Statistical Methods for Machine Learning
Assignment 3

Tudor Dragan (xlq880)
Nicolae Mariuta (rqt629)
Gabriel Carp (slp670)

March 20, 2015

III.1 Neural Networks

III.1.1 Neural network implementation

We have constructed a neural network with a hidden layer which contains three neurons and for the forward propagation we have implemented the following functions:

(i) for applying the weights between the input neurons and the neurons in the hidden layer

$$Z^{(2)} = XW^{(1)}$$

(ii) for applying the activation function in the hidden layer

$$a^{(2)} = f(Z^{(2)})$$

(ii) for applying the weights between the neurons in the hidden layer and the output neuron

$$Z^{(3)} = a^{(2)}W^{(2)}$$

Derivative of $h(a)$

$$h(a) = \frac{a}{1 + |a|}$$

The derivate of the $h(a)$ function has been calculated by applying the following formula:

$$\left(\frac{f}{g}\right)' = \frac{f'g - g'f}{g^2}$$

$$h(a)' = \frac{1}{1+|a|} - \frac{a}{(1+|a|)^2} \tag{1}$$
$$= \frac{1+|a|}{(1+|a|)^2} - \frac{a}{(1+|a|)^2} \tag{2}$$
$$= \frac{1+|a|-a}{(1+|a|)^2} \tag{3}$$
$$= \frac{1}{(1+|a|)^2}. \tag{4}$$

III.1.2 Neural Networks Training

We heavily modified our functions from the first try but haven't managed to get accurate results.

III.2 Support Vector Machines

III.2.1 Data Normalization

In order to normalize data to zero mean and unit variance, the mean, μ, and variance, σ^2 of the data is determined. From there the following mapping is used:

$$f_{norm}(\mathbf{x}) = \sum_i \frac{x_i - mu_i}{\sigma_i}\mathbf{x}_i \tag{5}$$

where x_i is the ith component of the vector $\mathbf{x}$. $\mathbf{x}_i$ here denotes the unitvector satisfying $x_i = \mathbf{x} \cdot \mathbf{x}_i$ and σ is the standard deviation equal to the square root of the variance.

The means for the train data set is represented by the *Means* variable and the standard deviation is displayed in $Stds$. For the normalized test data the means can be seen in *testMeansNorm* and the standard deviations in $testStdsNorm$.

means	stds	test_means_norm	test_stds_norm
155.9604	44.3036	-0.0786	0.8557
204.8212	98.1520	-0.1580	0.8455
115.0586	45.7557	0.0556	0.8931
0.0060	0.0040	0.1132	1.4108
0.0000	0.0000	0.0716	1.2908
0.0032	0.0024	0.0869	1.4618
0.0033	0.0023	0.1157	1.3865
0.0096	0.0071	0.0870	1.4621
0.0277	0.0159	0.2490	1.3311
0.2624	0.1627	0.2452	1.3524
0.0147	0.0087	0.2296	1.3105
0.0166	0.0101	0.2509	1.3334
0.0220	0.0133	0.3166	1.4799
0.0440	0.0260	0.2296	1.3105
0.0226	0.0298	0.1491	1.6319
22.0007	4.0632	-0.0568	1.1666
0.4948	0.0105	0.0736	1.0405
0.7157	0.0558	0.0868	0.9754
-5.7637	1.0304	0.1548	1.1030
0.2148	0.0758	0.3107	1.1674
2.3658	0.3694	0.0874	1.0746
0.1997	0.0816	0.1686	1.1894

III.2.2 Model selection using grid-search

We used the LIBSVM library for Support Vector Machines. Training the SVM will result in a model which is used for classifying the test data.

```
model = svmtrain(trainX, trainY, ...
        'autoscale',false, ...
        'boxconstraint',ones(size(trainX,1),1) * Cs(i), ...
        'kernel_function',kernel);
```

For the cross validation we used the same algorithm from the first assignment and we split the data into 5 parts. Each part was used to create a model and used the remaining data as a test set.
For each group we calculated the average error and we applied this algorithm for all the possible combination between the regularization parameters C and the kernel parameter γ.
For the non normalized data we obtained the best results for

$$C = 0.01$$

$$\gamma = 10^{-4}$$

For the normalized data we obtained the best results for

$$C = 1000$$

$$\gamma = 10^{-3}$$

Training data accuracy	0.7857
Test data accuracy	0.7526
Normalized training data accuracy	0.8980
Normalized test data accuracy	0.8144

After the normalization of the data set we have higher accuracy because each parameter has almost the same weight in creating the model.

III.2.3.1 Support vectors

We calculated the number of bounded and free support vectors for different values of C. As C increases the number of support vectors decrease and the number of bound support vectors tends to zero. If C is decreased, the number of support vectors increase and the number of free support vectors tends to zero.

As C is related to the slack variable, samples that violate the margin requirement is more likely to be accepted/neglected when the slack variable (or in this case C) increases, and thus get an α_i of zero. Thus the number of support vectors decrease with increasing C.

C	0.01	0.1	1	10	100	1000	10000
Bounded vectors	50	50	44	23	11	1	0
Free vectors	4	4	7	17	21	26	26

III.2.3.2 Scaling behavior

It is expected that the number of support vectors scales linearly with the number of training samples - the proportionality depending on the separation of the data. Training of the SVM has a polynomial complexity in the number of support vectors. Doubling the number of training samples should thus, at worst, quadruple the training time.