Applications of Multi-Layer Perceptrons

Introduction to Neural Networks : Lecture 11

1. Applications of Feed-Forward Networks
2. Brain Modelling

   What Needs Modelling?

   Development, Adult Performance, Neuropsychology

   Analysis of Hidden Unit Representations
3. Real World Applications

   Data Compression - PCA

   Time Series Prediction

   Character Recognition and What-Where

   Driving - ALVINN

Applications of Feed-Forward Networks

We have already noted that there are two basic goals for neural network research:

Brain modelling : 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 modelling, and the need for computational efficiency in artificial system building. Simple feed-forward neural networks, such as MLPs, are surprisingly effective for both.

Brain Modelling – What Needs Modelling?

It makes sense to use all available information to constrain our theories/models of real brain processes. This involves gathering as much empirical evidence about brains as we can (e.g. by carrying out psychological experiments) and comparing it with our models.

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.

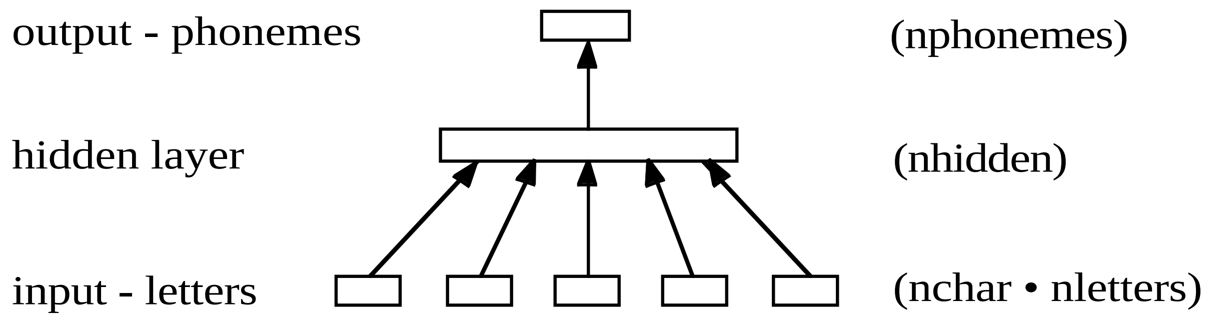
Adult Performance : Comparisons of our mature models with normal adult performance – exactly what is compared depends on what we are modelling.

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:

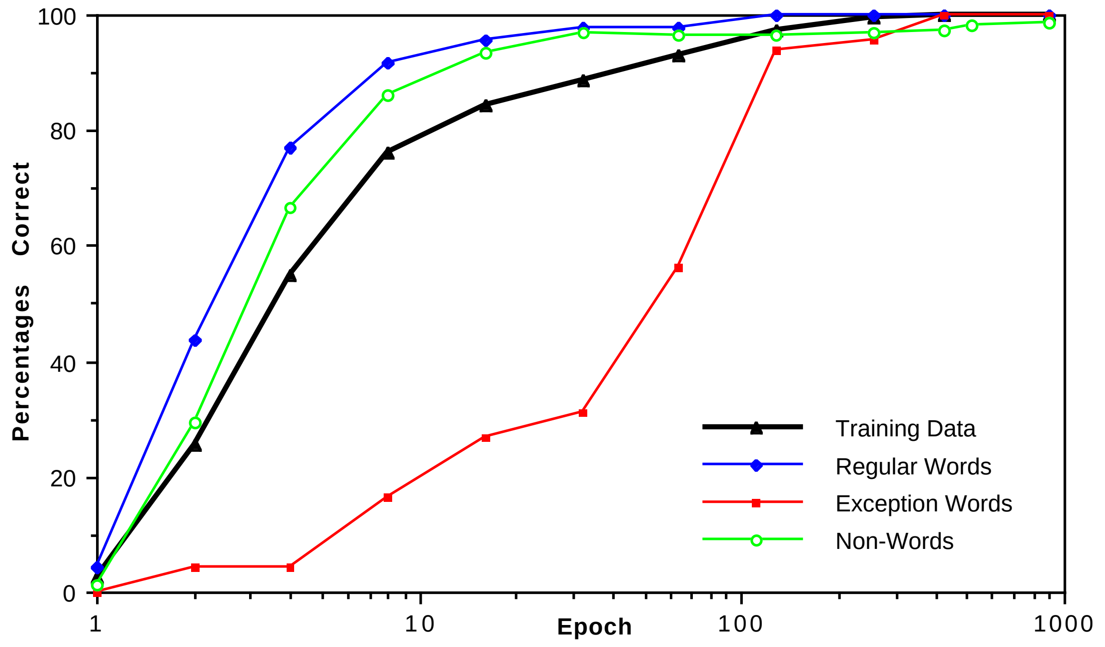


The network can also be set up to figure out the letter-phoneme alignments for itself by assuming the alignment that best fits in with its expectations (Bullinaria, 1997).

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.

Development = Network Learning

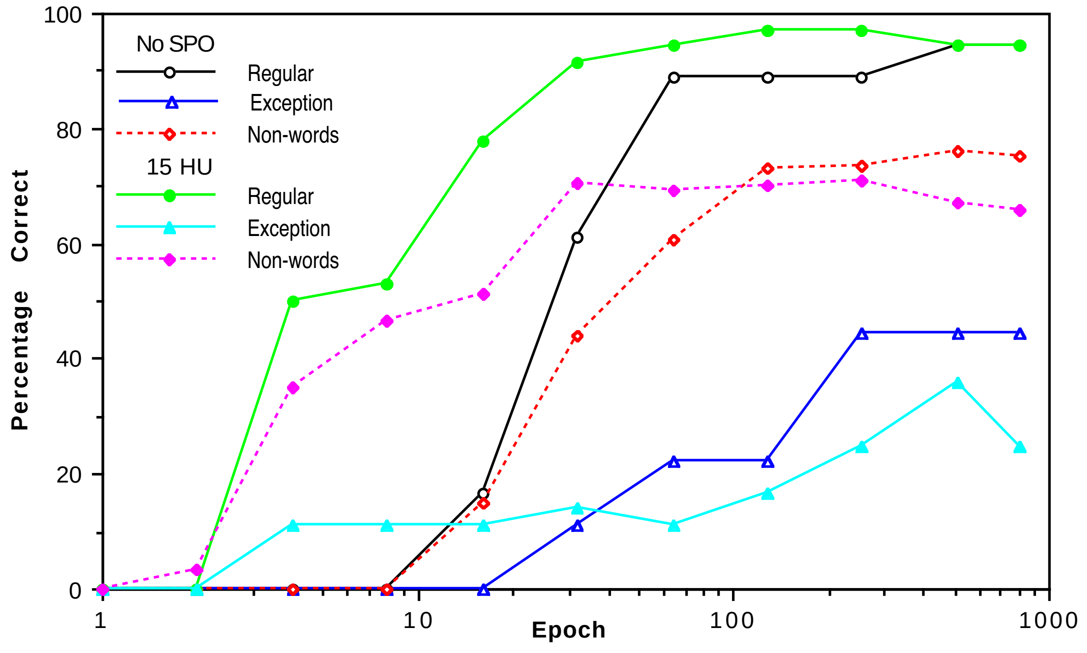
If neural network models are to provide a good account of what happens in real brains, we should expect their learning process to be similar to the development in children.



The networks find regular words (e.g. 'bat') easier to learn that exception words (e.g. 'yacht') in the same way that children do. It also learns human-like generalization.

Developmental Problems = Restricted Network Learning

Many dyslexic children exhibit a dissociation (performance difference) between regular and irregular word reading. There are many ways this can arise in network models:



1. Limitations on computational resources (e.g. only 15 hidden units)
2. Problems with learning algorithms (e.g. no SPO in learning algorithm)
3. Simple delay in learning (e.g. low learning rate η)

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 (vown fi gowpit?), or recognise an object from an unseen direction?

Reaction Times – response speeds and their differences, e.g. can we recognise one word type faster than another, or respond to one colour faster than another?

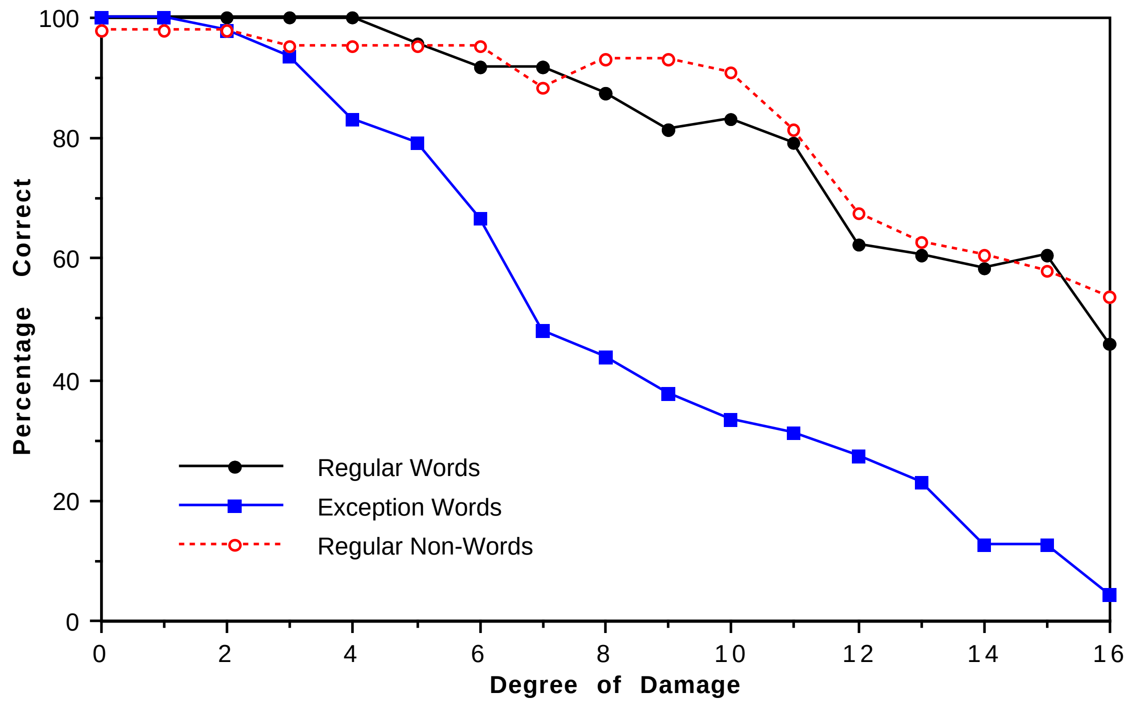
Priming – e.g. if asked whether *dog* and *cat* are real words, you tend to say yes to *cat* faster than if you were asked about *dot* and *cat* (this is *lexical decision priming*).

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.

Different performance measures will be appropriate to test different models.

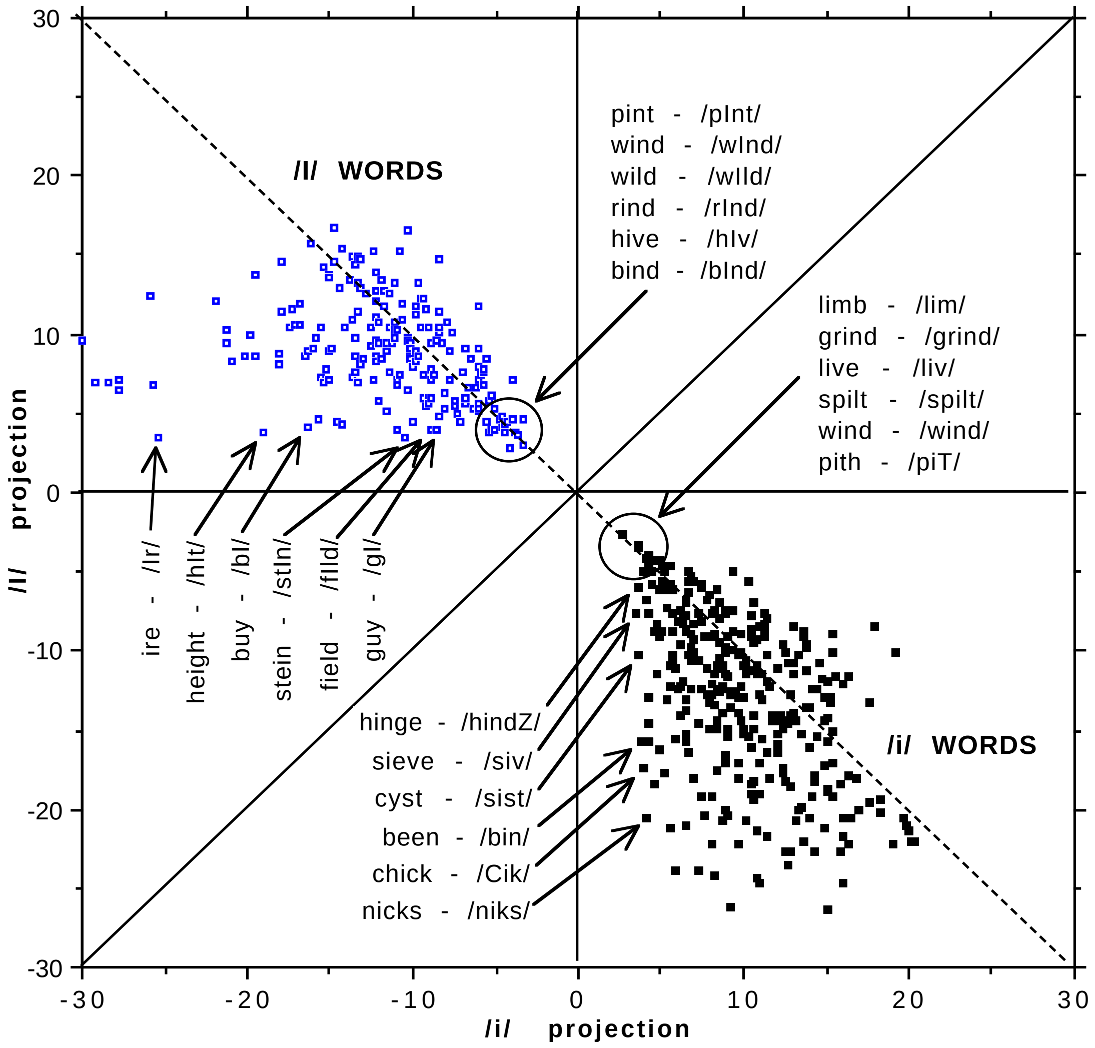
Brain Damage = Network Damage

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:



These neural network deficits are the same as seen in human acquired *Surface Dyslexia*.

Analysing the Internal Representations



One can look at the representations that the neural network learns to set up on its hidden units. Here we see the weight sub-space corresponding to the distinction between long and short ‘i’ sounds, i.e. the ‘i’ in ‘pint’ versus the ‘i’ in ‘pink’. The irregular words are closest to the border line. So, after net damage, it is these that cross the border line and produce errors first. Moreover, the errors will mostly be regularizations. This is exactly the same as is found with human surface dyslexics.

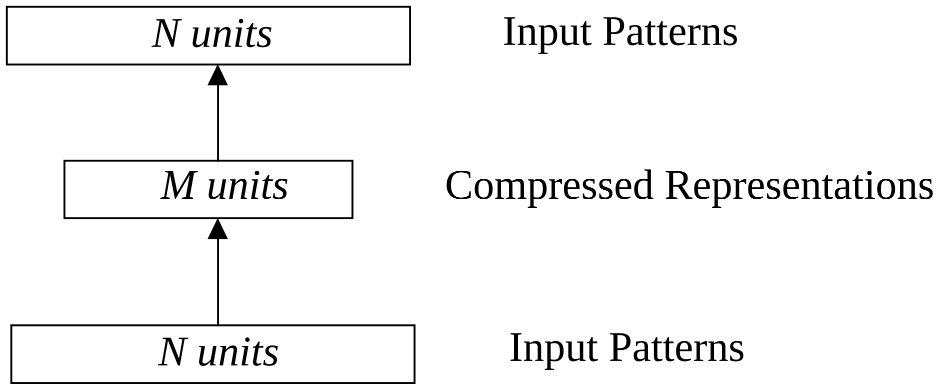
Real World Applications

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; Fausett, Sect. 7.4) •
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)

Data Compression - PCA

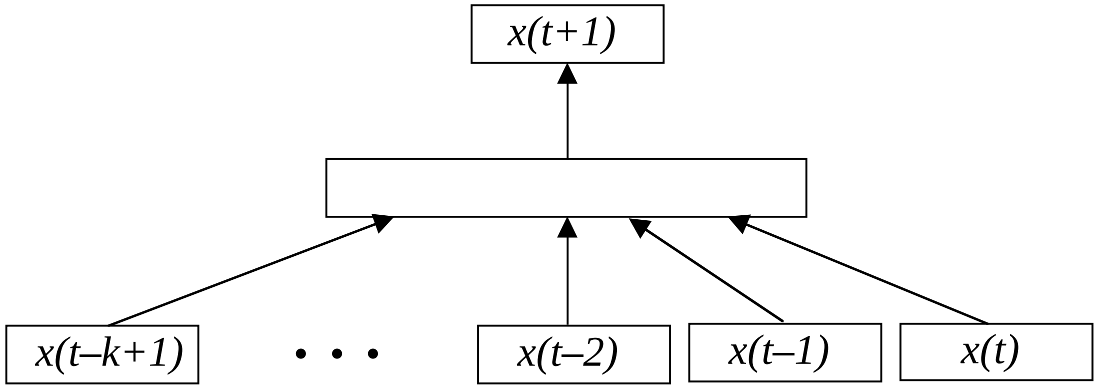
An ***auto-associator*** network is one that has the same outputs as inputs. If in this case we make the number of hidden units M smaller than the number of inputs/outputs N, we will have clearly compressed the data from N dimensions down to M dimensions.



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 input-output 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 $x(t+1)$.



Such networks can be extended in many ways, e.g. additional inputs for information other than the series *x(t)*, outputs for further time steps into the future, feeding the outputs back through the network to predict further into the future (Weigend & Gershenfeld, 1994).

Hand-written Character Recognition

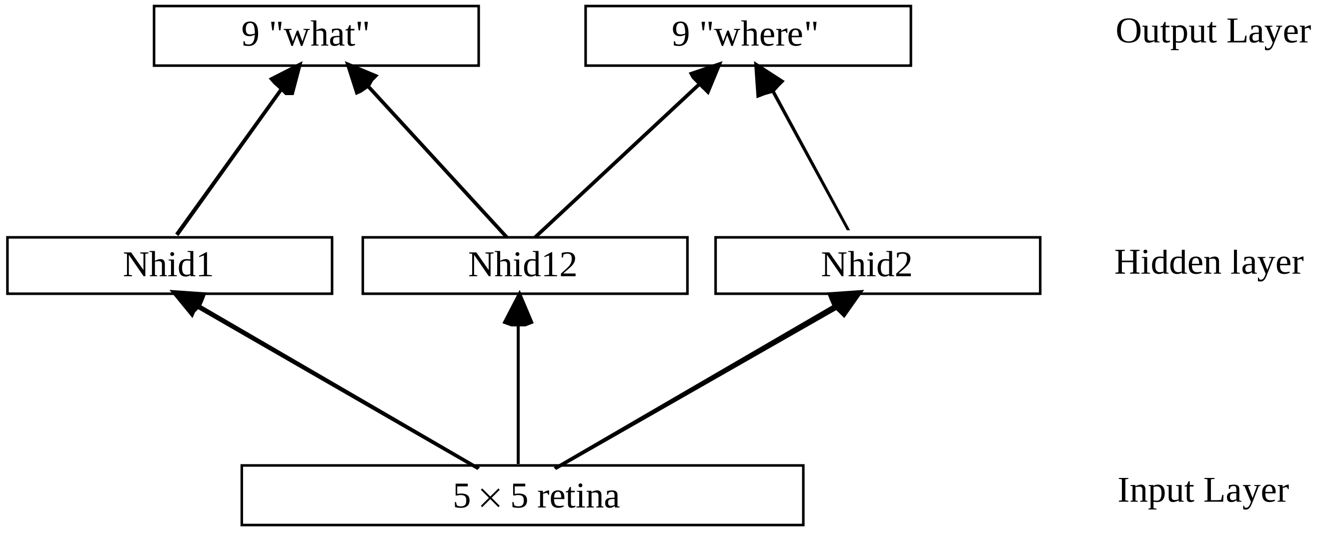
The neural network literature is full of ***pattern recognition*** applications. Typically one takes pixelated image values as the network input and that maps via layers of hidden units to a set of outputs corresponding to possible classifications of the image.

A typical example by Le Cun et al. (1989) was designed to recognise 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.

Processing “What” and “Where” Together

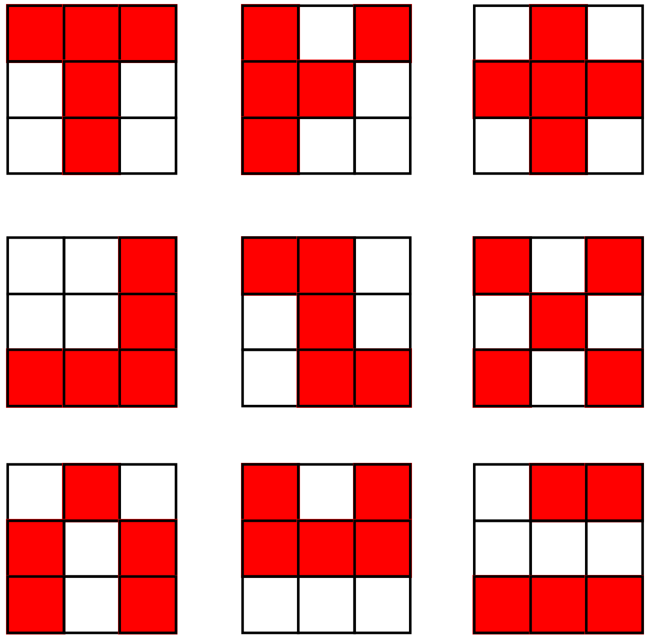
General pattern recognition requires the determination of “What” and “Where” at the same time, from the same pixelated images. It is not obvious if it is best to use one big network for both tasks, or separate networks. Experiments to test this have been carried out on simple 5×5 images on a network with parameterized architecture:



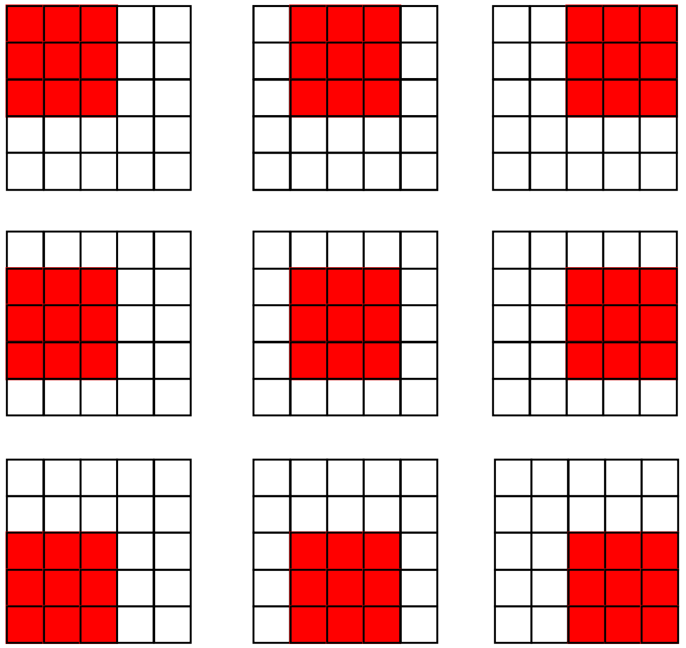
The conclusion depends on precisely what learning algorithm is used (Bullinaria, 2002).

Simplified "What-Where" Training Data

"What" = Nine 3 × 3 Patterns



"Where" = Nine Positions in 5 × 5 Retina



9 × 9 = 81 Training Patterns in total

e.g.	01110 00100 00100 00000 00000	100000000	010000000
	Input (5 × 5 = 25 units)	What (9 units)	Where (9 units)

Driving – ALVINN

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 neural network was running on.

In a later study the network learnt by watching how a human steered, and by using additional views of what the road would look like at positions slightly off course. After about three minutes of training, ALVINN was able to take over and continue to drive. ALVINN has successfully driven at speeds up to 70mph and for distances of over 90 miles on a public highway north of Pittsburgh. (Apparently, actually being inside the vehicle during the test drive was a big incentive for the researchers to develop a good neural network!)

References / Advanced Reading List

1. Bourland, H. & Kamp, Y. (1988). Auto-association by Multilayer Perceptrons and Singular Values Decomposition. *Biological Cybernetics*, **59**, 291-294.
2. Bullinaria, J.A. (1997). Modelling Reading, Spelling and Past Tense Learning with Artificial Neural Networks. *Brain and Language*, **59**, 236-266.
3. Bullinaria, J.A. (2002). To Modularize or Not To Modularize? In *Proceedings of the 2002 U.K. Workshop on Computational Intelligence: UKCI-02*, 3-10.
4. Le Cun, Y. et al. (1989). Back-propagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, **1**, 541-551.
5. Pomerleau, D.A. (1989). ALVINN: An Autonomous Land Vehicle in a Neural Network. In D.S. Touretzky (ed.), *Advances in Neural Information Processing Systems I*, 305-313. Morgan Kaufmann.
6. Sejnowski, T.J. & Rosenberg, C.R. (1987). Parallel Networks that Learn to Pronounce English Text. *Complex Systems*, **1**, 145-168.
7. Weigend, A.S. & Gershenfeld, N.A. (1994). *Time Series Prediction: Forecasting the Future and Understanding the Past.* Addison-Wesley.

Overview and Reading

1. We began by recalling the distinction between using neural networks for brain modelling and for artificial system building.
2. Then we looked at the main issues in brain modelling (development, adult performance, and neuropsychology) with reference to a reading model, and saw how one can better understand the operation of particular neural networks by looking at the patterns of activation on their hidden layer.
3. We ended by looking at a selection of real world applications, with five studied in particular detail: data compression, times series prediction, character recognition, the what-where task, and driving.

Reading

1. Hertz, Krogh & Palmer: Section 6.3
2. Gurney: Section 6.11
3. Beale & Jackson: Section 4.13
4. Ham & Kostanic: Chapters 6 to 10