The Perceptron

Instructor: Jessica Wu -- Harvey Mudd College

The instructor gratefully acknowledges Andrew Ng (Stanford), Eric Eaton (UPenn), David Kauchak (Pomona), and the many others who made their course materials freely available online.

Perceptron Basics

Learning Goals

- Describe the perceptron model
- Describe the perceptron algorithm
- Describe why the perceptron update works
- Describe the perceptron cost function
- Describe how a bias term affects the perceptron

The Perceptron

Assume there is a linear classifier

Start with simple learner and analyze with it does

Let $y \in \{-1, +1\}$

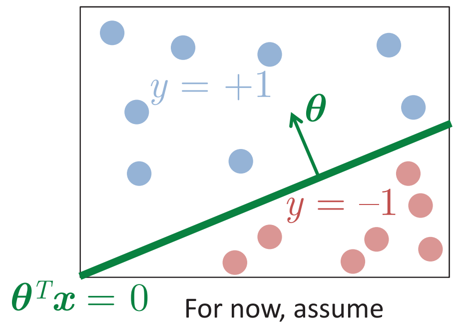
$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x})$$

where

$$g(z) = \mathrm{sgn}(z) \begin{cases} +1, & z \geq 0 \\ -1, & z < 0 \end{cases}$$

So if $\boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{x} \geq 0$, then $y = +1$

if $\boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{x} < 0$, then $y = -1$



For now, assume hyperplane through origin (no bias term).

Perceptron Algorithm

start with guess for $\boldsymbol{\theta}$ (typically $\boldsymbol{\theta} = \boldsymbol{0}$)

repeat until convergence ← possible criteria:

- all training examples correctly classified
- if average update $(||\boldsymbol{\theta}_{\text{new}} - \boldsymbol{\theta}_{\text{old}}||_2) < \epsilon$
- fixed # of iterations
- single pass over data

for $i = 1$ to n

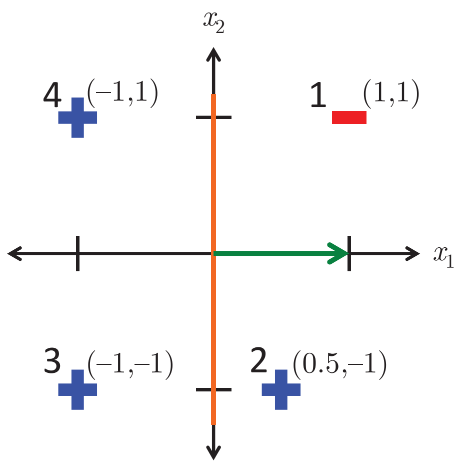
if $y^{(i)} \boldsymbol{\theta}^T \boldsymbol{x}^{(i)} \leq 0$ ← if mistake is made (why ≤ 0 rather than < 0?)

$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + y^{(i)} \boldsymbol{x}^{(i)}$ ← update step

Notes

- online learning algorithm
- guaranteed to find separating hyperplane if data is linearly separable (theorem later this lecture)

Perceptron Example



$\boldsymbol{\theta}_0 = [1,\ 0]^T$

Repeat until convergence

Process points in order 1,2,3,4

Keep track of $\boldsymbol{\theta}$ as it changes

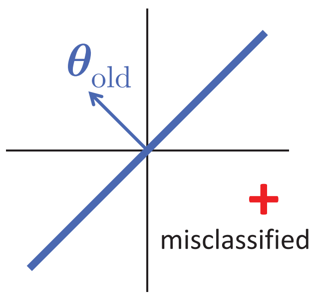
Redraw the hyperplane after each step

Based on slide by David Kauchak [originally by Piyush Rai]

(This slide intentionally left blank.)

Why the Perceptron Update Works

Geometric Interpretation



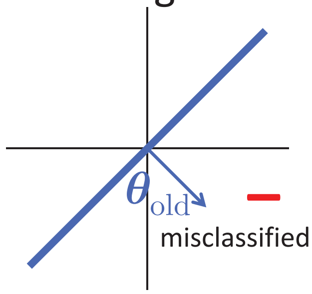
Why the Perceptron Update Works

Mathematic Proof

- Consider the misclassified example $y = +1$
  - Perceptron wrongly thinks $\boldsymbol{\theta}_{\text{old}}{}^T \boldsymbol{x} < 0$

Why the Perceptron Update Works

similar arguments for misclassified negative example



- Consider the misclassified example $y = -1$
  - Perceptron wrongly things $\boldsymbol{\theta}_{\text{old}}{}^T \boldsymbol{x} > 0$

Why the Perceptron Update Works

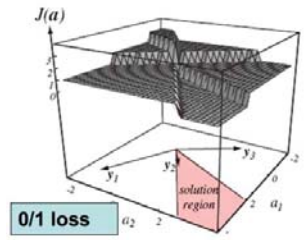
- Consider the misclassified example $y = -1$
  - Perceptron wrongly things $\boldsymbol{\theta}_{\text{old}}{}^T \boldsymbol{x} > 0$

Cost Function

What if we tried to minimize 0/1 loss?

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} \underbrace{\mathbb{I}[[y^{(i)}\boldsymbol{\theta}^T\boldsymbol{x}^{(i)} \leq 0]]}_{\substack{\text{indicator function} \\ \text{0 if prediction is correct} \\ \text{1 otherwise}}}$$

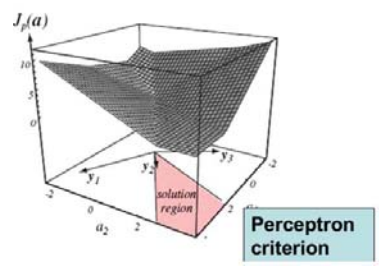
remember, prediction is correct if $y^{(i)}\boldsymbol{\theta}^T\boldsymbol{x}^{(i)} > 0$



Challenges
- NP-hard!
- small changes in $\boldsymbol{\theta}$ can induce large changes in $J(\boldsymbol{\theta})$, and change is not continuous
- lots of local minima
- not useful gradient: at any point, no information to direct us towards any minima

Idea: Use <u>surrogate</u> loss function

$$J(\boldsymbol{\theta}) = \sum_{i=1}^{n} \max\left(0, -y^{(i)}\boldsymbol{\theta}^T\boldsymbol{x}^{(i)}\right)$$



If prediction is correct
- $\max(0, -y^{(i)}\boldsymbol{\theta}^T\boldsymbol{x}^{(i)}) = 0$.

Otherwise
- it is confidence is mis-prediction
- walk in direction of negative gradient (GD!) (for single example, $-\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = y^{(i)}\boldsymbol{x}^{(i)}$)

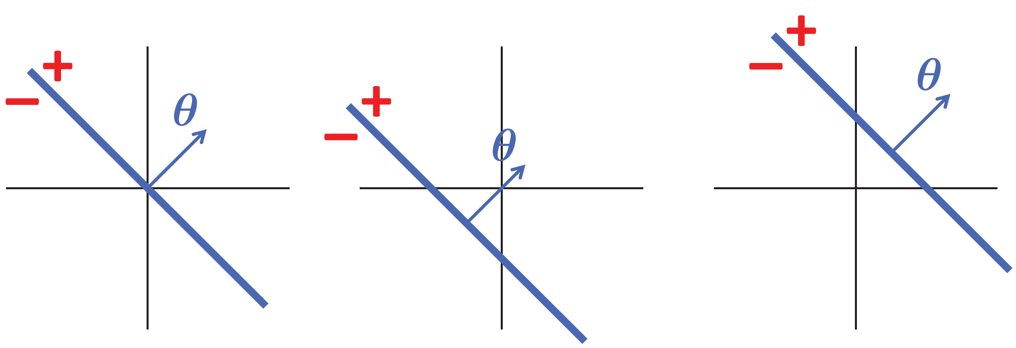
Based on slide by Eric Eaton [originally by Alan Fern]

Bias Term

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \operatorname{sgn}(\boldsymbol{\theta}^T\boldsymbol{x} + b)$$

Bias **shifts** hyperplane $-b$ units in direction of $\boldsymbol{\theta}$.

$b = 0$ $\quad$ $b > 0$ $\quad$ $b < 0$



Positive bias means more examples classified as positive.

Shift away from $\boldsymbol{\theta}$ means more space for positive classification.

Perceptron Convergence

Learning Goals

Does the perceptron converge?
If so, how long does it take?
(How many mistakes / updates?)

- State the perceptron convergence theorem
- State the implications of the theorem

Margins

For training set $\left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{n}$ and parameter vector $\boldsymbol{\theta}$:

- **functional margin** of i^{th} example

$$\hat{\gamma}^{(i)} = y^{(i)}\left(\boldsymbol{\theta}^T\boldsymbol{x} + b\right)$$

  - this is positive if $\boldsymbol{\theta}$ classifies $\boldsymbol{x}^{(i)}$ correctly
  - absolute value = "confidence" in predicted label (or "mis-confidence")

- **geometric margin** of i^{th} example

$$\gamma^{(i)} = \frac{\hat{\gamma}^{(i)}}{\|\boldsymbol{\theta}\|} = \frac{y^{(i)}\left(\boldsymbol{\theta}^T\boldsymbol{x} + b\right)}{\|\boldsymbol{\theta}\|}$$

  - signed distance of example to hyperplane (positive if example classified correctly)

- **margin of training set** = minimum geometric margin

$$\gamma = \min_i \gamma^{(i)}$$

Perceptron Convergence Theorem

Theorem (Block & Novikoff) $[b = 0]$

Assume

- $\exists\ \boldsymbol{\theta}^* \text{ s.t. } ||\boldsymbol{\theta}^*|| = 1$ and $\forall\ i,\ \gamma^{(i)} \geq \gamma^* > 0$
  - data is linearly separable with margin γ^* by unit-norm hyperplane $\boldsymbol{\theta}^*$
- $\forall\ i,\ ||\boldsymbol{x}^{(i)}|| \leq R$
  - examples are not "too big"

Then

- perceptron converges after at most $\dfrac{R^2}{(\gamma^*)^2}$ updates.

Convergence Theorem Guarantees

- Convergence rate
  - depends on margin γ^* and "size" of data R
  - but <u>not</u> on number of training examples n or data dimensionality d
- If perceptron is given data that is linearly separable with margin γ^*, it will converge to a solution that separates data and converge quickly if γ^* is large

Proof

- We assumed that $\boldsymbol{\theta}^*$ (a hyperplane consistent with data) exists
  - the classification problem is "easy" if γ^* is large
  - optimal case (maximum γ^*): $\boldsymbol{\theta}^*$ is the **maximum-margin separator**
- But we are <u>not</u> guaranteed to find $\boldsymbol{\theta}^*$ using perceptron algorithm
  - we will show how to find maximum-margin separator $\boldsymbol{\theta}^*$ using SVMs

Order of Examples

Does the order in which we traverse examples matter?

The Name "Perceptron"

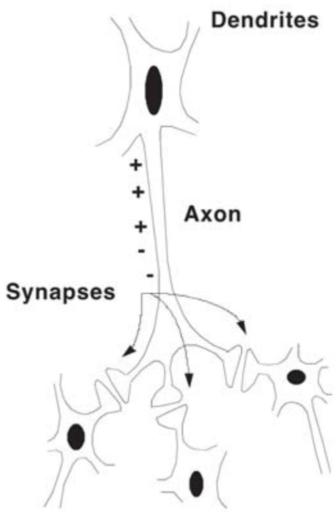
Learning Goals

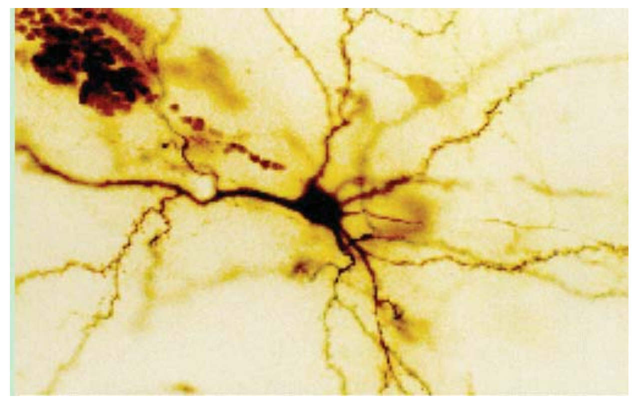
Why is it called the "perceptron" learning algorithm if it learns a line?

Why not "line learning" algorithm?

The Name "Perceptron"

Comes from our nervous system

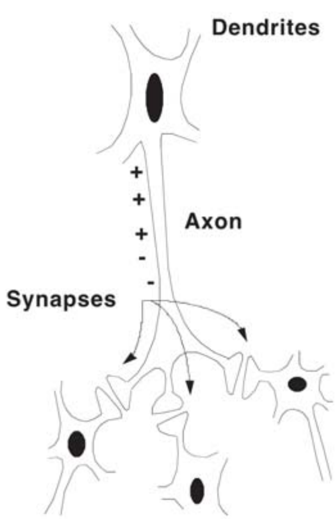




neuron

Based on slide by David Kauchak

Our Nervous System:



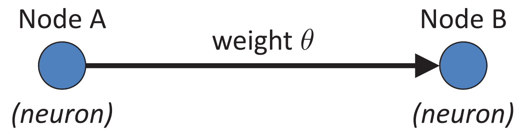
the human brain is a large collection of interconnected neurons

a **neuron** is a brain cell

- collects, processes, and disseminates electrical signals
- connected via synapses
- **fire** depending on conditions of neighboring neurons

Based on slide by David Kauchak

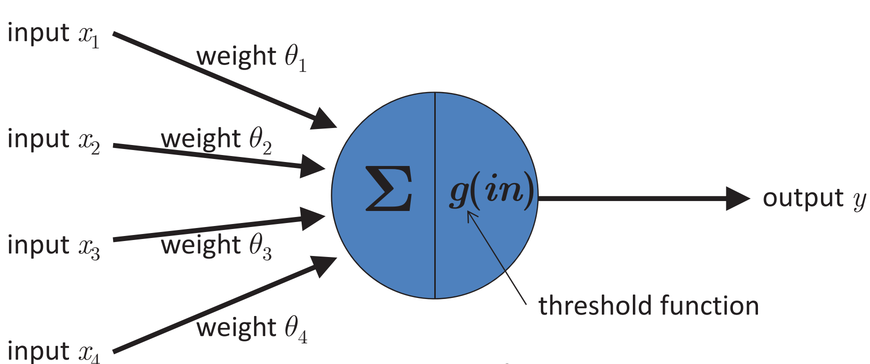
A Neuron



- θ is the strength of signal sent from A to B
- if A fires and θ is positive, then A stimulates B
- if A fires and θ is negative, then A inhibits B
- if B is stimulated enough, then it also fires
- amount of stimulation required is determined by its threshold

Based on slide by David Kauchak

A Single Neuron / Perceptron



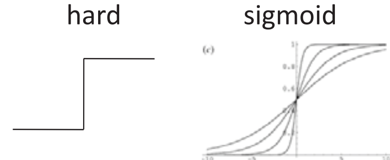
$in = \Sigma_j \, \theta_j x_j$

linear combination

possible thresholds

hard

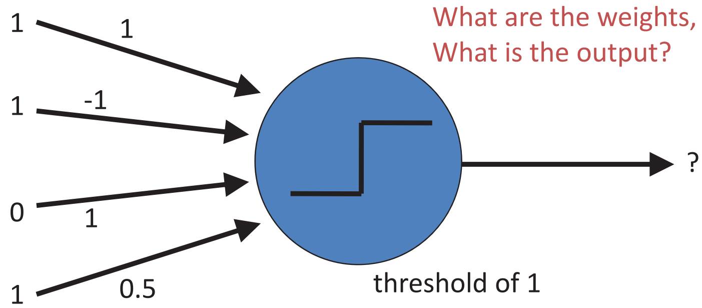
sigmoid

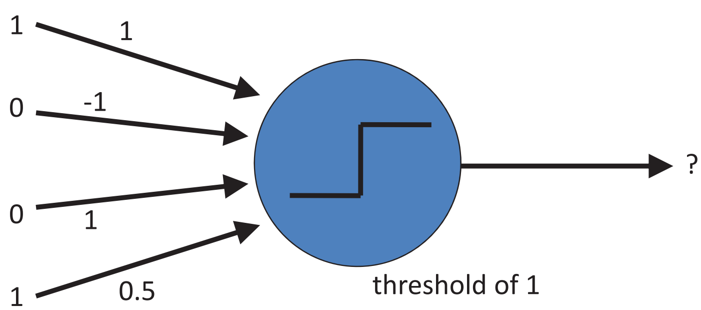


Based on slide by David Kauchak

Neuron Example

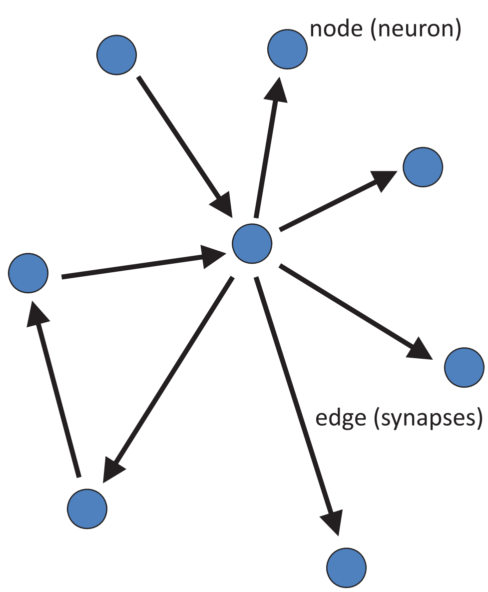
What are the weights, and what is *b*?
What is the output?

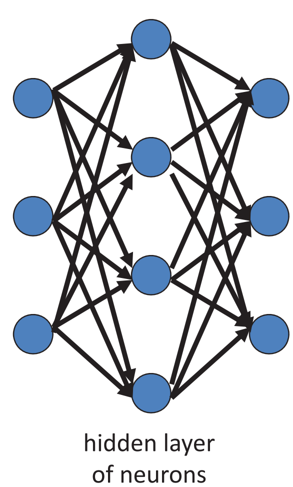




Based on slide by David Kauchak

Neural Networks





Based on slide by David Kauchak

(extra slides)

Proof of
Perceptron Convergence Theorem

Learning Goals

- Prove the Perceptron Convergence Theorem =D

Proof Overview

- $\exists\ \boldsymbol{\theta}^*$ s.t. data is linearly separable with margin γ^*
  - (we do not know $\boldsymbol{\theta}^*$ but we know that it exists)
- perceptron algorithm tries to find $\boldsymbol{\theta}$ that points roughly in same direction as $\boldsymbol{\theta}^*$
  - for large γ^*, "roughly" is very rough
  - for small γ^*, "roughly" is very precise
- every update, angle between $\boldsymbol{\theta}$ and $\boldsymbol{\theta}^*$ changes

recall $\cos\alpha = \dfrac{\boldsymbol{u}^T\boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|}$, we will prove that

- $\boldsymbol{u}^T\boldsymbol{v}$ increases a lot
- $||\boldsymbol{u}||$ and $||\boldsymbol{v}||$ do not increase very much
- so angle α decreases each update

Proof by Induction

(This slide intentionally left blank.)