

Deep Learning Lab Course 2017

Assignment 1.

Olesya Tsapenko

For the given dataset, I have chosen the next structure of my neural network:

#layer	Description	Number of units	Deviation for weights initialization	Activation function
0	Input layer	28*28	-	-
1	Fully connected layer	150	0.01	ReLU
2	Fully connected layer	100	0.01	ReLU
3	Fully connected layer	10	0.01	None
4	Softmax output layer	-	-	-

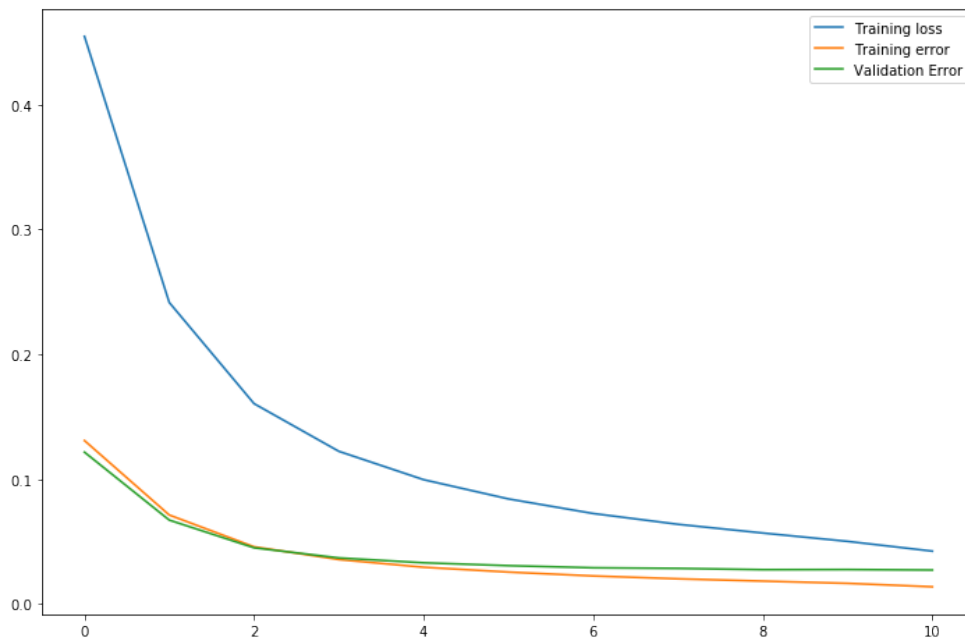
Changing the structure of the neural network did not bring significant improvements.

Thus, subsequent improvements were related only to other hyperparameters. I would like to show some of the configurations I had used:

- 1) **Gradient descent (GD)** with **vanilla** update. It provided very poor results. It requires unreasonable amount of time for just one updating of the current neural network and after 100 epochs, the training and validation error remains the same as it was after one epoch (around 89%).
- 2) **GD** with **Adam's** version of weight updating. This configuration of the neural networks improved results for GD. However, after 50 epochs the training and validation error remained at the level around 45%.
- 3) **Stochastic gradient descent (SGD)** with **vanilla** update, the size of the **batch 64**. This configuration tended to overfit a bit after 8 epoch but remained at the good result around 2% of validation error.
- 4) **SGD** with **Adam's** version of weight updating, the size of the **batch 64**. It had not improved somehow results of previous configuration. Therefore, at this moment, I had decided return SGD with vanilla updating because the Adam's version required storing running means of the gradient and the squared gradient.
- 5) **SGD** with **vanilla** update, the size of the **batch 10**. It started to overfit earlier (after 3 epochs).
- 6) **SGD** with **vanilla** update, the size of the **batch 100**. It required much more epochs for good results.

My final configuration for training was **SGD** with **vanilla** update, the size of the **batch** was **64**, and the number of **epochs** was **10**.


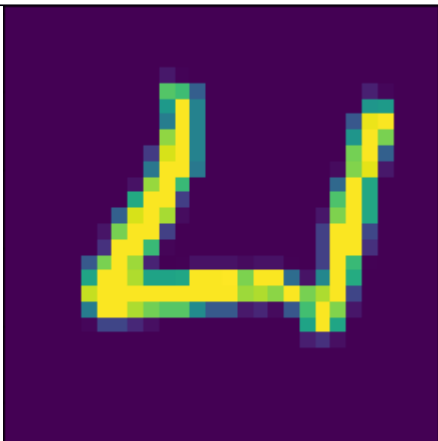
This configuration has led to the next results:

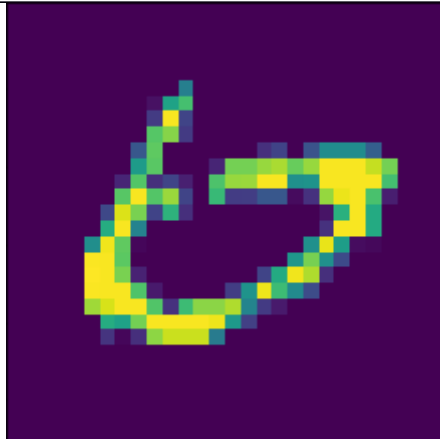


After achievement **2%** error on the validation set, I had trained this network on the complete data (train+validation sets), computed error on the test set and received **2.4%**.

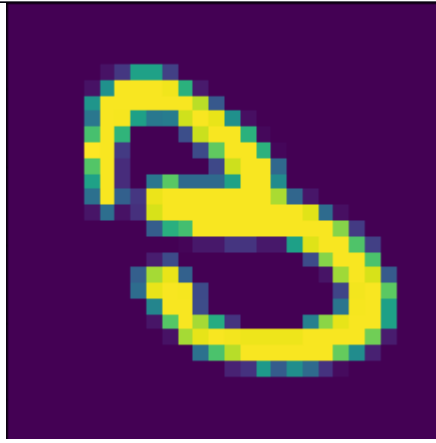
In conclusion, I would like to say that for successful classification this dataset, it was not necessary to implement an Adam's version of updating weights and complex structure of the neural network. Vanilla stochastic gradient descent with not very big batches and the neural network with three fully connected layers coped well with this role in just 10 epochs.

Some examples of classification:

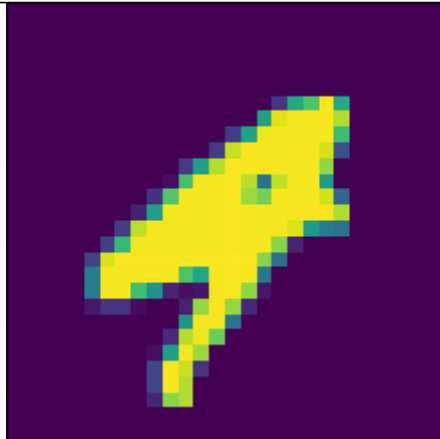
Wrong classification	Correct classification
 <p>Prediction label: 7 Ground truth label: 2</p>	 <p>Prediction label: 4 Ground truth label: 4</p>



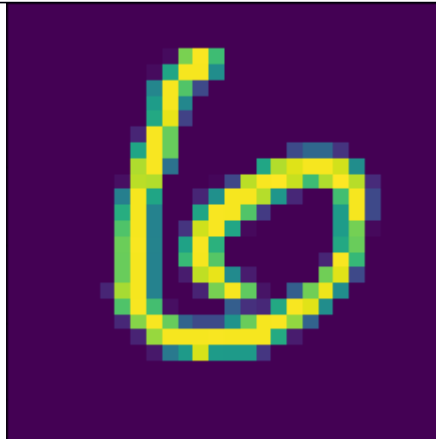
Prediction label: 0
Ground truth label: 6



Prediction label: 3
Ground truth label: 3



Prediction label: 9
Ground truth label: 4



Prediction label: 6
Ground truth label: 6