

## Introduction

Cognitive modelling - representations which model both human behaviour and the cognitive processes on which that behaviour depends. Many different types - e.g. connectionist modelling, symbolic (rule-based) modelling.

### *Rule based approaches*

**Anderson's ACT-R model** - simulation of human memory and arithmetic skills. Demonstrates psychological phenomena can be simulated by collections of rules. Other examples are expert systems - e.g. for clinical diagnosis.

### *Neural networks*

Based on the micro-structure of the brain, rather than explicitly programmed rules. Connectionist models do replicate some of the characteristics of networks of neurons - e.g. capable of generalising from many specific cases - a process that underlies models of concept formation. Their ability to generate generalisations is intrinsic - there doesn't need to be an add-on to the model for it to perform generalisation. Content-addressable memory is also inherent to connectionist models - a natural property of human cognition.

## Local and distributed representation

Adult brains lose thousands of neurons daily - but our cognitive processing is not affected. Increasing levels of damage rarely produce catastrophic impairment immediately - instead, a slow decline in performance tends to occur - graceful degradation - indicates distributed, rather than local representation.

### *Local representation*

Research by **Hubel and Wiesel** on neurons in cat's brains show they respond to more complex visual aspects as recordings are made more 'deeply' in the

Tim Holyoake 2010, <http://www.tenpencepiece.net/>

## Connectionism

visual cortex. Perhaps there is a cell therefore that only responds to your grandmother - the hypothetical 'grandmother cell'. Simple to criticise - if you lose that cell, you would no longer be able to recognise your grandmother - and this doesn't happen.

### *Distributed representation*

Seven-segment LED display as an example of a distributed representation. No single segment represents a single digit - all are involved either by being 'on' or 'off'. The digits 6,7 and 9 can have a damaged segment yet still be recognised - there is some redundancy in their encoding.

When large numbers of units (segments) are involved, as is likely to be the case in the brain, then the representations become very robust to such damage. They therefore demonstrate graceful degradation - just like the way the human brain performs.

### Parallel Processing

**Braisby** - people use similarity between stimuli to generalise from instances to an organising concept. Connectionist models also generalise from simple cues to general schema, using pattern matching and pattern association.

### *Pattern matching*

Connectionist models are good at this, particularly if a 'best fit' rather than a precise match is required. It is a key part of many cognitive models - e.g. **Braisby** - in some models of categorisation. Classical view - all features on a list present => match; prototype theories also require features to be matched to those on a list.

### *Pattern associators*

If retrieval from a memory system is cued, then the process of pattern matching is central to it. This

process has to check the cue against all stored memories and select the most appropriate one. e.g. the process that takes place when you see a familiar face and use the face as a cue to retrieve the person's name.

In connectionist models, the match is performed against all previously stored patterns in a single step - a **direct retrieval** of the appropriate pattern through **parallel processing**.

The connectionist model that can do this is known as a **pattern associator**.

Example - a three-element feature set - each element takes a 1 or 0 value:

Type of animal - Mammal = 0; Bird = 1

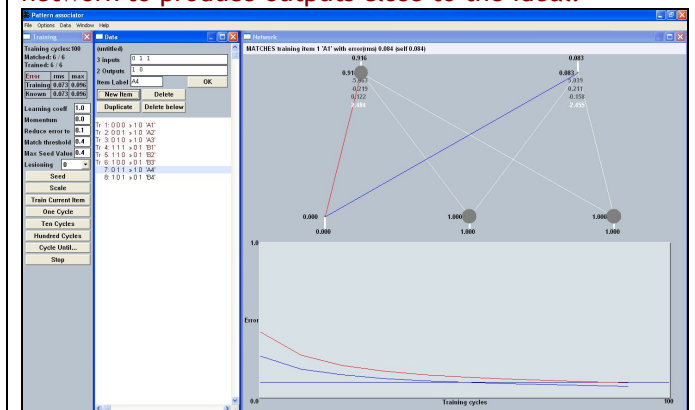
Ability to fly - Unable = 0; Able = 1

Northern/Southern hemisphere - North = 0; South = 1

Examples - Polar Bear = 0 0 0; King Penguin = 1 0 1.

These three digits (0/1) define the input.

Pattern associators often use neural nets as a classifier - e.g. the output will be a class into which each pattern should belong. The output is referred to as a 'teacher' - the ideal output is used to 'teach' a network to produce outputs close to the ideal.



Simple example - only the leftmost input node determines the output in this case - red and blue lines on the network diagram indicate the strongest connections between input and output nodes - in other words, they are the most strongly weighted connections.

Patterns A4 and B4 (resulting in matches to A1 and B1) show how a pattern the network has not been trained on can be matched and come to the right result.

### The auto associator

Pattern associators - used to form a link between one complete patterns and another (e.g. a visual one to an olfactory one - rose to its scent).

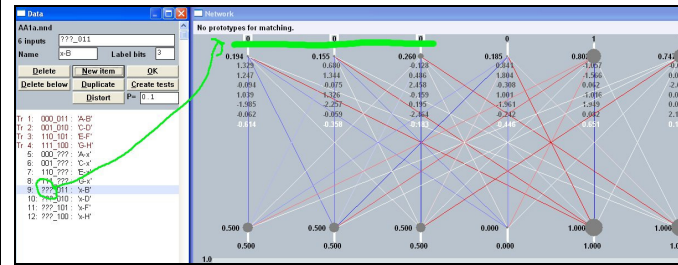
However, most common use is when you have a complete set of information to be held in memory - e.g. an event, which then has to be subsequently retrieved by a small part of that information - using a cue.

An important feature of human memory is that cues can be used to retrieve a complete set of information about something. The memory system is cued with part of the content of the memory trace. This ability is called content addressability. The connectionist model used to simulate this is the auto associator.

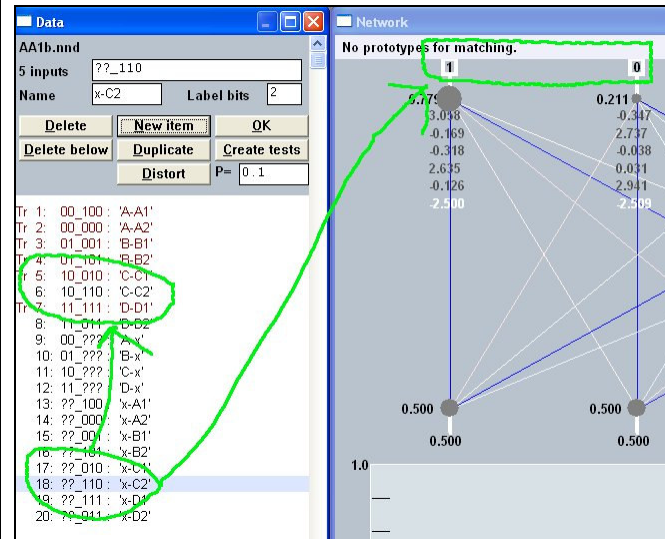
Example - each pattern is two letters together - e.g. A-B; C-D where each letter is represented by three elements e.g. A = 0 0 0; B = 0 1 1; C = 0 0 1; D = 0 1 0. So A-B = 0 0 0\_0 1 1; C-D 0 0 1\_0 1 0 etc.

Training the network enables it to answer questions of the form 000\_??? - to which the answer would be 001 - i.e. the cue A-x provides the answer A-B.

The network will answer equally well if the missing part of the pattern is on the rhs - indeed, any pattern of incompleteness should work - such as 0?0\_?1?



Auto-associator networks also do not need to be trained on patterns in order to reply with the 'correct' answer.



Above image shows C-C2 (10\_110) not trained, but matched on input ??\_110. Another example of spontaneous generalisation - it can generalise from six patterns to two more it has not seen.

### Learning by error reduction

Weights of the connections in the network change over time in response to training - so the error reduces with training. How quickly a network learns is determined by the learning coefficient. Larger coefficients often mean faster learning, but with the potential to have less stable learning within the networks if it is set too high.

The ability of connectionist networks to learn just from local operations is a desirable feature of a model that mimics the brain. In the brain, it is argued neurons learn by altering their connections based on activity in neighbouring neurons. As it seems unlikely that connections can be modified from activity occurring elsewhere in the brain, it implies the brain learns just by means of local operations too.

### What can connectionist models do for us?

#### Emergent properties of associative networks

1. Content addressability - an important emergent property of connectionist networks - equivalent to the human ability to retrieve information on the basis of partial cues. This is not directly 'programmed in', but is an inherent response of the network to partial cues. Behind much of the debate regarding the worth of connectionist models vs rule based models of cognition - **Stone**.

2. Multiple input patterns can be used for training - eventually, if all the patterns are presented, the weights of the network will converge on a set that will store all patterns perfectly (i.e. the error will drop to zero). One set of weights in a network can therefore store more than one pattern - similar in concept to the way that holograms can show more than one picture.

#### Example connectionist model of learning & memory

Prototype theory (**Braisby**) - a concept has a prototype - an idealised 'master' concept. All instances of the

concept (exemplars) are similar, but not identical - in other words, some features are different. There can therefore be good and poor exemplars of the prototype - e.g. a child learning about different types of cats and dogs.

**McClelland and Rumelhart** - connectionist model based on this to demonstrate the way in which a distributed model of memory would behave. Used a 24 unit auto-associator network.

Input patterns to represent exemplars are presented to the network, which learns using a procedure similar to the patter/auto-associator examples. After learning, network cued with pattern fragments - to see if it corresponds to the input pattern or an ideal prototype.

In some psychological models (e.g. **Rosch**) the prototype is a distinct item - something containing the 'most typical' properties of a concept. Exemplars are then compared to the prototype in order to determine category membership. The prototype is not a 'perfect' exemplar - but a collection of salient features. If enough are present in an exemplar, it is categorised as one of the concepts. Prototype theory therefore has a 'typical' set of features that represent the most 'typical dog'.

A connectionist network might therefore provide evidence of a mechanism that can produce a prototype from a series of exemplars.

**First demonstration** modelled a child seeing several different exemplars of a concept (such as a dog) and so learns the 'prototype' of the concept without ever seeing it.

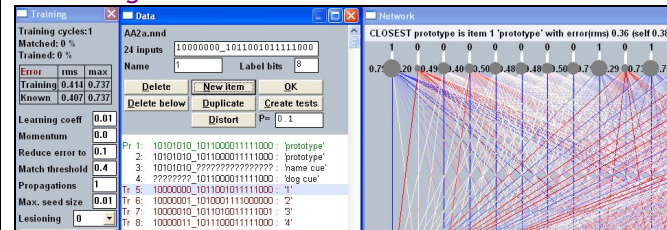
**McClelland and Rumelhart** - of the 24 units, the 1<sup>st</sup> 8 represent the name (e.g. Fido) and the next 16 represent dog features (e.g. waggy tail, barks ...)

As in the auto associator example, the actual features represented by the 24 input units are not

specified. In the **McClelland and Rumelhart** example, the 1<sup>st</sup> 8 units are uncorrelated (names) but the last 16 are similar to other exemplars - as in real dogs.

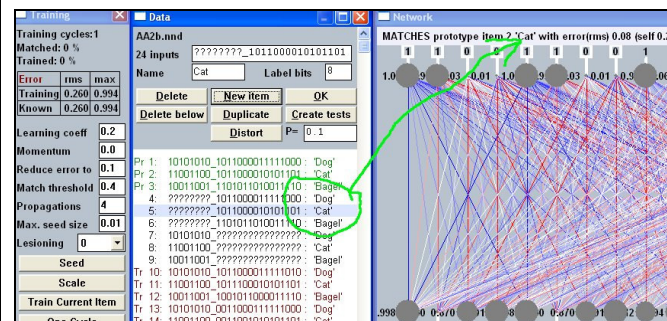
The prototype dog (not shown to the network) was defined arbitrarily as a pattern of the last 16 units - 1011000011111000.

After 1 cycle of training, the network responds to the prototype strongly for the prototype - and all of the 'seen' dogs!



**Second demonstration** - new prototypes created for the concepts 'cat' and bagel - using 16 units.

The cat (as a small mammal) shared some features with the dog ( $r=0.5$ ); bagel is not correlated. Prototypes also given names 'cat', 'dog', 'bagel' - a unique pattern in the first 8 units always associated with individual exemplars of that type.



Cueing the network with either the name or prototype part of the pattern produces the correct response (and as before, the network was not trained on these examples). Sometimes this requires a bit more tweaking

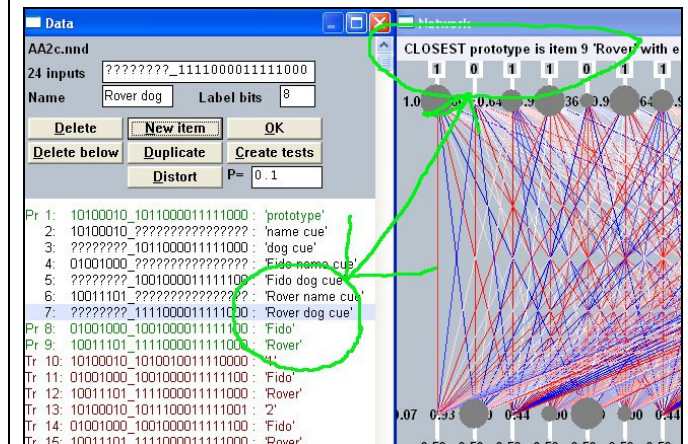
to work - e.g. 8 propagations instead of 4; cycle through the training examples a few times.

Particularly interesting result for the correlated 'cat' and 'dog' prototypes - shows a connectionist network can discriminate between correlated prototypes. The network derives its answer from the commonalities within the input patterns - a consequence of learning by error reduction.

**Third demonstration** - a model can learn simultaneously from a general prototype and specific exemplars. Scenario - two known dogs are called Fido and Rover. Other 'dogs' are also seen, but not named. The resulting network recognises dogs as dogs and Fido and Rover are also recognised correctly.

Input: Name pattern never changes; Fido differs from the 'dog' prototype in positions 11 and 22; Rover in position 10.

Fido and Rover are seen many times in the input (a person sees Rover and Fido many times); several other non-repeated distortions of the 'dog' prototype are the rest of the input set (many dogs are seen only once).



Small differences show the sensitivity of this distributed memory system - Fido differs by 2 units; Rover by 1 - yet both are retrieved rather than the

general 'dog' prototype when seen.

**McClelland and Rumelhart** have therefore provided evidence that a distributed memory system can learn from just a set of exemplars presented to it.

It can spontaneously generalise; learn and store many different patterns in the same network using a single set of connections - a differentiator of connectionist models over (say) rule based approaches. Interrogating the connectionist network produces one response - as if all stored patterns are processing in parallel.

It gives an insight into semantic/episodic memory distinction (**Rutherford**) - Specific information about particular dogs == episodic memory; the ability to generalise from individual exemplars to classify them all as dogs == semantic memory.

**McClelland and Rumelhart** therefore suggest that 'semantic memory may be just the residue of the superposition of episodic traces' - arguing for the position that there is no fundamental separation of the two in human memory.

### More powerful connectionist models: the simulation of human cognitive behaviour

The behaviour of the pattern matching, auto associator and prototype learning connectionist models is not directly comparable with the way humans behave - more powerful models are required to compare them with human performance on a specific task - e.g. % correct recall; reaction times ...

Example - presenting words (and non-words) to a neural network to simulate the way in which humans learn and then pronounce words.

Patterns of pronunciation are not linearly predictable (which is required by two-layer pattern associators and the other early examples). E.g. - the pattern for 'exclusive or' is not linearly predicable - you can't

get a two-layer network to learn:

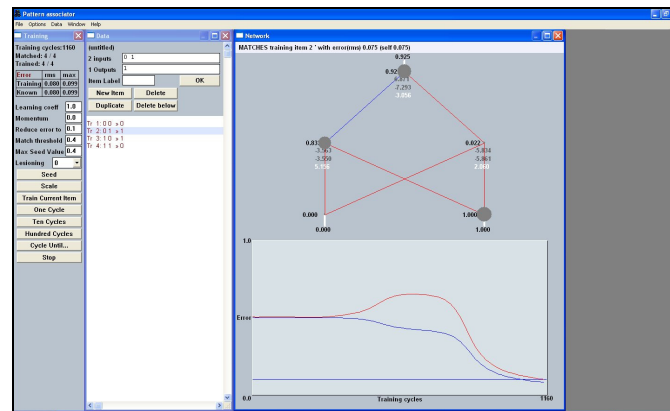
Input	Output
1 1	0
0 1	1
1 0	1
0 0	0

- each of the 'features' of the input are acting in combination, rather than independently of each other.

Another example is the letter string 'ave' in English - pronounced differently in have, gave, weaver - depends on the combination of other letters around it.

### Hidden layers

Allow this type of association to occur for patterns that are not linearly predictable - the configural problem, as discrimination between patterns relies on the total configuration of the inputs.



[Pattern associator with hidden layer for the XOR problem]

The calculation of new weights for the network 'propagates back' to reduce the error - the backward propagation of error (backprop) method of learning.

Preserves learning by local actions - modification of

weights is from information local to the connection ( - as in the human brain.)

### An example of a multi-level connectionist model

Seidenberg and McClelland - three layer network with hidden units to simulate the reading of English words.



Orthographic input units - used to take on words in the form of triples (to a scheme devised earlier by **Rumelhart and McClelland**) - e.g. make represented as \_ma, mak, ake, ke\_. Each input unit responds to a range of triples (10 possible first letters, 10 second, 10 third including the beginning and end markers).

Each individual input unit can therefore respond to 1000 different triples - but only a small number of the 400 input units are activated by each triple - around 20 in all - as they are set up to respond to different patterns of letters in the triples.

Feeds through the hidden units and on to the 460 output units. These have a similar representational scheme (based on **Wickelgren's** triple scheme from 1969). The encoded feature are **phonetic** - e.g. vowel, fricative, stop. Each unique phoneme triple corresponds to a unique pattern of excitation in around 16 of the 460 output units.

Importantly, input and output both use distributed representation over many units - a unique pattern of activation is produced by a given input word or a given output pronunciation.

The network was trained on 2,897 monosyllabic words, reflecting the **Kucera and Francis** word frequencies - e.g. "the" is 69,000 times more frequent than 'rake'. 150,000 learning trials used - most common words appear 230 times; least common 14.

The performance of the trained network was defined by its ability to produce correct phonology - as measured by the mean squared error.

Substantial error was found on low frequency exception words (seen less - similar to human learning). And, the network had far fewer errors on high frequency regular, exception and strange words.

Naming latency is known from **Waters and Seidenberg's** experiments on humans to be higher on strange and exception low frequency words than for regular words; with all three types of words showing similar (and much less) latency on high frequency examples.

Using the same words as for **Waters and Seidenberg's** experiment in the network, they found that the connectionist network was a good simulation of actual human behaviour.

Similar results when tested with Brown's 'unique' words (e.g. soap and curve - letter sequences not found in other monosyllabic words), which are less eccentric in pronunciation than 'strange' words.

Their main theoretical conclusion is not that their model is a complete one of the reading process; but that it is in contrast to some psychological theories of human reading (e.g. Ellis) - where regular words are pronounced directly by a rule-based system; and irregular words are looked up first in a mental lexicon which can then be used to look up an unambiguous pronunciation.

**Seidenberg and McClelland's** connectionist model does not require multiple routes to perform this task. Therefore, naming latency cannot be used to justify the multiple route stances alone. **Acquired dyslexia** (through brain damage) has been argued to provide the additional evidence require for a multiple route theory - but **Seidenberg and McClelland's** model can simulate this evidence too.

## Conclusion

Connectionist models are very different from rule-based systems (like ACT-R). Main advantage of both types of models is the requirement to be explicit about the computational architecture that is being used.

Neural network models stand or fall by the accuracy of their representation schemes. Later work by **Morris** using **Seidenberg and McClelland's** scheme showed problems and alternative schemes have been used - e.g. **Plaut et al.**

Comparison with human data is important - e.g. the effect of lesioning models.

Better connectionist models will result as improved understanding of brain structure and the way it handles neural connections is discovered.

=====

## Link to cognitive neuropsychology:

Connectionist (and other cognitive models) used to support research in neuropsychology as it is possible to lesion models to simulate brain damage and compare its behaviour to those of neuropsychological patients.

**Farah and McClelland** - connectionist model successfully created of some problems experienced by agnosia (object recognition) patients - e.g. those with more difficulty naming living than non-living things (a finding first documented by **Warrington & Shallice**)

... and **Coltheart** - recreation of a model of reading errors found in different types of dyslexia.

## Link to recognition:

IAC connectionist model of face recognition - **Burton et al; Burton & Bruce**. Strength of the model is that it can account for findings from both lab studies and the type of everyday mis-recognition errors categorised in **Young's** diary study

## Link to language processing:

**McClelland and Elman** - TRACE - lexical competition model of word recognition, using continuously varying levels of activation within the network reflecting the probability of the next word requiring recognition from the speech stream.