Training an Artificial Neural Network to Pilot a Sailboat

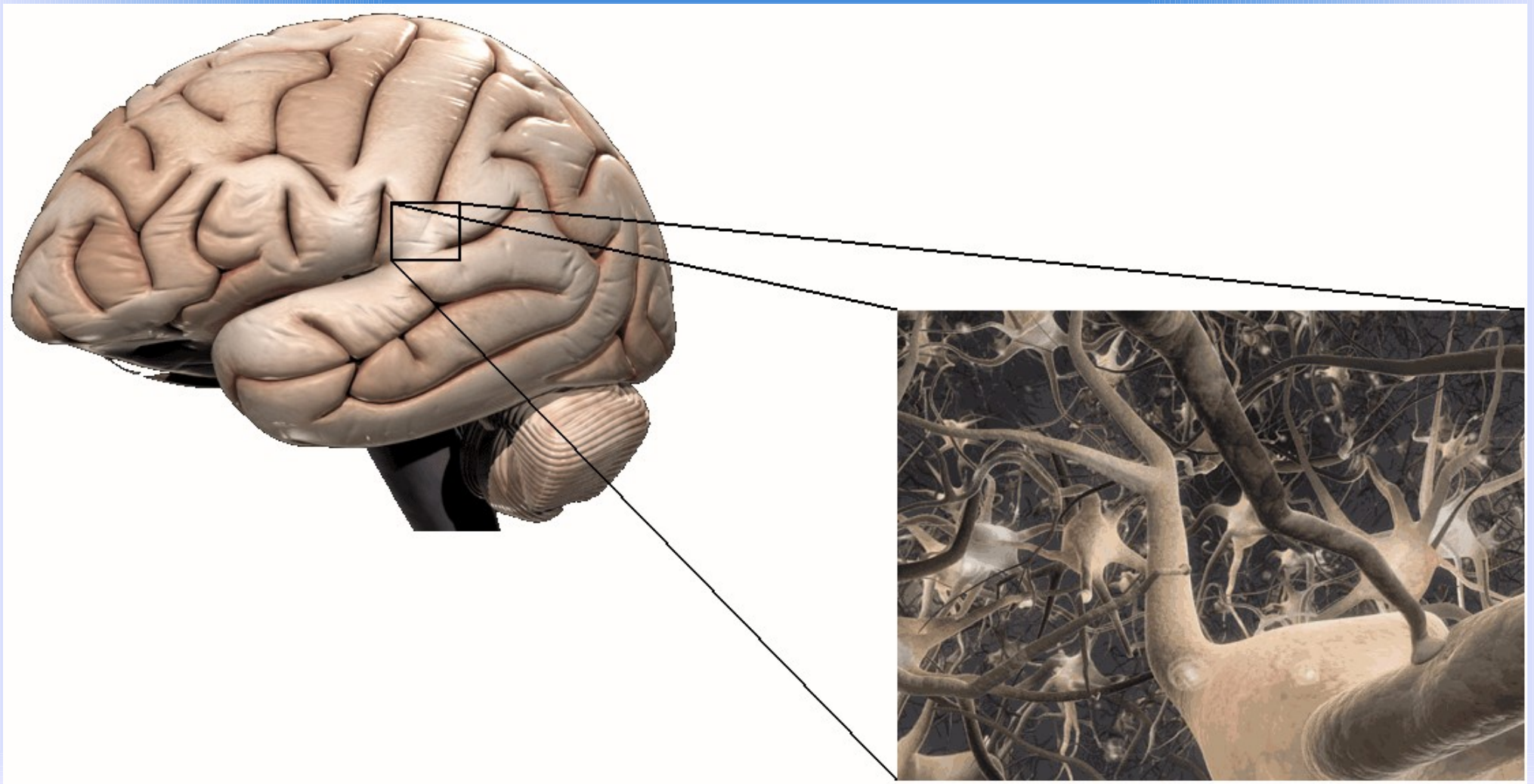
Motivation

The development of the ALVINN (Autonomous Land Vehicle In a Neural Network) system

What is an Artificial Neural Network?

- A network of simple processing elements, that exhibit complex global behavior, determined by the connections between the processing elements and element parameters

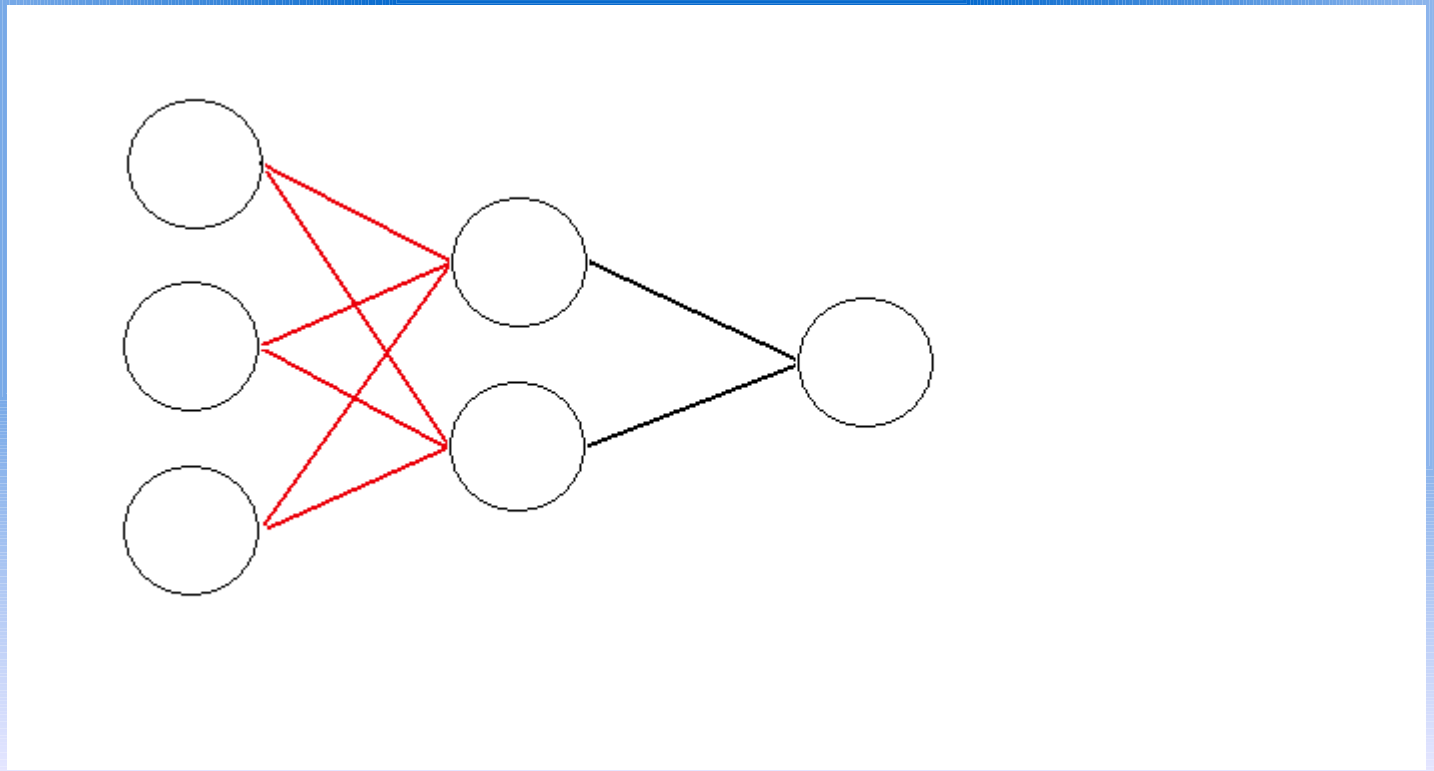
- Tries to simulate certain functions of the human brain, specifically, learning



Back-Propagation

Summary of the back-propagation technique:

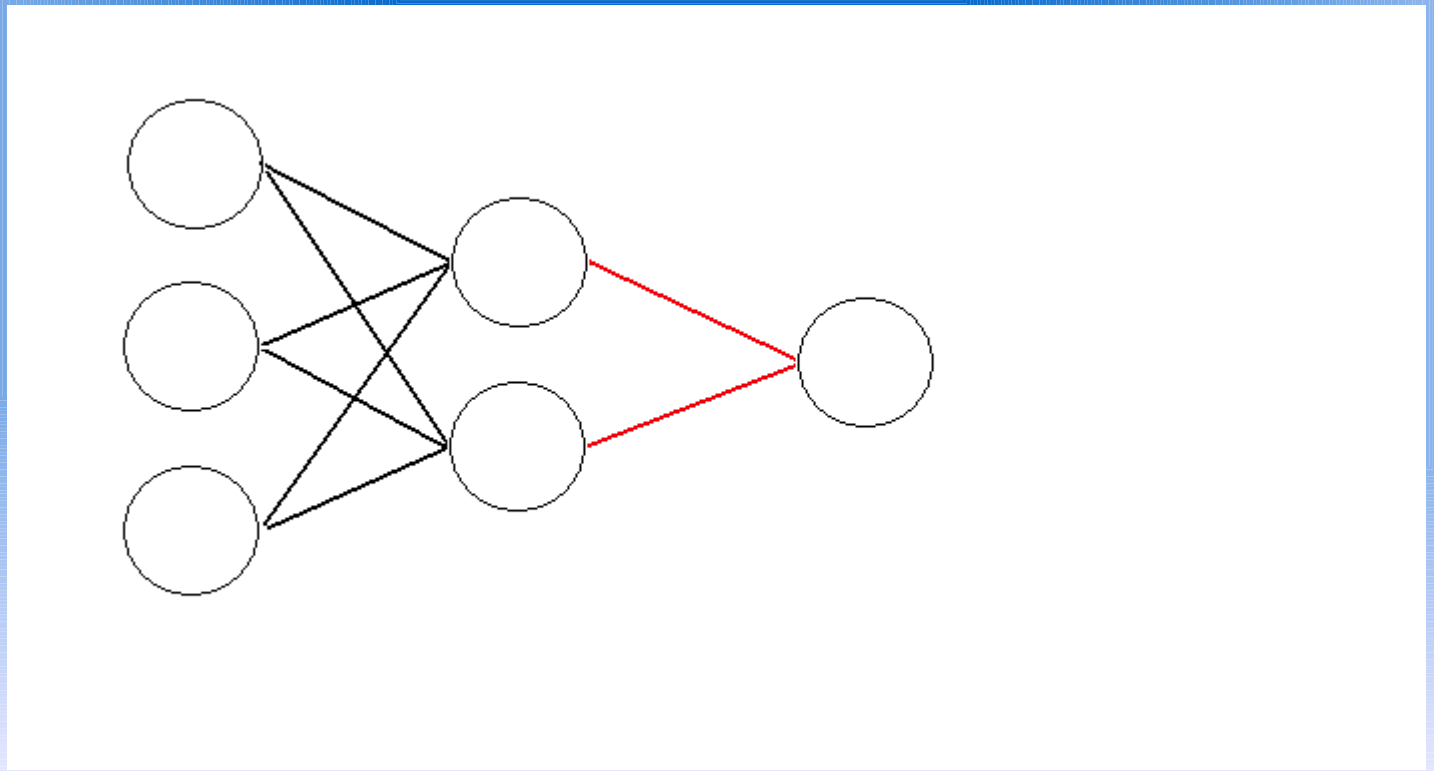
1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and how much the output must be adjusted to match the desired output.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat from step 3 on the neurons at the previous level, using each one's "blame" as its error.



Back-Propagation

Summary of the back-propagation technique:

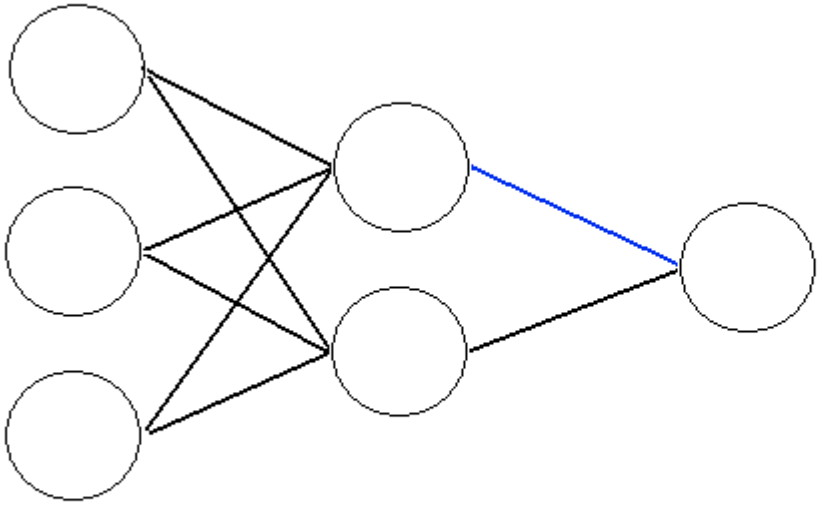
1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and how much the output must be adjusted to match the desired output.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat from step 3 on the neurons at the previous level, using each one's "blame" as its error.



Back-Propagation

Summary of the back-propagation technique:

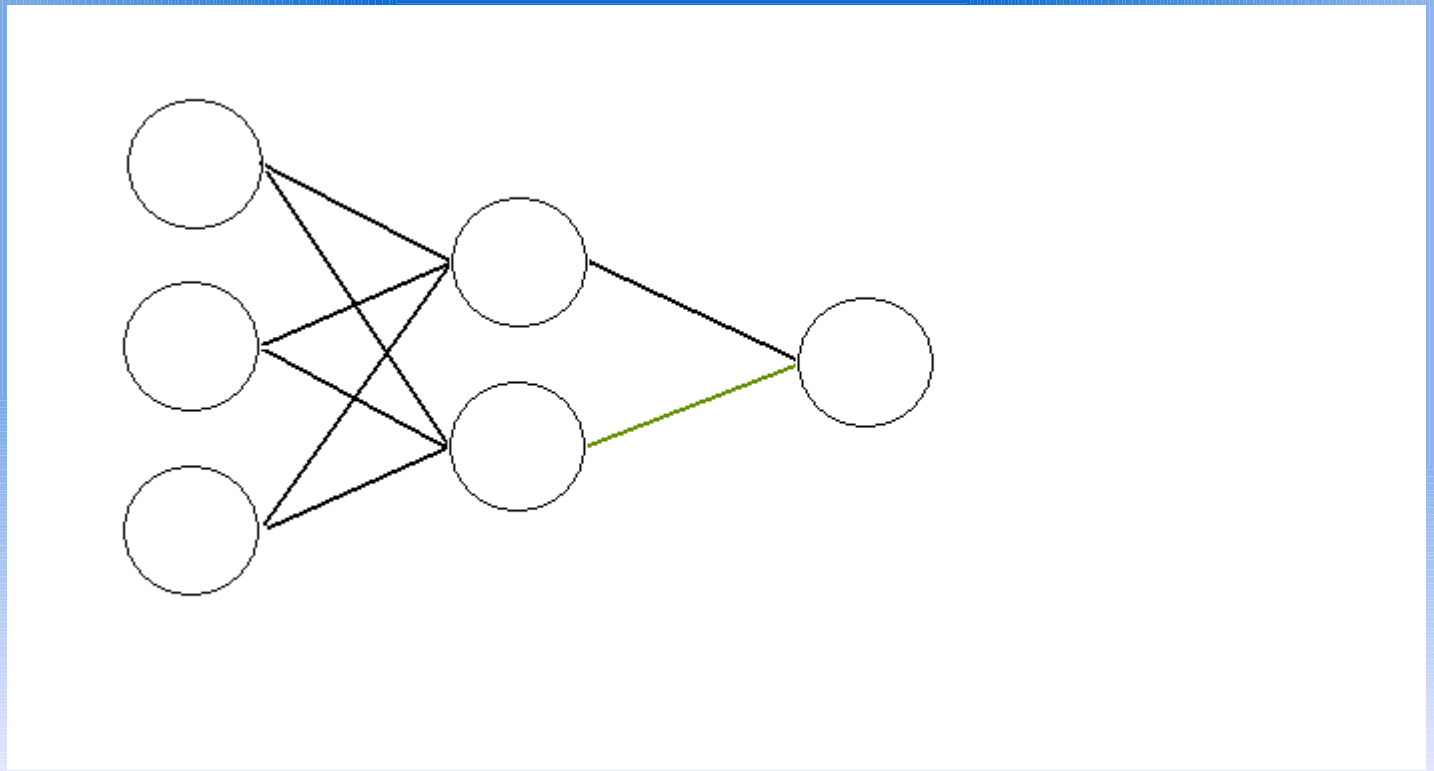
1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and how much the output must be adjusted to match the desired output.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat from step 3 on the neurons at the previous level, using each one's "blame" as its error.



Back-Propagation

Summary of the back-propagation technique:

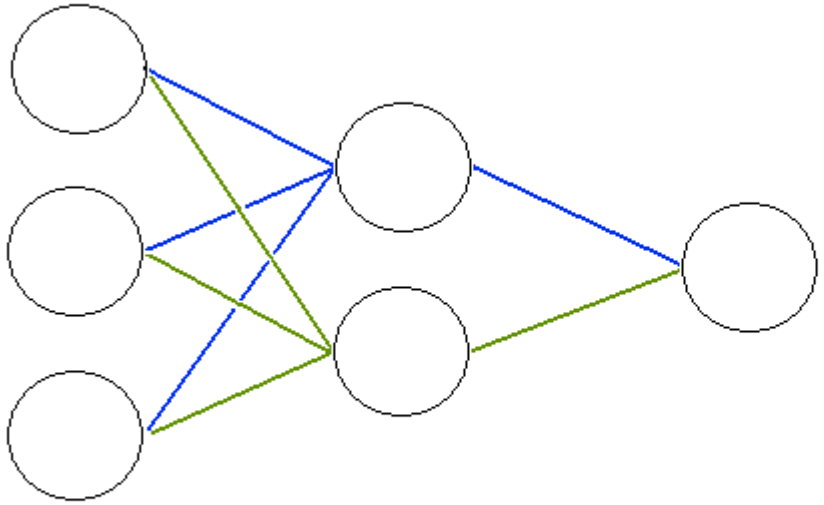
1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and how much the output must be adjusted to match the desired output.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat from step 3 on the neurons at the previous level, using each one's "blame" as its error.



Back-Propagation

Summary of the back-propagation technique:

1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and how much the output must be adjusted to match the desired output.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat from step 3 on the neurons at the previous level, using each one's "blame" as its error.

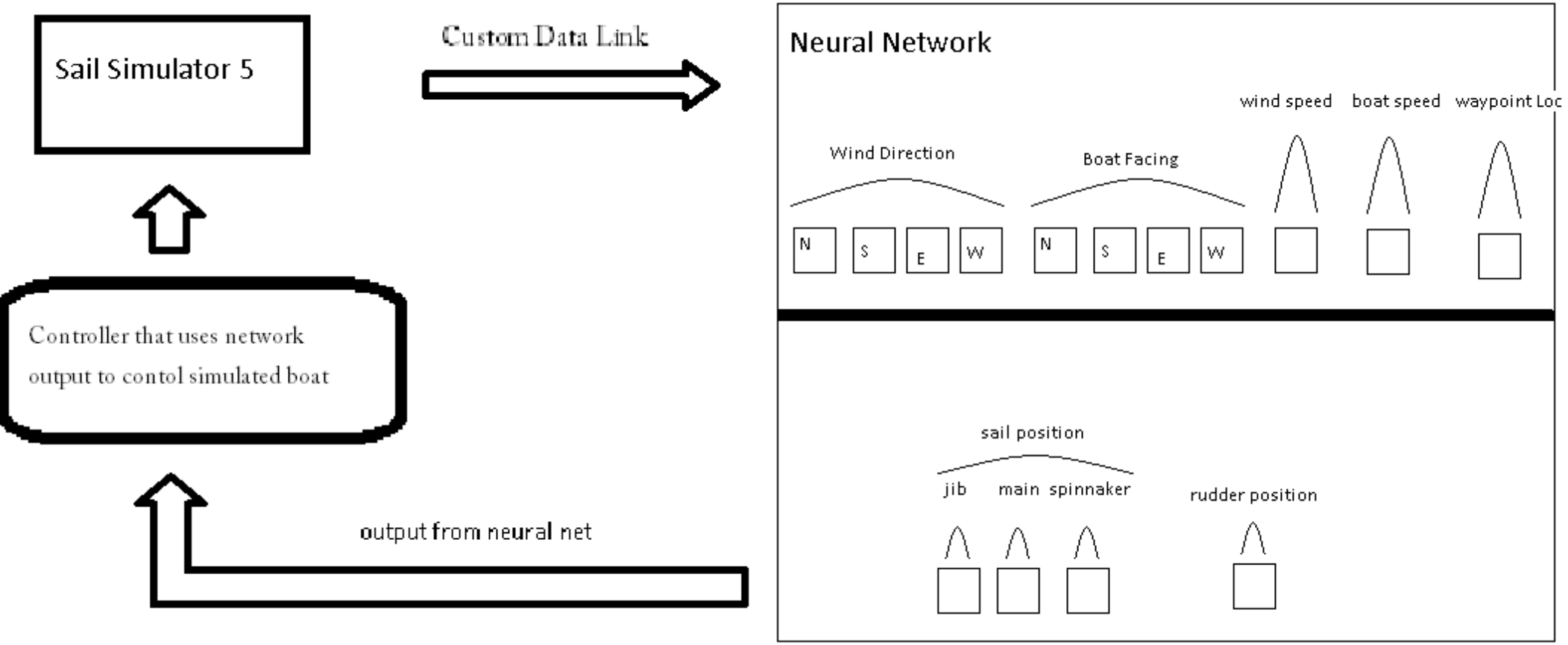


Learning Paradigm

- Supervised Learning
  - The network is presented with a set of input data and a set of desired output
  - The aim is for the network to discover how to manipulate the input data so that it maps to the desired output
  - This method requires that there be a “teacher” or something that knows the correct output for a certain input
  -

Program Interaction

Testing/Training



The Simulator

Sail Simulator 5
Developed by Stentec software
Based in the Netherlands

The simulator has very realistic environment physics and boat handling making it ideal to train a network that could sail an actual boat

http://www.youtube.com/watch?v=tsf3Q5dTuTQ

Custom Data Link

• The link sends data through the serial port, I use a COM port emulator and a com port monitoring program to capture the output as a text file.

•Data sent is: Boat position (latitude and longitude), rudder position, main sheet position, waypoint location, and wind speed and direction.

•Data is stored as hex strings
• Program to translate hex reads input from the text file, translates into decimal and feeds into the input layer of the neural network.

Controller

- ME!!!

- Program will analyze the output of the neural network and tell me to either tighten or loosen the sail or turn left or right

Resources

Stentec software: www.stentec.com

Josh Bongard: *Resilient Machines through Continuous Self Modeling*
http://www.sciencemag.org/cgi/content/full/314/5802/1118

Pomerleau, Dean: *Knowledge-based Training of Artificial Neural Networks*