CSC 561: Neural Networks and Deep Learning

Supervised learning, Perceptron

Marco Alvarez

Department of Computer Science and Statistics
University of Rhode Island

Spring 2024

THINK BIG  WE DO℠

Major paradigms

- Unsupervised learning
  - training data (no labels)
  - clustering, dimensionality reduction, density estimation
- Supervised learning
  - labeled training data
  - classification, regression
- Reinforcement learning
  - agent interacting with an environment and learning through trial and error (to maximize rewards)
  - rewards from good actions and penalties for bad actions
- Semi-supervised learning
  - small amount of labeled data to guide the learning process on a large amount of unlabeled data

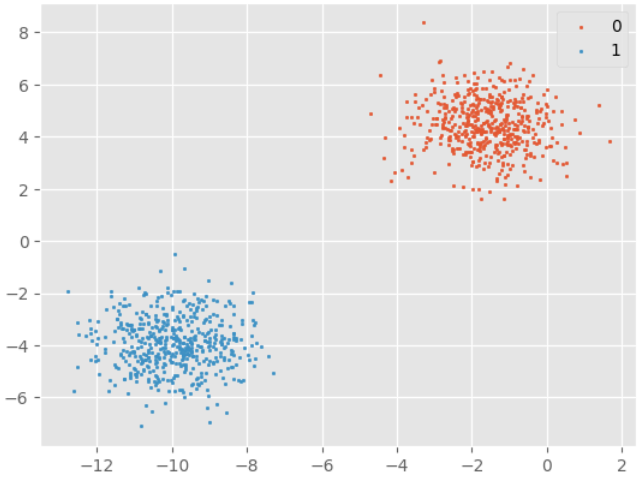
Supervised learning

Credit: Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville

Binary classification

```
[[ -3.05837272   4.48825769]              [[0]
 [ -8.60973869  -3.72714879]               [1]
 [  1.37129721   5.23107449]               [0]
 [ -9.33917563  -2.9544469 ]               [1]
 [-11.57178593  -3.85275513]               [1]
 [-11.42257341  -4.85679127]               [1]
 [-10.44518578  -3.76476563]               [1]
 [-10.44603561  -3.26065964]               [1]
 [ -0.61947075   3.48804983]               [0]
 [-10.91115591  -4.5772537 ]               [1]
 [-11.6173582   -1.94820802]               [1]
 [-11.28684254  -3.8435656 ]               [1]
 [ -2.33022219   4.78405366]               [0]
 [ -1.0749133    4.73076411]               [0]
 [ -0.23594548   5.31048904]]             [0]]
```

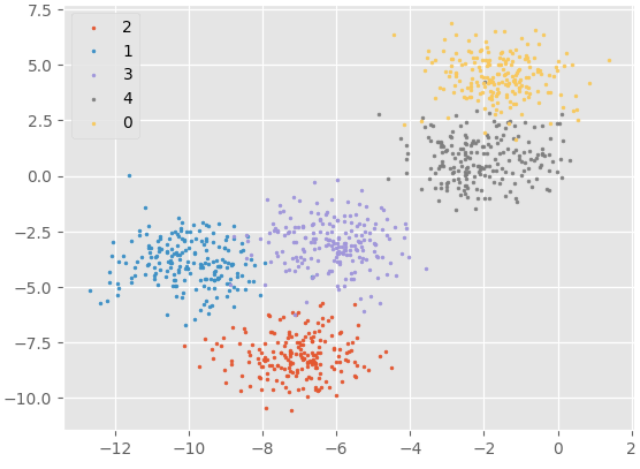
Binary classification

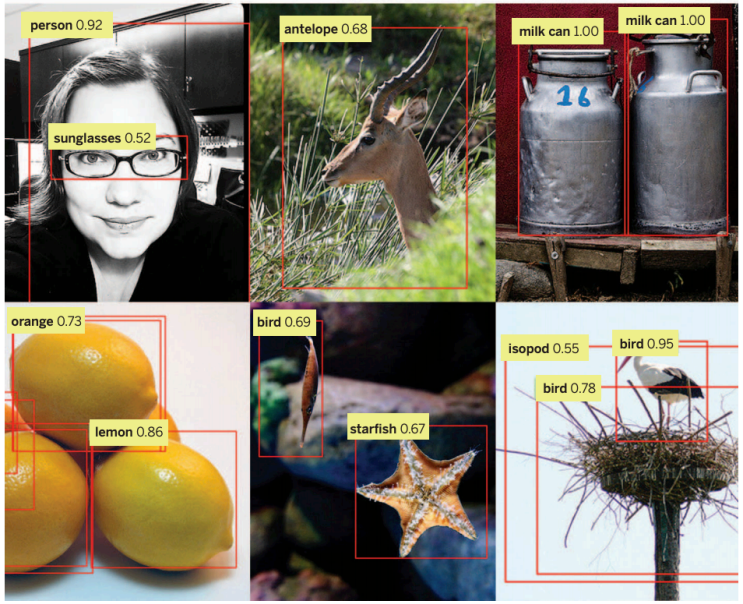


Multiclass classification

```
[[ -7.23632337  -7.96629106]                [[2]
 [-12.22482514  -5.65268215]                 [1]
 [ -6.00096868  -3.25886902]                 [3]
 [ -5.39859591  -2.96553   ]                 [3]
 [ -1.87897516   1.41336   ]                 [4]
 [ -1.76459926   0.21286177]                [4]
 [ -7.07408373  -7.48661742]                 [2]
 [ -0.97134388   0.0301104 ]                 [4]
 [ -1.1583891    2.24295942]                 [4]
 [ -2.22488863   1.4986148 ]                 [4]
 [ -2.48972149   2.43473402]                 [4]
 [ -0.46064203   4.59164629]                 [0]
 [ -1.09679881   4.64722696]                 [0]
 [ -7.29859509  -2.52102072]                 [3]
 [-12.07451453  -3.00584738]]                [1]]
```


Multiclass classification





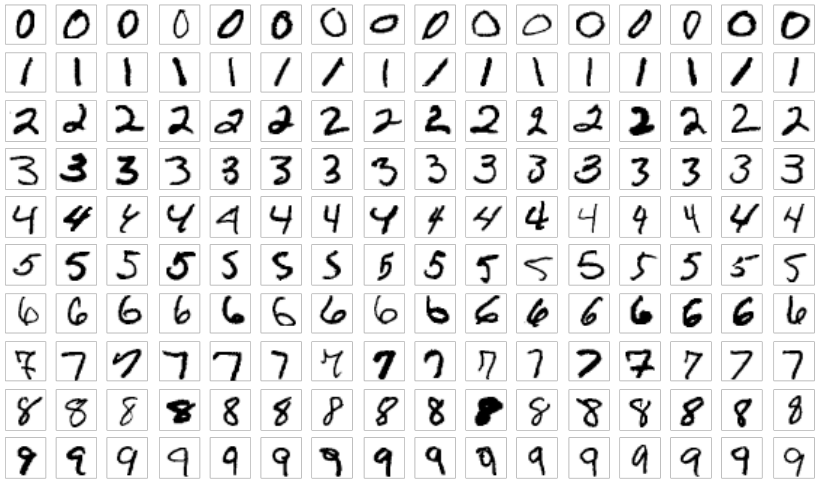
"Machine learning: Trends, perspectives, and prospects", M. I. Jordan and T. M. Mitchell

Components of supervised learning

- Data
  - input space:
    - $\mathcal{X}$ (set) — in general $\mathcal{X} = \mathbb{R}^d$
  - output space:
    - $\mathcal{Y}$ (set)
  - data instance:
    - $(\mathbf{x}, y), \mathbf{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$
  - dataset
    - $\mathcal{D} = \{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)\} \subseteq \mathcal{X} \times \mathcal{Y}$
- Hypothesis
  - function $h : \mathcal{X} \mapsto \mathcal{Y}, h \in \mathcal{H}$

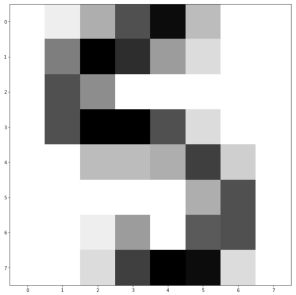
Data instances are independently drawn from an unknown joint distribution $P(X, Y)$

Example: MNIST dataset



https://en.wikipedia.org/wiki/MNIST_database

Example: MNIST dataset



Image

```
[[ 0.  1.  5. 11. 15.  4.  0.  0.]
 [ 0.  8. 16. 13.  6.  2.  0.  0.]
 [ 0. 11.  7.  0.  0.  0.  0.  0.]
 [ 0. 11. 16. 16. 11.  2.  0.  0.]
 [ 0.  0.  4.  4.  5. 12.  3.  0.]
 [ 0.  0.  0.  0.  0.  5. 11.  0.]
 [ 0.  0.  1.  6.  0. 10. 11.  0.]
 [ 0.  0.  2. 12. 16. 15.  2.  0.]]
```

Matrix representation

```
[ 0.  1.  5. 11. 15.  4.  0.  0.  0.  8. 16. … 11.  0.  0.  0.  2. 12. 16. 15.  2.  0.]
```

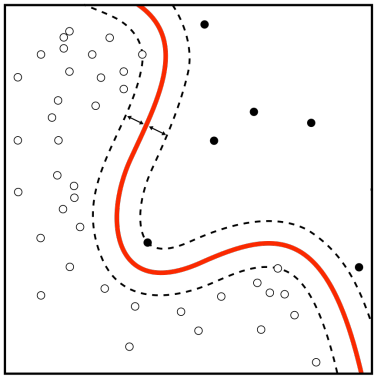
Vector representation

$\mathcal{X} = \mathbb{R}^{64}$ $\mathcal{Y} = \{0,\ldots,9\}$

Hypothesis spaces

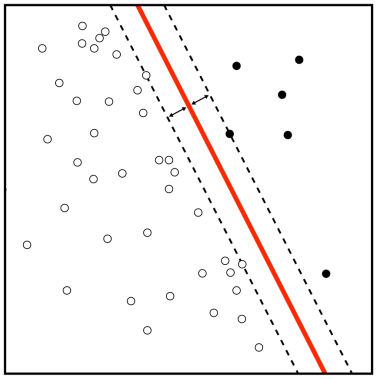
- Hypotheses are functions that belong to a hypothesis space
  - the space is defined by the machine learning technique, i.e., decision trees, neural networks, support vector machines, etc.
- Machine learning approach
  - define the hypothesis space $\mathcal{H}$, restricting the space of all possible functions
  - then find the **best function** within this space
    - a **loss function** is used to evaluate and select hypotheses

Examples



$$h(\mathbf{x}) = \text{sgn}\left(\mathbf{w}^T \text{relu}\left(W\mathbf{x}\right)\right)$$

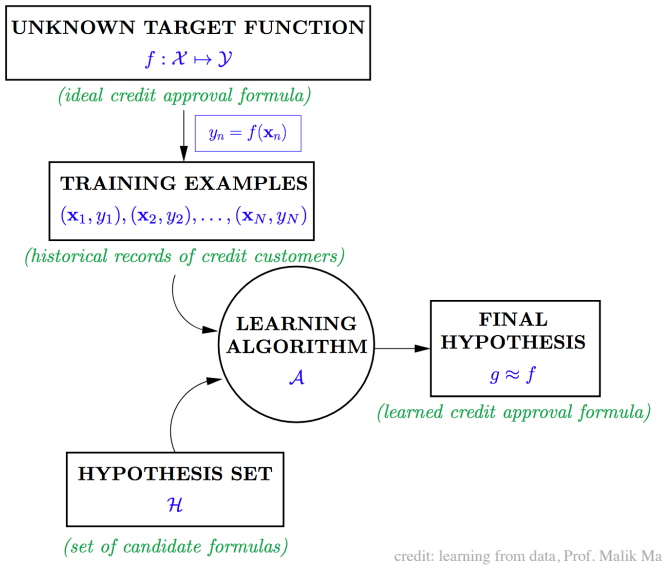
Hypothesis space: all possible two-layer neural nets



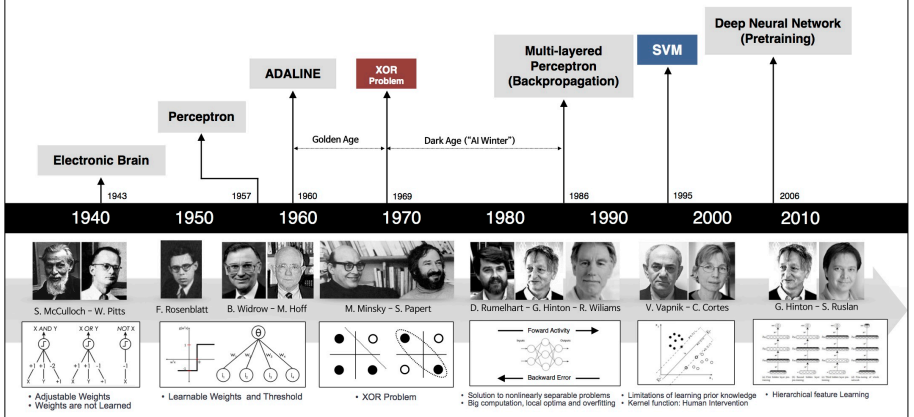
$$h(\mathbf{x}) = \text{sgn}(\mathbf{w}^T\mathbf{x})$$

Hypothesis space: all possible linear functions

Summary



The Perceptron

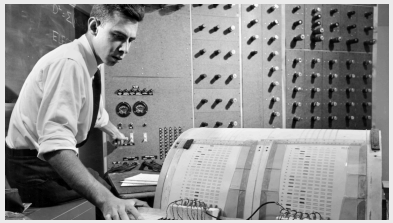


Rosenblatt (1958)

- Perceptron introduced by Frank Rosenblatt (psychologist, logician)
  - based on work from McCulloch-Pitts and Hebb
  - very powerful **learning** algorithm with high expectations

NEW NAVY DEVICE LEARNS BY DOING; Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7, 1958 (UPI) -- The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.



https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon

Journal Information
Journal TOC

Search APA PsycNET

PsycARTICLES: Journal Article

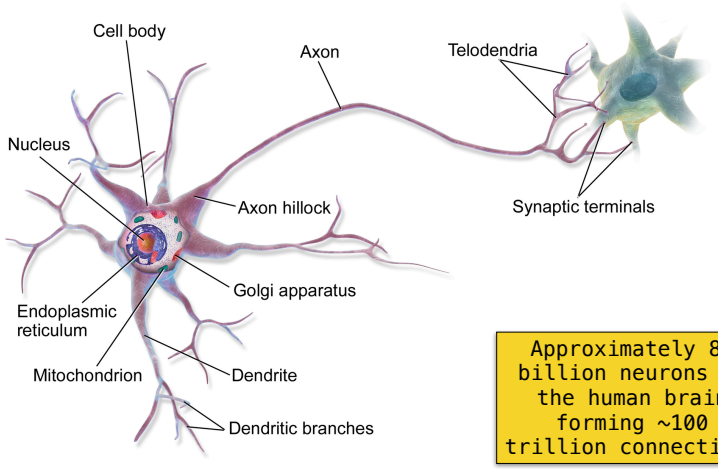
The perceptron: A probabilistic model for information storage and organization in the brain.

Rosenblatt, F.

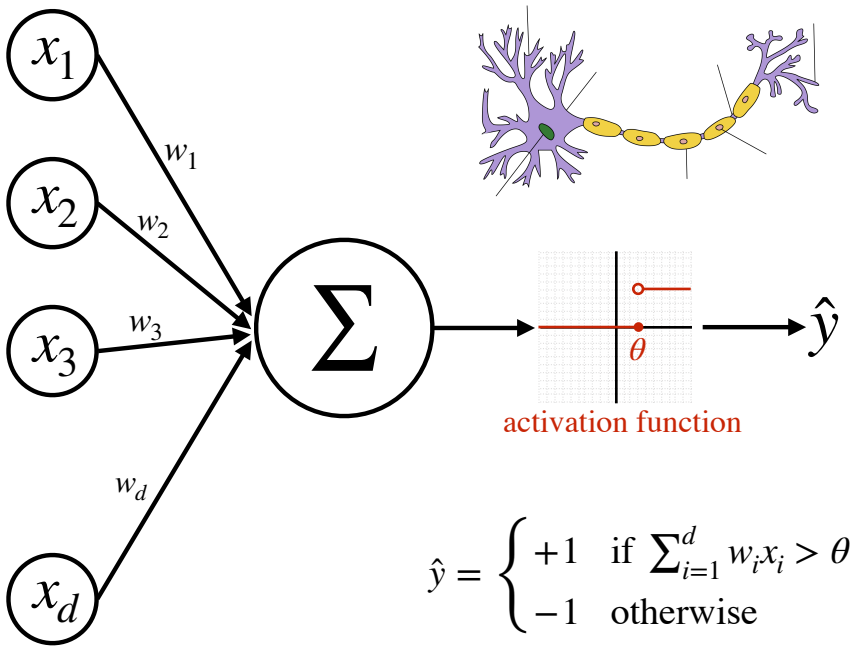
Psychological Review, Vol 65(6), Nov 1958, 386-408

To answer the questions of how information about the physical world is sensed, in what form is information remembered, and how does information retained in memory influence recognition and behavior, a theory is developed for a hypothetical nervous system called a perceptron. The theory serves as a bridge between biophysics and psychology. It is possible to predict learning curves from neurological variables and vice versa. The quantitative statistical approach is fruitful in the understanding of the organization of cognitive systems. 18 references. (PsycINFO Database Record (c) 2016 APA, all rights reserved)

Neuron

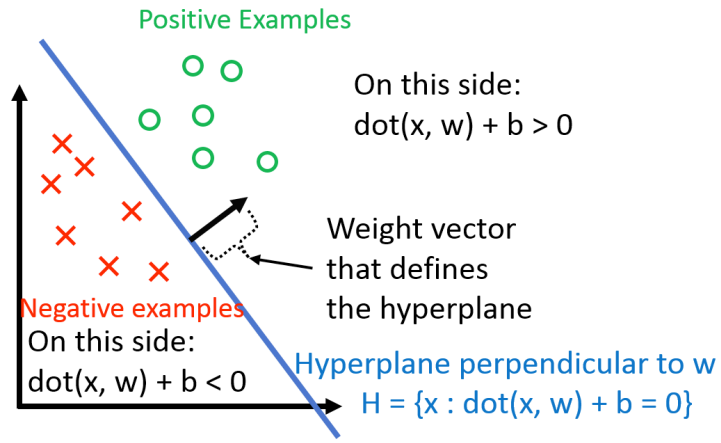


Approximately 86 billion neurons in the human brain forming ~100 trillion connections



$$\hat{y} = \begin{cases} +1 & \text{if } \sum_{i=1}^{d} w_i x_i > \theta \\ -1 & \text{otherwise} \end{cases}$$

Interpretation



Absorbing the threshold/bias

$$\hat{y} = \begin{cases} +1 & \text{if } \sum_{i=1}^{d} w_i x_i > \theta \\ -1 & \text{otherwise} \end{cases}$$

$$x_0 = +1, \quad w_0 = -\theta$$

For convenience we can 'absorb' the threshold into the weight vector by adding an extra dimension

$$\hat{y} = \begin{cases} +1 & \text{if } \sum_{i=0}^{d} w_i x_i > 0 \\ -1 & \text{otherwise} \end{cases}$$

Summary

$$h_{\mathbf{w}}(\mathbf{x}) = \sigma\left(\sum_{i=0}^{d} w_i x_i\right) = \sigma(\mathbf{w}^T\mathbf{x})$$

$$\sigma(z) = \begin{cases} +1 & \text{if } z > 0 \\ -1 & \text{if } z \leq 0 \end{cases}$$

$$\mathcal{H} = \{h_w : \mathbf{w} \in \mathbb{R}^{d+1}\}$$

set of all functions $h_{\mathbf{w}} : \mathbb{R}^{d+1} \mapsto \{-1, +1\}$ defined by $\mathbf{w}$

The algorithm

- Start with a null vector **w**
- Repeat for **T** epochs
  - shuffle the data instances
  - for all examples **(x_i, y_i)** in training data
    - if **x_i** misclassified
      - if **y_i** equals +1
        - update **w** by adding **x_i** to **w**
      - else
        - update **w** by subtracting **x_i** from **w**
- Return **w**

Practice

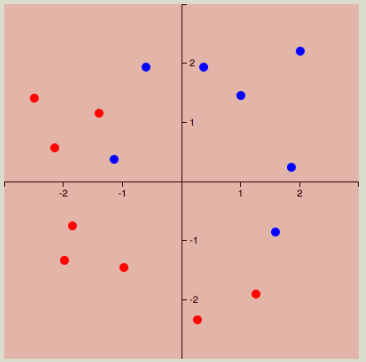
- Write the pseudocode or python for the perceptron algorithm?

Demo

Interactive Perceptron Training Toy

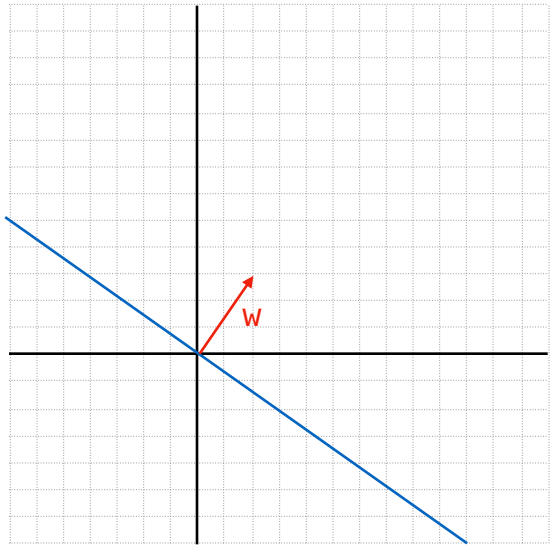
Monday September 7, 2015

A little while ago I contributed a simple perceptron model for the Simple Statistics JavaScript library. This makes possible things like this interactive perceptron model training environment, in which you can get a "hands-on" feel for how the model works in two dimensions.

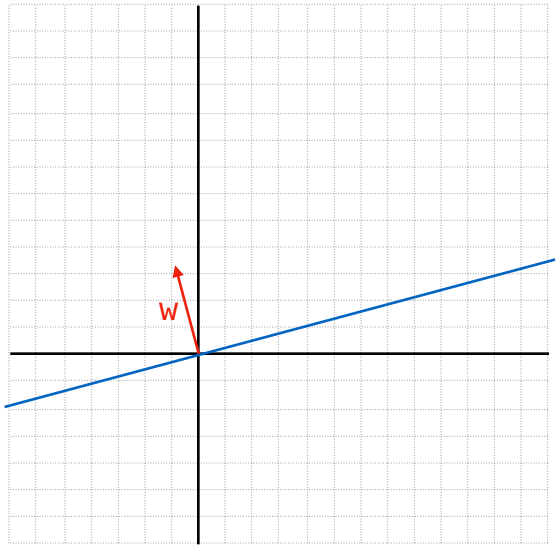


https://planspace.org/20150907-interactive_perceptron_training_toy/

Mistake on a positive (update)



Mistake on a negative (update)



Intuition

- Suppose a mistake on the positive side:
  - $y = +1 \qquad \mathbf{w}^T x \leq 0$
- After 1 update the new weight vector will be:
  - $\mathbf{w}_{t+1} = \mathbf{w}_t + \mathbf{x}$
- Classifying the same datapoint with the new weight vector:
  - $\mathbf{w}_{t+1}{}^T \mathbf{x} = (\mathbf{w}_t + \mathbf{x})^T \mathbf{x} = \mathbf{w}_t^T \mathbf{x} + \mathbf{x}^T \mathbf{x} \geq \mathbf{w}_t^T \mathbf{x}$

`can use the same idea for mistakes on the negative side`

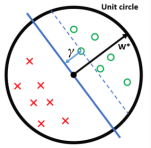
Convergence theorem

The argument goes as follows: Suppose $\exists \mathbf{w}^*$ such that $y_i(\mathbf{x}^\top \mathbf{w}^*) > 0$ $\forall (\mathbf{x}_i, y_i) \in D$. Now, suppose that we rescale each data point and the $\mathbf{w}^*$ such that

$$||\mathbf{w}^*|| = 1 \quad \text{and} \quad ||\mathbf{x}_i|| \leq 1 \ \forall \mathbf{x}_i \in D$$

Let us define the Margin γ of the hyperplane $\mathbf{w}^*$ as $\gamma = \min_{(\mathbf{x}_i, y_i) \in D} |\mathbf{x}_i^\top \mathbf{w}^*|$.

A little observation (which will come in very handy): For all $\mathbf{x}$ we must have $y(\mathbf{x}^\top \mathbf{w}^*) = |\mathbf{x}^\top \mathbf{w}^*| \geq \gamma$. Why? Because $\mathbf{w}^*$ is a perfect classifier, so all training data points $(\mathbf{x}, y)$ lie on the "correct" side of the hyper-plane and therefore $y = sign(\mathbf{x}^\top \mathbf{w}^*)$. The second inequality follows directly from the definition of the margin γ.



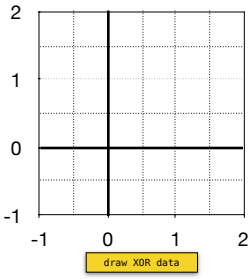
- All inputs $\mathbf{x}_i$ live within the unit sphere
- There exists a separating hyperplane defined by $\mathbf{w}^*$, with $||\mathbf{w}||^* = 1$ (i.e. $\mathbf{w}^*$ lies exactly on the unit sphere).
- γ is the distance from this hyperplane (blue) to the closest data point.

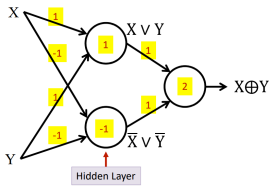
Theorem: If all of the above holds, then the Perceptron algorithm makes at most $1/\gamma^2$ mistakes. **Proof:**

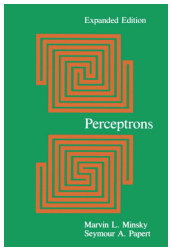
https://www.cs.cornell.edu/courses/cs4780/2024sp/lectures/lecturenote03.html

Minsky & Papert (1969)

- Perceptrons
  - AI winter — significant impact, shifting focus on symbolic approaches
  - core message: perceptrons can only learn linear decision boundaries







Remarks

- Assumes data is **linearly separable**
  - does not converge if classes are not linearly separable
- Different **correct** solutions can be found
  - most are not optimal in terms of **generalization**
- Averaged Perceptron
  - returns a **weighted average** of earlier hypotheses