*Christoph F. Eick*

**Wednesday, April 8, 202019**

**“Short” Review for COSC 4368 Midtem2 Exam**

Remark: Only questions 1 and 3 are discussed on April 8.

**1) Classification and Supervised Learning in General**

a) What is overfitting? What are the characteristics of overfitting? What can be done in the context of decision trees to battle overfitting?

The model is too complex, the testing/generalization error is not optimal, although the training error is low.

b) What is the key contribution of the backpropagation algorithm? What problems does it solve?

It measures and associates an error with the nodes of the intermediate layers that then can be used to learn the weights of incoming connections of nodes of the intermediate layers.

c) What is the purpose of training, test and validation sets in Supervised Learning? What else can be said about their relationship?

Training set used for model learning

Validation set is used to learn the best (hyper-)parameters for the method that generates the model (e,g, C and kernel function parameters in the case of the SVM)

Test set is used to determine the accuracy of the learnt model; e.g. to determine accuracy, testing/generalization error

All three sets should be disjoint🡺otherwise cheating

**2) Supervised Learning in General [8]**

a) What is the purpose of using N-fold Cross Validation? Explain in a few sentences how 2-fold cross validation works! [4]

To determine the generalization error/training accuracy of the learn model (if they just say just accuracy give them on 0.5 points) [1.5]

Correct description of 2 fold cross validation [2.5]

b) Deep Neural Networks usually employ very complex models; what can be done to alleviate the problem of overfitting when using deep neural networks? [2]

Use very large training sets[2].

**3) Neural Networks**

Describe how multi-layer neural networks, consisting of 3+ layers learn a model for a training set! Limit you answer to at most 9 sentences! [7]

Neural network learning tries to find weights that minimize the error in the neural network prediction for a training set [1]. Neural networks employ gradient decent hill climbing to find the “best” weights. [1]. In particular, Neural network learning adjust weights using the gradient of the error function of the training set [1]; the search starts with a random initial weight vector and weights are adjusted in the direction of the steepest negative gradient of this error function---that is weights are updated accordingly moving in the direction that reduces the error the most [2]; The step width of this weight update depends on the learning rate and other factors [1]. In order to apply this procedure the error for each none-input node has to be known. As this error is not initially given intermediate for intermediate layer nodes, it is computed using the back-propagation algorithm [2].

Other observation might deserve credit. At most 7 points!

**4) Reinforcement Learning [13]**

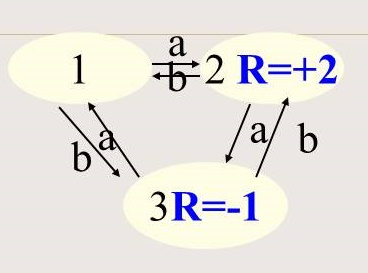
a) What are the main differences between supervised learning and reinforcement learning? [5]

SL: assumes a static world[0.5], correct answer/action is known and described in training sets from which models are learnt![1.5]

RL: can deal with dynamic changing worlds/can adapt [1]; needs to learn from indirect, sometimes delayed feedback/rewards[1]; suitable for exploration of unknown worlds[1]; temporal analysis/worried about the future/interested in an agent’s long term wellbeing[1], needs to carry out actions to find out if they are good—which actions/states are good is (usually) not know in advance1[1]

Other answers might deserve credit, might also use answer from the RL-Paper paragraph on that matter (page 239)! At most 5 points!

b) Assume the following world ABC is given:



Assume that SARSA is used for the ABC World; the initial q-values are all 0, the learning rate α is 0.5 and the discount rate γ is 1. The agent begins in state 2 and applies aaaa (a four times). Additionally, you can assume that according the agent’s policy operator a is always applied in state 3. How does the Q-table look like after aaaa has been applied? [4]

Q(a,s) 🡨 Q(a,s) +

α [ R(s) + γ\*Q(a’,s’) - Q(a,s) ]

Q(a,2)= 0 + 0.5\*(2 + 0 – 0)=1

Q(a,3)= 0 + 0.5\*(-1 + 0 – 0)=0.5

Q(a,1)=0+ 0.5\*(0+1-0)=0.5

Q(a,2)=1+ 0.5\*(20.5-1)=1+0.5\*0.5=1.25

One error at most 2 points; 2 errors at most 0.5 points!

|  |  |  |
| --- | --- | --- |
|  | Value | SARSA-  Update |
| q(a,1) | 0 |  |
| q(b,1) | 0 |  |
| q(a,2) | 0 |  |
| q(b,2) | 0 |  |
| q(a,3) | 0 |  |
| q(b,3) | 0 |  |

c) In which cases would you prefer SARSA over Q-Learning? [4]

When the employed policy is quite different from a greedy policy [4]; in world that change significantly and need adaptation [2.5], when dealing with unknown worlds that need to be explored.[1.5]