EPART summer semester 2005/2006

Problems to think through before the final exam

It is advised to use normalized data while training neural networks (mean=0, variance=1 in the training set). For what conditions starting weight values should be taken randomly from the range $(-1/\sqrt{d}, 1/\sqrt{d})$, where d is the number of neuron's inputs. What is explanation of these recommendations?

What will happen if we set the same weights to two neurons in the same layer? What will happen if weights will be 0 (zero)?

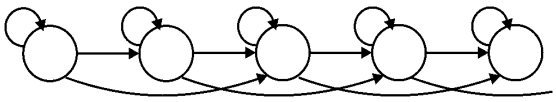
Explain the method of computing impurity change in the nodes of decision trees:

$$\Delta i(N) = i(N) - P_L i(N_L) - P_R i(N_R)$$

What is the maximum value of $\Delta i(N)$ in binary tree for two category case?
What is the maximum number of nodes of the binary tree for a training set of size *N*?

Compare techniques used in neural networks and decision trees preventing overfitting of the classifier.

Quite often Markov processes are used, in which transitions are possible only between two or three neighboring states. On the diagram you can see example of the process, in which only two next states are reachable from the current state (process can remain also in the current state).



Propose modifications of the *forward* procedure, which exploits transition restrictions. What is computational complexity of the new procedure? What modifications can be applied to the *Viterbi* algorithm? What about memory usage?

In which circumstances can we suppose in advance, that adding new classifier will degrade metaclassifier efficiency? (Or, in other words, what should we know about classifier, to know that it will be spoiling the results of the other classifiers).

We have two classifiers producing measurement level output, i.e. in the rank list we have also – beside class label – probability that this classification decision is correct. Propose a method of merging results of the both classifiers and describe the implementation steps.

Propose a method of approximate searching in the text for a pattern, allowing only one error (deletion, insertion and substitution of a character). There should be no text preprocessing.
As above, but only allowing for substitutions.
What is the complexity of these algorithms?

I plan to use trigrams to correct OCR results. Trigrams are without probability (i.e. only contain information that given 3 character sequence is possible or not). Design data structures and algorithm for correcting classification results. What results level of a classifier will match best your solution?

In what circumstances neighborhood generation is sensible for approximate text search? (Calculations would be welcomed.)

Design effective data structure to represent suffix tree built for the text using 4-character alphabet (for example *a c g t*).

Please, remember that tasks during the exam will cover the whole semester material – you should think again about the previous problems!