# Homework 7 Sample Solutions, 15-681 Machine Learning

**1.**

- Given the specified representation of hypotheses, the quantity to be minimized according to the Minimum Description Length principle is $a(\log_2 n) + e(1 + \log_2 m)$, where $a$ is the number of attributes specified by a hypotheses and $e$ is the number of examples it misclassifies.
- The key here is to make $m$ much smaller than $n$, so a single negative example with many attributes should do the trick. Given the following training set:

  `-: < Premature, Proactive, Preternatural, Paranoid >`

  In this case, any consistent hypothesis will have to specify at least one attribute to exclude the example. An example of such a hypothesis is `< Mature, ?, ?, ? >`, which specifies one attribute and misclassifies no examples, so its encoding length is $1(\log_2 4) + 0(1 + \log_2 1) = 2$. On the other hand, the maximally general, *every-instance-is-positive* hypothesis, while inconsistent, specifies no attributes and misclassifies only one example, so its encoding length is $0(log_2 4) + 1(1 + \log_2 1) = 1$, and would thus be preferred by the MDL principle.
- (Optional extra credit) The algorithm picks the hypothesis that minimizes $a(\log_2 n) + e(1 + \log_2 m)$, while the MAP hypothesis maximizes $P(h)P(D|h)$. To determine distributions for which these two are equivalent, just work backwards through the derivation in the MDL handout:

$$\begin{aligned} -\log_2 P(h) &= a\log_2 n \\ P(h) &= 2^{-a\log_2 n} \\ &= \tfrac{1}{2^{a\log_2 n}} \\ &= \tfrac{1}{n^a} \end{aligned}$$

$$\begin{aligned} -\log_2 P(D|h) &= e(1+\log_2 m) \\ P(D|h) &= 2^{-e(1+\log_2 m)} \\ &= (2^{-e})(2^{-e\log_2 m}) \\ &= \tfrac{1}{2^e}\tfrac{1}{2^{e\log_2 m}} \\ &= \tfrac{1}{2^e}\tfrac{1}{m^e} \end{aligned}$$

**2.**

In the interest of getting this on-line ASAP, let's skip the drawing and go with a verbal description.

The resulting network has three units, corresponding to the horn clauses for *pass_final*, *know_material*, and *wide_awake*, and four inputs, corresponding to *slept_last_night*, *studied*, *ate_breakfast*, *passed_midterm*.

The *know_material* unit has one connection of weight $W$ from the input *studied*, a bias-unit connection weight of $-W + 0.5W$, and three connections of neglible weight from the other input values.

The *wide_awake* unit has one connection of weight $W$ from the input *slept_last_night*, a bias-unit connection weight of $-W + 0.5W$, and three connections of neglible weight from the other input values.

The *pass_final* unit has two connections of weight $W$ from the *know_material* and *wide_awake* units and a bias-unit connection weight of $-2W + 0.5W$.

**3.**

As was pointed out in class, the circumstances under which crossover will be a useful operator in evolutionary search are not well agreed upon. Therefore, full credit was given to any clear, well-reasoned examples.

One argument that can be made is that a search space in which recombination of partial solutions is likely to yield an improved solution is one in which crossover would be more likely to be useful.

For example, if the search space is the space of "recipies" for making a fixed quantity of some product, e.g. 18 metric tons of egg-nog, and the representation employed is a vector of quantities of the individual ingredients, we might expect that cross-over might be helpful for combining one solution that has close to the right amount of nutmeg with another that has close to the right amount of milk.

On the other hand, imagine a search for a fixed plan in some domain to accomplish some goal, with solutions being represented as an ordered vector of "moves". If it turns out that in this domain there are several alternative, very different sequences that might lead to the goal, we might expect that recombining two moderately successful but very different sequences may very frequently yield two wholly ineffective sequences.

Note that, at the same time, arguments may be made *against* the utility of crossover in the former domain and *for* its utility in the latter.