Introduction to Artificial Neural Networks

Lecture 6:

Applications of Multi-Layer Perceptrons

By: Ali Motie Nasrabadi

Outline

- 1.Applications of Feed-Forward Networks
- 2. Brain Modeling
  - What Needs Modeling?
  - Development, Adult Performance, Neuropsychology
  - Analysis of Hidden Unit Representations
- 3. Real World Applications
  - Data Compression - PCA
  - Time Series Prediction
    - Predicting hourly temperature
  - Character Recognition
  - Driving - ALVINN

Applications of Feed-Forward Networks

- Brain modeling **: The scientific goal of building models of how real brains work. This can potentially help us understand the nature of human intelligence, formulate better teaching strategies, or better remedial actions for brain damaged patients.**
- Artificial System Building **: The engineering goal of building efficient systems for real world applications. This may make machines more powerful, relieve humans of tedious tasks, and may even improve upon human performance.**

These should not be thought of as competing goals. We often use exactly the same networks and techniques for both. Frequently progress is made when the two approaches are allowed to feed into each other. There are fundamental differences though, e.g. the need for biological plausibility in brain modeling, and the need for computational efficiency in artificial system building. Simple feed-forward neural networks are surprisingly effective for both.

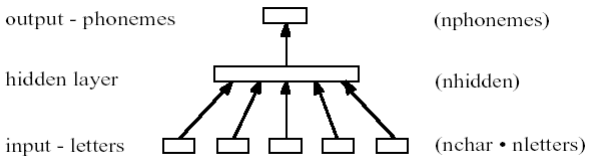
Brain Modelling – What Needs Modelling?

- **The comparison between brains and models fall into three broad categories:**
- Development **: Comparisons of children's development with that of our models – this will generally involve both maturation and learning.**
- Adult Performance **: Comparisons of our mature models with normal adult performance – exactly what is compared depends on what we are modeling.**
- Brain Damage / Neuropsychological Deficits **: Often performance deficits, e.g. due to brain damage, tell us more about normal brain operation than normal performance.**

We shall consider these issues for a particularly simple and familiar task – "reading aloud" or "text to phoneme conversion". Similar considerations will apply to a wide range of psychological tasks that can be reduced to forms of input output mappings.

The NETtalk Model of Reading

- **The *NETtalk model* of Sejnowsi & Rosenberg (1987) is a basically a MLP which generates output phonemes corresponding to the letter in middle of an input window:**



Input: 29 bits per character, 7 character window (for context). *one-of-n-encoding* — for each character, 1 bit is on, the rest are off.
Hidden units: One layer consisting of 80 hidden units, completely connected to inputs and outputs.
Output: 26 units to represent 54 phonemes. A distributed representation based on articulatory features.
Training: Training on dictionary pronunciation and on transcriptions of speech

The NETtalk Model of Reading

- **The network can also be set up to figure out the letter-phoneme alignments for itself by assuming the alignment that best fits in with its expectations (Bullinaria, 1997). We'll look at the results from a typical series of simulations. The network training data consisted of all 2998 English monosyllabic words, and the testing data was a standard set of 166 made up pronounceable non-words. It was trained using a standard learning algorithm (back-propagation with momentum) with 300 hidden units.**
- Results: **Understandable speech after 10 learning cycles. After 50 training cycles, apparent error was 5%; generalization error was about 22%. Existing expert system (DecTalk) performed better, but required about 10 person-years of linguistic analysis to generate rules; Nettalk require about 1 month to produce**

Points about Nettalk

- Representation — useful to put knowledge into the representation to
- make problem easier to learn (output representation). Where this cannot
- be done, use one-of-n encoding.

Lecture 6-7

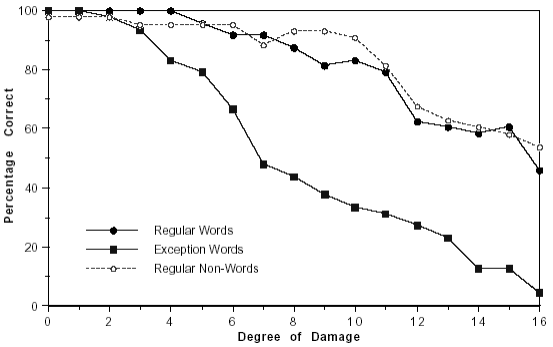
Adult Performance = Trained Network Performance

- Having succeeded in building accurate models of children's development, one might think that our adult models (fully trained neural networks) require little further testing. In fact, largely due to better availability and reliability of the empirical data, there are a range of adult performance measures that prove useful for constraining our models, such as:
  - Accuracy – basic task performance levels, e.g. how well are particular aspects of a language spoken/understood, or how well can we estimate a distance?
  - Generalization – e.g. how well can we pronounce a word we have never seen before , or recognize an object from an unseen direction?
  - Reaction Times – response speeds and their differences, e.g. can we recognize one word type faster than another, or respond to one color faster than another?
  - Priming – e.g. if asked whether *dog* and *cat* are real words, you tend to say yes to *cat* faster than if you were asked about *dot* and *cat* (this is *lexical decision priming*).
  - Speed-Accuracy Trade-off – across a wide range of tasks your accuracy tends to reduce as you try to speed up your response, and vice-versa.

Lecture 6-8

Brain Damage = Network Damage

- **Neural network models have natural analogues of brain damage – removal of sub-sets of neurons and connections, adding noise to the weights, scaling the activation functions. If we damage the reading model, the regular items are more robust than the irregulars:**



Lecture 6-9

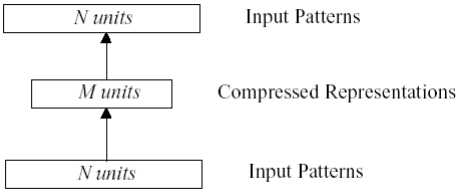
Real World Applications

- **The real world applications of feed-forward networks are endless. Some well known ones that get mentioned in the recommended text books are:**
  - **1.Airline Marketing Tactician (Beale & Jackson, Sect. 4.13.2)**
  - **2. Backgammon (Hertz et al., Sect. 6.3)**
  - **3. Data Compression – PCA (Hertz et al., Sect. 6.3; Bishop, Sect. 8.6)**
  - **4. Driving – ALVINN (Hertz et al., Sect. 6.3)**
  - **5. ECG Noise Filtering (Beale & Jackson, Sect. 4.13.3)**
  - **6. Financial Prediction (Beale & Jackson, Sect. 4.13.3; Gurney, Sect. 6.11.2)**
  - **7. Hand-written Character Recognition (Hertz et al., Sect. 6.3)**
  - **8. Pattern Recognition/Computer Vision (Beale & Jackson, Sect. 4.13.5)**
  - **9. Protein Secondary Structure (Hertz et al., Sect. 6.3)**
  - **10. Psychiatric Patient Length of Stay (Gurney, Sect. 6.11.1)**
  - **11. Sonar Target Recognition (Hertz et al., Sect. 6.3)**
  - **12. Speech Recognition (Hertz et al., Sect. 6.3)**

Lecture 6-10

Data Compression - PCA

- **An *auto-associator* network is one that has the same outputs as inputs. If in this case we make the number of hidden units *M* smaller than the number of inputs/outputs *N*, we will have clearly compressed the data from *N* dimensions down to *M* dimensions.**

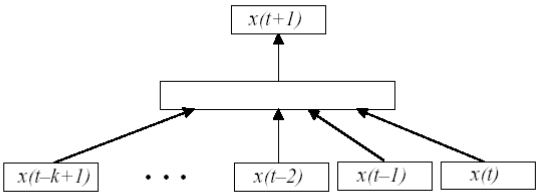


Data Compression - PCA

Such data compression networks have many applications where data transmission rates or memory requirements need optimising, such as in image compression. They clearly work by removing the redundancy that exists in the data. It can be shown that the hidden unit representation spans the *M* *principal components* of the original *N* dimensional data, so standard PCA considerations apply (Bourland & Kamp, 1988).

Time Series Prediction

- **Neural networks have been applied to numerous situations where** *time series prediction* **is required – predicting weather, climate, stocks and share prices, currency exchange rates, airline passengers, etc. We can turn the temporal problem into a simple inputoutput mapping by taking the time series data $x(t)$ at k time-slices t, $t-1$, $t-2$, …, $t-k+1$ as the inputs, and the output is the prediction for $x(t+1)$.**



- Examples:
- **Co-winner of the Santa Fe Prediction Contest, 1989.**
- **Prediction of hourly temperature to aid in prediction of energy**
- **consumption for a power company.**
- **Tax Revenues for the Utah Legislature since 1997. MLP's combine with**
- **other techniques to get as accurate prediction as possible.**

Predicting hourly temperature

- **Reference: IEEE Trans. on Power Systems Vol. 11(2) pp. 870–876, 1996.**
- **Motivation:**
  - **Power utility companies want to predict demand throughout the day**
  - **Demand is strongly related to temperature**
  - **Hourly temperature predictions acquired from meteorological services may be too expensive.**
- **Goal: Predict hourly temperature from past hourly temperatures and predicted daily highs and lows. Prediction for the next seven days.**

System description

- Inputs:
  - **24 hourly temperatures on the day prior to the first forecast day, ending at midnight.**
  - **high and low temperatures for two days prior to the first forecast (4 in all),**
  - **forcasted high and low temperatures for the days to be forecast, as provided by a weather service. (This is to use information about cold fronts etc which could not be inferred from previous day's temperatures).**
- Outputs: **Hourly temperatures, starting at 1am for the next 7 days (168 predictions).**

- Constituents: **A cascade of seven daily forecasters. 24 hourly predictions and high and low of predictor *i* feeding into predictor *i* 1. Each consists of a 28–n–24 MLP with sigmoid units. The number of hidden units n is not reported.**
- Input scaling: **Input is scaled to lie between 0 and 1.**
- Output scaling: **Outputs (which lie between 0 and 1) are scaled each day so that the predicted high and low are the same as the "desired" high and low (actual if they are known; predicted by the weather service otherwise).**
- Training: **is incremental and on-line. Prediction error is computed after each day when the actual temperatures become available. At this points the weights are adjusted.**

Lecture 6-17

Performance

- **Trained on 3 years of data provided by eight utility companies.**
- **Trained on historic data, replacing the high/low forecast (which were not available) with the actual highs and lows.**
- **Performance criteria:** $e(k) = T_{scaled}(k) - T_{actual}(k).$
  - **Error:**
  - **Mean Absolute Error (MAE):** $\frac{1}{n}\sum_{k=1}^{n} |e(k)|$
  - **Standard deviation of error.**
  - **4. Mean Absoluter Error in time: mean absolute error between the predicted time of the high (or low) and the time it actually occurred.**

Lecture 6-18

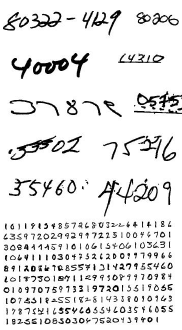
Results

- **MAE for 1-day ahead forecasts: 1.48deg*F*, compared to 1.89deg for baseline predictor.**
- **Error is 0, 29.4% of the time,**
- **At the time of actual high (low) MAE is 0.55deg (0.46deg).**
- **Mean absolute error in predicted high (low) is 1.25 hours (1.38 hours) for 1-day ahead forecasts.**
- **Unusual days – Onset of cold or warm front is indicated by a large change in the high or low between two subsequent days. MAE is larger on such days. However, error does not increase much. Let D be the larger of the differences between the highs on two subsequent days and the lows on the same two days.**

Hand-written Character Recognition

- **The neural network literature is full of *pattern recognition* applications. Typically one takes pixelated image values as the network input and that maps via layers of hidden units to a set of outputs corresponding to possible classifications of the image.**
- **A typical example by Le Cun et al. (1989) was designed to recognize hand-written ZIP codes (numerical postal codes). The input was a 16 ´ 16 array that received pixelated images of hand-written digits scaled to a standard size, and this fed through three layers of hidden units to ten output units each corresponding to one of the digits 0–9. The first layer of hidden units contained 12 feature detectors (8 ´ 8), and the second contained 12 feature detectors (4 ´ 4).**

- Each unit in each detector had a 5 ´ 5 receptive field in the earlier layer, and weight sharing ensured that they all detected the same feature in different parts of the retina. The third hidden layer had 30 units fully connected to the second hidden layer and the outputs.
- The network was trained on 7300 digits with ~1% errors and tested on 2000 digits with~5% errors. Pruning by Optimal Brain Damage improved the performance further.



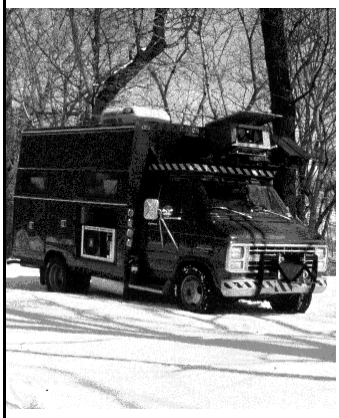
Lecture 6-21

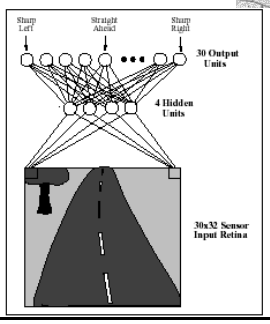
Driving – ALVINN (Autonomous Land Vehicle In a Neural Network)

- Pomerleau (1989) constructed a neural network controller ALVINN for driving a car on a winding road. The inputs were a 30 ´ 32 pixel image from a video camera, and an 8 ´ 32 image from a range finder. These were fed into a hidden layer of 29 units, and from there to a line of 45 output units corresponding to direction to drive.
- The network was originally trained using back-propagation on 1200 simulated road images. After about 40 epochs the network could drive at about 5mph – the speed being limited by the speed of the computer that the network was running on.

Lecture 6-22

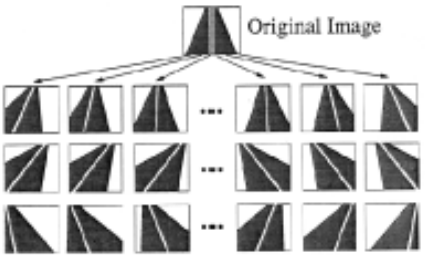
ALVINN

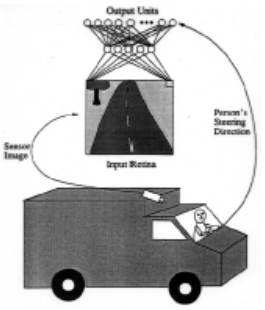




Lecture 6-23

ALVINN





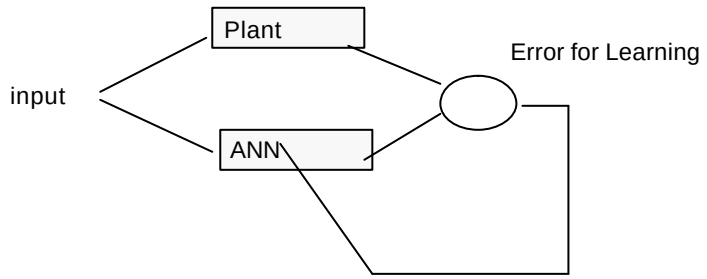
Lecture 6-24

ALVINN

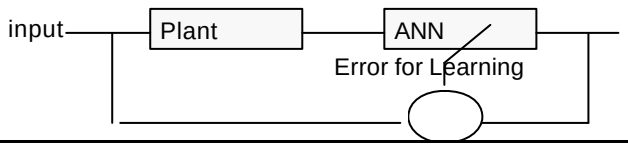
- Input: **Video image (30*32) + range-finder distance(8*32).**
- Output: **45 units representing steering angle.**
- Hidden: **45 units completely connected to input and output.**
- Training: **Originally on simulated images. Then in real use.**
- Results: **Drives 3mph along woodland road.**
- Current Developments: **Recently up to 15 decisions per second, good for 55mph driving along dirt, gravel, and other difficult surfaces. Trained by a real driver. Working on passing.**

System Identification

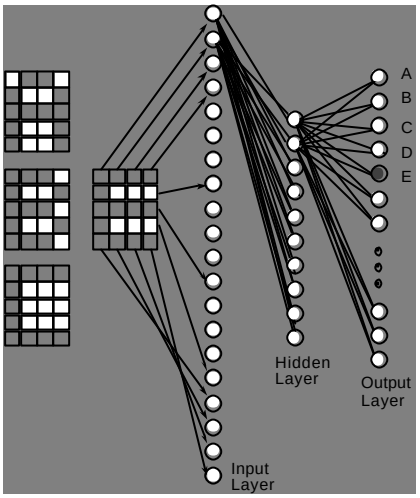
Model of Plant



Inversed Model of Plant



Aside on biological plausibility



V's