3)what happen when you increase the number of internal hidden neurons ?

Ans : Increasing the number of hidden layers might improve the accuracy or might not, it really depends on the complexity of the problem that you are trying to solve. Where in the left picture they try to fit a linear function to the data.

4)what happen when you increase the size of batch computation.?

Ans :

larger batch sizes make larger gradient steps than smaller batch sizes for the same number of samples seen. for the same average Euclidean norm distance from the initial weights of the model, larger batch sizes have larger variance in the distance.

5)why we adopt regularization to avoid overfitting?

Ans : Regularization refers to techniques that are used to calibrate machine learning models in order to minimize the adjusted loss function and prevent overfitting or underfitting. Using Regularization, we can fit our machine learning model appropriately on a given test set and hence reduce the errors in it.One of the most powerful features to avoid/prevent overfitting is cross-validation. The idea behind this is to use the initial training data to generate mini train-test-splits, and then use these splits to tune your model. In a standard k-fold validation, the data is partitioned into k-subsets also known as folds.

6)what do you mean by underfitting in neural networks ?

Ans : Your model is underfitting the training data when the model performs poorly on the training data. This is because the model is unable to capture the relationship between the input examples (often called X) and the target values (often called Y).

Your model may be underfitting simply because it is not complex enough to capture patterns in the data. Using a more complex model, for instance by switching from a linear to a non-linear model or by adding hidden layers to your neural network, will very often help solve underfitting.

7)what are lost & cost Functions in deep learning ?

Ans : In mathematical optimization and decision theory, a loss function or cost function (sometimes also called an error function) is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event.

8)why we use dropout in neural network?

Ans : By using dropout, in every iteration, you will work on a smaller neural network than the previous one and therefore, it approaches regularization. Dropout helps in shrinking the squared norm of the weights and this tends to a reduction in overfitting.