Artificial Neural Networks

Žiga Trojer - 63200440

Implementation of Artificial Neural Networks.

We successfully implemented ANN for classification and regression. For both, we used the Sigmoid function on all layers, except the last one. Implementation differs in activation on the last layer, how we calculate the loss function and in number of neurons in the last layer.. In the classification, we used Softmax activation for the last layer, and the loss function was `log loss`. In the last layer, there are n neurons, where n is number of classes - for each observation, we get probabilities for each class. For regression, we used a linear function with mean square error loss. There is only one neuron in the final layer, which return continuous values.

Numerical verification of gradients.

Before each fit, we approximated the gradients and compared them with the gradients, used in the network. We approximated the gradients of the loss function by computing the following expression:

$$\frac{f(x+\varepsilon)-f(x)}{\varepsilon}$$

for some small ε. This expression was used on each direction. We then verified if this expression coincide with the calculated gradient for the correct direction. If the expressions differs in some direction by some pre-defined tolerance δ, we stop the calculations. For each direction, we check the correctness 10 times, so we are very certain that the gradients are computed correctly.

Comparison with other methods.

We were given two different datasets, containing housing data for regression and classification. On the first one, we performed regression with ANN and the kernelized ridge regression. For both methods, we used optimal parameters, gotten from the CV (in this assignment, we only performed CV for the ANN, since we already know optimal parameters for support vector regression from the previous assignment). First, we shuffled rows and we splitted data into train an test - 80:20. We performed 5-fold cross validation on train dataset, where we standardized each dataset for each fold separately. From CV we got the parameters, that gives the lowest loss - `units` $=[10,20,10]$ and $\lambda=0.1$. Average loss from cross validation was 27.90. We also tried to estimate the CI95, by splitting the test data into 5 folds. 95% confidence interval for ANN regression was $[0,175.15]$ with the average MSE 50.15 and $[0,155.14]$ for SVR with average MSE 59.11. If we would not estimate the CI, we would say that ANN performs better, but as we can see, the SD for ANN is a lot higher than for SVR. For the classification problem, we compared ANN for classification and multinomial logistic regression. Similarly as before, optimal parameters for ANN were chosen with 5-fold CV. Best parameters are: `units` $=[10]$ and $\lambda=0.001$, where the loss was 0.321. CI95 for ANN is $[0.060,0.538]$ with the average log loss 0.299. CI95 for multinomial logistic regression: $[0.134,0.448]$ with the average log loss 0.291.

Huge dataset.

We standardized both datasets and then on the train, performed CV with 5-cross validation. Best parameters were `units` $=[20,20]$ and $\lambda=0.0001$ (log loss = 0.558). We tried 5 different structures of hidden layers and 5 different regularization parameters. CV process took about 4 hours to complete (125 different training and predictions). We think that predictions would be better if we would have more neurons in the layers, but then the calculation would take too much time, so we did not try those settings. We also did not calculate the uncertainty (95CI for the results), as calculation would take too much time. Log loss on the whole train dataset is 0.474. Predictions for the test dataset are in the `final.txt` file.