DS 4440

Beyond Linear Models

Last time: Linear Regression from an "ML" perspective – i.e., $\nabla \mathcal{L}$ / SGD

$$\nabla_w \mathcal{L} = \underbrace{-2x^T y}_{d \times n \;\; n \times 1} + \underbrace{2x^T x}_{d \times d} \underbrace{w}_{d \times 1}$$

Analytic Solution

$$w^* = (x^T x)^{-1} x^T y$$

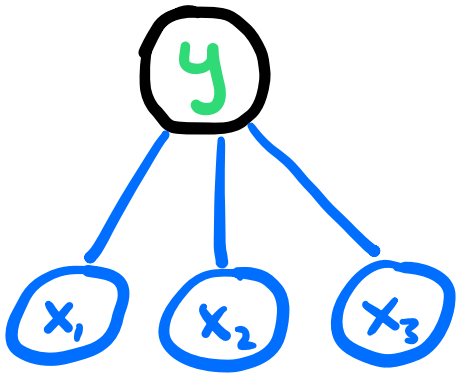
Gradient Descent

$$\hat{w}^{(t+1)} \leftarrow \hat{w}^{(t)} - \underbrace{\alpha}_{\text{learning rate}} \nabla_w \mathcal{L}(\underbrace{x, y}_{\text{usually batched}} | w^{(t)})$$

Let's see in Colab...

(exercise / NB review)

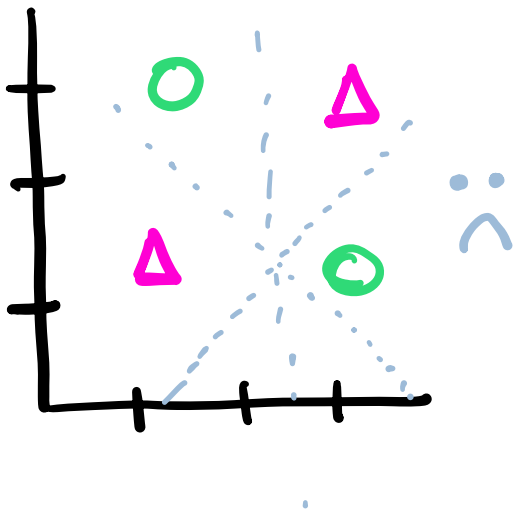
So far we have assumed



$$y = f(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3)$$

This implies a line or hyperplane that separates classes.

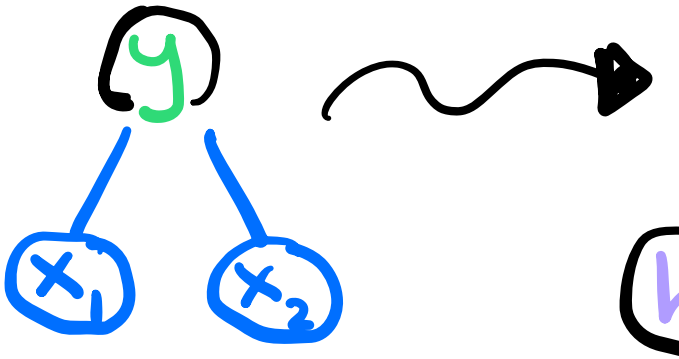
But consider:

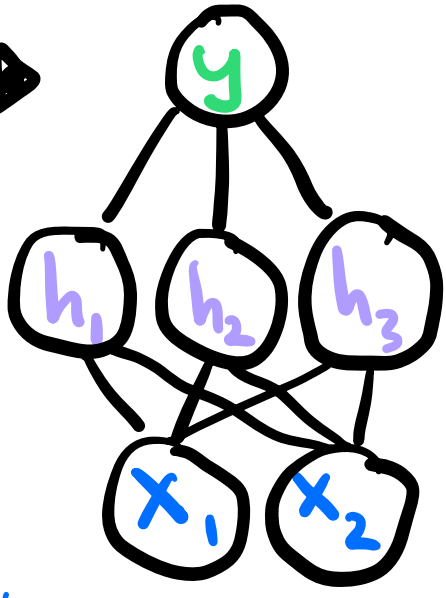


$\Delta = 1$

$\text{O} = \emptyset$

How can we address this?





$W^y \in \mathbb{R}^{1\times 3}$

$W^h \in \mathbb{R}^{3\times 2}$

Layers!

$$\begin{bmatrix} & | & \\ & | & \\ & | & \end{bmatrix} \begin{matrix} h_1 \\ h_2 \\ h_3 \end{matrix} \qquad \begin{bmatrix} & & \\ - & - & - \\ & & \end{bmatrix} \begin{matrix} x_1 \\ x_2 \end{matrix}$$

$$W^h \cdot x = h$$

But this is still linear!

$$y = w_1^y \cdot h_1 + w_2^y \cdot h_2 + w_3^y \cdot h_3$$

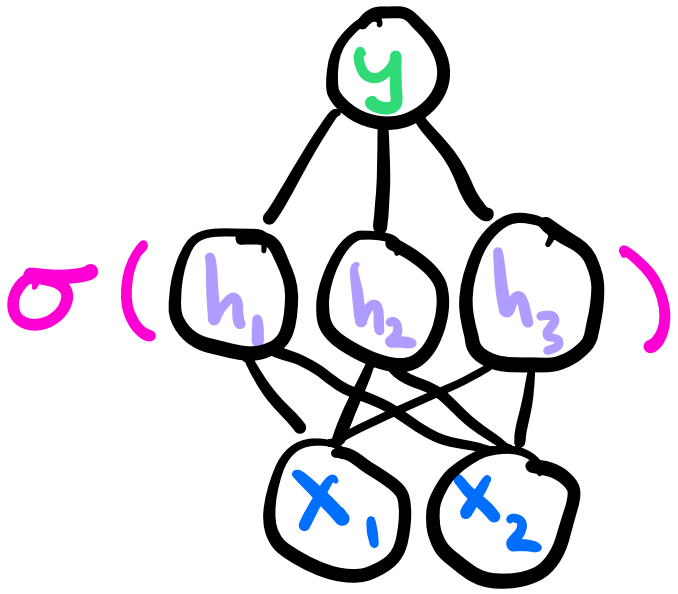
$$\hookrightarrow = w_{1:}^h \cdot x$$

$$= w_1^y (w_{1:}^h \cdot x) + w_2^y (w_{2:}^h \cdot x) + w_3^y (w_{3:}^h \cdot x)$$

So just a weirdly parameterized

linear model! [See Colab NB.]

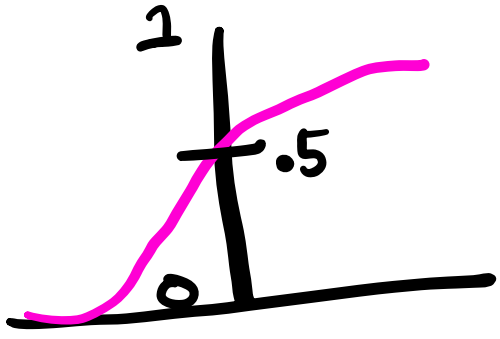
Enter activation functions σ



Applied element wise.

One choice for σ: Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

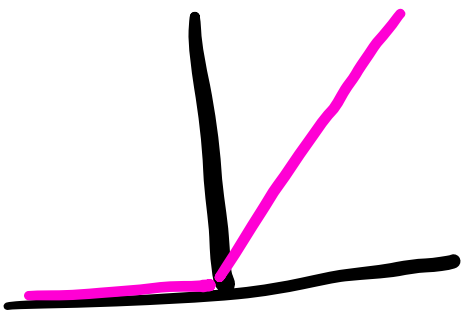


$$\frac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x))$$

Another (popular) option:

Rectified Linear Unit (ReLU)

$$\sigma(x) = \begin{cases} x & \text{if } x > 0 \\ \emptyset & \text{otherwise} \end{cases}$$



$$\frac{d}{dx}\sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ \emptyset & \text{otherwise} \end{cases}$$

(See notebook on activations)

Note SoftMax

$$SM(z)_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

z: vector