Can Computers Think?

Week 9, Th

Kristina Striegnitz

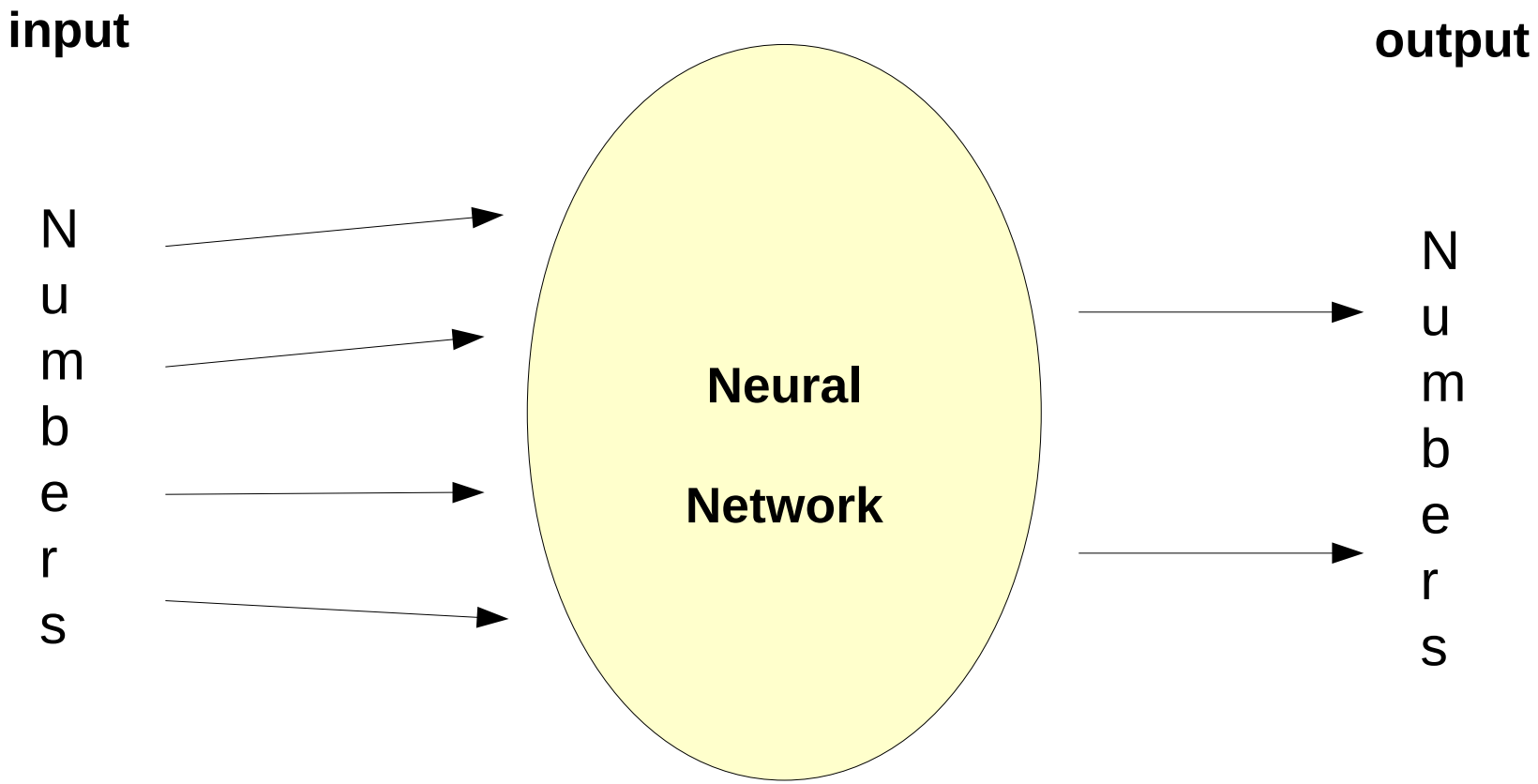
cs.union.edu/~striegnk

striegnk@union.edu

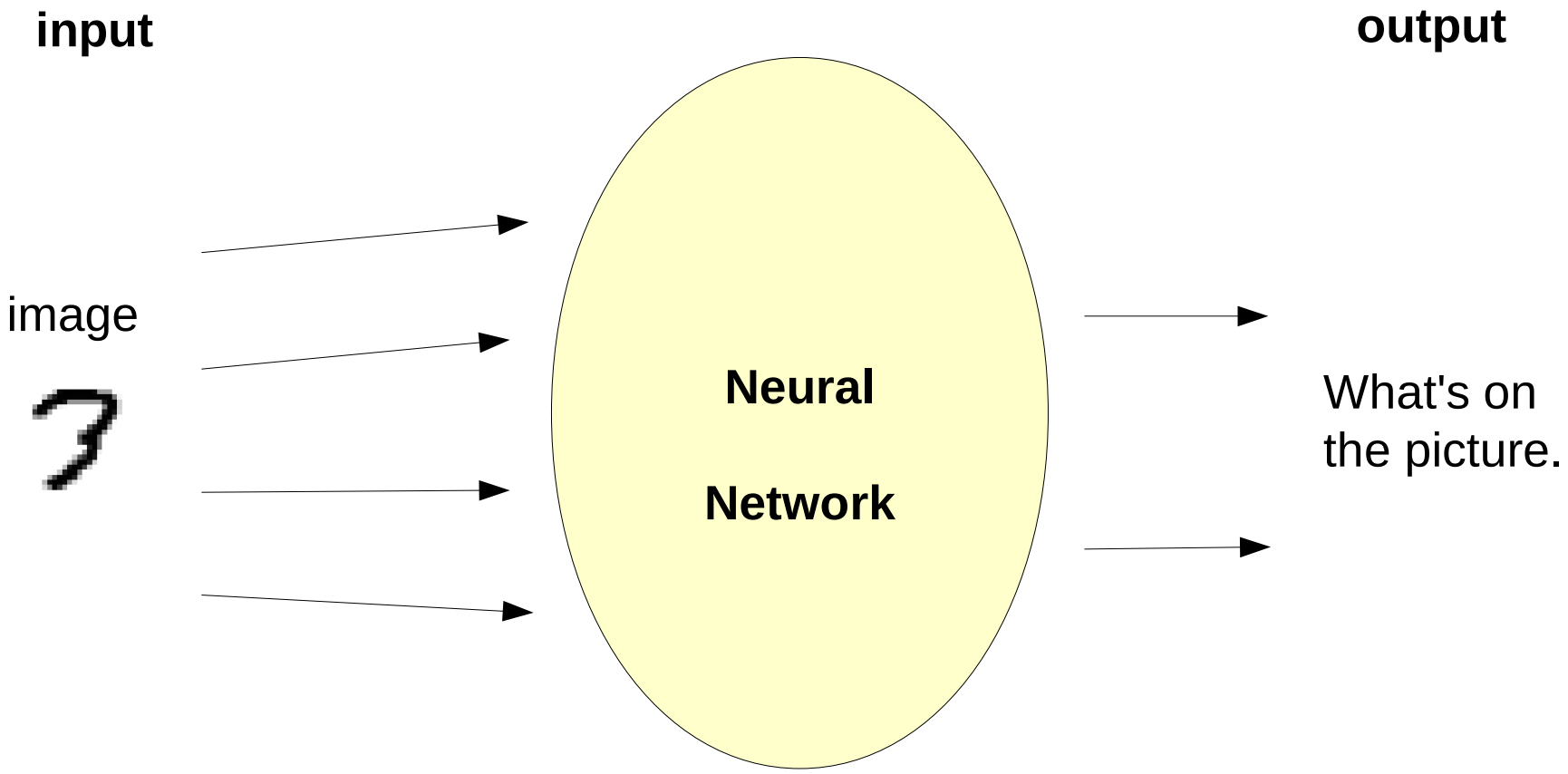
Steinmetz 233

office phone: x6554

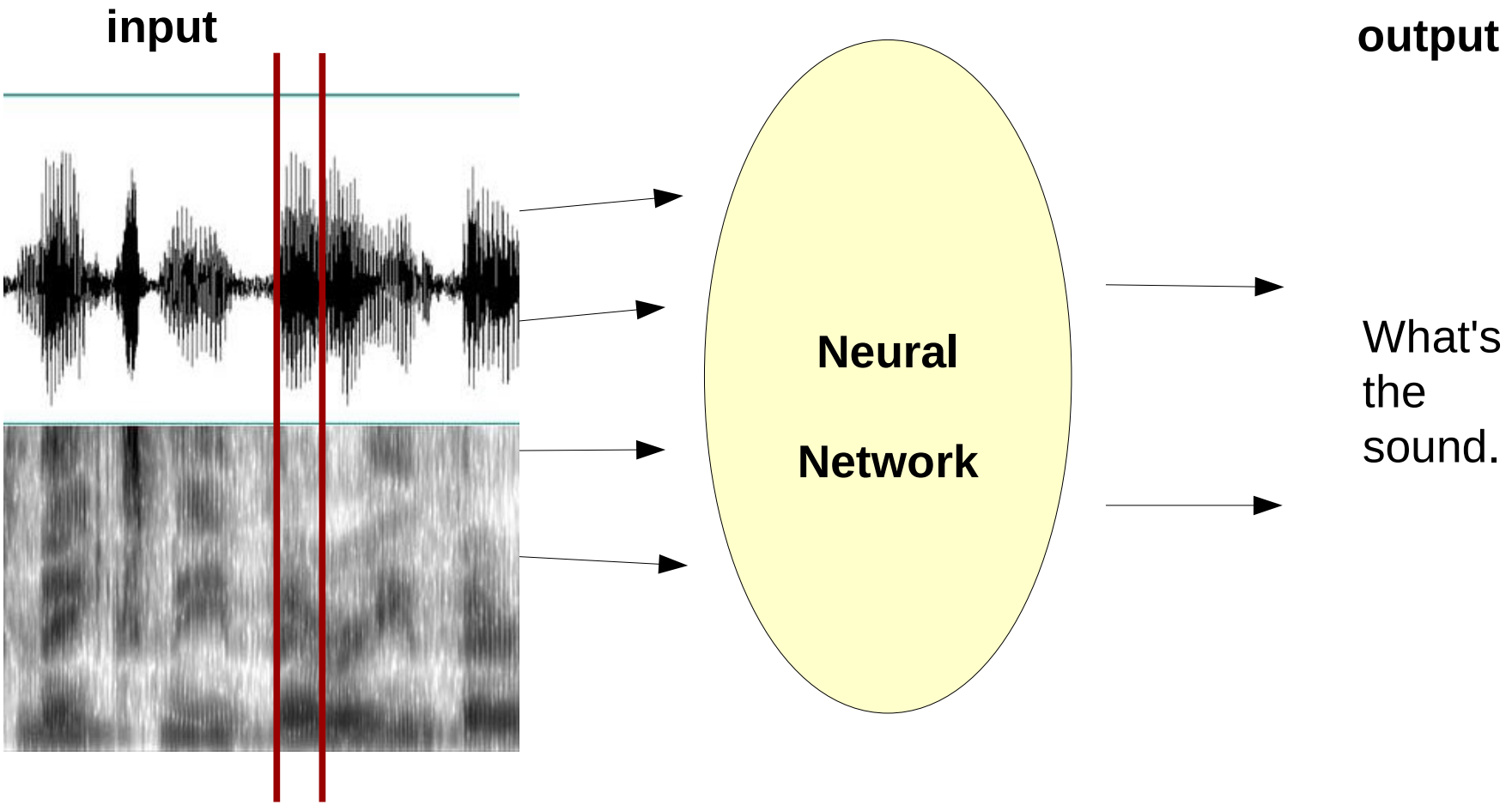
Artificial Neural Networks



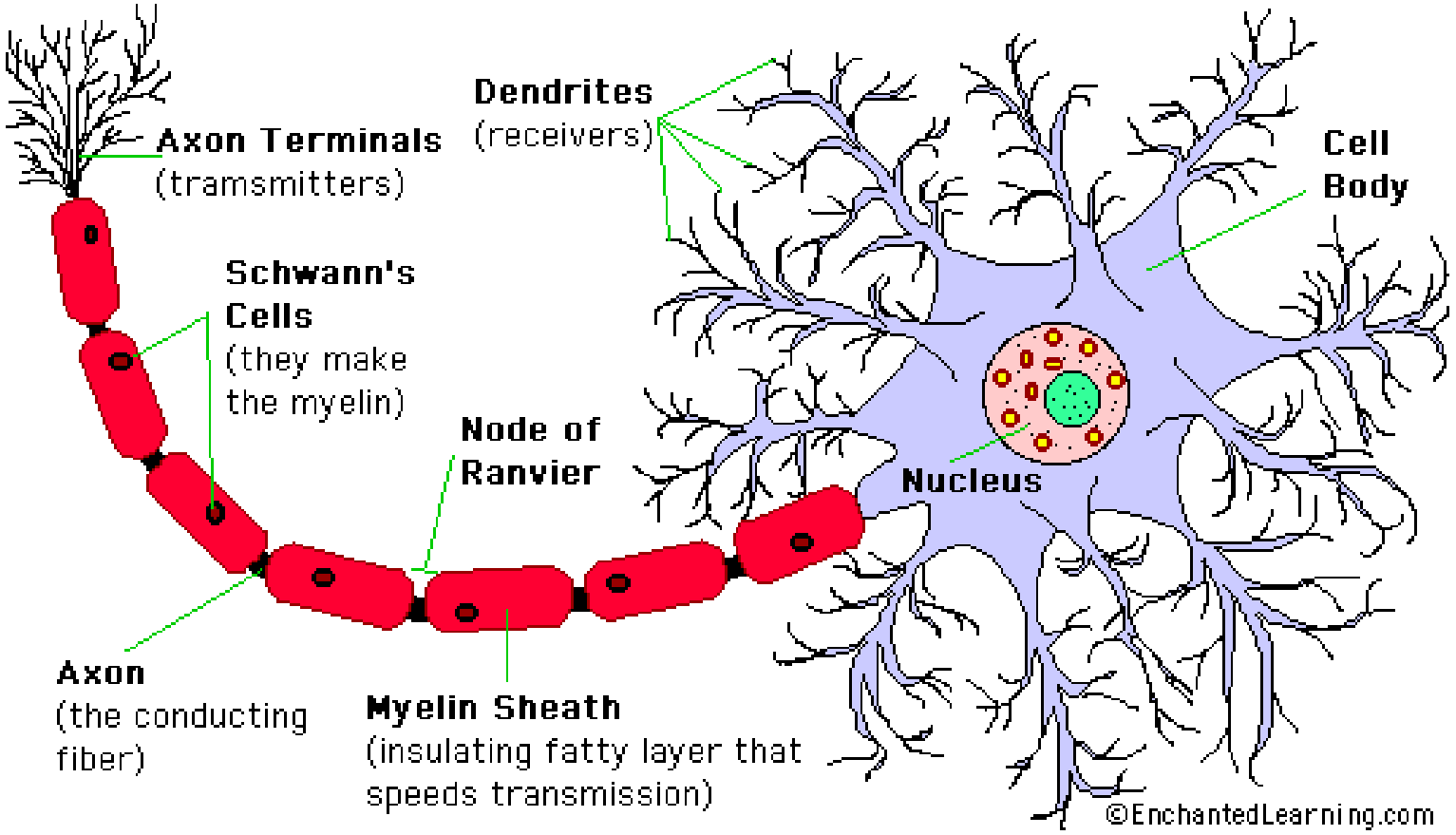
Artificial Neural Networks



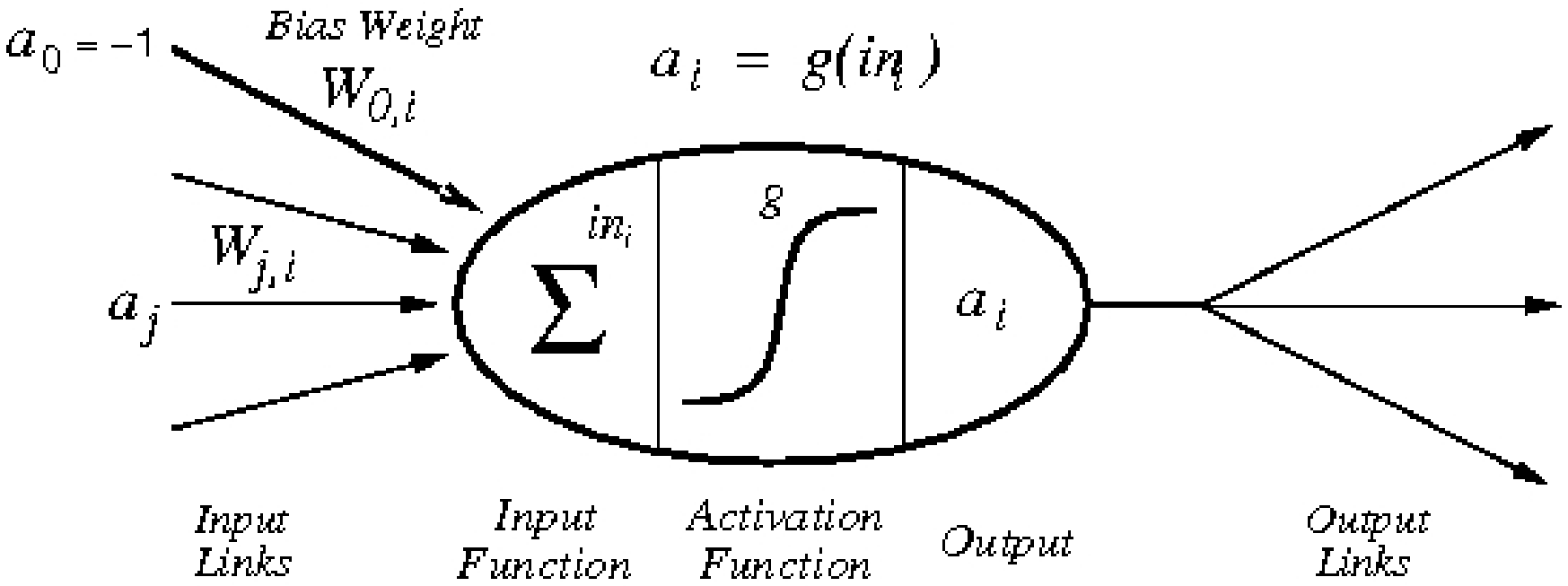
Artificial Neural Networks



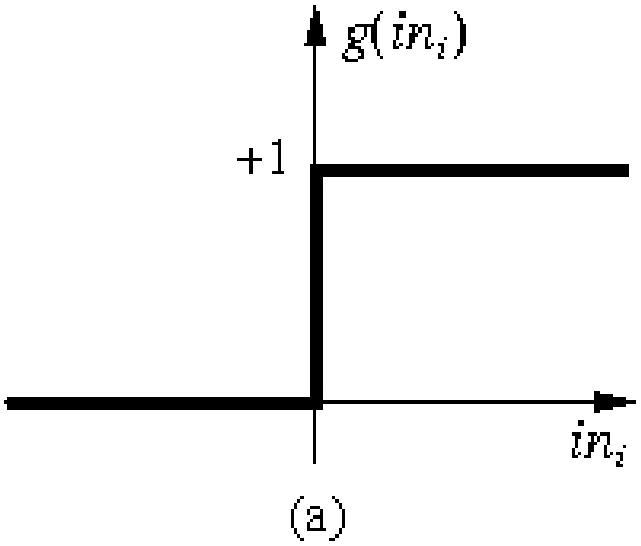
Neurons


Axon Terminals
(tramsmitters)
Dendrites
(receivers)
Cell
Body
Schwann's
Cells
(they make
the myelin)
Node of
Ranvier
Nucleus
Axon
(the conducting
fiber)
Myelin Sheath
(insulating fatty layer that
speeds transmission)
©EnchantedLearning.com

A computational model of a neuron

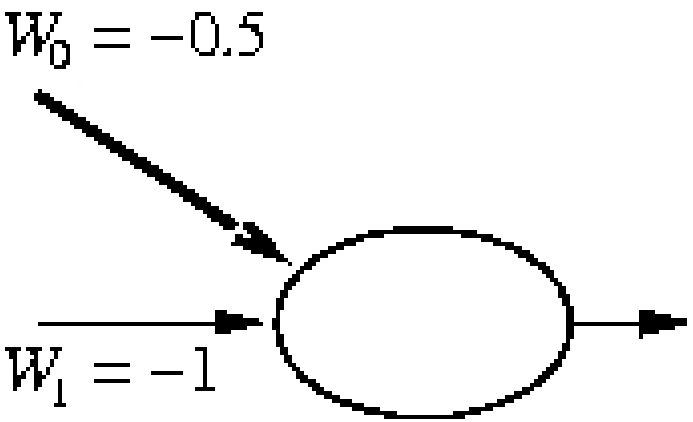


A simple threshold activation function

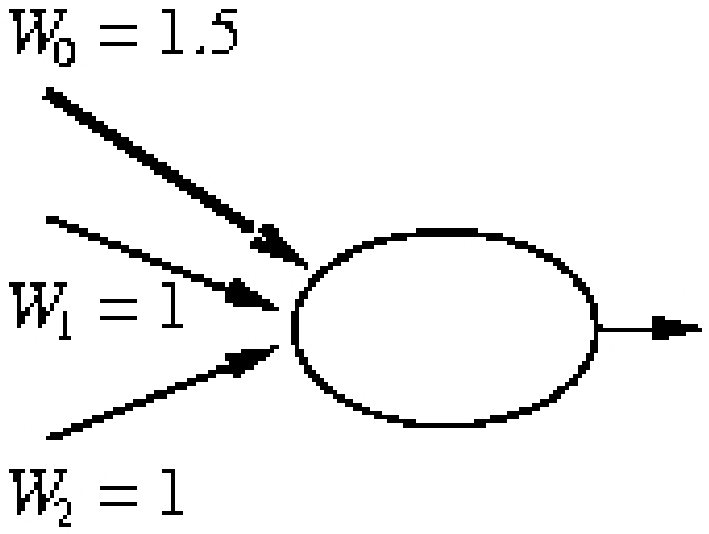


(a)

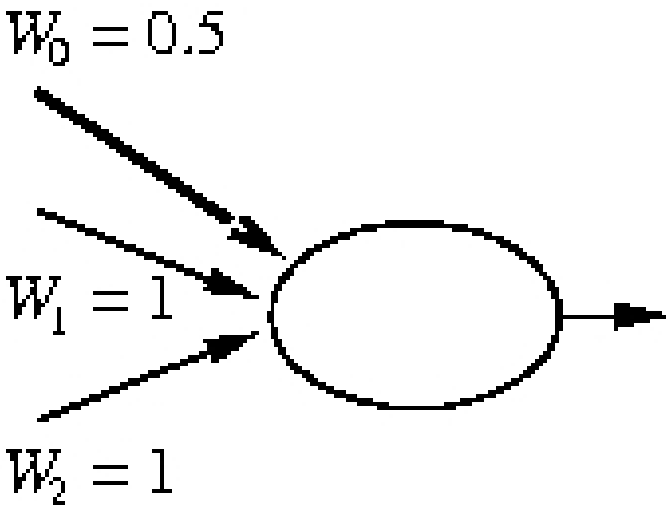
An example neuron



Example 2



Example 3

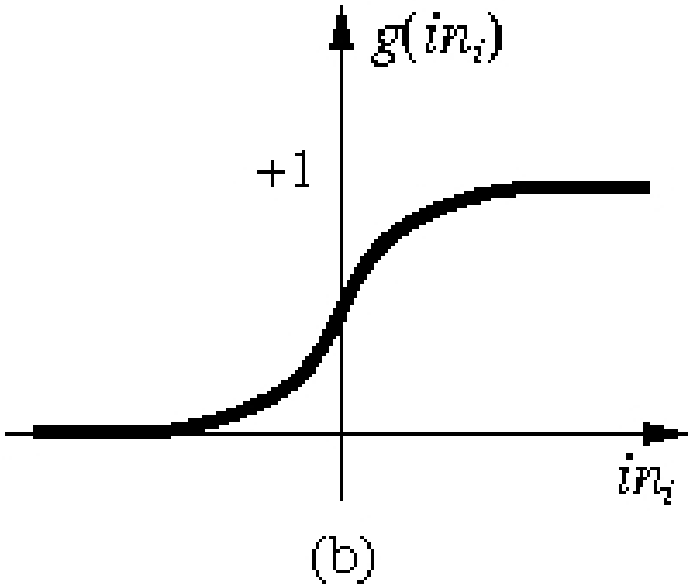


Exercises

1) Design a neuron that takes two inputs (plus the bias) and outputs 1 if both inputs are 0 and outputs 0 otherwise.

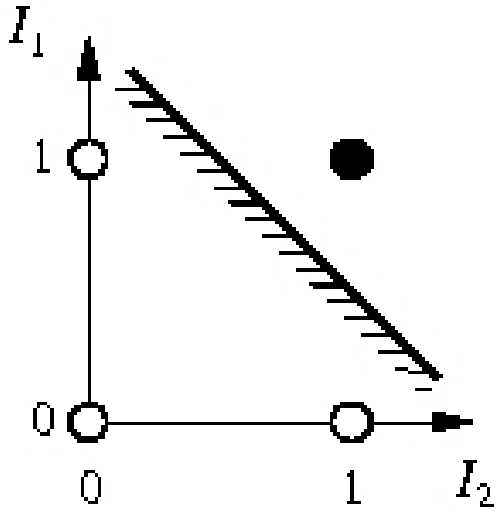
2) Design a neuron that takes two inputs (plus the bias) and outputs 0 if both inputs are 1. In all other cases it outputs 1.

A more common activation function

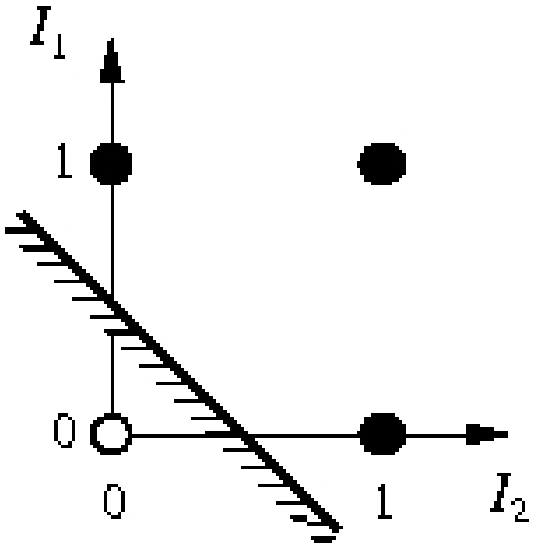


(b)

Linear separability

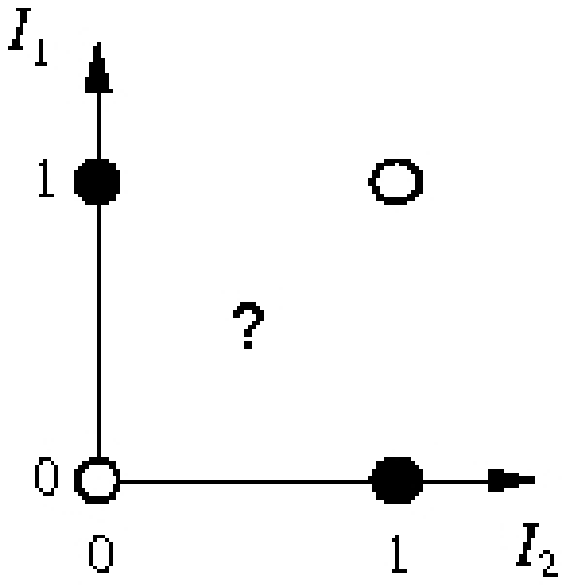


(a) I_1 **and** I_2



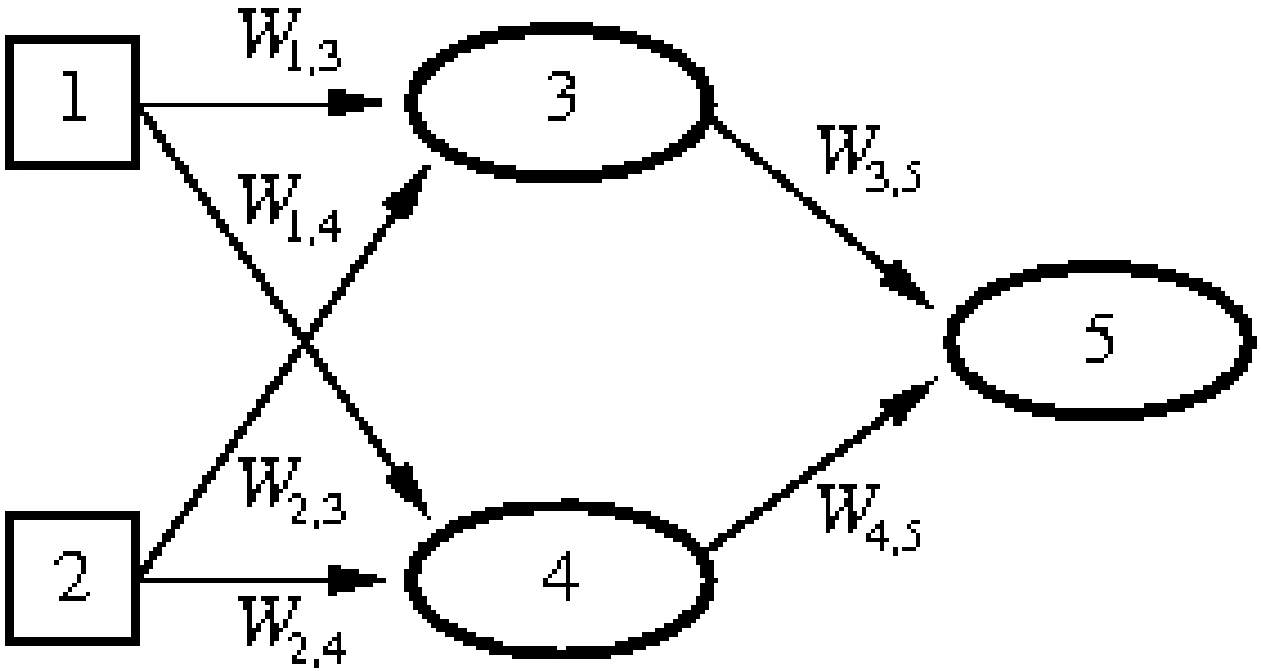
(b) I_1 **or** I_2

Something our artificial neuron cannot express



(c) I_1 **xor** I_2

Networks of Neurons



XOR neural net

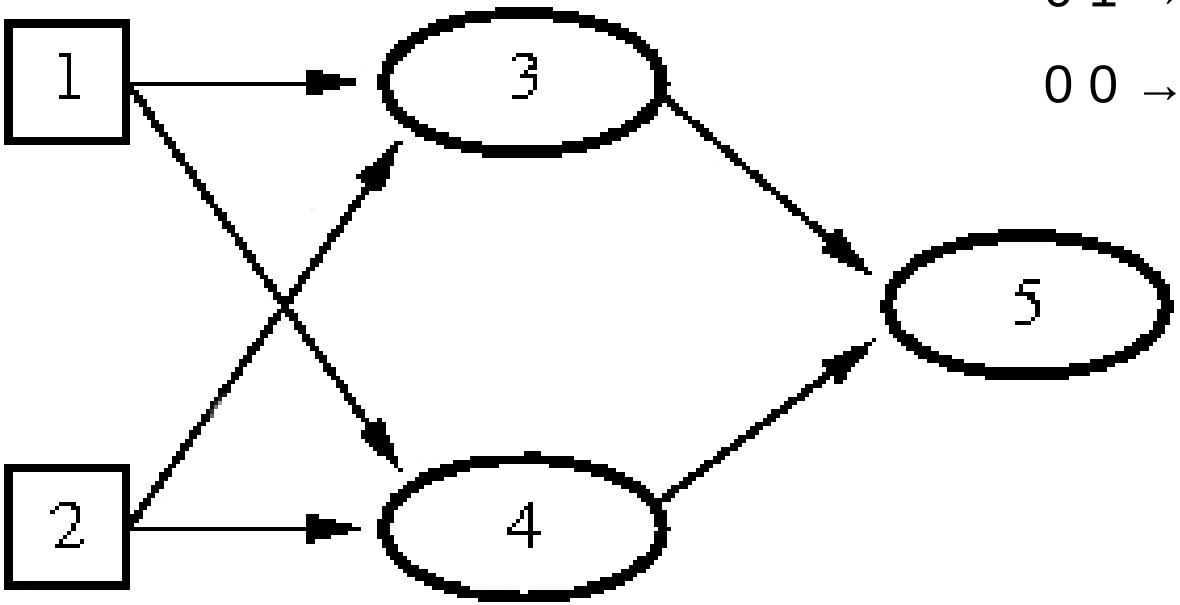
desired behavior:

1 1 → 0

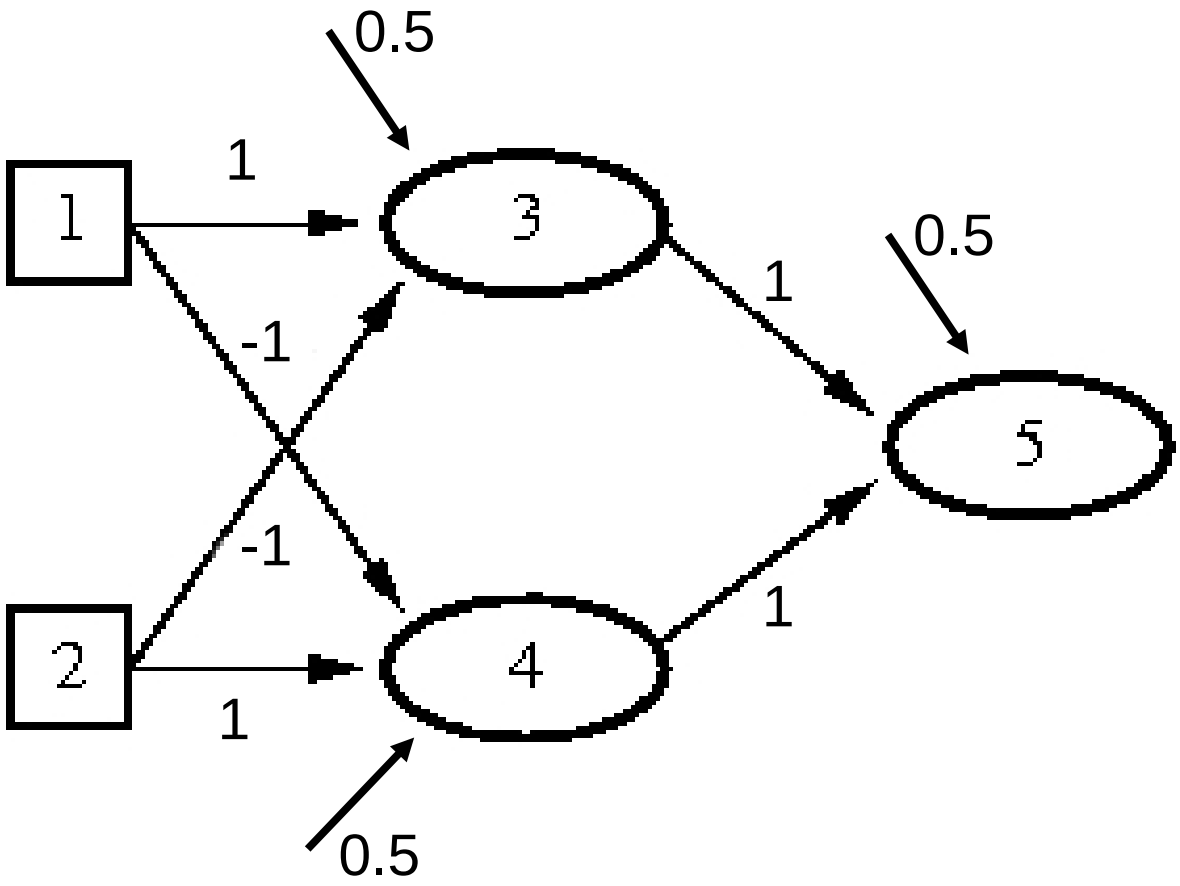
1 0 → 1

0 1 → 1

0 0 → 0

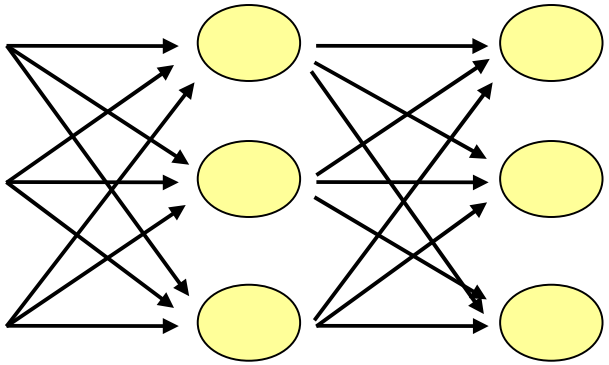


XOR neural net

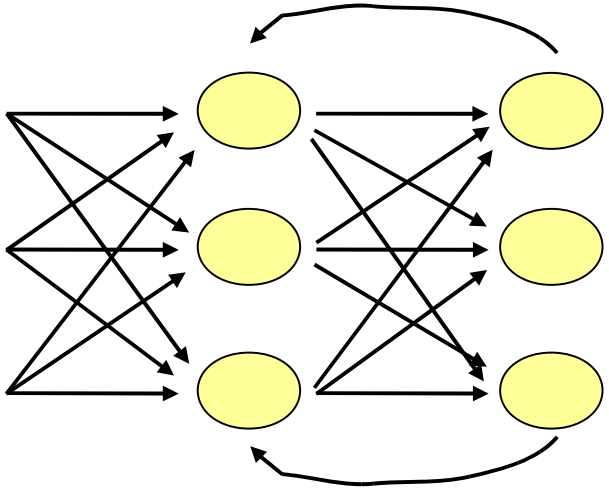


Network Architectures

feed-forward:



recurrent:



Most common: feed-forward networks with one hidden layer.

Where do the weights come from?

Given :

lots of example inputs + the desired output for each of these inputs

Learning strategy (backpropagation):

1. randomly assign small weights

– for each pattern:

– feed pattern into the NN

– compare actual output to desired output and adjust weights to make actual output closer to desired

1. repeat from step 2 until the weight changes get very small (or some other stopping criterion is met)

Exercises

- Download xor.py.
- Add a function which builds, trains and tests a neural network that computes a boolean *and*.
- Add a function which builds, trains and tests a neural network that decides whether the inputs represent a binary number which is divisible by 3. Use four input units – this will allow you to represent the numbers 0,...,15.