

Alireza Mohammadshafie-Machine learning 4th assignment- short answer questions.

1. Please briefly explain PCA and NMF, and describe their similarities and differences.

Both PCA and NMF aim to reduce computation and simplify data by finding correlations. However, a key difference is that PCA can produce negative numbers, whereas NMF only generates non-negative values.

2. Please explain why the initialization process of weights and bias is important for neural networks. (1 point)

The initialization of weights and bias is crucial because it can affect how well the network learns. Proper initialization helps to prevent issues like vanishing or exploding gradients, allowing the model to converge more effectively during training.

3. How would you prevent overfitting when designing an artificial neural network? (1 point)

To prevent overfitting in neural networks, you can use techniques like dropout, where random neurons are turned off during training, and also apply weight regularization to penalize overly complex models. Additionally, using data augmentation or early stopping can help too.

4. Please describe the key differences between convolutional neural network and multilayer perceptron? (1 point)

The main difference is that convolutional neural networks use convolutional layers to process grid-like data (like images). In contrast, multilayer perceptrons (MLPs) connect every input to every neuron, treating inputs as flat vectors with no spatial relation.

5. What is the “vanishing gradient” problem with neural networks based on sigmoid non-linearities? (1 point)

The vanishing gradient problem occurs in neural networks using sigmoid activation functions, where gradients become very small during backpropagation, making it difficult for the model to learn, especially in the earlier layers. This can slow down or even halt the training process.

Coding question summary part:

I worked on my personal laptop instead of using Google Colab. For the modifications, I added an extra layer to the model and adjusted the learning rate to **0.002**. I also increased the number of epochs to **20** and switched the activation function from **tanh** to **ReLU**. Additionally, I changed the structure of the model by increasing the number of nodes in any layers. I used the **Adam** optimizer and applied the **Dropout** technique for regularization with a probability of **0.2** for the classifier layers. I used augmentation technique too.

The accuracy I could obtain for the test is: 77.98%