

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
10	400 tanh	400 tanh	400 tanh		10 none	0.4	15	60	0,00	2,25%	15	
10	50 tanh	50 tanh	50 tanh		10 none	0.4	15	60	0,4%	3,3%	20	

### Observations:

- Training error can be brought to zero, at validation error of ~2,2%. Slight progress in validation error after this point.
- Tanh works better than relu and sigmoid
- Smaller Batch sizes converge faster
- Batch size 1.000 and 20 both bad results
- No benefit from adding additional layer with 300 points
- But all 3 layers with 200 instead of 100 brought fast convergence to 1,9x% (zero training error)
- With units 400 per layer gets painfully slow, and no better performance
- 50 units per layer significant decline in performance
- After training as in run no 9, resulting in 2,04% validation Error, the Test Error war nearly identical with 2,05%