HW6 report

Maša Kljun, 63170011

Results

Task 1

We implement this problem in `hw6.py` and implement it in a way that classification and regression ANNs share a lot of code. The only difference between the models is in the activation functions of the last layer and in the loss functions. In classification there are *n_unique* output neurons, where *n_unique* is the number of unique classes in the target variable, while in regression there is only one neuron in the output layer. The activation of the last layer in classification is the softmax function, which enables us to get probabilities of each class, while the activation of the last layer in regression is the identity function. As mentioned before, the second difference is in the loss functions. In classification we use cross entropy, while in regression we use mean squared error. Most of the other code (computing gradient, etc.) is shared among both models.

Task 2

We verify the gradient with the help of the following definition of the derivative:

$$f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}. \tag{1}$$

We loop through the whole vector of weights and biases wb and in each iteration obtain vector $\bar{wb}^{(i)}$, where the i-th component of vector wb is changed by adding a small value $h = 1e-06$, while leaving other components of the vector as they are. We then use the above Equation 1 and compute the i-th component of the estimated gradient:

$$\nabla f^{(i)}(wb) = \frac{f(\bar{wb}^{(i)}) - f(wb)}{h}. \tag{2}$$

We compare the obtained component $\nabla f^{(i)}(wb)$ with the true value of derivative obtained by the backpropagation function $\nabla f_{bp}^{(i)}(wb)$. If the absolute difference between $\nabla f^{(i)}(wb)$ and $\nabla f_{bp}^{(i)}(wb)$ is smaller than predefined threshold $thresh = 1e-02$ we conclude the gradient for i-th component is correct. If gradient for all components is correct, we conclude the gradient is correct. We verify the gradient on the initial set of weights everytime we fit the model.

Task 3

We apply `housing2r` and `housing3` data sets to the regression and the classification models, respectively. We use 5-fold cross-validation for both data sets in order to find the best values for regularization parameter λ and the number of hidden layers $units$. We choose some options for these two parameters manually, by checking which settings work, and then apply 5-fold cross-validation. We perform training and optimal parameter search on 80% of the data and then evaluate the model on the remaining 20% of the data. Also note that we shuffle the data before $80/20$ split and scale them after using StandardScaler.

For the regression model we see from the CV that the best setting is $units : [10]$ (one hidden layer with 10 nodes) and $\lambda = 0.01$. We then evaluate the obtained model with the test set and get $MSE = 0.53$. We compare the obtained results with the Support Vector Regression (SVR) with RBG kernel and the following settings: $\sigma = 5, \lambda = 1, \varepsilon = 1$.

Discussion

Similarly to the previous homework we believe that using a model with RBF kernel is a better choice opposed to using Polynomial kernel, as we can achieve lower RMSE with it. We also believe that SVR is a slightly better model than Kernelized ridge regression (KRR) for our task, as we achieve much lower RMSE especially when using Polynomial kernel on the *housing2r* dataset. RMSE obtained by using RBF kernel does not differ much from that achieved in the previous homework. We believe that the key advantage of KRR is that it is easier to implement due to its closed-form solution and it is also fitted faster (probably due to the fact that we had a small-medium sized dataset). On the other hand, the solution of SVR is sparse (and the one of KRR is non sparse), which is an advantage of SVR that was shown in faster prediction time.