**Assignment - 04**

1. What is the concept of cyclical momentum?

Ans: Concept of cyclical momentum:

Cyclical momentum is a technique used in optimization algorithms, particularly in gradient descent-based optimization methods such as SGD (Stochastic Gradient Descent) with momentum.

It involves dynamically adjusting the momentum parameter during training, where the momentum value varies in a cyclical manner over epochs.

The purpose of cyclical momentum is to introduce variation in the momentum term, which can help accelerate convergence and prevent the optimization process from getting stuck in local minima or saddle points.

1. What callback keeps track of hyperparameter values (along with other data) during training?

Ans: Callback that keeps track of hyperparameter values during training:

The callback that keeps track of hyperparameter values (along with other data) during training is typically a TensorBoard callback.

TensorBoard is a visualization toolkit provided by TensorFlow that logs various metrics, including hyperparameters, loss, accuracy, and other data, during the training process.

This callback allows users to monitor and analyze the training progress using interactive visualizations.

3. In the color dim plot, what does one column of pixels represent?

Ans: Representation of one column of pixels in color dim plot:

In the color dim plot, one column of pixels represents the intensity values of a specific color channel (e.g., red, green, or blue) across all rows of the image.

Each pixel in the column corresponds to the intensity value of the corresponding row and the specified color channel.

4. In color dim, what does "poor teaching" look like? What is the reason for this?

Ans: "Poor teaching" in color dim and its reason:

"Poor teaching" in color dim refers to ineffective learning or slow convergence during the training process.

It may manifest as slow changes or lack of improvement in the pixel intensity values across different color channels over epochs.

This could occur due to issues such as vanishing gradients, inadequate model capacity, or improper hyperparameter settings.

5. Does a batch normalization layer have any trainable parameters?

Ans: Trainable parameters in batch normalization layer:

Yes, a batch normalization layer has trainable parameters.

It includes trainable parameters such as scale (gamma) and shift (beta) parameters, which are learned during training to scale and shift the normalized activations.

6. In batch normalization during preparation, what statistics are used to normalize? What about during the validation process?

Ans: Statistics used for normalization in batch normalization:

During training, batch normalization uses batch statistics (mean and variance) calculated over each mini-batch to normalize the activations.

During the validation process or inference, running statistics (moving averages) computed during training are used for normalization instead of batch statistics.

7. Why do batch normalization layers help models generalize better?

Ans: Benefits of batch normalization layers for model generalization:

Batch normalization layers help models generalize better by reducing internal covariate shift, making optimization more stable and accelerating convergence.

They normalize activations, making them less sensitive to changes in parameter initialization and learning rates, which can lead to improved generalization performance.

8.Explain between MAX POOLING and AVERAGE POOLING is number eight.

Ans: Difference between MAX POOLING and AVERAGE POOLING:

Max pooling retains the maximum value within each pooling region, resulting in sharper feature maps and preserving strong activation responses.

Average pooling calculates the average value within each pooling region, resulting in smoother feature maps and averaging out noise or small variations.

9. What is the purpose of the POOLING LAYER?

Ans: Purpose of the POOLING LAYER:

The pooling layer reduces the spatial dimensions (width and height) of the input feature maps while preserving important information.

It helps in controlling overfitting, reducing computational complexity, and increasing the receptive field of the network.

10. Why do we end up with Completely CONNECTED LAYERS?

Ans: Reason for Completely CONNECTED LAYERS at the end:

Fully connected layers (also known as dense layers) at the end of the neural network integrate features learned by previous layers and perform classification or regression tasks.

They provide a global view of the features and their relationships across the entire input, enabling the network to make final predictions based on learned representations.

11. What do you mean by PARAMETERS?

Ans: Meaning of PARAMETERS:

Parameters in a neural network refer to the weights and biases associated with each neuron or connection between neurons.

These parameters are learned during the training process through optimization algorithms such as gradient descent.

12. What formulas are used to measure these PARAMETERS?

Ans: Formulas used to measure PARAMETERS:

Parameters in a neural network are typically measured by the number of weights and biases.

For a fully connected layer, the number of parameters can be calculated as (number of input units) \* (number of output units) + (number of output units) for weights and biases, respectively.