Neural Networks in Cognitive Science

Jeff Yoshimi

August 28, 2011

Chapter 1: Introduction

The phrase "neural network" has several meanings. A biological neural network is an actual set of interconnected neurons in an animal brain. Figure 1 shows a biological neural network. However, "neural network" as we will use the phrase means "artificial neural network," that is, a computer model that has certain things in common with biological neural networks. In particular, artificial neural networks contain simple processing units or "nodes", that are similar to neurons in the brain, and are connected by "weights", which are similar to synaptic connections in the brain. Neural networks can be made to do many fascinating things, and have become a core tool in contemporary cognitive science. In this chapter I give a general introduction to neural networks, and survey some of the different ways they are used. After considering uses of neural networks in biology and engineering, I focus on cognitive science. The course as a whole is on neural networks in cognitive science.[1]

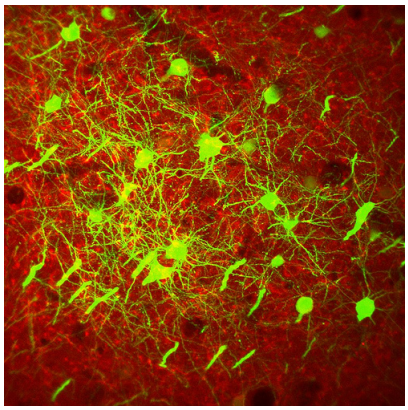

Figure 1: A biological neural network (Mark Miller, Nelson Lab, Brandeis University. Licensed Under: CC BY-ND)

[1]Though much of the main work we will do involves basic tools and concepts that are common to all applications of neural networks.

Note that there are a few different ways of referring to artificial neural networks, including: "connectionist network," "PDP (parallel distributed processing) network," or simply "neural network." These phrases differ slightly in their meaning, but will be treated as synonyms in what follows.

1 Overview and Simple Example

In figure 2 some simple neural networks are shown. Both networks were built using a software package called "Simbrain", which is used in demonstrations throughout these chapters. The circles with numbers in them represent artificial neurons, and the lines with filled disks at the end represent artificial synaptic connections between neurons (there is no completely standard way of visually representing neural networks; for example, compare figures 9 and 10). The neural network on the left is a "feed-forward" neural network, which is a layered network, where each layer is a group of nodes that is totally connected to the next layer in a sequence. Activity in this kind of network flows from an "input layer" through one or more "hidden layers", and then to an "output layer". The network on the right is a "recurrent" network, where the connections don't just go forward, but interconnect in such a way that activity can flow in repeating cycles. Recurrent networks display complex dynamical behaviors that don't occur in feed-forward networks.

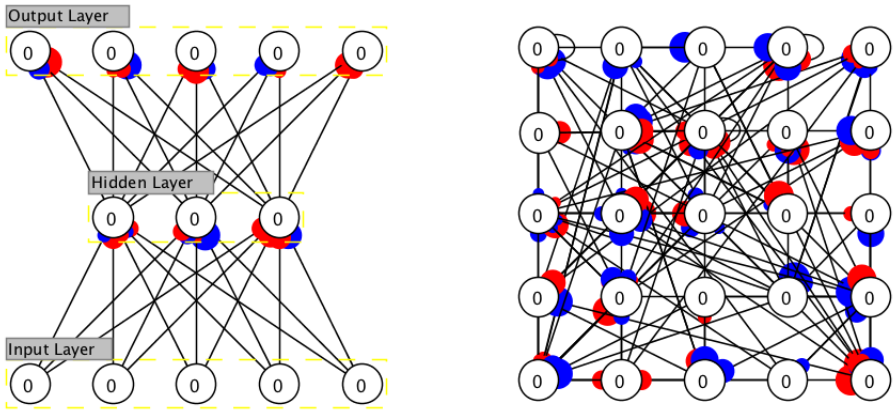

Figure 2: Some common types of neural network: a feed-forward network (left), and a recurrent network (right).

To get a feel for how networks work, let's try an example (you can run this example in Simbrain and follow along; the simulation is in *simulations > threeObjectsDist.zip*). A screenshot of this simulation is shown in figure 3. The simulation has a neural network with three nodes in the input layer, five nodes in the hidden layer, and three nodes in the output layer. The mouse in the

simulated world on the right is hooked up to this network. When the mouse is moved around, the activation in the input nodes changes. This simulates the way odor molecules impact the inner lining of the nose, causing sensory neurons to fire at different levels. So the input layer is a kind of simulated nose. The job of this network is to distinguish the three objects on the basis of those sensory inputs. Depending on which object the mouse is near a different output node should be activated.

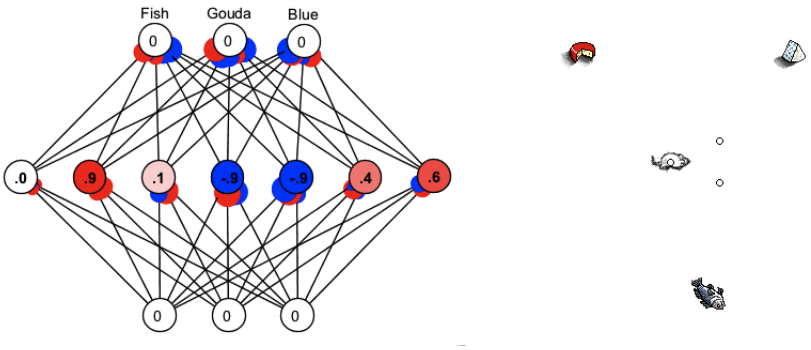

Figure 3: Simple feed-forward network that recognizes three objects.

This may not seem like much, and in fact it is a pretty simple network. However, it illustrates several basic principles of neural networks that will be important below.

First, the network is brain-like, but is not an actual simulation of a brain. It takes a pattern of inputs, and transforms those inputs through a network of connections. It learns by slowly adjusting those connections. This is similar to the way information processing occurs in the brain. But it is not a realistic simulation of a brain circuit.

Second, nobody "programmed" the network with rules, the way you would program a typical computer.[2] The way this network works is that we train it. We tell the network what we want it to do, and it learns to do it. Roughly, we put the mouse near the fish, and say, "Ok, when you smell something like this, fire your first node," and similarly with the Gouda and blue cheese. It then adjusts its weights to produce the correct input-output mapping. At first it will make mistakes but over time it well get better and better, something like the way humans gradually get better at doing things with training.

Third, the network does well even in the face of various kinds of noise and damage. It is not "brittle" in the way a digital computer is. You can start deleting synapses and the network still works ok (try it!). In a similar way,

[2]Of course,you are most likely running the neural network on a traditional computer. But that's just a matter of convenience. What is important here is the formal structure of the neural network, not the nature of the computer "implementing" the neural network program. Also note that silicon neural networks are available. In fact a new one was recently released from IBM!

humans can lose a few neurons and function well. A digital computer, by contrast, could completely stop working if it lost a single transistor. Of course, if a network loses enough connections its performance will suffer, but the change in performance happens in a gradual way. This is called "graceful degradation."

2 Types of Neural Network Research

In practice, neural networks are used in two main ways: (1) as engineering tools, to solve problems and build useful things, like machines at the post office which recognize hand-written zip codes on envelopes, and (2) as scientific tools, to understand how the mind and brain work. To see the difference between (1) and (2), consider the status of errors. Errors are bad for engineering. A good calculator does not make mistakes! But they are useful for cognitive science. A neural network model of adding and multiplying would be better if it made the *same kinds of errors* as humans do.[3]

Scientific uses of neural networks can be further divided depending on whether the neural network is supposed to model biological or behavioral data (see figure 4).

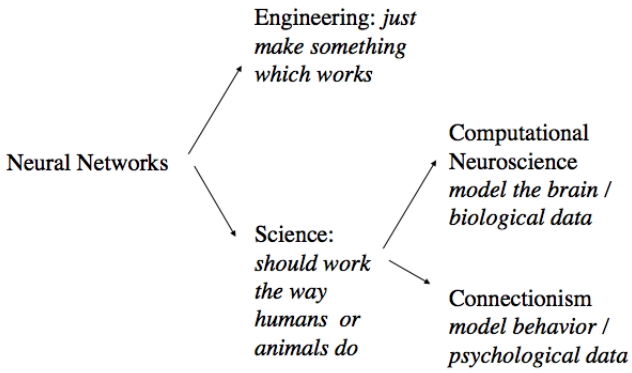

Figure 4: Tree of positions and uses of neural networks.

[3]Sometimes determining what kind of neural network model is being used is tricky. For example, consider the following title of a journal article: "Use of Neural Networks in Brain SPECT to Diagnose Alzheimer's Disease" (Page, et. al. 1995). At first, this sounds like it might be a computational neuroscience or connectionist article, since it mentions the brain and Alzheimer's. However, the article is actually about how neural networks can be used to determine whether a person has Alzheimer's. The neural network is not being used as a model of the brain or any cognitive processes, but rather as an engineering tool to help diagnose Alzheimer's based on brain images.

2.1 Neural Networks as Engineering Tools

Neural networks in engineering are tools which solve problems. It turns out that neural networks are good at doing certain kinds of things, like finding patterns in noisy data, controlling certain kinds of machines, and in general dealing with problems that involve numerous variables that are hard to deal with explicitly. As two workers in the field put it:

> The most natural application areas for [neural networks] are obviously tasks in which appropriate transformations from certain inputs to certain outputs should be established, but the transformations cannot be discovered analytically due to a variety of reasons. Therefore it is no wonder that the most successful applications of the [neural networks] can be found in the areas of machine vision, pattern recognition, motor control, signal processing, etc., where such input to output transformations dominate the problem solving (Heikkonen and Lampinen).

Engineers who use neural networks for these purposes dont care about how the brain works or how humans think: they want to make a machine that works. A neural-network based rice cooker better make good rice, whether it cooks like humans do or not.

Consider an example. The lumber industry has a need to classify pieces of wood into different grades based on what type of defects they contain, according to some pre-set standard. The Finnish standards involve over 30 categories of wood defect, which are in turn based on different "knot classes." Some examples are shown in figure 5.

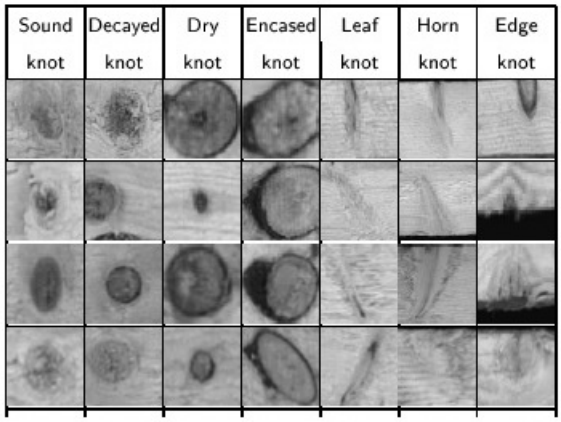

Figure 5: The kind of data a knot classification network has to deal with. From: Heikkonen, J., Lampinen, J. Building Industrial Applications with Neural Networks Licensed Under CC BY-NC

A human can classify these knots reasonably well, but it is time-consuming, error-prone (look at how subtle some of the differences are in the knots above), and expensive. The lumber industry would like a way to automate the process.

So, how can we automatically classify knots? It's hard to classify them according to any kind of explicit rule. But a neural network can be used to solve the problem reasonably well, in fact better than human graders. A neural network has about 90% accuracy in this process, compared with 70-80% accuracy for humans. How does it achieve this? It is shown a set of examples, and told how they should be classified. Its connections are then incrementally updated to reduce the number of mistakes it makes. The network picks up on statistical regularities in the data it's presented with, and ends up doing a pretty good job. Figure 6 shows a flowchart of the system the engineers used to solve the problem. The raw image on the left is input, and a classification of the knot (on the right) is the output.

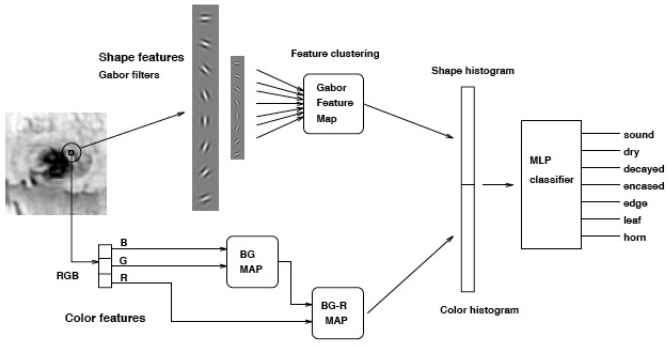

Figure 6: An industrial neural network, which recognizes different kinds of knots in lumber. From: Heikkonen, J., Lampinen, J. Building Industrial Applications with Neural Networks Licensed Under CC BY-NC

Notice that the neural network is buried inside the system. It is the "MLP classifier" towards the right ("MLP" means "multi-layer-perceptron," and is similar to the simple feed-forward example network above). This system takes a picture of a piece of wood, does some "pre-processing" on the resulting pixel image, and then summarizes features and colors of that image as a list of numbers, a "histogram" or "vector" (we'll talk more about vectors later). This list of numbers is then fed to the neural network as input. The neural network transforms these numbers into another list of numbers which describe how decayed, burnt, dry, round, and so forth each sample is–a "feature vector". This feature vector can then be used to classify the knot.

2.2 Neural Networks as Scientific Models

In science, neural networks are used as a modeling tool. As with other scientific models, neural networks are valuable because they they provide a simulation

of phenomena that are difficult to directly measure or observe. It is hard to observe neurons or synapses in living organisms, but easy to create and observe simulated neurons on a computer. Compare the use of computer models in studying the weather: it's hard to measure certain aspects of the weather, but relatively easy to program a simulation of the weather. So computer simulations are a useful tool for meteorologists. Similarly here.

Neural networks are used to model real neural networks in the brain (computational neuroscience), and human and animal behavior (connectionism). In both cases the models are constrained to be consistent with observed features of a system of interest. Neural network models in biology must behave in the same way real neurons do when they are measured. Neural network models of behavior must produce the same kinds of behaviors humans and animals do. In both cases the model should also shed new light on relevant processes (they should have "explanatory power"), helping us to understand how those processes work, and in some cases generating new, testable predictions.

2.2.1 Computational neuroscience

The use of neural networks to model biological data is associated with "computational neuroscience." The goal of computational neuroscience is to understand how the brain works, with the help of computer models. Neural network models in computational neuroscience are typically designed to reproduce electrophysiological data, that is, recordings of real neurons either in a laboratory dish ("in vitro") or in a living animal ("in vivo"). Not all models in computational neuroscience are network models: many focus on single neurons (or even on individual parts of neurons), which can be quite complex (see figure 7). When compared with more classical neural networks (as in the figures above), the model neurons used in computational neuroscience tend to be more complex, and are designed to mimic the electrochemical properties of real nerve cells.

2.2.2 Connectionism

The use of neural networks to model psychological data is sometimes called "connectionism".[4] The general idea with a connectionist model is to reproduce some aspect of human or animal behavior. For example, a connectionist model of a 4-year old child reading aloud (converting written symbols to appropriate noises) should be able to perform about as well as a 4-year old child: it should find the same things difficult an average child that age does, it should make the same mistakes a child that age does, etc. Of course, what it means for a task to be "difficult" for a neural network is not obvious, and must be specified. Such models are usually meant to *suggest* how a given task is accomplished by the brain, but they are not usually intended as a direct models of the underlying neuroscience. In what follows, we will focus on connectionist models.

[4]Not everyone using the term "connectionism" in this way, but it is a fairly standard usage. A more precise phrase would be "connectionist model of a cognitive process".

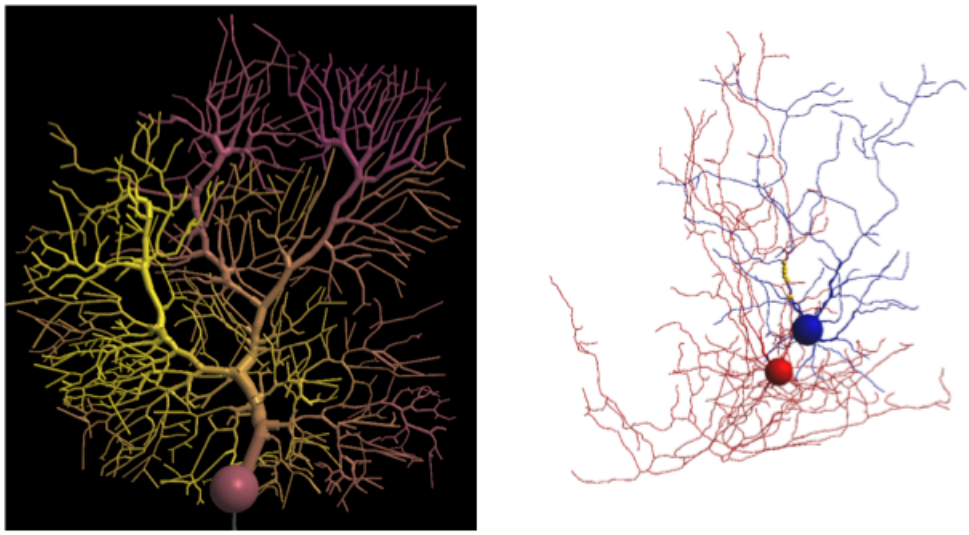

Figure 7: Some computational neuroscience models: a simulated Purkinge cell from the cerebellum (L); a simulated pyramidal cell from the cerebral cortex (R).

To get a better sense of what's involved in connectionist modeling, suppose someone gives you a journal article and asks you to determine: (1) what the structure of the neural network in it is, and (2) what kind of behavioral data it models. If the paper describes a connectionist model, there should be answers to these questions, though the answers are not always obvious. The first question is about the structure or "architecture" of the network. Is it feed-forward or recurrent? What kind of nodes and weights are used? How was it trained? (These questions may not make much sense now, but later, once we've acquired the relevant concepts, they will). The second question is about what kind of behavioral data the network captures. Sometimes this is obvious: for example, a network that can play chess like a human is modeling the human ability to play chess. In other cases it is more subtle. For example, some networks are supposed to model how long it takes subjects to perform various tasks (e.g., it takes longer to say some types of words than others). In humans we can measure this as reaction time in milliseconds. But what does this correspond to in a neural network? Sometimes it is number of "processing cycles". Other times it is even less direct, as we'll see. There are many other kinds of data that a neural network can capture. For example, some networks are "lesioned" in a way that is supposed to mimic certain kinds of brain damage in humans, and they are then tested to see if their performance is changed in the same way

human performance is in similar conditions.

Let us consider two examples that show how these questions would be answered, for two famous papers in the history of connectionism.

A first example is a model of reading aloud due to Seidenberg and McLelland (1989), of looking at words on a page and converting them in to speech. With respect to question (1), regarding architecture, the neural network is a variant on a feed-forward network, similar to the network on the left side of figure 2. It has a lot more nodes: 400 input units, which represent "orthographic" features of words (the way words are written) and 460 output units, which represent "phonological" features of words (the way words sound). It was trained to pronounce all one-syllable words in English using a method called "backpropogation" (more on what that is later).

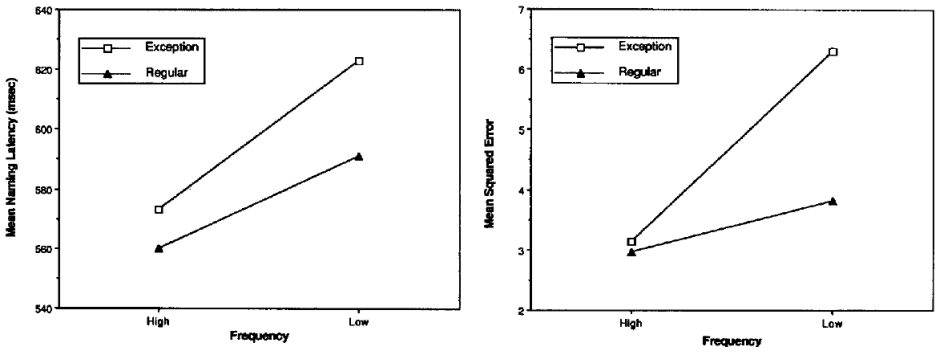

Figure 8: Data associated with Seidenberg and McLelland (1989)'s reading model. Human data are on the left, neural network data are on the right.

With respect to question (2), the model reproduced a number of known linguistic phenomena, including the "word frequency" effect (common words are pronounced more quickly than uncommon words), and the "frequency-regularity interaction", whereby words that have regular pronunciation patterns (e.g. GAVE or MUST), are pronounced faster than exception words (e.g. HAVE or PINT). However, this regularity effect only occurs for infrequent words. Human data showing these effects are on the left side of figure 8. Note that the low frequency words take longer to pronounce than the high frequency words (regularity effect) and that irregular words also take longer to pronounce, for infrequent words. The neural network data is on the right. The data was generated by showing the network many different words in these various categories (regular, irregular, frequent, infrequent), and then counting how many mistakes the network made in these various categories. They then assumed that this error score for the neural network corresponded to reaction time. When you line the two graphs up next to each other, they look the same. As you can see, the neural network made the most errors for irregular, infrequent words, just like humans take the longest to pronounce those words. So this is taken to be

evidence in favor of the model. Many questions come up here, and you might not be satisfied (in fact, this model was involved in a kind of war between connectionist and non-connectionist models of reading). But the main thing I want to emphasize here is that the model is supposed to capture some kind of human, behavioral data. It is a bit convoluted (since we are supposed to assume that human reaction times correspond to mean network errors), but nonetheless this illustrates how connectionist networks can be used to model behavioral data.

A second example is an "IAC" or "Interactive Activation and Competition" network. These networks were among the first used to demonstrate what neural networks could do after the resurgence of interest in neural networks that began in the 1980's (more in chapter 2). Let's look at a famous example of an IAC network, the Jets and Sharks model (McClelland, 1981), which was used to model certain features of human memory, e.g. an ability to recall who a person is that has certain traits, and an ability to guess what properties a person has based on his or her other properties. As an example (which was outdated even then!), McClelland created a network with knowledge of two fictional 1950's gangs, the Jets and Sharks from *West Side Story*.

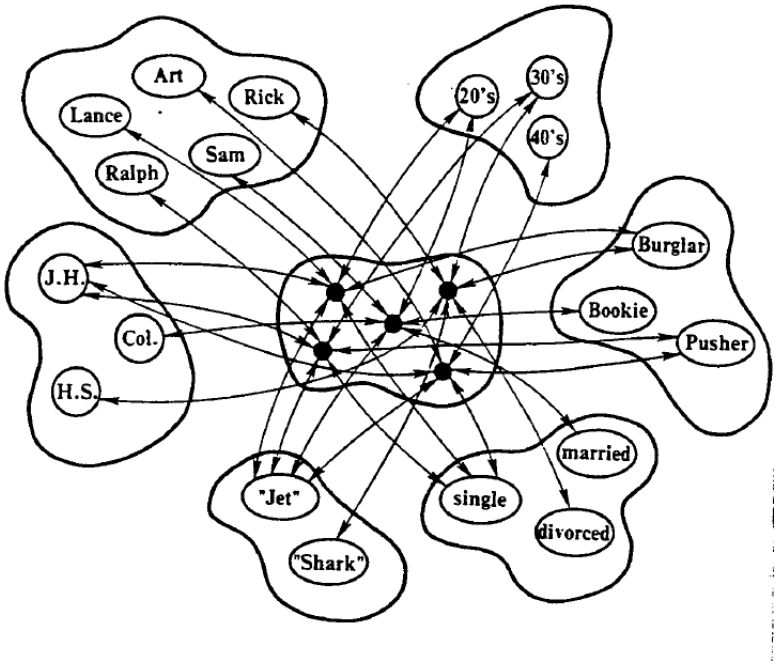

Figure 9: A fragment of the Jets and Sharks model.

In terms of question (1), architecture, an IAC model contains a set of "pools" of nodes. In figure 9, these pools are shown as groups of nodes surrounded by squiggly lines: there are 7 pools of nodes, including the central pool in which nodes are shown as filled black circles without labels. The neurons in each

pool are connected to each other with "inhibitory" connections, and the result is that each pool has a "winner take all" structure. The idea is that after a while, only one node in each pool will be active above 0. (We will study this type of network in more detail later). Each pool represents one trait of a gang member: name, age, education level, marital status, job in the gang, etc. To encode knowledge about a gang member, connections are made between a person, his name (these were all male gang members), his job, etc, by way of the "instance nodes" in the center. I don't expect you to fully understand this network, since I'm only giving a sketch here. But the overall idea is that the connections correspond to "associations" between ideas. When one node / idea is activated and the network is run, all the associated nodes / ideas are activated, and inconsistent nodes / ideas are suppressed. This is also called a "spreading activation" network. Over time all the ideas associated with the original idea should be activated.

In terms of question (2), this model captures a simpler form of data than the reading network. Rather than modeling behavioral data gathered from experiments, the network just needs to be able to do certain general kinds of things humans do. Here are a few examples of what this network can do. First, if the Jets node is activated and the network is run, the standard characteristics of the Jets will light up: they tend to be in their 20's, with a junior high school education, and single. This is like asking the network "what do the Jets tend to be like?" Second, the network can find an individual based on specific properties. For example, if the 20's node and the junior high education node are activated, then the name nodes for Lance, Jim, John, and George all light up. This is like asking "Who is in their 20's with a junior high education?" and being told "Well, that could be Lance, Jim, John, or George." Third, the network can tell you the properties of an individual. If the Lance node is activated, the Jets node, 20's node, Junior high education node, and burglar node are active. This is like asking "Tell me about Lance?" and being told about him.

2.2.3 Hybrid Approaches

Connectionism and computational neuroscience are really two ends of a continuum. Though models in computational neuroscience tend to ignore animal behavior (focusing instead on single neurons or small groups of neurons), some do attempt to capture behavior. Even though connectionist models are just "biologically inspired," some connectionist models do try to produce behaviors in a more neurally realistic way. For example, it is not uncommon for a connectionist researcher to say something like "this part of the network models executive function in the frontal lobes" or "this corresponds to motion processing in area MT of the extrastriate temporal cortex."

Consistently with this, many recent neural network models are "in the middle" between computational neuroscience and connectionism. One prominent example is the whole field of Computational Cognitive Neuroscience (CCN), which has flourished in recent years. These models capture various aspects of cognition (e.g. visual attention, semantic and episodic memory, priming, fa-

miliarity, and cognitive control), using groups of neurons that are explicitly associated with specific of the brain. An example is shown in figure 10. Many researchers hope that over time computer models of brain and behavior will converge, and that future models will increasingly capture both neural and behavioral data, and thereby reveal how the dynamics of the brain give rise to the dynamics of cognition.

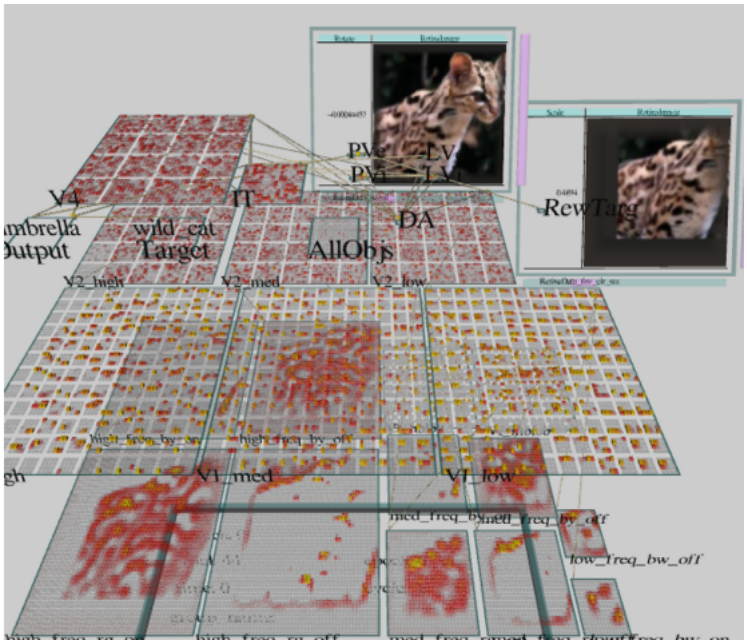

Figure 10: A screenshot from an Emergent simulation of visual processing, with labels indicating which brain areas each group of neurons represents.

3 Links

http://grey.colorado.edu/CompCogNeuro/index.php/CCNBook/Main

http://plato.stanford.edu/entries/cognitive-science/

http://plato.stanford.edu/entries/connectionism/

http://www.nici.kun.nl/Publications/1998/11339.html (Siena)

References

[1] Heikkonen, Jukka, and Lampinen, Jouko. 1999. 'Building Industrial Applications with Neural Networks.'

[2] McClelland, J. 1981. 'Retrieving General and Specific Information from Stored Knowledge of Specifics.' *Proceedings of the third annual conference of the Cognitive Science Society.*.

[3] McLeod, P., Plunkett, K. and Rolls, E. 1998. *Introduction to Connectionist Modelling of Cognitive Processes.* Oxford: Oxford University Press.

[4] Michael P.A. Page, Robert J. Howard, John T. O'Brien, Muriel S. Buxton-Thomas and Alan D. Pickering 1995. 'Use of Neural Networks in Brain SPECT to Diagnose Alzheimer's Disease'. *Journal of Nuclear Medicine* 37:2, pp. 195-200.

[5] Seidenberg, M and McClelland, J. 1989. 'A Distributed, Developmental, Model of Word Recognition and Naming.' *Psychological Review.*