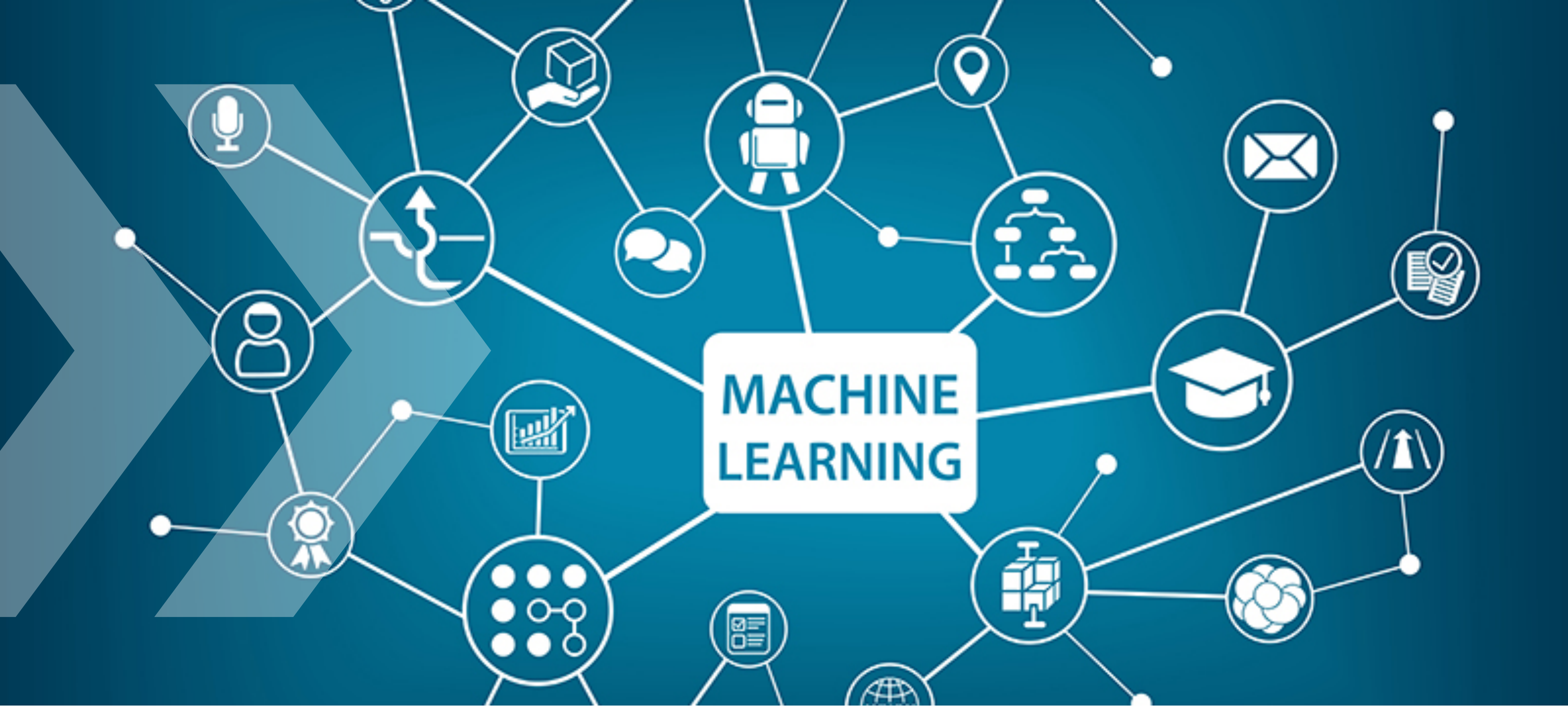


CS489/698: Intro to ML

Lecture 01: Perceptron

Yao-Liang Yu

Outline

- Announcements
- Perceptron
- Projects
- Next

UNIVERSITY OF WATERLOO

Outline

- Announcements
- Perceptron
- Projects
- Next

UNIVERSITY OF WATERLOO

Announcements

- Enrollment
  - Undergrad (CS489): talk to your advisors
  - Grad (CS698): already long waiting list…

- Computer Resource
  - Email me a project plan to get a sharcnet account

- Assignment 1 unleashed tonight

Outline

- Announcements
- Perceptron
- Projects
- Next

Spam filtering example

	and	viagra	the	of	nigeria	y
$\mathbf{x}^1$	1	1	0	1	1	+1
$\mathbf{x}^2$	0	0	1	1	0	-1
$\mathbf{x}^3$	0	1	1	0	0	+1
$\mathbf{x}^4$	1	0	0	1	0	-1
$\mathbf{x}^5$	1	0	1	0	1	+1
$\mathbf{x}^6$	1	0	1	1	0	-1

- Training set ($X = [\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^n]$, $\mathbf{y}=[y_1, y_2, \ldots, y_n]$)
  - $\mathbf{x}^i$ in $X = R^d$: instance i with d dimensional features
  - y_i in $Y = \{-1, 1\}$: instance i is spam or ham?
- Good feature representation is of uttermost importance

Batch vs. Online

- Batch learning
  - Interested in performance on test set X'
  - Training set (X, $\mathbf{y}$) is just a means
  - Statistical assumption on X and X'

- Online learning
  - Data comes one by one (streaming)
  - Need to predict y before knowing its true value
  - Interested in making as few mistakes as possible
  - "Friendliness" of the sequence $(\mathbf{x}^1, y_1), (\mathbf{x}^2, y_2), \ldots$

- Online to Batch conversion

UNIVERSITY OF WATERLOO

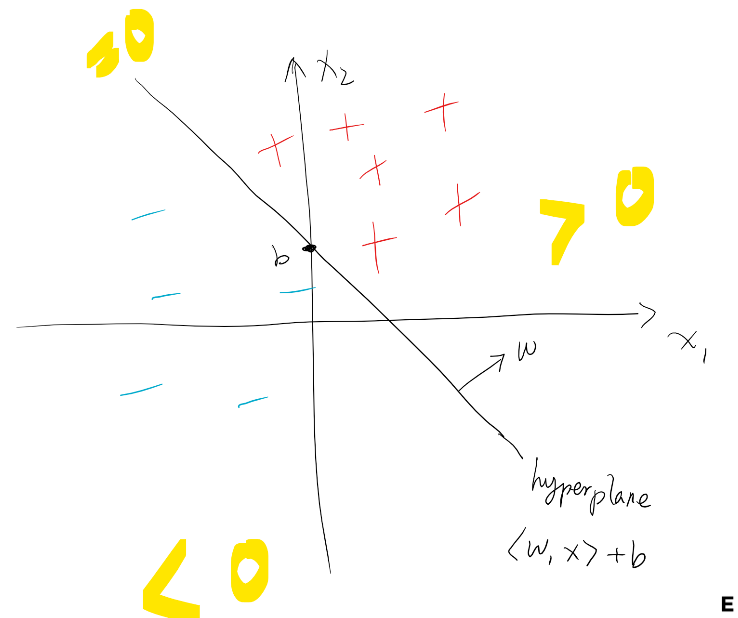
Linear threshold function

- Find (**w**, b) such that for all i:

$$y_i = \text{sign}(\langle \mathbf{w}, \mathbf{x}^i \rangle + b)$$

  - **w** in R^d: weight vector for the separating hyperplane
  - b in R: offset (threshold, bias) of the separating hyperplane
  - sign: thresholding function

$$\text{sign}(t) = \begin{cases} 1, & t > 0 \\ -1, & t \leq 0 \end{cases}$$



Simplification

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = \left\langle \boxed{\begin{pmatrix} \mathbf{w} \\ b \end{pmatrix}}, \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \right\rangle$$

Also denote as **w**…

- Padding constant 1 to the end of each **x**

$$y \cdot \operatorname{sign}(\langle \mathbf{w}, \mathbf{x} \rangle) = \operatorname{sign}(\langle \mathbf{w}, y\mathbf{x} \rangle)$$

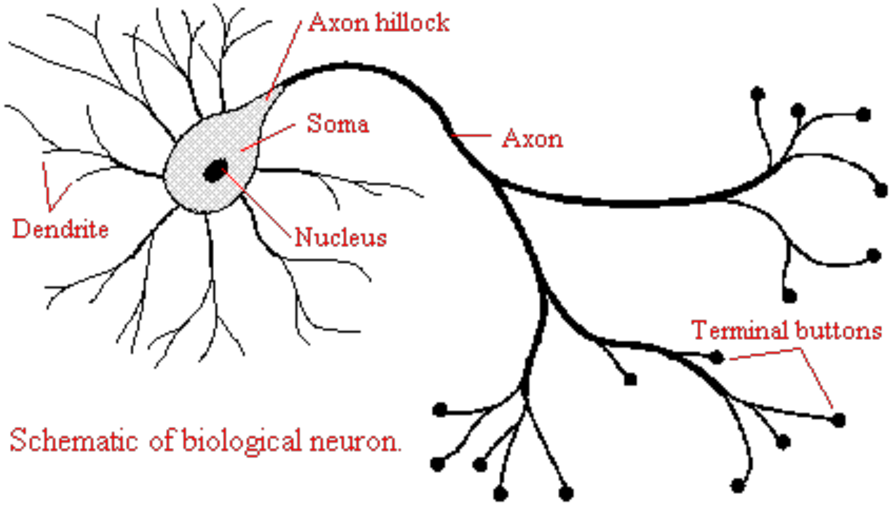
- Pre-multiply each **x** with its label y

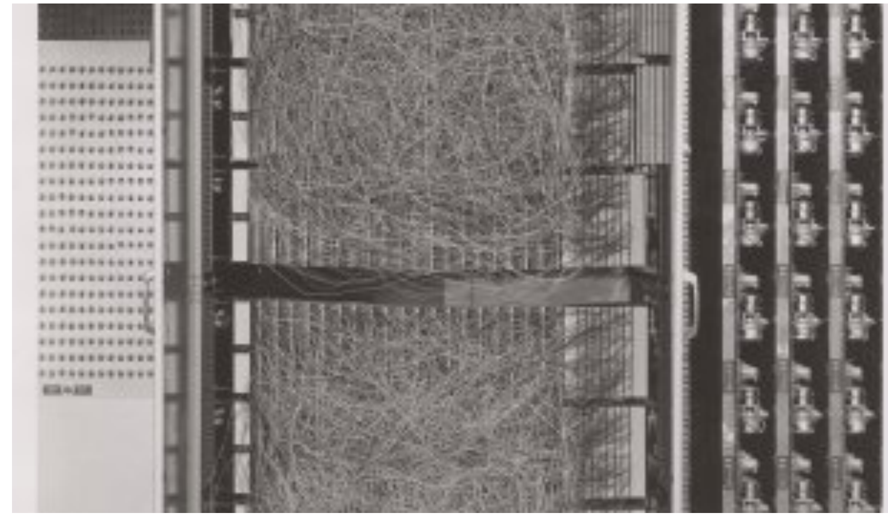
$$A = [\mathbf{a}^1, \ldots, \mathbf{a}^n]$$

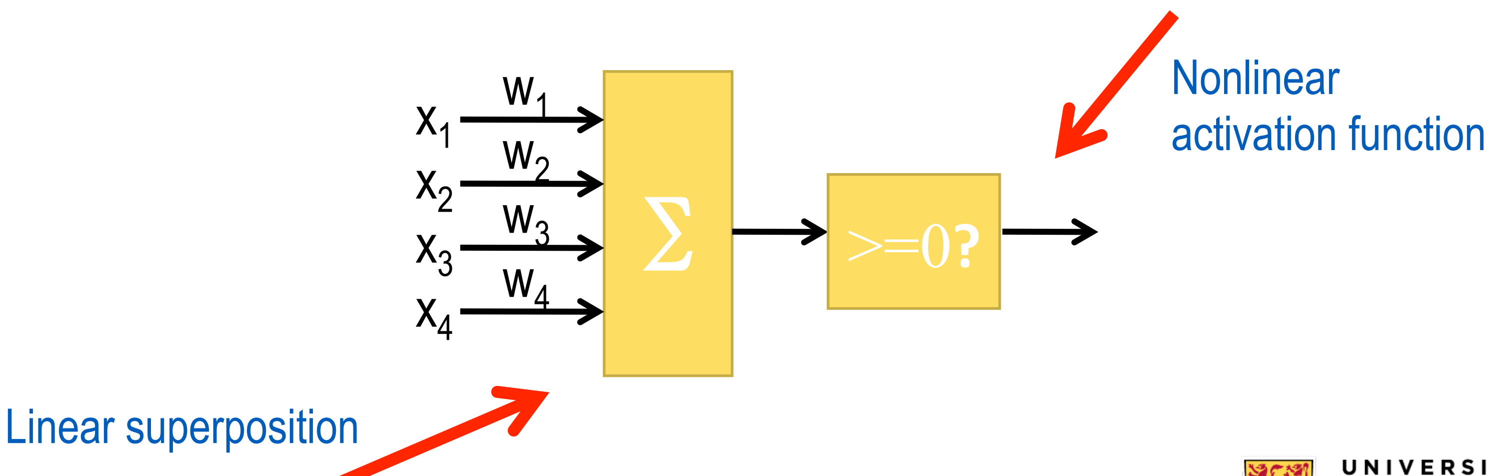
- Find **w** such that $A^T\mathbf{w} > \mathbf{0}$

$$\mathbf{a}^i = \begin{pmatrix} y_i \mathbf{x}^i \\ y_i \end{pmatrix}$$

Perceptron [Rosenblatt'58]







The perceptron algorithm

Algorithm 1: The perceptron algorithm (Rosenblatt 1958)

Input: $A \in \mathbb{R}^{(d+1)\times n}$, threshold $\delta \geq 0$, initialize $\mathbf{w}_0 \in \mathbb{R}^{d+1}$ arbitrarily

repeat
  select some column $\mathbf{a}$ of A;
  if $\langle \mathbf{a}, \mathbf{w}_t \rangle \leq \delta$ **then**
    $\mathbf{w}_{t+1} = \mathbf{w}_t + \mathbf{a}$;   `// update only when making a mistake`
    $t \leftarrow t + 1$
until *convergence*;

- Typically $\delta = 0$, $\mathbf{w}_0 = \mathbf{0}$
  - $\langle a, w \rangle = y(\langle x, w \rangle + b) < 0$ iff $\overset{\text{truth}}{y} \neq \underbrace{\mathrm{sign}(\langle x, w \rangle + b)}_{\text{prediction}}$
- Lazy update: if it ain't broke, don't fix it

$$\langle \mathbf{a}, \mathbf{w}_{t+1} \rangle = \langle \mathbf{a}, \mathbf{w}_t + \mathbf{a} \rangle = \langle \mathbf{a}, \mathbf{w}_t \rangle + \|\mathbf{a}\|_2^2 > \langle \mathbf{a}, \mathbf{w}_t \rangle$$

Does it work?

	and	viagra	the	of	nigeria	y
$\mathbf{x}^1$	1	1	0	1	1	+1
$\mathbf{x}^2$	0	0	1	1	0	-1
$\mathbf{x}^3$	0	1	1	0	0	+1
$\mathbf{x}^4$	1	0	0	1	0	-1
$\mathbf{x}^5$	1	0	1	0	1	+1
$\mathbf{x}^6$	1	0	1	1	0	-1

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

- $w_0 = [0, 0, 0, 0, 0]$, $b_0 = 0$, pred on $\mathbf{x}^1$ undefined
- $w_1 = [1, 1, 0, 1, 1]$, $b_1 = 1$, pred 1 on $\mathbf{x}^2$, wrong
- $w_2 = [1, 1, -1, 0, 1]$, $b_2 = 0$, pred on $\mathbf{x}^3$ undefined
- $w_3 = [1, 2, 0, 0, 1]$, $b_3 = 1$, pred 1 on $\mathbf{x}^4$, wrong
- $w_4 = [0, 2, 0, -1, 1]$, $b_4 = 0$, pred 1 on $\mathbf{x}^5$, correct
- $w_4 = [0, 2, 0, -1, 1]$, $b_4 = 0$, pred -1 on $\mathbf{x}^6$, correct

Perceptron Convergence Theorem

Theorem (Block'62; Novikoff'62). Assume there exists some $\mathbf{w}$ such that $A^T\mathbf{w} > 0$, then the perceptron algorithm converges to some $\mathbf{w}^*$. If each column of A is selected indefinitely often, then $A^T\mathbf{w}^* > \delta$.

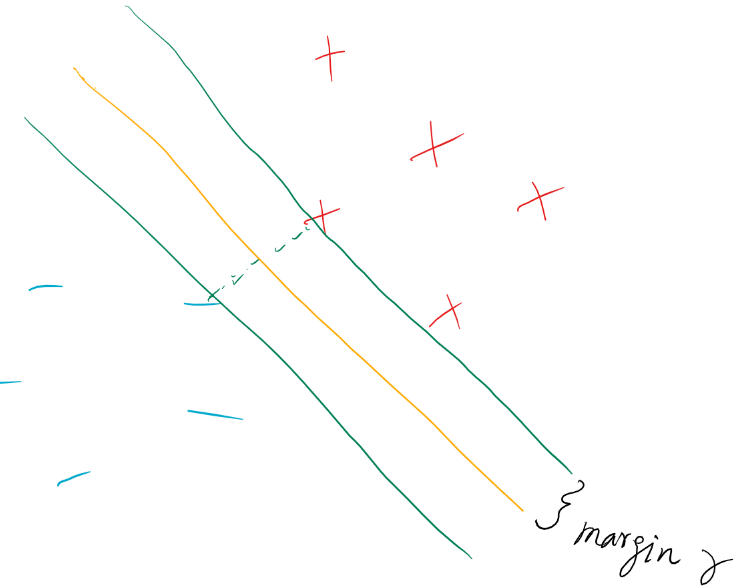
Corollary. Let $\delta = 0$ and $\mathbf{w}_0 = \mathbf{0}$. Then perceptron converges after at most $(R/\gamma)^2$ steps, where

$$R = \max_i \|A_{:i}\|_2, \quad \gamma = \max_{\mathbf{w}:\|\mathbf{w}\|_2 \le 1} \min_i \langle \mathbf{w}, A_{:i} \rangle$$

UNIVERSITY OF WATERLOO

The Margin

$$
\begin{aligned}
\min_{\mathbf{w}^\star : A^\top \mathbf{w}^\star \geq \gamma \mathbf{1}} \frac{\|\mathbf{w}^\star\|_2^2}{\gamma^2} &= \min_{\mathbf{w}: A^\top \mathbf{w} \geq \mathbf{1}} \|\mathbf{w}\|_2^2 \\
&= \min_{(\mathbf{w}, t): \|\mathbf{w}\|_2 \leq t, A^\top \mathbf{w} \geq \mathbf{1}} t^2 \\
&= \min_{(\mathbf{w}, t): \|\mathbf{w}\|_2 \leq 1, A^\top \mathbf{w} \geq \frac{1}{t} \mathbf{1}} t^2 \\
&= \min_{\mathbf{w}: \|\mathbf{w}\|_2 \leq 1, A^\top \mathbf{w} > 0} \frac{1}{\min_{\mathbf{a} \in A} \langle \mathbf{a}, \mathbf{w} \rangle^2} \\
&= \Bigg[\frac{1}{\underbrace{\max_{\mathbf{w}: \|\mathbf{w}\|_2 \leq 1} \min_{\mathbf{a} \in A} \langle \mathbf{a}, \mathbf{w} \rangle}_{\text{the margin } \gamma}} \Bigg]^2
\end{aligned}
$$



What does the bound mean?

Corollary. Let $\delta = 0$ and $\mathbf{w}_0 = \mathbf{0}$. Then perceptron converges after at most $(R/\gamma)^2$ steps, where

$$R = \max_i \|A_{:i}\|_2, \quad \gamma = \max_{\mathbf{w}:\|\mathbf{w}\|_2 \le 1} \min_i \langle \mathbf{w}, A_{:i} \rangle$$

- Treating R and γ as constants, then

  # of mistakes independent of n and d!

- Otherwise may need exponential time…

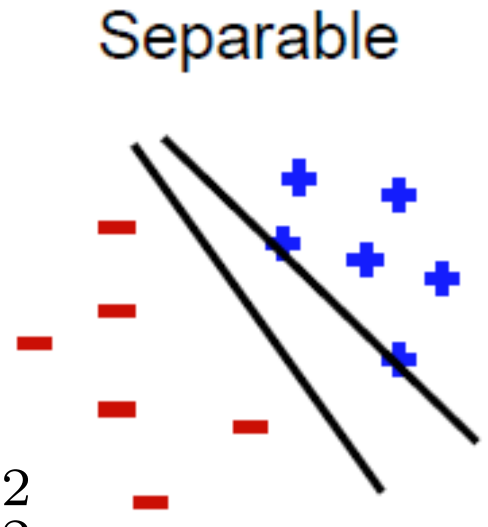
- Can we improve the bound?

UNIVERSITY OF WATERLOO

But

✓ The larger the margin, the faster perceptron converges

x But perceptron stops at an arbitrary linear separator…



- Which one do you prefer?

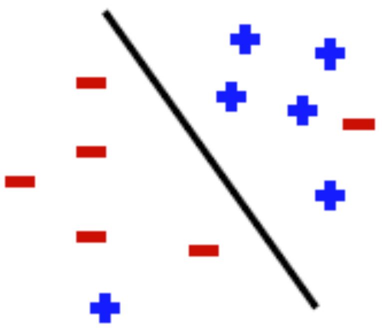
Support vector machines

$$\min_{A^\top \mathbf{w} \geq 1} \frac{1}{2}\|\mathbf{w}\|_2^2 \approx \min_{y_i(\langle \mathbf{w}, \mathbf{x}^i \rangle + b) \geq 1} \frac{1}{2}\|\mathbf{w}\|_2^2$$

What if non-separable?

- Find a better feature representation
- Use a deeper model
- Soft margin

Non-Separable



$$\forall \mathbf{w}^\star, \forall \gamma > 0, \text{ and } \forall \mathbf{a} \in A: \quad \langle \mathbf{a}, \mathbf{w}^\star \rangle \geq \gamma - (\gamma - \langle \mathbf{a}, \mathbf{w}^\star \rangle)_+$$

Perceptron Boundedness Theorem

- Perceptron convergence requires the existence of a separating hyperplane
  - how to check this assumption in practice?
  - What if it fails? (trust me, it will)

Theorem (Minsky and Papert'67; Block and Levin'70).

The iterates of the perceptron algorithm are always bounded. In particular, if there is no separating hyperplane, then perceptron cycles.

"…proof of this theorem is complicated and obscure. So are the other proofs we have since seen..." --- Minsky and Papert, 1987

When to stop perceptron?

- Online learning: never

- Batch learning
  - Maximum number of iteration reached or run out of time
  - Training error stops changing
  - Validation error stops decreasing
  - Weights stopped changing much, if using a diminishing step size $\eta_{t,}$ i.e., $w_{t+1} \leftarrow w_t + \eta_t \, y_i \, \mathbf{x}^i$

Multiclass Perceptron

- One vs. all
  - Class c as positive
  - All other classes as negative
  - Highest activation wins: pred = $\arg\max_c \mathbf{w}_c^T \mathbf{x}$

- One vs. one
  - Class c as positive
  - Class c' as negative
  - Voting

Winnow (Littlestone'88)

Algorithm 2: The Winnow algorithm (Littlestone 1988)

Input: $A \in \mathbb{R}^{d' \times n}$, threshold $\delta \geq 0$, step size $\eta > 0$, initialize $\mathbf{w}_0 \in \mathrm{int}\,\Delta_{d'-1}$

```
repeat
    select some column a of A;
    if ⟨a, w_t⟩ ≤ δ then
        w_{t+1} = w_t ⊙ exp(ηa) ;          // update only when making a mistake
        w_{t+1} ← w_{t+1}/||w_{t+1}||_1 ;  // normalize
        t ← t + 1
until convergence;
```

- Multiplicative vs. additive

Theorem (Littlestone'88). Assume there exists some nonnegative **w** such that $A^T\mathbf{w} > 0$, then winnow converges to some $\mathbf{w}^*$. If each column of A is selected indefinitely often, then $A^T\mathbf{w}^* > \delta$.

Another example: pricing

- Selling a product Z to user x with price y = f(x, w)
- If y is too high, user won't buy, update w
- If y is acceptable, user buy [update w]
- How to measure our performance?

Outline

- Announcements
- Perceptron
- Projects
- Next

UNIVERSITY OF WATERLOO

Projects

- Using the perceptron algorithm to solve linear programming (Dunagan and Vempala 2008) or conic programming (Belloni et al. 2009)
- Smooth analysis of the perceptron algorithm (Blum and Dunagan 2002)
- Second-order perceptron (Cesa-Bianchi et al. 2005)
- Forgetron (Dekel et al. 2008)
- Better proof for the boundedness theorem (Amaldi and Hauser 2005)

Outline

- Announcements
- Perceptron
- Projects
- Next

Linear regression

- Perceptron is a linear rule for classification
  - y in a finite set, e.g., {+1, -1}

- What if y can be any real number?
  - Known as regression
  - Again, there is a linear rule

empirical risk

regularization

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2}\|X\mathbf{w} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2$$

hyperparameter

- Regularization to avoid overfitting
- Cross-validation to select hyperparameter