Review of Machine Learning (see info earlier in course about Watson).

Humans create Machine Learning AI systems when humans themselves don’t know the rules about how humans use to solve a real world problem. For example, humans don’t know rules that can be used to tell in which of previously unseen digitized photographs of a person the person is wearing (or is not wearing) sunglasses. To an artificial system, a digitized photo is just a rectangle containing many rows and columns of decimal numbers (one decimal number per pixel). For another example, humans don’t know a rule that would say what steering wheel position should be used based on each of 960 different numbers coming from the corresponding pixels in a video image, where the video camera is aiming at a road ahead of a vehicle. Humans have developed several strategies (ANNs are one such) to program Machine Learning systems that learn on their own from lots of examples. Although humans can write a program (that humans can comprehend) to enable a Machine Learning system to be trained (from many examples) and then used, the resulting trained AI system itself is usually NOT comprehensible at all to humans, since the resulting system is typically just a huge bunch of decimal numbers none of which seems to have any meaning in and of itself. (Likewise a human brain that is thinking consists, among other things, of huge bunches of neurons that have certain charges on them, together with the connections among neurons.)

Review of ANNs

Often an ANN (Artificial Neural Network) is simply called a “neural network” by Machine learning system developers, since the “artificial” is understood to be the case by the users.

The two main parts of an ANN are:
- nodes (in a diagram, a node is drawn as a small circle)
- arcs (in a diagram, an arc is drawn as an arrow)

Each arc goes from one particular node to another particular node. A node is analogous to a biological neuron. An arc is analogous to a biological axon and one of its dendrites. A node can have many arcs aiming at it, and many arcs leaving from it. Some practical ANNs have hundreds of nodes and thousands of arcs. (The human brain has about 50 billion neurons and about a thousand connections per neuron.)

There are three kinds of nodes:
- input nodes, often visualized on the left side of an ANN diagram
- hidden nodes, can all be in one ‘hidden layer’ or in several such layers, often shown on the extreme right
- output nodes, often shown on the extreme right

Each arc into a node contains a decimal number and is viewed as part of the input to the node. The input to the entire ANN (the “overall input to the ANN”) is the combined input to each of the input nodes in the ANN.

Each arc out of a node contains a decimal number and is viewed as the output of that node. The output of the entire ANN (the “overall output of the ANN”) is the combined output of all of the output nodes in the ANN.

Recall that a biological neuron sums up electrical charges arriving on its dendrites and “fires” an electrical charge toward adjacent neurons, when the charge exceeds a threshold. Some very simple ANNs simulate this process by letting each node sum up decimal numbers that arrive on the arcs pointing to the node and if that sum exceed a certain decimal number (called the threshold of the node), the node outputs the number 1 on all arcs leaving from the node, otherwise the node outputs the number 0 on all arcs leaving from the node. A node in a more sophisticated ANN transforms the sum of the decimal numbers that arrive on the arcs pointing to the node into an output (from that node) that is a decimal number strictly between 0 and 1.

Once properly trained to adjust the weights appropriately to give the desired overall output for each of numerous “training” examples of overall input, when the ANN is then given an overall input of any example it hasn’t seen before, the ANN transforms that overall input into an overall output.

Two applications of successful ANNs, developed in 1970’s and 1980’s.

- The ability to speak printed words. (my grandmother’s house)
- This is called “NERTalk”, which is demonstrated in “The Thinking Machine” video.

A successful ANN had 203 input nodes, 80 hidden nodes, 26 output nodes. Also with diagrams showing inputs to outputs going from left to right. The diagram in that video showed inputs to outputs going from the top to the diagram to the bottom.

Driving on roads in the Pittsburgh area. (The vehicle was called “AVINN”.)

A successful ANN had 960 input nodes, 4 hidden nodes, 30 output nodes. The 960 input nodes received the pixel values from essentially a 30 by 32 video camera that aimed at the scene in front of the vehicle. The 30 output nodes indicated 30 possible directions (from far left to right) that the steering wheel could be turned.

Today there are even more impressive robot cars, especially a robot car developed by Google that has traveled hundreds of miles on California roads, navigating through the usual daily traffic in cities and on expressways, including at night.

ANNs continue to be considered as very practical Machine Learning systems, such as for recognizing handwritten digits, letters of the alphabet, and photographs. For instance:

- When given 10 million digital images from YouTube videos with no explanation of what in the images might be ‘interesting’, an ANN developed in Summer 2012 by Stanford’s Andrew Ng and by scientists at Google discovered its own concept of a cat’s face. (This is an example of Machine Learning called ‘unsupervised learning’, since the images were not labeled with a category indicating what the images were about.)
- The Ng team developed the ANN using a huge amount of processing, involving 16,000 separate CPUs. ANNs that learn patterns in the speech of individual people are also being used by Google. According to Google Research in August 2012: "... when you speak to your Android phone, chances are, you are talking to a neural network trained to recognize your speech. ...

Using neural networks for speech recognition is nothing new ... after what can only be described as a 20-year dry-spell (from the 1980s until around the year 2000) evidence that the technology could scale to modern computing resources has recently begun to emerge. ...

What changed?
Access to larger and larger databases of speech, advances in computing power ... fast distributed computing clusters such as the Google Compute Engine ... and a better understanding of how to scale the algorithms to make them effective learners.

An ANN developed in Fall 2012 by a University of Toronto team lead by Professor Geoffrey Hinton won an international competition for classifying photographs. The competition was in a kind of Machine Learning called ‘supervised learning’ since the competitors were given over a million examples of such digitized photographs, with each photograph already correctly labeled with the appropriate ‘one of 1000 classifications’.

There were at least 500 different examples of photographs illustrating each of the classifications. The winning Machine Learning system had to have the lowest error in predicting the classification of photographs the MS system had not previously seen. The Toronto team’s ANN had more than 6 layers of nodes, about 650,000 nodes and about 60 million arcs. The Hinton team used one powerful computer to train the ANN, and one week of computer time to train the ANN using the given photographs.

(See other side.)
Three different ANNs appear above (slide courtesy of Stanford's Andrew Ng). They illustrate design choices when researchers decide on how many nodes an ANN will have, and on how many of those nodes are input nodes (shown on the left of each of the three ANNs) and how many are output nodes (shown on the right of each of the three ANNs) and how many are hidden nodes, and also, how those hidden nodes will be arranged in one, two, or three hidden layers. (More than three hidden layers is also possible.) It's difficult to notice the arrow heads on the arcs, because so many arcs are aiming at any particular node, but the arrow heads are there. The numbers, such as 3-5-4 under an ANN indicate how many nodes are in the layer above that number.

Thus, you can think of each of the three diagrams as a diagram of a single, very simple, kind of artificial brain.

--- Features
The number of input "features" is 10,000 if the input is a digitized photo that has 100 rows and 100 columns of pixels (notice that 100 times 100 is 10,000).

The number of input "features" is 960, if the input is a single-frame of a digitized video image of 960 pixels of a scene in front of a vehicle (960 is what you get by multiplying 30 rows and 32 columns of decimal numbers, with each decimal representing a single pixel).

--- Classes
The number of output "classes" for an application determining whether a person in a photo is or is not wearing sunglasses is 2, so there would be 2 output nodes of the overall ANN for such an application.

The number of output "classes" for an application determining which of 30 positions on a steering wheel is the correct position (for the corresponding overall input into the ANN, coming from a video camera) is 30. Thus there would be 30 output nodes of the overall ANN for such an application.

--- Not all Machine Learning systems are ANNs

Here are recent examples of Machine Learning systems that weren't ANNs:
- The Netflix contest winner, which was a hundred different ML systems working together as a single system.
- Watson, which was a combination of about a thousand different ML systems.
- Narwhal, used by the Obama presidential campaign.
- Orca, used by the Romney presidential campaign.

The Netflix contest was a one million dollar, international contest, to develop an ML system that was 10% better than the existing Netflix system for recommending movies to customers.

Narwhal is the name of a huge AI system used by the 2012 Obama presidential campaign, to learn a great deal of publicly-available info about Obama supporters and best how to motivate them to help Obama win.

Orca is the name of the AI system used by the 2012 Romney presidential campaign. Orca crashed badly on election day because it could not handle the vast amount of data that needed to be processed.