**Report Series 2d:**

**CNN**



Non-permutated MNIST



permutated MNIST



**Figure 1:** Accuracy and loss of a CNN algorithm applied on the standard MNIST data set (left column) and the a permutated MINST data set. Instead of using the validation set, the test set was used as comparison. Different optimal learning rates were detected depending on the data set.

In case of the CNN algorithm we see that the accuracy on the training set reaches 1 quite fast when applied on the non-permutated data set. A high accuracy can also be reached when applying the same CNN algorithm on the non-permutated test set. By accident, we used the train set instead of the validation set to verify the accuracy of the algorithm with a different data set. However, this should not undermine our general observations. When using a slightly modified CNN algorithm (different learning rate) on the permuted data set, we can observe a drastic change of the accuracies. For the training data set it barely reaches 0.32 and it also increases slower compared to the standard MNIST. In case of the permutated test set the accuracy of the CNN stays more or less constant but never exceeds 0.23.

**MLP**



**Figure 2:** Accuracy and loss of a MLP algorithm applied on the standard MNIST data set (left column) and the a permutated MINST data set.

When applied on the standard MNIST data set the accuracy of MLP algorithm increases rather slow compared to the respective CNN. The maximum accuracy reached on the training and test respectively are 0.97 and 0.925, which are lower than for the CNN. The accuracy on the validation set seems to reach a plateau in contrast to the training set curve. The spiky shape of both curves further indicate that the accuracies fluctuate more compared to the CNN algorithm. Used on the test set the MLP had an accuracy of 0.897.

However, the shape of the curves is more smoothly when the MLP was used on the permutated data sets. The maximum accuracies are similar to those observed on the standard data set, but a plateau is reached faster in both cases. Applied on the test set the MLP had an accuracy of 0.969.

The CNN algorithm performs worse on the permutated data set than the MLP. This might be due to the fact, that CNN uses spatial information for the classification. When we assume that the permutation randomly shuffled the pixels, this feature might become a disadvantage for the CNN, since there is no clear spatial pattern for different digits visible anymore. For example, a CNN might identify images where two circular shape lie close to each other as the digit 8, but when permutated this unique spatial information might get lost and the classification fails. In contrast, an MLP does not rely on spatial information because the input is flattened to a one-dimensional vector. When classifying the permutated data set, the MLP probably rather uses the number of dark pixels as classification criterion, which could explain its better performance. However, we cannot explain why we get a higher accuracy on the permutated test set than on the standard one in case of the MLP. We probably made a mistake somewhere, but we could not detect it.

The Multi-Layer Perceptron (MLP) has a couple of disadvantages compared to a Convolutional Neural Network (CNN). One of them are the fully connected layers in the MLP, which can cause the number of parameters to grow very high. Such high dimensions can cause redundancy and hence inefficiency during the learning process.

Additionally, a larger number of parameters results in a higher risk of overfitting. Furthermore, the standard MLP disregards spatial information since it takes flattened vectors as inputs.

In the CNN approach however, spatial information is not lost as it can take 2D images as input. I could imagine that this property makes the learning process of a CNN more “flexible”, meaning that the learned categorization techniques are more general and can be successfully applied to varying data sets. Hence, I would expect to see the CNN perform better on the permutated MNIST data set than the MLP.

A CNN algorithm is also more efficient during the learning process than the MLP since the weights are smaller and shared (the layers are not fully connected). Maybe this could also cause a higher accuracy of the CNN.

The lower parameter number probably also reduces the risk of overfitting the CNN algorithm.

In general, I would expect the CNN to perform with a higher accuracy the MLP, for the reasons stated above. It would make sense if the difference in accuracy would be bigger on the permutated data, due the lack of spatial information in case of the MLP. That overfitting is the reason for any performance difference of CNN and MLP, is rather unlikely in my opinion, since we optimized the corresponding parameters in 2b and 2c.