

Neural Networks

July 6, 2006

CS 486/686

University of Waterloo

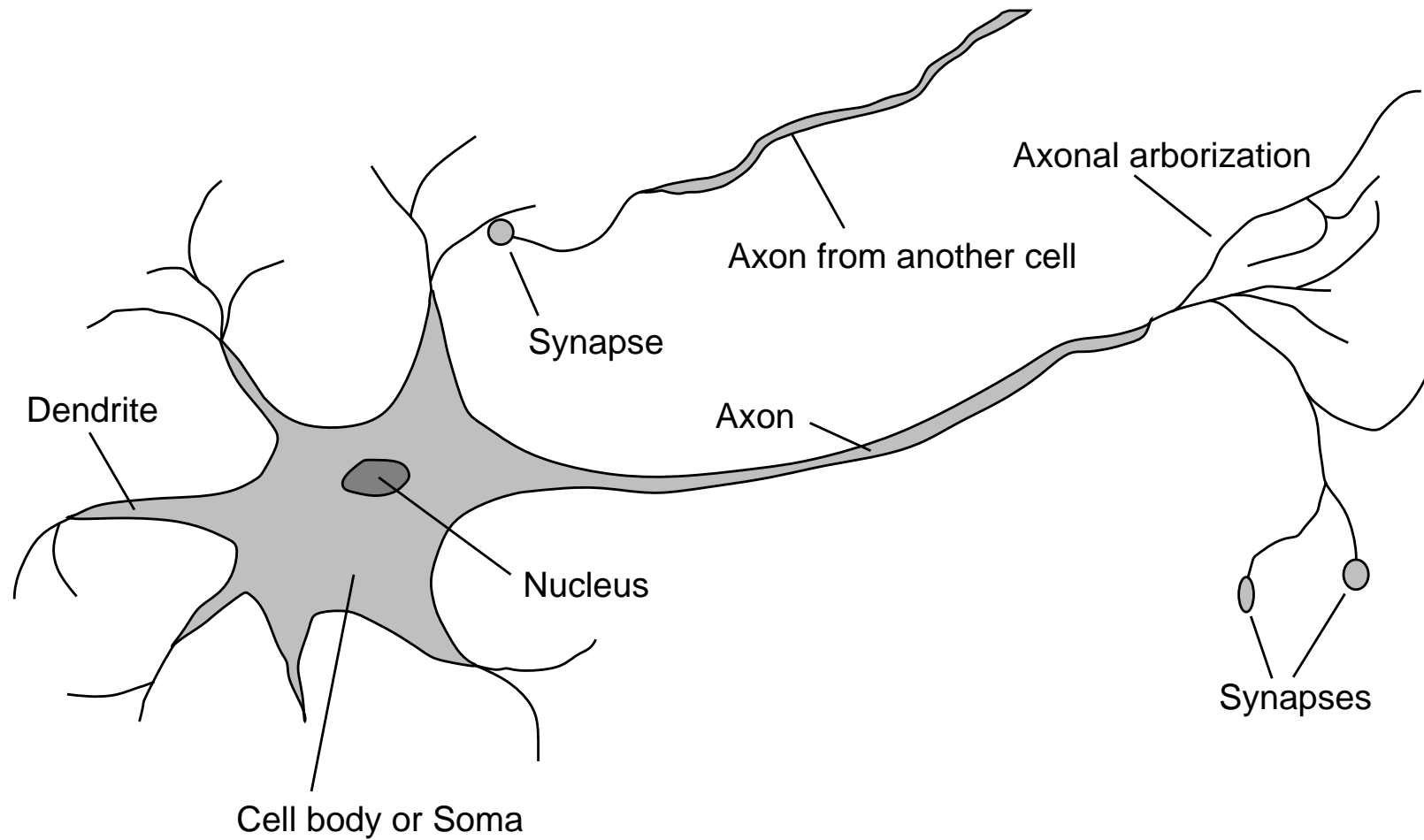
Outline

- Neural networks
 - Perceptron
 - Supervised learning algorithms for neural networks
- Reading: R&N Ch 20.5

Brain

- Seat of human intelligence
- Where memory/knowledge resides
- Responsible for thoughts and decisions
- Can learn
- Consists of nerve cells called **neurons**

Neuron



Comparison

- Brain
 - Network of neurons
 - Nerve signals propagate in a neural network
 - Parallel computation
 - Robust (neurons die everyday without any impact)
- Computer
 - Bunch of gates
 - Electrical signals directed by gates
 - Sequential computation
 - Fragile (if a gate stops working, computer crashes)

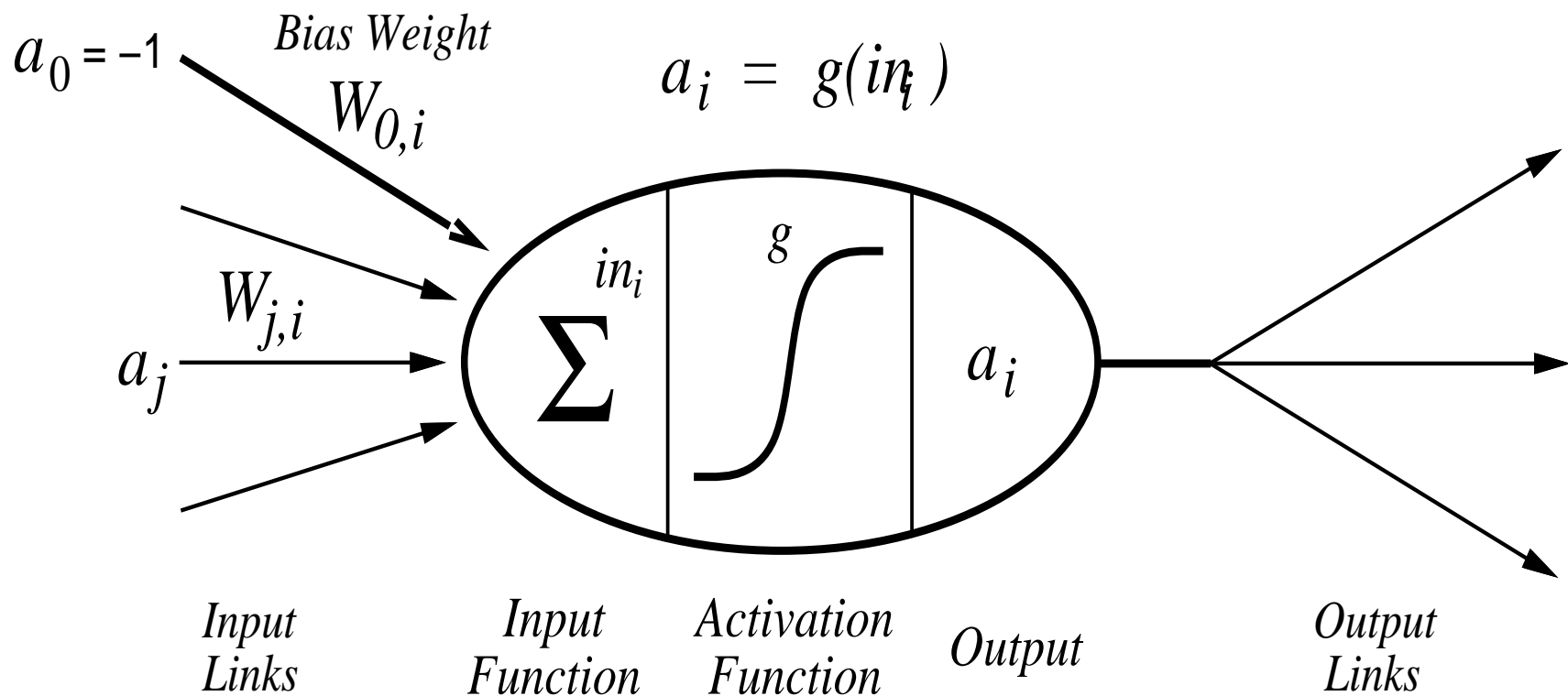
Artificial Neural Networks

- Idea: **mimic the brain to do computation**
- Artificial neural network:
 - Nodes (a.k.a units) correspond to neurons
 - Links correspond to synapses
- Computation:
 - Numerical signal transmitted between nodes corresponds to chemical signals between neurons
 - Nodes modifying numerical signal corresponds to neurons firing rate

ANN Unit

- For each unit i :
- **Weights: W_{ji}**
 - Strength of the link from unit j to unit i
 - Input signals a_j weighted by W_{ji} and linearly combined: $in_i = \sum_j W_{ji} a_j$
- **Activation function: g**
 - Numerical signal produced: $a_i = g(in_i)$

ANN Unit

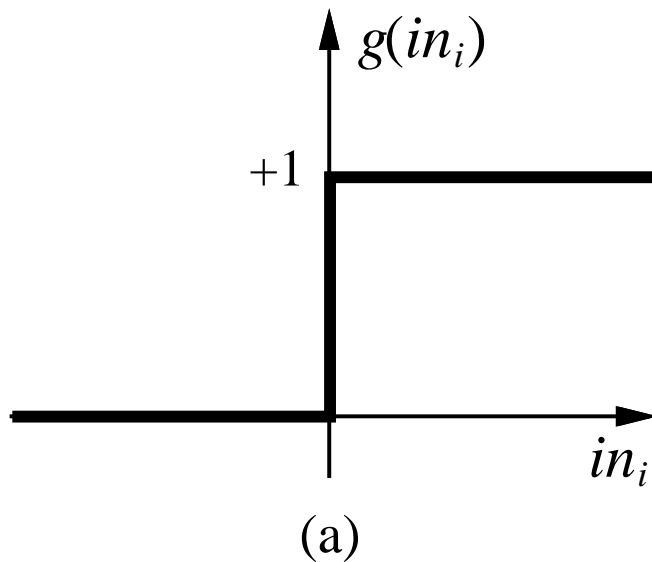


Activation Function

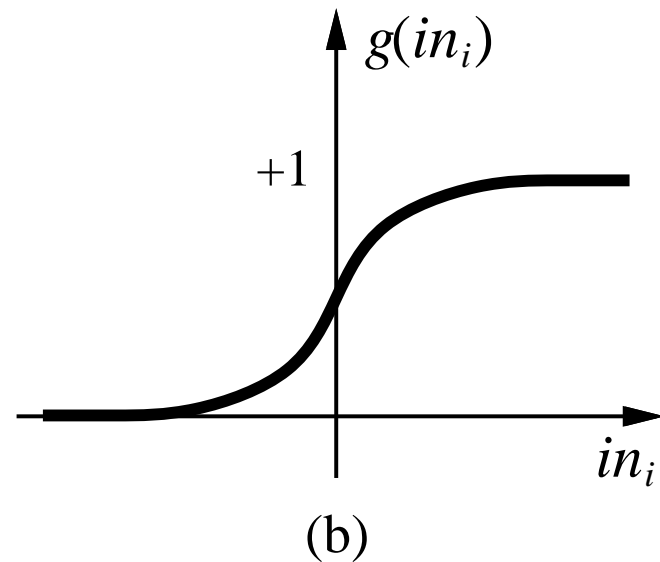
- Should be nonlinear
 - Otherwise network is just a linear function
- Often chosen to mimic firing in neurons
 - Unit should be "active" (output near 1) when fed with the "right" inputs
 - Unit should be "inactive" (output near 0) when fed with the "wrong" inputs

Common Activation Functions

Threshold



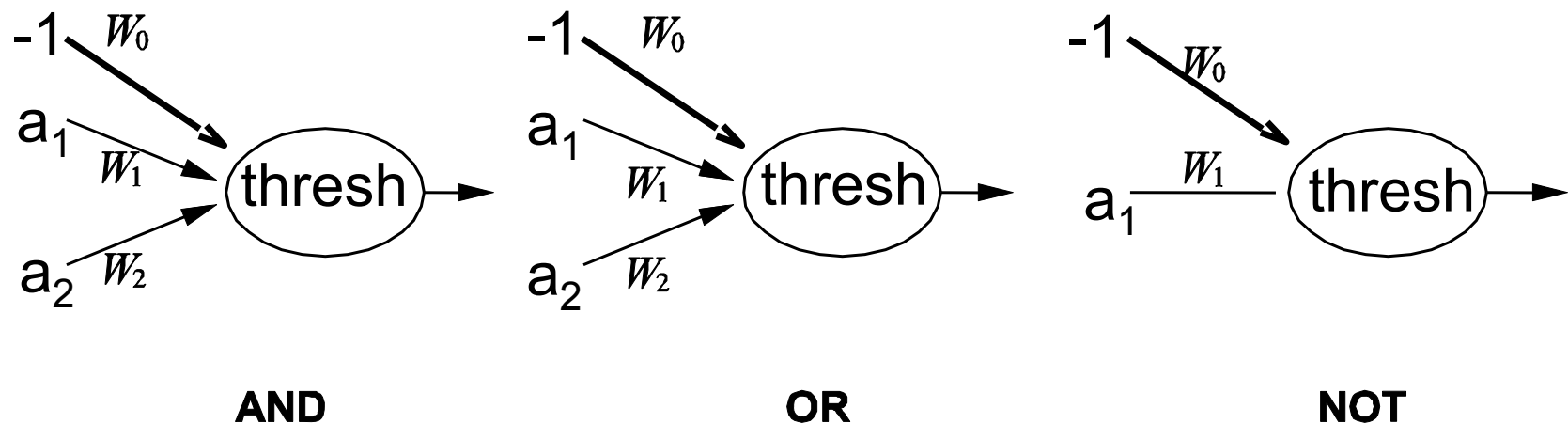
Sigmoid



$$g(x) = 1/(1+e^{-x})$$

Logic Gates

- McCulloch and Pitts (1943)
 - Design ANNs to represent Boolean fns
- What should be the weights of the following units to code AND, OR, NOT?

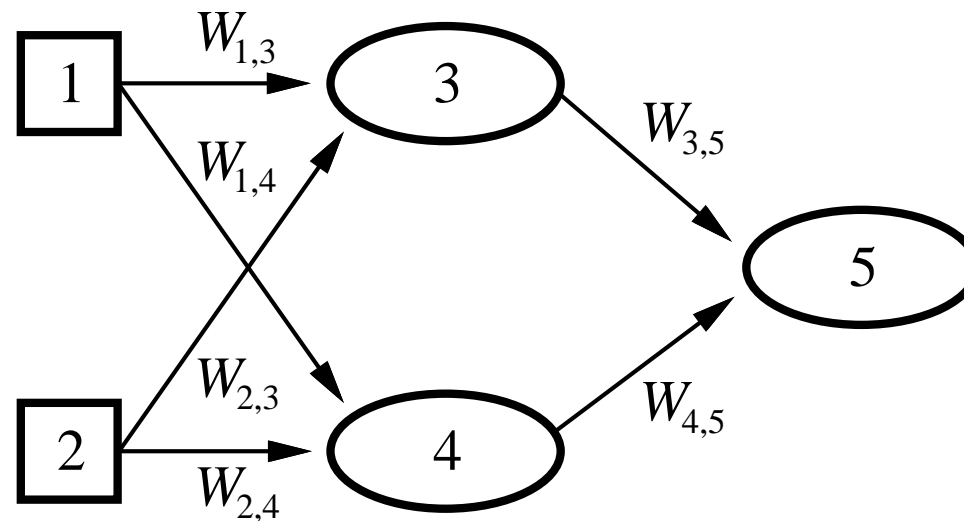


Network Structures

- Feed-forward network
 - Directed **acyclic** graph
 - No internal state
 - Simply computes outputs from inputs
- Recurrent network
 - Directed **cyclic** graph
 - Dynamical system with internal states
 - Can memorize information

Feed-forward network

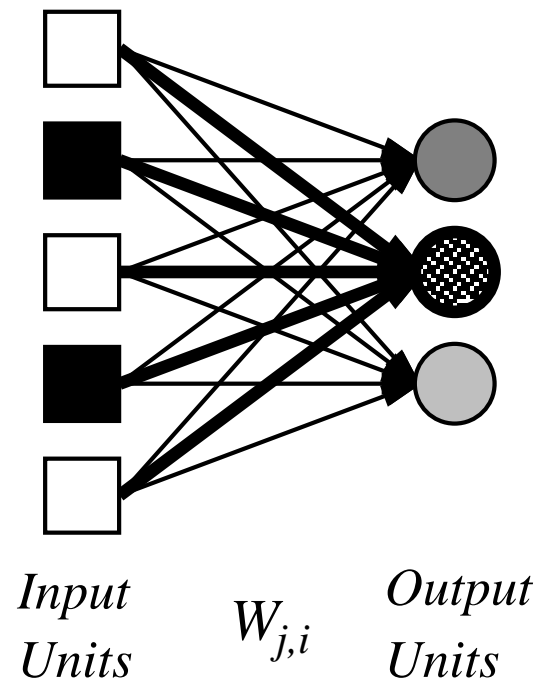
- Simple network with two inputs, one hidden layer of two units, one output unit



$$\begin{aligned} a_5 &= g(W_{3,5}a_3 + W_{4,5}a_4) \\ &= g(W_{3,5}g(W_{1,3}a_1 + W_{2,3}a_2) + W_{4,5}g(W_{1,4}a_1 + W_{2,4}a_2)) \end{aligned}$$

Perceptron

- Single layer feed-forward network



Supervised Learning

- Given list of <input,output> pairs
- Train feed-forward ANN
 - To compute proper outputs when fed with inputs
 - Consists of adjusting weights W_{ji}
- Simple learning algorithm for threshold perceptrons

Threshold Perceptron Learning

- Learning is done separately for each unit
 - Since units do not share weights
- Perceptron learning for unit i :
 - For each $\langle \text{inputs}, \text{output} \rangle$ pair do:
 - Case 1: correct output produced
 - $\forall_j W_{ji} \leftarrow W_{ji}$
 - Case 2: output produced is 0 instead of 1
 - $\forall_j W_{ji} \leftarrow W_{ji} + a_j$
 - Case 3: output produced is 1 instead of 0
 - $\forall_j W_{ji} \leftarrow W_{ji} - a_j$
 - Until correct output for all training instances

Threshold Perceptron Learning

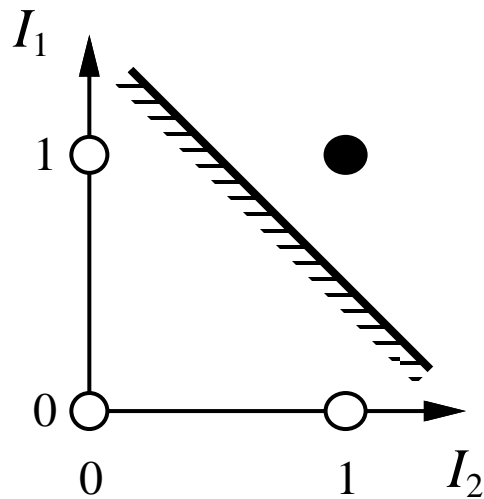
- Dot products: $a \bullet a \geq 0$ and $-a \bullet a \leq 0$
- Perceptron computes
 - 1 when $a \bullet W = \sum_j a_j W_{ji} > 0$
 - 0 when $a \bullet W = \sum_j a_j W_{ji} < 0$
- If output should be 1 instead of 0 then
 - $W \leftarrow W + a$ since $a \bullet (W + a) \geq a \bullet W$
- If output should be 0 instead of 1 then
 - $W \leftarrow W - a$ since $a \bullet (W - a) \leq a \bullet W$

Threshold Perceptron Hypothesis Space

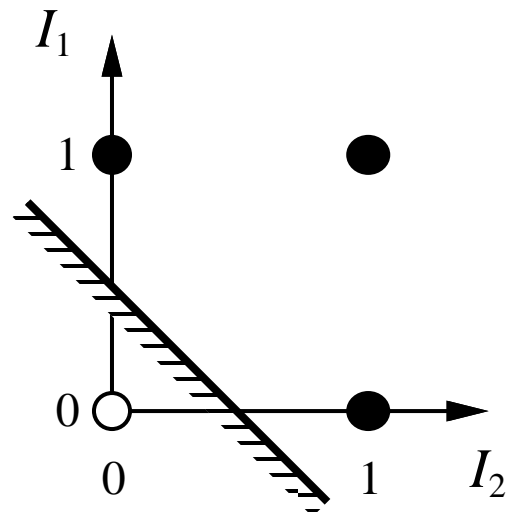
- Hypothesis space h_W :
 - All binary classifications with param. W s.t.
 - $a \bullet W > 0 \rightarrow 1$
 - $a \bullet W < 0 \rightarrow 0$
- Since $a \bullet W$ is linear in W , perceptron is called a **linear separator**

Threshold Perceptron Hypothesis Space

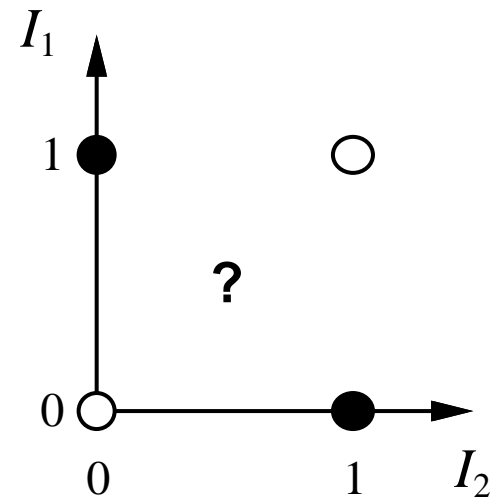
- Are all Boolean gates linearly separable?



(a) I_1 **and** I_2



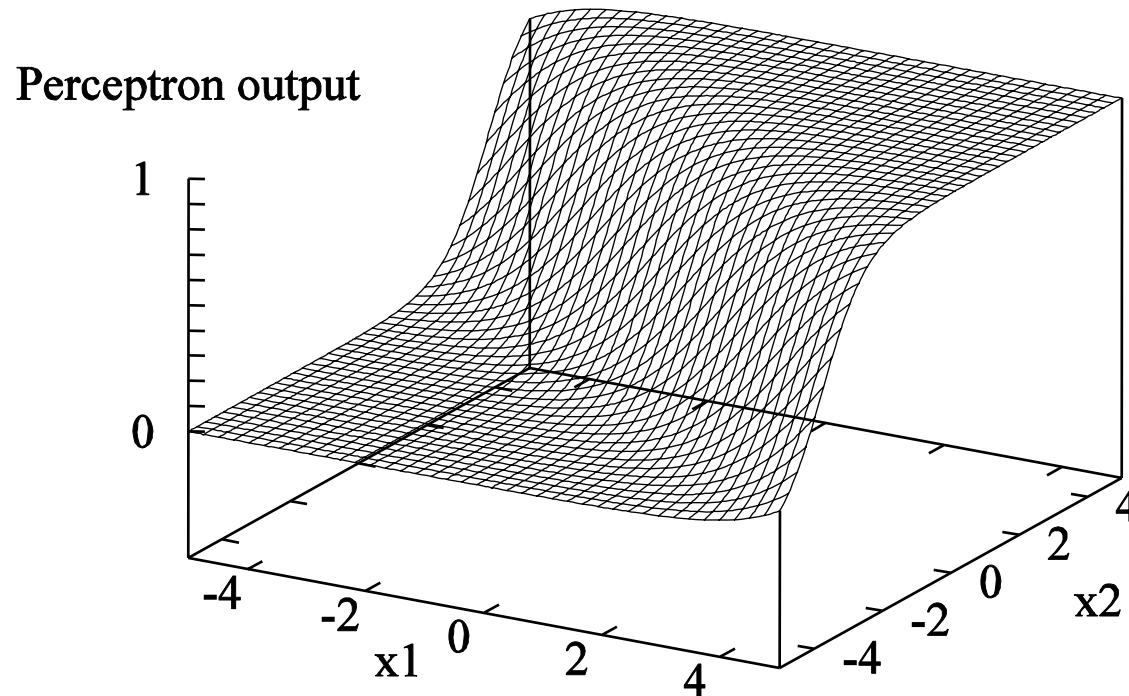
(b) I_1 **or** I_2



(c) I_1 **xor** I_2

Sigmoid Perceptron

- Represent “soft” linear separators



Sigmoid Perceptron Learning

- Formulate learning as an optimization search in weight space
 - Since g differentiable, use gradient descent
- Minimize squared error:
 - $E = 0.5 \text{Err}^2 = 0.5 (y - h_w(\mathbf{x}))^2$
 - \mathbf{x} : input
 - y : target output
 - $h_w(\mathbf{x})$: computed output

Perceptron Error Gradient

- $E = 0.5 \text{ Err}^2 = 0.5 (y - h_w(\mathbf{x}))^2$
- $$\begin{aligned}\partial E / \partial W_j &= \text{Err} \times \partial \text{Err} / \partial W_j \\ &= \text{Err} \times \partial (y - g(\sum_j W_j x_j)) \\ &= -\text{Err} \times g'(\sum_j W_j x_j) \times x_j\end{aligned}$$
- When g is sigmoid fn, then $g' = g(1-g)$

Perceptron Learning Algorithm

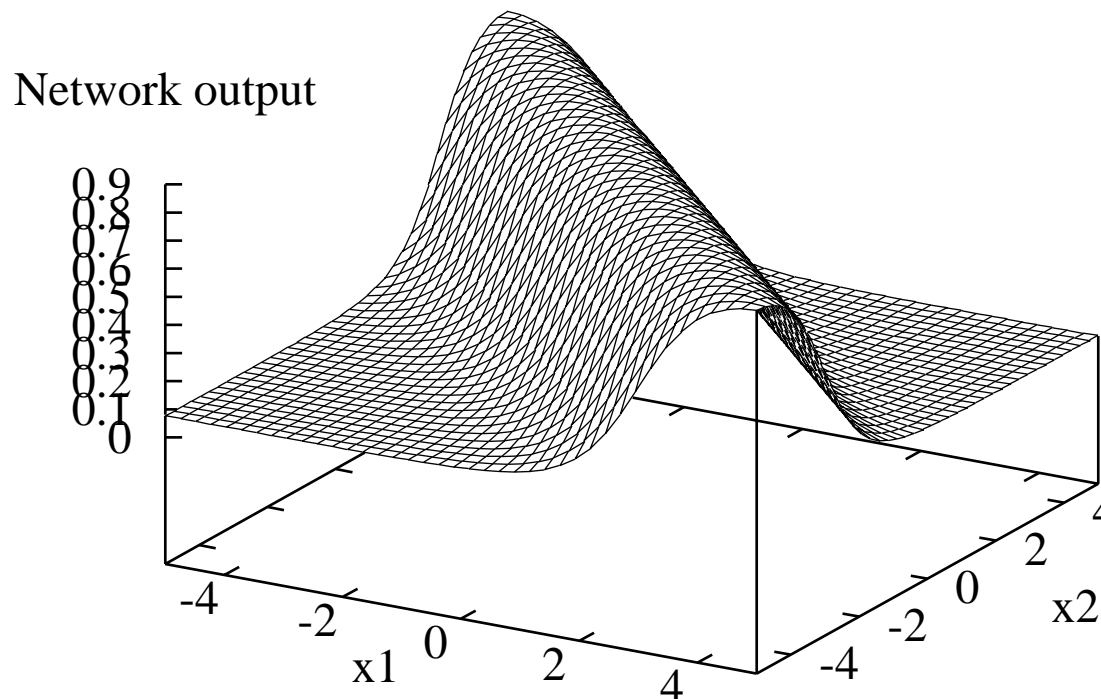
- `Perceptron-Learning(examples, network)`
 - Repeat
 - For each e in examples do
 - $in \leftarrow \sum_j W_j x_j[e]$
 - $Err \leftarrow y[e] - g(in)$
 - $W_j \leftarrow W_j + \alpha \times Err \times g'(in) \times x_j[e]$
 - Until some stopping criteria satisfied
 - Return learnt network
- N.B. α is a learning rate corresponding to the step size in gradient descent

Multilayer Feed-forward Neural Networks

- Perceptron can only represent (soft) linear separators
 - Because single layer
- With multiple layers, what fns can be represented?
 - Virtually any function!

Multilayer Networks

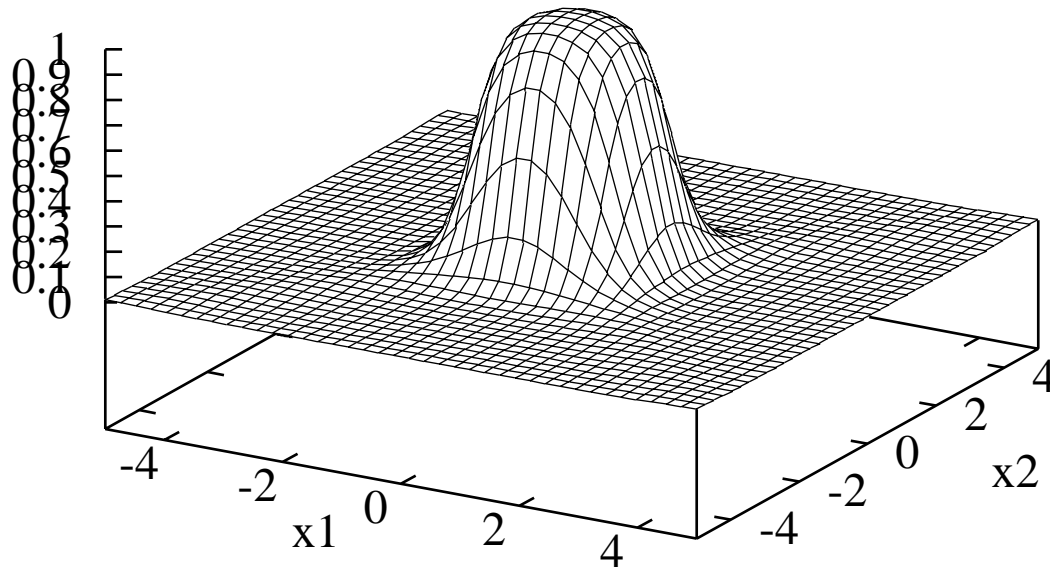
- Adding two sigmoid units with parallel but opposite "cliffs" produces a ridge



Multilayer Networks

- Adding two intersecting ridges (and thresholding) produces a bump

Network output



Multilayer Networks

- By tiling bumps of various heights together, we can approximate any function
- Training algorithm:
 - Back-propagation
 - Essentially gradient performed by propagating errors backward into the network
 - See textbook for derivation

Neural Net Applications

- Neural nets can approximate any function, hence 1000's of applications
 - NETtalk for pronouncing English text
 - Character recognition
 - Paint-quality inspection
 - Vision-based autonomous driving
 - Etc.

Neural Net Drawbacks

- Common problems:
 - How should we interpret units?
 - How many layers and units should a network have?
 - How to avoid local optimum while training with gradient descent?

Next Class

- Next Class:
 - Ensemble learning
 - Russell and Norvig Sect. 18.4