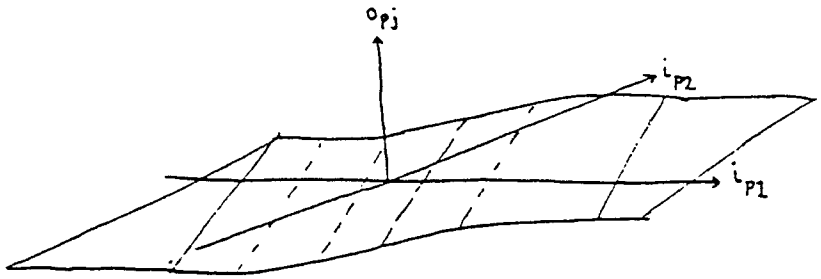
11 May 1989

Neil E Cotter

Universal Approximation

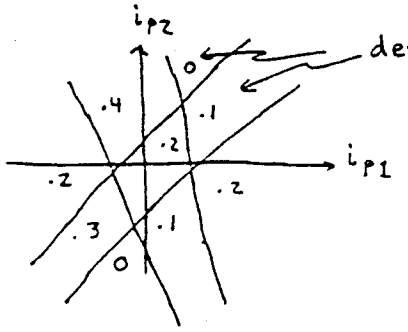
Perceptrons – ~~[illegible]~~



• Response surface for 2-input neuron

A Four-layer Network Can Compute Any Function of N var:

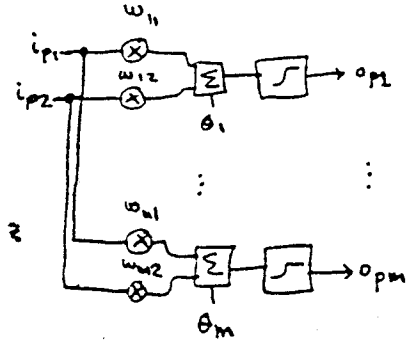
ex:



First layer:

Suppose we make a topographic map of the function and draw lines that approximate the contour lines of equal altitude:

Let each line correspond to the decision line of a neuron on the first layer of the network. Each neuron has inputs $i_{p1}, \ldots, i_{pN}$.



• First layer of network
• Use large synaptic weights so neuron output nearly binary.

Universal Approximation (cont.)

11 May 1989

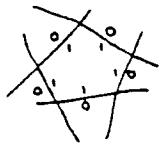
Neil E Cotter

Perceptrons – ~~multilayer perceptrons and other perceptron topics~~

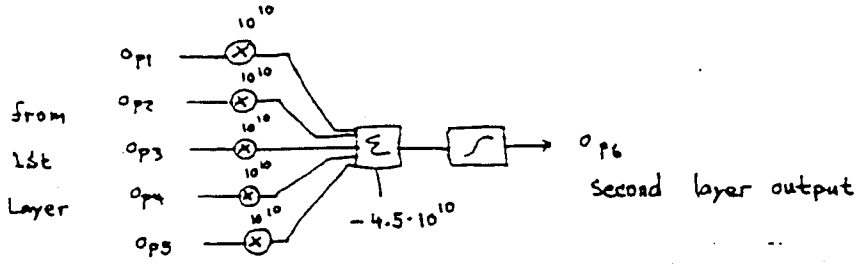
2nd layer

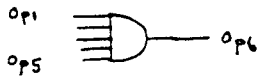
Use AND gates to determine when an input point is inside a polygonal region.

e.g.



A point is inside the pentagon if and only if all five of the neurons on the first layer are outputting ones, "1's". So use a five input AND gate:





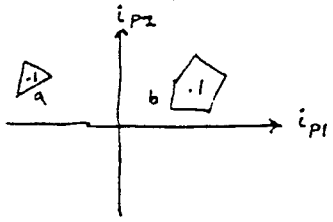
Divide the input space into polygons by adding more neurons to the first layer if necessary.

The output of the second layer tells us which polygonal region we are in. One and only one neuron in the second layer should be outputting a 1 at any given time.

We can assume that each neuron on the second layer receives inputs from all the neurons on the first layer but may ignore some inputs (synaptic weight = 0).

11 May 1989 Perceptrons - ~~Planning~~ Universal Approximation (cont.)
Neil E Cotter 3 rd layer

Suppose points in either of two polygonal regions should give the same network output:



On the 2nd layer there are neurons a and b whose outputs are 1 when the input point is in polygon a or polygon b, respectively.

OR the outputs of neurons a and b to get an output of a neuron that is 1 if and only if the input point is in a region that should give an overall output value of .1.
i.e. combine regions with the same output values.

So the third layer contains a neuron whose output is 1 if the input point is in a or b. There is one such neuron for each of the possible network output values.

4 th layer

Remove the sigmoid squashing function from the neurons on this layer of network. Multiply 3rd layer outputs by the value the network should output. e.g. multiply the a OR b neuron output by .1. The same is done for all the other possible output values. With enough neurons we can get arbitrary accuracy.