TP6 : Deep learning for point clouds

Dany MOSTEFAI, Elhadi CHITER

February 2025

1 MLP

For the simple 2D spiral dataset, the MLP manages in 1 epoch to correctly discriminates between the inner circle and the outer ring (see figure 1).

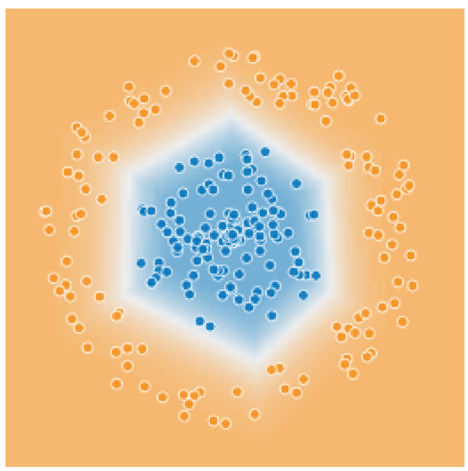

Figure 1: Simple MLP trained to discriminate non linear data.

2 Setup

We made our experiments on the dataset "ModelNet40_PLY" only.

3 Question 1

We trained our simple MLP for 50 epochs with a learning rate of 0.0005. Figure 2 shows the evolution of the train loss and the test loss during training. You can clearly see that the model just tries to learn by heart the training set, and completely fails to generalize on the test dataset.

The architecture is clearly not fitted for our classification problem.

The best accuracy on test dataset during training for this model is 11%.

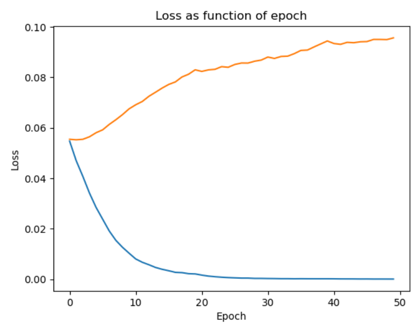

Figure 2: Evolution of the losses for the MLP during training

4 Question 2

We trained our basic point net for 50 epochs with same learning parameters as the MLP (50 epochs, learning rate of 0.0005). The test accuracy of 81% is very encouraging, and is close to what is shown in article [1].

Figure 3 shows the evolution of the losses during training.

The model generalizes well, and seems to start overfitting at around 50 epochs.

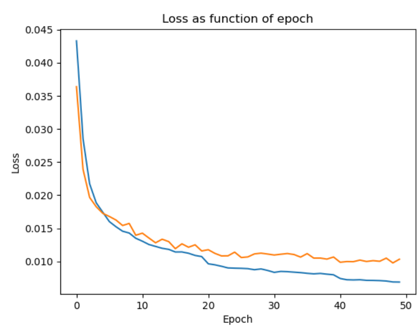

Figure 3: Evolution of the losses for the basic pointnet during training

5 Question 3

We've added a Tnet that outputs a 3x3 rotation matrix, and the loss has been adapted with a regularizer that ensures that the matrix yielded by the Tnet is orthogonal.

The hyperparameters are the same as before.

We've trained for 967 (50 epochs) seconds our model, and reached a maximum test accuracy of 79.7 %. Notice how the test loss 4 is noisier due to the added Tnet, and this behavior is exacerbated with the test accuracy (The accuracy can vary a lot from one epoch to the next one).

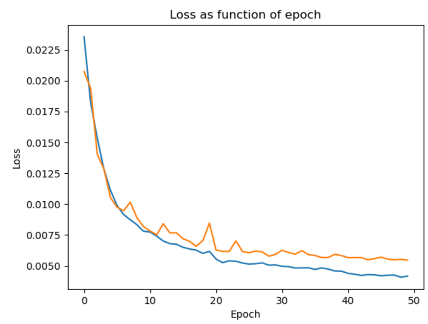

Figure 4: Evolution of the losses for the pointnet during training

6 Question 4

We had many ideas for data augmentation, inspired by article [2] which experiments various data augmentation techniques for point clouds segmentation and classification.

Our added data augmentation is pretty simple, we made a random rescaling on each axis of the point cloud.

Moreover, we've changed slightly the hyperparameters : we've reduced the stepsize of the scheduler from 20 epochs to 10 epochs, because the pointnet is very instable as shown in figure 4.

The best accuracy we've got is 78.5 %, with 1200 seconds of training.

The losses behave well 5, and the test loss is less "erratic" than before.

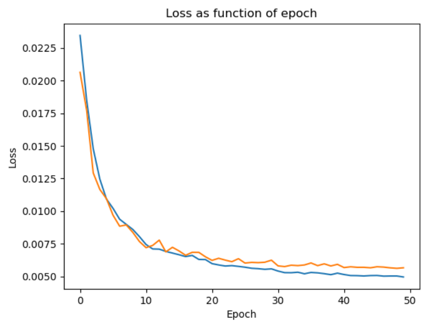

Figure 5: Evolution of the losses for the pointnet during training with added data augmentation.

For a fair comparison, we retrained the pointnet full with base data augmentation with the same hyperparameters than the model trained with our data augmentation.

We've gained approximatively 200 seconds during training compared to our data augmentation. Also, the accuracy reached is roughly the same (78.6 %). However, we can notice in figure 6 that the test loss is much farther away from the training loss than in figure 5, indicating a more prounonced overfitting.

To conclude, we don't notice any meaningful change with our added data augmentation given our number of epochs (except the fact that the training is slower). However, if we've continued our training to 100 epochs, the model with our data augmentation would certainly overfit later than the one without our data augmentation.

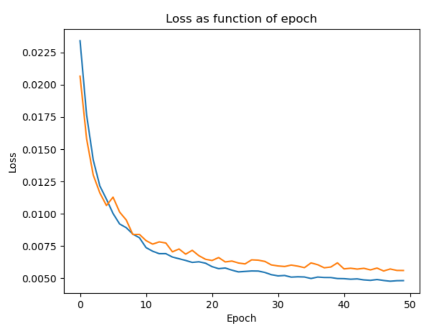

Figure 6: Evolution of the losses for the pointnet during training with base data augmentation.

Our pointnet is slightly better in accuracy than the baseline, which is a MLP trained on extracted features from the point clouds, proposed in [1].

7 Question Bonus

After numerous hours of debugging, we weren't able to use Minkowski engine (In fact, we could use it, but on cpu, which is very slow).

References

[1] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CoRR*, abs/1612.00593, 2016.

[2] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Abed Al Kader Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies, 2022.