Exam neural networks

Wojtek Kowalczyk
wojtek@liacs.nl
27.05.2016

It is a "closed book" exam: you are not allowed to use any notes, books, etc. For each problem you get some points (in total up to 90 points). Additionally, you get 10 points for free. The final grade for this exam is the total number of points you get divided by 10.

1) Bayes theorem (10 points: 3+7)

a) Formulate the Bayes theorem.

b) Let us suppose that a certain diagnostic classifier (perhaps a neural network) almost perfectly recognizes cancer: if you have cancer then with the probability 99% the classifier says "Cancer", if you don't have cancer, the classifier says "NoCancer", also with the probability 99%. Fortunately, cancer is not so common: it happens once per 10.000 persons. Now suppose that you have been diagnosed by the classifier as having cancer. What is the probability that you really have cancer? Write down a formula that expresses this probability.

2) Modeling probability distributions (15 points: 5+5+5)
At the beginning of the course we discussed the importance of learning probability distributions from data. The following questions are related to this part of the course.

Let us consider the problem of classifying images of 3 types of objects: *A*, *B*, *C*, assuming that each image is represented by a vector of 10 real-valued features $x = (x_1,... ,x_{10})$. Additionally, let us assume that you have data on 1.000 images of each type (i.e., 3.000 images in total).

a) How could you model $p(x|A)$ with help of a histogram? What is the main limitation of this method?

b) How could you model $p(x|A)$ with help of a Gaussian distribution? Which (and how many) parameters would you have to estimate? How would you estimate them? What is the main limitation of this method?

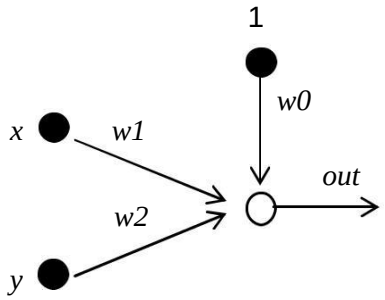
c) Suppose that you somehow managed to model the three density functions: $p(x|A)$, $p(x|B)$, $p(x|C)$. How would you proceed to classify an image represented by vector x? What additional information would you need?

3) Improvements of backpropagation (10 points: 5+5)
Describe in detail 2 techniques for speeding up the backpropagation algorithms:

a) adaptive gradient descent,

b) line search.

4) A simple network (2 + 10 + 3 points)
Let us consider a single unit network with two inputs, one output, and the cubic activation function $f(a)=a^3$:



a) Express the output of the network, *out*, as a function of five variables: x, y, w_0, w_1, w_2 (write a formula).

b) Derive the weight update rules for our network, assuming that we want to minimize the sum squared error, i.e., the error made by the network on input (x, y) and the target value t is given by $(out(x, y, w_0, w_1, w_2)-t)^2$. In other words, find partial derivatives of $(out(x, y, w_0, w_1, w_2)-t)^2$ over w_0, w_1, w_2 and formulate the update rules.

c) Can this network be trained to solve the XOR-problem? Justify your answer.
Here we assume that the logical values *true* and *false* are represented by +1 and -1, respectively, and the output of the network is interpreted as *true* if *out*>0 and *false* otherwise.

5) Convolutional neural networks (15 points: 5x2 + 3 + 2)

a) The first convolutional layer of Net5 consists of 6 feature maps, generated by 6 receptive fields of size 5x5, where each map receives inputs from a 32x32 input layer. Answer the following questions:

I what is the size of each feature map?

II how many connections are between the input layer and this convolutional layer (include one bias node for each receptive field)?

III what is the number of trainable parameters?

IV how many trainable parameters would we have if the input layer was fully connected to all the nodes of the convolutional layer?

V what is the role of subsampling (or pooling) layers?

b) Explain the concept of "dropout". What is its main role?

c) What is data augmentation? What is the purpose of using it and how does it affect the training time and network accuracy?

6) Restricted Boltzmann machines and autoencoders (25 points: 5x5)

Let us consider an RBM with m visible nodes $x_1, ..., x_m$, n hidden nodes $h_1, ..., h_n$, and an mxn weight matrix $W=(w_{ij})$, where w_{ij} denotes the weight of the connection between x_i and h_j. For simplicity, let us ignore the biases of the visible and the hidden layer (assuming that they are 0). Additionally, let us assume that both the visible and the hidden nodes take values 0 or 1.

a) As we know, an RBM defines a joint probability distribution over all possible $2^m 2^n$ binary vectors x, h, $P(x, h)$. Define $P(x, h)$ as a function of all weights (write down a formula). Next, write down formulas for: $P(h_j|x)$ and $P(x_i|h)$.

b) What is the purpose of training the RBM on a training set X? Which function has to be optimized (give a formula)? Describe the contrastive divergence algorithm for training an RBM. How many multiplications are needed (by the contrastive divergence algorithm) to calculate a single update of weights for a single input vector x? (Assume m input nodes, n hidden nodes, no biases.)

c) Describe the concept of a deep belief network. Explain in detail, how such a network can be trained to recognize hand-written digits. For simplicity, let us assume that the network consists of an input layer, two RBM layers and one softmax layer.

d) Describe how can you build a multi-layer RBM that is trained (in an unsupervised way) on images of hand-written digits, and then used to generate even more images of digits.

e) Describe the concept of a stacked denoising autoencoder.