

Perceptron

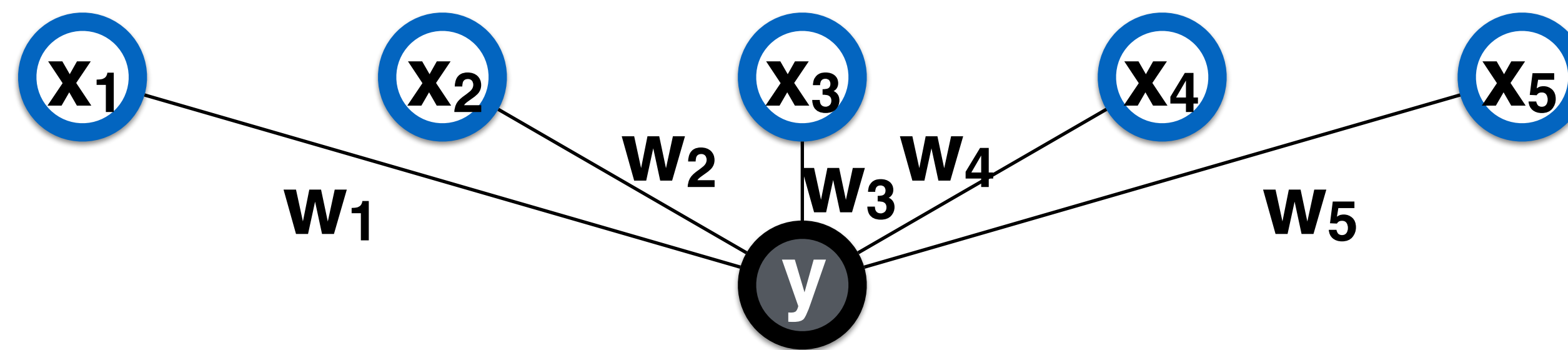
Machine Learning
CS5824/ECE5424
Bert Huang
Virginia Tech

Plan

- Perceptron
- Multi-class perceptron

Perceptron

- AKA single-layer neural network



- linear classifier

$$f(x) = \text{sign} \left(\sum_i w_i x_i \right) = \begin{cases} +1 & \text{if } \sum_i w_i x_i \geq 0 \\ -1 & \text{if } \sum_i w_i x_i < 0 \end{cases}$$

- online learner

Online Training

- Get example \mathbf{x} and oracle label \mathbf{y}
- Try current classifier $\mathbf{f}(\mathbf{x})$ $\mathbf{f}(\mathbf{x}) = \text{sign}(\mathbf{w}_1\mathbf{x}_1 + \mathbf{w}_2\mathbf{x}_2 + \mathbf{w}_3\mathbf{x}_4 + \dots)$
- If $\mathbf{f}(\mathbf{x}) = \mathbf{y}$, celebrate if $\mathbf{y} = +1$ and $\mathbf{f}(\mathbf{x}) = -1$, $\mathbf{w} \cdot \mathbf{x}$ is too small
- If $\mathbf{f}(\mathbf{x}) \neq \mathbf{y}$, fix it if $\mathbf{y} = -1$ and $\mathbf{f}(\mathbf{x}) = +1$, $\mathbf{w} \cdot \mathbf{x}$ is too big

How to Fix Your Broken Perceptron

~~$$w \leftarrow w - x \text{ if } f(x) = +1 \text{ and } y = -1$$~~

~~$$w \leftarrow w + x \text{ if } f(x) = -1 \text{ and } y = +1$$~~

$$w \leftarrow w \text{ if } f(x) = y$$

$$w \leftarrow w + yx \text{ if } f(x) \neq y$$

$$f(x) = \text{sign}(w_1 x_1 + w_2 x_2) = +1$$

Example:

$$w = [2, 1], \quad x = [3, 1], \quad y = -1$$

$$f(x) = \text{sign}(2(3) + 1(1))$$

Update:

$$W_1 = 2 - 3$$

$$W_2 = 1 - 1$$

$$w = [-1, 0]$$

$$\text{New } f(x) = -1(3) + 0(1)$$

How to Fix Your Broken Perceptron

~~$w \leftarrow w - x$ if $f(x) = +1$ and $y = -1$~~

~~$w \leftarrow w + x$ if $f(x) = -1$ and $y = +1$~~

$w \leftarrow w$ if $f(x) = y$

$w \leftarrow w + yx$ if $f(x) \neq y$

$y = +1, f(x) = -1$

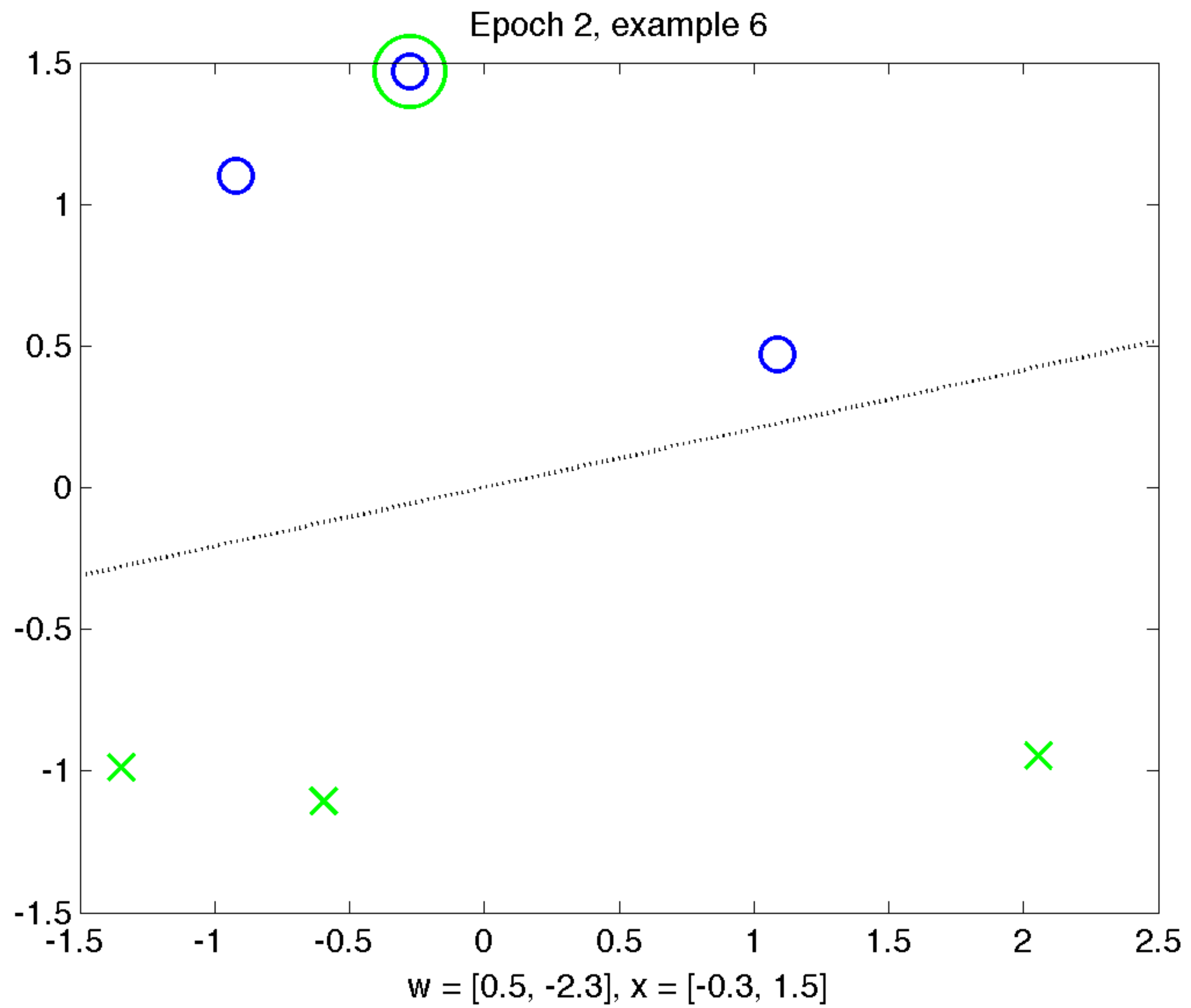
$$w \cdot x \leq (w + x) \cdot x$$

$$w \cdot x + x \cdot x$$

$y = -1, f(x) = +1$

$$w \cdot x \geq (w - x) \cdot x$$

$$w \cdot x - x \cdot x$$



Binary Perceptron Learning

$$\text{sign}(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$$

bias

$$\mathbf{f}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x})$$

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \mathbf{y} \mathbf{x} \text{ if } \mathbf{f}(\mathbf{x}) \neq \mathbf{y}$$

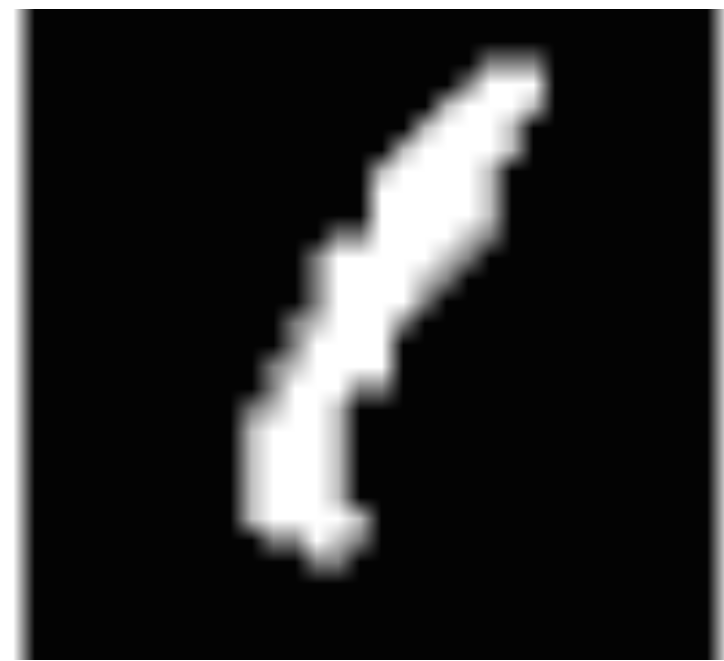
α = learning rate

$$\mathbf{b} \leftarrow \mathbf{b} + \alpha \mathbf{y} \text{ if } \mathbf{f}(\mathbf{x}) \neq \mathbf{y}$$

Multiclass Output



-1



+1

Multiclass Output



Multiclass Perceptron

- Separate weights for each class: $\mathbf{w}^0, \mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3, \mathbf{w}^4, \mathbf{w}^5, \mathbf{w}^6, \dots$
- Predict according to maximum scoring class:

$$\mathbf{f}(\mathbf{x}) = \operatorname{argmax}_{y'} \mathbf{w}^{y'} \cdot \mathbf{x}$$

- Learning update:

$$\begin{aligned} \mathbf{w}^y &\leftarrow \mathbf{w}^y + \mathbf{x} \\ \mathbf{w}^{\mathbf{f}(\mathbf{x})} &\leftarrow \mathbf{w}^{\mathbf{f}(\mathbf{x})} - \mathbf{x} \end{aligned}$$

Multiclass Perceptron

- Learning update:

$$\mathbf{w}^y \leftarrow \mathbf{w}^y + \mathbf{x}$$

$$\mathbf{w}^{f(\mathbf{x})} \leftarrow \mathbf{w}^{f(\mathbf{x})} - \mathbf{x}$$

\mathbf{x}



$$y = 2$$

$$f(\mathbf{x}) = 6$$

$$\mathbf{w}^2 \leftarrow \mathbf{w}^2 + \mathbf{x} \quad (\text{make } \mathbf{w}^2 \cdot \mathbf{x} \text{ greater})$$

$$\mathbf{w}^6 \leftarrow \mathbf{w}^6 - \mathbf{x} \quad (\text{make } \mathbf{w}^6 \cdot \mathbf{x} \text{ lesser})$$

Multiclass Perceptron

- Learning update:

$$\mathbf{w}_y \leftarrow \mathbf{w}_y + \mathbf{x}$$

$$\mathbf{w}_{f(\mathbf{x})} \leftarrow \mathbf{w}_{f(\mathbf{x})} - \mathbf{x}$$

\mathbf{x}



$$y = 2$$

$$f(\mathbf{x}) = 2$$

$$\mathbf{w}^2 \leftarrow \mathbf{w}^2 + \mathbf{x}$$

$$\mathbf{w}^2 \leftarrow \mathbf{w}^2 - \mathbf{x}$$

Multiclass Perceptron

- Learning update:

$$\mathbf{w}_y \leftarrow \mathbf{w}_y + \mathbf{x}$$

$$\mathbf{w}_{f(\mathbf{x})} \leftarrow \mathbf{w}_{f(\mathbf{x})} - \mathbf{x}$$

\mathbf{x}



$$y = 2$$

$$f(\mathbf{x}) = 2$$

$$\mathbf{w}^2 \leftarrow \mathbf{w}^2 + \mathbf{x} - \mathbf{x} = \mathbf{w}^2$$

Perceptron Summary

- Linear classifier multiplies **weights** by input **features**
- Learn by updating when wrong
 - If score too low, make score higher
 - If score too high, make score lower
- Multiclass: use **multiple weight vectors**, choose max scoring
 - Learn by adjusting score of predicted class and true class