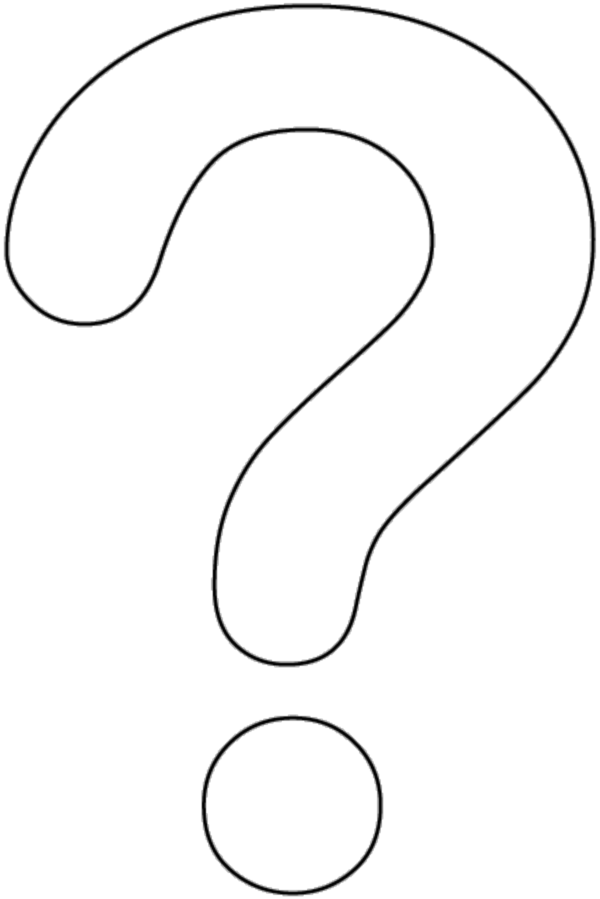
IOWA STATE UNIVERSITY

Department of Computer Science

COM S 573: Machine Learning

Lecture 5: Linear Perceptron

Linear Regression: Assumptions



Linear Regression: Assumptions

- Data are linear

$$\hat{y}_i = \boldsymbol{w}^T \boldsymbol{x}_i$$

- Residuals are Independent and identically distributed

$$p(r_i, r_j) = p(r_i)p(r_j)$$

- Each residual follows normal distribution

$$r_i \sim \mathcal{N}(0, \sigma^2)$$

Linear Models

- Linear regression: predict a scalar
  - House price
  - Weight of a planet
- Linear perceptron: classifier
  - Predict an animal is a dog or not
  - Predict an image contains a square or not
- Logistic regression: classifier based on a probability
  - Predict how likely a team win
  - Predict how likely tomorrow is sunny

Linear Perceptron: Example

- Credit approval or denial
  - Task: Approve or deny credit (binary)
  - Features: Salary, debt, years in residence, etc.

Linear Perceptron: Representation

- Input: $\boldsymbol{x} = [x_1, x_2, \cdots, x_d]^T$
- Give different weights to different features

$$\text{credit score} = \sum_{i=1}^{d} w_i x_i$$

- Approve if the credit score is larger than threshold

$$\text{Approve credit if } \sum_{i=1}^{d} w_i x_i > \text{threshold}$$

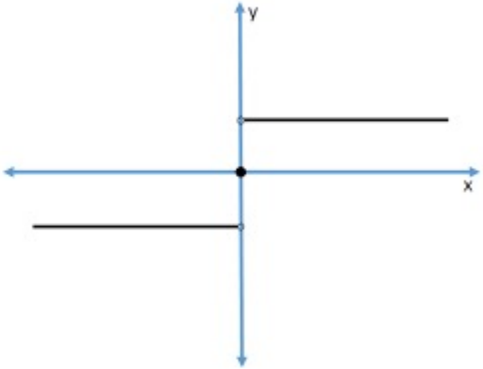
$$\text{Deny credit if } \sum_{i=1}^{d} w_i x_i < \text{threshold}$$

Linear Perceptron: Representation

- Approve if the credit score is larger than threshold

$$\text{Approve credit if } \sum_{i=1}^{d} w_i x_i > \text{threshold}$$

$$\text{Deny credit if } \sum_{i=1}^{d} w_i x_i < \text{threshold}$$



- Can be rewritten as

$$h(\boldsymbol{x}) = \text{sign}((\sum_{i=1}^{d} w_i x_i) + w_0)$$

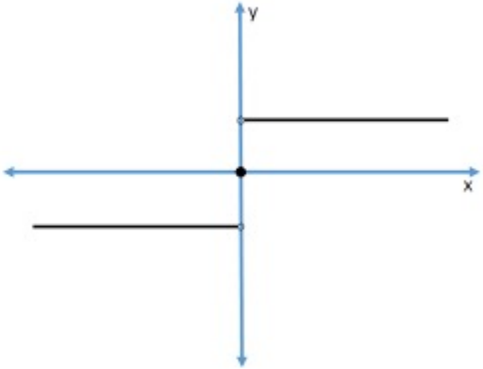
What is w0?

Linear Perceptron: Representation

- Approve if the credit score is larger than threshold

$$\text{Approve credit if } \sum_{i=1}^{d} w_i x_i > \text{threshold}$$

$$\text{Deny credit if } \sum_{i=1}^{d} w_i x_i < \text{threshold}$$



- Can be rewritten as

$$h(\boldsymbol{x}) = \text{sign}((\sum_{i=1}^{d} w_i x_i) + w_0)$$

- The bias corresponds to the threshold $w_0 = -\text{threshold}$

Linear Perceptron: Representation

- Input: $\boldsymbol{x} = [x_1, x_2, \cdots, x_d]^T$
- We want to learn a set of weights

$$h(\boldsymbol{x}) = \text{sign}((\sum_{i=1}^{d} w_i x_i) + w_0)$$

$$\boldsymbol{w} = [w_0, w_1, w_2, \cdots, w_d]^T$$

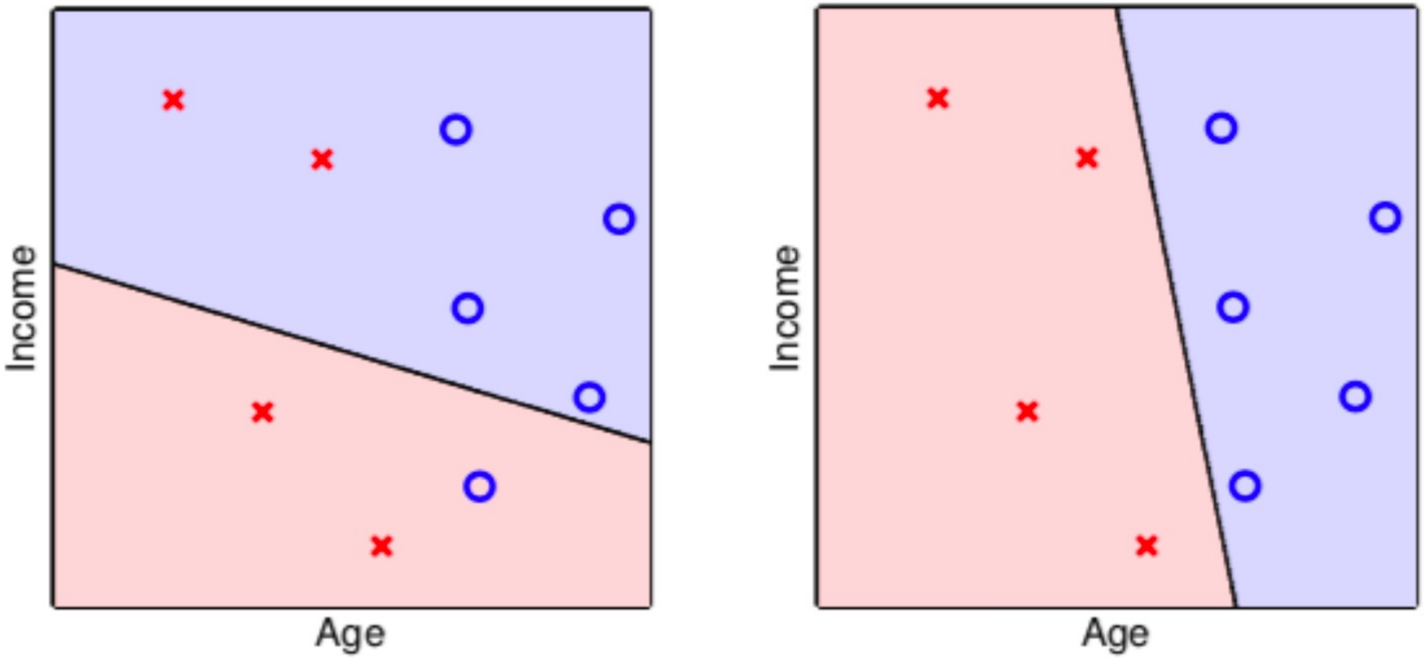
$$\boldsymbol{x} = [1, x_1, x_2, \cdots, x_d]^T$$

$$h(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T \boldsymbol{x})$$

How to train weights? Use derivative = 0?

Linear Perceptron: Representation

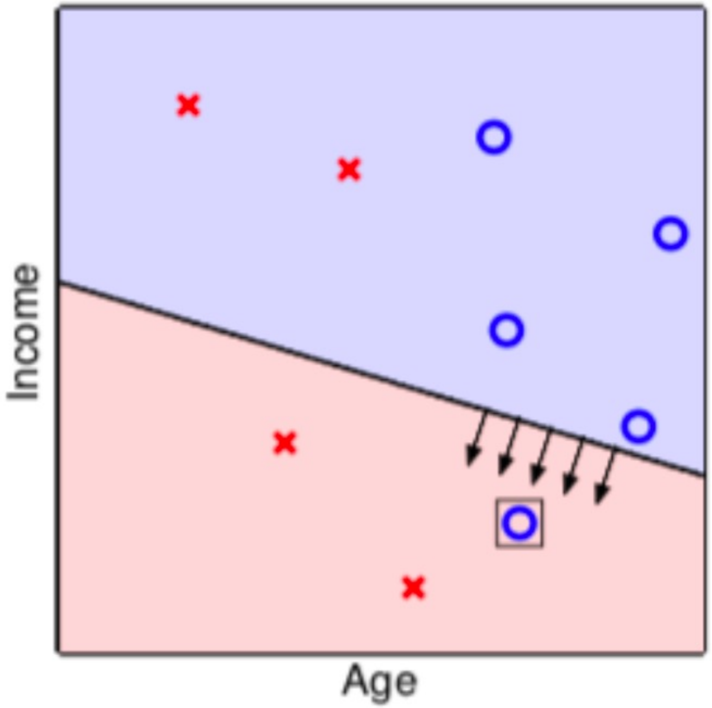
- A perceptron uses a line to separate data



- How to find a hyperplane that separates the data?

Linear Perceptron: Representation

- Idea: Start from some random weights and then improve it



Perceptron Learning Algorithm (PLA)

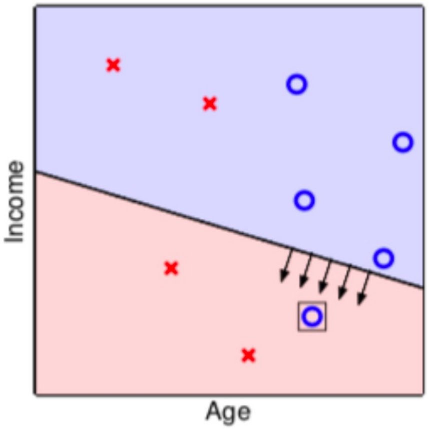
- A simple iterative method
- Incremental learning on single example at a time

1. Initialization $\mathbf{w}(\mathbf{0}) = 0$ (or any other vector)

Perceptron Learning Algorithm (PLA)

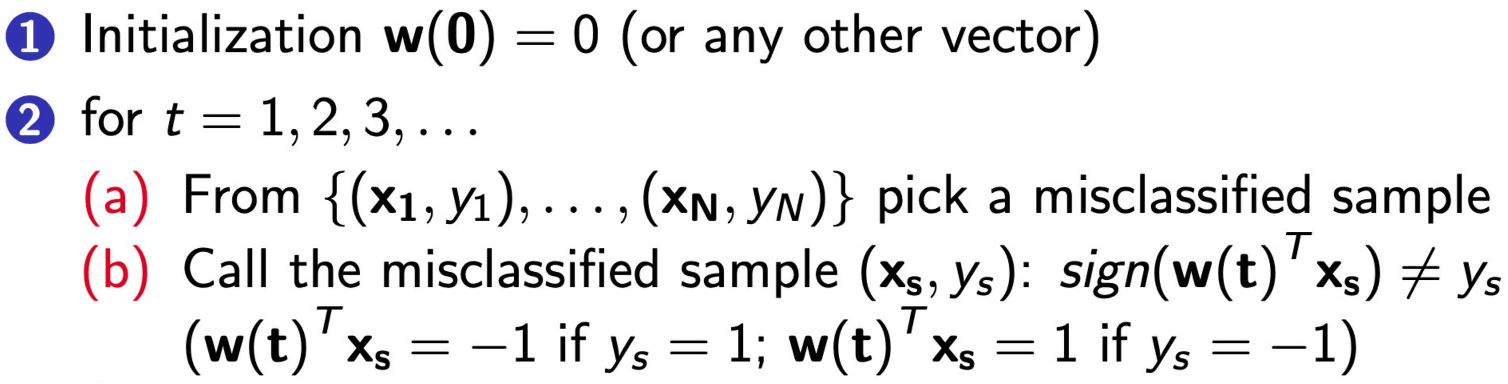
- A simple iterative method
- Incremental learning on single example at a time

```
❶ Initialization $\mathbf{w}(\mathbf{0}) = 0$ (or any other vector)
❷ for $t = 1, 2, 3, \ldots$
    (a) From $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\mathbf{N}, y_N)\}$ pick a misclassified sample
```



Perceptron Learning Algorithm (PLA)

- A simple iterative method
- Incremental learning on single example at a time



```
1 Initialization $\mathbf{w(0)} = 0$ (or any other vector)
2 for $t = 1, 2, 3, \ldots$
    (a) From $\{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_N}, y_N)\}$ pick a misclassified sample
    (b) Call the misclassified sample $(\mathbf{x_s}, y_s)$: $sign(\mathbf{w(t)}^T \mathbf{x_s}) \neq y_s$
        $(\mathbf{w(t)}^T \mathbf{x_s} = -1$ if $y_s = 1$; $\mathbf{w(t)}^T \mathbf{x_s} = 1$ if $y_s = -1)$
```

Perceptron Learning Algorithm (PLA)

- A simple iterative method
- Incremental learning on single example at a time

1. Initialization $\mathbf{w(0)} = 0$ (or any other vector)
2. for $t = 1, 2, 3, \ldots$
   (a) From $\{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_N}, y_N)\}$ pick a misclassified sample
   (b) Call the misclassified sample $(\mathbf{x_s}, y_s)$: $sign(\mathbf{w(t)}^T \mathbf{x_s}) \neq y_s$
   $(\mathbf{w(t)}^T \mathbf{x_s} = -1$ if $y_s = 1$; $\mathbf{w(t)}^T \mathbf{x_s} = 1$ if $y_s = -1)$
   (c) Update the weight:
   $\mathbf{w(t+1)} = \mathbf{w(t)} + y_s \mathbf{x_s}$

Perceptron Learning Algorithm (PLA)

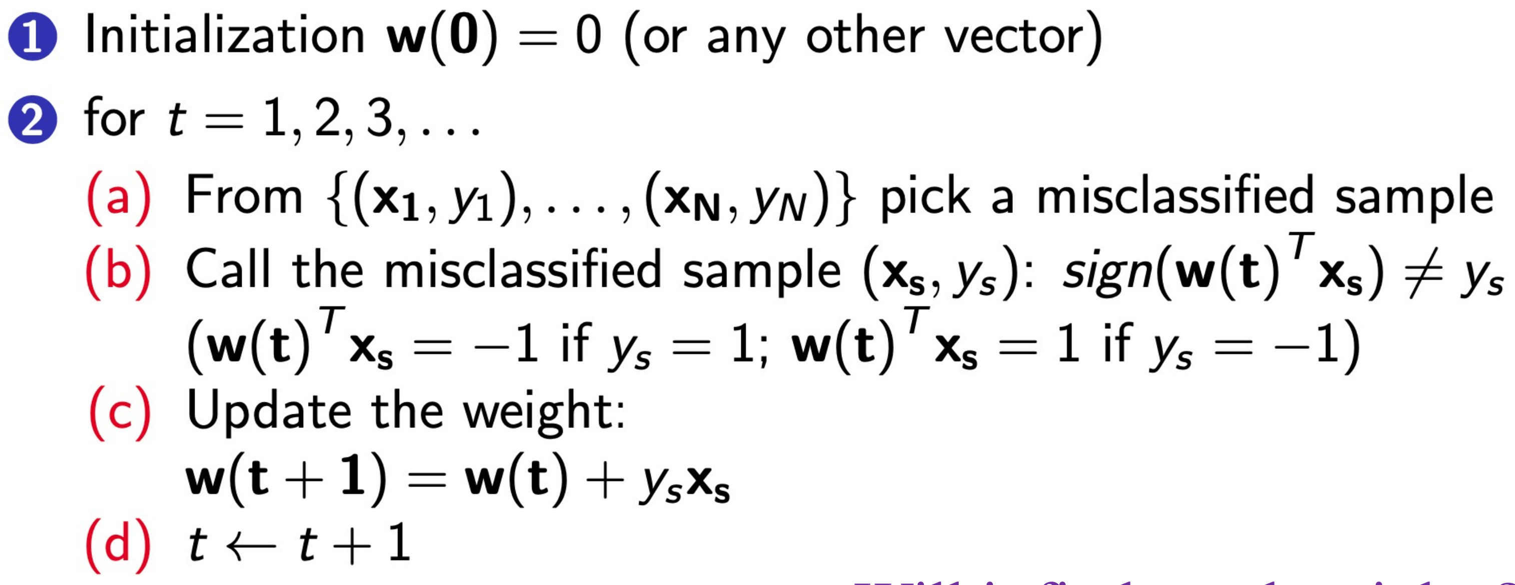
- A simple iterative method
- Incremental learning on single example at a time

1. Initialization $\mathbf{w(0)} = 0$ (or any other vector)
2. for $t = 1, 2, 3, \ldots$
   (a) From $\{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_N}, y_N)\}$ pick a misclassified sample
   (b) Call the misclassified sample $(\mathbf{x_s}, y_s)$: $sign(\mathbf{w(t)}^T \mathbf{x_s}) \neq y_s$
   $(\mathbf{w(t)}^T \mathbf{x_s} = -1 \text{ if } y_s = 1; \mathbf{w(t)}^T \mathbf{x_s} = 1 \text{ if } y_s = -1)$
   (c) Update the weight:
   $\mathbf{w(t+1)} = \mathbf{w(t)} + y_s \mathbf{x_s}$
   (d) $t \leftarrow t + 1$

Potential Issues?

Perceptron Learning Algorithm (PLA)

- Issue: The rule update considers a training sample at a time and may "destroy" the classification of other samples

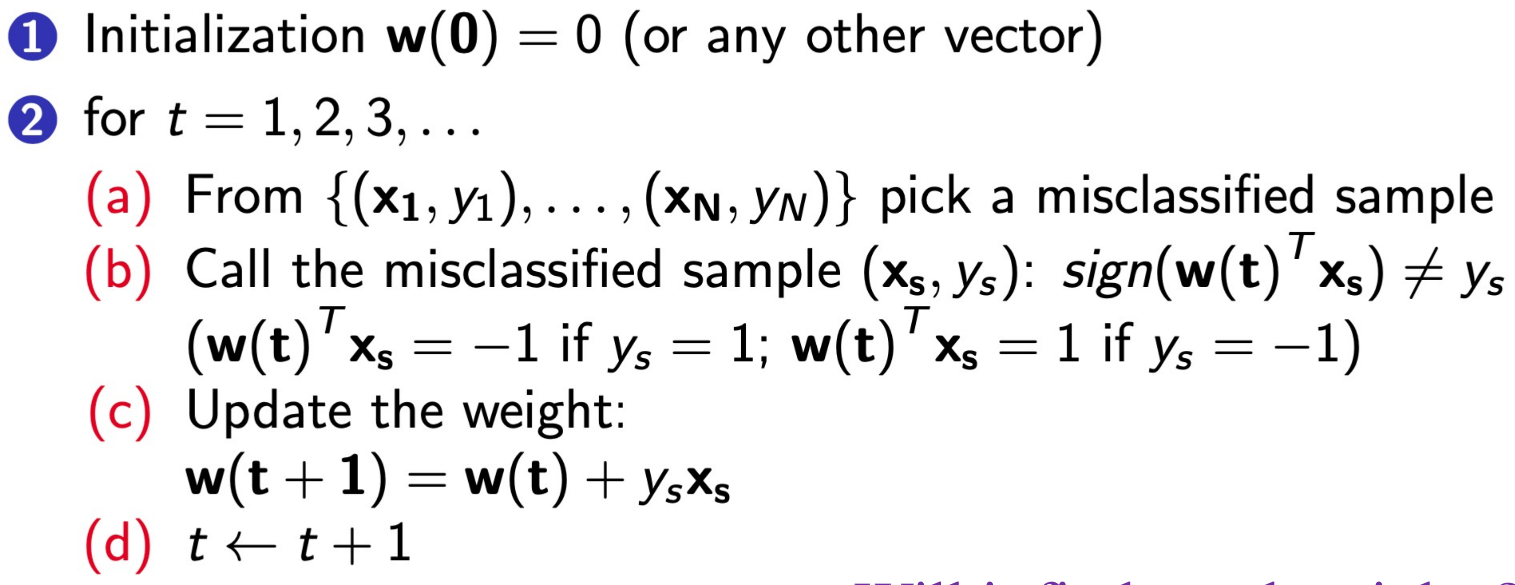


1. Initialization $\mathbf{w(0)} = 0$ (or any other vector)
2. for $t = 1, 2, 3, \ldots$
    (a) From $\{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_N}, y_N)\}$ pick a misclassified sample
    (b) Call the misclassified sample $(\mathbf{x_s}, y_s)$: $sign(\mathbf{w(t)}^T \mathbf{x_s}) \neq y_s$ $(\mathbf{w(t)}^T \mathbf{x_s} = -1$ if $y_s = 1$; $\mathbf{w(t)}^T \mathbf{x_s} = 1$ if $y_s = -1)$
    (c) Update the weight: $\mathbf{w(t+1)} = \mathbf{w(t)} + y_s \mathbf{x_s}$
    (d) $t \leftarrow t + 1$

Will it find good weights?

Perceptron Learning Algorithm (PLA)

- If the data can be fit by a linear separator (linearly separable), then after some finite number of steps, PLA is **guaranteed** to arrive to a correct solution.

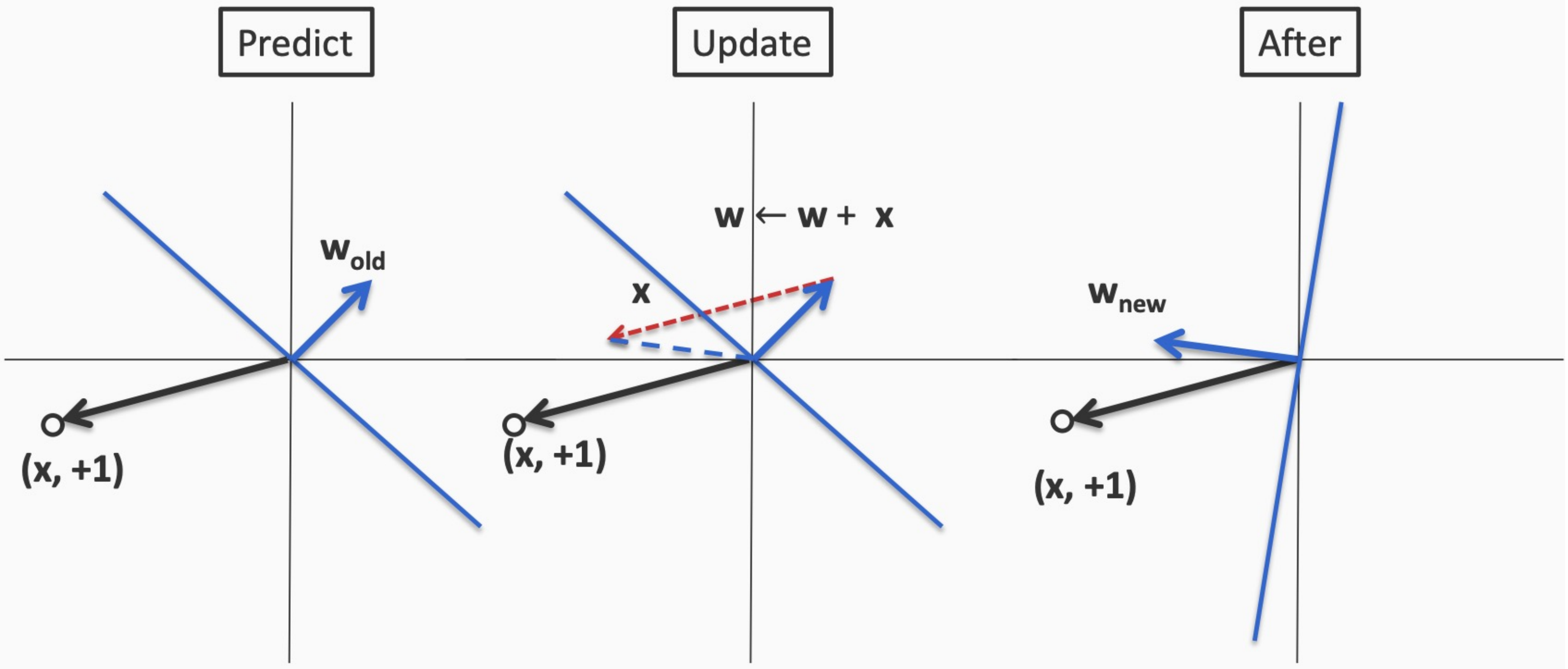


```
1 Initialization $\mathbf{w(0)} = 0$ (or any other vector)
2 for $t = 1, 2, 3, \ldots$
    (a) From $\{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_N}, y_N)\}$ pick a misclassified sample
    (b) Call the misclassified sample $(\mathbf{x_s}, y_s)$: $sign(\mathbf{w(t)}^T\mathbf{x_s}) \neq y_s$
        $(\mathbf{w(t)}^T\mathbf{x_s} = -1$ if $y_s = 1$; $\mathbf{w(t)}^T\mathbf{x_s} = 1$ if $y_s = -1)$
    (c) Update the weight:
        $\mathbf{w(t+1)} = \mathbf{w(t)} + y_s\mathbf{x_s}$
    (d) $t \leftarrow t + 1$
```

Will it find good weights?

Perceptron Learning Algorithm (PLA)

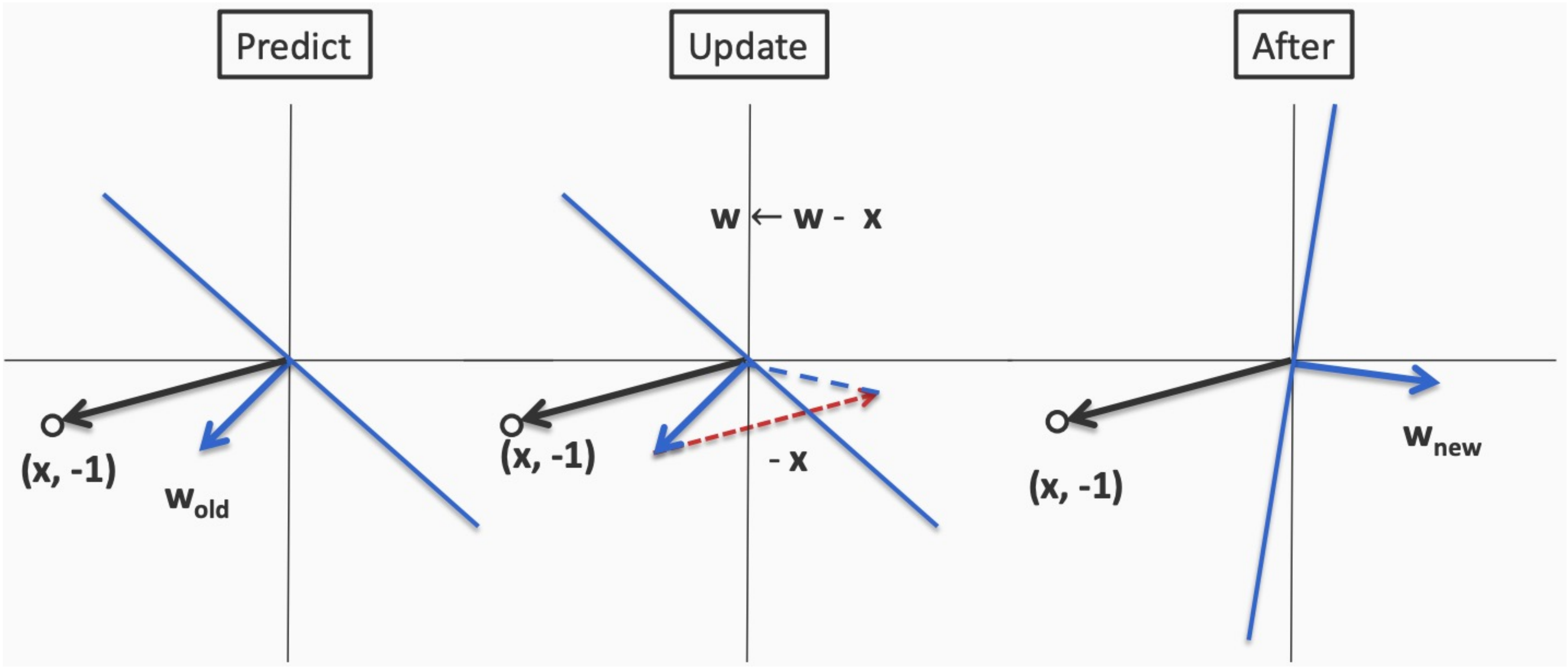
- For a mistake on a **positive** example



$$\mathbf{w}(\mathbf{t}+1) = \mathbf{w}(\mathbf{t}) + y_s \mathbf{x}_s$$

Perceptron Learning Algorithm (PLA)

- For a mistake on a **negative** example



$$\mathbf{w}(\mathbf{t}+\mathbf{1}) = \mathbf{w}(\mathbf{t}) + y_s\mathbf{x}_s$$

Linear Models: Summary

- Linear regression: predict a scalar

$$\hat{y}_i = \boldsymbol{w}^T \boldsymbol{x}_i$$

- Linear perceptron: predict $\{1, -1\}$

$$\hat{y}_i = sign(\boldsymbol{w}^T \boldsymbol{x}_i)$$

Linear Models

- Linear regression: predict a scalar
  - House price
  - Weight of a planet
- Linear perceptron: classifier
  - Predict an animal is a dog or not
  - Predict an image contains a square or not
- Logistic regression: classifier based on a probability
  - Predict how likely a team win
  - Predict how likely tomorrow is sunny