1. Explain the Activation Functions in your own language

1. sigmoid
2. tanh
3. ReLU
4. ELU
5. LeakyReLU
6. swish

2. What happens when you increase or decrease the optimizer learning rate?

3. What happens when you increase the number of internal hidden neurons?

4. What happens when you increase the size of batch computation?

5. Why we adopt regularization to avoid overfitting?

6. What are loss and cost functions in deep learning?

7. What do ou mean by underfitting in neural networks?

8. Why we use Dropout in Neural Networks?

Answer:

1. Activation Functions are mathematical functions applied to the output of a neuron in a neural network to introduce non-linearity into the model. Some of the commonly used activation functions are:

a) Sigmoid: It is an S-shaped function that maps any input value to a range of 0 to 1. It is commonly used in binary classification problems. b) Tanh: It is a hyperbolic tangent function that maps any input value to a range of -1 to 1. It is commonly used in binary classification problems and can be used in the hidden layers of the network. c) ReLU: It is a Rectified Linear Unit function that maps any input value less than zero to zero and maintains any value greater than zero. It is the most commonly used activation function in deep learning as it provides faster convergence and is computationally efficient. d) ELU: It is an Exponential Linear Unit function that is similar to the ReLU function, but with a more gradual curve when the input is negative. It helps reduce the vanishing gradient problem and is useful for deep neural networks. e) LeakyReLU: It is a variation of ReLU that allows for a small, non-zero output for negative input values, which helps to mitigate the dying ReLU problem. f) Swish: It is a newer activation function that is similar to the ReLU function but with a smoother curve, which helps to produce better results in some cases.

1. The learning rate of an optimizer determines the step size of the optimization algorithm towards the optimal solution. Increasing the learning rate can help the model converge faster, but if it is too high, the optimizer may overshoot the optimal solution and diverge. On the other hand, decreasing the learning rate may help the optimizer to reach the optimal solution, but the model may take longer to converge.
2. Increasing the number of internal hidden neurons can increase the capacity of the model, which can lead to better performance and the ability to learn more complex patterns. However, if the number of neurons is too high, the model may overfit the training data and perform poorly on new data.
3. Increasing the size of batch computation can help to increase the efficiency of the training process, as it allows the model to process multiple examples at once. However, if the batch size is too large, the model may have difficulty converging, and the training process may become unstable.
4. Regularization techniques, such as L1 and L2 regularization, are used to prevent overfitting in neural networks. Overfitting occurs when the model learns the noise in the training data, which leads to poor performance on new data. Regularization helps to limit the complexity of the model, and penalizes large weights in the network, which helps to prevent overfitting.
5. Loss functions and cost functions are used in deep learning to evaluate the performance of a model. The loss function measures the error between the predicted output and the actual output, and the goal is to minimize the loss function during training. The cost function is the average of the loss function over all training examples.
6. Underfitting occurs when the model is not able to capture the patterns in the data, leading to poor performance on both the training and test data. This can occur when the model is too simple, or if the training data is not representative of the underlying patterns in the data.
7. Dropout is a regularization technique that randomly drops out some of the neurons during training to prevent overfitting. This helps to reduce the reliance of the model on any particular set of neurons, and encourages the model to learn more robust features.