TP 6: Deep Learning

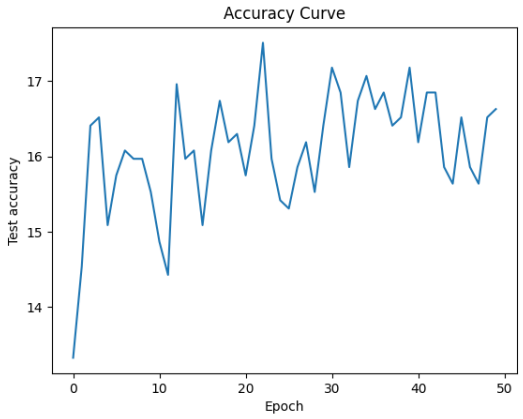
Martin Jolif

February 27, 2025

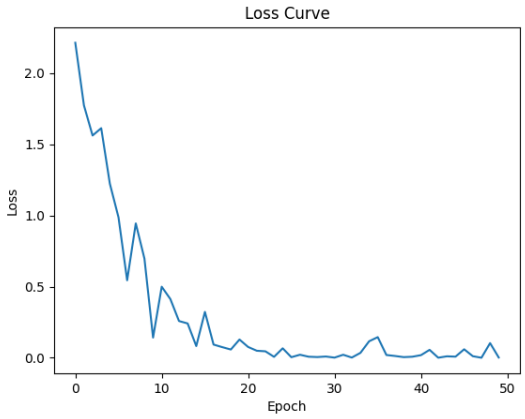
1 Question 1

The experiments were done with a learning rate value of 0.001, and 50 epochs.

For the *ModelNet10_PLY* dataset, I obtained the following results:



Test accuracy

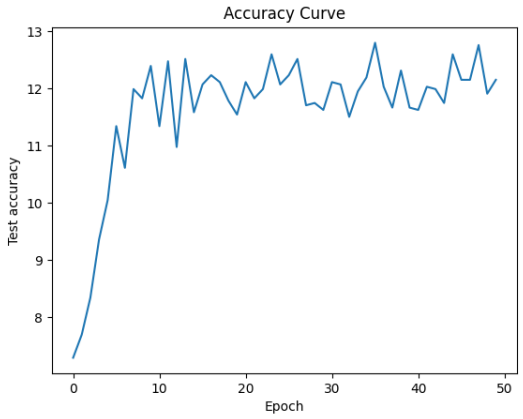


Train loss

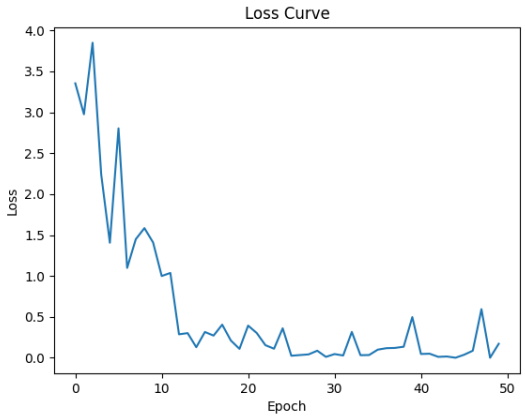
Figure 1: ModelNet10_PLY dataset

The test accuracy is around 16% for the *ModelNet10_PLY* dataset. For the *ModelNet40_PLY* dataset, I obtained the following results:

The test accuracy is around 12% for the *ModelNet40_PLY* dataset. The results are a bit better than a random choice, however they are not very good. It seems that our MLP model is over-fitting on our training set as the training loss quickly converges to zero and the accuracy stabilizes at the same time. These results were expected as our MLP model doesn't have any information about the 3D structure of the data. Moreover, this MLP model isn't invariant by permutation as it should be to process 3D data.



Test accuracy

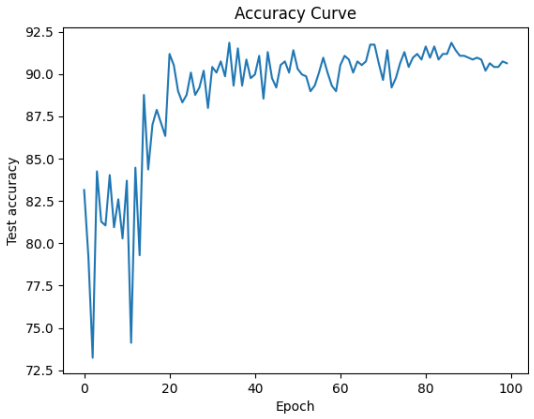


Train loss

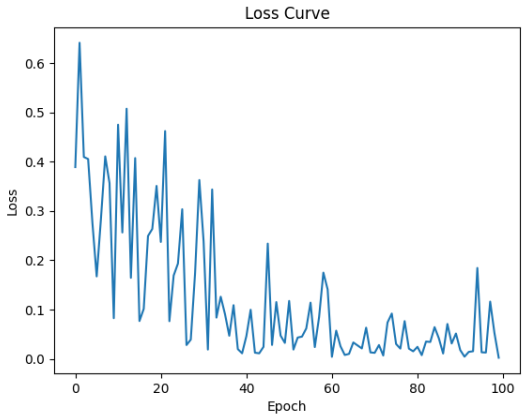
Figure 2: ModelNet40_PLY dataset

2 Question 2

The experiments were done with a learning rate value of 0.001, and 100 epochs. The test accuracy is around 90% for the *ModelNet10_PLY* dataset:



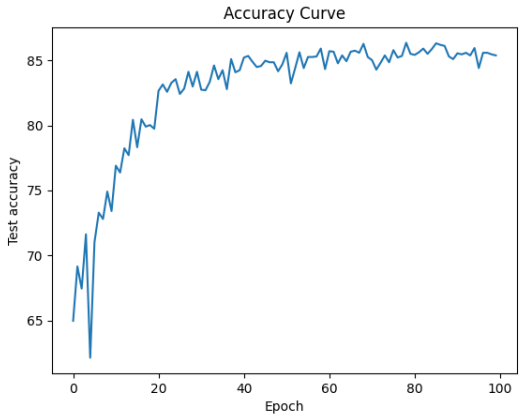
Test accuracy



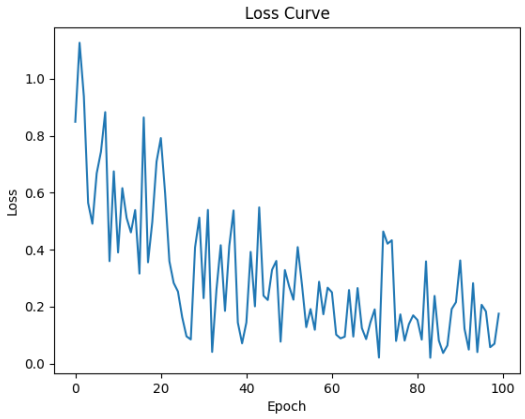
Train loss

Figure 3: ModelNet10_PLY dataset

This time, the test accuracy is around 85% for the *ModelNet10_PLY* dataset. This model clearly outperforms the MLP. This was expected as the Pointnet model is invariant to permutation thanks to the max pooling layer. Moreover, the spatial information is also conserved as the MLP layers operate on the unflattened data (shared MLP layers).



Test accuracy

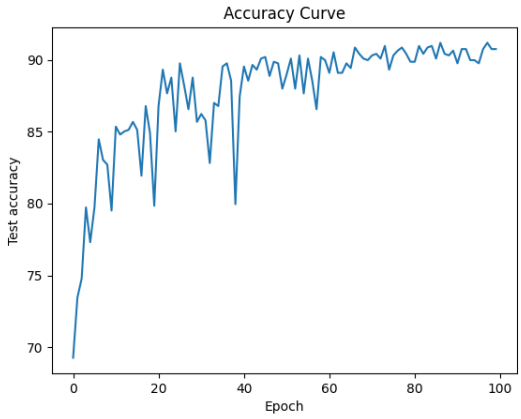


Train loss

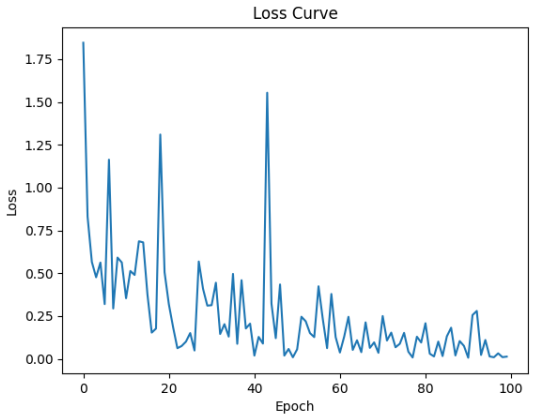
Figure 4: ModelNet40_PLY dataset

3 Question 3

The experiments were done with a learning rate value of 0.001, and 100 epochs.



Test accuracy



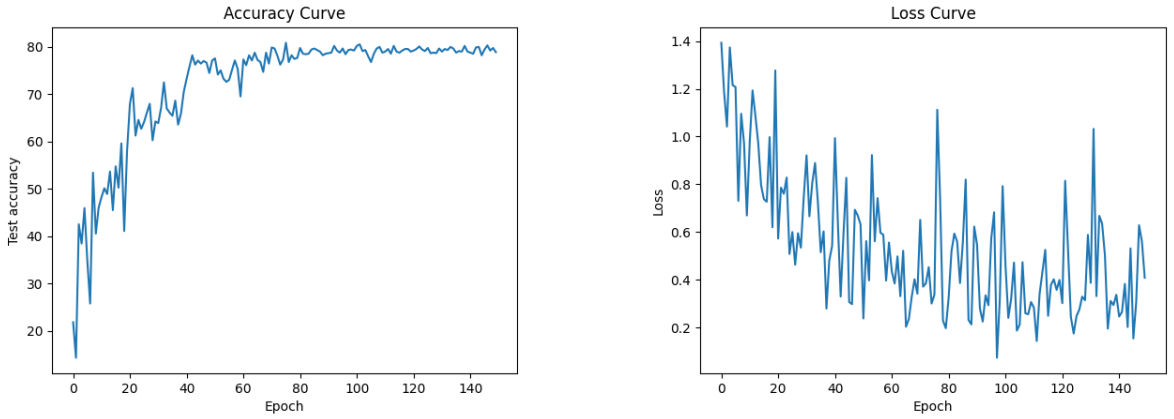
Train loss

Figure 5: ModelNet10_PLY dataset

With this T-Net network, the test accuracy is also around 90% as it was the case with the basic version of PointNet. However, this with this T-Net network, the model has a lot more parameters and requires much longer training. Therefore, we should use the basic version of PointNet as it gives the same results with less training time. The fact that the results are not improved with the T-Net network can maybe explained by the data augmentation. Indeed, the goal of the T-Net network is to rotate the data such that they are all aligned. But, with the data augmentation, we can give to the network different point cloud with different alignment, then the network learn to process it automatically.

4 Question 4

The experiments were done with a learning rate value of 0.001, and 150 epochs by training the basic PointNet (as it should give good results and a relatively fast training as explained in the last question).



Test accuracy Train loss

Figure 6: ModelNet10_PLY dataset

For my personal data augmentation, I have chosen to do random rotations along the three axes (x-y-z) and to add a bit of random noise to the data.

The results are around 80% test accuracy for this data augmentation. This is below the 90% test accuracy corresponding to the default data augmentation. This lower score can be explained by the fact that rotation along the z-axis, preserves the idea of the gravity and the semantics of the object from the 3D point cloud. In contrast, rotations along the x- or y-axis don't preserves the idea of the gravity, this could change the semantics associated with the object. For example, a dresser rotated along the x-axis could also look like a table.