Laboratory 7

Training a Deep Neural Network on CIFAR Data

Due Date: Beginning of Week 9 Lab

Concepts:

- Convolution Deep Neural Network
- Image classification on CIFAR dataset

Total Points: 100 points

Objectives:

Convolutional neural networks (CNNs) are one of the most popular algorithms for deep learning. They are a category of neural networks that have been proven to be very effective in the areas of image recognition and classification. Students will implement two different deep CNN models to conduct image classification for the CIFAR dataset.

Files Needed:

- Lab7_Part1.mlx
- Lab7_Part2.mlx
- CIFAR_3.mat

Assignment: Implement Convolution Neural Networks on CIFAR data

Part1: Convolution Neural Network Model 1

In Lab7, you will develop an image classification model by training a Convolutional Neural Network on CIFAR dataset using Matlab Neural Network toolbox. The provided data set, "CIFAR_3.mat" is a subset of the CIFAR-10 dataset and contains 18,000 32x32 color images in 3 classes {'airplane', 'automobile', 'horse'}.

1) Load the CIFAR Dataset

Load the CIFAR training and test data from the provided file "CIFAR_3.mat". Each image is a 32x32 RGB image and there are 15,000 training samples and 3,000 test samples. The training images are saved in the variable name, 'XTrain' and the corresponding categorical labels are saved as 'YTrain'. Similarly, the test images and the corresponding labels are saved in the variables, 'XTest' and ' YTest', respectively. Each of the color images is stacked in a 4D matrices (image height, image width, channel, sample index) in the variables, "XTrain" and "XTest" as shown below:

```
Name            Size                     Bytes  Class         Attributes

XTest           32x32x3x3000              9216000  uint8
XTrain          32x32x3x15000            46080000  uint8
YTest           3000x1                       3384  categorical
YTrain          15000x1                     15384  categorical
```

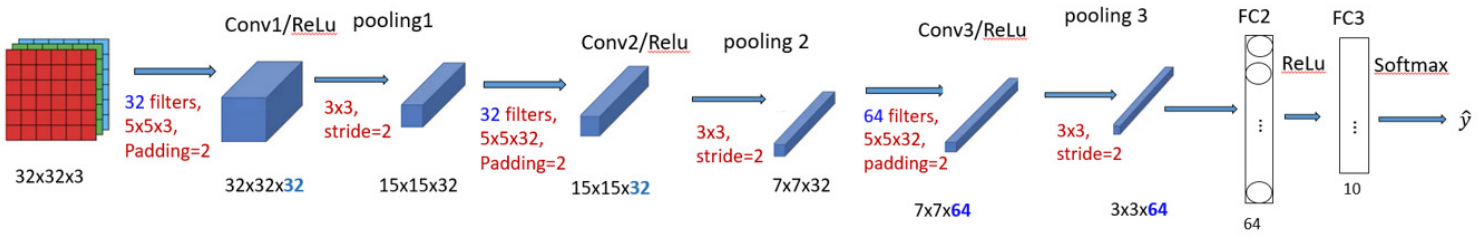
2) Visualize the CIFAR Dataset

As a first step, the first 100 training examples are visualized. They correspond to small images of three different classes. These images are best processed using a neural network architecture specific to images, for example, a convolutional neural network.

- Display the randomly selected 100 image samples from the training dataset. Generate 100 numbers in the range of [1, number of the training samples] using the **randperm**() function. Then, extract the image samples from the training data using the indexes generated by the **randperm**() function.

3) Create a Convolutional Neural Network (CNN)

Task1: Implement a convolutional neural network composed of 3 convolution layers, interleaved ReLU activation function and max-pooling layers. The architecture of the network is shown below:



A CNN is composed of a series of layers, where each layer defines a specific computation. The Neural Network Toolbox™ provides functionality to easily design a CNN layer-by-layer. In this example, the following layers are used to create a CNN:

- imageInputLayer - Image input layer
- convolution2dLayer - 2D convolution layer for Convolutional Neural Networks
- reluLayer - Rectified linear unit (ReLU) layer
- maxPooling2dLayer - Max pooling layer
- fullyConnectedLayer - Fully connected layer
- softmaxLayer - Softmax layer
- classificationLayer - Classification output layer for a neural network

- **imageInputLayer** : The network starts with an imageInputLayer. The input layer defines the type and size of data the CNN can process. In this example, the CNN is used to process CIFAR images, which are 32x32 RGB images.

- **Convolution Layers** :Feature Extraction
  - The middle layers are made up of repeated blocks of convolutional, ReLU (rectified linear units), and pooling layers. These 3 layers form the core building blocks of convolutional neural networks.
  - The convolutional layers define sets of filter weights, which are updated during network training.
  - The ReLU layer adds non-linearity to the network, which allow the network to approximate non-linear functions that map image pixels to the semantic content of the image.
  - The pooling layers downsample data as it flows through the network. In a network with lots of layers, pooling layers should be used sparingly to avoid downsampling the data too early in the network.
- **The final fully connected layers:** The final layers of a CNN are typically composed of fully connected layers and a softmax layer.

The overall CNN layers will created by combining the input, Convolution Layers, and final fully connected Layers.

4) Train CNN Using CIFAR-10 Data

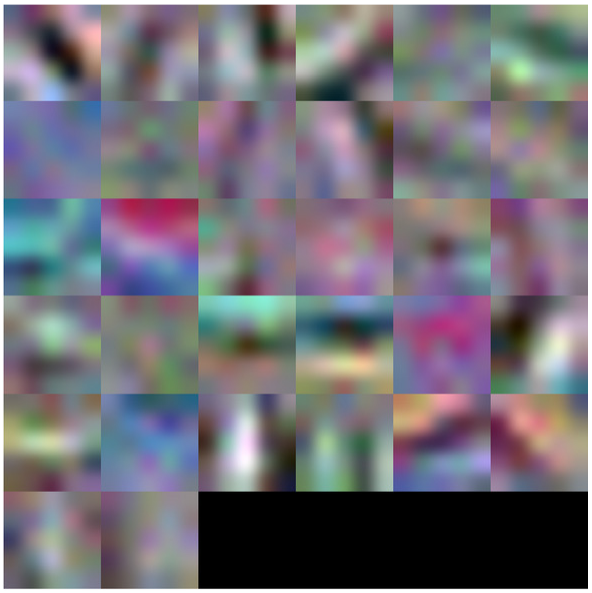
Now that the network architecture is defined, it can be trained using the CIFAR training data. First, set up the network training algorithm using the `trainingOptions` function. The network training algorithm uses Stochastic Gradient Descent with Momentum (SGDM) with an initial learning rate of 0.001. The training algorithm is run for 20 epochs.

Train the network using the trainNetwork function. This is a computationally intensive process that takes 10-30 minutes to complete.

5) Validate CIFAR Network Training

After the network is trained, it should be validated to ensure that training was successful. First, a quick visualization of the first convolutional layer's filter weights can help identify any immediate issues with training.

The first layer weights should have some well defined structure. If the weights still look random, then that is an indication that the network may require additional training. As shown in the below figure, the first layer filters have learned edge-like features from the CIFAR training data.

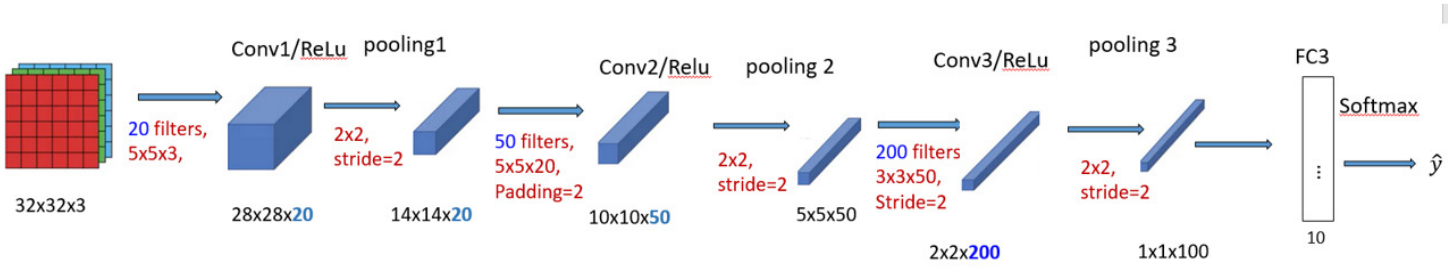


To completely validate the training results, use the CIFAR test data to measure the classification accuracy of the network. A low accuracy score indicates additional training or additional training data is required. The goal of this example is not necessarily to achieve 100% accuracy on the test set, but to sufficiently train a network for use in training an object detector.

Question1: What is the accuracy of the trained CNN model in Part 1?

Part2: Convolution Neural Network Model 2

Task2: Given the architecture of the network below, implement a convolutional neural network model #2 which is composed of 3 convolution layers, interleaved ReLU activation function and max-pooling layers. To complete Part2, you need to repeat the steps defined in Part 1 with this CNN architecture.



Question2: What is the accuracy of the trained CNN model in Part 2?

You can work in groups of up to two people

What to Submit to Blackboard:

- **one** ZIP file that includes :
  1) **A report (10pts): contains answers for Question1 –Question2.**
  2) (45 pts): The live script file, "Lab7_Part1.mlx", in that all outputs are generated either inline or on the right side.
  3) (45 pts): The live script file, "Lab7_Part2.mlx", in that all outputs are generated either inline or on the right side.

- **Pay attention to the zip file name convention:**
  Lab7_Student1 Lastname_Student2 Lastname.zip
  Ex) Lab7_Green_Smith.zip