Machine Learning

A Geometric Approach

CUNY Graduate Center, Spring 2013

Lectures 2-4 Online Learning: Perceptron and its Extensions

(including voted/avg perc, (aggressive) MIRA, multiclass, and feature preprocessing)

Professor Liang Huang

huang@cs.qc.cuny.edu

Alex Smola (CMU) slides

Liang Huang (CUNY) slides

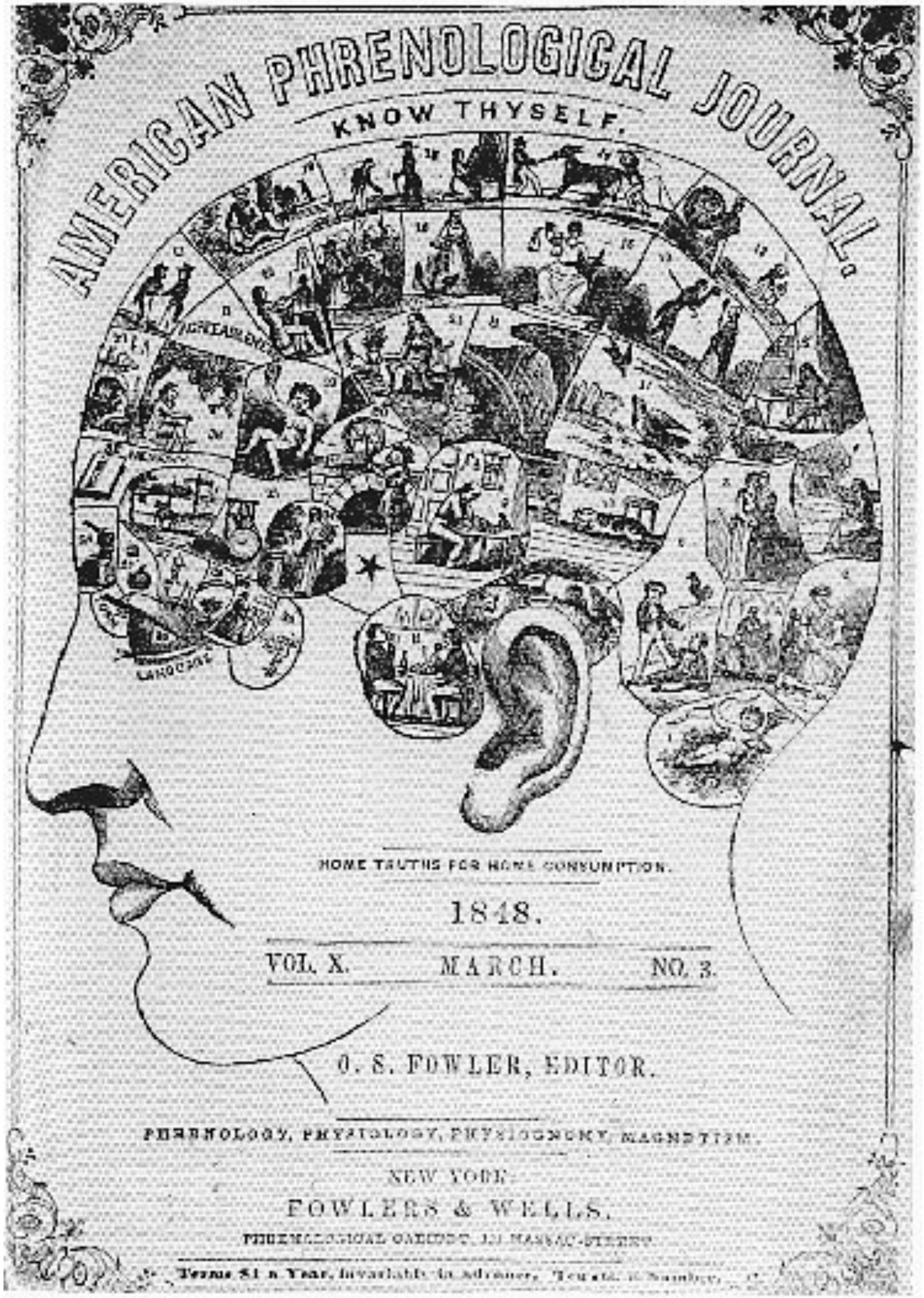
http://acl.cs.qc.edu/~lhuang/teaching/machine-learning

Outline

- Perceptron
  - Classification in Augmented Space
  - Perceptron Algorithm
  - Convergence Proof
- Extensions of Perceptron
  - Voted/Averaged, MIRA, passive-aggressive, *p*-aggressive MIRA
  - Multiclass Perceptron
- Features and preprocessing
  - Nonlinear separation
  - Perceptron in feature space
- Kernels
  - Kernel trick
  - Kernelized Perceptron in Dual (Kai)
  - Properties

Perceptron

Frank Rosenblatt



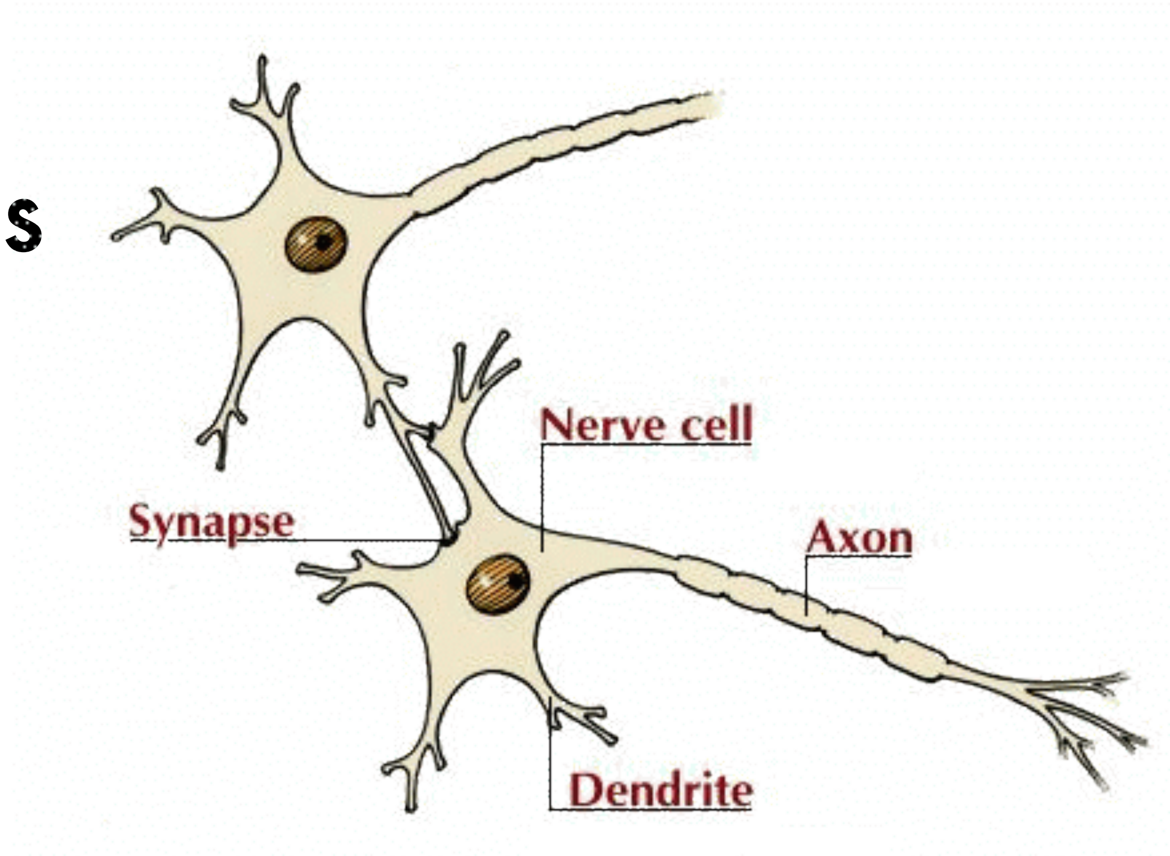
early theories of the brain

Biology and Learning

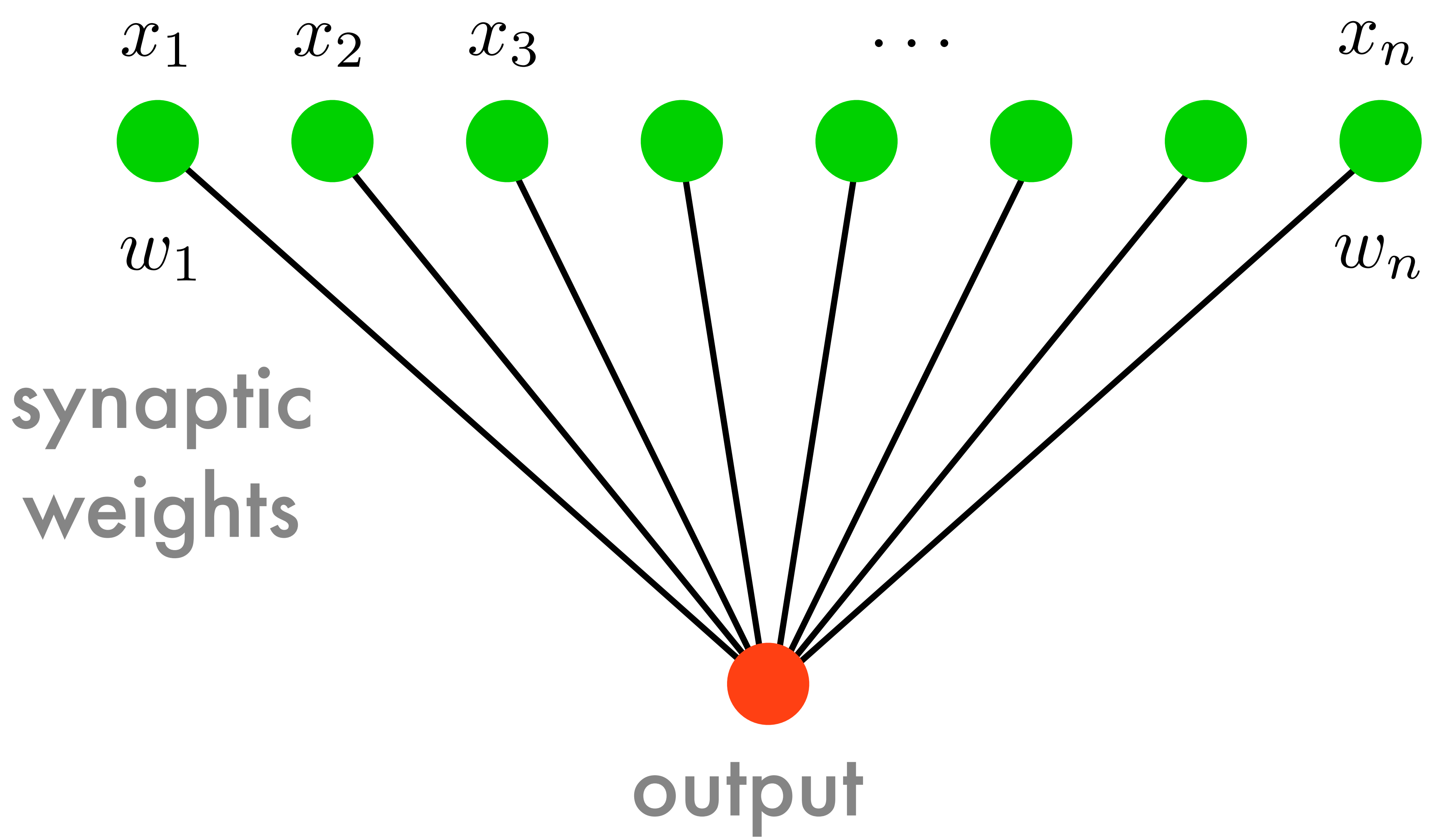
- Basic Idea
  - Good behavior should be rewarded, bad behavior punished (or not rewarded). This improves system fitness.
  - Killing a sabertooth tiger should be rewarded ...
  - Correlated events should be combined.
  - Pavlov's salivating dog.
- Training mechanisms
  - Behavioral modification of individuals (learning)
    Successful behavior is rewarded (e.g. food).
  - Hard-coded behavior in the genes (instinct)
    The wrongly coded animal does not reproduce.

Neurons

- Soma (CPU)
  Cell body - combines signals
- Dendrite (input bus)
  Combines the inputs from several other nerve cells
- Synapse (interface)
  Interface and parameter store between neurons
- Axon (output cable)
  May be up to 1m long and will transport the activation signal to neurons at different locations

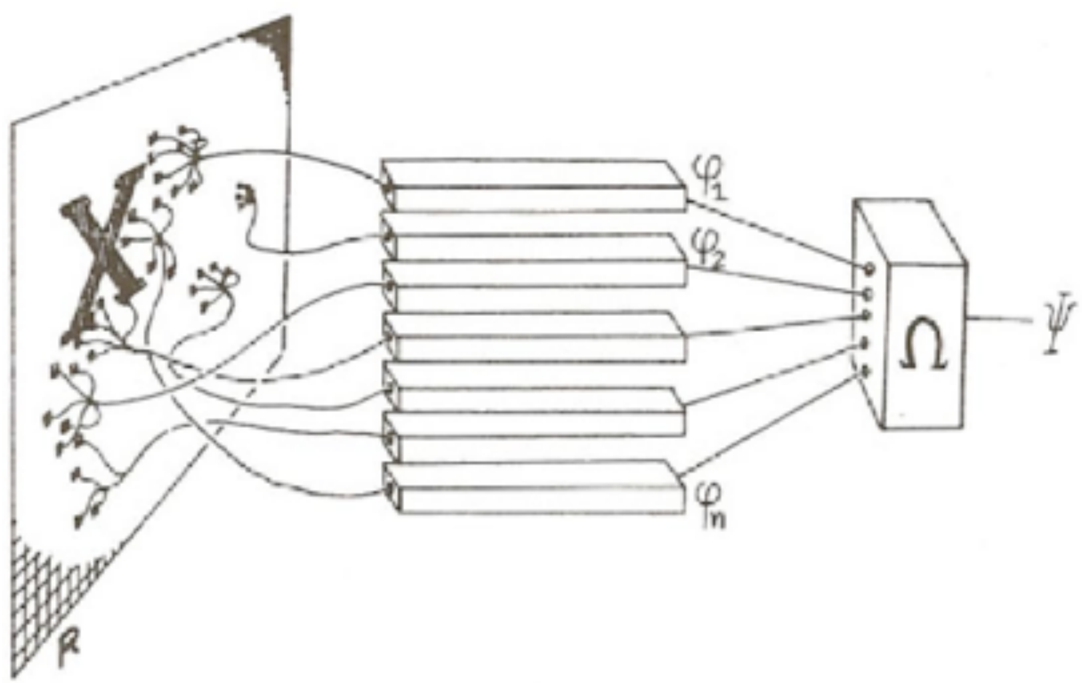


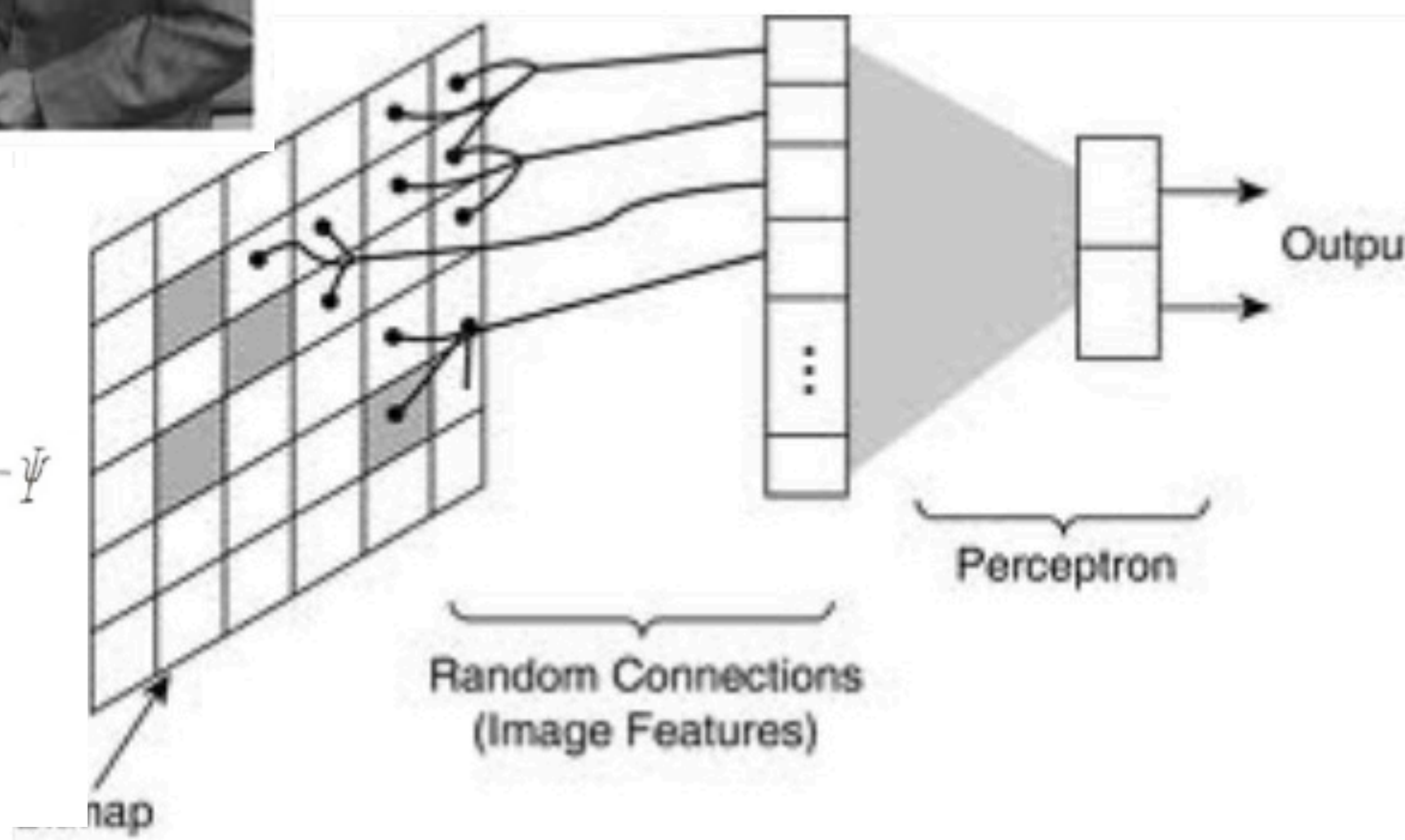
Neurons



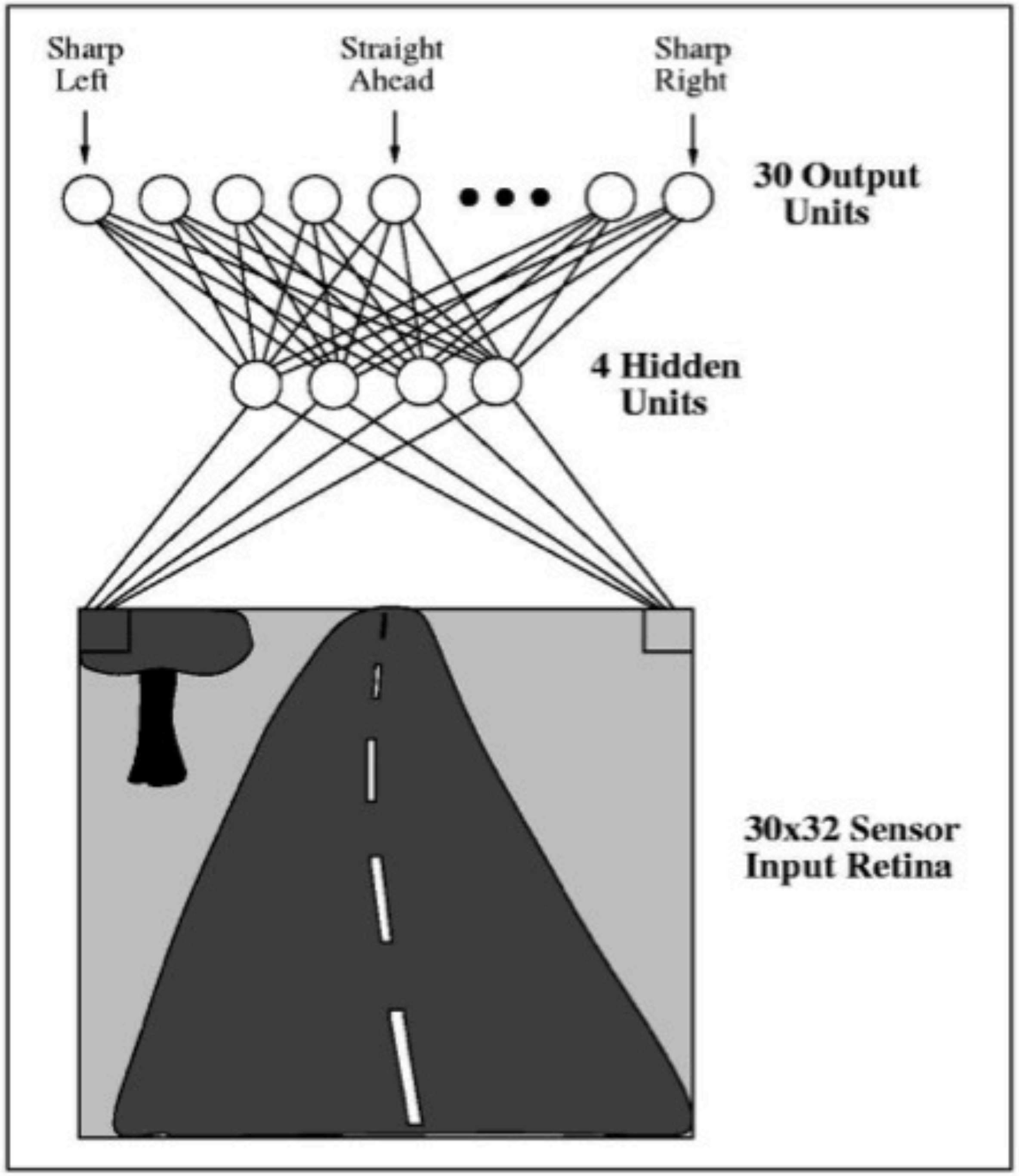
$$f(x) = \sum_i w_i x_i = \langle w, x \rangle$$

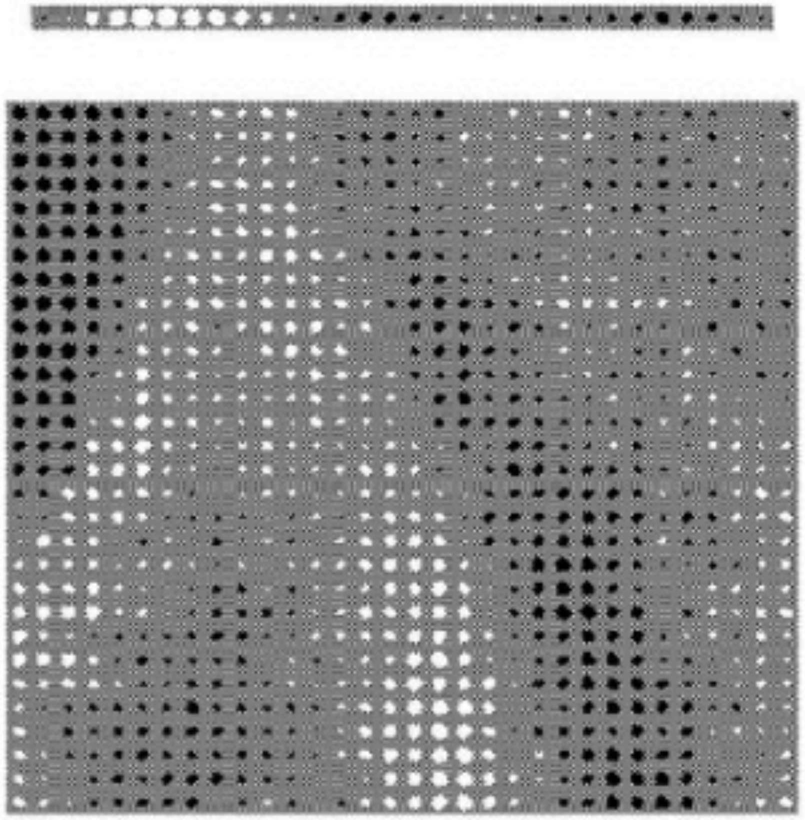
Frank Rosenblatt’s Perceptron





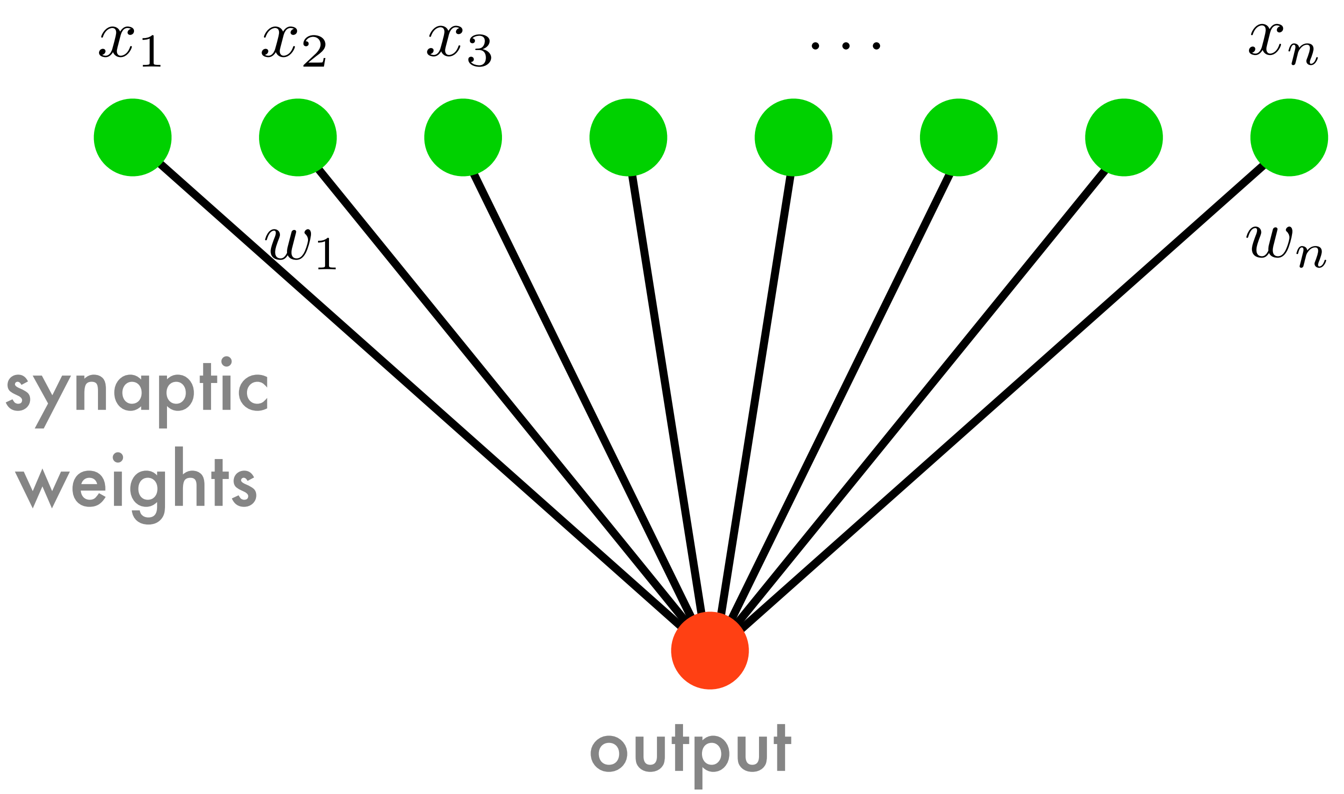
Multilayer Perceptron (Neural Net)





Perceptron w/ bias

- Weighted linear combination
- Nonlinear decision function
- Linear offset (bias)

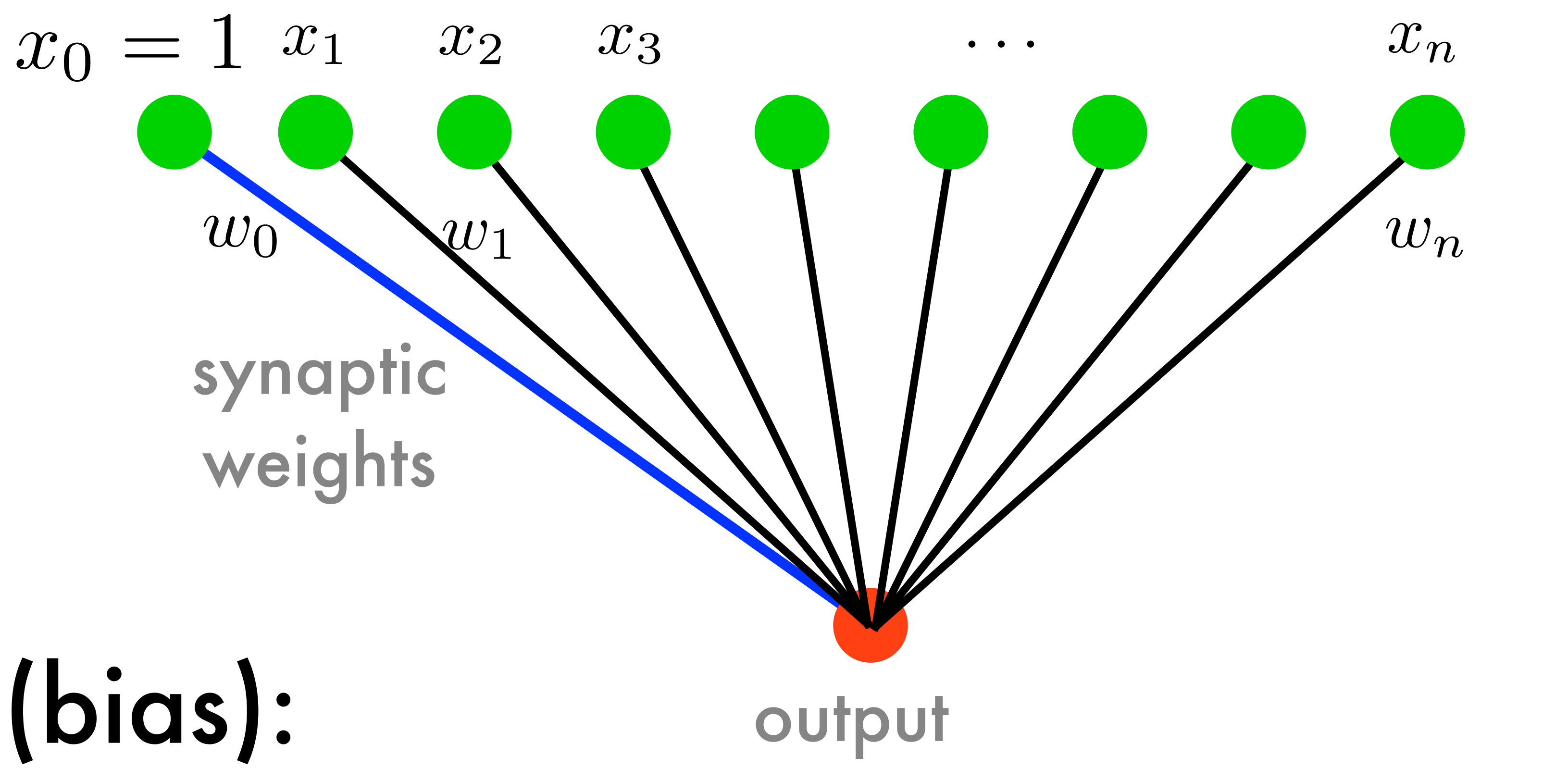


$$f(x) = \sigma\left(\langle w, x\rangle + b\right)$$

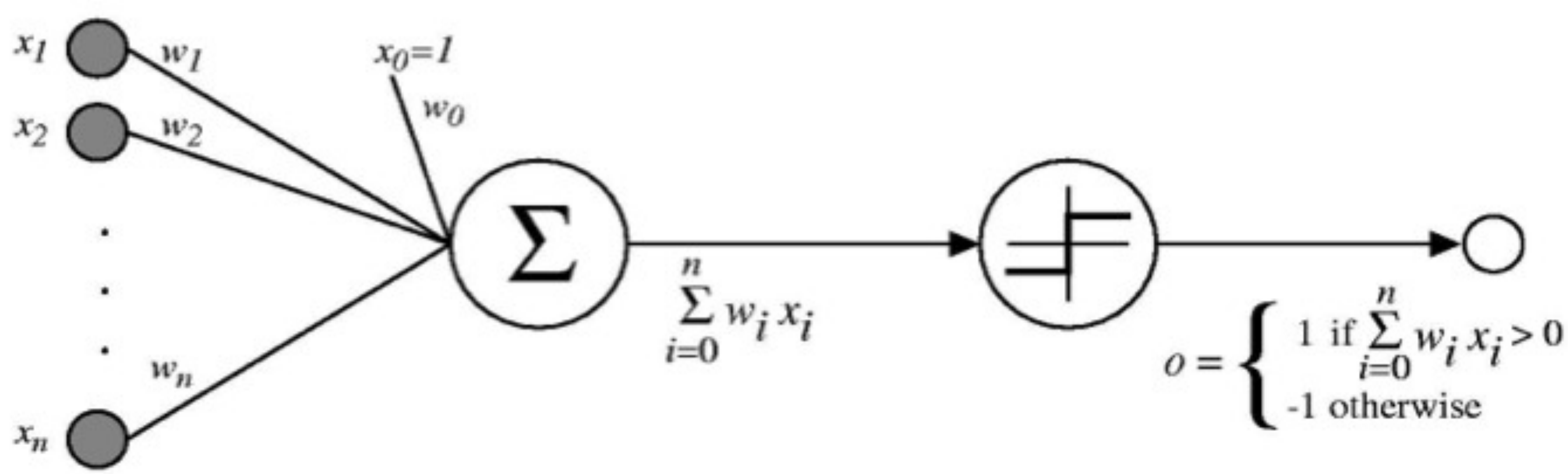
- Linear separating hyperplanes (spam/ham, novel/typical, click/no click)
- Learning: w and b

Perceptron w/o bias

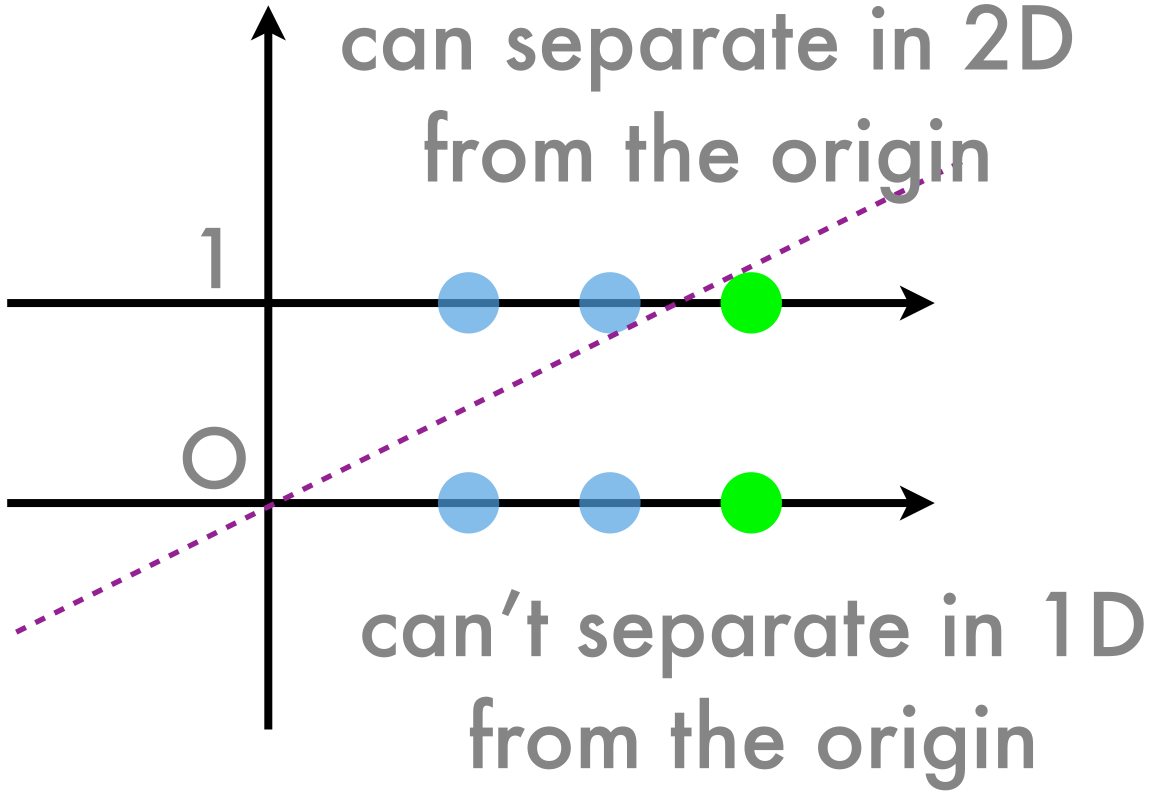
- Weighted linear combination
- Nonlinear decision function
- No Linear offset (bias): hyperplane through the origin
- Linear separa
  (spam/ham, n
- Learning: w

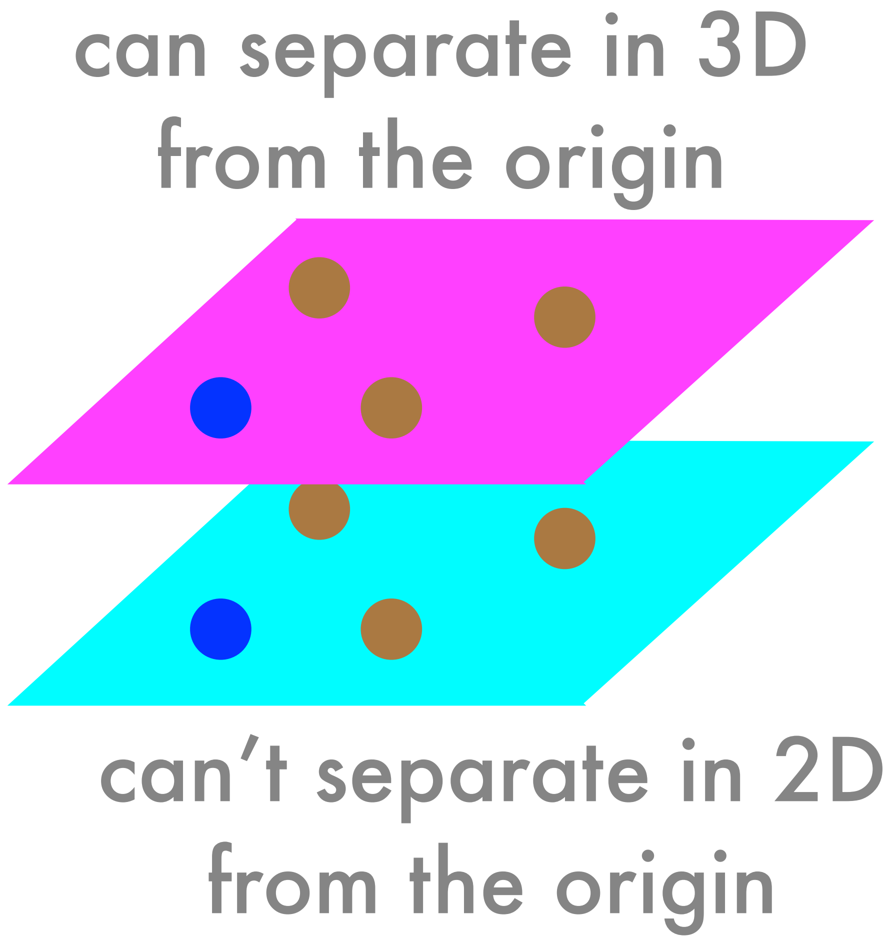


$$f(x) = \sigma\left(\langle w, x\rangle + b\right)$$

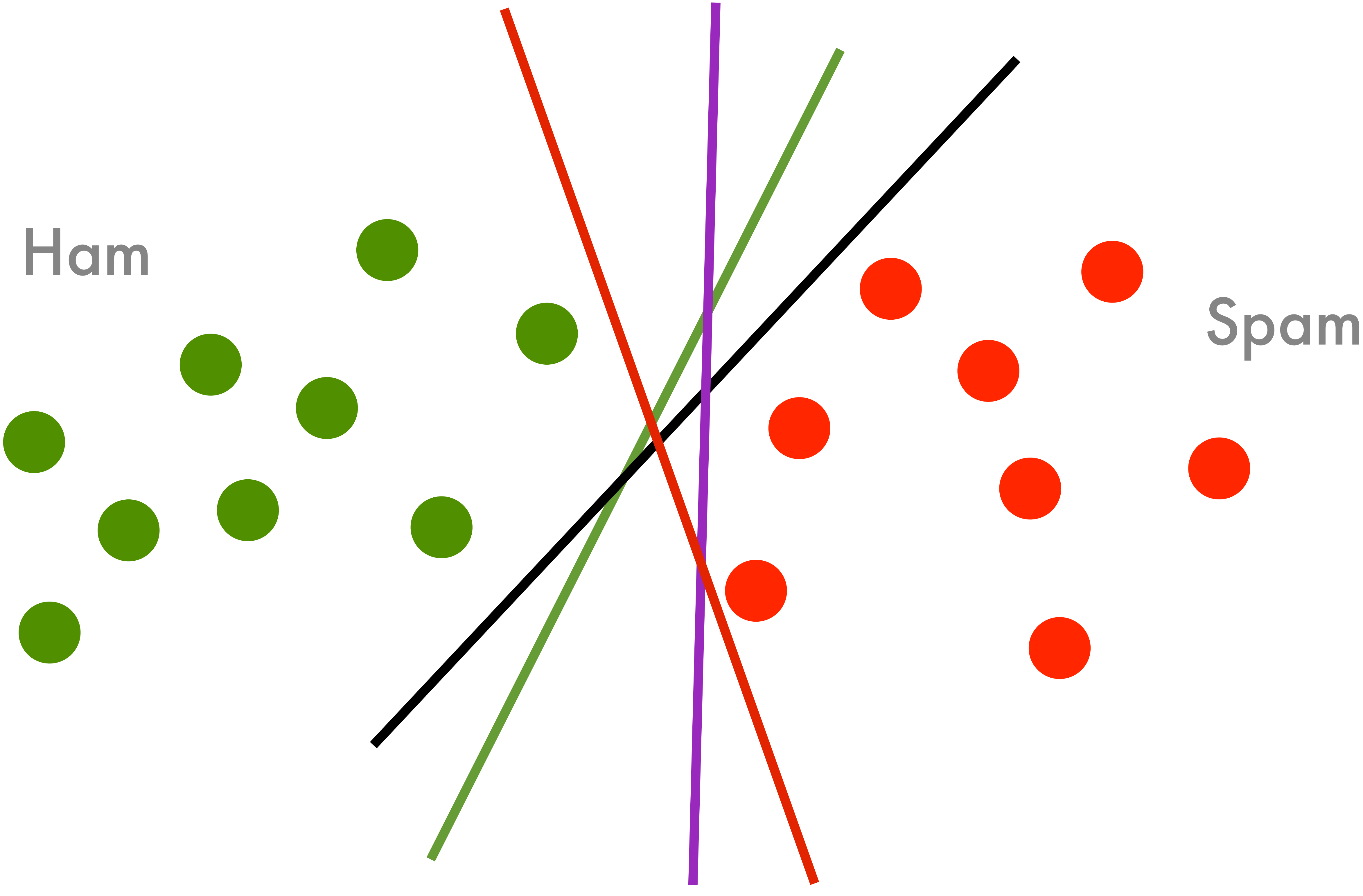


Augmented Space





Perceptron



The Perceptron w/o bias

initialize $w = 0$ and $b = 0$
repeat
  if $y_i \left[\langle w, x_i \rangle + b\right] \leq 0$ **then**
    $w \leftarrow w + y_i x_i$ and $b \leftarrow b + y_i$
  end if
until all classified correctly

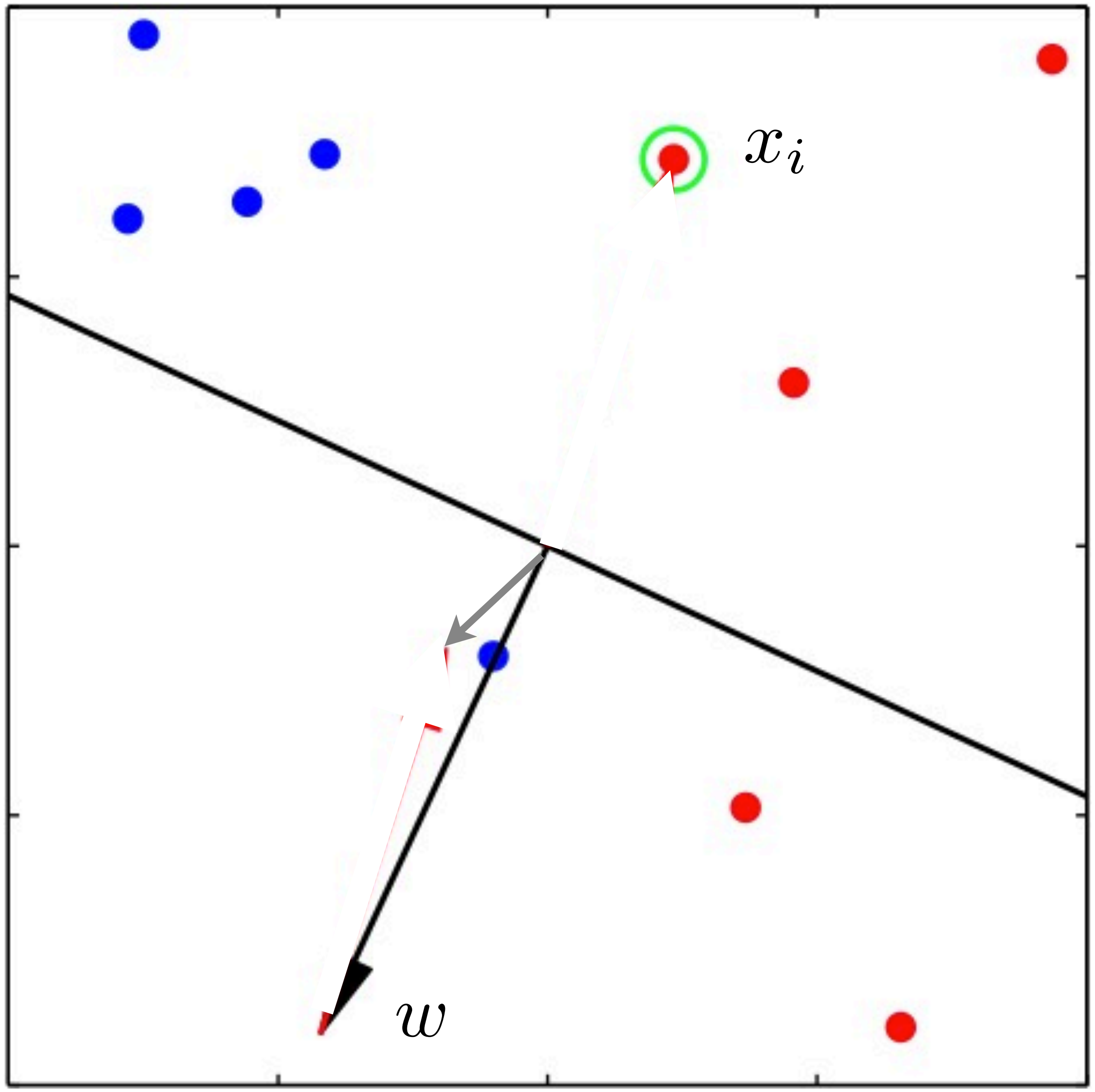
- Nothing happens if classified correctly
- Weight vector is linear combination $w = \sum_{i \in I} y_i x_i$
- Classifier is linear combination of inner products $f(x) = \sum_{i \in I} y_i \langle x_i, x \rangle + b$

The Perceptron w/ bias

initialize $w = 0$ and $b = 0$
repeat
  if $y_i \left[\langle w, x_i \rangle + b\right] \leq 0$ **then**
    $w \leftarrow w + y_i x_i$ and $b \leftarrow b + y_i$
  end if
until all classified correctly

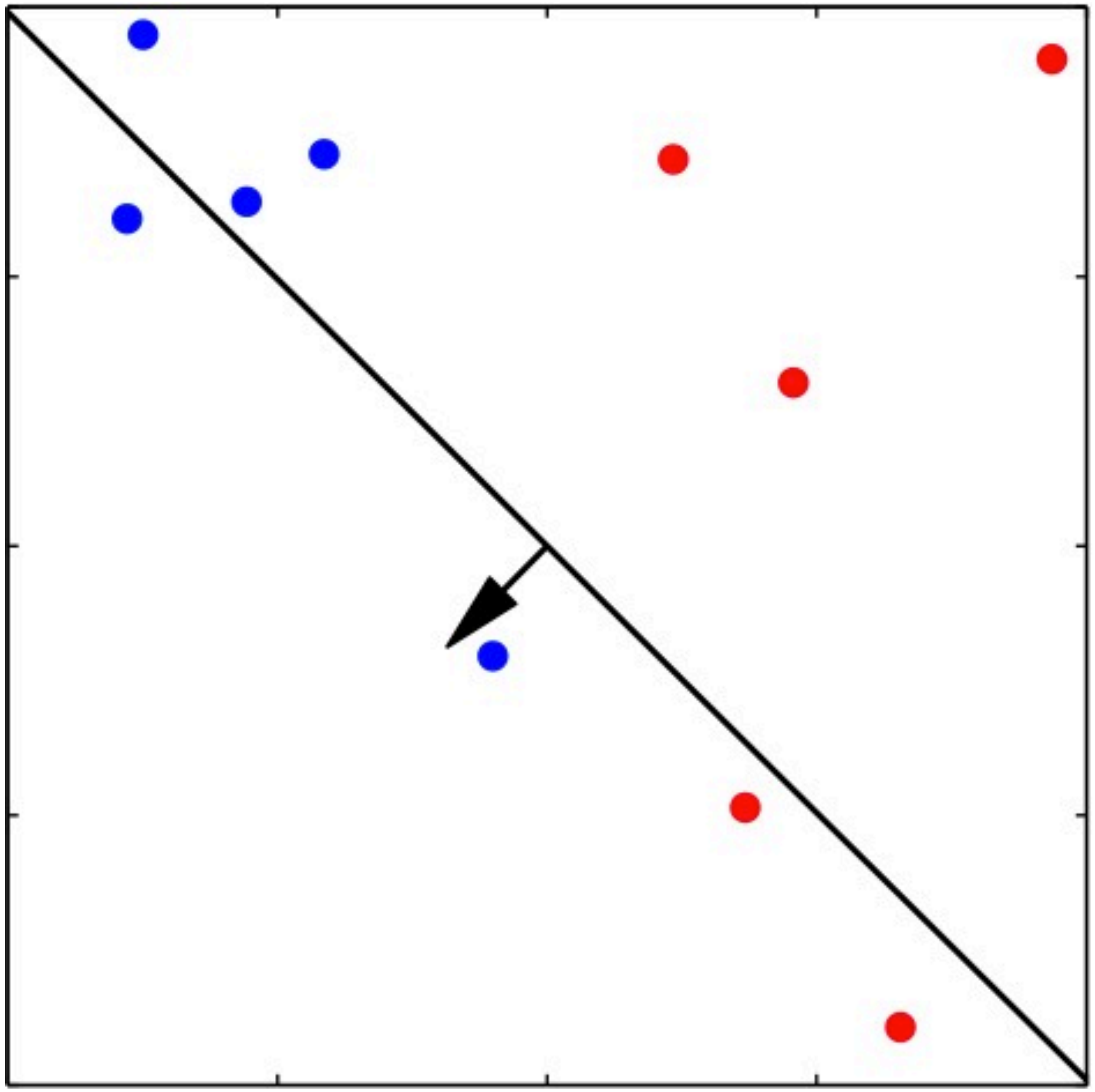
- Nothing happens if classified correctly
- Weight vector is linear combination $w = \sum_{i \in I} y_i x_i$
- Classifier is linear combination of inner products $f(x) = \sum_{i \in I} y_i \langle x_i, x \rangle + b$

Demo

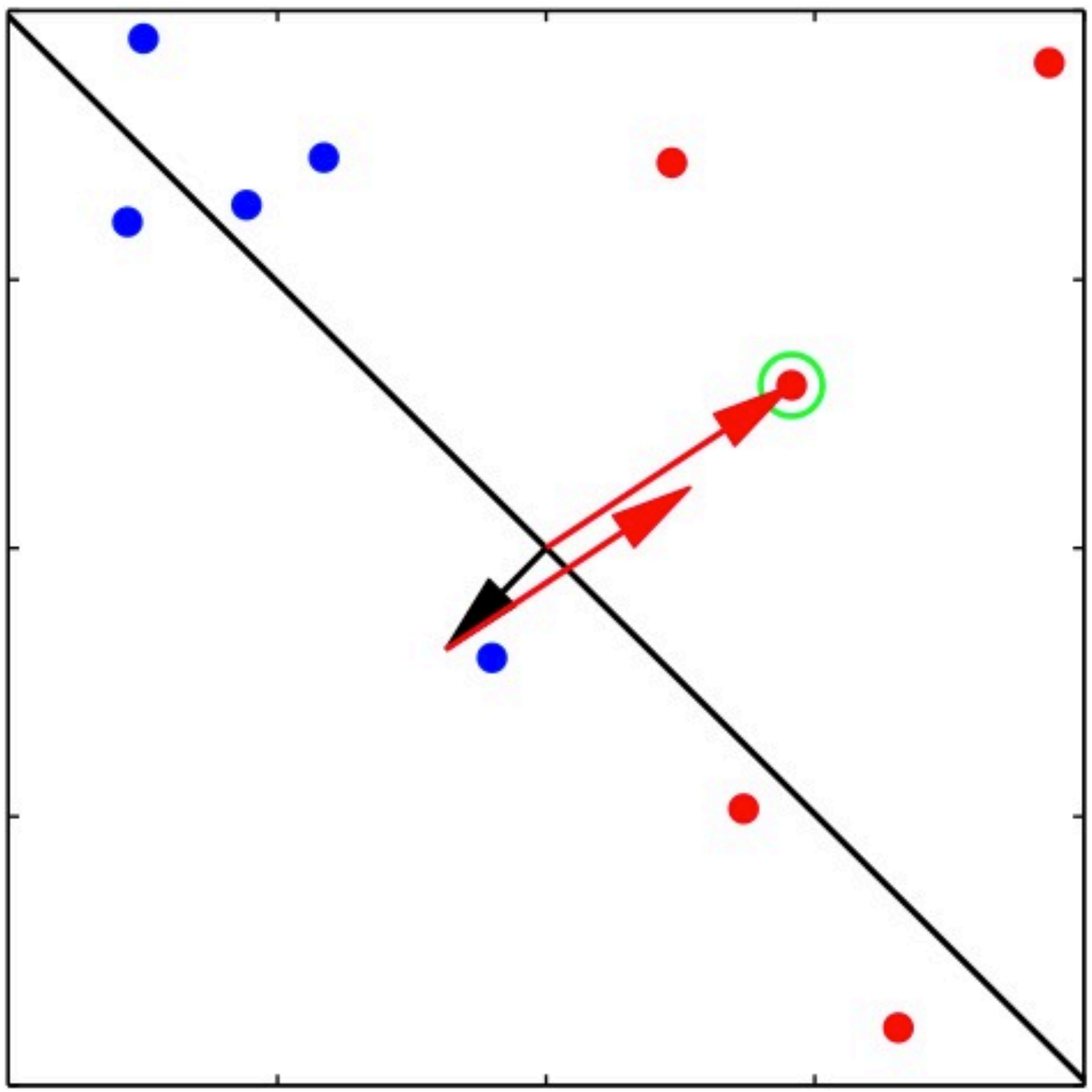


(bias=0)

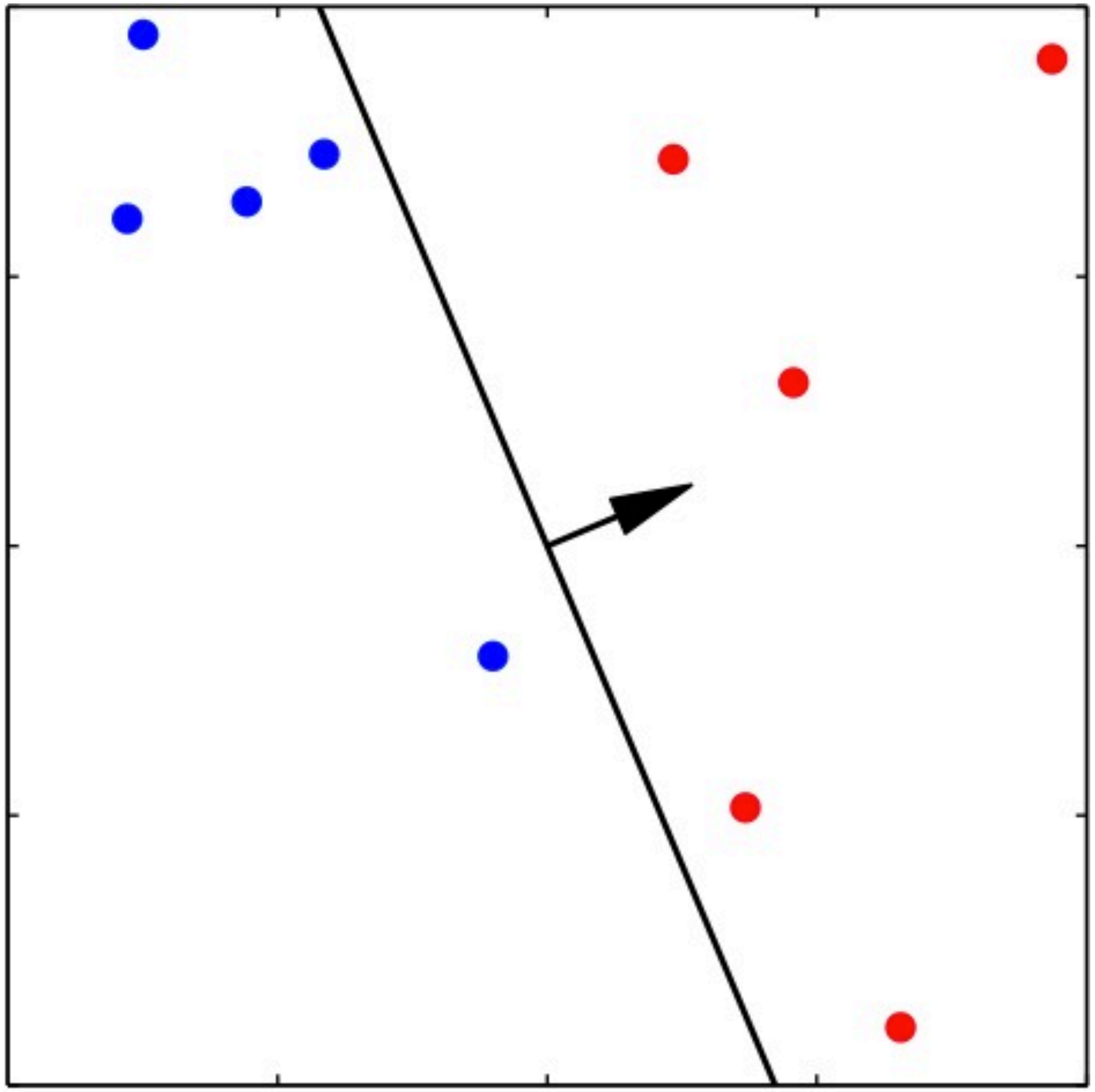
Demo

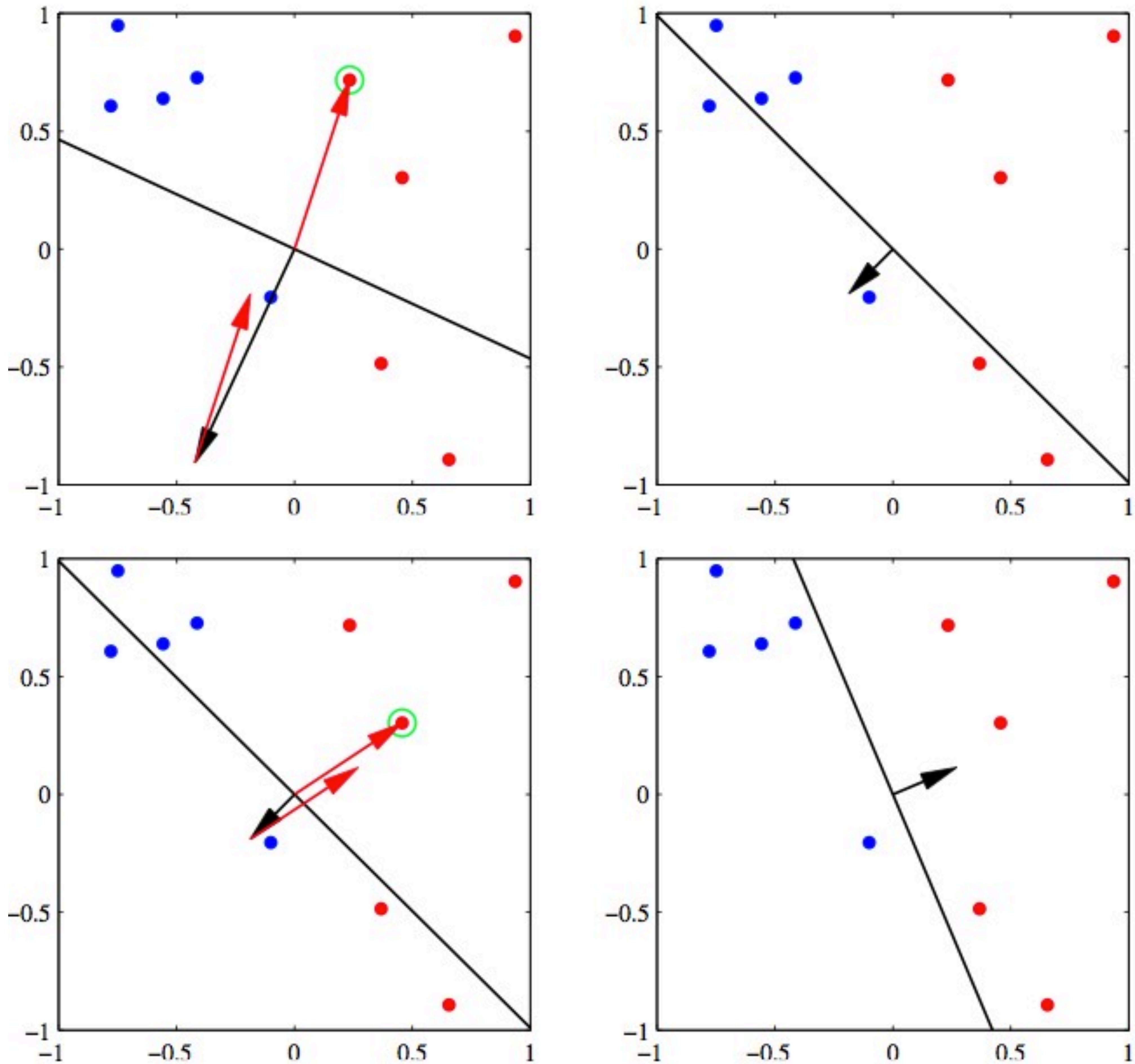


Demo



Demo





1
0.5
0
−0.5
−1
−1
−0.5
0
0.5
1

Convergence Theorem

- If there exists some oracle unit vector $u : \|u\| = 1$

$$y_i(u \cdot x_i) \geq \delta \text{ for all } i$$

then the perceptron converges to a linear separator after a number of steps bounded by

$$R^2/\delta^2 \text{ where } \; R = \max_i \|x_i\|$$

- Dimensionality independent
- Order independent (i.e. also worst case)
- Scales with 'difficulty' of problem

Geometry of the Proof

- **part 1: progress (alignment) on oracle projection**

assume w_i is the weight vector **before** the ith update (on $\langle x_i, y_i \rangle$) and assume initial $w_0 = 0$

$$w_{i+1} = w_i + y_i x_i$$

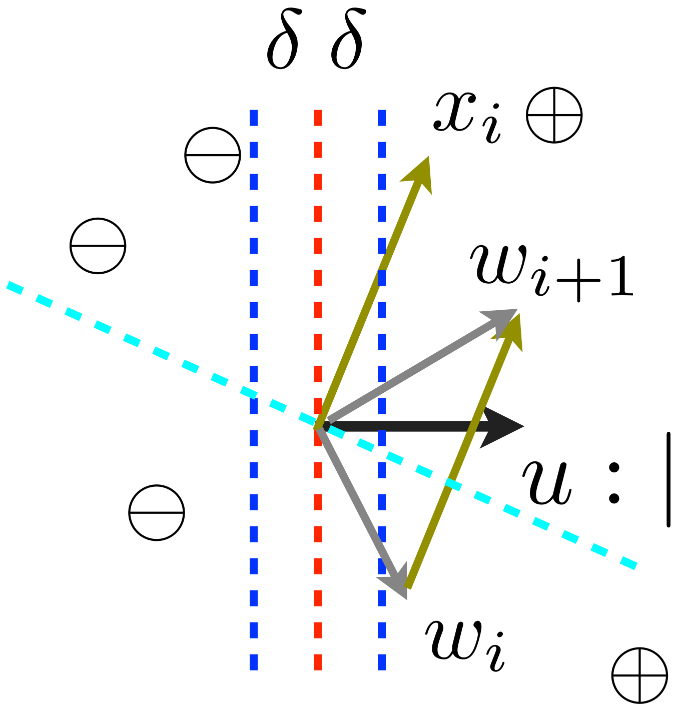
$$u \cdot w_{i+1} = u \cdot w_i + y_i (u \cdot x_i)$$

$$u \cdot w_{i+1} \geq u \cdot w_i + \delta$$

$$u \cdot w_{i+1} \geq i\delta$$

$y_i (u \cdot x_i) \geq \delta$ for all

projection on *u* increases!
(more agreement w/ oracle)



$$\|w_{i+1}\| = \|u\| \|w_{i+1}\| \geq u \cdot w_{i+1} \geq i\delta$$

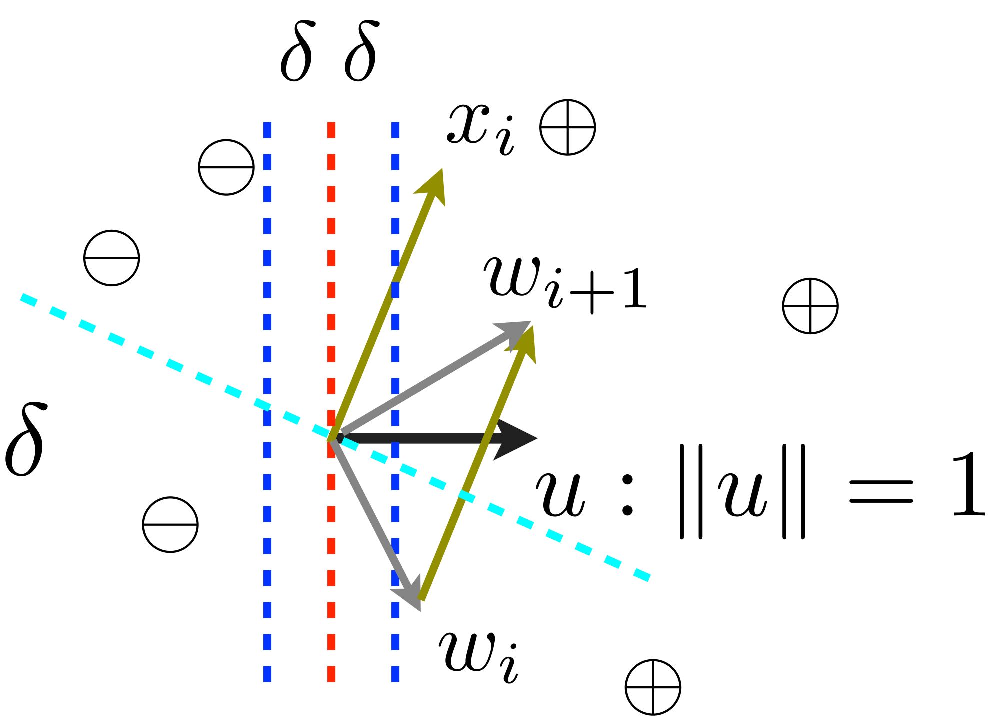
Geometry of the Proof

- part 2: bound the norm of the weight vector

$$
\begin{aligned}
w_{i+1} &= w_i + y_i x_i \\
\|w_{i+1}\|^2 &= \|w_i + y_i x_i\|^2 \\
&= \|w_i\|^2 + \|x_i\|^2 + 2y_i(w_i x_i) \\
&\leq \|w_i\|^2 + R^2 \\
&\leq iR^2
\end{aligned}
$$

"mistake on x_i"

(radius)



Combine with part 1

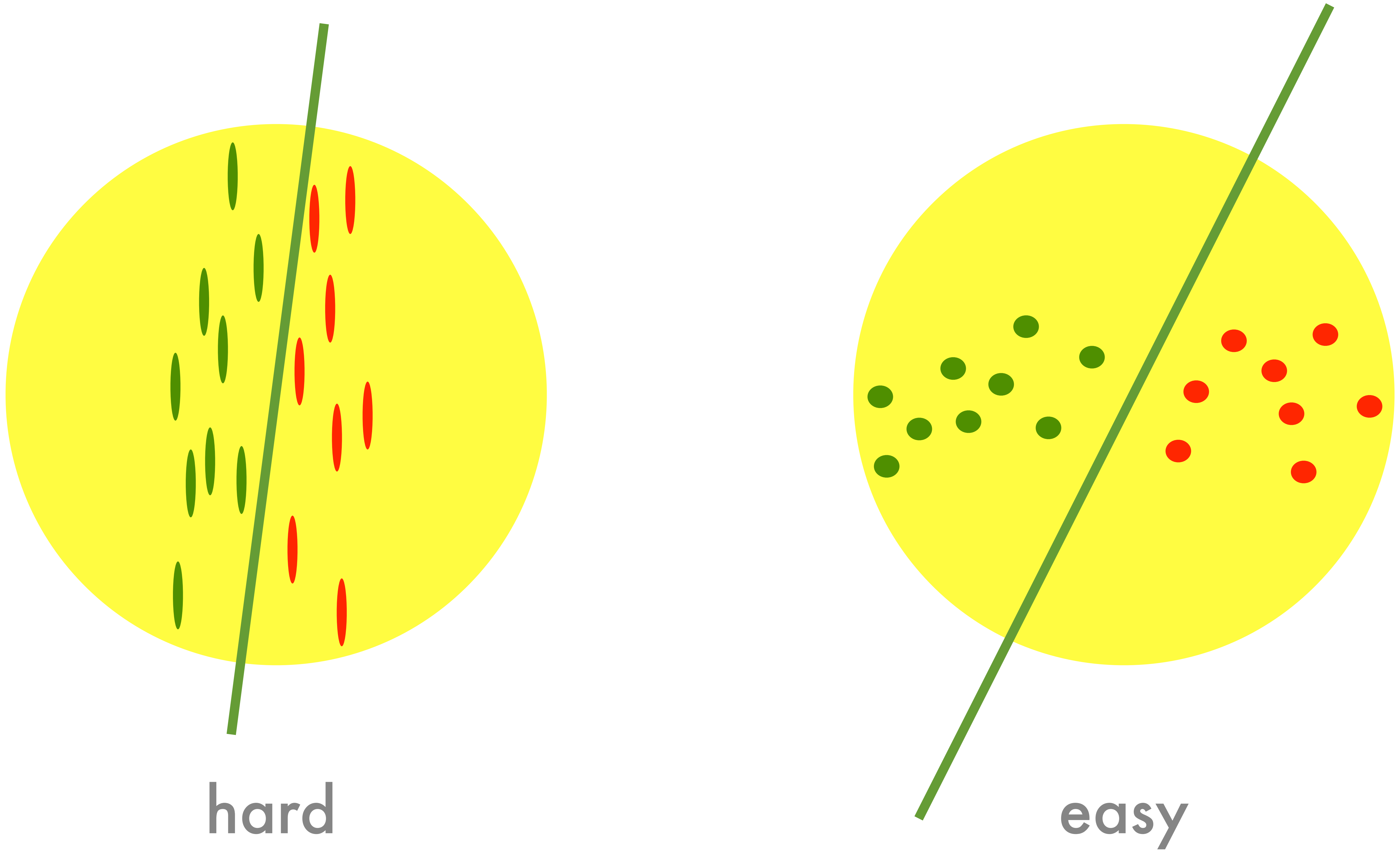
$$\|w_{i+1}\| = \|u\|\|w_{i+1}\| \geq u \cdot w_{i+1} \geq i\delta$$

$$i \leq R^2/\delta^2$$

Convergence Bound R^2/δ^2

- is independent of:
  - dimensionality
  - number of examples
  - starting weight vector
  - order of examples
  - constant learning rate
- and is dependent of:
  - separation difficulty
  - feature scale
- but test accuracy is dependent of:
  - order of examples (shuffling helps)
  - variable learning rate (1/total#error helps)
    - can you still prove convergence?

Hardness
margin vs. size



Consequences

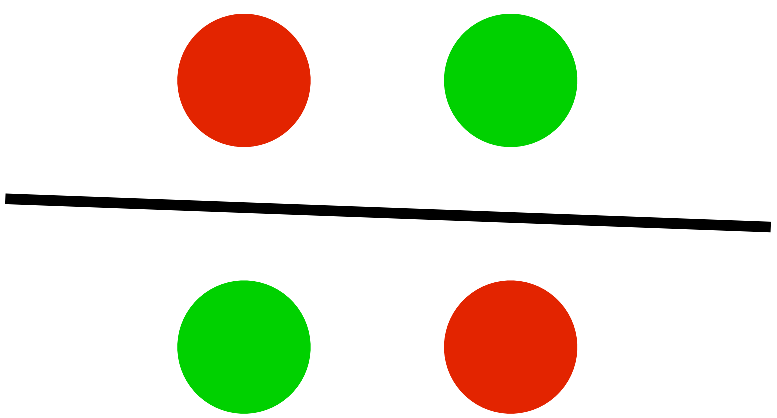
- Only need to store errors.
  This gives a compression bound for perceptron.
- Stochastic gradient descent on hinge loss

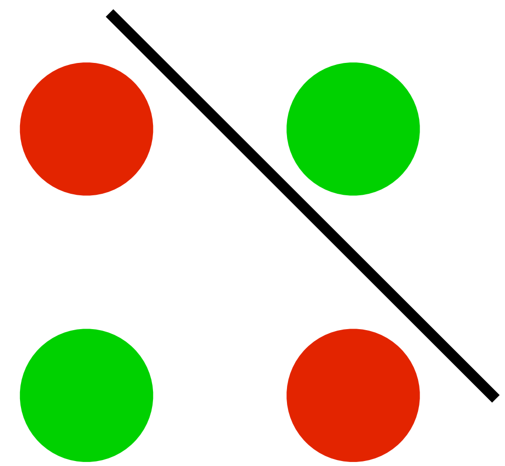
$$l(x_i, y_i, w, b) = \max\left(0, 1 - y_i \left[\langle w, x_i \rangle + b\right]\right)$$

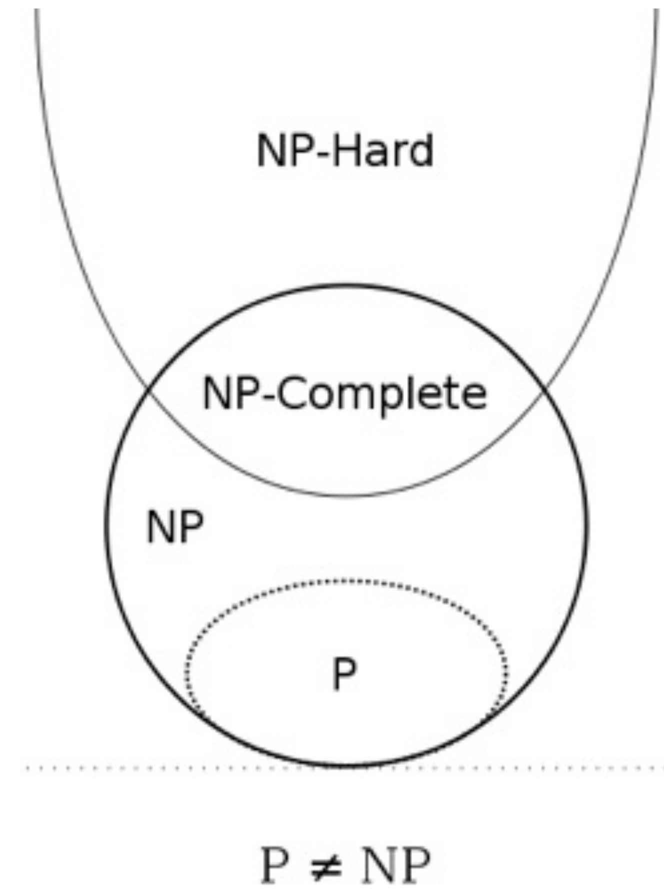
- Fails with noisy data

do NOT train your avatar with perceptrons

XOR

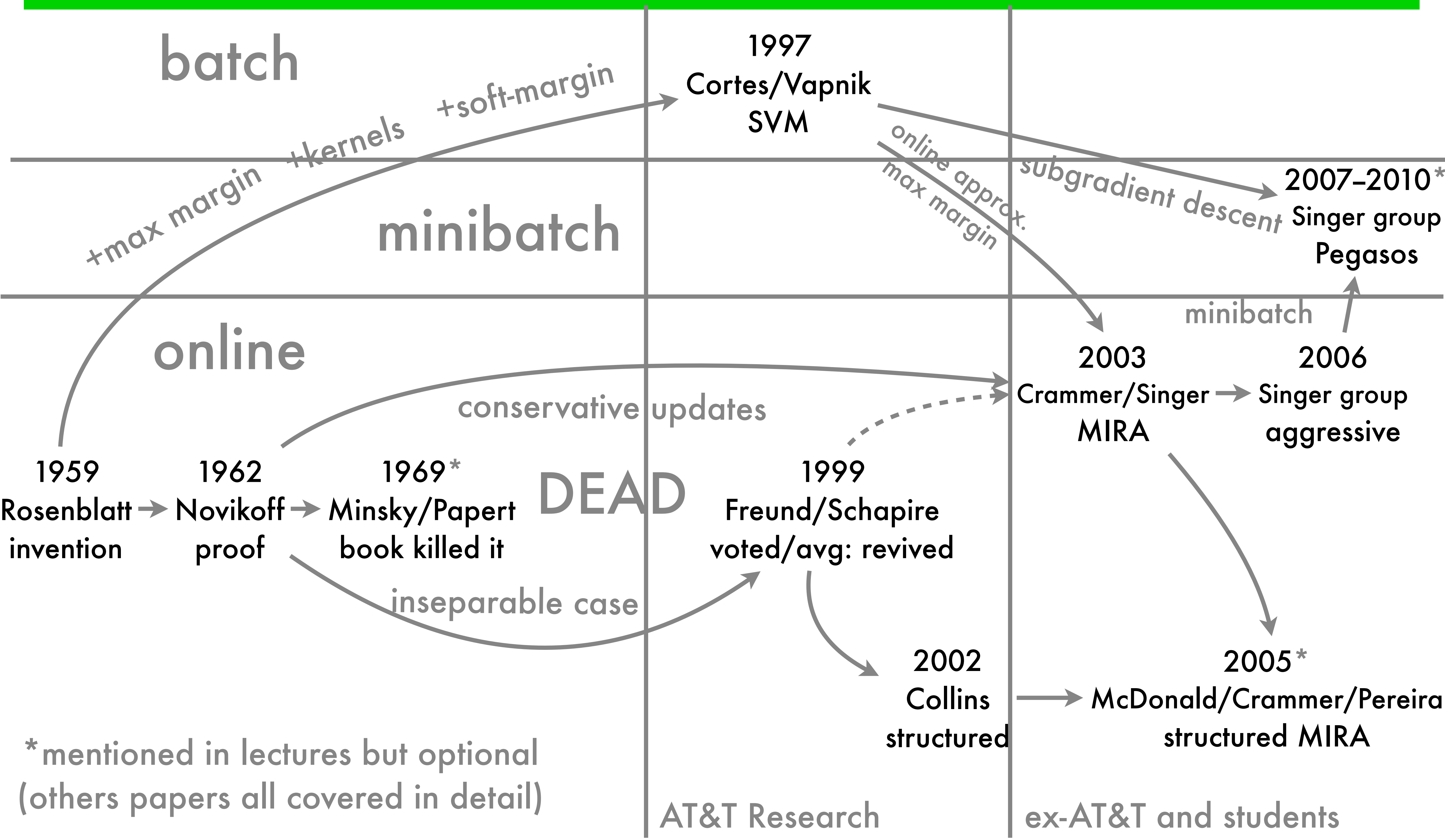






- XOR - not linearly separable
- Nonlinear separation is trivial
- Caveat from "Perceptrons" (Minsky & Papert, 1969) Finding the minimum error linear separator is NP hard (this killed Neural Networks in the 70s).

Brief History of Perceptron



*mentioned in lectures but optional
(others papers all covered in detail)

Extensions of Perceptron

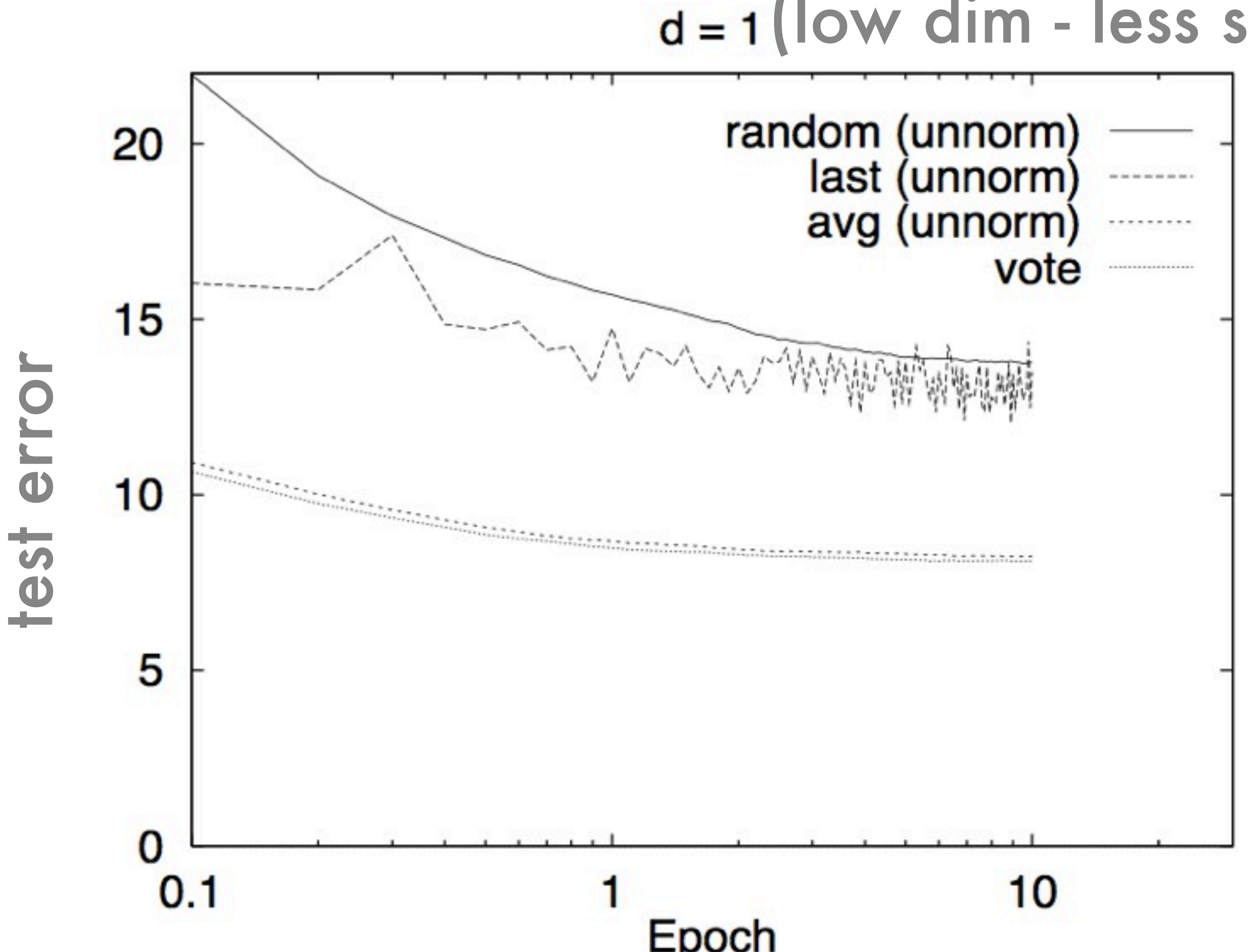
- Problems with Perceptron
  - doesn’t converge with inseparable data
  - update might often be too “bold”
  - doesn’t optimize margin
  - is sensitive to the order of examples
- Ways to alleviate these problems
  - voted perceptron and average perceptron
  - MIRA (margin-infused relaxation algorithm)
  - passive-aggressive

Voted/Avged Perceptron

- motivation: updates on later examples taking over!
- voted perceptron (Freund and Schapire, 1999)
  - record the weight vector after each example
    - (not just after each update)
  - and vote on a new example
  - shown to have better generalization power
- averaged perceptron (from the same paper)
  - an approximation of voted perceptron
  - just use the average of all weight vectors
  - can be implemented efficiently

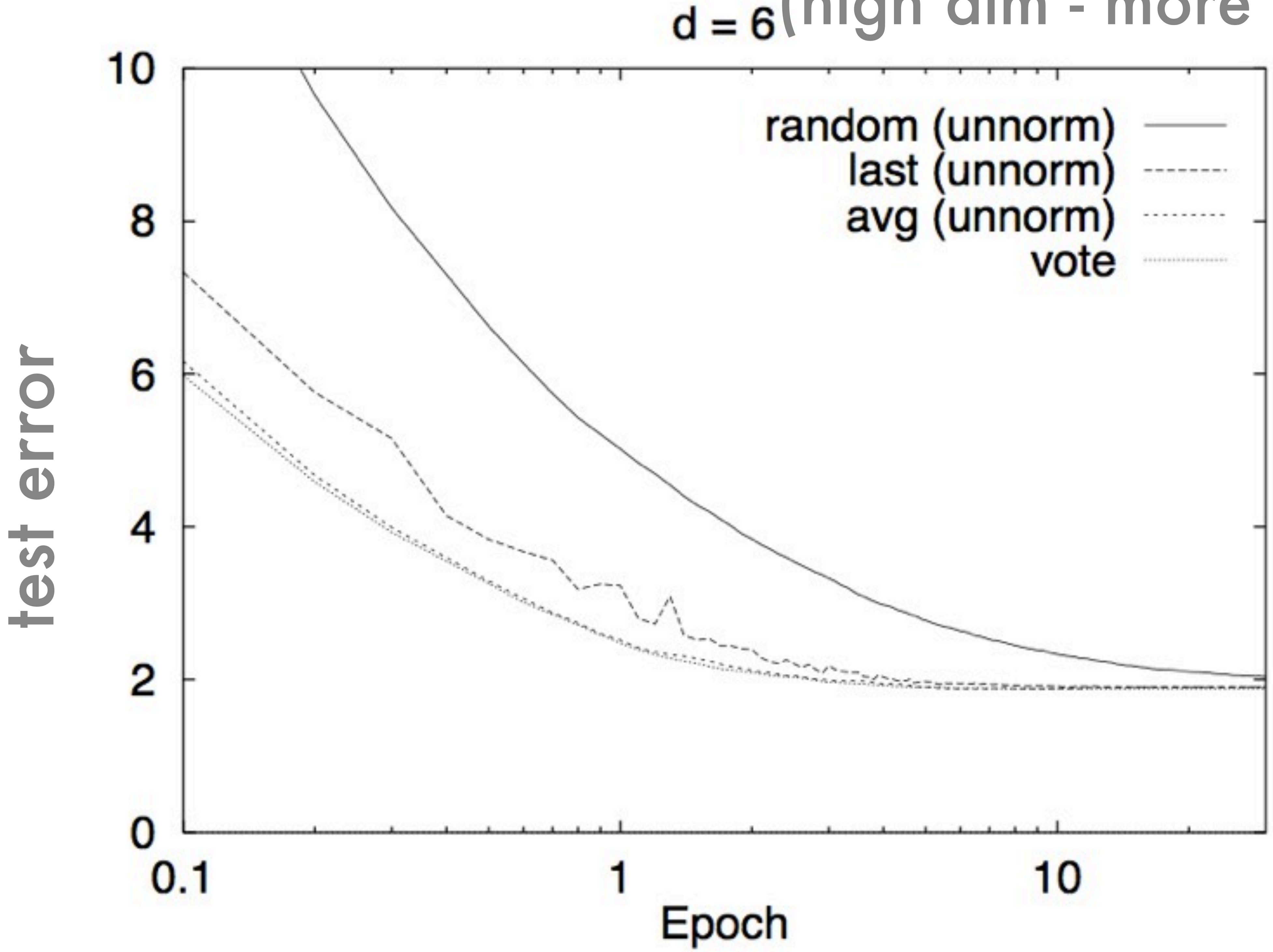
Voted/Avged Perceptron

(low dim - less separable)



Voted/Avged Perceptron

(high dim - more separable)



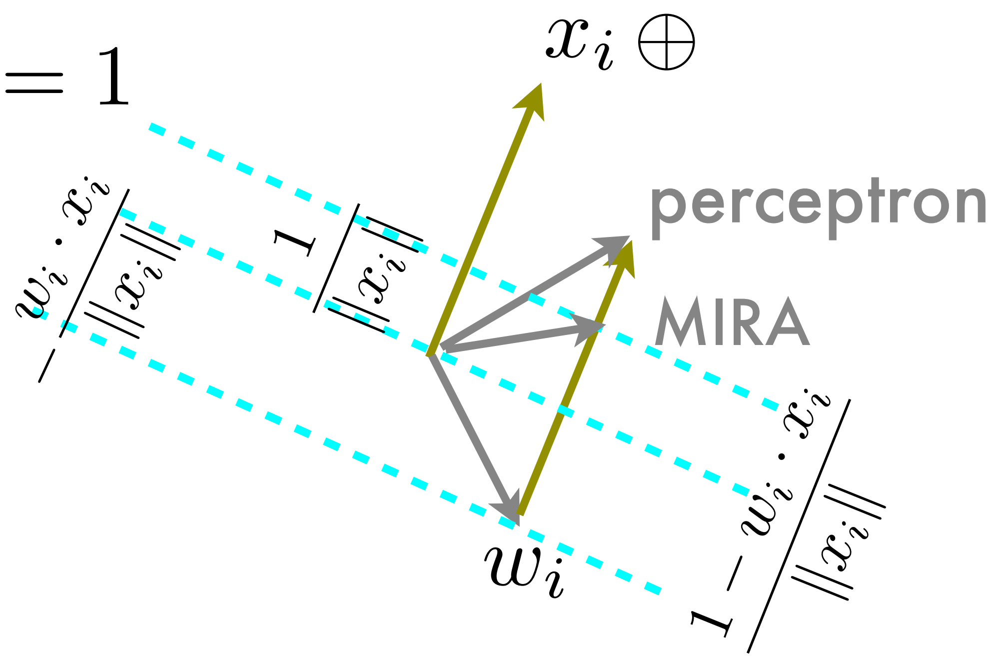
MIRA

- perceptron often makes too bold updates
  - but hard to tune learning rate
- the smallest update to correct the mistake?

$$w_{i+1} = w_i + \frac{y_i - w_i \cdot x_i}{\|x_i\|^2} x_i$$

easy to show:

$$y_i(w_{i+1} \cdot x_i) = y_i \left(w_i + \frac{y_i - w_i \cdot x_i}{\|x_i\|^2} x_i\right) \cdot x_i = 1$$



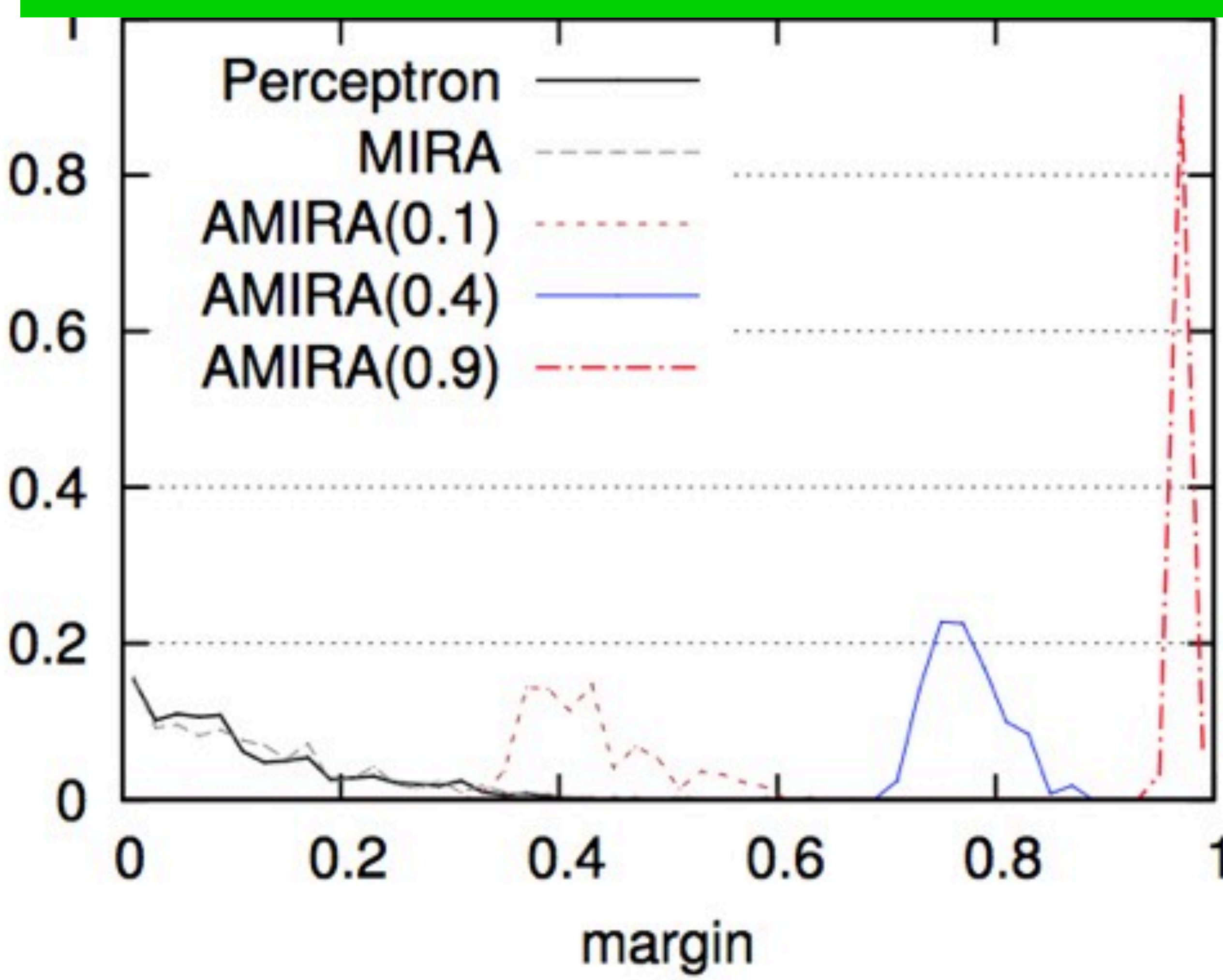
Aggressive MIRA (AMIRA)

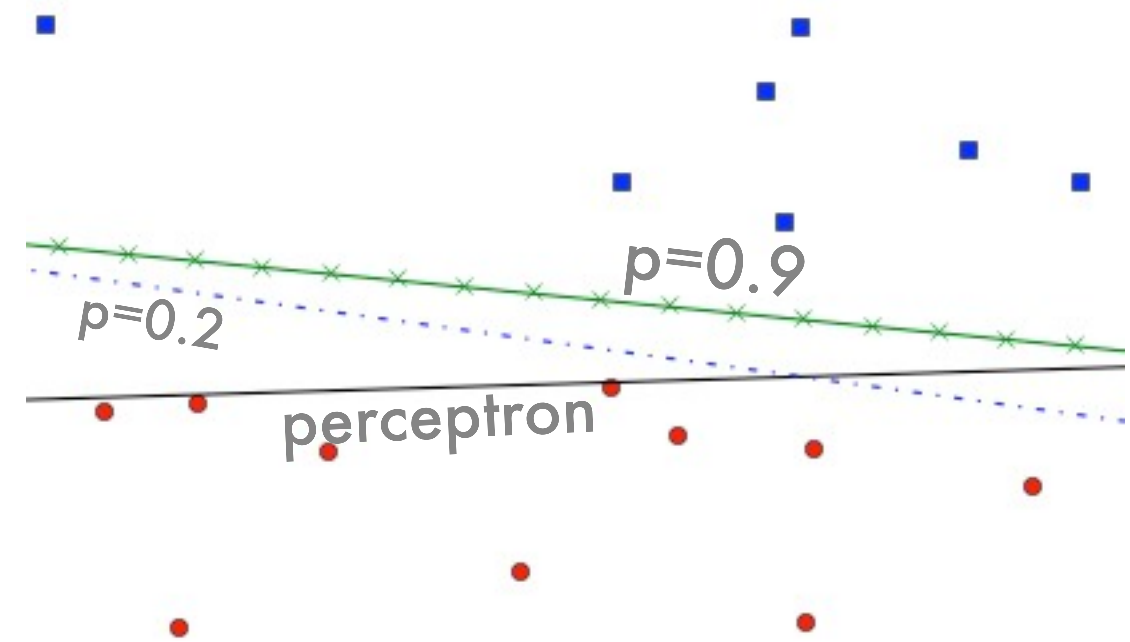
- aggressive version of MIRA
  - also update if correct but margin not big enough
- functional margin: $y_i(\mathbf{w} \cdot x_i)$
- geometric margin: $\frac{y_i(\mathbf{w} \cdot x_i)}{\|\mathbf{w}\|}$

what if we replace functional here by geometric?

- update if functional margin is <=*p* (0<=*p*<1)
  - update rule is same as MIRA
  - called AMIRA*p* or *p*-aggressive MIRA. (MIRA: *p*=0)
- larger *p* leads to a larger geometric margin
  - but slower convergence

Aggressive MIRA (AMIRA)





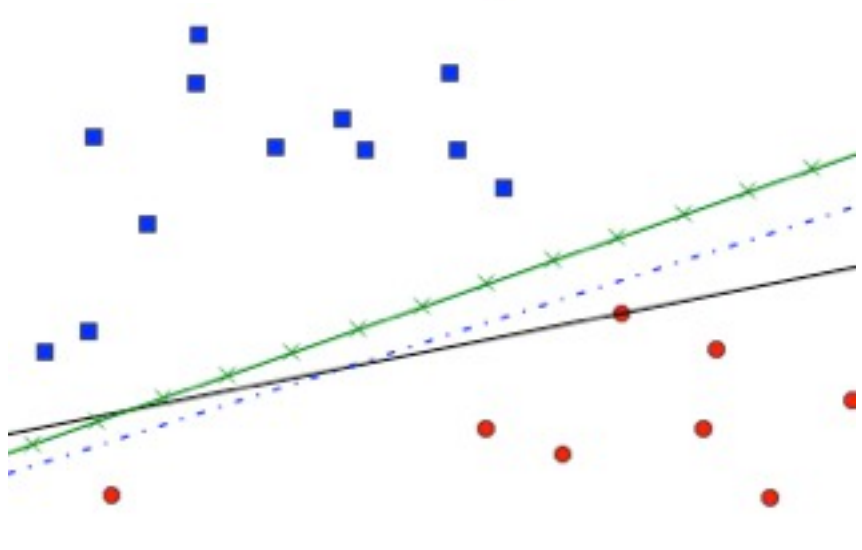
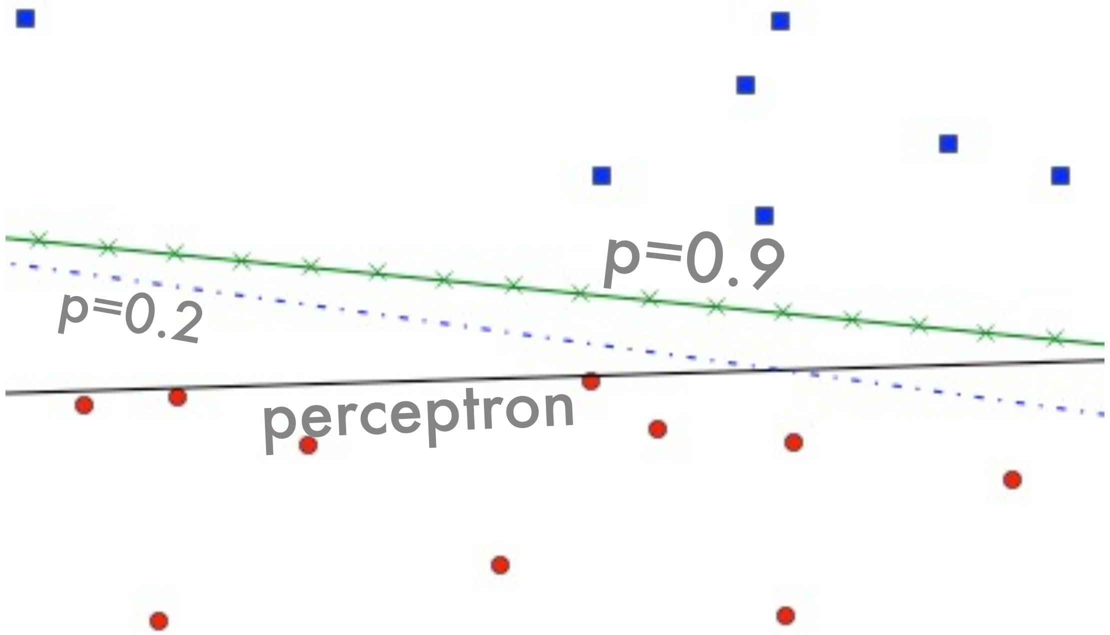


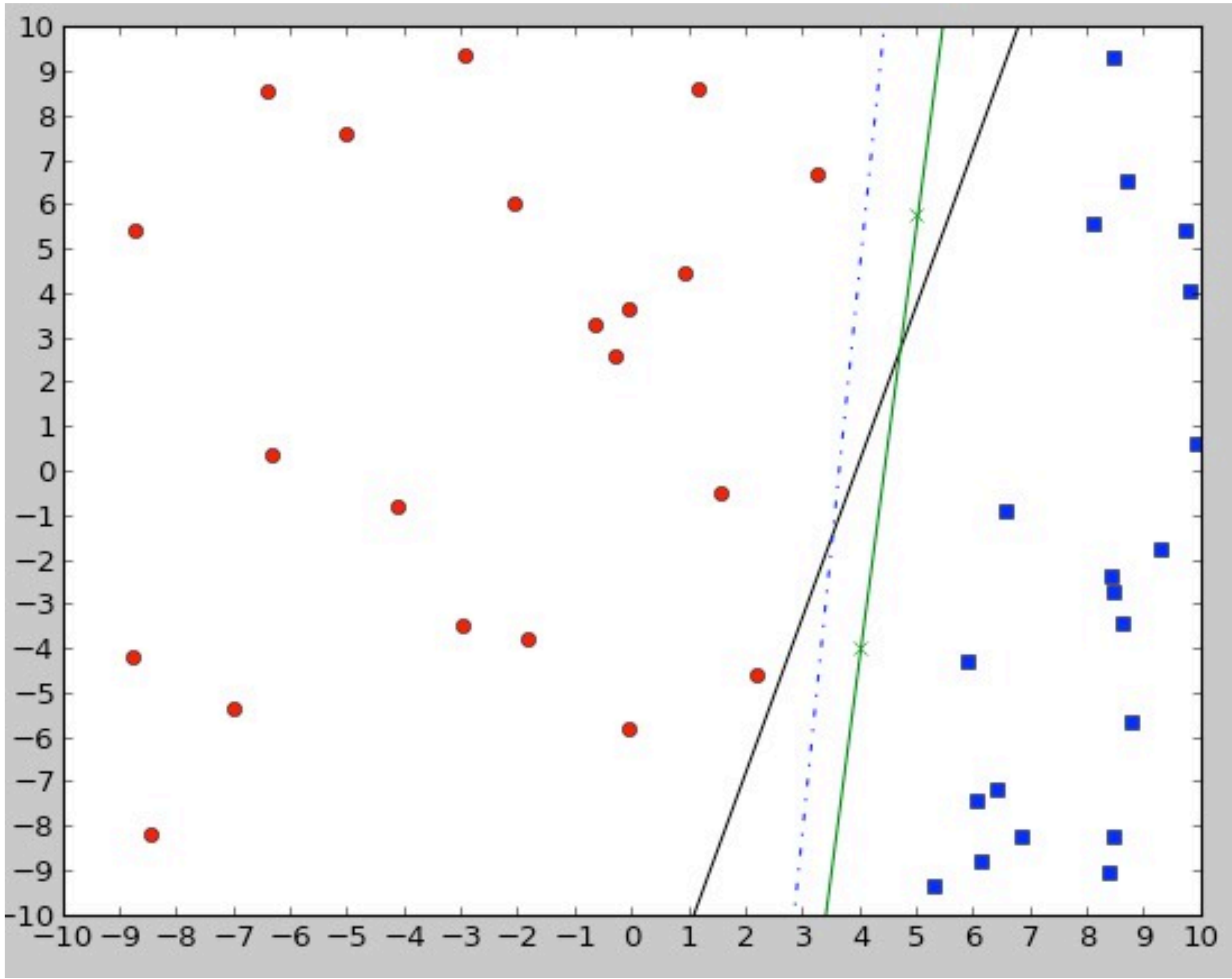
Table 3. Error rates on MNIST dataset. Both ROMMA and Aggressive ROMMA use a scale of 1100. The numbers in parentheses denote the aggressive parameters for AMIRA.

Epoch	1	2	3	4
Perceptron	2.98%	2.32%	1.94%	1.88%
Perceptron(avg.)	2.16%	1.85%	1.73%	1.69%
ROMMA	2.48%	1.96%	1.79%	1.77%
aggr-ROMMA	**2.14%**	1.82%	1.71%	1.67%
MIRA	2.56%	2.03%	1.74%	1.70%
bin AMIRA(0.1)	2.20%	**1.78%**	**1.67%**	**1.64%**

Demo

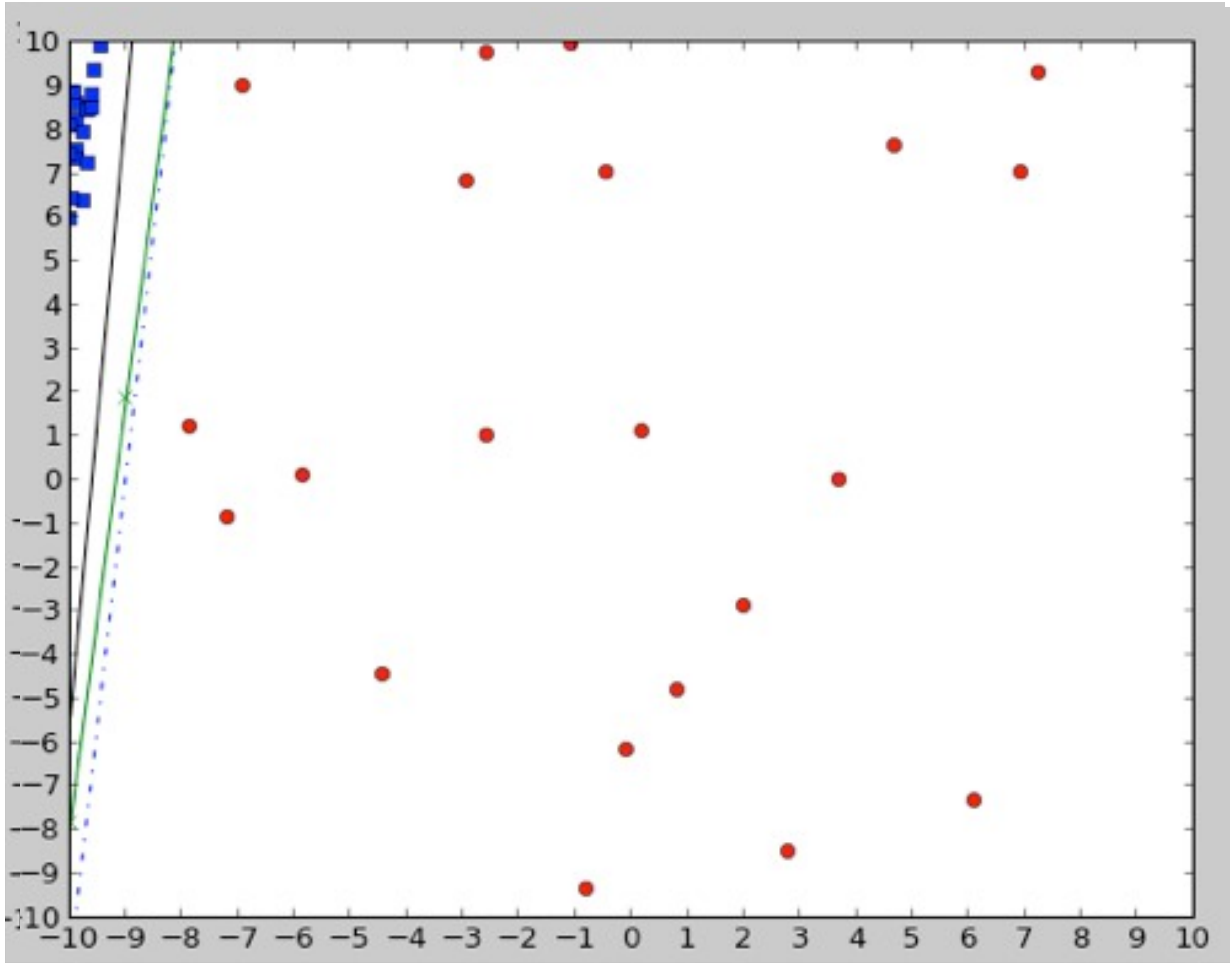
- perceptron vs. 0.2-aggressive vs. 0.9-aggressive



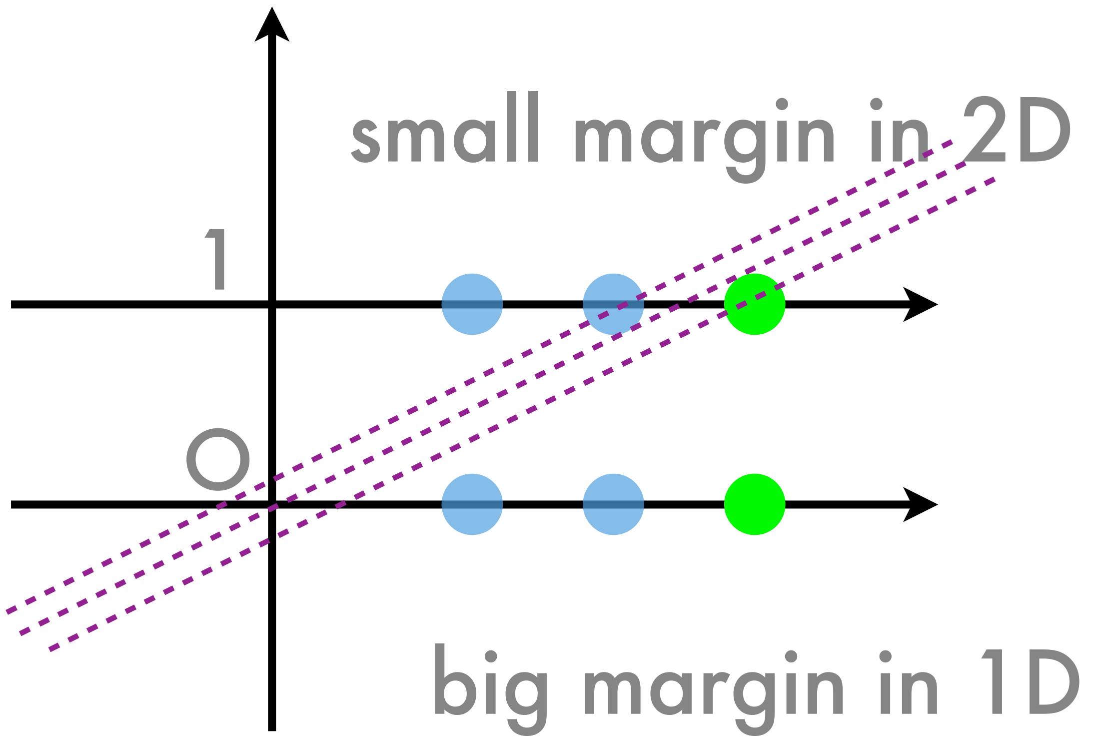


Demo

- perceptron vs. 0.2-aggressive vs. 0.9-aggressive
- why does this dataset so slow to converge?
  - perceptron: 22, p=0.2: 87, p=0.9: 2,518 epochs

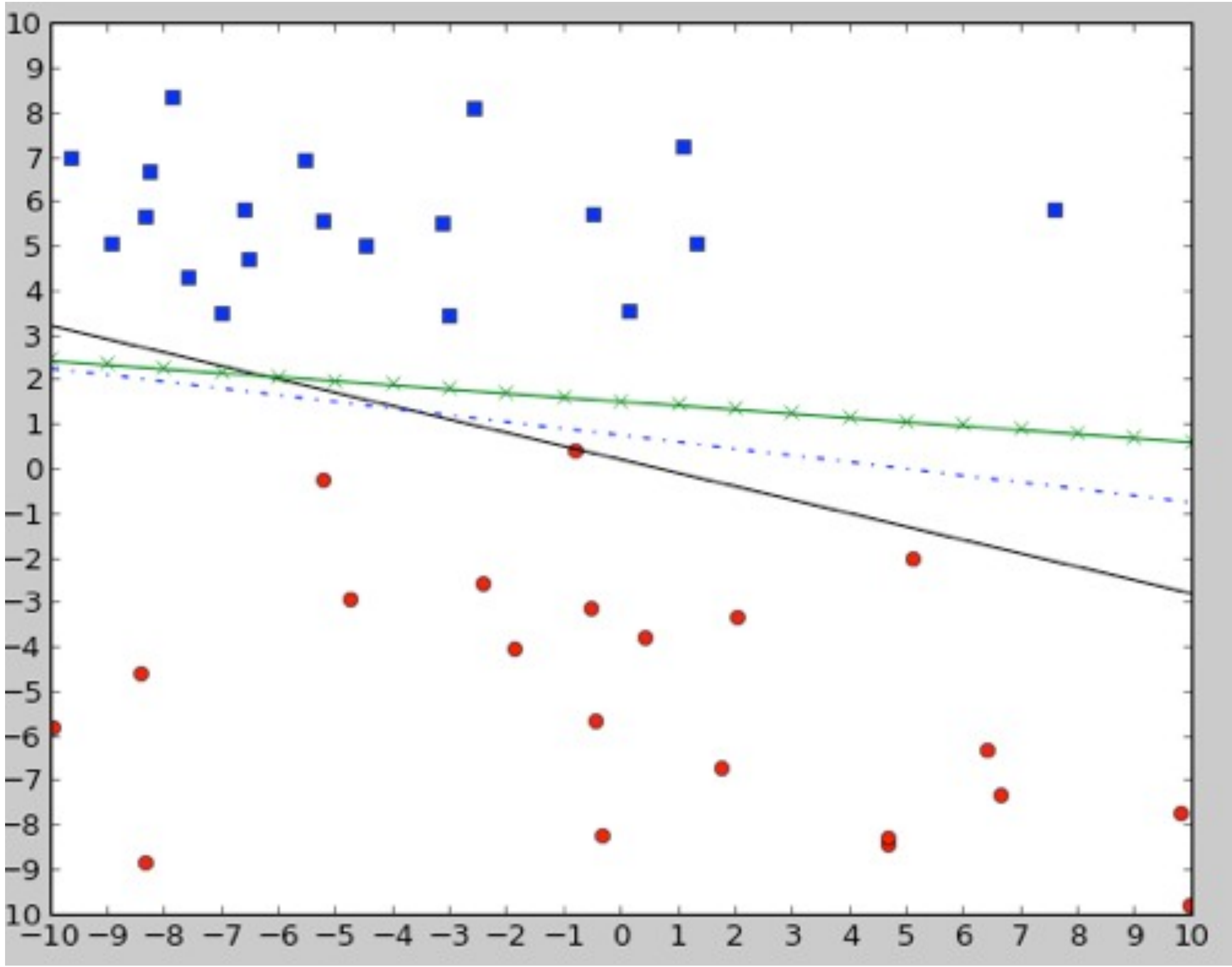


answer: margin shrinks in augmented space!

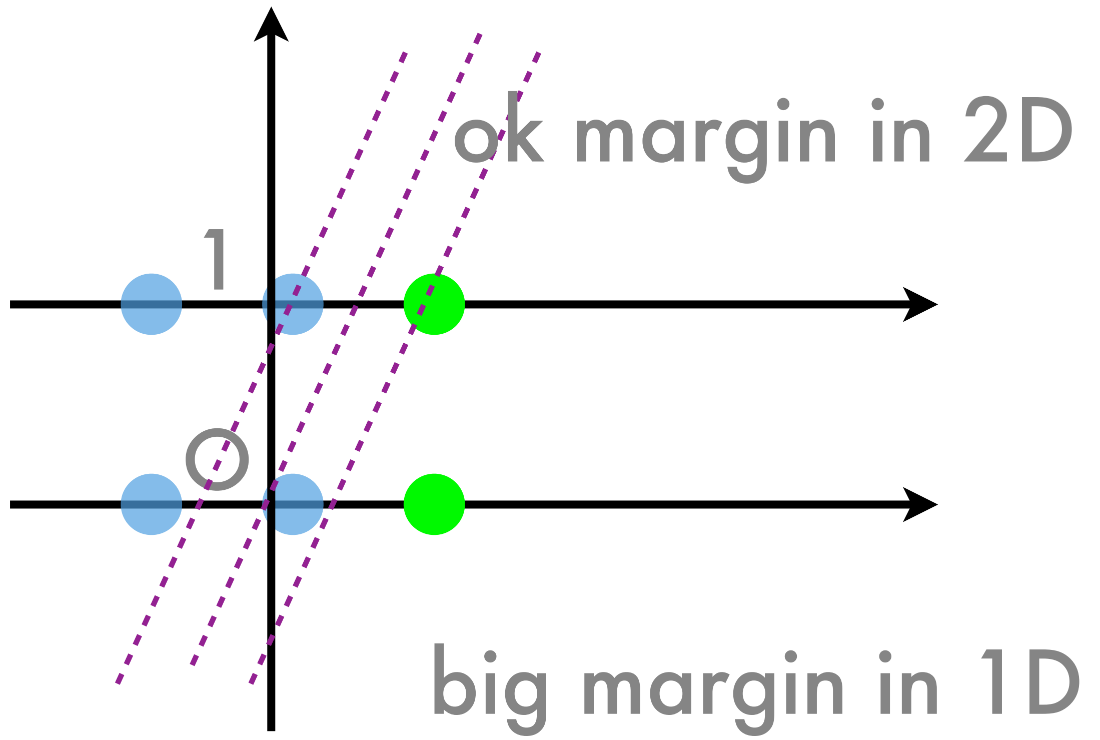


Demo

- perceptron vs. 0.2-aggressive vs. 0.9-aggressive
- why does this dataset so fast to converge?
  - perceptron: 3, p=0.2: 1, p=0.9: 5 epochs



answer: margin shrinks in augmented space!

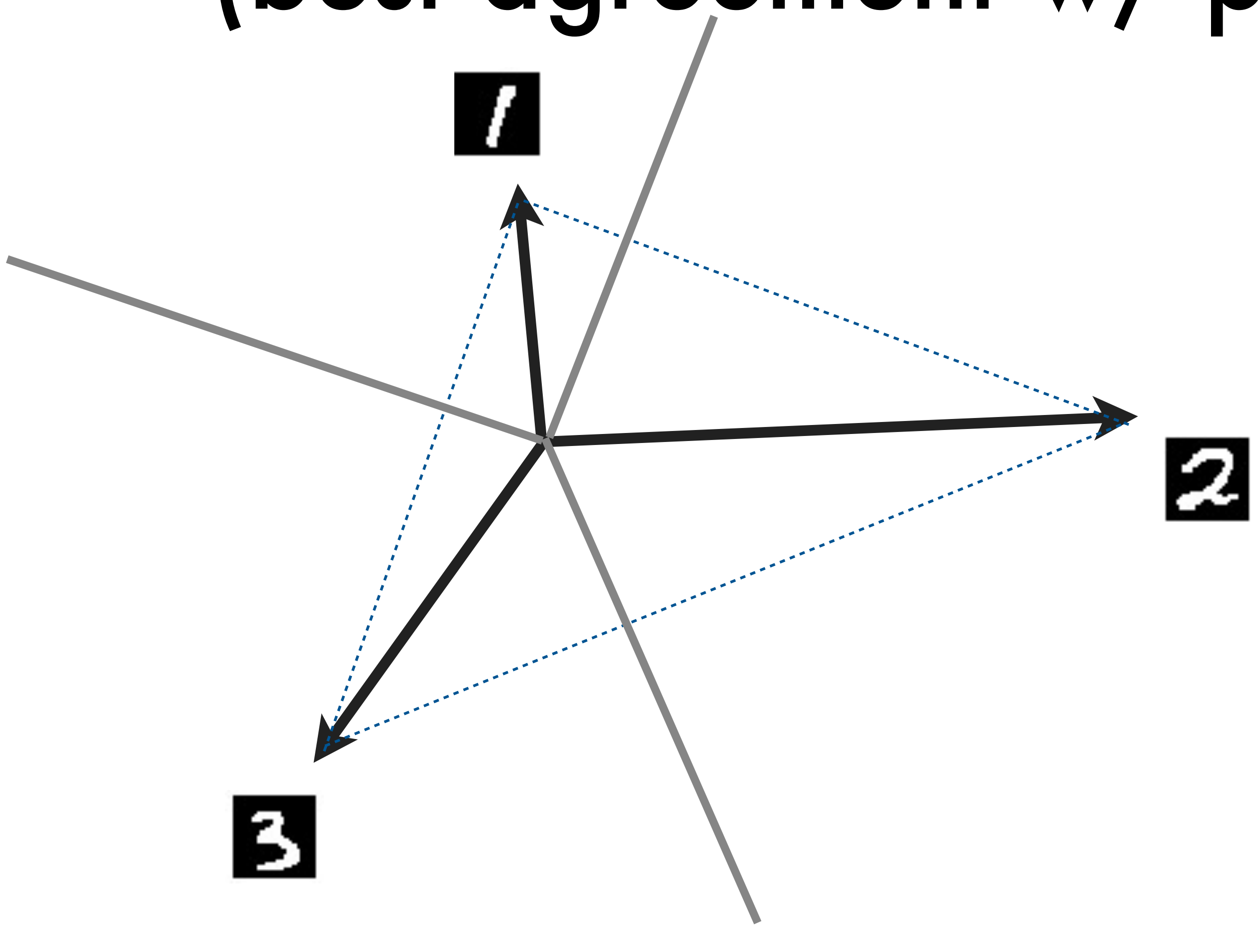


Multiclass Classification

- one weight vector ("prototype") for each class:

$$\mathbf{w} = (\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(M)}),$$

- multiclass decision rule: $\hat{y} = \underset{z \in 1 \ldots M}{\mathbf{argmax}}\, w^{(z)} \cdot x$
  (best agreement w/ prototype)

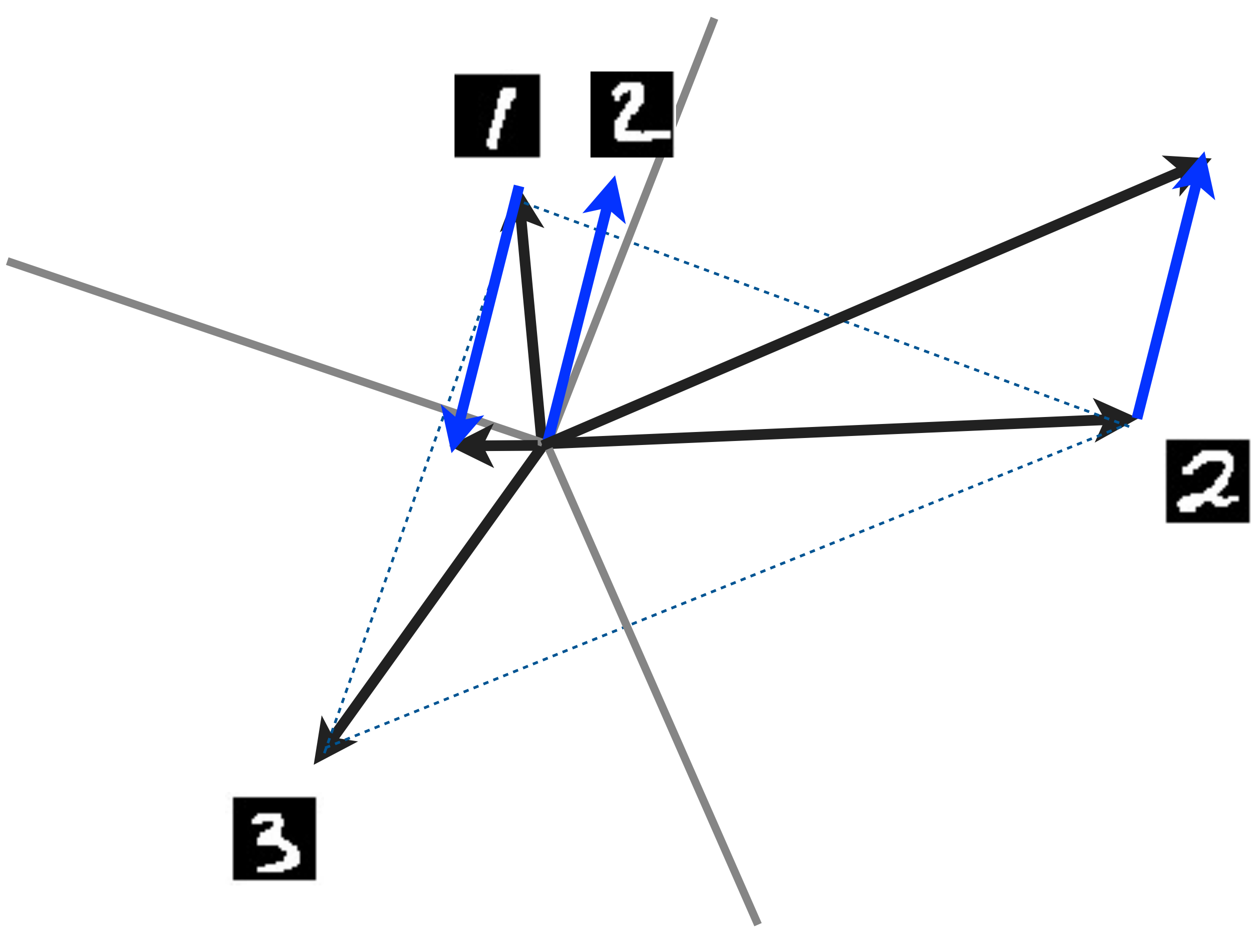


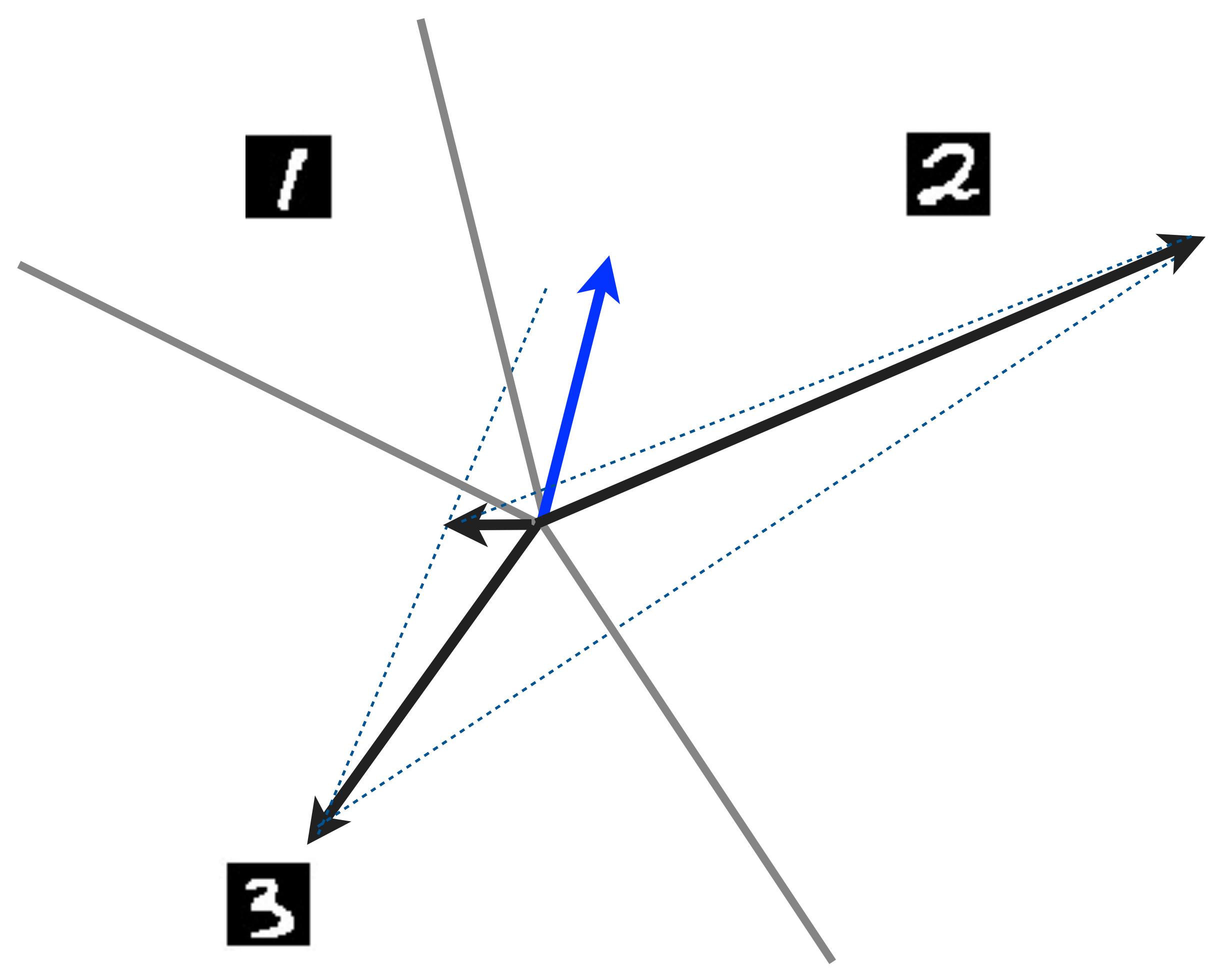
Q1: what about 2-class?

Q2: do we still need augmented space?

Multiclass Perceptron

- on an error, penalize the weight for the wrong class, and reward the weight for the true class





Convergence of Multiclass

$$\mathbf{w} = (\mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots, \mathbf{w}^{(M)}),$$

where $\mathbf{w}^{(i)}$ is used to calculate the functional margin for training example with label i;

for a given training example $\mathbf{x}$ and a label y, we define feature map function $\mathbf{\Phi}$ as

$$\mathbf{\Phi}(\mathbf{x}, y) = (\mathbf{0}^{(1)}, \ldots, \mathbf{0}^{(y-1)}, \mathbf{x}, \mathbf{0}^{(y+1)}, \ldots, \mathbf{0}^{(M)}).$$

such that $\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}, y) = \mathbf{w}^{(y)} \cdot \mathbf{x}$.

We also define that, with a given training example $\mathbf{x}$, the difference between two feature vectors for labels y and z as $\Delta\mathbf{\Phi}$:

$$\Delta\mathbf{\Phi}(\mathbf{x}, y, z) = \mathbf{\Phi}(\mathbf{x}, y) - \mathbf{\Phi}(\mathbf{x}, z).$$

update rule:

$$\mathbf{w} \leftarrow \mathbf{w} + \Delta\mathbf{\Phi}(\mathbf{x}, y, z)$$

separability:

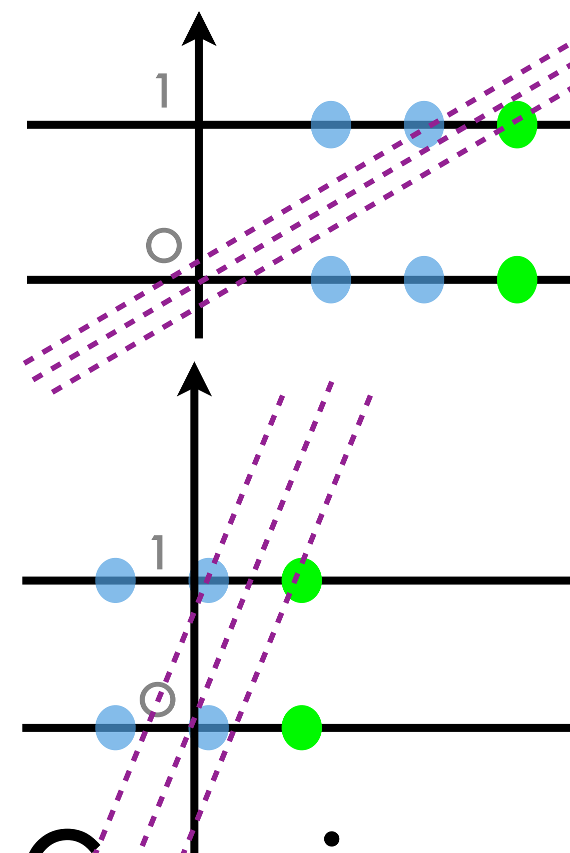
$$\exists \mathbf{u}, \text{ s.t. } \forall (\mathbf{x}, y) \in D, z \neq y$$

$$\mathbf{u} \cdot \Delta\mathbf{\Phi}(\mathbf{x}, y, z) \geq \delta$$

Useful Engineering Tips:

shuffling, variable learning rate, fixing feature scale

- shuffling at each epoch helps a lot
- variable learning rate often helps (constant: useless)
    - 1/(total#updates) or 1/(total#examples) helps
    - any requirement in order to converge?
        - how to prove convergence now?
- centering of each feature dim helps
    - why? => R smaller, margin bigger
- unit variance also helps (why?)
    - 0-mean, 1-var => each feature ≈ a unit Gaussian



Useful Engineering Tips:

feature bucketing (binning/quantization), categorical=>binary

- HW1 Adult income dataset: <=50K, or >50K?
  - age: older means more $$$?
    - bin: `Young` (0-25), `Middle-aged` (26-45), `Senior` (46-65) and `Old` (66+).
  - educational level: 1 to 9 (i think higher is better)
  - hours-per-week: more hours means more $$$?
    - bin: `Part-time` (0-25), `Full-time` (25-40), `Over-time` (40-60) and `Too-much` (60+).
  - native-country: split into X binaries for X countries
  - gender: binary; no need to split into two binaries!
  - type-of-work or position: split into many binaries
  - ...

Brief History of Perceptron