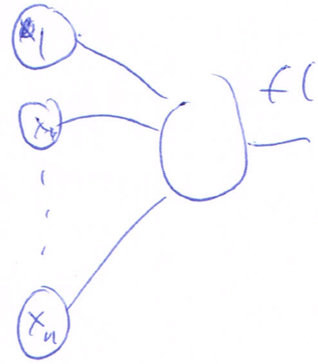
April 11

Perceptron

features = inputs
attributes



$$f(1, x_1, \cdots, x_n) = \begin{cases} 1 & \text{if } \sum_{i=0}^{n} w_i x_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Perceptron learning rule

Current hypothesis: $h_{\bar{w}}(x)$

Initially $w_0 = w_1 = \cdots = w_n = 0$

(more generally set to random weights)

Repeat (randomly reorder the data)

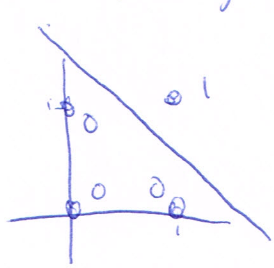
For $i = 1$ to n <for each example>

For $j = 1$ to n <for each feature>

$$w_j \leftarrow w_j + \alpha \bar{x}_j^i (y_j^i - h_{\bar{w}}(\bar{x}^i))$$

Until $h_{\bar{w}}(\bar{x})$ gets all data correct

Example And gate



	x_0	x_1	x_2	$f(x_0, x_1, x_2)$
1	1	0	0	0
2	1	0	1	0
3	1	1	0	0
4	1	1	1	1

Y

$$w_j \leftarrow w_j + 0.3\, x_j^i \left(y^i - h_{\bar{w}}(\bar{x}^i)\right)$$

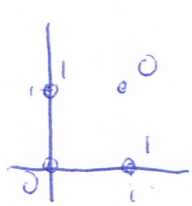
Example	w_0	w_1	w_2	
	0	0	0	
1	-.3	0	0	
2	-.3	0	0	no change
3	-.3	0	0	no change
4	0	.3	.3	
1	-.3	.3	.3	
2	-.6	.3	0	
3	-.6	.3	0	no change
4	-.3	.6	.3	
1	-.3	.6	.3	no change
2	-.6	.6	0	
3	-.9	.3	0	
4	-.6	.6	.3	~~Success~~

⋮

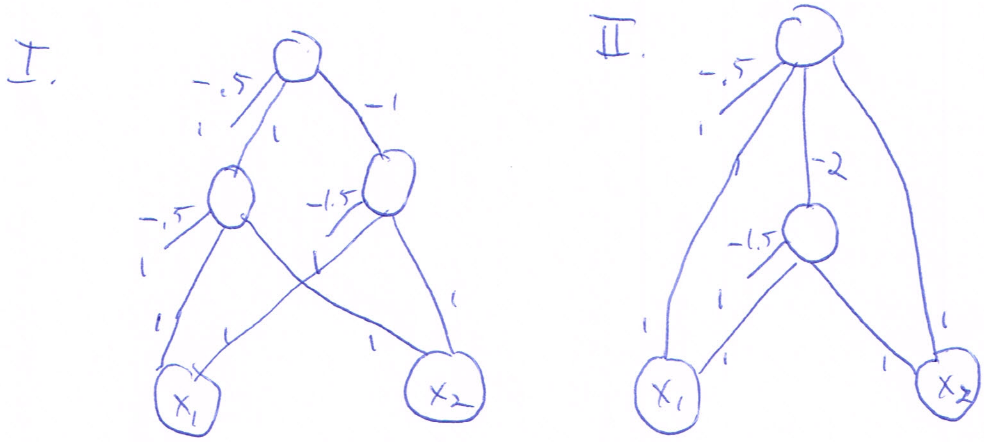
Perceptron learning rule: Convergence

If a set of training data are linearly separable then the perceptron learning rule algorithm will stop after a finite number of steps with a linear surface that correctly separates the data. (more in next lecture)

Linear Separability: XOR - fail!



Multilayer neural network for XOR



Perceptron problems

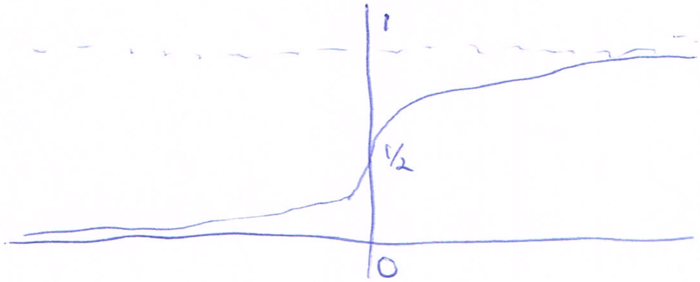
1. Problem: data that are not linearly separable
   Solution: use multilayer network ✓
2. Problem: Unclear how to update weights for multilayer perceptrons
   Solution: Use differentiable function that approximates a hard threshold

Old: $f(x_0, x_1, \cdots x_n) = 1$ if $\sum_{i=1}^{n} w_i x_i \geq 0$

0 otherwise

New: $$f(x_0, x_1, \cdots x_n) = \frac{1}{1 + e^{-\sum_{i=1}^{n} w_i x_i}}$$

"Logistic Function"



Idea behind backpropagation:

Consider error on data:

$$\sum_{i=1}^{m} \left(f(\bar{x}^i) - h_{\bar{w}}(\bar{x}^i) \right)^2$$

"L_2 norm"