## Report

During implementing and preliminary tests for debugging I worked with an extremely small toy data set (4 samples of (4,3) arrays.

After successfully setting up the network I started testing with the predefined settings.

First tests showed that Stochastic Gradient Descent converged much faster and was clearly preferable to Gradient Descent as expected. So all further tests were performed with SGD, with varying batch sizes.

Training with more than 30 epochs did not lead to further improvements, with most settings reaching a saturation of the validation error and very small or zero training error between 15 and 25 epochs.

The duration of training on 30 epochs varied between 3 and 6 minutes, depending on the settings.

**Observations during tests with the network:**

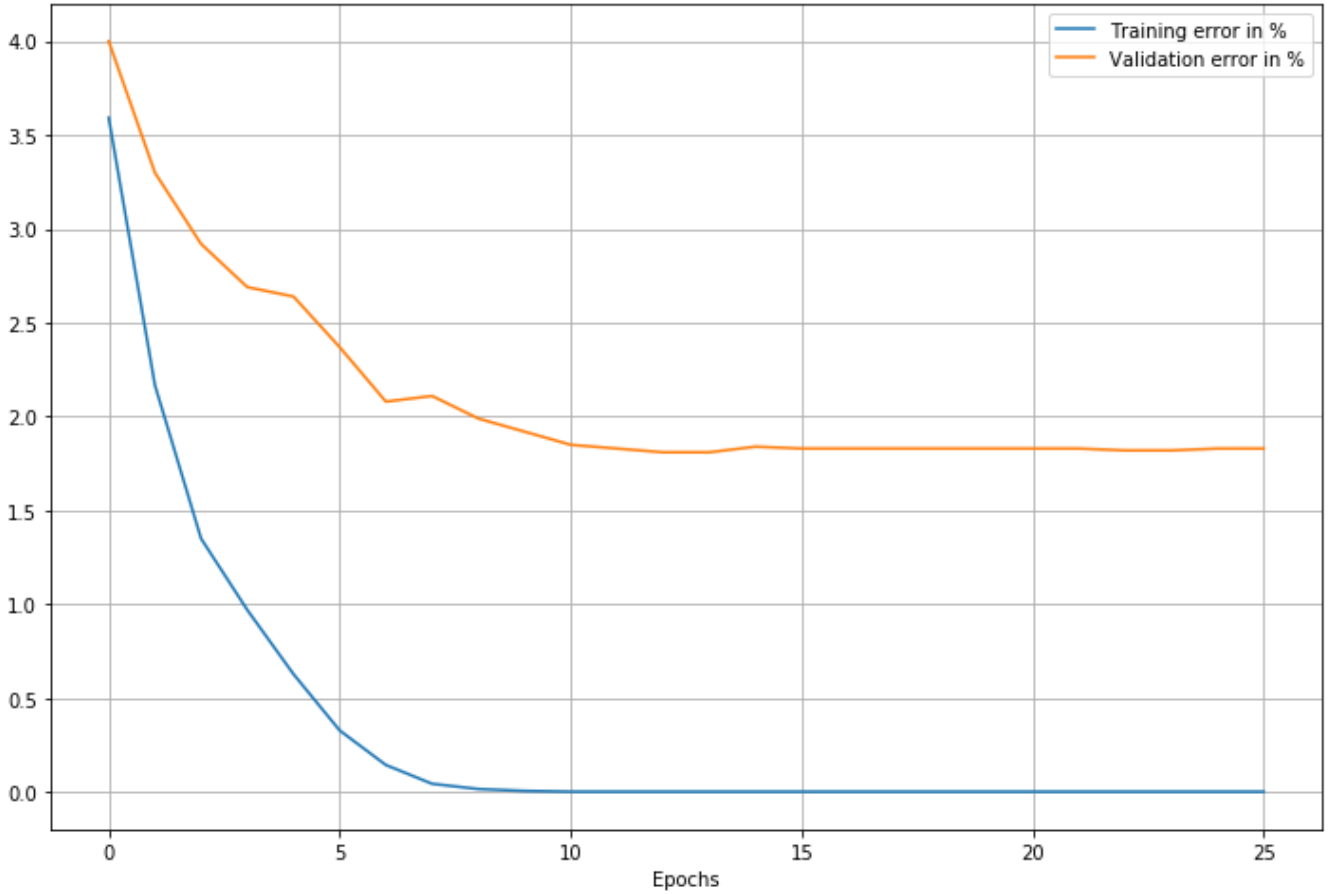
- Tanh Activation led to lower error rates than relu and sigmoid, with not significant difference in speed
- Smaller Batch sizes generally lead to faster convergence, as expected.
- A Batch size of 1.000 led to poor results (3,1% validation error, slow convergence)
- A Batch size of 20 led to a high error rate (4% validation error), after very fast convergence (below 10 epochs).
- Batch sizes of 60 and 120 yielded similar results, 60 slightly better, so 60 was chosen for the search for optimal settings
- Tests where started with 3 fully connected layers (without the last layer with 10 nodes and without activation)
- Adding an additional layer with 300 units did not improve the results, so further tests were conducted with the initial 3 plus 1 layer.
- But all 3 layers with 200 instead of 100 brought fast convergence to 1,9x% (zero training error) and a slightly better error rate.
- With 400 units per layer training gets painfully slow, and no better performance (even a slight decrease)
- At 50 units per layer a significant decline in performance was observed.
- After training with the best found configuration in the final run, the Test Error war slightly higher then the validation error (2,12% instead of 1,83%).

Jochen Sautter, Exercise 1, Deep Learning Lab WS 2017, 03.11.2017

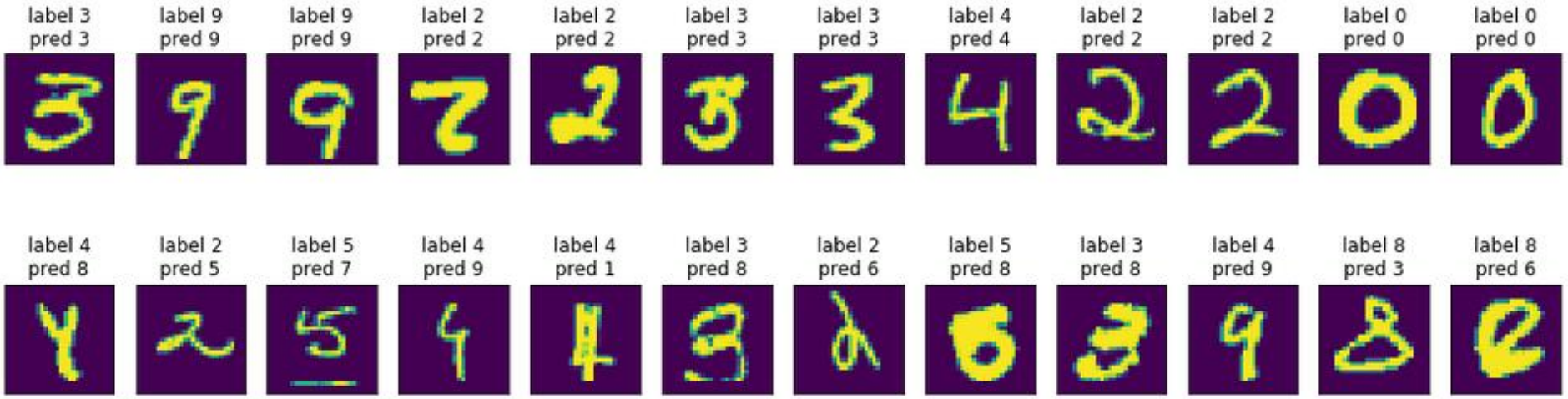
**Overview over the test runs**

| Run | Layer 1 | Layer 2 | Layer 3 | Layer 4 | Layer 5 | L-rate | Epochs | Batch size | Train_err | valid_err | Stop converging | remarks |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 100 tanh | 200 tanh | 100 tanh | | 10 none | 0.4 | 30 | 120 | 0.01% | 2,11% | 30 | |
| 2 | 100 relu | 200 relu | 100 relu | | 10 none | 0.4 | 30 | 120 | 0.53% | 2,8% | 20 | Stop converging |
| 3 | 100 tanh | 200 tanh | 100 tanh | | 10 none | 0.1 | 30 | 120 | 0.05% | 2,54% | 30 | |
| 4 | 100 tanh | 200 tanh | 100 tanh | | 10 none | 0.4 | 30 | 1000 | 0.59% | 3.1% | 30 | Slow convergence |
| 5 | 100 tanh | 200 tanh | 100 tanh | | 10 none | 0.4 | 30 | 60 | 0,00 | 2.05% | 20 | Best pick so far |
| 6 | 100 tanh | 200 tanh | 300 tanh | 100 tnh | 10 none | 0.4 | 30 | 60 | 0,00 | 2.07% | 20 | |
| 7 | 100 tanh | 100 tanh | 100 tanh | | 10 none | 0.4 | | 20 | 2,2% | 4,0% | 10 | |
| 8 | 100 sigm | 100 sigm | 100 sigm | | 10 none | 0.4 | | 60 | 0.06% | 2.6% | 30 | Slow convergence |
| 9 | 200 tanh | 200 tanh | 200 tanh | | 10 none | 0.4 | 30 | 60 | 0,00 | 1,95% | 12 | Best pick so far |
| 10 | 400 tanh | 400 tanh | 400 tanh | | 10 none | 0.4 | 15 | 60 | 0,00 | 2,25% | 15 | |
| 11 | 50 tanh | 50 tanh | 50 tanh | | 10 none | 0.4 | 15 | 60 | 0,4% | 3,3% | 20 | |
| 12 | 200 tanh | 200 tanh | 200 tanh | | 10 none | 0.4 | 30 | 60 | 0,002% | 1,83% | 12 | Final run |

**Training and Validation errors at the final run**



**Correctly and incorrectly classified Samples**




**Final remarks**

It seems, that the Network cant be significantly improved by more layers / units or other configuration of the given parameters (Activations Functions, Batch Size).

But a look at the misclassified images shows, that most of those can be identified quite safely by a human. So there should be room for significant improvement. This could eventually be achieved by other approaches, like convolutional networks.