Deep Learning

Bastien BRIER
bastien.brier@student.ecp.fr

March 5, 2017

Assignment 3

1 Training Neural Networks

1.1 Understanding the code

We found the symbols associated to the variables slide 142 in the 4th presentation.
- `d_output_d_activation` = $\frac{\partial \hat{y}_c}{\partial b_k}$
- `d_loss_d_activation` = $\frac{\partial L}{\partial b_k}$
- `d_loss_d_output` = $\frac{\partial L}{\partial \hat{y}_c}$

We are here applying the L2-regularization cost on the weights to modify them accordingly during the back-propagation process, a larger regularization means that we decrease the importance of each neuron on the output. We start from 2 because we added a constant vector and we do not want to regularize it.

We are here computing the error term for each output, which means how much a node is responsible for the error. In the case of one hidden layer, it is first computed between the output and layer, and then layer and input. This corresponds to the slide 149 of the 4th presentation.

1.2 Extension: Using Softmax

In the provided code, we modified the objective function from the sigmoid to the softmax. For this, we implemented the function softmax.m and modified parts of the code nnet.m that are indicated by doSoftMax. Then, if doSoftMax is set to true, the softmax is used in the neural network.

1.3 Extension: Using ReLUs

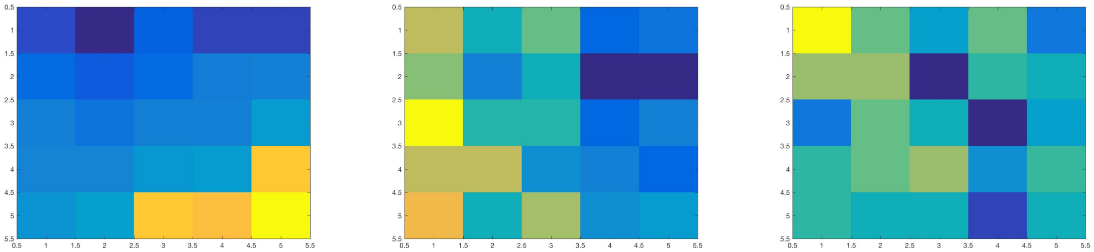
We modified the code by implementing a new activation function, a rectified linear unit (ReLU). We implemented the functions ReLU.m and reluGradient.m and also modified parts of the code nnet.m that are indicated by doReLU. Similarly, is this is set to true, the ReLU activation is used.

1.4 Experiment: Cross-validate over regularization cost and number of intermediate neurons

Thanks to the template provided, we implemented a 10-fold cross-validation on the regularization cost λ and the number of neurons N for three different models : Softmax, ReLU and Softmax-ReLU.

We obtained really satisfactory results for the three models. The graphs corresponds to the results for the different parameters: the ordinate is the regularization cost : the lower, the higher the regularization. The abscissa is the number of neurons : from left to right correspond to 5, 10, 20, 50 and 100 neurons. For a better comprehension, a darker and bluer color corresponds to a lower error.

Figure 1: Cross-validation error for softmax (left), relu (middle) and softmax-relu (right) combinations.


0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 5.5
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 5.5
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
0.5 1 1.5 2 2.5 3 3.5 4 4.5 5 5.5