Deep Learning Intro

- trees
- KNN
- Kernels

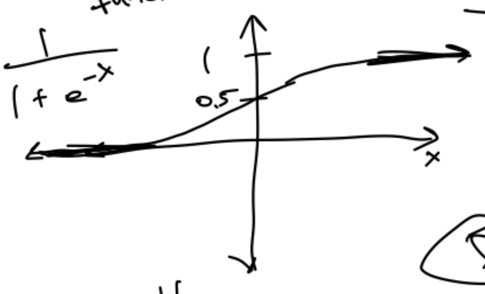
(Rosenblatt, 1958)

$$h(x) = \underset{\text{feat. req.}}{\phi(x)^T} \underset{\text{hyperplane}}{\underline{w}} + \underline{b}$$

$$\phi(x) = \underset{\text{function}}{\sigma}(\overset{\text{matrix}}{A}x + \underset{\text{vector}}{\underline{c}})$$

Sigmoid

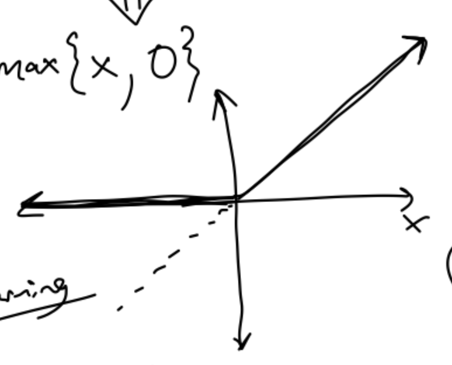
$$\sigma(x) = \frac{1}{1+e^{-x}}$$



ReLU

$$\Downarrow$$

$$\sigma(x) = \max\{x, 0\}$$



More Layers: Deep Learning

$$h(x) = w^T \phi(x) + b$$

layers

$$\phi(x) = \sigma(A\phi'(x) + c)$$

$$\phi'(x) = \sigma(A'\phi''(x) + c')$$

$$\phi''(x) = \sigma(A''x + c'')$$

Why not 1-layer

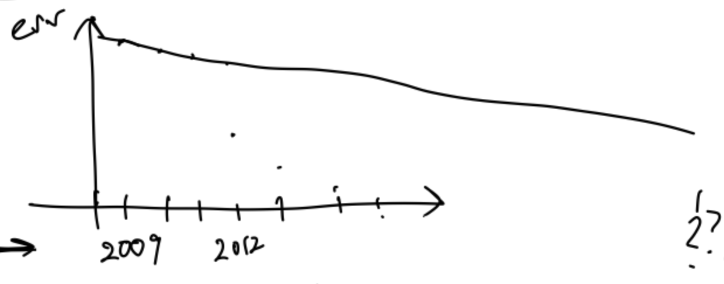
$$A \in \mathbb{R}^{\text{million/billion} \times p}$$

Key Changes to NNs

- signoid ⟶ ReLU
- GPUs
- SGD
- rebranding

90s : SVMs much faster!

ImageNet



¿?!

Speech Recognition

Deep Learning

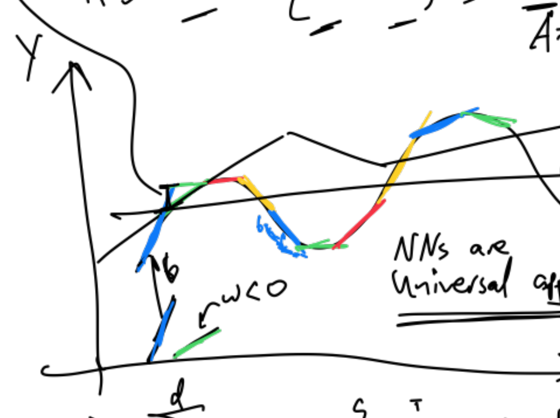
$$h(x) = \underline{w}^T \phi(x) + \underline{b}$$

$$\phi(x) = \sigma(\underline{A}x + \underline{c})$$

$$\mathcal{L}(h) = \sum_{i=1}^{n} \ell(h(x_i), y_i) = \sum_{i=1}^{n} (h(x_i) - y_i)^2$$

$$h(x) = \underline{w}^T \max\{\underline{A}x + \underline{c}, 0\} + \underline{b}$$

$$A = \begin{bmatrix} - & a_1 & - \\ - & a_2 & - \\ & \vdots & \\ - & a_d & - \end{bmatrix} \quad c = \begin{bmatrix} c_1 \\ \vdots \\ c_d \end{bmatrix}$$



NNs are Universal approximators

$$h(x) = \sum_{j=1}^{d} w_j \max\{a_j^T x + c_j, 0\} + b$$