Assignment 2

Computational Intelligence

Team Members		
Last name	First name	Matriculation Number
Merdes	Malte	01331649
Riedel	Stefan	01330219

1 Regression with Neural Networks

1.1 Simple Regression with Neural Networks

a) For two hidden neurons (figure 2) we can clearly observe underfitting. For 8 (figure 2) and 40 hidden neurons (figure 3) we can fit the data quite accurately.

As an experiment we also plotted the regression for $n_h = 1000$ and observed overfitting. However, this relates to the number of iterations ($max_{iter} = 200$) which is not enough to fit all parameters. Increasing the number of iterations to 5000 which results in an accurate fitting of the data.

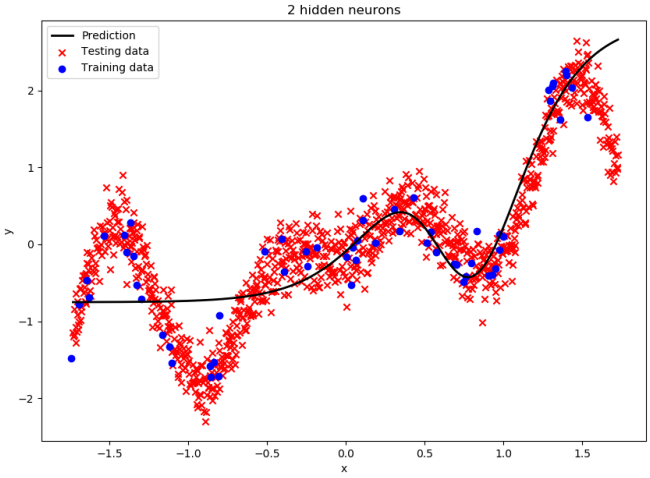

Figure 1: Regression with 2 hidden neurons

b) For 40 hidden layers:
Minimum MSE (training set): 0.04059
Maximum MSE (training set): 0.05428
Mean MSE (training set): 0.04537
Std. deviation MSE (training set): 0.004247
Minimum MSE (test set): 0.12713

The min MSE is not obtained for the same seed, the min on the training set is at index 7, the test set at index 0.
A validation set is needed to make sure that the seed is also optimal for unseen data.
Concerning linear and logistic regression the convex MSE function has one global minimum that can be found independently of the seed (initial parameters). Re-

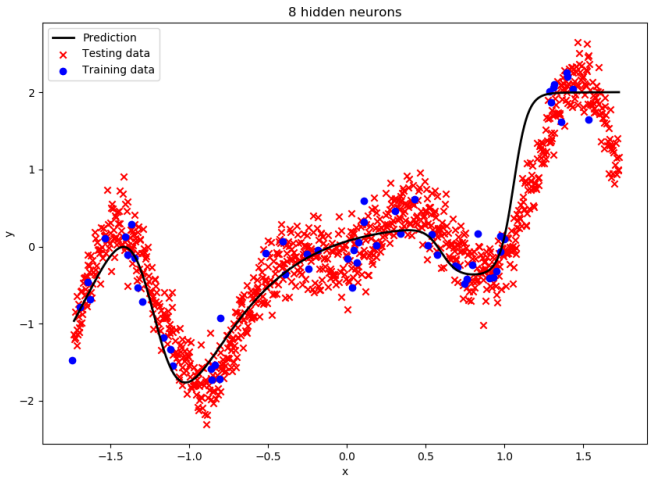

Figure 2: Regression with 8 hidden neurons

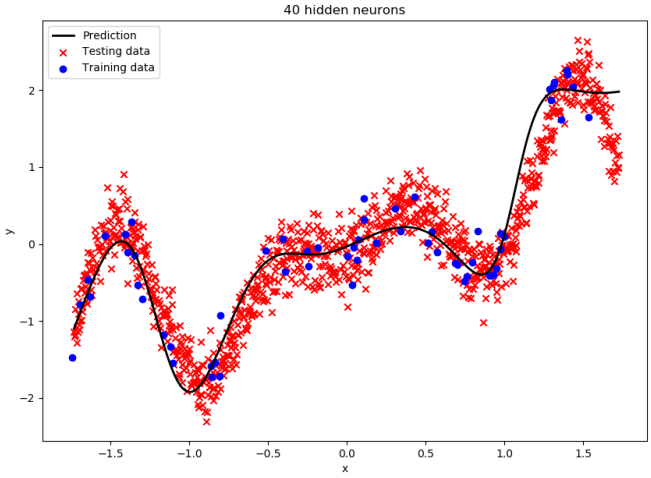

Figure 3: Regression with 40 hidden neurons

garding neuronal networks, one can get stuck in a local minimum depending on the seed, because the MSE function is not convex.

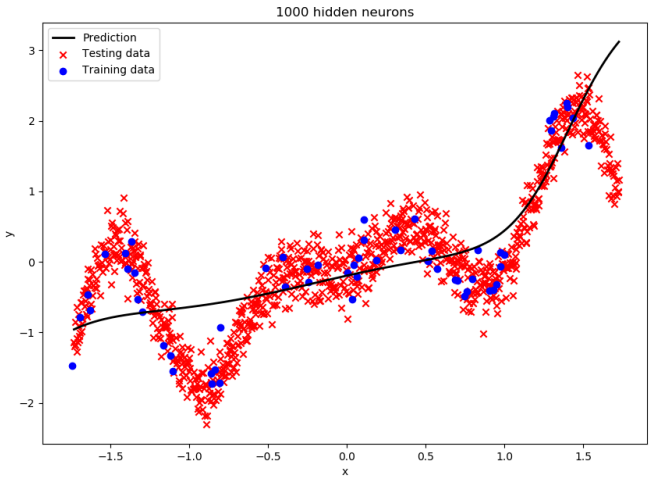

Figure 4: Regression with 1000 hidden neurons, 200 iterations

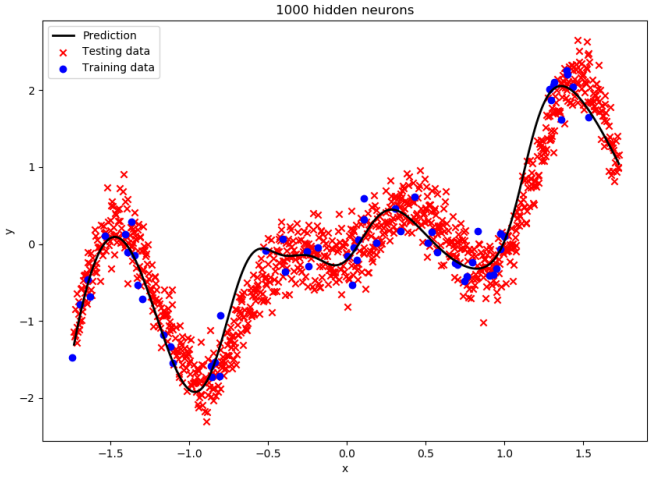

Figure 5: Regression with 1000 hidden neurons, 5000 iterations

The source of randomness through SGD is due to the fact that after each sample from the training set, we use the calculated error gradient for weight

update.
The randomness of the initial parameters will persist if SGD is replaced by standard Gradient Decent.

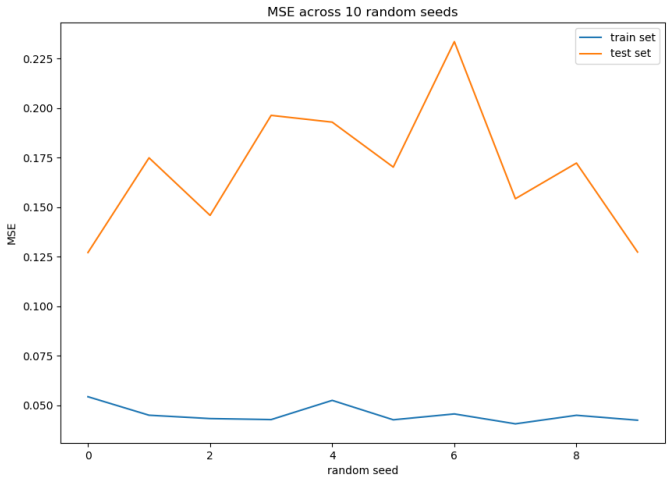

Figure 6: Training and test MSE as a function of random seeds (40 hidden neurons, 200 iterations)

c) Unlike in task a) with only 200 iterations (default tolerance of 1e-4) we can now clearly see overfitting the data (testing MSE ¿¿ training MSE) as we allow 10000 iterations and a tolerance of 1e-8 (figure 8).
The best number of hidden neurons independently of the choice of the random seed is 6, as seen in figure 7.

The lbfgs solver performs best in terms of MSE.
The stochastic algorithms tend to overcome overfitting as they only learn from one sample at a time.
We can see that the lbfgs manages to rapidly converge also for larger numbers of hidden neurons, which suggests a real life application (deep neural networks).

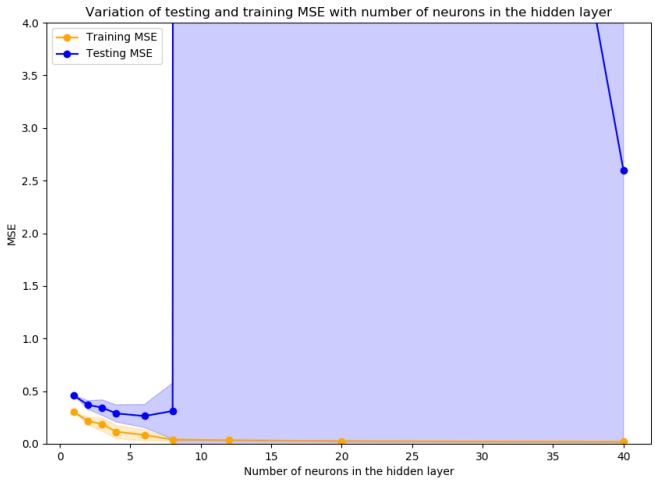

Figure 7: Training and test MSE as a function of the number of neurons in the hidden layer (10000 iterations, tolerance 1e-8)

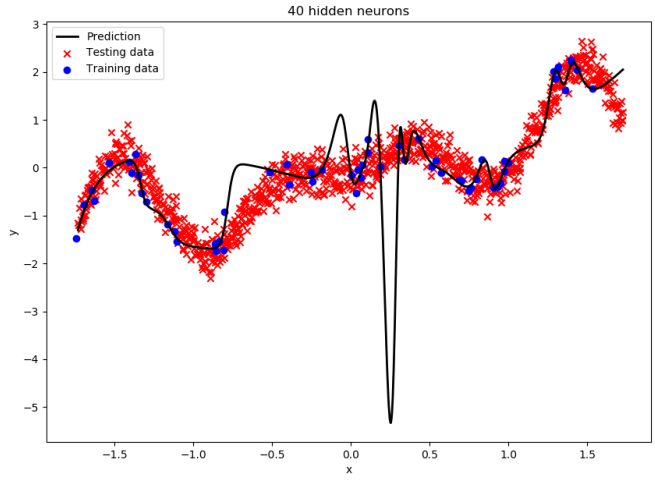

Figure 8: Learned function for 40 hidden neurons (10000 iterations, tolerance 1e-8)

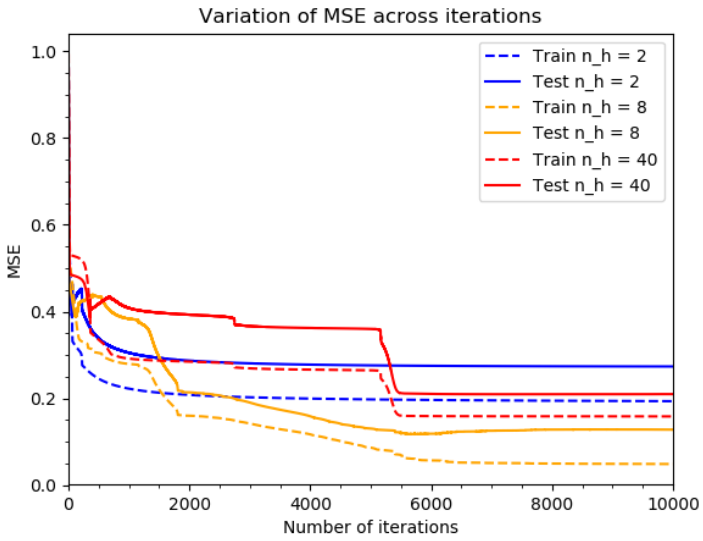

Figure 9: Variation of MSE across iterations for lbfgs solver

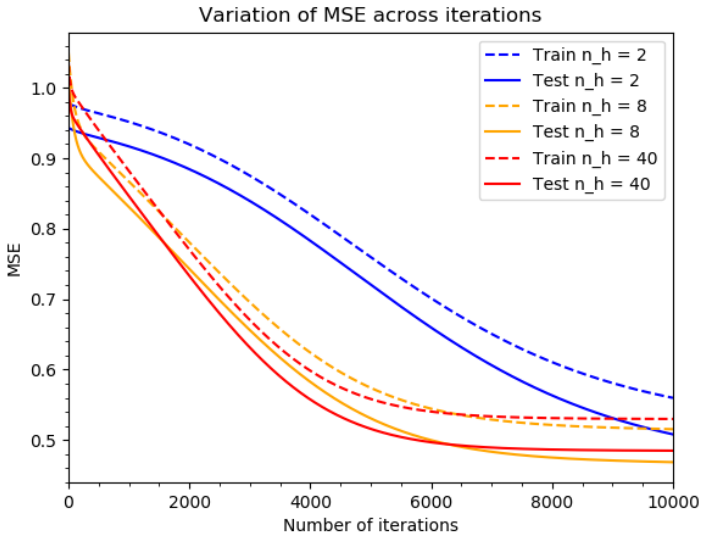

Figure 10: Variation of MSE across iterations for sgd solver

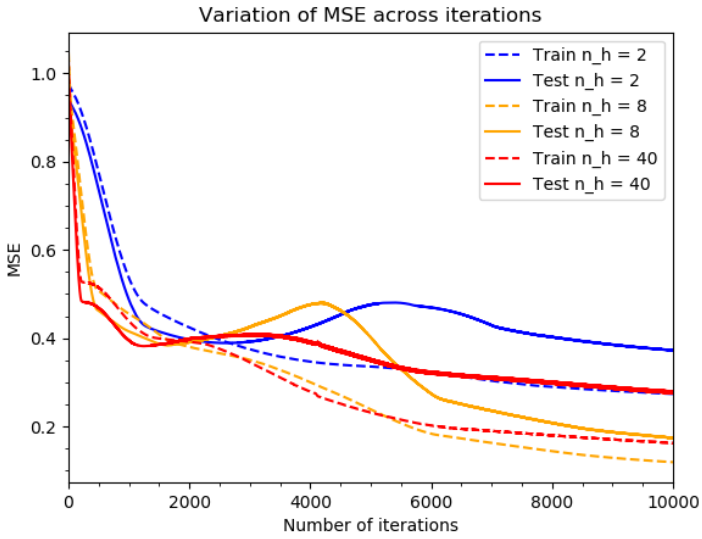

Figure 11: Variation of MSE across iterations for adams solver

1.2 Regularized Neural Networks

a)
We now trained the network with different values of the regularization parameter α and 40 hidden neurons.

Figure 12 shows plots of the variation of MSE with the value of α. It shows that $\alpha = 10^{-2}$ gives the lowest mean value, while $\alpha = 10^{-3}$ gives a lower standard deviation around a similarly low mean value. Regularization prevents overfitting as it keeps the higher complexity parameters low (close to zero) in order to minimize the penalty term introduced to the cost function.

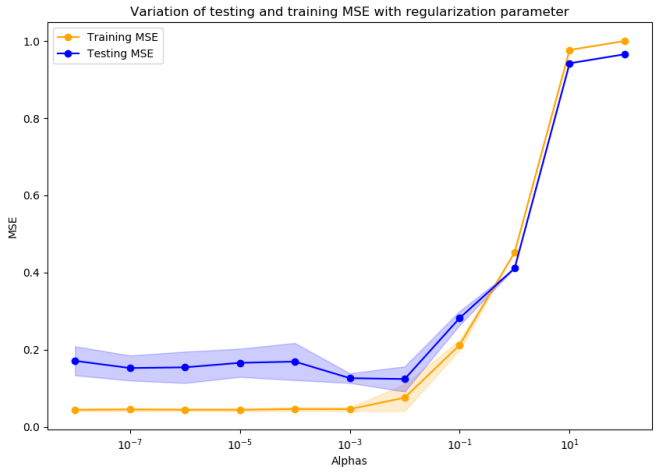

Figure 12: Variation of MSE as a function of the regularization parameter α

b)
Figure 13 shows a comparison of the errors on the test sets at the last training iterations, at early stopping and when it is minimal. Comparing the test MSE we do not expect the early stopping to happen at the same iteration number for the different random seeds, as all represent different starting points on the error surface. We could verify this by printing the early stopping index.
Taking the overall minimum of the validation mse assures that we find the global minimum, but is only possible in offline computation. The standard approach allows for online computation, but can result in a local minimum.

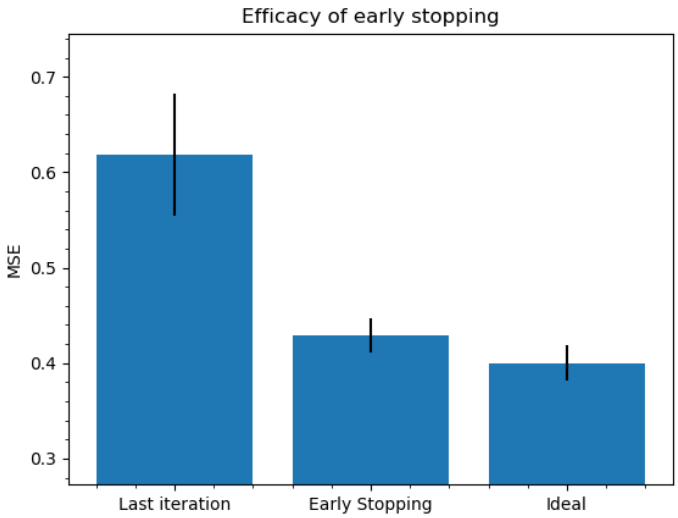

Figure 13: Comparison of test MSE (after all iterations, early stopping and overall minimum)

c)
optimal seed: 8

train mse @ optimal seed: 0.03130562290097121

test mse @ optimal seed: 0.11975190986310334

valid mse @ optimal seed: 0.07377662786946931

Mean training error: 0.03022203966371964

Std deviation training error: 0.002524435007833558

Mean testing error: 0.12717236776033153

Std deviation testing error: 0.0043720599363846055

Mean validation error: 0.08385110336627687

Std deviation validation error: 0.005406117225450046

2 Face Recognition with Neural Networks

2.1 Pose Recognition

pose straight: class 1
pose left: class 2
pose right: class 3
pose up: class 4

Confusion matrix: $\begin{bmatrix} 122 & 4 & 6 & 9 \\ 2 & 135 & 1 & 3 \\ 1 & 0 & 133 & 4 \\ 8 & 3 & 2 & 131 \end{bmatrix}$

Looking at the diagonal values of the confusion matrix, we find that the pose left and and right can be classified more accurately than pose straight and up. The neural network tends to confuse straight and up, as those poses are represented by similar picture data.

Figure 14: Hidden layer weights: Each block represents the weights connected to one of the six hidden neurons

Figure 14 shows that the weights representing the face contours are very important. In contrast, the "skin area" weights are not as large, as the skin are doesn't contain significant information. Obviously, the corners of these pictures are irrelevant, as all the persons are centered. Therefore the "corner weights" are always small.

2.2 Face Recognition

Figure 15 shows that there is one best performing seed that achieves 96% testing accuracy. Most seeds achieved 95% accuracy.
Different performances across seeds can be explained by different initial weigths, of which not all converge to the global error minimum. This assumption could be confirmed by increasing the iteration number to 5000, which still yields the same results (no further optimized convergence).

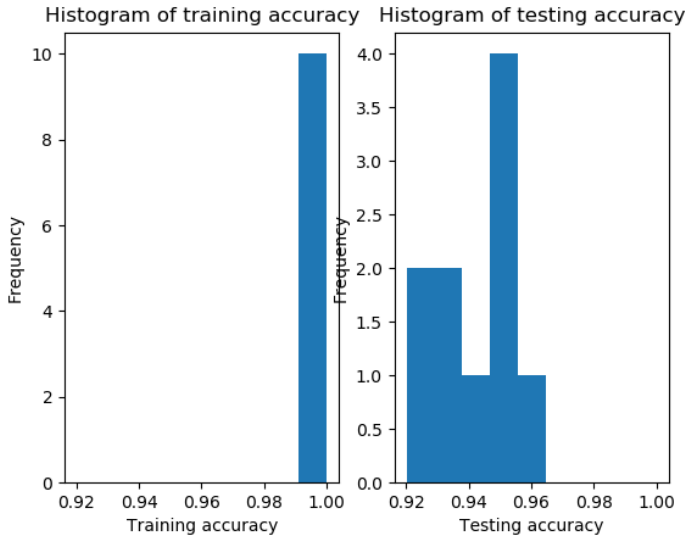

Figure 15: Histogram of training and testing accuracy for 10 different random seeds.

Confusion matrix for for the best network (random state = 1):

[27 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 29 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 28 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 1 0 27 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 24 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 28 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 8 19 0 0 1 0 0 0 0 0 0 0 0 0 0]
[0 6 0 0 0 0 0 23 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 29 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 28 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 28 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 23 0 0 0 2 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 29 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 27 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 29 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 29 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 27 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 29 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 29 0]
[0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 28]

Figure 16: Three misclassified images of our best performing network.

Figure 16 shows three example pictures of misclassified persons. We can see that sunglasses and looking up make it harder for the network to identify characteristic features and classifiy correctly.