Artificial Neural Networks

SECTIONS 18.6 & 18.7
COSC 370
SPRING 2013
ALAN C. JAMIESON

SOME SLIDE CONTENT FROM RUSSELL & NORVIG PROVIDED SLIDES

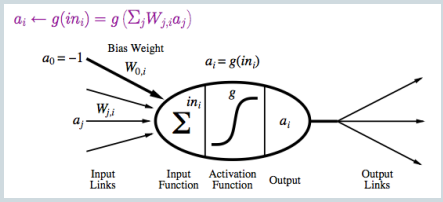
- Neural Networks
- Linear Classification
- Feed-forward Networks
- Recurrent Networks (briefly)

Neural Networks

- AI field based on and an offshoot of neuroscience.
- Original basis from work by Alexander Bain and William James in the late 19th century
- Artificial Neural Networks spawned from work in the 1940s.
- Also known as – connectionism, threshold logic, neural computation.
- Great for pattern recognition and machine learning tasks.

McCulloch-Pitts Neural Unit

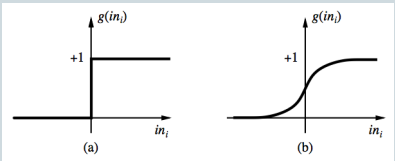
- We're basically using the same model from 1943:



- Linear function of the inputs.

Activation Functions

- Step function/threshold function (a) or sigmoid function (b)



- Changing the bias weight will change where the threshold "step" happens.

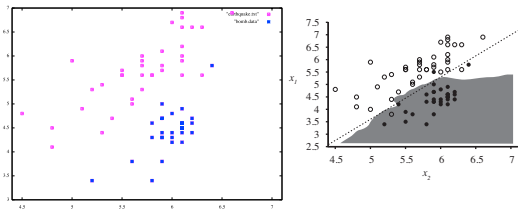
Linear Classification/Regression

- Approach to modeling the relationship between two scalars – useful for forecasting.
- Given a selection of linear attributes, a classifier identifies an object's grouping.
- If you map the data, a boundary line (or surface for higher dimensions) will be apparent based on a hypothesis that splits the data. This leaves an equation:

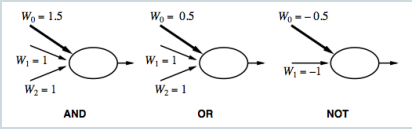
  $4.9 + 1.7x - y = 0$

- This equation is a linear separator.

Linear Classification Example



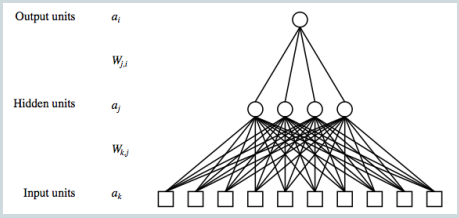
Boolean Gate Implementations



Feed-Forward Network

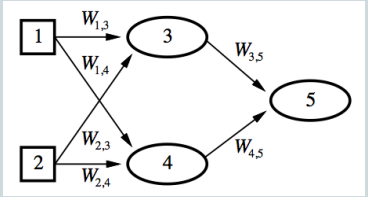
- Directed acyclic graph topology – each vertex is a "neuron".
- Implementation of functions only, no state!
- Single-layer perceptrons – single layer, output units all operate separately, no shared weights.
- Multi-layer perceptrons – multiple layer, each layer is fully connected within hidden units between layer, each layer has the same weight.

Multi-layer Example



Simple Feed-Forward

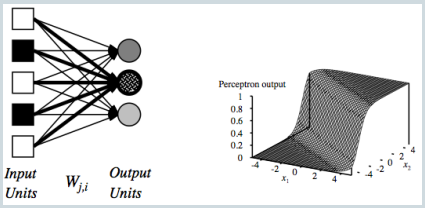
- Parameterized family of nonlinear functions.



- Adjusting the weights changes the function – make alterations for learning this way!

Single-Layer

- Output units all operate separately!
- Adjusting weights moves the location, orientation, and steepness of the cliff.



Handling Learning

Learn by adjusting weights to reduce error on training set

The squared error for an example with input $\mathbf{x}$ and true output y is

$$E = \frac{1}{2}Err^2 \equiv \frac{1}{2}(y - h_{\mathbf{W}}(\mathbf{x}))^2 ,$$

Perform optimization search by gradient descent:

$$\frac{\partial E}{\partial W_j} = Err \times \frac{\partial Err}{\partial W_j} = Err \times \frac{\partial}{\partial W_j}(y - g(\Sigma_{j=0}^{n} W_j x_j)) = -Err \times g'(in) \times x_j$$

Simple weight update rule:

$$W_j \leftarrow W_j + \alpha \times Err \times g'(in) \times x_j$$

E.g., +ve error $\Rightarrow$ increase network output
$\Rightarrow$ increase weights on +ve inputs, decrease on -ve inputs

Recurrent Networks (briefly)

- Hopfield nets – symmetric weights, holographic associative memory
- Boltzmann Machines – stochastic activators, basic probabilistic model in ANN.
- Internal state, cyclic, can oscillate.

Exercise

- Construct an ANN: 18.19