MLDS: Homework 6

Andraž De Luisa

May 17, 2020

1. The implementation for both neural networks, classification and regression, share quite a lot of code. They differ just in the loss functions and in the activation functions of the last layers. For regression, the output layer is a single neuron with the identity function as the activation, while the optimized loss function is the mean squared error. For classification, the output layer is formed by m neurons (m is the number of classes in the target variable) with the softmax function as the activation (hence their values represent the probability of each class), while the optimized loss function is the cross-entropy. Therefore the function and its gradient can be computed in the same way for both types of networks, only the last layer must be different (i. e. the first computed layer during the back propagation).

2. For the numerical verification of the correctness of the gradient the definition of the derivative is used: $f'(x) = \lim_{h->0} \frac{f(x+h)-f(x)}{h}$. The loss function has several variables, therefore we apply this definition in each direction of the n-dimensional space. We choose $\epsilon > 0$, $tol > 0$ and a random $x_0 \in \mathbb{R}^n$. Then we compute the gradient at x_0 and its approximation with $\frac{f(x_0+\epsilon \cdot e_i)-f(x_0-\epsilon \cdot e_i)}{2\cdot\epsilon}$ for each $i \in \{1, \dots n\}$. If

   $$\left|\nabla f(x_0)_i - \frac{f(x_0 + \epsilon \cdot e_i) - f(x_0 - \epsilon \cdot e_i)}{2 \cdot \epsilon}\right| > tol$$

   for some i, then the gradient isn't correct. We repeat the same procedure for 10 random $x0$ (to decrease the probability of an error). This numerical verification is performed automatically before the fitting of a model.

3. For both datasets a 5-fold cross-validation is used for the choice of the best user defined parameters (size of hidden layer and regularization parameter λ) in the neural networks.

   - Classification ('housing3' dataset): we compare the neural network with a simple logistic regression (without polynomial expansion). The evaluated loss (cross-entropy) on the test set (20% of the data) is 0.326 for the logistic regression and 0.240 for the neural network (single hidden layer with 5 neurons, regularization parameter $\lambda = 0.001$).
   - Regression ('housing2r' dataset): we compare the neural network with the support vector regression (). The evaluated loss (mean squared error) on the test set (20% of the data) is 41.490 for the support vector regression and 42.136 for the neural network (single hidden layer with 10 neurons, regularization parameter $\lambda = 0.1$).

4. Data from both 'train' and 'test' dataset were standardized (with mean and standard error estimated using only the training data). The best parameters for the neural network were chosen with a 5-fold cross validation (a single hidden layer with 20 neurons, regularization parameter $\lambda = 0.0001$). The whole cross validation procedure run in 53 minutes (20 combinations of parameters, therefore 100 fittings of the network), while the fitting of the final model takes 110 seconds. The cross-entropy of the final model evaluated on the training set is 0.5027, while the estimated cross-entropy for the test dataset (estimated through cross-validation) is 0.5588. The predictions are saved in the file 'final.txt'.