Statistics of Big Data, Spring 2018

Homework 3

Due date: 23 May 2018

1. **Basic Neural Nets**

   (a) Assume we are given a modeling problem with $x \in \mathbb{R}^p$ and $y \in \{0, 1\}$, which are can treat as a regression or classification problem (but prediction is always by comparing predictions to 0.5 and predicting either 0 or 1). For the following popular models, describe a neural network that implements them:

   - Standard linear regression
   - Logistic regression

   Explain in what sense the network implements them. Specifically, do we expect to get the same model from the network as from the regular model when applied to data? Why yes or why not?

   (b) The code HW3-1.r reads the South African heart dataset, divides it into training and test sets, and uses Keras to apply a NN with one hidden neuron and logit (=sigmoid) activation. It also applies and tests logistic regression. Use this skeleton to:

   i. Implement all four models described in the previous part
   ii. Prepare 2*2 confusion tables of predicted vs. actual labels
   iii. Briefly discuss the results compared to your expectations from the previous section

   (c) Implement a more complex architectures (e.g., a hidden layer with three nodes, and then an output layer, see commented code in the file) and apply it to the data. You may play with some of the parameters if necessary. Discuss its test-set performance.

   (d) Implement a network with a hidden layer with three nodes and an output layer, where all activations are linear. What form does the final model have? What functions of the original x variables are being fitted?

Resources for this problem:
Keras help
Keras in R

2. **Word2Vec Deep Learning**
In this problem, we apply the Word2Vec architecture we saw in a class to a somewhat simplified version of the problem. We have a vocabulary of V words, each with a true representation as a d dimensional vector w_v. We generate a set of three-word "sentences" (A, B, C), where the middle word B is generated uniformly with probability $1/V$ and the other two words depend on it as:

$$P(A = a|B = b) = \frac{\exp(w_b^t w_a)}{\sum_{v=1}^{V} \exp(w_b^t w_v)} \ , \ P(C = c|B = b) = \frac{\exp(w_b^t w_c)}{\sum_{v=1}^{V} \exp(w_b^t w_v)}.$$

(A, C are conditionally independent given B).
We want to build a "Deep Learning" system that will find the w vectors from data using gradient descent.

(a) Given N sentences, write down the log-likelihood ℓ function for the vectors $w_1, \ldots, w_V$, and find an explicit expression for the gradient of ℓ (that is, its derivative relative to w_{vj} , $\forall v = 1, \ldots, v \ j = 1, \ldots, d$).

(b) The code HW3-2.r simulates the probabilistic setting describes here and implements gradient descent. It is missing the calculation of the log likelihood and the derivative.

i. Implement these two calculations into the code

ii. Run it at least five times (you can change the settings) and report your results relative to the truth – are they close? Keep in mind that the problem has some symmetries that have to be taken into account when judging "closeness".

iii. If not close, try to find out if changing the parameters like learning rate and number of epochs, or starting from a different random initialization improves the results (or you may have a bug...).