Machine Learning (CSE 446): Perceptron

Noah Smith

University of Washington
nasmith@cs.washington.edu

October 9, 2017

Happy Medium?

Decision trees (that aren't too deep): use relatively few features to classify

K-nearest neighbors: all features weighted equally.

Today: use all features, but weight them.

Happy Medium?

Decision trees (that aren't too deep): use relatively few features to classify

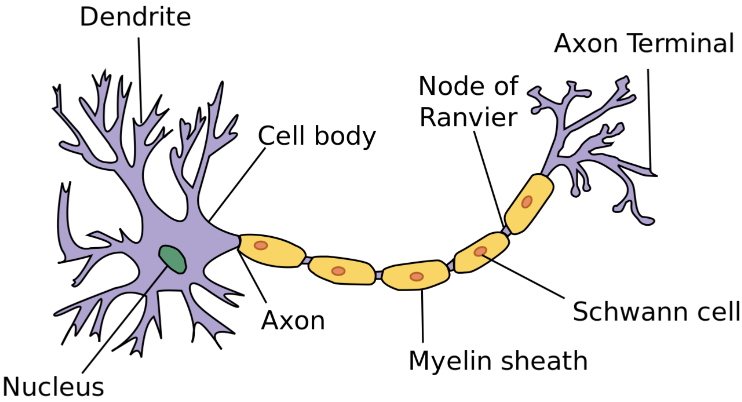
K-nearest neighbors: all features weighted equally.

Today: use all features, but weight them.

For today's lecture, assume that $y \in \{-1, +1\}$ instead of $\{0, 1\}$, and that $\mathbf{x} \in \mathbb{R}^d$.

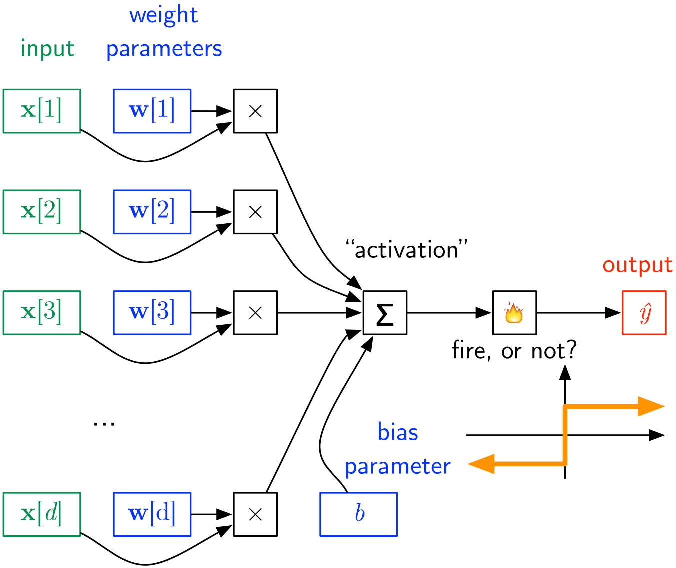
Inspiration from Neurons

Image from Wikimedia Commons.

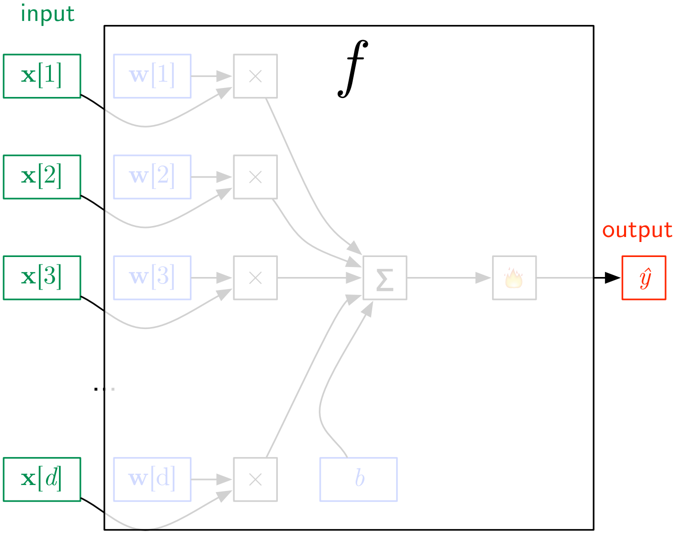


Input signals come in through dendrites, output signal passes out through the axon.

Neuron-Inspired Classifier



Neuron-Inspired Classifier



Neuron-Inspired Classifier

$$f(\mathbf{x}) = \text{sign}\left(\mathbf{w} \cdot \mathbf{x} + b\right)$$

remembering that: $\mathbf{w} \cdot \mathbf{x} = \sum_{j=1}^{d} \mathbf{w}[j] \cdot \mathbf{x}[j]$

Neuron-Inspired Classifier

$$f(\mathbf{x}) = \text{sign}\left(\mathbf{w} \cdot \mathbf{x} + b\right)$$

$$\text{remembering that: } \mathbf{w} \cdot \mathbf{x} = \sum_{j=1}^{d} \mathbf{w}[j] \cdot \mathbf{x}[j]$$

Learning requires us to set the weights $\mathbf{w}$ and the bias b.

Perceptron Learning Algorithm

Data: $D = \langle(\mathbf{x}_n, y_n)\rangle_{n=1}^{N}$, number of epochs E
Result: weights $\mathbf{w}$ and bias b
initialize: $\mathbf{w} = \mathbf{0}$ and $b = 0$;
for $e \in \{1, \ldots, E\}$ **do**
  for $n \in \{1, \ldots, N\}$, *in random order* **do**
    # predict
    $\hat{y} = \text{sign}(\mathbf{w} \cdot \mathbf{x}_n + b)$;
    if $\hat{y} \neq y_n$ **then**
      # update
      $\mathbf{w} \leftarrow \mathbf{w} + y_n \cdot \mathbf{x}_n$;
      $b \leftarrow b + y_n$;
    end
  end
end
return $\mathbf{w}, b$

Algorithm 1: PerceptronTrain

Parameters and Hyperparameters

This is the first supervised algorithm we've seen that has **parameters** that are numerical values ($\mathbf{w}$ and b).

Parameters and Hyperparameters

This is the first supervised algorithm we've seen that has **parameters** that are numerical values ($\mathbf{w}$ and b).

The perceptron learning algorithm's sole hyperparameter is E, the number of epochs (passes over the training data).

Parameters and Hyperparameters

This is the first supervised algorithm we've seen that has **parameters** that are numerical values ($\mathbf{w}$ and b).

The perceptron learning algorithm's sole hyperparameter is E, the number of epochs (passes over the training data).

Can you think of a clever way to efficiently tune E using development data?

Perceptron Updates

Suppose $y_n = 1$ but $\mathbf{w} \cdot \mathbf{x}_n + b < 0$; this means $\hat{y} = -1$.

The new weights and bias will be:

$$\begin{aligned} \mathbf{w}' &= \mathbf{w} + \mathbf{x}_n \\ b' &= b + 1 \end{aligned}$$

If we immediately made the prediction again, we'd get activation:

$$\begin{aligned} \mathbf{w}' \cdot \mathbf{x}_n + b' &= (\mathbf{w} + \mathbf{x}_n) \cdot \mathbf{x}_n + (b + 1) \\ &= \mathbf{w} \cdot \mathbf{x}_n + b + \|\mathbf{x}_n\|_2^2 + 1 \\ &\geq \mathbf{w} \cdot \mathbf{x}_n + b + 1 \end{aligned}$$

Perceptron Updates

Suppose $y_n = 1$ but $\mathbf{w} \cdot \mathbf{x}_n + b < 0$; this means $\hat{y} = -1$.

The new weights and bias will be:

$$\begin{aligned} \mathbf{w}' &= \mathbf{w} + \mathbf{x}_n \\ b' &= b + 1 \end{aligned}$$

If we immediately made the prediction again, we'd get activation:

$$\begin{aligned} \mathbf{w}' \cdot \mathbf{x}_n + b' &= (\mathbf{w} + \mathbf{x}_n) \cdot \mathbf{x}_n + (b + 1) \\ &= \mathbf{w} \cdot \mathbf{x}_n + b + \|\mathbf{x}_n\|_2^2 + 1 \\ &\geq \mathbf{w} \cdot \mathbf{x}_n + b + 1 \end{aligned}$$

So we know we've moved in the "right direction."

Geometric Interpretation

For every possible $\mathbf{x}$, there are three possibilities:

$\mathbf{w} \cdot \mathbf{x} + b > 0$	classified as positive
$\mathbf{w} \cdot \mathbf{x} + b < 0$	classified as negative
$\mathbf{w} \cdot \mathbf{x} + b = 0$	on the decision boundary

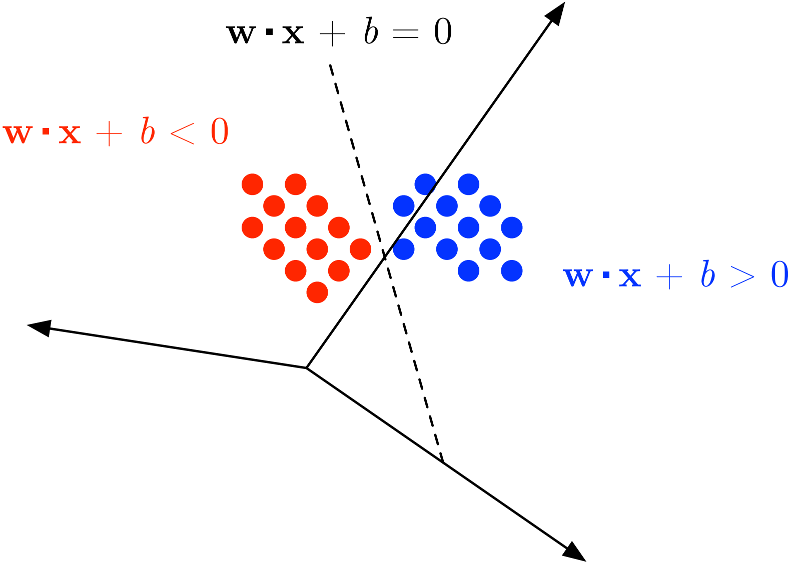
Geometric Interpretation

For every possible $\mathbf{x}$, there are three possibilities:

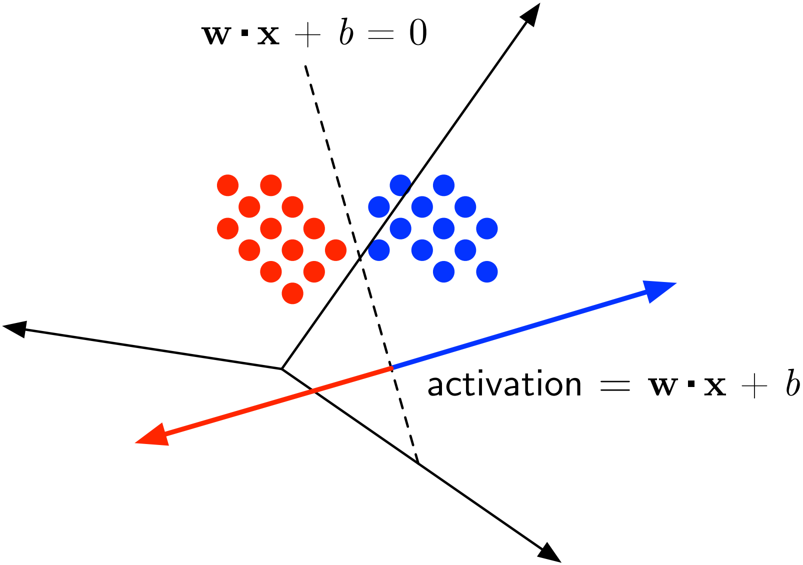
$\mathbf{w} \cdot \mathbf{x} + b > 0$	classified as positive
$\mathbf{w} \cdot \mathbf{x} + b < 0$	classified as negative
$\mathbf{w} \cdot \mathbf{x} + b = 0$	on the decision boundary

The decision boundary is a $(d-1)$-dimensional hyperplane.

Linear Decision Boundary



Linear Decision Boundary



Interpretation of Weight Values

What does it mean when . . .

Interpretation of Weight Values

What does it mean when . . .

- $w_{12} = 100$?

Interpretation of Weight Values

What does it mean when ...

- $w_{12} = 100$?
- $w_{12} = -1$?

Interpretation of Weight Values

What does it mean when ...

- $w_{12} = 100$?
- $w_{12} = -1$?
- $w_{12} = 0$?