Deep Learning Assignment

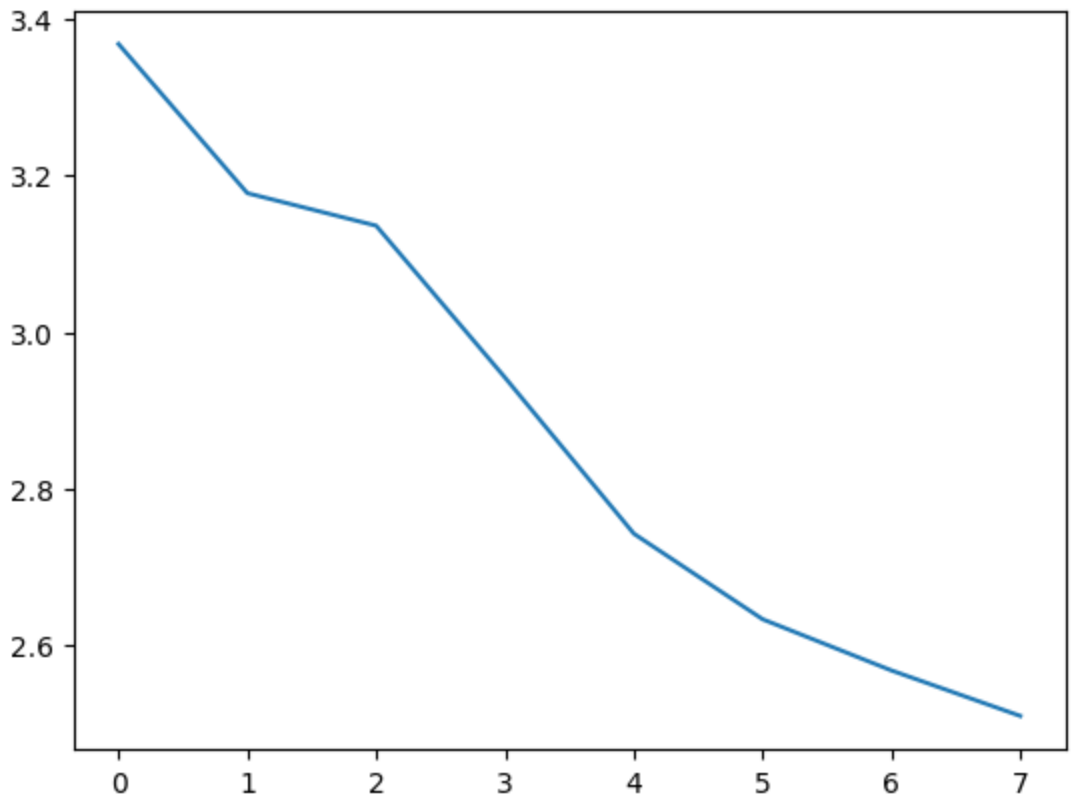
Part - 1
Vyom Goyal (2021101099)

- **Task :** Predict the sum of all the digits present in the image
- **Dataset :** The given dataset contains images with multiple digits and the sum of these digits as labels.
- **Baseline Model :**
  - We can use a simple CNN model and train it for this classification problem with possible classes being 0-36 as the numbers are 4-digit numbers.
  - The Loss function used is Cross entropy loss and the optimizer used is Adam.

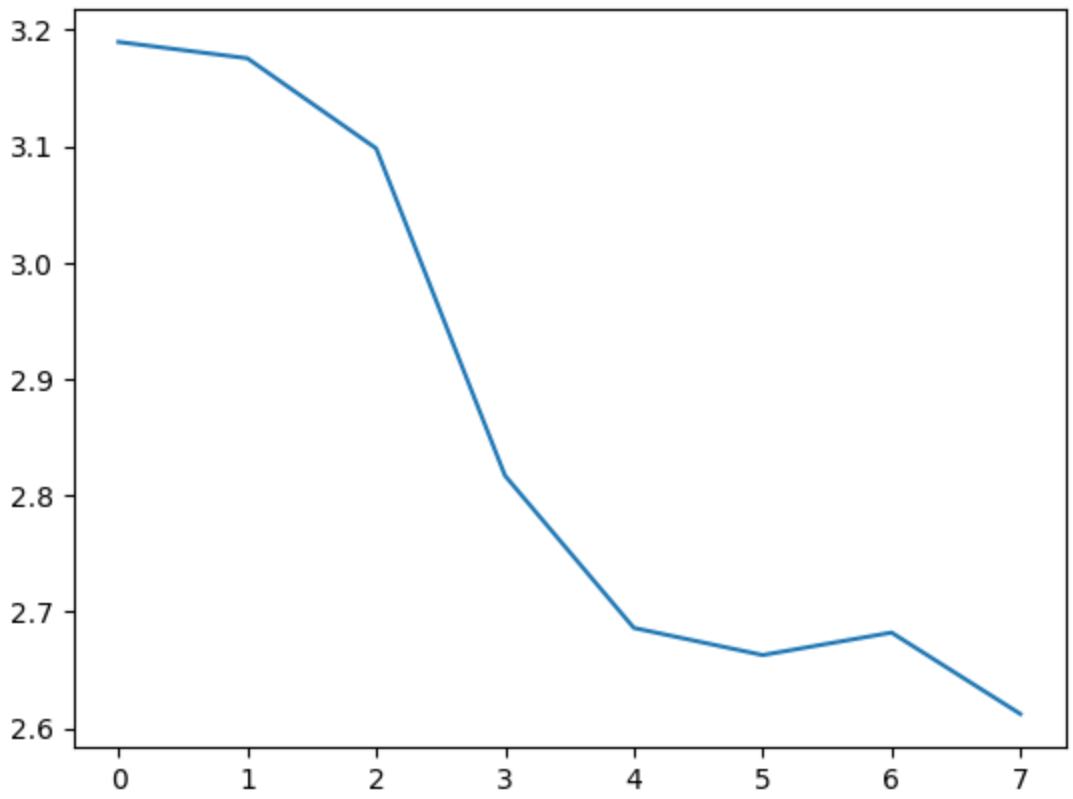
```
class CNNModel(nn.Module):
    def __init__(self, num_classes=37):
        super(CNNModel, self).__init__()

        self.conv1 = nn.Conv2d(in_channels=1, out_channels=32
        self.conv2 = nn.Conv2d(in_channels=32, out_channels=6
        self.conv3 = nn.Conv2d(in_channels=64, out_channels=1

        self.fc1 = nn.Linear(128 * 5 * 21, 256)
        self.fc2 = nn.Linear(256, num_classes)

        self.pool = nn.MaxPool2d(kernel_size=2, stride=2)
        self.dropout = nn.Dropout(0.7)

    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
```

```
        x = self.pool(F.relu(self.conv3(x)))

        x = x.view(x.size(0), -1)

        x = F.relu(self.fc1(x))
        x = self.dropout(x)
        x = self.fc2(x)

        return x
```

- **Results Obtained:** The accuracy obtained on the validation dataset is ~ 13.5% . The results obtained are not very good as the simple CNN model cannot learn the task accurately and requires more sophisticated techniques.
    - Training Loss:



- Val Loss:



- **Hyper parameters used:**
  - batch size: `64`
  - learning rate: `1e-3`
  - num_epochs: `8`
  - dropout rate: `0.7`
- **File Structure:**
  - `training.ipynb`: Loads, data and trains our CNN model. The trained model is then saved.
  - `inference.ipynb` : Loads the model and performs testing.
  - `model.pth`: Saved CNN model