Machine Learning
written examination

Monday, May 17, 2010
Polacksbacken, 9\(^{th}\) - 14\(^{th}\)

Allowed help material: Pen, paper and rubber, dictionary

Please, answer (in Swedish or English) the following questions to the best of your ability!

Any assumptions made, which are not already part of the problem formulation, must be stated clearly in your answer!

Write your name on top of each page.

The maximum number of points is 40. To get the grade 3 (pass) a total of 20 points is required. The grade 4 requires 27 points and grade 5 requires 32 points.

Your teacher will drop in around between 10.00 and 10.30 to answer questions.

In this exam, some concepts may be called by different names than the ones used in the book. Here is a list of useful synonyms and acronyms:

- Perceptron = summation unit = SU = conventional neuron
- Binary perceptron = perceptron with binary activation function
- Multilayer perceptron = MLP = Feedforward network of summation units
- Back propagation = Generalized delta rule

Good luck!

47 students wrote the exam. 13 failed, 12 passed with grade 3, 10 passed with grade 4 and 12 passed with grade 5. The top score was 37, the average was 23.3.

I knew that unusually many students would take this exam so I tried to make it easy to mark and grade, encouraging students to write short answers. A side effect was that some students could do the exam in a very short time and/or space. One student passed the exam with grade 3 in 1 hour 10 minutes. Another only needed 3 pages of text and 1 hour and 40 minutes to receive grade 5. On the other hand, 13 students failed the exam which is a record both in absolute and relative numbers.

Counting, for each question, the number of student who received the maximum score, question 6 was the most difficult – only 3 students got full credit. Comparing the average score with the maximum, the most difficult question was question 9 – the only question where the average score was less than half of the maximum.
1. The binary perceptron
   a. What is the boolean logic function (write down the expression) implemented by the binary perceptron in the margin to the left? \( a, b, \) and \( c \) are binary inputs, either 0 (false) or 1 (true). ................... (3)

   The boolean function is \( a+bc \), i.e. \( a \) OR (b AND c)

   b. Binary perceptrons can be trained by the Perceptron Convergence Procedure. How are the weights updated in this algorithm if the perceptron output is 0 and the desired output is 1? ....................... (2)

   The corresponding input value is added to the weight, to increase the weighted sum.

   Many students confused PCP for the delta rule, but the delta rule is not directly applicable to a binary perceptron, since the activation function is not differentiable.

   In the special case where we assume that the node is linear (which it isn't) and \( \eta=1 \) the effect is the same, but it was PCP which was asked for here.

   This question was intended to be a general question, not specifically about the network from a), as some students apparently assumed.

   46 answers, 11 max scores (5), average 3.5

2. Multilayer perceptrons and back propagation

   Under the most common assumptions, the back propagation algorithm updates the weights of a multilayer perceptron as follows:

   \[
   \Delta w_{ji} = \eta \delta_j x_i \\
   \delta_j = \lambda y_j (1 - y_j) (\delta_j - y_j), \text{ if } <\text{CONDITION 1}> \\
   \delta_j = \lambda y_j (1 - y_j) \sum_k w_{kj} \delta_k, \text{ if } <\text{CONDITION 2}>
   \]

   a. What is CONDITION 1 and CONDITION 2 here? .................. (1)

   CONDITION 1 is that node \( j \) is an output node, CONDITION 2 is that node \( j \) is a hidden node.

   b. Explain the sum (over \( k \)) in the last row! What is \( k \) an index of? . (2)

   The sum is over all nodes in the adjacent node layer which is closer to the outputs (or are the outputs). This is the node layer for which we computed the \( \delta \)-values, in the previous step in the back propagation procedure.

   Students who claimed that \( k \) is a sum over the output nodes got full credit only if they explicitly assumed (in a drawing or in words) that the network had only one hidden layer.

   c. These equations assume that the neurons have logistic activation functions. Which part or parts would change if we used another activation function?.................................................. (2)

   \( \lambda y_j (1-y_j) \) would change in both rows where it occurs. This is the derivative of the logistic function. \( \lambda \) is the slope parameter of the logistic function. Answers which did not include \( \lambda \) received full credit anyway, since \( \lambda \) is very often equal to 1.
d. Another assumption made above is that the objective function to be minimized is the squared error. Which part or parts would change if we replaced this by another objective function? .............................. (2)

\[(d_j - y_j)\] in the \(\delta\)-update for the output layer nodes, would change. The update for the hidden layer is not affected (other than indirectly through the sum over \(k\)).

To be precise the output layer \(\delta\)-update also changes its sign if we do this, since the derivative of the objective function is actually \(-(d_j - y_j)\), but full credit was given also for answers which did not mention this.

e. During training it is common to plot the error as it changes over time in a graph. What is the typical effect seen in this graph, if the learning rate (gain, \(\eta\)) is set too high? ................................. (1)

The effect is a very noisy error graph, noisy here meaning the opposite of smooth, where the error fluctuates (oscillates) a lot.

3. Explain K-fold cross validation in supervised learning! How is it done and why? ................................................................. (3)

K-fold cross validation: Divide the data you have in \(K\) equal sized parts. Then, for each part \(i\) (\(1 \leq i \leq \) \(K\)), train an all the data except part \(i\) and test on part \(i\). Then compute average error over these \(K\) experiments. This is done when we have too little data to afford separate training and test sets. By using K-fold cross validation, we can use all the data for training and still get a generalization measure.

An extreme (but common) case of this is the so called leave-one-out principle, where \(K\) = the size of the training set, i.e. each part contains only one pattern, but I don't think I mentioned this variant on lecture.

Points were deducted for failing to tell that the result is the average over the \(K\) tests, and for failing to answer the why-question. Some students had confused K-fold cross validation with early stopping, which is another form of cross validation. Both try to avoid overfitting - early-stopping by deciding when to stop training, K-fold cross validation by trying to maximize the training set size.

Cross validation techniques are not specifcally for neural networks. They are applicable to any supervised learning system trained on a finite set of data.

39 answers, 17 max scores (3), average 2.0
4. Self Organizing Feature Maps can be described as an extension to Standard Competitive Learning to implement topologically preserving dimension reduction.

   a. Explain what it means to be topologically preserving!............... (2)

   To reduce dimensionality in a way which maintains the statistical distribution of the data, so that, for example, two points that are close to each other in the high-dimensional input space are still close to each other also in the lower-dimensional output space (and vice versa).

   b. How is the Standard Competitive Learning Rule extended to do implement this?........................................................... (2)

   Instead of just moving the winner (the weight vector which is closest to the input vector), we move all weight vectors towards the input, but to a degree which depends on how far that node is from the winner in the network structure (usually a 2-dimensional grid).

   This is done by defining a neighbourhood function f(j,k) where j is the index of the node to move and k is the index of the winner node. This function should have a max=1 for the winner itself (j=k) and then decrease with distance between j and k. A Gaussian function, for example. This neighbourhood function is used as a multiplier in the update formula:

   \[ \Delta w_j = \eta f(j,k) (x_i - w_j) \]

   Points were deducted for not describing the neighbourhood function (just calling it by name is not enough), for not mentioning that all nodes are moved (weighted by f), or for not telling how f is used in the update formula (as a multiplier). An explanation in words is fine but should at least state that the neighbourhood function is multiplied, not just “included”.

   Some students had confused the SOFM update rule with the update rule for GNG (Growing Neural Gas), which moves all neighbours by the same amount (but different from the winner).

   44 answers, 6 max scores (4), average 2.4

5. A Q-value in Q-Learning can be defined as follows: \( Q(s,a) \) is a discounted sum of future rewards, after taking action \( a \) in state \( s \), and assuming that all future actions are greedy.

   a. What does it mean to be greedy here? ............................................. (1)

   The greedy action is the action which is currently believed to be the best one (has the greatest “merit” or value). In other words, it is the opposite of an exploratory action.
b. The definition can be rewritten on a simple recursive form, as an equation in terms of $Q$ itself. How? Write down the equation and explain its parts. ................................................................. (2)

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a')$$  (*)

where $r$ is the immediate reward received after taking action $a$ in state $s$, $s'$ is the new state we came to and the max operator is over all possible actions in that state(s'). In other words, $a'$ is the greedy action in state $s'$ and $\max Q(s', a')$ is the $Q$-value of that action. $\gamma \in [0,1]$ is a discount factor which ensures that the sum of future rewards are finite, even in the infinite horizon case.

Many students wrote down the update equation for Q-Learning instead, which gave partial credit, but was not asked for here. It is however a direct consequence of (*), given by the answer to c) below.

c. How is this equation then used to define the temporal difference error? ............................................................................................. (2)

The temporal difference error here (for Q-Learning) is the right hand side minus the left hand side of the above equation, i.e. $Err_{TD} = r + \gamma \max_{a'} Q(s', a') - Q(s, a)$. It is in proportion to this error we update $Q(s,a)$ in the update equation, but that was not the question here.

46 answers, 5 max scores (5), average 2.5

6. Learning to play a board game, such as chess, is an example of an application where you could chose to use supervised learning or reinforcement learning. What are the main advantages/disadvantages of each in that case? ........................................................................................................... (3)

Supervised learning (SL) gets more specific feedback and is therefore, in general, more efficient and faster than reinforcement learning. On the other hand SL requires an expert (to give the correct answer) and that expert can be difficult to find, difficult to get clear answers from and/or expensive to hire. Furthermore the SL system can never become better than that expert since the goal is to imitate the expert, not to beat it. (from a SL perspective, beating the expert would actually be wrong!)

Reinforcement learning (RL) learns by trial and error, and therefore requires no expert and may very well become better than such an expert, if there is one. In many applications, for example board games, where the task is to compete against a similar opponent, RL is also more autonomous since the agent can learn from playing against itself over and over again, without the need of an operator/supervisor to control learning. On the other hand, learning by trial-and-error is to learn from a very weak/noisy feedback so training may take very long time. This is essentially RL’s only drawback, but it is a big one.
Many students correctly pointed out that the two forms of learning can be combined — training supervised first and then continue using reinforcement learning to improve from there. You still need the expert in the beginning though, and the RL session will be strongly biased. There is a great risk that the initial SL session ends up in a local minimum from which the subsequent RL session can not escape.

Most students received 2 credits for noting the speed difference and that RL has no upper limit (can become better than the teacher). 3 credits required some note of the drawbacks of requiring an expert at all in SL, or about the autonomy advantage of RL.

Some students claimed that SL requires a differentiable objective function and/or that SL must be trained off-line. These are shortcomings of some SL algorithms, not consequences of SL as such.

45 answers, 3 max scores (3), average 1.9

7. Evolutionary computation

a. Select a (non-trivial) crossover point and perform a one point crossover on the two genotypes (character strings) in the margin to the left! What is the result? ........................................................... (2)

Pick a selection point and swap the content of the tails, creating two new genotypes. For example:

```
abbccabca
bcaabbacc
```

Some students swapped the heads instead of the tails — that works too, of course, and gave full credit, though it is not how one point crossover is usually described.

Some students only swapped the letters at the crossover point, which is not one point crossover — it is in fact a special case of two point crossover.

b. One big challenge in evolutionary computation is to avoid premature convergence. This is less likely to occur if we use rank selection instead of fitness selection. Why? .................................................. (2)

Premature convergence is when the population converges to one spot, losing diversity (and therefore the point of having a population at all).

Consider an extreme case where one individual X in the population has a much greater fitness than any other. In fitness selection, X is very much more likely to be selected for recombination/ reproduction/mutation due to the great difference in fitness. Thus, its genotype, or parts of it, tends to spread rapidly in the population, in effect pulling all other individuals closer to it.

In rank selection, since X is ranked number one, it is still the most likely to be selected, as it should be, but the probability does not depend on how much greater fitness it has. So, less fit individuals have a slightly greater probability of being selected in rank selection, and the whole population is therefore more likely to maintain diversity.

Some students suggested that fitness selection is deterministic — that low fit individual have 0 probability of being selected or (similarly) that individuals with very high fitness are guaranteed to be selected (=elitism). This is not the case.

45 answers, 17 max scores (4), average 2.8
8. Swarm intelligence

a. Define stigmergy in one sentence, and then explain why it is of particular interest to computer science! ........................................ (2)

Stigmergy = Indirect communication and coordination by local modification and sensing of the environment.

The most common example (it was not a requirement here to give examples) is how pheromones are used by ants to control the behaviour of other ants, but many other land living animals, including humans, also use stigmergy. Dogs leave messages to other dogs by urinating on points of interest, and humans sometimes leave messages to family members on kitchen tables. Road signs is another human example.

It is of particular interest to computer science because it is an extremely scalable form of communication. An ant colony may consist of millions of individuals, so point-to-point communication would not be feasible. Furthermore, the individual agents can be made very simple but yet the system as a whole can solve fairly complicated coordination tasks.

b. In the Particle Swarm Optimization variant called lbest, the particle velocities are updated by a weighted sum of two parts – a trade-off between a cognitive component and a social component. What are these two components trying to achieve? (what are they striving for?) ............................................................................................... (2)

The cognitive component strives for the best position found so far by this particle, i.e. the previously visited position in space which had the best fitness.

The social component strives for the best position found so far by the neighbours of this particle in a neighbourhood graph. The neighbourhood is social, not topological – it is a network of friends, rather than a network of neighbours. In the extreme case (gbest) this neighbourhood graph covers the whole swarm, i.e. the social component strives for the best position found so far by any particle in the swarm. This is usually not a good idea though. Local interaction works better.

42 answers, 8 max scores (4), average 2.2

9. Particle Swarm Optimization and Genetic Algorithms

a. Particle Swarm Optimization and Genetic Algorithms are similar in that they can both be applied to the same type of problems. One of them is likely to require more computing time (and memory) than the other, though. Which, and why? ................................................................. (2)

Genetic algorithms would typically require more computing power/time since the populations are usually much larger (10 times larger or more) than in particle swarm optimization. This means that many more individuals have to be evaluated and this evaluation (the fitness function) is where both algorithms will have to spend most of their computing time.

PSO must reevaluate each particle every generation, though, which is not necessarily the case in GA. Some students claimed the opposite. PSO must do this, since all particles move all the time and each one has to check if it found a new personal best. In GA, if an individual is reproduced (copied as it is) to the next generation and if the optimization problem is stationary, there is no reason to reevaluate it. However, both this difference and that PSO particle must remember their personal bests, is overshadowed by the difference in population size.

Some students had confused PSO with ACO (Ant Colony Optimization) and/or GA with genetic programming.
b. Particle Swarm Optimization and Genetic Algorithms have in common that they do not depend on gradients of the objective function to work, in contrast to many neural network algorithms. This independence is usually considered an advantage, but the “No Free Lunch” theorem tells us that there must be cases where it is a disadvantage. Describe such a case! ............................................. (2)

For problems where the objective function is smooth and has few local optima and/or plateaus, gradient methods work very well and are hard to beat. An extreme example is the bowl function shown on the PSO lecture.

Throwing away the gradient information also means that PSO and GA are not very good for fine-tuning. They may find the tallest hill faster than the gradient methods, but may then have difficulties finding the exact position of the hill-top. (By the way, this applies to RPROP as well – it also throws away most of the gradient information, but still requires differentiability!)

Some students implied in their answers that any problem with only one optimum would do. You can not know that without knowing more about the surface. It is not necessarily solvable by a gradient method at all (differentiable), or may contain plateaus (saddle points) where we could get stuck (due to zero derivatives), etc.

A few students suggested that gradient descent might be better if we want to find the local optima (which they tend to get stuck in). This was creative and gave partial credit. However the whole idea is kind of contradictory in that the objective function is supposed to express what we want, which it does not in that case.

44 answers, 7 max scores (4), average 1.9