1. Explain the Activation Functions in your own language

a) sigmoid

b) tanh

c) ReLU

d) ELU

e) LeakyReLU

f) swish

**a) Sigmoid: The sigmoid activation function squashes input values to a range between 0 and 1. It's often used in binary classification problems as it can model probabilities.**

**b) tanh (Hyperbolic Tangent): Tanh is similar to the sigmoid but squashes input values to a range between -1 and 1. It's often used in hidden layers of neural networks.**

**c) ReLU (Rectified Linear Unit): ReLU is a popular activation function that outputs the input if it's positive and zero otherwise. It helps introduce non-linearity to the model and addresses the vanishing gradient problem.**

**d) ELU (Exponential Linear Unit): ELU is similar to ReLU but with a smooth curve for negative values. It can mitigate the dying ReLU problem and is robust against noisy data.**

**e) LeakyReLU: LeakyReLU is a variant of ReLU that allows a small gradient for negative values, preventing neurons from becoming inactive. It can help address some of the limitations of ReLU.**

**f) Swish: Swish is a newer activation function that has been found to perform well in some deep learning tasks. It's a smooth function that combines elements of ReLU and sigmoid.**

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

**Increasing the learning rate can make the optimization process converge faster but may lead to overshooting the optimal solution or diverging.**

**Decreasing the learning rate makes the optimization process more stable and accurate but can slow down convergence. It's a trade-off between speed and accuracy.**

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

**Increasing the number of hidden neurons can increase the model's capacity to learn complex patterns from the data.**

**However, it can also lead to overfitting if the network becomes too complex for the available data.**

**It may require more data to train effectively.**

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

**Increasing the batch size can speed up training as more data is processed in parallel, utilizing GPU capabilities efficiently.**

**Smaller batches can lead to noisier gradient estimates but might generalize better.**

**The optimal batch size depends on the specific problem and available resources.**

5. Why we adopt regularization to avoid overfitting?

**Regularization techniques (e.g., L1, L2 regularization, dropout) introduce constraints on the model parameters, preventing them from fitting the training data too closely.**

**Overfitting occurs when a model learns to memorize the training data rather than generalize to unseen data.**

**Regularization helps improve the model's ability to generalize and reduce overfitting.**

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

**Loss and cost functions measure the error or discrepancy between the predicted values and the actual target values.**

**They quantify how well the model is performing during training.**

**Common loss functions include mean squared error (MSE), cross-entropy, and hinge loss.**

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

**Underfitting occurs when a model is too simple to capture the underlying patterns in the data.**

**The model fails to fit the training data adequately and performs poorly on both training and test data.**

**It indicates that the model's capacity is insufficient for the complexity of the problem.**

8. Why we use Dropout in Neural Networks?

**Dropout is a regularization technique where random neurons are temporarily dropped out (set to zero) during each training iteration.**

**It helps prevent co-adaptation of neurons and reduces overfitting.**

**Dropout encourages the network to learn more robust features and improves generalization.**