CS 374 Assignment #4
Neural Networks and the Backpropagation Algorithm

Due the week of February 25, 2008

—

Of all machine learning techniques, neural networks are probably best known, at least in name, by the general public. Foundational work on the implementation of artificial neurons (and networks) was done in the 1940s, before "Artificial Intelligence" was established as a field of study. (The year 2006, by the way, marked the fiftieth anniversary of the Dartmouth Conference, at which AI was formally established as a field of research and was given its name.) Since that time, they have found support among computer scientists as a mechanism for solving optimization problems, while also finding popularity in the world of science fiction. Neural networks are of central importance, for example, in *2001: A Space Odyssey*, in which they are highlighted as the key reason for the existence of HAL.

This week we will study artificial neural networks and the backpropagation learning algorithm. Note that it will be another role-playing week, though this time the focus will be on analysis of a research paper and critique of the analysis.

1 Backpropagation

1.1 Reading

Please read

- Mitchell, Chapter 4 (at least through Section 4.6.2), and
- Witten and Frank, pages 227-233.

Alpaydin's text also includes a chapter on this topic, which you're welcome to read as well. The relevant sections are 11.1-11.9. (The first four will be familiar to you from Assignment 1.) If you read both Mitchell and Alpaydin, you'll find that they derive slightly different rules for updating multilayer perceptrons. As you know from your first assignment, determining the delta rule for updating a perceptron involves computing the gradient of the error surface. So the delta rule is dependent on how error is defined and that, in turn, is dependent upon the output function of the perceptron. Since Mitchell and Alpaydin begin with different functions, they derive different delta rules. I bring this to your attention for two reasons – first, to avoid any confusion and second, because you will implement backpropagation this week and I would like you to follow Mitchell's explanation of the algorithm.

1.2 Exercises

This week your primary exercise will involve the implementation of the backpropagation algorithm. While this algorithm is conceptually fairly simple (the big idea, after all, is walking along the error surface to find the weights that minimize squared error), the details can be tricky. The algorithm is certainly non-trivial to debug. Please don't wait until the last minute to begin your implementation.

As all of the texts above point out, neural networks are not necessarily the representation of choice for all classification tasks. They are difficult to understand, and they take a long time to train. They are, however, fairly robust to noise, can be trained incrementally, and are flexible enough to be applied to both classification and regression. So they are most definitely worthwhile to study.

We will focus on applying neural networks to classification tasks. We do this in order to better understand how neural networks and backpropagation compare to the techniques we have already studied and will study later.

Please implement the stochastic version of backpropagation as outlined in Table 4.2 in Mitchell. In order to constrain the task somewhat, you may make the following assumptions:

- You never need to model a neural network with more than one hidden layer.
- The neural network will be used to represent classification tasks.
- The inputs and classes are nominal- (i.e., discrete-)valued.
- The input to your system will be a data file in the "ARFF" format. This will give you an opportunity to clean up or reuse your file-reading code from the earlier Naive Bayes assignment.

In order to handle nominal attributes and classes in a neural network, you will need to treat the values as if they are numeric. Say that you have a binary-valued attribute or class. This is easy – you can simply treat such an attribute or class as if it has one of the two possible values 0 and 1. If an attribute or class has multiple possible values, then you will need to create a node for each possible value. Say, for instance, that you are implementing a neural network for the "weather" data. The "outlook" attribute can take one of three values – sunny, rainy, overcast. You would handle this by creating three input nodes. For any given example, only one of these nodes would have the value 1; the other two would be 0.

You can test your implementation on the weather and contact lens data sets. Recall that both of these are quite small – in fact, when you implemented Naive Bayes, I had you use the full data sets for both training and test. If you do that for this assignment, you won't get any sense of the generalization capabilities of multilayer perceptrons. So I'd like you to do a "leave one out" cross-validation for the weather data set and a similar evaluation for the contact lens data set. In a "leave one out" evaluation, you remove an example from the training set and reserve it for testing. Since the weather data set has 14 examples, you will do this 14 times.

2 Weka

You might find it useful to run an implementation of backpropagation to get a sense of the learning results you might expect for each data set. Weka is a wonderful resource that includes implementations of many popular machine learning algorithms. To use it, begin by copying weka.jar, which you can find in

```
~andrea/shared/cs374/
```

Then type

```
java -jar weka.jar
```

This will start up a GUI. Click on the button that's labelled "Explorer". This will open another window that will allow you to select machine learning algorithms and datasets on which you can test them.

Begin by selecting a data set. You can do this by clicking on "Open file..." and then selecting an "aarf" file of your choice. Once you've selected a data file, click on "Classify" and then "Choose". In the "Functions" folder, click on MultilayerPerceptron. Then click on "Start". The output of the classifier will appear in the right half of the window.

If you have any trouble working with Weka, let me know.

3 Multitask Learning

Witten and Frank tell us that:

> The same technique can be applied to predict several targets, or attribute values, simultaneously by creating a separate output unit for each one. Intuitively, this may give better predictive accuracy than building a separate classifier for each class attribute if the underlying learning tasks are in some way related.

This idea – *multitask learning* – has been extensively studied.

3.1 Reading

Please read

- "Multitask Learning", a paper by Rich Caruana that appeared in the journal *Machine Learning* in 1997.

You can find the link to this paper by going to the Assignments page for this course. This paper describes multitask learning first in the context of backpropagation but then extends it to other learning techniques. More specifically, Section 5 talks about multitask learning in the context of k-nearest neighbor and decision trees. Some of you may already be familiar with these. If not, don't worry about it. We'll cover these later in the semester. In any case, I would like you to focus on the discussion of multitask learning and backpropagation.

3.2 Exercise

At this point in the semester you have accumulated experience with a few different machine learning algorithms. And (as of week 3) have had some experience reading and discussing machine learning research. This should place you in a good position to critically analyze the multitask learning paper. For the tutorial session you should be prepared to not only discuss the general results presented in the paper, but should be able to contribute to a lively discussion that critiques the work.

In order to better prepare for this discussion, one member of each group will write and present a critical analysis of the paper. The others will write and present a critique of the presenters' analysis.

3.3 Reviewing Machine Learning Research Papers

In order to help you prepare for this task, I have attached four of my own reviews here. These include both journal and conference submissions, both positive and negative reviews. I have removed information that might identify the authors wherever possible. I have also (largely) removed sections of my reviews that provide simple editorial suggestions, such as grammar and spelling corrections. The four reviews are as follows:

- a very negative review of an article submitted to a machine learning journal
- a positive first review of an article submitted to a journal (I was asked to do a second review of this article, which has now been published.)
- a very negative review of a conference submission
- a mildly negative review of a conference submission

As you'll see, a good review, should include both a description of the work and a critique. The description is important, as those reading the review can then assess the extent to which you actually understood the work. The critique, of course, is essential as well. You should include both positive and negative comments. You should comment on such areas as importance, technical soundness, and clarity.

As you read the reviews, notice that while I describe the work, I spend far less time on description than I do on issues and questions raised by the work. Note also that my style of writing is less like an essay and more like a laundry list of ideas. While this is appropriate review style, I expect that your reviews will be written in essay form. (Typically, my reviews are written in the context of having to consider on the order of ten papers simultaneously, so I focus less on my prose and more on the points I need to make.) You will also note that I often include references to other work in the field. If you are able to do this in your review, that's great. I do not expect it, however, as your knowledge of the field is clearly limited to the work we have done this semester (or in the AI class last semester). Finally, I like to include editorial (grammar and spelling) corrections to help the authors. You should not include such comments in your final review. You should, however, comment on the clarity of presentation and on the writing style in general.

Please type your report, which should be 2-4 pages long (12-point font, 1.5 spacing).