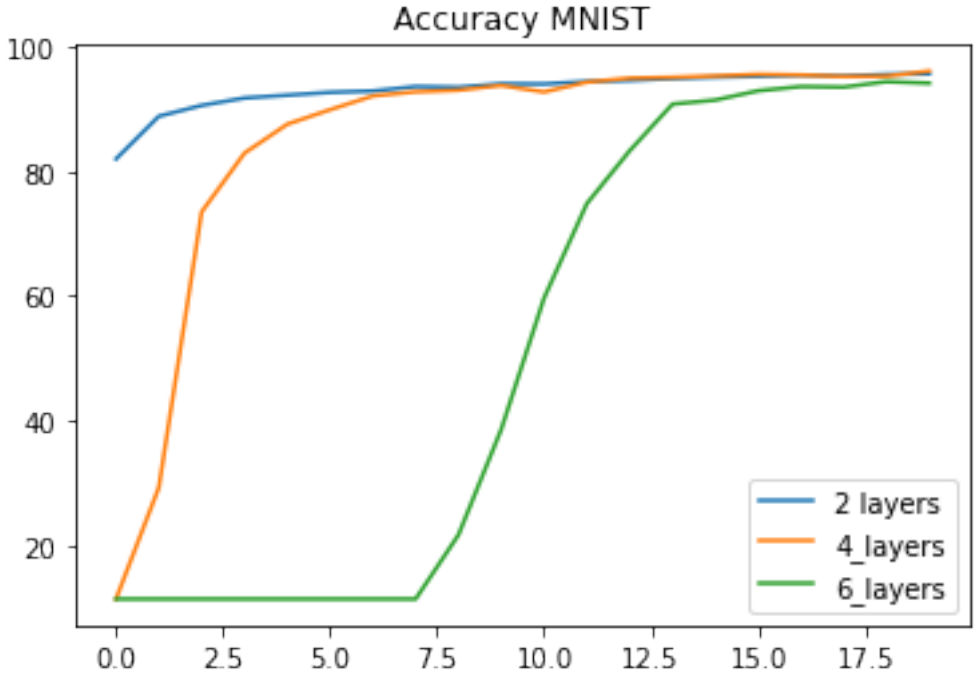
Costrita Alexandru – GP3 – ESITech

Report

Building a classificator for the Fashion MNIST dataset

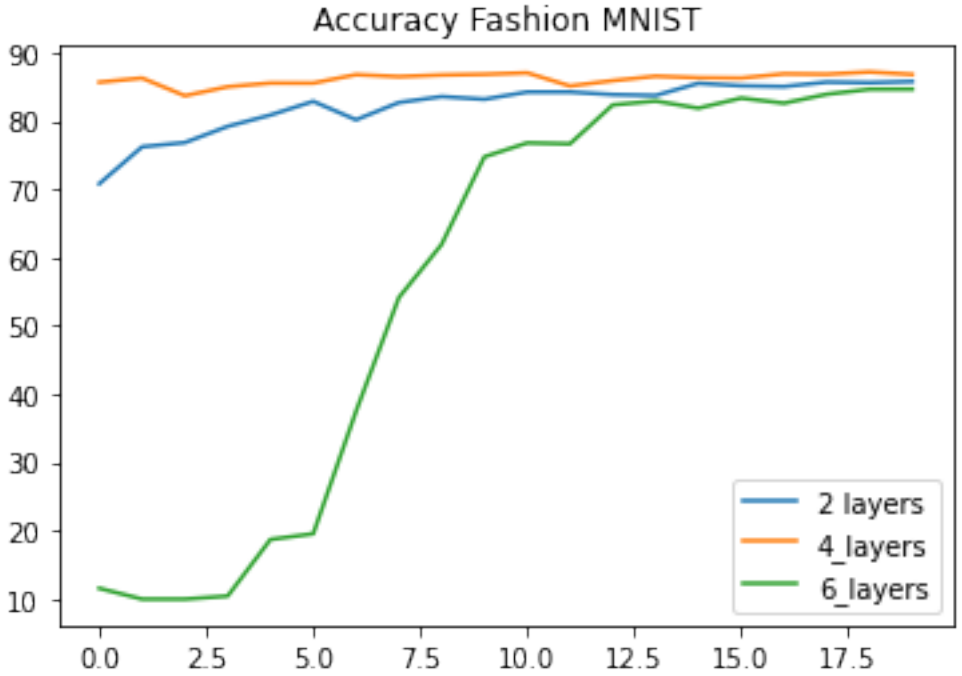
In the following report we will build a classificator for the fashion MNIST dataset which a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes.

The first curve that we will analyze will be the accuracy of the MNIST with the hyperparameters of the depth set to 2, 4, 6; the optimizer will be sgd and the model type mlp.

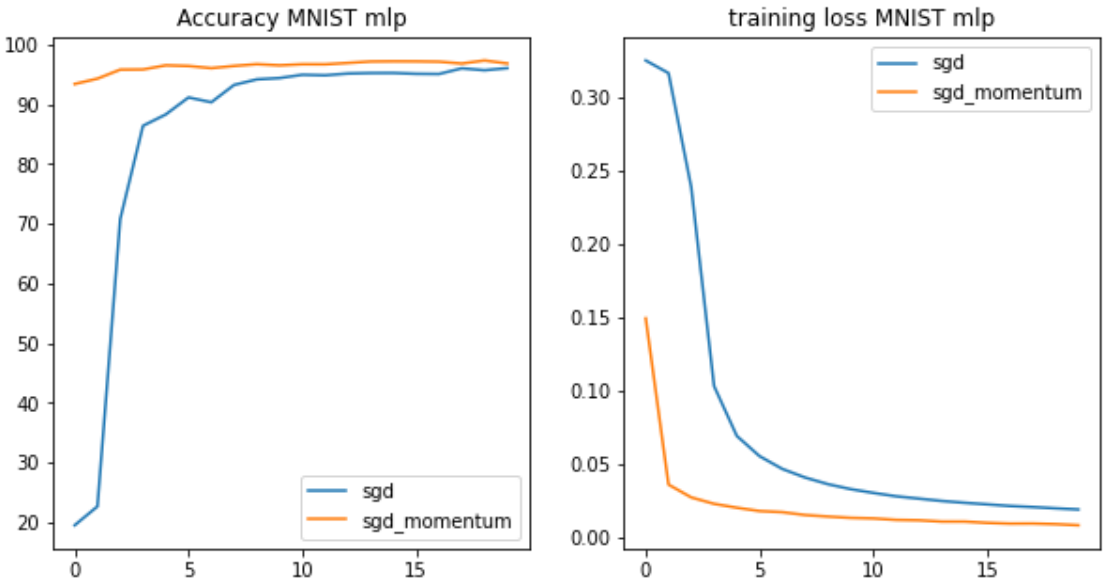


We can observe that each time we adjust layers, our accuracy takes more time to actually understand and reproduce the image. Besides that, the 2-layer depth seems to be the most efficient and accurate with a starting accuracy of 82% and with a constant show of good results with high efficiency. Also, we can mention that the full 100% accuracy wasn't attended in with the MNIST protocol.

The second curve will be the accuracy of the Fashion MNIST with the hyperparameters: sgd,2-4-6 layers, and mlp as a model type. We obtain the following graph:



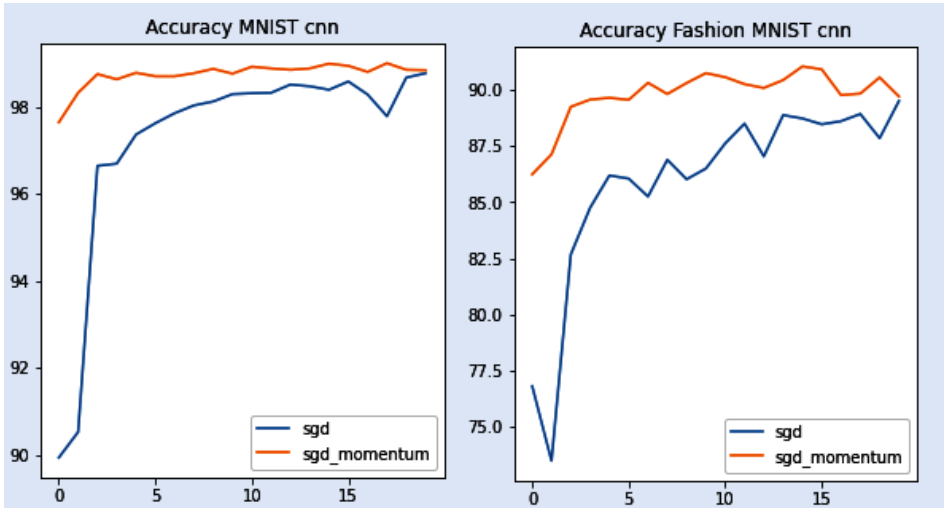
The first thing that we can notice is the high initial accuracy of the 4-layer method and a constant performance for both the 2- and 4-layer protocol. For the 6-layer method however we conclude that the first half of the tries were difficult to spot and a low efficiency in the calculus, which means that the 6-layer method needs more time to show actual results, so I think it isn't the right choice into working with hyper-big datasets.

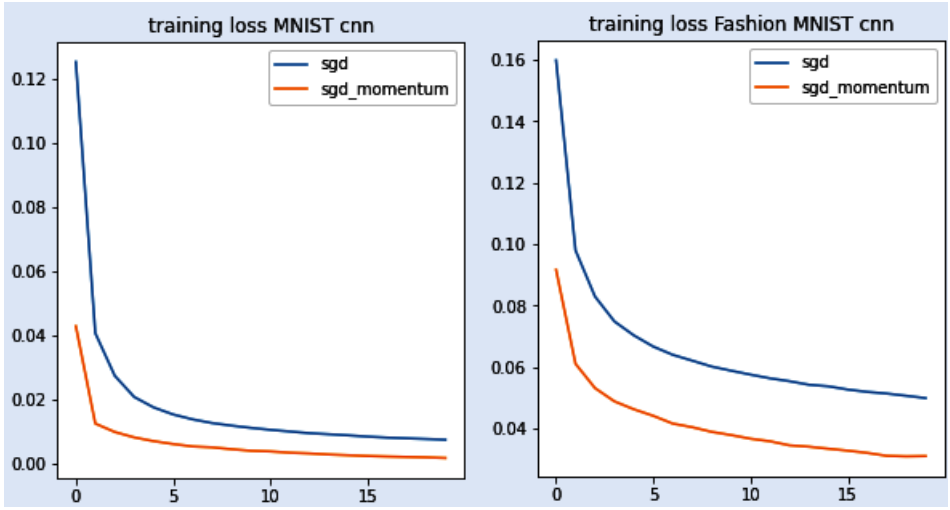


From the graphs above we can conclude that the sgd momentum hyperparameter assure a good and very fast learning rate and if we compare to sgd we can see a

higher efficiency in a lesser time, in the following projects sgd momentum will be a better choice.

In the following graphs we changed the model type mlp with cnn in order to see if there is any difference between the methods and compare their efficiency. The results show:





The cnn model type for Fashion Mnist is a little tricky because it shows constant ups and downs so the volatility of the analyzed data's efficiency is quite high. For the Mnist it doesn't change that much, the efficiency is high and the time is quick.

If we will have to conclude on the setup of the different hyperparameters and how they influence our system, we will prefer to make a table and put all the known information. This is why the table of accuracy containing rows for the dataset and columns of model and a table summarizing the hyperparameters used for both models will represent our conclusion:

	mlp	**cnn**
Mnist	93%+ efficiency during all learn rate	98%+ efficiency during all learn rate
Fmnist	85% efficiency during all learn rate	From 77.4% to 88%: long learning rate, efficiency not that high

And the table summarizing the hyperparameters:

	cnn	**mlp**
depth	No need	2-4-6 layers
optimizer	Sgd/sgd-momentum	Sgd/sgd-momentum

 # depth = This controls the number of layers
#sgd = Stochastic Gradient Descent
#sgd_momentum = Stochastic Gradient Descent with momentum

With the constants

- batch_size = 64
- n_epochs = 20
- learn_rate = 1e-1