Neural Computation : Exercise Sheet 3

John A. Bullinaria - 2015

The following questions are of the kind that may come up in the exam this year. They are designed to help you monitor your progress – try to answer the questions without your notes, and then use your notes to check whether your answers are correct. The percentages indicate the corresponding fraction of a 1.5 hour exam.

Question 1

(a) Explain how a *gradient descent algorithm* can be used to train a *Multi-Layer Perceptron* neural network. **[8%]**

(b) Outline the principal techniques that can be applied to speed up the training when using such algorithms. Which of them is usually easiest, and why? **[7%]**

(c) Outline the principal techniques for improving the generalization ability of neural networks trained in that way. Which of them is usually easiest, and why? **[10%]**

Question 2

(a) In the context of feed-forward neural networks, explain what the following equation, and each symbol in it, represents:

$$E = -\sum_p \left[targ^p . \log\left(out^p\right) + \left(1 - targ^p\right) . \log\left(1 - out^p\right) \right] \qquad \textbf{[5\%]}$$

(b) Describe the basic ideas underlying *gradient descent learning algorithms,* and how and when the above equation would be used for such training. **[8%]**

(c) Explain how, in general, one could estimate the expected generalization ability of a trained neural network. **[6%]**

(d) Describe one way in which the above equation could be modified to improve the generalization ability of the trained neural network. **[6%]**

Question 3

(a) What is *k-fold cross validation*? Explain the assumptions involved and how it is relevant to the optimization of neural network generalization performance. **[8%]**

(b) Describe in detail how and why *early stopping* can be used to improve generalization with back-propagation training, and how the stopping point can be determined. **[12%]**

(c) What are the major advantages of early stopping over alternative procedures for improving generalization? **[5%]**

Question 4

(a) Explain in words what the various terms in the following equation signify, and what the equation as a whole tells us:

$$\mathcal{E}_D\left[\left(\mathcal{E}[y \mid x_i] - net(x_i, W, D)\right)^2\right]$$
$$= \left(\mathcal{E}_D\left[net(x_i, W, D)\right] - \mathcal{E}[y \mid x_i]\right)^2 + \mathcal{E}_D\left[\left(net(x_i, W, D) - \mathcal{E}_D\left[net(x_i, W, D)\right]\right)^2\right]$$
$$= \quad (\text{bias})^2 \quad + \quad (\text{variance})$$

[10%]

(b) Using a simple example, such as a Multi-Layer Perceptron trained to perform function approximation (regression), describe the extreme cases of *over-fitting* and *under-fitting* within the framework underlying the above equation. **[5%]**

(c) With reference to the above equation, explain three distinct approaches that might lead to improved generalization ability for a trained Multi-Layer Perceptron. **[10%]**

Question 5

(a) In the context of Multi-Layer Perceptron (MLP) training, explain carefully the relation between *Weight Decay* and cost function *Regularization*. **[5%]**

(b) Give an intuitive argument of why one might expect weight decay to help improve the generalization ability of an MLP. **[5%]**

(c) Explain why one should not also decay the thresholds/biases. **[4%]**

(d) Describe how one could use a *validation set* to choose the appropriate level of weight decay for a given classification problem. **[5%]**

(e) Briefly outline two approaches other than weight decay that one could expect to improve the generalization ability of an MLP. **[6%]**

Question 6

(a) Why might one want to add *noise/jitter* to the training data when training a neural network? Should the noise be added to the inputs, the target outputs, or both? **[8%]**

(b) What is meant by the terms *Hard Weight Sharing* and *Soft Weight Sharing*? Suggest how and when such processes might be usefully implemented in a neural network for practical applications. **[8%]**

(c) Explain what is meant by the terms *Constructive Algorithm* and *Pruning Algorithm*, and outline when and why these algorithms might be useful. **[9%]**