# Pattern Recognition Group Exercise 2d

Davud Bucher, Timo Bürk, Félicien Hêche, Aleksandar Lazic, Zahkar Tymchenko

April 20, 2020

## 1 MLP on Scambled MNIST

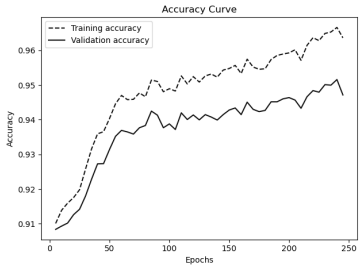
### 1.1 Expectation

As the features in an MLP are independant (there is no spatial idententification) and the permutation constant on the dataset, we do not expect significant difference with the result obtained in task 2b. If we take our trained MLP on the normal data set, there is conceptualy exactly one permutation on the parameter of the first layer for which the resulting network would give the exact same prediction on the permutated data.

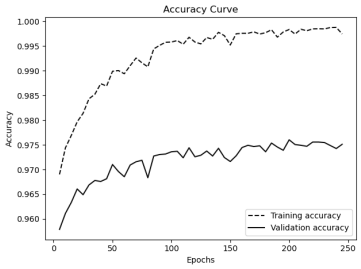
$$MLP(x) = MLP'(x')$$

### 1.2 Results and Interpretation

As we can see the validation accuracy are very similar (0.946 on permutated versus 0.975 on nomral MNIST), the same goes for the validation loss (0.06 versus 0.03). But the crurves have another shape. Both the validation and training curves for accuracy and loss on the MNIST set looks like a logarithmic function. But On the permutated MNIST curves show a little angle, allowing us to identify two phases, one with fast progress and one with small progress. This can be explain by the fact we did not perform parameters optimization for the 2d task and just re-used the same as in 2b. We did observe similar effect during the parameter tunning phase in task 2b.

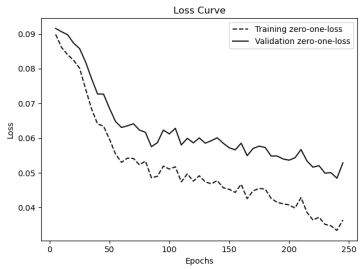



(a) Permutated MNIST

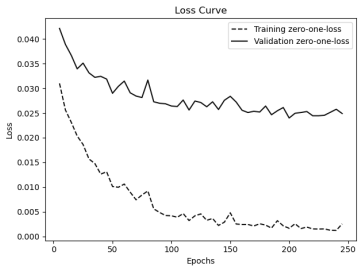



(b) MNIST

Figure 1: MLP Accuray




(a) Permutated MNIST




(b) MNIST

Figure 2: MLP Loss

# 2 CNN on Scambled MNIST

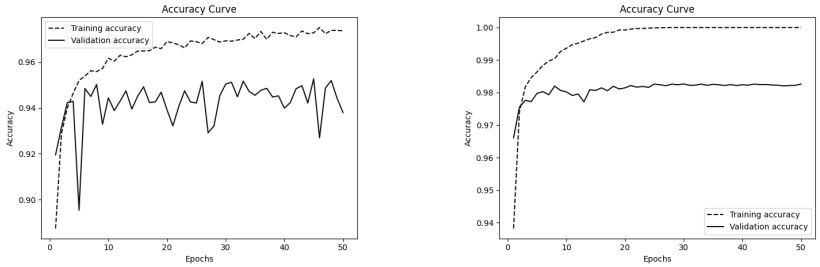
## 2.1 Expectation

The idea behind CNN is that the relevant information could be localize anywhere in the picture and has some spatial structure. A kernel could be training to recognize circle, straigth line, intersetcion, all kind of small element present in digit. On the Scramble dataset, we cannot see those pieces anymore, but since it is the same permutation applied on each picture, there are still paterns to learn, just not the one human eyes *sees*. In that regard we do not expect much different results. **But** CNN depends on the number of kernels and most importantly here, on the size of the kernels. Let us consider two pixels that are close to each other in the original data set (for example, having high chances to be black on a picture from class 2, but white in class in class 3) - potentialy important to learn meaningful pattern. Now consider the image of these pixels in the permutated set, as a pair they still share information but depending on how far they are from each other, they may never be processed by the kernel at the same time. This is a potential problem. Now some theoretical work could be done: considering kernel size, picture size, average resulting distance between corrolated pixels,..., how does the average accuracy change ? Lets just

point out that we are using kernels of size 7 on a picture of size 28, kernels *see* only 1/16 of the image at the same time, which is not a lot, specialy because in the pictures *small*, there are more pixel that belongs to the **background** than pixels **relevant** to the class. Thus we expect a significant drop in accuracy, but still the CNN should be able to perform somewhat good predicitions.

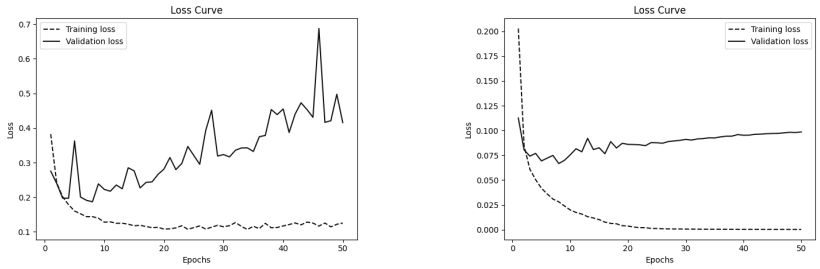
## 2.2 Results

As we can see, the figures are very similar, and the trained CNN have similar accuracy (0.93 on permutated versus 0.985 on normal). This small difference is surprising. One hypothesis is that the white pixels are used to learn pattern. As mentionned, the MNISt data set is *very nice*, grey scale, no noise, etc. While the human eyes recognize classes based on the shape of the black pixels, nothing prevent the machine to recognize white circle rectangle, and since most pixels are white, the relative small size of the kernels may find patterns this way.
One interesting experiment would be to redoo the entire 2c/2d task with a preprocess steps, where we use RGB input channels and randomly colorize each white pixels and colorize with one unique random color all the black one (using a grey threshold to define black and white).




(a) Permutated MNIST (b) MNIST

Figure 3: CNN Accuracy




(a) Permutated MNIST (b) MNIST

Figure 4: CNN Loss