PMDS603P Deep Learning Lab Experiment 7

September 2025

1 Work to do today

Note: Make a single PDF file of the work you are doing in a Jupyter notebook. Upload with the proper format. Please mention your name and roll no properly with the Experiment number on the first page of your submission.

Today we will look at how we can define a simple CNN model to fit MNIST dataset and prepare a classification model for the same. Our image dimension is $28 \times 28 \times 1$ since its a grey scale image.

Question 1: First try to fit a simple ANN model with atleast 2 hidden layers and get the accuracy of your model. So that you can compare how the CNN models you are going to fit next will work. Include early stopping in your model as well.
Question 2: Now let us try to fit a CNN model to accomplish the same task and look at the improvements. Import necessary modules and functions first. Here you can see we are including the Conv2D and MaxPool2d layers from keras.layers.

```
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Flatten, Conv2D, MaxPool2D
from keras.optimizers import RMSprop
from keras.callbacks import EarlyStopping
from sklearn.model_selection import train_test_split
```

load the mnist data and normalize the data and reshape. Now you can check the shape of xtrain and xtest etc which you are loading. we need to provide $28 \times 28 \times 1$ images to the the CNN so we need to reshape as (28,28,1) which is done below.

```
(xtrain, ytrain), (xtest, ytest) = mnist.load_data()

xtrain = xtrain.astype('float32')
xtest = xtest.astype('float32')

xtrain=xtrain/255
xtest=xtest/255

xtrain = xtrain.reshape(-1, 28, 28, 1)
xtest = xtest.reshape(-1, 28, 28, 1)
```

Now, create a CNN model as you see below. , and compile the model and fit the model

```
model = Sequential()
model.add(Conv2D(26, 5, strides=(1,1), activation='relu', padding='valid', input_shape=(28,28,1)))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2), padding='valid'))
model.add(Conv2D(20, 3, strides=(1,1), activation='relu', padding='valid'))
model.add(MaxPool2D(pool_size=(2,2), strides=(1,1), padding='valid'))
model.add(Conv2D(10, 3, strides=(1,1), activation='relu', padding='valid'))
model.add(MaxPool2D(pool_size=(2,2), strides=(1,1), padding='valid'))
model.add(Flatten())
model.add(Dense(100, activation='relu'))
model.add(Dense(10, activation='softmax'))
```

and check the performance. Also include early stopping in your model. Since we have a classification problem we have used the softmax activation in the final layer. Here Padding = valid means no padding is done. Also, if you want to do padding, then you can use padding = 'same' with a stride of 1. Here in the first convolution layer, we are using 26 kernels of size 5×5 with a stride of 1, both horizontal and vertical. Similar way further its given. Maxpooling layer

Question 3: Now you can try to fit a CNN model with CIFAR10 dataset we have seen in the previous lab with an appropriate model. See whether you get an improved accuracy for the model when you are using CNN models.