Perceptron, again

Homogeneous linear classifiers

- Homogeneous linear classifier: $w \in \mathbb{R}^d$ (*weight vector*)

$$f_w(x) = f_{w,0}(x) = \begin{cases} +1, & \langle x, w \rangle > 0 \\ -1, & \langle x, w \rangle \leq 0 \end{cases}$$

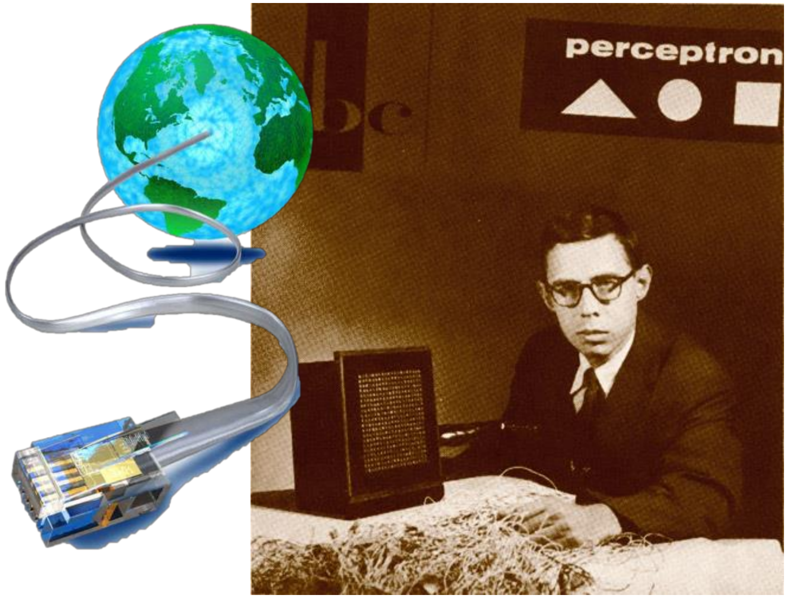
Perceptron (Rosenblatt, ‘58)

Input: training data S

- **Let** $w_1 = \vec{0}$.
- **For** $t = 1,2, \ldots$:
  - **If** there is $(x_t, y_t) \in S$ such that $f_{w_t}(x_t) \neq y_t$, **then**:
    - **Update**: $w_{t+1} \coloneqq w_t + y_t x_t$
  - **Else**: **return** w_t

> If S is separable with margin $\gamma > 0$, and $R \coloneqq \max_{(x,y)\in S} \|x\|$,
>
> then Perceptron terminates after $\left(\frac{R}{\gamma}\right)^2$ updates with linear separator for S.

Online Perceptron



Input: training data S as an *input stream*.

- **Let** $w = \vec{0}$.
- **For** each $(x, y) \in S$:
  - **If** $f_w(x) \neq y$, **then**:
    - **Update**: $w \coloneqq w + yx$
- **Return** w

Online Perceptron

- **Always terminates**: in fact, just makes a *single pass through the data*!
- Does it return a linear separator (assuming one exists)? Maybe not.

- **However**:

If S is separable with margin $\gamma > 0$, and $R \coloneqq \max_{(x,y)\in S} ||x||$,

then Online Perceptron makes at most $\left(\frac{R}{\gamma}\right)^2$ mistakes (and updates).

upates is over the the wrong classification

What good is a mistake bound?

- **Mistake bound**: upper-bound on number of mistakes made by an *online learning algorithm* on an arbitrary sequence of examples.

- **Online learning algorithm** (for our purposes): algorithm that operates on a stream of examples, and always has a “current classifier” in hand.

- **Amazing fact**: online learning algorithms with small mistake bounds can be used to produce classifiers with small classification error!

Voted-Perceptron (Freund and Schapire, '99)

Input: training data S as an *input stream*.

- **Let** $w_1 = \vec{0}, c_1 = 0, t = 0.$
- **For** each $(x, y) \in S$:
  - **If** $f_{w_t}(x) \neq y$, **then**:
    - **Update**: $w_{t+1} := w_t + yx,$
      $c_{t+1} := 0,$
      $t := t + 1.$
  - **Else**: $c_t := c_t + 1$
- **Return** $\big((w_1, c_1), (w_2, c_2), \dots, (w_t, c_t)\big)$

c_t represents # of examples that w_t correctly classifies.

A.K.A. "survival time".

note the survival time in c fore each classifier w

Voted-Perceptron (Freund and Schapire, '99)

What is the final classifier based on $(w_1, c_1), (w_2, c_2), \dots, (w_t, c_t)$?

Input: test point x

- **Compute score**: $z \coloneqq \sum_{s=1}^{t} c_s \, f_{w_s}(x)$
- **Compute prediction**: $\hat{y} \coloneqq \mathrm{sign}(z)$

c_s represents # of examples that w_s correctly classifies.

A.K.A. "survival time".

the computed score z should against
which value ?

Voted-Perceptron: classification error

- Assume S is a sequence of n i.i.d. examples (x, y) from P.
- Also assume there exists $w_\star$ with $\|w_\star\| = 1$ and $\gamma, R > 0$ such that

$$\Pr_{(x,y)\sim P}(y\langle w_\star, x\rangle \geq \gamma \wedge \|x\| \leq R) = 1.$$

- If $\hat{f}$ denote the classifier returned by Voted-Perceptron on input S, then:

$$\mathbb{E}\left[\mathrm{err}\left(\hat{f}\right)\right] \leq \frac{2(R/\gamma)^2}{n+1}$$

Other variants

- What determines final classifier?
  1. Just run Online Perceptron and return final w
  2. Voted-Perceptron, based on survival times c_i
  3. Weighted Perceptron: $\hat{w} := \sum_{i=1}^{t} c_i w_i$
- How to use the training data?
  1. Make a single pass through S.
  2. Make multiple passes through S.

note: the order of the S is matter

Experimental results

Test error(online - P)
the classifier w' could be easily distorted by a sigular feature vector, not stable~~~

- Using OCR digits data, binary classification problem of distinguishing "9" from other digits.

even the sample is IID, what if the sample is sorted~ 54000 negative, then 6000 positive~~

- # training examples: 60000 (about 6000 are from class "9").

# passes	0.1	1	2	3	4	10
Test error (online-P)	0.079	0.064	0.057	0.063	0.058	0.059
Test error (voted-P)	0.045	0.039	0.038	0.038	0.038	0.037
Test error (average-P)	0.045	0.039	0.038	0.038	0.038	0.037