MATHÉMATIQUES
VISION
APPRENTISSAGE

Dauphine UNIVERSITÉ PARIS | PSL

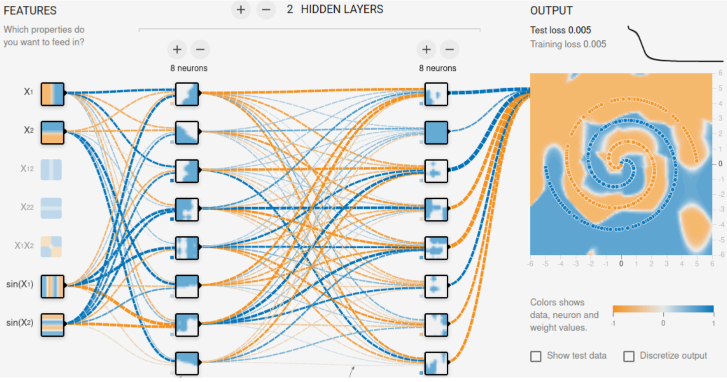
TP6: Deep Learning

Louis Martinez
louis.martinez@telecom-paris.fr

Mohamed Ali Srir
mohamed.srir@telecom-paris.fr

Exercise A: Understand MLP for geometric 2D data

Fitting the spiral dataset is tougher than the other datasets. To get decent results, we need to extensively tune the parameters.



2 HIDDEN LAYERS

8 neurons

8 neurons

Test loss 0.005
Training loss 0.005

Figure 1: Classification results obtained on the Spiral dataset.

Exercise B: MLP on point clouds in PyTorch

*We use the same training hyper-parameters across all experiments. Therefore, every model is trained for **250** epochs. We use a learning rate of **0.001** and a batch size of **32**.*

Question 1:

Metric	ModelNet10_PLY	ModelNet40_PLY
Test Accuracy	18.9%	12.3%

Table 1: Results obtained with a basic MLP

The basic MLP implementation provides poor results on both ModelNet versions. Test accuracy is barely improved across epochs. As expected though, we get better performances on ModelNet 10 since the classification task is easier.

Exercise C: PointNet in PyTorch

Question 2:

Metric	ModelNet1c0_PLY	ModelNet40_PLY
Test Accuracy	91.7%	86.9%

Table 2: Results obtained with the basic PointNet implementation (without T-NET)

The basic version of PointNet provides satisfying results. From the first epoch, test accuracy lies above 70% and keeps increasing.

Question 3:

Metric	ModelNet10_PLY	ModelNet40_PLY
Test accuracy w. T-NET	**92.3%**	85.2%
Test accuracy wo. T-NET	91.7%	**86.9%**

Table 3: Results obtained with PointNet on both ModelNet datasets. We compare accuracies obtained with (top) and without (bottom) T-Net module.

This improved version of PointNet still delivers satisfying results. However, adding the 3×3 input transform does not seem to significantly improve test accuracy. While it yields a 0.6% improvement on ModelNet10, performance drops by more than 1.5% on ModelNet40. This suggests that the transform may not scale well with the number of classes. For example, if the learned orientation is poorly suited for even one of the 40 classes in ModelNet40, overall accuracy could suffer.

Question 4:
We added a small random translation to every point-cloud. The translation vector is obtained by sampling each component independently. It is then normalized and multiplied by a pre-defined magnitude coefficient.

The goal of this data augmentation procedure is to take into account the fact that, within the same class, characteristic features from one sample to another may be slightly shifted. Intuitively, this augmentation may provide better generalization capabilities when the number of classes increases.

Metric	ModelNet10_PLY	ModelNet40_PLY
Test accuracy w. translation	91.4%	**86.4%**
Test accuracy wo. translation	**92.3%**	85.2%

Table 4: Results obtained with the full PointNet implementation (with T-Net). We compare accuracies with (top) and without (bottom) random data translation.

While results on ModelNet10 are comparable to the basic implementation of PointNet (event slightly worse), test accuracy on ModelNet40 has been improved by 1.2%. Although the improvement is slight, it's important enough not to be a consequence of randomness during training. Therefore, it confirms the aforementioned intuition on the expected result.