Math 3934 Lec 5 2/14/18

Perceptron

why $p+1$?

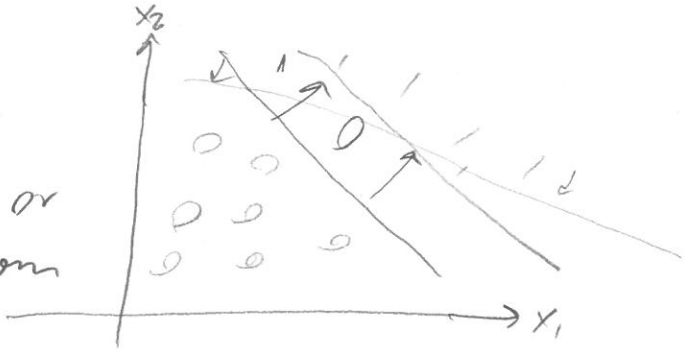
$\vec{x}_i = [1 \; x_{i1} \; x_{i2} \; \dots \; x_{ip}]$

$$\mathcal{H} = \left\{ \mathbb{1}_{\vec{w} \cdot \vec{x}_i > 0} \; : \vec{w} \in \mathbb{R}^{p+1} \right\}$$

$g = A(\mathcal{D}, \mathcal{H})$ How does $A$ produce $g$?

It's very difficult to search all $\vec{w} \in \mathbb{R}^{p+1}$ to minimize misclassification error, so we use a "heuristic" (rule of thumb / approximate) algorithm called the perceptron learning algorithm

Here's how it works...




(1) Initialize the weights to be $\vec{w}^{t=0} = \vec{0}$ or random

(2) Calculate $\hat{y}_i$

$$\hat{y}_i = \mathbb{1}_{\vec{w}_0 \cdot \vec{x}_i > 0}$$

(3) Update all weights $j = 0, \ldots, p$

$$w_0^{t=1} = w_0^{t=0} + (y_i - \hat{y}_i)\ (1)$$

$$w_1^{t=1} = w_1^{t=0} + (y_i - \hat{y}_i)\, x_{i,1}$$

$$\vdots$$

$$w_p^{t=1} = w_p^{t=0} + (y_i - \hat{y}_i)\, x_{i,p}$$

note: there is only an update on a misclassification o/w $y_i - \hat{y} = 0$!

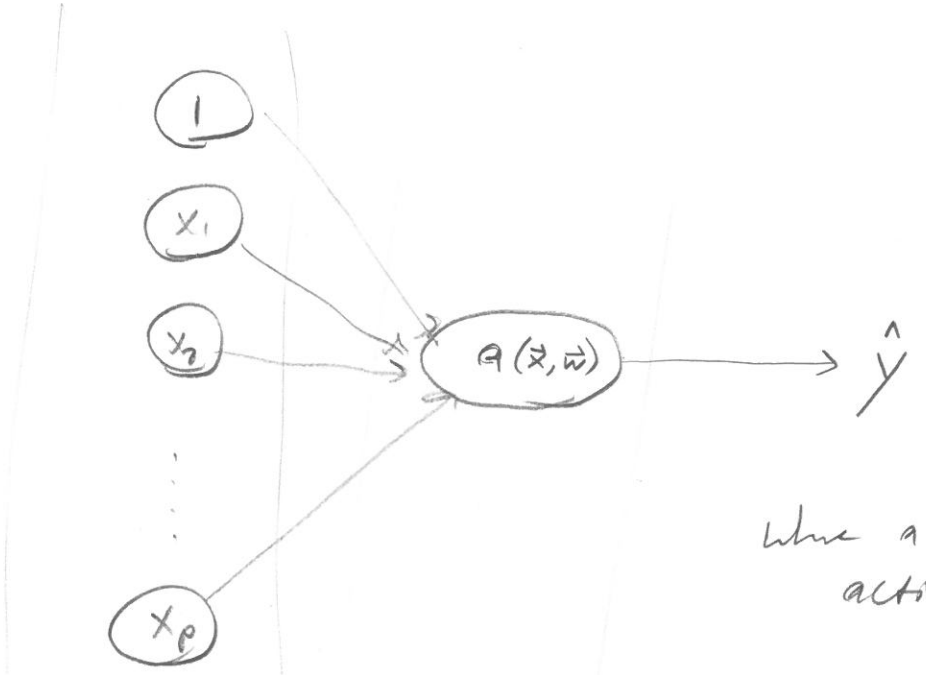
(4) Repeat steps 2 and 3 $\forall\, i \in \{1, \ldots, n\}$.

(5) Repeat steps 2-4 until error hits threshold or until # of iterations is reached.

Note: if "linearly separable", error will be 0.

In practice with real data this is unlikely with just the $p$ features.

How the perceptron is usually illustrated



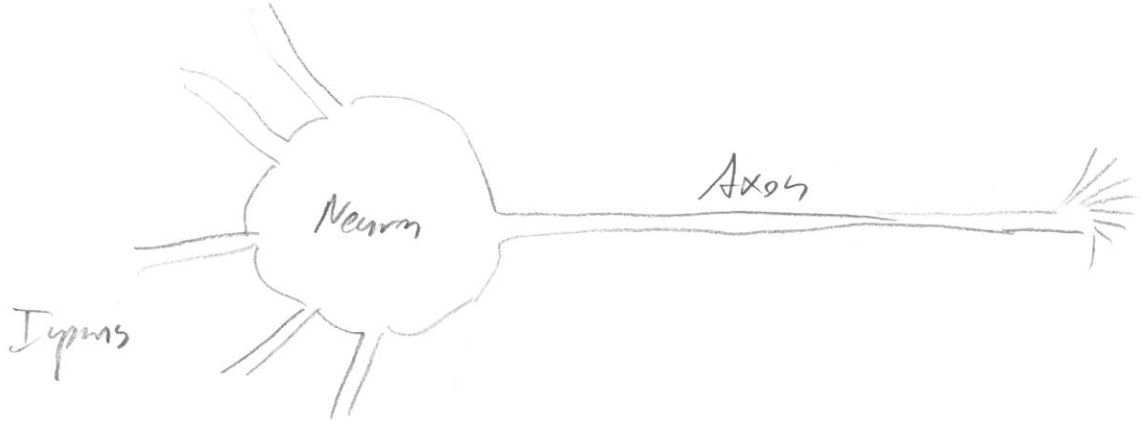

where $a$ is the activation function

"Input Layer" "Output Layer"

In a perceptron $a(\vec{x}, \vec{w}) = \mathbb{1}_{\vec{w} \cdot \vec{x}}$ binary step/ the Heaviside (indicator)

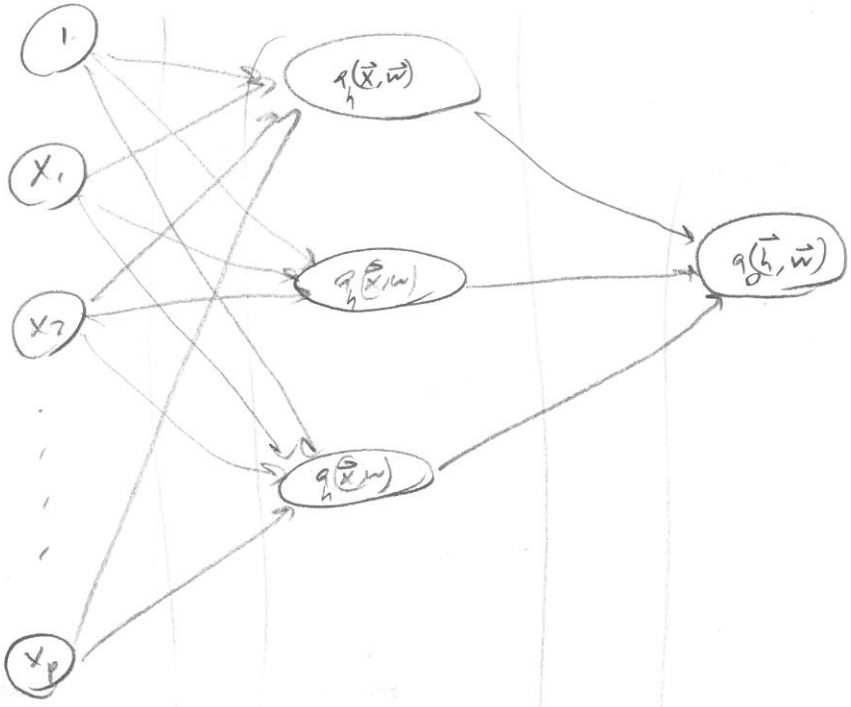
This is also called a "single layer neural network" with activation function. We will return to these later.

Why "neural network?" Loosely inspired by the neuron



Teaser...

You can imagine a setup that looks like:




Input Layer

"Hidden Layer"
we don't observe its output

Output Layer

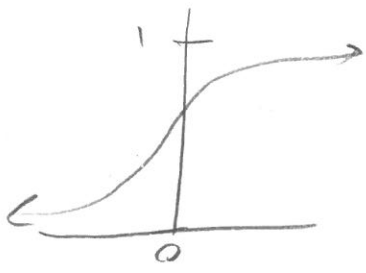
"depth 2" (input layer doesn't count)

higher depth
$\Rightarrow$ "deep"

Since we're learning from data
$\Rightarrow$ "deep learning"

Diving Step

It makes sense for $g_0$ to be the Heaviside/Indicator function because $y = \{0,1\}$. But what about $g_h$. It wouldn't be efficient to have 0's & 1's only... why not...




Sigmoid / Logistic

$$g_h(\vec{x}) = \frac{1}{1+e^{-\vec{w}\cdot\vec{x}}}$$

or many others....

Hopefully we will return to these...

Big project to find
$3\vec{w}$'s and an $\vec{h}$
parameter dim $= 3(p+1)+3$