CS 374 Assignment #5
Neural Networks and the Backpropagation Algorithm
A First Look at Deep Learning

Due the week of March 22, 2021

—

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. (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. So we're in year 64 now.) Since that time, artificial neural networks have found popularity in the world of science fiction. They 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. On a note more relevant to this course, artificial neural networks have found support among computer scientists as a mechanism for solving optimization problems, and most recently, have seen new enthusiasm with the success of deep learning.

This week we will study artificial neural networks and the backpropagation learning algorithm. We will also consider the relationship of these to deep learning.

1 Backpropagation

1.1 Reading

Please read

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

1.2 Exercises

This week your primary exercise will involve the implementation of backpropagation for training a feedforward neural network. While backpropagation 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. I also strongly recommend working with your tutorial partner(s). Two minds and two sets of eyes to find bugs will be very helpful here.

As 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 problems. 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 for classification only – not for regression.
- 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. (Of course, binary-valued attributes can be handled in this way as well.)

You can test your implementation on the weather, contact lens, and Titanic data sets, available at:

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

For the weather and contact lens data sets, you should do at least two types of evaluation: train and test on the full set, as well as leave-one-out cross validation. For Titanic, you can do both as well, but as this data set is bigger, a 10-fold cross validation is preferable to leave-one-out. To do this, you need to partition the data into 10 sets. Hold out one set at a time for testing, training on the remaining nine folds.

In all cases, output the number correct, number incorrect, and percentage correct.

Of course, you should feel free to try your implementation on other data sets. You can find more in my shared 374 directory:

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

Turn in both paper and electronic versions of your code and be prepared to present it in the tutorial meeting.

2 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.

2.1 Reading

Please read

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

This paper investigates the utility of multitask learning first in the context of backpropagation but then extends it to other learning techniques.

2.2 Exercise

At this point in the semester you have accumulated experience with a few different machine learning algorithms. And you have had some experience reading and discussing machine learning research. This should place you in a good position to critically analyze the paper.

2.3 Reviewing Machine Learning Research Papers

In week 3, I provided questions to help guide your thinking as you reviewed the "Skewing" paper. This week we'll expand on those to get you even closer to the point of writing a complete review of a machine learning paper.

A good review should include both a description of the work (claims made by the authors, how they support the claims, etc.) and a critique. The description is important, as those reading the review can then assess the extent to which you actually understood the work. However, don't make the summary too long; focus on the "big ideas." The critique, of course, is essential. You should include both positive and negative comments.

In general, a critique should comment on such areas as importance, references (i.e., is the work placed in context?), technical soundness, and clarity. As I said in week 3, you probably don't have enough experience yet to *really* comment on importance and references, but there may be some things you can say about them! Also, in week 3 I did *not* mention clarity, as I wanted you to focus on the technical aspects of the paper. Technical matters are still most important, but from now on please do comment on clarity as well.

I am less concerned with the overall "flow" of your critique than I am with your ability to read and analyze carefully, to critique fairly, and to articulate your comments (both positive and negative) as clearly as possible. You can choose to do this in the form of an essay or paper, or you may choose to do it in the form of sections addressing topics such as those below.

- **Summary of the paper:** What is the problem that the authors are addressing? What are their claims? How do they go about providing support for those claims?
- **Importance/Novelty:** How important is the problem that the authors address? How significant are the results? Does the paper break new ground? Does it identify a new problem? Does it present promising new ideas for a well-known problem?
- **Approach:** Does the paper present the approach to address the problem in sufficient detail? Are new ideas, theoretical results, or experiments described in sufficient detail? If, for instance, the paper presents an experiment, would you be able to replicate it?
- **Validation:** Does the paper show convincing evidence to support the claims? Do the authors present solid arguments (descriptive, theoretical, empirical) to support their conclusions? Is the work technically sound?
- **Context:** Do the authors place the research in the context of previous work and cite relevant work appropriately?
- **Readability:** Is the paper well-written and easy to understand? Is it well-organized and easy to follow?
- **Current Status/Limitations:** Does the paper clearly state how much work has been done to date? Does it identify its own limitations?

Please type your review which should be 2-4 pages long (12-point font, 1.5 or 2.0 spacing). The write-up will be the foundation for the discussion between you and your tutorial partner.

3 Deep Learning

In all of our learning problems so far, we have assumed that someone is able to provide training data already formulated as vectors of attribute values. We haven't explicitly considered whether the chosen attributes are the best or most appropriate for the problem we're solving. Deep Learning is concerned, among other things, with the problem of learning representations.

Please read

- Chapter 14 in the online *Deep Learning* book.

I do not expect you to understand it in detail. For now, I want you to get the "big ideas". Be prepared to discuss and answer questions about autoencoding at a high level.

4 What you will be turning in this week

This week, please turn in

- A pdf with your summary and critique of the Caruana paper.

- All of your code to read data, run Backprop, and test your classifier. This should be in a single tar'd and compressed file.

To turn these in, follow the link on the course website or in the Glow module for Week 5.