Name: Akash verma
Rollno: 2018101011

Question 5: Report result

Question 1: Model Building

Build a multi-layered perceptron (MLP) in Pytorch that inputs that takes the (224x224 RGB) image as input, and predicts the letter (You may need to flatten the image vector first). Your model should be a subclass of nn.Module. Explain your choice of neural network architecture: how many layers your network has? What types of layers does it contain? What about other decisions like use of dropout layers, activation functions, number of channels / hidden units.

Results: *In this question we have used neural network architecture with 5 layers. The type of layer it contains is LINEAR. There are no dropout layers. Here the activation function is ReLU.*

Question 2: Training Code

Write code to train your neural network given some training data. Your training code should make it easy to tweak hyperparameters. Make sure that you are checkpointing your models from time to time (the frequency is up to you). Explain your choice of loss function. Ensure that your code runs on GPU.

Results: *The choice for the loss function is Cross Entropy as it is good for classification because it helps to reduce the distance between the groups.*

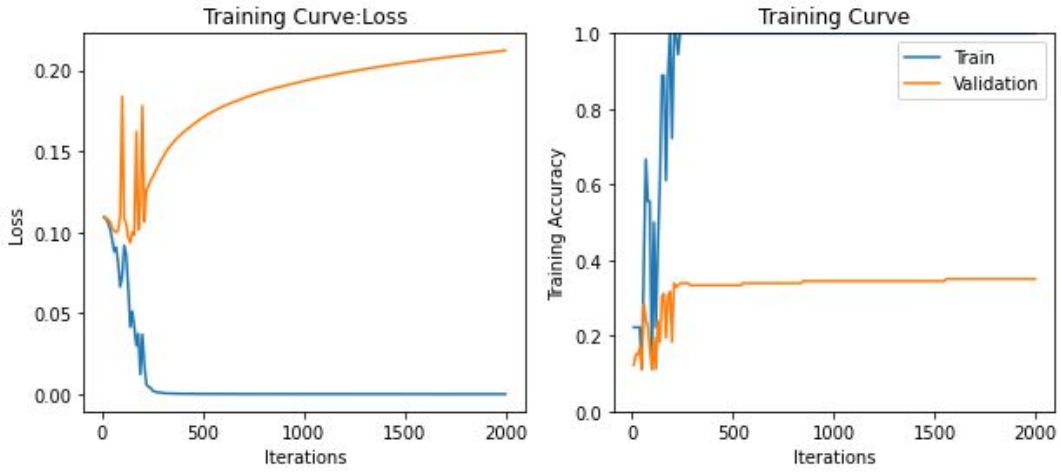
Question 3: Overfit to a Small Dataset

Part a:

One way to sanity check our neural network model and training code is to check whether the model is capable of overfitting a small dataset. Construct a small dataset (e.g. 1-2 image per class). Then show that your model and training code is capable of overfitting on that small dataset. You should be able to obtain a 100% training accuracy on that small dataset relatively quickly.

If your model cannot overfit the small dataset quickly, then there is a bug in either your model code and/or your training code. Fix the issues before you proceed to the next step.

Results:

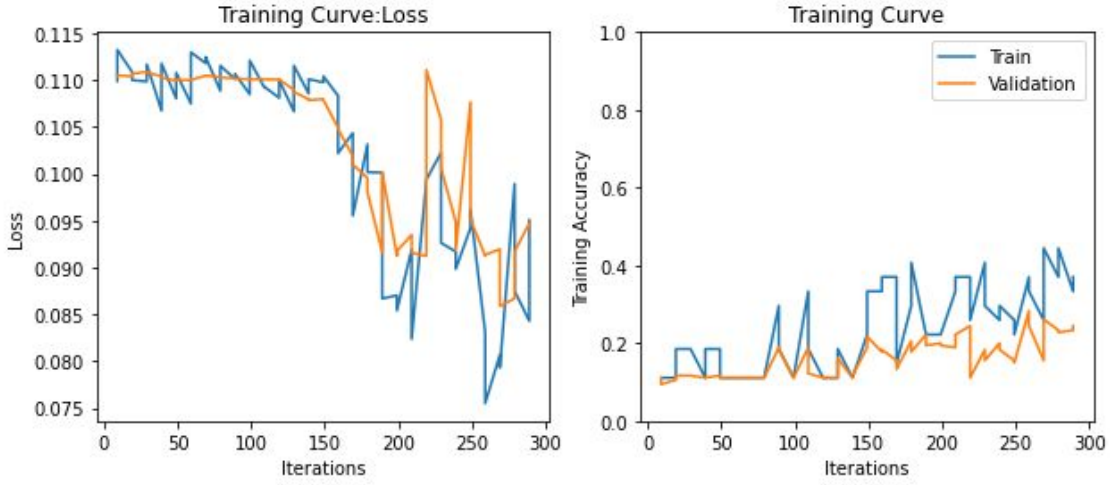



```
Final Training Accuracy: 1.0
Final Validation Accuracy: 0.35
```

Part b:

Once you are done with the above part, try to reduce the effect of overfitting by using techniques discussed in the previous lecture.

Results:




```
Final Training Accuracy: 0.37037037037037035
Final Validation Accuracy: 0.24444444444444444
```

Question 4: Finetuning

For many image classification tasks, it is generally not a good idea to train a very large deep neural network model from scratch due to the enormous compute requirements and lack of sufficient amounts of training data.

In this part, you will use Transfer Learning to extract features from the hand gesture images. Then, train the last few classification layers to use these features as input and classify the hand gestures. As you have learned in the previous lecture, you can use AlexNet architecture that is pretrained on 1000-class ImageNet dataset and finetune it for the task of understanding American sign language.

Results: