CogSci 109: Lecture 22

Fri Nov 30, 2007

Multilayer artificial neural networks, examples, and applications

Outline for today

- Announcements
- Homework announcement
- PLU demo code
  - **(not using the neural network toolbox in matlab)**
- Completing the PLU details - XOR, NOT
- How do we resolve this issue?
  - **Historical notes**
- Feedforward/feedback structures
  - **Important note for network topologies**
  - **Big dog**
  - **Inverted pendulum**
- Multilayer neural networks
  - **Some of the typical network topologies**
- Matlab neural network toolbox demos

Announcements

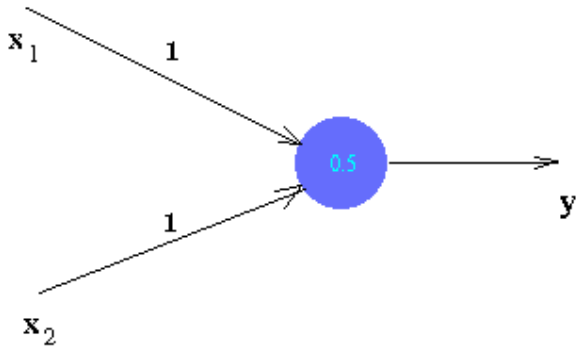
- Homework 6
- Homework 5 notes
  - **How to download papers for free using your UCSD student access**
  - **Previous homework and midterm returns**
    - Come discuss issues
- Grade program online
  - **Blank midterm short answer sections**
  - **Student who may not be registered**
  - **About grade changes/late midterms/homeworks 0's**

Generic PLA matlab code (not using neural network toolbox)

- Original TLU's did not have learning rule - weights had to be designed
- 50's Rosenblatt's main contributions were the perceptron learning rule
- Demo/explanation
  - **Binary classifier**

Limitations of a single neuron

- XOR problem -
  - build a single layer, single unit perceptron which takes 2 boolean inputs and outputs the XOR of them. What we want is a perceptron which will output 1 if the two inputs are different and 0 otherwise.
  - Consider the following perceptron as an attempt to solve the problem



Input	Input	Desired Output
0	0	0
0	1	1
1	0	1
1	1	0

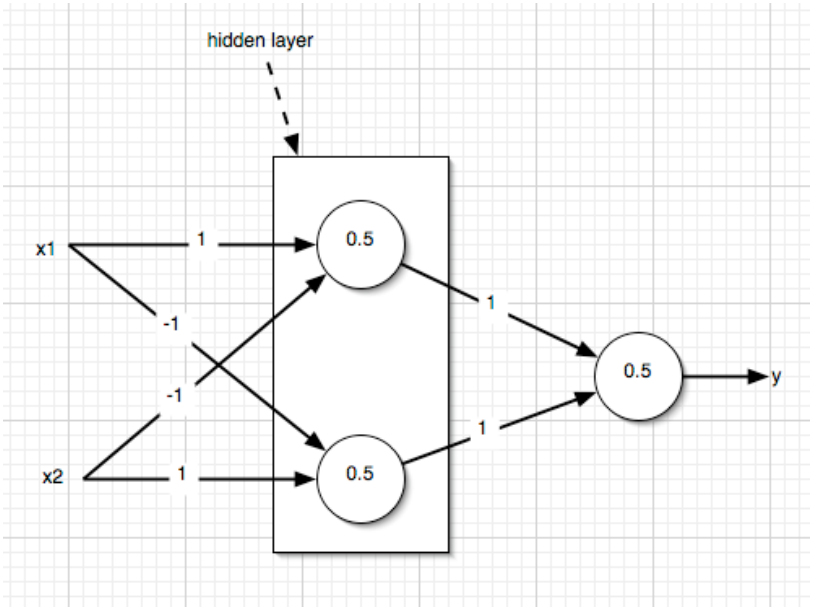
•If the inputs are both 0, then net input is 0 which is less than the threshold (0.5). So the output is 0 - desired output.
•If one of the inputs is 0 and the other is 1, then the net input is 1. This is above threshold, and so the output 1 is obtained.
•But the given perceptron fails for the last case

Limitations of single layer perceptrons (II)

- Widely publicized in the book Perceptrons [MiPa69] by Marvin Minsky and Seymour Papert
- It was not until the 1980s that these limitations were overcome with im- proved (multilayer) perceptron networks and associated learning rules
  - The funding and thus literature for ANN's slowed to a crawl until then!

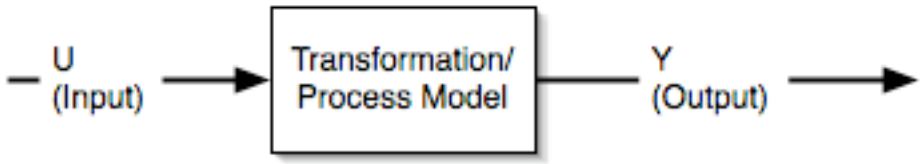
How do we resolve this?

- Feedforward multilayer networks
  - **Simple implementation**
  - **Computational capability**
  - **Input-output data**
  - **No feedback (signals only travel forward)**
- It can be shown that by connecting together multiple TLU's in a two layer network we can solve the XOR problem
  - **Implements two linear decision boundaries**

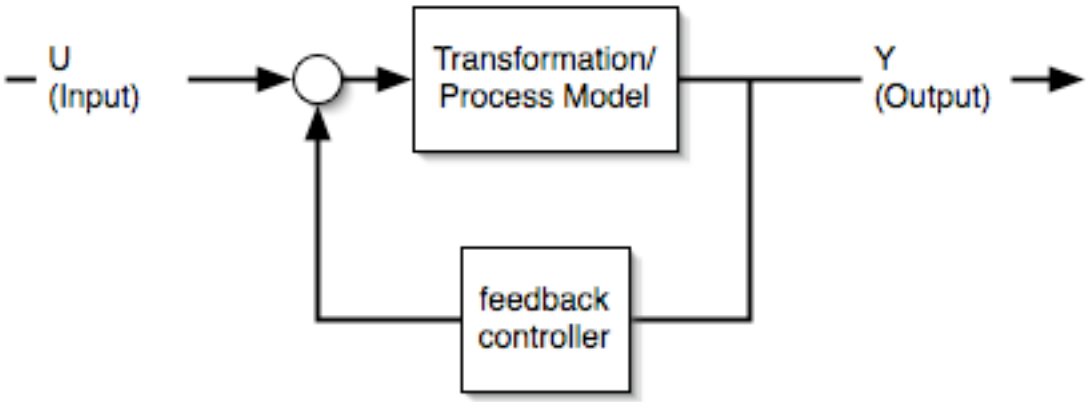


An important concept…

- Feedforward system



- Feedback system



Feedforward-feedback example

- Position control of a motor angle or human limb joint angle
- Path planning
- Feedforward has advantages and drawbacks
  - **Main drawback -** ***model is never perfect, and noise can cause severe drift over time, leading to inaccuracies***
    - *Any small error in the model tends to cause massive inaccuracies*
    - *Any disturbances cause errors - noise or external inputs*
  - **Advantage -** ***simplicity in computation and sensor requirements***
- Feedback has advantages of robustness and error correction

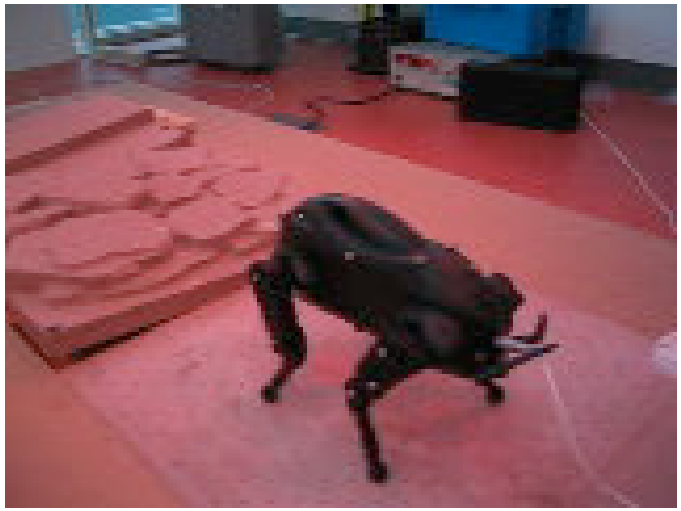
A common feedback example - inverted pendulum control

- People standing or walking can be modeled as inverted pendulums

Another example - robotics application

- Big dog video

- Littledog video

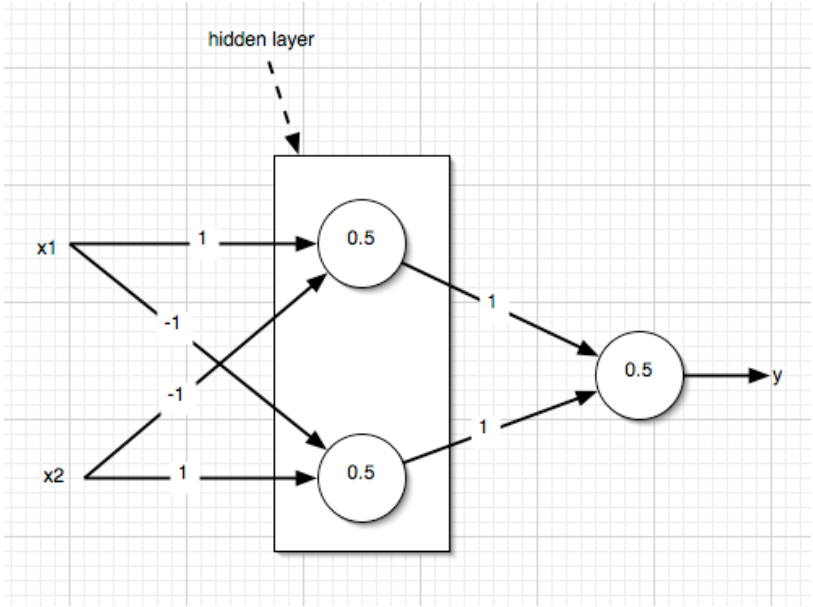


Back to neural networks…

- Now that we have a concept of feedforward and feedback, and how single unit perceptrons work, let's move on to combinations of units to multi-layer networks
- More details next time but main applicatios of ANN's are
  - **Function fitting**
    - Fit this data without an equation!!!
  - **Classification**
    - blue cat or red cat?

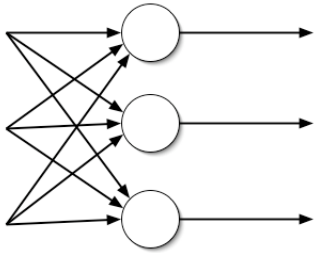
Multilayer networks

- Hidden nodes/layers - intermediate node layers which are NOT directly connected to the outside world (input or output)

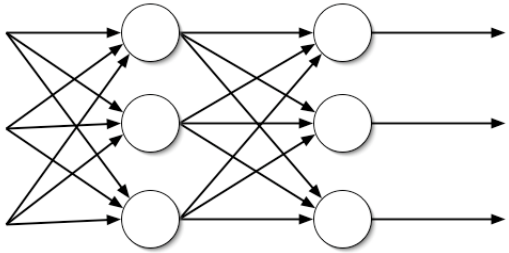


Some typical network topologies

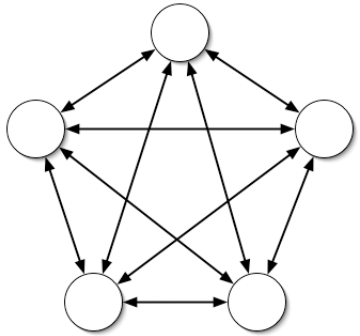
Single layer perceptron



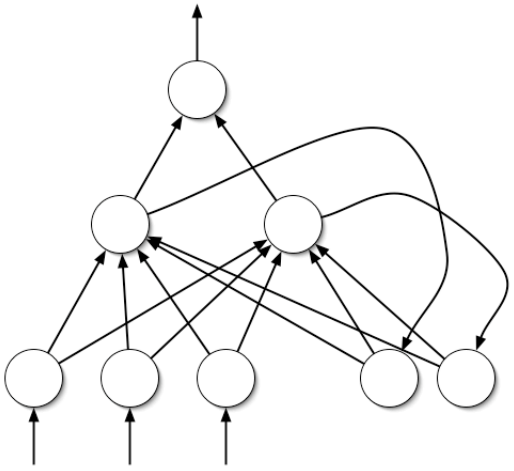
Multi-layer perceptron



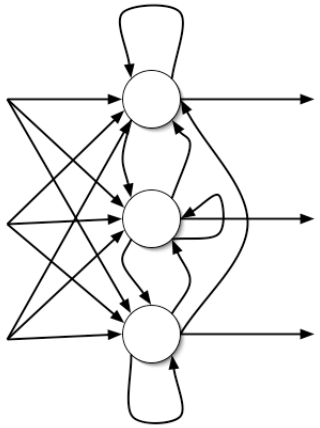
Hopfield network



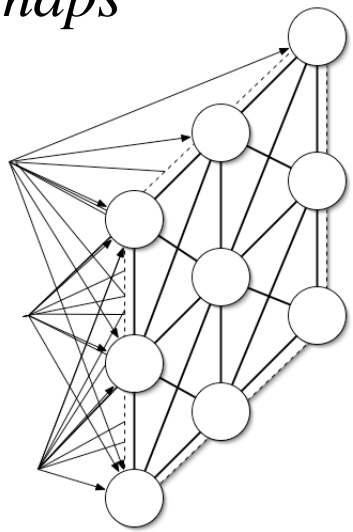
Elman recurrent network



Competitive networks



Self-organizing maps



Other activation function concepts

- Threshold
- Sigmoid
- Logarithmic
- Linear
- Many others

Neural Network Demos in matlab

- In matlab (you need the Neural Network Toolbox)
  - **nnd2n1 One-input neuron demonstration.**
  - **nnd2n2 Two-input neuron demonstration.**
  - **nnd4db Decision boundaries demonstration.**
  - **nnd4pr Perceptron rule demonstration.**
  - **nnd9sdq Steepest descent for quadratic function demonstration.**
  - **nnd11nf Network function demonstration.**
  - **nnd11bc Backpropagation calculation demonstration**
  - **nnd11fa Function approximation demonstration.**
  - **nnd11gn Generalization demonstration.**