10-601 Machine Learning, Fall 2012
Homework 3

Instructors: Tom Mitchell, Ziv Bar-Joseph

TA in charge: Mehdi Samadi
email: msamadi@cs.cmu.edu

Due: Monday October 15, 2012 by 4pm

Instructions There are 4 questions on this assignment – no programming. Please hand in a hard copy of your completed homework to Sharon Cavlovich (GHC 8215) by 4 PM on Monday, October 15th, 2012. Don't forget to include your name and email address on your homework.

1 Neural Networks

1.1 Expressiveness of Neural Networks [10 points]

As discussed in class, neural networks are built out of units with real-valued inputs $X_1 \ldots X_n$, where the unit output Y is given by

$$Y = \frac{1}{1 + \exp(-(w_0 + \sum_i w_i X_i))}$$

Here we will explore the expressiveness of neural nets, by examining their ability to represent boolean functions. Here the inputs X_i will be 0 or 1. Of course the output Y will be real-valued, ranging anywhere between 0 and 1. We will interpret Y as a boolean value by interpreting it to be a boolean 1 if $Y > 0.5$, and interpreting it to be 0 otherwise.

1. Give 3 weights for a single unit with two inputs X_1 and X_2, that implements the logical OR function $Y = X_1 \vee X_2$.

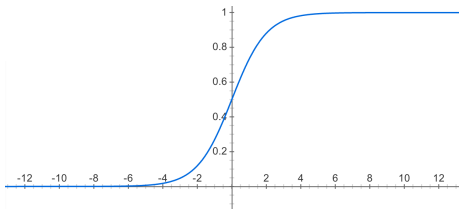

Figure 1: $\frac{1}{1+e^{-x}}$.

★ **SOLUTION:** Figure 1 shows the value of $y = \frac{1}{1+e^{-x}}$ for different values of x. Note that $y \geq 0.5$ if $x \geq 0$, and $y \leq 0.5$ if $x \leq 0$. Given this, we need to choose w_i so that $w_0 + w_1 * x_1 + w_2 * x_2$ will be greater than 0 when $x_1 \vee x_2$ is equal to 1. One candidate solution is $[w_0 = -0.5, w_1 = 1, w_2 = 1]$.

2. Can you implement the logical AND function $Y = X_1 \wedge X_2$ in a single unit? If so, give weights that achieve this. If not, explain the problem.

★ **SOLUTION:** Similar to previous part, we can obtain $[w_0 = -1.5, w_1 = 1, w_2 = 1]$

3. It is impossible to implement the EXCLUSIVE-OR function $Y = X_1 \oplus X_2$ in a single unit. However, you can do it using a multiple unit neural network. Please do. Use the smallest number of units you can. Draw your network, and show all weights of each unit.

★ **SOLUTION:** It can be represented by a neural network with two nodes in the hidden layer. Input weights for node 1 in the hidden layer would be $[w_0 = -0.5, w_1 = 1, w_2 = -1]$, input weights for node 2 in the hidden layer would be $[w_0 = -0.5, w_1 = -1, w_2 = 1]$, and input weights for the output node would be $[w_0 = -0.8, w_1 = 1, w_2 = 1]$.

4. Create a neural network with only one hidden layer (of any number of units) that implements $(A \vee \neg B) \oplus (\neg C \vee \neg D)$. Draw your network, and show all weights of each unit.

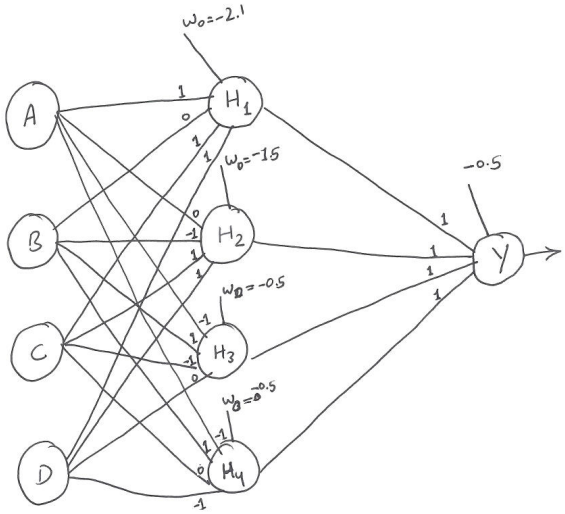

Figure 2: An example of neural network for problem 1.4

★ **SOLUTION:** Note that XOR operation can be written in terms of AND and OR operations: $p \oplus q = (p \wedge \neg q) \vee (\neg p \wedge q)$. Given this, we can rewrite the formula as $(A \wedge C \wedge D) \vee (\neg B \wedge C \wedge D) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg D)$. This formula can be represented by a neural network with one hidden layer and four nodes in the hidden layer (one unit for each parenthesis). An example is shown in Figure 2.

1.2 MCLE, MAP, Gradient descent [15 points]

In class we showed the derivation of the gradient descent rule to train a *single* logistic (sigmoid) unit to obtain a Maximum Conditional Likelihood Estimate for the unit weights $w_0 \ldots w_n$. (See the slides from the lecture on neural networks: http://www.cs.cmu.edu/~tom/10601_fall2012/slides/NNets-9_27_2012.pdf, especially the slides on pages 4 and 5).

1. The slide at the top of page 5 claims that if we want to place a Gaussian prior on the weights, to obtain a MAP estimate instead of a Maximum likelihood estimate, then we must choose weights that minimize the expression E:

$$E = \quad c\sum_i w_i^2 \; + \; \sum_l (y^l - \hat{f}(x^l))^2$$

where w_i is the i^{th} weight for our logistic unit, y^l is the target output for the l^{th} training example, x^l is the vector of inputs for the l^{th} training example, $\hat{f}(x^l)$ is the unit output for input x^l, and c is some constant.

Show that this claim is correct, by showing that minimizing E is equivalent to maximizing the expression: $\ln P(W) \prod_l P(Y^l|X^l; W)$. Here W is the weight vector $\langle w_0 \ldots w_n \rangle$. In particular, assume each weight w_i in the single unit follows a zero-mean Gaussian prior, of the form:

$$p(w_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{w_i - 0}{\sigma}\right)^2\right)$$

So that $P(W) = P(w_0, \ldots, w_n) = \prod_{i=0}^{n} P(w_i)$.

★ **SOLUTION:**

$$\begin{aligned}
\hat{W}_{MAP} &= \arg\max_W \ln\left(P(W)\prod_l P(Y^l|X^l;W)\right) \\
&= \arg\max_W \ln P(W) + \ln\prod_l P(Y^l|X^l;W) \\
&= \arg\max_W \ln P(W) + \ln\prod_l \left(\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2}\left(\frac{Y^l - f(X^l)}{\sigma}\right)^2\right)\right)
\end{aligned} \tag{1}$$

From the definition of $P(W)$ we can write:

$$\begin{aligned}
P(W) &= \prod_{i=0}^{n} \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2}\left(\frac{w_i - 0}{\sigma}\right)^2\right) \\
\ln P(W) &= \frac{n\ln(2\pi\sigma^2)}{2} + \left(\frac{-\sum_{i=0}^{n} w_i^2}{2\sigma^2}\right)
\end{aligned} \tag{2}$$

By removing constant values from the above two formulas (since they don't change the maximization) and removing the negative sign, we would have:

$$\begin{aligned}
\hat{W}_{MAP} &= \arg\min_W c\sum_{i=0}^{n} w_i^2 + \sum_{l=1}^{M}(Y^l - f(X^l))^2 \\
&= \arg\min_W E
\end{aligned} \tag{3}$$

2. Derive the gradient you would use to obtain the map estimate, for a single unit with two inputs X_1 and X_2. In other words, give formulas for each of the three partial derivatives

$$\left[\frac{\partial E}{\partial w_0}, \; \frac{\partial E}{\partial w_1}, \; \frac{\partial E}{\partial w_2}\right]$$

Hint: the slide at the bottom of page 5 of the handout does part of your task. If you get stuck, the slides on linear regression might also be helpful.

★ **SOLUTION:**

$$
\begin{aligned}
E &= c\sum_{i=0}^{n} w_i^2 + \sum_{l=1}^{M}(Y^l - f(X^l))^2 \\
&= cw_0^2 + cw_1^2 + cw_2^2 + \sum_{l}(Y^l - f(X^l))^2 \\
\frac{\partial E}{\partial w_0} &= 2cw_0 - \sum_{l} 2(Y^l - f(X^l))\frac{\partial}{\partial w_0}(f(X^l))
\end{aligned} \tag{4}
$$

In class we showed that $\frac{\partial}{\partial w_0}(f(X^l)) = f(X^l).(1 - f(X^l))$. So we can write:

$$
\frac{\partial E}{\partial w_0} = 2cw_0 - \sum_{l} 2(Y^l - f(X^l))f(X^l)(1 - f(X^l)) \tag{5}
$$

Similarly, we could derive formulas for w_1 and w_2.

2 Bayesian Networks [20 points]

2.1 Representation and Inference [12 points]

Consider the following Bayes net:

$$A \rightarrow B \rightarrow C \leftarrow D$$

1. Write the joint probability $P(A, B, C, D)$ for this network as the product of four conditional probabilities.

★ **SOLUTION:**

$$P(A, B, C, D,) = P(A)P(B|A)P(D)P(C|B, D) \tag{6}$$

2. How many independent parameters are needed to fully define this Bayesian Network?

★ **SOLUTION:** We need 8 independent variables.

3. How many independent parameters would we need to define the joint distribution $P(A, B, C, D)$ if we made *no* assumptions about independence or conditional independence?

★ **SOLUTION:** $2^4 - 1$

4. [6 pts] Consider the even simpler 3-node Bayes Net

$$A \rightarrow B \rightarrow C$$

Give an expression for $P(B = 1|C = 0)$ in terms of the *parameters of this network*. Use notation like $P(C = 1|B = 0)$ to represent individual Bayes net parameters.

★ **SOLUTION:** Using Bayes' Rule, we have:

$$\begin{aligned} P(B=1|C=0) &= \frac{P(C=0|B=1)P(B=1)}{P(C=0)} \\ &= \frac{(1-P(C=1|B=1))P(B=1)}{P(C=0)} \end{aligned} \tag{7}$$

Equations for $P(B=1)$ and $P(C=0)$ can be derived by:

$$\begin{aligned} P(B=1) &= P(B=1|A=0)P(A=0) + P(B=1|A=1)P(A=1) \\ P(C=0) &= 1 - P(C=1) \\ &= 1 - (P(C=1|B=0)P(B=0) + P(C=1|B=1)P(B=1)) \end{aligned} \tag{8}$$

Note that 5 independent parameters for this Bayesian Network are: $P(A=1), P(B=1|A=0), P(B=1|A=1), P(C=1|B=0), P(C=1|B=1)$. The above equations can be written in terms of these parameters using complement rule in probability.

2.2 Learning Bayes Nets [8 points]

Suppose you want to learn a Bayes net over two binary variables X_1 and X_2. You have N training pairs of X_1 and X_2, given as $\{(x_1^1, x_2^1), (x_1^2, x_2^2), (x_1^3, x_2^3), \ldots, (x_1^N, x_2^N)\}$. Given two datasets A and B, we know that the data in B is generated by $x_2^j = F(x_1^j, \theta) + \epsilon$ for all training instances j where θ and ϵ are two unknown parameters. We don't have any information on how dataset A is generated. Let BN denote the Bayes Net with no edges, and BN' denote the BN with an edge from X_1 to X_2. For both of these Bayes net, we learn its parameters using maximum likelihood estimation.

1. Which Bayes net is better to model the dataset A? Explain your answer.

★ **SOLUTION:** BN'. We shouldn't make any independence assumption between variables since we don't know how data has been generated.

2. Which Bayes net is better to model the dataset B? Explain your answer.

★ **SOLUTION:** BN' since we know that X_1 and X_2 are not independent.

3 Expectation Maximization (EM) [15 points]

Consider again the simple Bayes Network from question 2: $A \rightarrow B \rightarrow C$. You must train this network from partly observed data, using EM and the following training examples:

example 1: A=1, B=1, C=0
example 2: A=1, B=?, C=0
example 3: A=0, B=0, C=1
example 4: A=0, B=1, C=1

Assume that we begin with *each independent parameter of this network initialized to 0.6* (recall that you enumerated these in question 2).

1. As we execute the EM algorithm, what gets calculated during the first E step?

   ★ **SOLUTION:** For each training example k:

$$E[B_k] = P(B_k = 1|A_k, C_k, \theta) = \frac{P(B_k = 1, A_k, C_k|\theta)}{P(B_k = 1, A_k, C_k|\theta) + P(B_k = 0, A_k, C_k|\theta)} \tag{9}$$

2. Give the value for this quantify, as calculated by the first E step.

   ★ **SOLUTION:** For $k = 2$:

$$\begin{aligned} E[B_2] &= \frac{\theta_{C=0|B=1}\theta_{B=1|A=1}\theta_{A=1}}{\theta_{C=0|B=1}\theta_{B=1|A=1} + \theta_{C=0|B=0}\theta_{B=0|A=1}\theta_{A=1}} \\ &= \frac{0.4 * 0.6 * 0.6}{0.4 * 0.6 * 0.6 + 0.4 * 0.4 * 0.6} = 0.6 \end{aligned} \tag{10}$$

3. What gets calculated during the first M step?

   ★ **SOLUTION:**

$$\begin{aligned} \theta_{A=1} &= \frac{\sum_{k=1}^{N} \delta(A_k = 1)}{N} \\ \theta_{B=1|A=0} &= \frac{\sum_{k=1}^{N} \delta(A_k = 0)E[B_k]}{\sum_{k=1}^{N} \delta(A_k = 0)} \\ \theta_{B=1|A=1} &= \frac{\sum_{k=1}^{N} \delta(A_k = 1)E[B_k]}{\sum_{k=1}^{N} \delta(A_k = 1)} \\ \theta_{C=1|B=1} &= \frac{\sum_{k=1}^{N} \delta(C_k = 1)E[B_k]}{\sum_{k=1}^{N} E[B_k]} \\ \theta_{C=1|B=0} &= \frac{\sum_{k=1}^{N} \delta(C_k = 1)(1 - E[B_k])}{\sum_{k=1}^{N} (1 - E[B_k])} \end{aligned} \tag{11}$$

4. Give the value for this set of quantities, as calculated by the first M step.

   ★ **SOLUTION:**

$$\begin{aligned} \theta_{A=1} &= 0.5 \\ \theta_{B=1|A=0} &= 0.5 \\ \theta_{B=1|A=1} &= 0.8 \\ \theta_{C=1|B=0} &= 0.71 \\ \theta_{C=1|B=1} &= 0.38 \end{aligned} \tag{12}$$

4 Midterm Review Questions [15 points]

Here are short questions (some from previous midterm exams) intended to help you review for our midterm on October 18.

4.1 True or False Questions [9 points]

If true, give a 1-2 sentence explanation. If false, a counterexample.

1. As the number of training examples grows toward infinity, the MLE and MAP estimates for Naive Bayes parameters converge to the same value in the limit.

★ **SOLUTION:** False, since we have not made any assumption about the prior. A simple counterexample is the prior which assigns probability 1 to a single choice of parameter θ

2. As the number of training examples grows toward infinity, the probability that logisitic regression will overfit the training data goes to zero.

★ **SOLUTION:** True.

3. In decision tree learning with noise-free data, starting with the wrong attribute at the root can make it impossible to find a tree that fits the data exactly.

★ **SOLUTION:** False.

4.2 Short Questions [6 points]

1. The Naive Bayes algorithm selects the class c for an example x that maximizes $P(c|x)$. When is this equivalent to selecting the c that maximizes $P(x|c)$?

★ **SOLUTION:** $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$, so finding the c that maximizes $P(c|x)$ is equivalent to finding the c that maximizes $P(x|c)$, if the prior $P(c)$ is uniform.

2. Imagine you have a learning problem with an instance space of points on the plane. Assume that the target function takes the form of a line on the plane where all points on one side of the line are positive and all those on the other are negative. If you are asked to choose between using a decision tree or a neural network with no hidden layer, which would you choose? Why?

★ **SOLUTION:** Neural network. A decision tree is not able to learn some of the linear functions on a plane (e.g., $y = x$).