problems1

November 14, 2018

1 Problem sheet 1

1.1 1

Visit http://playground.tensorflow.org and familiarize yourself with neural networks!

1.2 2

Implement an MLP for the iris dataset as shown in the lecture. 1. Implement the MLP explicitly via matrices, biases and non-linearities. 2. Implement the MLP via `keras` and `tensorflow`.

https://www.tensorflow.org/get_started/premade_estimators

1.3 3

Implement an MLP consisting of one hidden layers with ReLU nonlinearities for the Boston housing dataset. 1. Implement the MLP explicitly via matrices, biases and non-linearities. 2. Implement the MLP via `keras` and `tensorflow`.

1.4 4

Estimate the computational complexity of an MLP. 1. How does it grow in terms of the number of layers, the number of neurons and the activation functions. 2. What about the amount of memory required to store the model? 3. Suppose you want to deploy the MLP in an IoT device. What are possibilities to reduce computational and memory complexities?

1.5 5

Show that in the standard linear regression model, maximum likelihood estimation is equivalent to minimizing the mean-squared error.

More precisely, let $\{(x_i, y_i)\}_{i \le n} \subset \mathbb{R}^p \times \mathbb{R}$ be training data coming from the linear model $y_i = wx_i + \varepsilon_i$, where the $\{\varepsilon_i\}_{i \le n}$ are iid standard normal and $w = (w_1, \ldots, w_p)$ is a weight vector. Show that the loss function $\sum_i (y_i - wx_i)^2$ attains its minimum at the maximum-likelihood estimate for w.

1.6 6

Show that in the logistic regression model, maximum likelihood estimation is equivalent to minimizing the binary crossentropy.

More precisely, let $\{(x_i, y_i)\}_{i \le n} \subset \mathbb{R}^p \times \{0, 1\}$ be iid training data, where the conditional probability of observing $y_i = 1$ given x_i equals $f(x_i)$. The function $f = f_w$ depends on parameters w. Show that the loss function $-\Big(\sum_i y_i \log(f_w(x_i)) + (1 - y_i)\log(1 - f_w(x_i))\Big)$ attains its minimum at the maximum-likelihood estimate for w.

1.7 7

Consider the following model in `keras`.

```
In [1]: library(keras)
        model <- keras_model_sequential()
        model %>%
          layer_dense(3, input_shape = 2) %>%
          layer_dense(4) %>%
          layer_dense(1)
```

How many parameters does the model have in which layers, and why?

1.8 8

Train the models from problems 2 and 3 in keras and inspect their weights. Repeat the step with data simulated from a known model.