# SSU Assignment 2
# Backpropagation

Marija Pajdaković

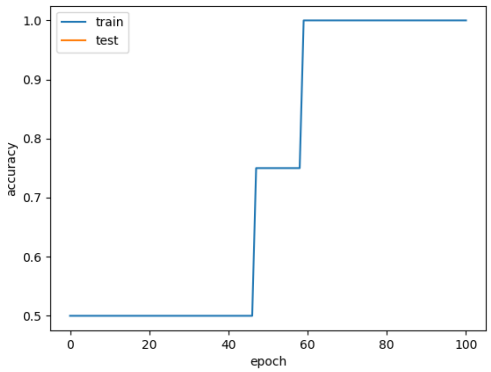
December 8, 2022

## 1 Assignment 1

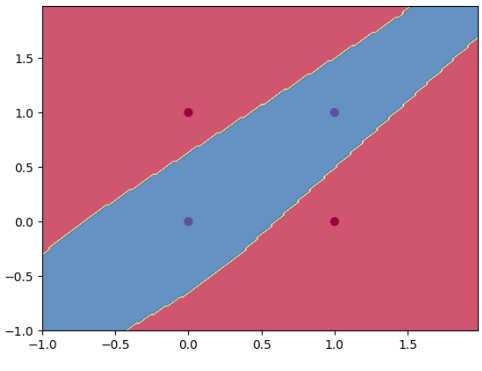
Our first assignment was to implement all messages in *LinearLayer*, *ReLULayer*, *SoftmaxLayer* and *LossCrossEntropy*, and than to train MLP classifier on *experiment spiral* dataset. To start off, we could use a popular debugging tool - a *XOR* dataset. We can see the performance of the model trained on this dataset on the Figures below. On the first Figure, we can see how accuracy was improving during the training epochs, while on the second one it is rather clear that our two classes here are separated correctly. After we assured ourselves that the layer messages are implemented properly, we tested it on the *experiment spiral* dataset. On the following Figure, we can see how training looked like for three different values of hyper parameter alpha.

The results are the best for alpha = 1, slightly worse for alpha=0.2, but definitely the worse for alpha=5. It is not surprising that we have got these results for alpha=5, since learning rate is a configurable hyperparameter used in the training that has a small positive value, often in the range between 0 and 1. This parameter is not learned by the network, so we achieve the optimal one by trying different configurations and checking the results.

On the Figure below, we can see how the accuracy was improving over traning for *mnist unstable* dataset.




(a) Accuracy during training




(b) Model classification

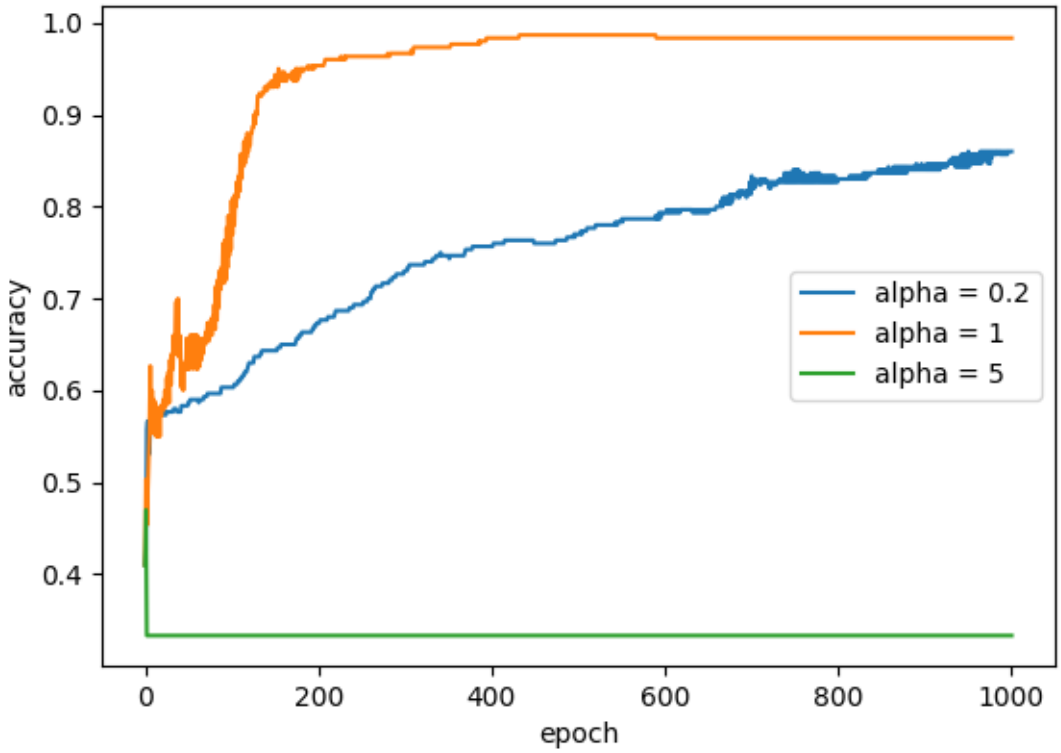



Figure 2: Three settings of the learning rate for the spirals dataset

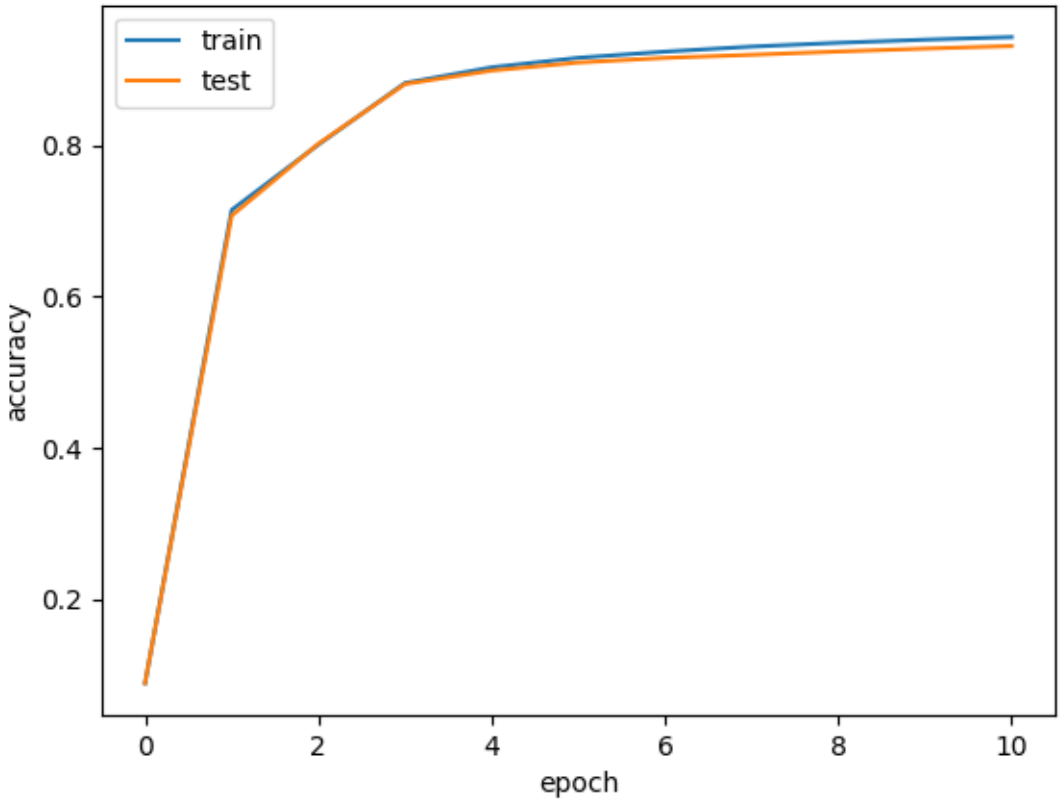



Figure 3: MNIST unstable dataset

# 2 Assignment 2

In this task, we had to give a forward and backward message of the compound layer of *SoftMax Layer* and *CrossEntropyLoss layer*. The solution is given on the images in this Zip file.

# 3 Assignment 3

In this task, we had to implement the forumlas that we derived in the Assignment 2 and use this combined *LossCrossEntropyForSoftmaxLogits* loss for the unstable dataset - *experiments MNIST*. By only looking at the formula, we can see that division is replaced by subtraction/addition, which makes it more numerically stable. We prevent overflows that were happening when we used *SoftmaxLayer* and *LossCrossEntropy* separately. Accuracy of this classifier was 0.99, loss: 0.02 for 100 epochs.

# 4 Conclusion

In this task, we have impelmented a multilayer perceptrom alghoritm and trained it on several datasets.