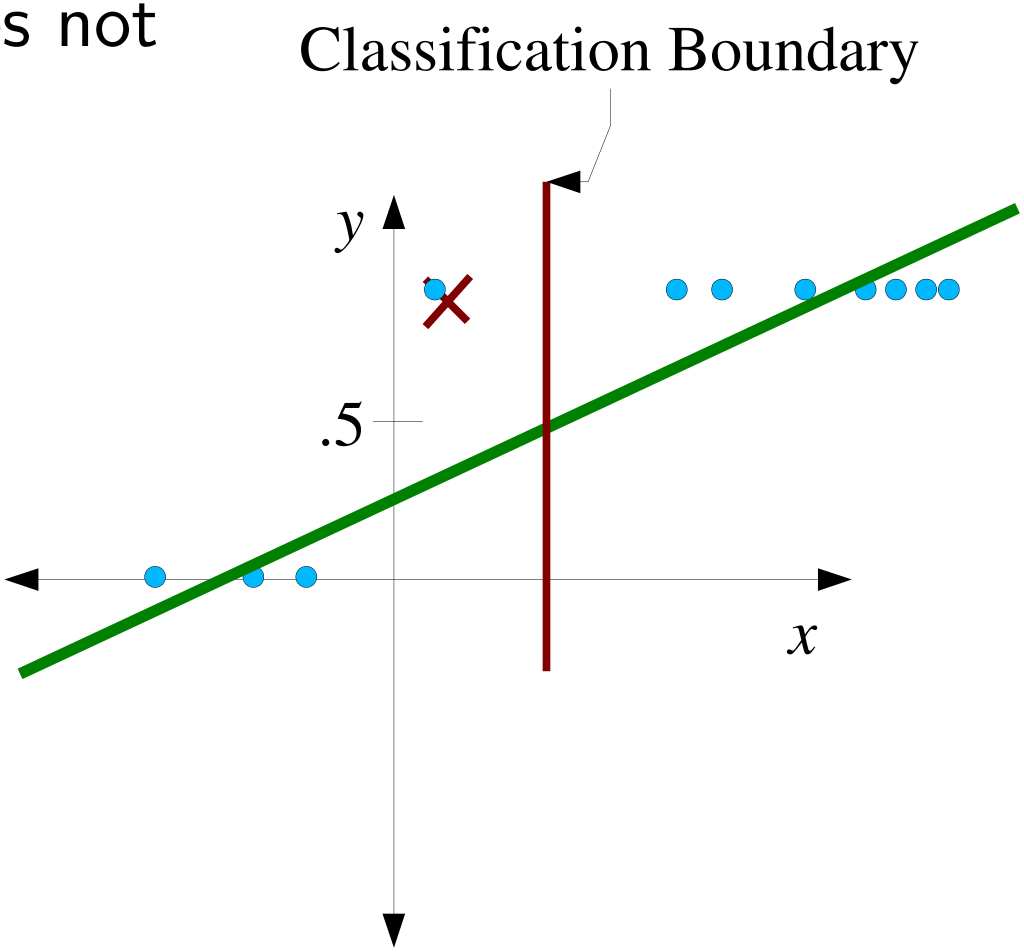
Single-Layer Logistic Network

Regression vs. Classification

- Now we have the machinery to fit a line (plane, hyperplane) to a set of data points - regression.
- What about classification?
- First thought:
  - For each data point $\boldsymbol{x}$, set the value of y to be 0 or 1, depending on the class
  - Use linear regression to fit the data.
  - During classification assume class 0 if $y < .5$, assume class 1 if $y >= .5$.

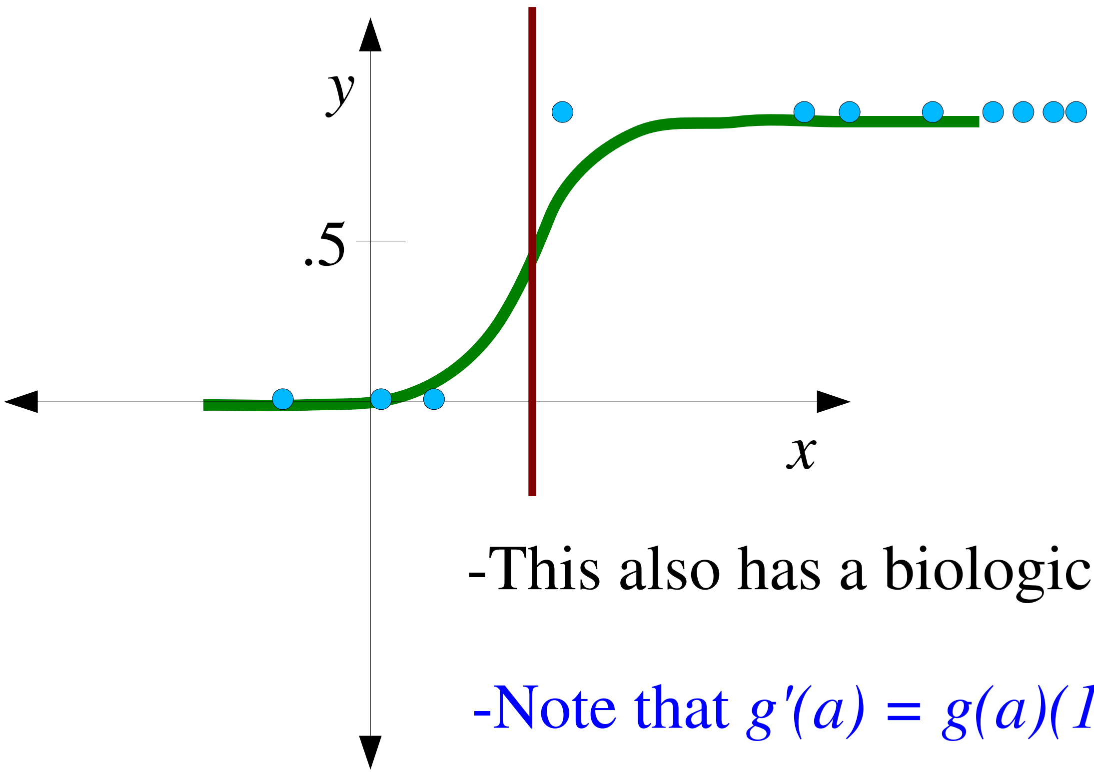
Classification Example

- The least squares fit does not necessarily lead to good classification.



Apply a Sigmoid to the Output

- Let's apply a squashing function to the output of the network: $h(x) = g(\boldsymbol{w}^T \boldsymbol{x})$, where $g(a) = \frac{1}{(1+e^{-a})}$



-This also has a biological motivation

-Note that $g'(a) = g(a)(1-g(a))$

The New Update Rule...

- The partial derivative (for a particular example):

$$Error(w) = \frac{1}{2}\left(y - g(\boldsymbol{w}^T \boldsymbol{x})\right)^2$$

$$\frac{\partial\, Error(w)}{\partial w_i} = \left(y - g(\boldsymbol{w}^T \boldsymbol{x})\right) \frac{\partial}{\partial w_i}\left(\left(y - g(\boldsymbol{w}^T \boldsymbol{x})\right)\right)$$

$$= -\left(y - g(\boldsymbol{w}^T \boldsymbol{x})\right) g'(\boldsymbol{w}^T \boldsymbol{x})\, x_i$$

- The new update rule: $w_i \leftarrow w_i + \eta\left(y - g(\boldsymbol{w}^T \boldsymbol{x})\right) g'(\boldsymbol{w}^T \boldsymbol{x})\, x_i$
- Vector version: $\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta\left(y - g(\boldsymbol{w}^T \boldsymbol{x})\right) g'(\boldsymbol{w}^T \boldsymbol{x})\, \boldsymbol{x}$

(This is a version of “logistic regression” a classical technique from statistics.)

Perceptrons

- Late 50's to mid 60's – Rosenblatt's Perceptrons
  (Original paper: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Psychological Review, 65:386-408)

- Original perceptron formulation used a threshold instead of a sigmoid:

$$g(a) = \begin{cases} 1 \; if \; a > 0 \\ 0 \; if \; a \leq 0 \end{cases}$$

- Learning rule: $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha\left(t - g\left(\boldsymbol{w}^{T} \boldsymbol{x}\right)\right)\boldsymbol{x}$

The Rise and Fall of Perceptrons

- 1969 – Minsky and Papert write Perceptrons.
  - Pretty much kills off neural network research.

The Problem...

- The perceptron (any single layer neural network) only works if the classes are linearly separable.
- XOR is a problem:

A	B	OUT
0	0	0
0	1	1
1	0	1
1	1	0