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?**

Increasing or decreasing the learning rate in a neural network can have a significant effect on the performance of the model. In general, a higher learning rate can help the model learn faster, while a lower learning rate can help the model learn more accurately.

When you increase the learning rate, the optimizer will make larger updates to the model parameters with each iteration, allowing the model to learn more quickly. This can be useful if the model is not learning fast enough, but it can also cause the model to converge to a suboptimal solution or even diverge entirely if the learning rate is set too high.

On the other hand, decreasing the learning rate can help the model learn more accurately by making smaller updates to the model parameters with each iteration. This can help the model avoid overshooting the optimal solution and can also improve the model's ability to generalize to new data. However, a lower learning rate can also slow down the learning process, so it's important to find a balance that works well for your specific problem.

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

Increasing the number of internal hidden neurons in a neural network can improve the model's ability to learn complex patterns in the data. Hidden neurons are the "internal" layers of a neural network, between the input and output layers. These neurons take in inputs from the previous layer, perform some computation on them, and pass them on to the next layer.

By increasing the number of hidden neurons, you are increasing the network's capacity to learn complex patterns. This can improve the model's performance on the training data and can also improve its ability to generalize to new data. However, it's important to strike a balance between having enough hidden neurons to capture the relevant patterns in the data, but not so many that the model overfits to the training data and becomes too complex.

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

In general, increasing the size of the batch in a neural network can speed up the computation and training of the model. Batch size refers to the number of samples from the training data that are processed at once by the model.

When you increase the batch size, the model is able to process more data in each iteration, which can speed up the training process. This is because larger batch sizes allow the model to make more efficient use of the computational resources available, such as the CPU or GPU.

5. **Why we adopt regularization to avoid overfitting?**

Regularization is a technique that is used in machine learning to prevent overfitting of the model to the training data. Overfitting occurs when a model learns the noise or random fluctuations in the training data, rather than the underlying relationship. This can cause the model to perform poorly on new, unseen data.

Regularization helps to avoid overfitting by introducing additional constraints on the model parameters, such as limiting their magnitude or forcing them to take on specific values. This can prevent the model from learning noise in the training data and can help it to generalize better to new data.

There are several different types of regularization methods, including L1 and L2 regularization, which add a penalty on the magnitude of the model parameters, and dropout regularization, which randomly drops out neurons from the network during training. These methods can help to improve the performance of the model on new data and can also make the model more interpretable.

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

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

Underfitting in a neural network occurs when the network is not able to capture the underlying trend in the data. This can happen when the network is not complex enough, or when it is not trained for enough time. Underfitting can lead to poor performance on the training data, and it can also cause the network to perform poorly on unseen data. In general, underfitting is a problem that can be addressed by increasing the complexity of the network, or by training the network for longer to allow it to learn the underlying patterns in the data.

8. **Why we use Dropout in Neural Networks?**

Dropout is a regularization technique for neural networks. It helps to prevent overfitting by randomly dropping out, or setting to zero, a certain number of activations in a layer during training. This has the effect of making the model more robust, and also reduces the number of trainable parameters, which can help to prevent overfitting. Dropout can be applied to any layer in a neural network, and it is typically used in conjunction with other regularization techniques, such as weight decay and early stopping.