# Machine Learning (lecture) Fall 2014: Exercise sheet 6

*Return your results on paper in the Friday class on November 14. Please typeset and print or write legibly (we will downgrade for illegible handwriting).*

*You may team up in groups of two.*

*The total point number achievable is 110. If you reach more than 100 points, the surplus counts in the final grade calculation.*

**Problem 1 (25 points)** Here is a quote from the neural networks FAQ at http://www.faqs.org/faqs/ai-faq/neural-nets/part3/ :

```
Neural Network practitioners often use nets with many times as many
parameters as training cases. E.g., Nelson and Illingworth (1991, p.
165) discuss training a network with 16,219 parameters with only 50
training cases! The method used is called "early stopping" or
"stopped training" and proceeds as follows:

1. Divide the available data into training and validation sets.
2. Use a large number of hidden units.
3. Use very small random initial values.
4. Use a slow learning rate.
5. Compute the validation error rate periodically during training.
6. Stop training when the validation error rate "starts to go up".
```

Early stopping is one "quick and dirty" but simple and fast approach to tackle the overfitting problem. It uses only a single, possibly oversized network, whose capacity is not fully exploited. A more cautionary but more expensive method to prevent overfitting would be to train networks of various sizes, check their generalization abilities with some cross-validation scheme, and choose the network size (and network) that gives best generalization (or employ a regularization scheme).

Early stopping is often suspected to give poorer results (larger risk) than a careful selection of a properly sized network (or regularized), which is then trained to minimal training error. Explain why this might happen. Illustrate your explanation by drawing hypothetical network output graphs into the diagram below, one graph coming from a carefully sized, fully trained network and one graph coming from an early-stopped large network. Assume that both nets have been trained on the training samples indicated in the figure, and that the validation error was the same for both (i.e. validation error at stopping time for the first approach and validation error after proper model selection or regularization for the second).

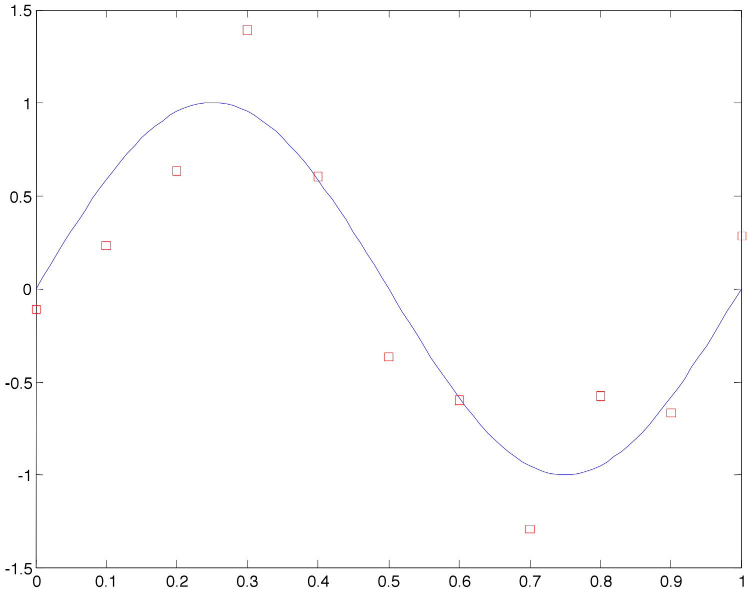

Figure: a nonlinear true function (line) and a noisy training sample (squares).

**Problem 2 (30 pts)**. Let $q_1, ..., q_n$ be the states of a stationary Markov chain $M$. Let $P^M(s_1 ... s_k)$ denote the probability that in this chain the sequence $s_1 ... s_k$ occurs in an observation window of length $k$. Define the *reverse* process $M_{rev}$ by

$$(*) \qquad P^{Mrev}(s_1\, s_2 ...\, s_k) = P^M(s_k\, s_{k\text{-}1} ...\, s_1).$$

Show that the reverse process is also a Markov process.

**Problem 3 (25 pts).** I once had a strange experience that apparently contradicts common machine learning wisdom. On a 1000-step long symbol sequence which was generated by a very complex stochastic sequence generation mechanism, I trained (i) a 5-state HMM and (ii) a 3-dimensional OOM ("observable operator model", a model type that is related to HMMs). The correct log probability of the training sequence (known only to me) was – 313.5; the log probability assigned to the training data by the trained models were – 314.2 for the 5-state HMM and – 324.0 for the OMM. This led me to believe that the 5-state HMM would be the better model, because, so I reasoned, it had a better training error while not overfitting. However, on independent test data the OOM turned out to be superior by a large margin. Can you offer an explanation? (Target size of plain English answer: 150 words.)

**Problem 4. (30 points)** An old gambler with bad eyesight suspects that the die he has recently bought is loaded. Unfortunately, when he sees the face of the die that has come up he knows that he might misread it (due to his weak eyes); even more unfortunately, he does not know the probabilities by which he misreads a certain true outcome for another, perceived outcome, that is, he has no clue whatsoever about the probabilities $q_{ij} = P(\text{perceived face is } j \mid \text{true face is } i)$. All he can do is throw the die many times and record what he *thinks* he sees. So he throws the die 10,000 times and gets a sequence of observations $D = y(1), ..., y(10\,000)$...

**(a)** (15 points) Describe the structure of a HMM for this process, by specifying a suitable set of hidden states and observable events. Assuming that the die displays the 6 faces with probabilities $p_1$, ..., $p_6$, what are the transition probabilities $p_{ii'}$ of your Markov transition matrix? What are the emission probabilties $e_{ij}$?

**(b)** (15 points) The gambler dimly remembers from his young days as an IUB student that one can train a HMM from an observation sequence. Could he indeed infer from a learnt HMM whether the die is loaded? If yes, how? if not, why not? Or could it work sometimes, but not in all cases?