

Neural Networks

- These representations are inspired by neurons and their connections in the brain.
- Artificial neurons, or **units**, have inputs, and an output. The output can be connected to the inputs of other units.
- The output of a unit is a parameterized non-linear function of its inputs.
- Learning occurs by adjusting parameters to fit data.
- Neural networks can represent an approximation to any function.



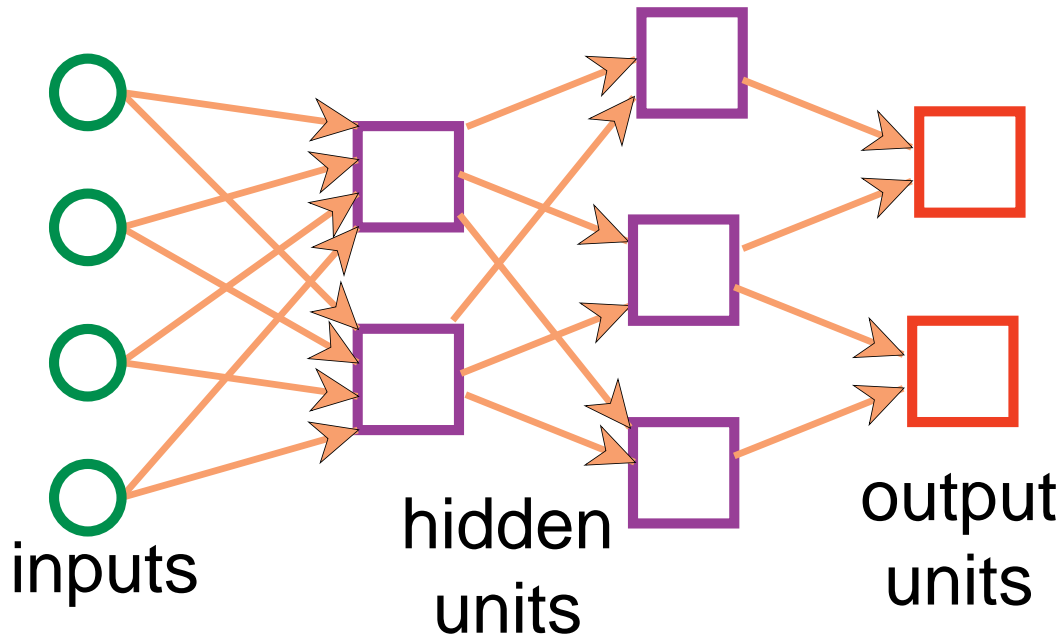
Why Neural Networks?

- As part of neuroscience, in order to understand real neural systems, researchers are simulating the neural systems of simple animals such as worms.
- It seems reasonable to try to build the functionality of the brain via the mechanism of the brain (suitably abstracted).
- The brain inspires new ways to think about computation.
- Neural networks provide a different measure of simplicity as a learning bias.



Feed-forward neural networks

- Feed-forward neural networks are the most common models.
- These are directed acyclic graphs:



The Units

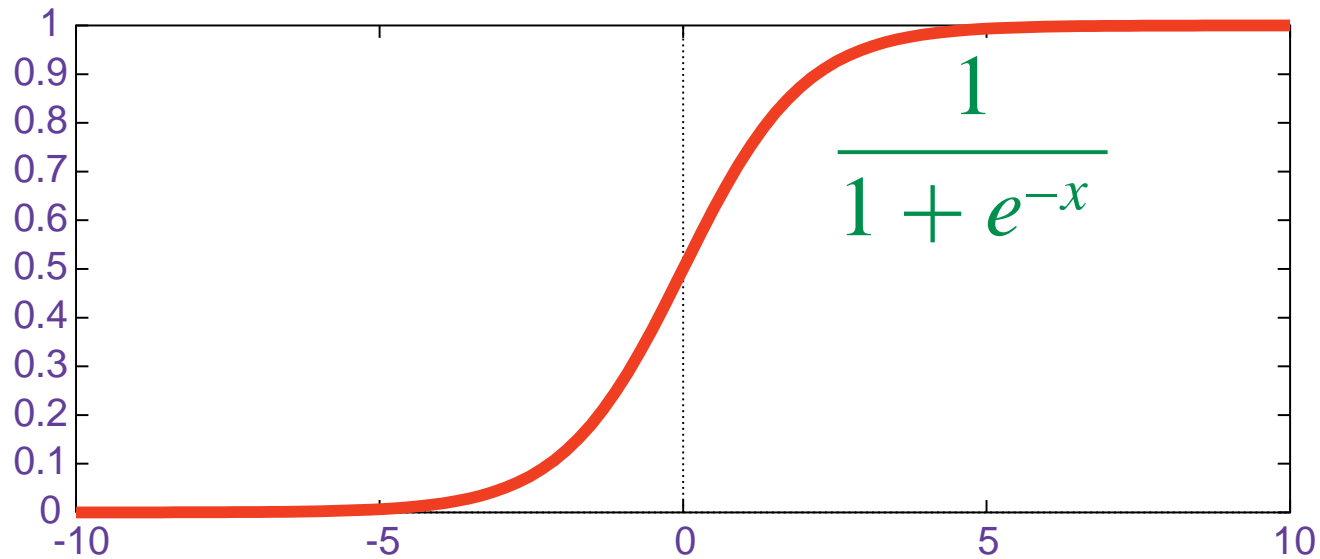
A unit with k inputs is like the parameterized logic program:

$$\text{prop}(\text{Obj}, \text{output}, V) \leftarrow$$
$$\text{prop}(\text{Obj}, \text{in}_1, I_1) \wedge$$
$$\text{prop}(\text{Obj}, \text{in}_2, I_2) \wedge$$
$$\dots$$
$$\text{prop}(\text{Obj}, \text{in}_k, I_k) \wedge$$
$$V \text{ is } f(w_0 + w_1 \times I_1 + w_2 \times I_2 + \dots + w_k \times I_k).$$

- I_j are real-valued inputs.
- w_j are adjustable real parameters.
- f is an activation function.

Activation function

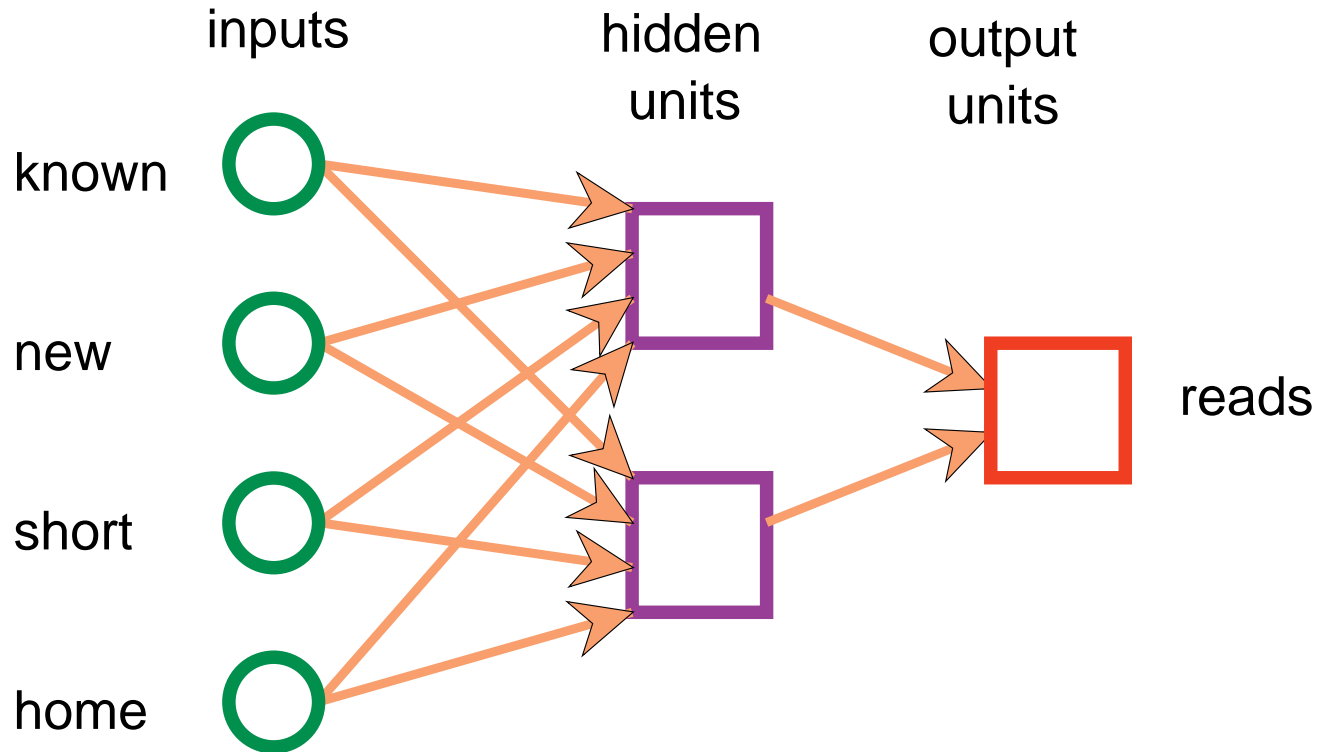
A typical activation function is the **sigmoid** function:



$$f(x) = \frac{1}{1 + e^{-x}}$$

$$f'(x) = f(x)(1 - f(x))$$

Neural Network for the news example



Axiomatizing the Network

- The values of the attributes are real numbers.
- Thirteen parameters w_0, \dots, w_{12} are real numbers.
- The attributes h_1 and h_2 correspond to the values of hidden units.
- There are 13 real numbers to be learned. The hypothesis space is thus a 13-dimensional real space.
- Each point in this 13-dimensional space corresponds to a particular logic program that predicts a value for *reads* given *known*, *new*, *short*, and *home*.



predicted_prop(Obj, reads, V) ←

prop(Obj, h₁, I₁) ∧ prop(Obj, h₂, I₂) ∧

V is f(w₀ + w₁ × I₁ + w₂ × I₂).

prop(Obj, h₁, V) ←

prop(Obj, known, I₁) ∧ prop(Obj, new, I₂) ∧

prop(Obj, short, I₃) ∧ prop(Obj, home, I₄) ∧

V is f(w₃ + w₄ × I₁ + w₅ × I₂ + w₆ × I₃ + w₇ × I₄).

prop(Obj, h₂, V) ←

prop(Obj, known, I₁) ∧ prop(Obj, new, I₂) ∧

prop(Obj, short, I₃) ∧ prop(Obj, home, I₄) ∧

V is f(w₈ + w₉ × I₁ + w₁₀ × I₂ + w₁₁ × I₃ + w₁₂ × I₄)

Prediction Error

- For particular values for the parameters $\bar{w} = w_0, \dots, w_m$ and a set E of examples, the **sum-of-squares error** is

$$Error_E(\bar{w}) = \sum_{e \in E} (p_e^{\bar{w}} - o_e)^2,$$

- $p_e^{\bar{w}}$ is the predicted output by a neural network with parameter values given by \bar{w} for example e
- o_e is the observed output for example e .
- The aim of neural network learning is, given a set of examples, to find parameter settings that minimize the error.

Neural Network Learning

- Aim of neural network learning: given a set of examples, find parameter settings that minimize the error.
- **Back-propagation learning** is gradient descent search through the parameter space to minimize the sum-of-squares error.

Backpropagation Learning



Inputs:

- A network, including all units and their connections
- Stopping Criteria
- Learning Rate (constant of proportionality of gradient descent search)
- Initial values for the parameters
- A set of classified training data



Output: Updated values for the parameters

Backpropagation Learning Algorithm

- Repeat
 - evaluate the network on each example given the current parameter settings
 - determine the derivative of the error for each parameter
 - change each parameter in proportion to its derivative
- until the stopping criteria is met

Gradient Descent for Neural Net Learning

- At each iteration, update parameter w_i

$$w_i \leftarrow \left(w_i - \eta \frac{\partial error(p_i)}{\partial p_i} \right)$$

η is the learning rate

- You can compute partial derivative:

- numerically: for small Δ

$$\frac{error(p_i + \Delta) - error(p_i)}{\Delta}$$

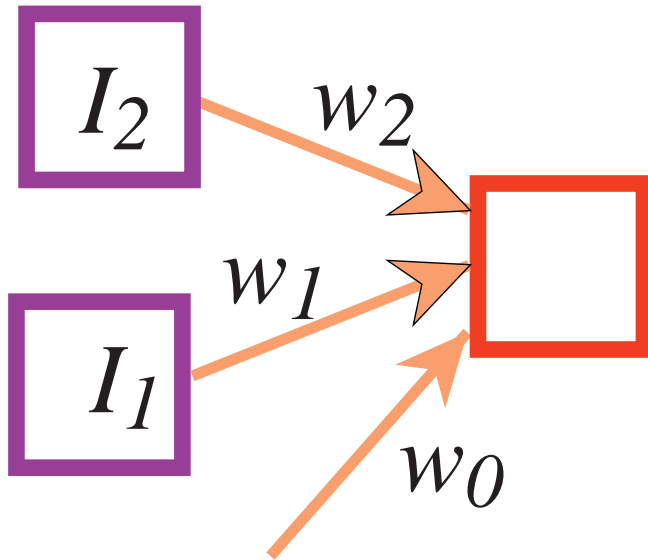
- analytically: $f'(x) = f(x)(1 - f(x))$ + chain rule

Simulation of Neural Net Learning

Parameter	iteration 0		iteration 1	iteration 80
	Value	Deriv	Value	Value
w_0	0.2	0.768	-0.18	-2.98
w_1	0.12	0.373	-0.07	6.88
w_2	0.112	0.425	-0.10	-2.10
w_3	0.22	0.0262	0.21	-5.25
w_4	0.23	0.0179	0.22	1.98
Error:	4.6121		4.6128	0.178



What Can a Neural Network Represent?



w_0	w_1	w_2	Logic
-15	10	10	and
-5	10	10	or
5	-10	-10	nor

Output is $f(w_0 + w_1 \times I_1 + w_2 \times I_2)$.

A single unit can't represent *xor*.



Bias in neural networks and decision trees

- It's easy for a neural network to represent “at least two of I_1, \dots, I_k are true”:

$$\begin{array}{cccc} w_0 & w_1 & \cdots & w_k \\ \hline -15 & 10 & \cdots & 10 \end{array}$$

This concept forms a large decision tree.

- Consider representing a conditional: “If c then a else b ”:
 - Simple in a decision tree.
 - Needs a complicated neural network to represent $(c \wedge a) \vee (\neg c \wedge b)$.



Neural Networks and Logic

- Meaning is attached to the input and output units.
- There is no a priori meaning associated with the hidden units.
- What the hidden units actually represent is something that's learned.

