Assignment 2
Optimisation for Computer Science
Group 33

Sven Celin & Gabriela Ožegović

December 2019

1 Tasks

1. The size of the matrices is:
   W^0 : N_H x 3
   W^1 : 4 x N_H
   b^0 : N_H
   b^1 : 4 x 1

   N_H is the number of hidden layers.
   3 is input dimension.
   4 is output dimension.
   1 because it is a vector.

$$N_O = 4, N_I = 3$$
$$\begin{bmatrix} z_1^{(1)} \\ \vdots \\ z_{N_H}^{(1)} \end{bmatrix} = \begin{bmatrix} W_{1,1}^{(0)} & \dots & W_{1,N_I}^{(0)} \\ \vdots & \ddots & \vdots \\ W_{N_H,1}^{(0)} & \dots & W_{N_H,N_I}^{(0)} \end{bmatrix} \begin{bmatrix} a_1^{(0)} \\ \vdots \\ a_{N_I}^{(0)} \end{bmatrix} + \begin{bmatrix} b_1^{(0)} \\ \vdots \\ b_{N_H}^{(0)} \end{bmatrix}$$
$$\begin{bmatrix} z_1^{(2)} \\ \vdots \\ z_{N_O}^{(2)} \end{bmatrix} = \begin{bmatrix} W_{1,1}^{(1)} & \dots & W_{1,N_H}^{(1)} \\ \vdots & \ddots & \vdots \\ W_{N_O,1}^{(1)} & \dots & W_{N_O,N_H}^{(1)} \end{bmatrix} \begin{bmatrix} a_1^{(1)} \\ \vdots \\ a_{N_H}^{(1)} \end{bmatrix} + \begin{bmatrix} b_1^{(1)} \\ \vdots \\ b_{N_O}^{(1)} \end{bmatrix}$$

2. N/A

3. Parameters are initialized in method initNormalDistribution. There was an idea to make normal distribution for values of the $W^{(0)}, W^{(1)}, b^{(0)}, b^{(1)}$ for values $\mu = 0$ and $\sigma = 0.05$.

4. Forward pass is implemented in neural network with one hidden layer. Dimensions of that hidden layer are N_H and after passing trough Softmax and Softplus functions we get a and z.

5. Backward pass using back-propagation method is implemented in back-prop function. After passing trough back-propagation function we get derivatives of the parameters and we use them to check results with $\tilde{y}$.

6. We make list of vectors from y and $\tilde{y}$ and after that we iterate trough that list of vector pairs. For 1000 iterations we search for steepest descent function. Step we choose is $k^k = 0.01$.

7.

$$\mathcal{A} = \frac{1}{S}\sum_{n=1}^{S}\delta(y^s, \tilde{y}^s) \quad \text{with } \delta(u,v) = \begin{cases} 1 & \text{if } u = v \\ 0 & \text{else,} \end{cases}$$

We train data $W^{(0)}, W^{(1)}, b^{(0)}, b^{(1)}$. Using them we run forward feed from where we get $\tilde{y}$. $\tilde{y}$ is result of one pair of the vectors. Then we compare $\tilde{y}$ with y test in every iteration in vector list. From that we get accuracy.

8. We train parameters $W^{(1)}, b^{(1)}$ using x-train and y-train. After training the parameters we put last feed forward where we check accuracy of x-test and y-test and we get numbers that signifies difference.

9. We created 3D scatter plot in drawPlot method that gives us visual representation of trained and real model.
Dots which are colored with red are x^s dots with corresponding $x^s = 0$ true values.
Dots which are colored with blue are x^s dots with corresponding $y^s = 1$ true values.
Dots which are colored with green are x^s dots with corresponding $y^s = 2$ true values.
Dots which are colored with yellow are x^s dots with corresponding $y^s = 3$ true values.
In case of correct value there would be plus signs matching the points in correct places.

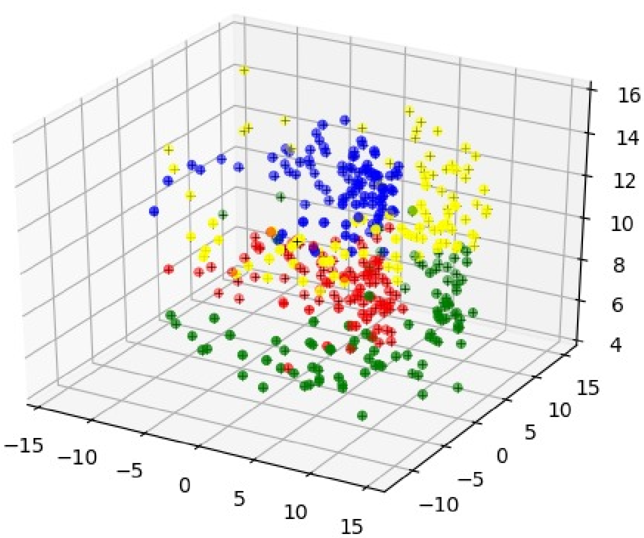

Figure 1: Graph