**Convolution Neural Networks (CNN)**

**Instructions:**

Please share your answers filled in-line in the word document. Submit code separately wherever applicable.

Please ensure you update all the details:

**Name: K Sai Manoj Batch ID:** 29072024 10 AM

**Topic: Convolutional Neural Networks**

**Guidelines:**

**1. An assignment submission is considered complete only when the correct and executable code(s) and documentation explaining the method and results are submitted. Failing to submit either of those will be considered an invalid submission and not a correct submission.**

**2. Ensure that you submit your assignments correctly and in full. Resubmission is not allowed.**

**3. Post the submission you can evaluate your work by referring to the keys provided. (will be available only post the submission).**

**Hints:**

1. **Business Problem**
   1. **What is the business objective?**
   2. **Are there any constraints?**
2. **Work on each feature of the dataset to create a data dictionary as displayed in the below image:**

**Make a table as shown above and provide information about the features such as its data type and its relevance to the model building. And if not relevant, provide reasons and a description of the feature.**

1. **Data Pre-processing**

**3.1 Data Cleaning, Feature Engineering, etc.**

**3.2 Outlier treatment if applicable.**

1. **Model Building**
   1. **Build a convolution neural network model.**
   2. **Train and test the model.**
   3. **Briefly explain the model output in the documentation.**
2. **Write about the benefits/impact of the solution - in what way does the business (client) benefit from the solution provided?**
3. **Use Tensorflow for this assignment. Depending on your system configuration, either Tensorflow GPU or Tensorflow CPU versions.**

**Problem Statement: -**

1. Build a CNN model on the CIFAR-10 dataset by applying a few regularization techniques like dropout and data augmentation. Download the data set from a library called tensorflow. Finally do the deployment on streamlit application
2. Find out the differences between the Convnet filter and the Maxpool layers.

**Ans:**

ConvNet (Convolutional Neural Network) filters and MaxPool layers are two fundamental components in CNNs, serving distinct purposes:

ConvNet Filter (Convolutional Layer)

1. Applies a learnable filter (kernel) to input data.

2. Performs convolution operation, scanning the input data spatially.

3. Generates feature maps, highlighting local patterns and features.

4. Typically uses a small, learnable filter (e.g., 3x3, 5x5).

5. Output: Feature maps with spatial hierarchy.

MaxPool Layer

1. Reduces spatial dimensions of feature maps.

2. Retains maximum value within a specified window (pool size).

3. Downsamples feature maps, reducing spatial resolution.

4. Typically uses a fixed window size (e.g., 2x2, 3x3).

5. Output: Reduced feature maps with retained important features.

Key differences:

1. Purpose:

- ConvNet filter: Extracts local features.

- MaxPool layer: Reduces spatial dimensions.

2. Operation:

- ConvNet filter: Convolution.

- MaxPool layer: Max value extraction.

3. Output:

- ConvNet filter: Feature maps.

- MaxPool layer: Reduced feature maps.

4. Filter size:

- ConvNet filter: Small, learnable.

- MaxPool layer: Fixed window size.

5. Effect on spatial dimensions:

- ConvNet filter: Preserves spatial dimensions.

- MaxPool layer: Reduces spatial dimensions.

ConvNet filters and MaxPool layers work together:

1. ConvNet filters extract local features.

2. MaxPool layers reduce spatial dimensions, retaining important features.

3. This process repeats, forming a hierarchical representation.

1. **If the input of an image is 64x64x3 which has been convolved by 10 5x5 filters with stride 1 and padding 2:**

**Ans:**

- Input image: 64x64x3

- Convolutional layer:

- Number of filters (kernels): 10

- Filter size: 5x5

- Stride: 1

- Padding: 2

1. **How many activation maps are obtained?**

The number of activation maps is equal to the number of filters. Therefore, there are 10 activation maps.

1. **What is the size of the activation maps?**

To calculate the output size, use the formula:

Output size = (Input size + 2 × Padding - Filter size) / Stride + 1

For each dimension (height and width):

Output height = (64 + 2 × 2 - 5) / 1 + 1 = 64

Output width = (64 + 2 × 2 - 5) / 1 + 1 = 64

So, the size of each activation map is 64x64.

1. **How many parameters are calculated?**

The number of parameters in a convolutional layer includes:

1. Weights: Filter size × Number of filters × Input channels

= 5 × 5 × 10 × 3 = 750

2. Biases: Number of filters

= 10

Total parameters = Weights + Biases

= 750 + 10

= 760

Therefore:

- 10 activation maps are obtained.

- Size of each activation map: 64x64.

- 760 parameters are calculated.

1. **What are the different techniques that need to be applied to overcome the issue of overfitting? Provide brief explanations of how these techniques address the issue.**

Here are techniques specific to Convolutional Neural Networks (CNNs) to overcome overfitting:

**Regularization Techniques**

1. Dropout: Randomly drops neurons during training.

2. Weight Decay (L2 Regularization): Penalizes large weights.

3. L1 Regularization: Penalizes non-zero weights.

**Data Augmentation**

1. Random Cropping: Crops input images randomly.

2. Random Flipping: Flips input images horizontally/vertically.

3. Color Jittering: Changes image brightness, contrast, saturation.

4. Rotation: Rotates input images.

**Architecture Modifications**

1. Batch Normalization: Normalizes inputs to each layer.

2. Depthwise Separable Convolutions: Reduces parameters.

3. Dilated Convolutions: Increases receptive field.

**Training Techniques**

1. Early Stopping: Stops training when validation accuracy plateaus.

2. Learning Rate Scheduling: Adjusts learning rate during training.

3. Transfer Learning: Leverages pre-trained models.

**Other Techniques**

1. Data Preprocessing: Removes noise, normalizes inputs.

2. Ensemble Methods: Combines predictions from multiple models.

3. Attention Mechanisms: Focuses on relevant regions.

**Specific CNN Architectures**

1. Residual Connections (ResNets): Eases training deep networks.

2. Inception Modules: Increases depth/width efficiency.

3. MobileNet: Uses depthwise separable convolutions.

By applying these techniques, you can reduce overfitting and improve your CNN's performance.

**Some popular CNN architectures that inherently address overfitting include:**

1. ResNet

2. Inception

3. MobileNet

4. ShuffleNet

5. DenseNet

These architectures incorporate techniques like residual connections, depthwise separable **convolutions, and batch normalization to mitigate overfitting.**