Artificial Intelligence

Chapters Reviews

Readings: Chapters 8-20 of Russell & Norvig.

Exercise Problems

Chap. 20: 15, 17.
Chap. 14: 2, 3, 4, 8.

Topics covered in the midterm

First-order logic (Chap. 8, 9)
- Syntax and Semantics (8.2, 8.3)
- Converting formulas into clauses (9.1)
- Unification (9.2)
- Logic Programming (9.4)
- Resolution and strategies (9.5)

Knowledge Representation (Chap 10)
- Situation calculus and frame problem (10.3)
- Closed World Assumption and Negation as Failure (10.7)

Planning (Chap 11)
- The STRIPS Language (11.1)
- Forward state-space search (11.2)
- Partial-order planning (11.3)

Uncertainty (Chap 13, 14)
- Probability Theory (13.1-13.6)
- Bayesian Networks (14.1-14.3)
- Fuzzy Logic (14.7)

Learning (Chap. 18, 20.5)
- Inductive learning and decision trees (18.1-18.3)
- Neural networks (20.5)
Exercise 20.15

Consider the following set of examples, each with six inputs and one target output:

| I_1  | 1 1 1 1 1 0 1 0 0 0 0 0 0 0 |
| I_2  | 0 0 0 1 0 1 0 1 0 1 0 1 1 1 |
| I_3  | 1 1 1 0 1 0 0 1 1 0 0 0 1 1 |
| I_4  | 0 1 0 0 1 0 0 1 1 0 0 1 1 0 |
| I_5  | 0 0 1 1 0 1 0 1 0 1 0 1 1 1 |
| I_6  | 0 0 0 1 0 1 0 1 0 1 0 1 1 1 |
| T    | 1 1 1 1 1 1 0 1 0 0 0 0 0 0 |

b. Run the perceptron learning rule on these data and show the final weights.

Assuming $\alpha = 1$, we have

$$\text{Sum} = W_0 (\text{I_0}) + W_1 (\text{I_1}) + W_2 (\text{I_2}) + \ldots + W_6 (\text{I_6})$$

$$\text{Out} = \text{step} (\text{Sum}), \text{Err} = T - \text{Out}, \text{and} \ W_j \leftarrow W_j + I_j \times \text{Err}.$$

b. Run the decision tree learning rule, and show the resulting decision tree.

Choosing $I_i$ first such that $\text{Remainder}(I_i)$ is minimal.

$$\text{Remainder}(I_i) = \frac{c_0}{p+n} I(\langle p_0/n_0, c_0 \rangle) + \frac{c_1}{p+n} I(\langle p_1/n_1, c_1 \rangle)$$

where $I_i$ divides the examples into $c_0$ 0-cases and $c_1$ 1-cases, $c_0 = p_0 + n_0$, $c_1 = p_1 + n_1$.

$$I(\langle p_0, p_1 \rangle) = -(p_0 \log_2 p_0 + p_1 \log_2 p_1)$$

Exercise 20.17

Suppose you had a neural network with linear activation functions.

a. Assume that the network has one hidden layer. For a given assignment to the weights $W$, write down equations for the value of the units in the output layer as a function of $W$ and the input layer $I$, without any mention to the output of the hidden layer. Show that there is a network with no hidden units that computes the same function.

b. Repeat the calculation in part (a), this time for a network with any number of hidden layers. What can you conclude about linear activation functions?