Practical Lecture 4 - Multi-layer Perceptron and the Backpropagation algorithm

Machine Learning Course - 2nd Semester 2019/2020

Instituto Superior Tecnico, Universidade de Lisboa

1) Consider a network with three layers: 5 inputs, 3 hidden units and 2 outputs where all units use a sigmoid activation function.

a) Initialize all connection weights to 0.1 and all biases to 0. Using the squared error loss do a **stochastic gradient descent** update (with learning rate $\eta = 1$) for the training example

$$\left\{ \mathbf{x} = \begin{pmatrix} 1 \ 1 \ 0 \ 0 \ 0 \end{pmatrix}^T, \mathbf{t} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right\}$$

b) Compute the MLP class for the query point $\mathbf{x} = \begin{pmatrix} 1 \ 0 \ 0 \ 0 \ 1 \end{pmatrix}^T$.

2) Consider a network with four layers with the following numbers of units: $4, 4, 3, 3$. Assume all units use the hyperbolic tangent activation function.

a) Initialize all connection weights and biases to 0.1. Using the squared error loss do a **stochastic gradient descent** update (with learning rate $\eta = 0.1$) for the training example:

$$\left\{ \mathbf{x} = \begin{pmatrix} 1 \ 0 \ 1 \ 0 \end{pmatrix}^T, \mathbf{t} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\}$$

b) Reusing the computations from the previous exercise do a **gradient descent** update (with learning rate $\eta = 0.1$) for the batch with the training example from the a) and the following:

$$\left\{ \mathbf{x} = \begin{pmatrix} 0 \ 0 \ 10 \ 0 \end{pmatrix}^T, \mathbf{t} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

c) Compute the MLP class for the query point

$$\left\{ \mathbf{x} = \begin{pmatrix} 1 \ 1 \ 1 \ 0 \end{pmatrix}^T, \mathbf{t} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

using both models from a) and b). Which has smallest squared error? Which model has better classification accuracy?

3) Let us repeat the exact same exercise as in 2) but this time we will change:

- The output units have a softmax activation function
- The error function is cross-entropy

a) Initialize all connection weights and biases to 0.1. Using the squared error loss do a **stochastic gradient descent** update (with learning rate $\eta = 0.1$) for the training example:

$$\left\{ \mathbf{x} = \begin{pmatrix} 1 \ 0 \ 1 \ 0 \end{pmatrix}^T, \mathbf{t} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \right\}$$

b) Reusing the computations from the previous exercise do a **gradient descent** update (with learning rate $\eta = 0.1$) for the batch with the training example from the a) and the following:

$$\left\{ \mathbf{x} = \begin{pmatrix} 0 \ 0 \ 10 \ 0 \end{pmatrix}^T, \mathbf{t} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

c) Compute the MLP class for the query point

$$\left\{ \mathbf{x} = \begin{pmatrix} 1 \ 1 \ 1 \ 0 \end{pmatrix}^T, \mathbf{t} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

using both models from a) and b). Which has smallest cross-entropy loss? Which model has better classification accuracy?

3 Thinking Questions

a) What are the main differences between using squared error and cross-entropy?

b) Try to repeat the problems in the lecture with invented (differentiable) activation functions.

c) How do you think the MLP decision boundary will look like in two-dimensional problems?

d) Think about how the number of parameters to estimate grows with the number of layers. Do we need more or less data to estimate them properly?

e) Can we use MLP for a regression problem?

f) Try to apply an MLP to exercises from previous lectures. Notice that it can solve non-linearly separable problems.