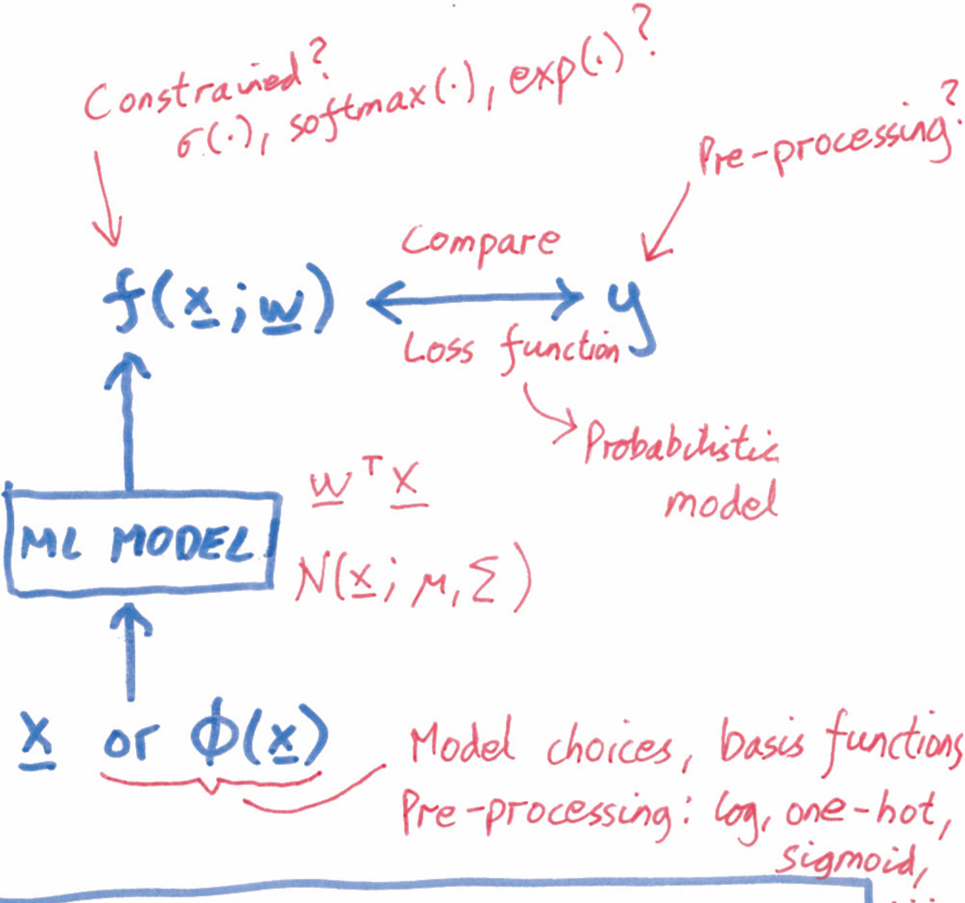


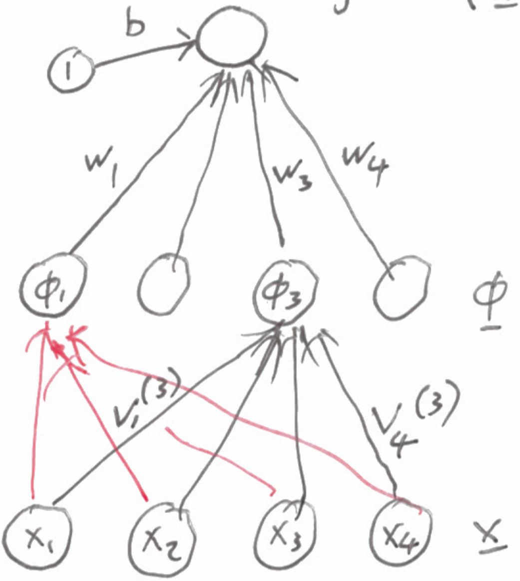
Training, validation, testing

Neural Networks

First example:

$$f = \sigma(\underline{w}^\top \underline{\phi} + b)$$

Logistic regression

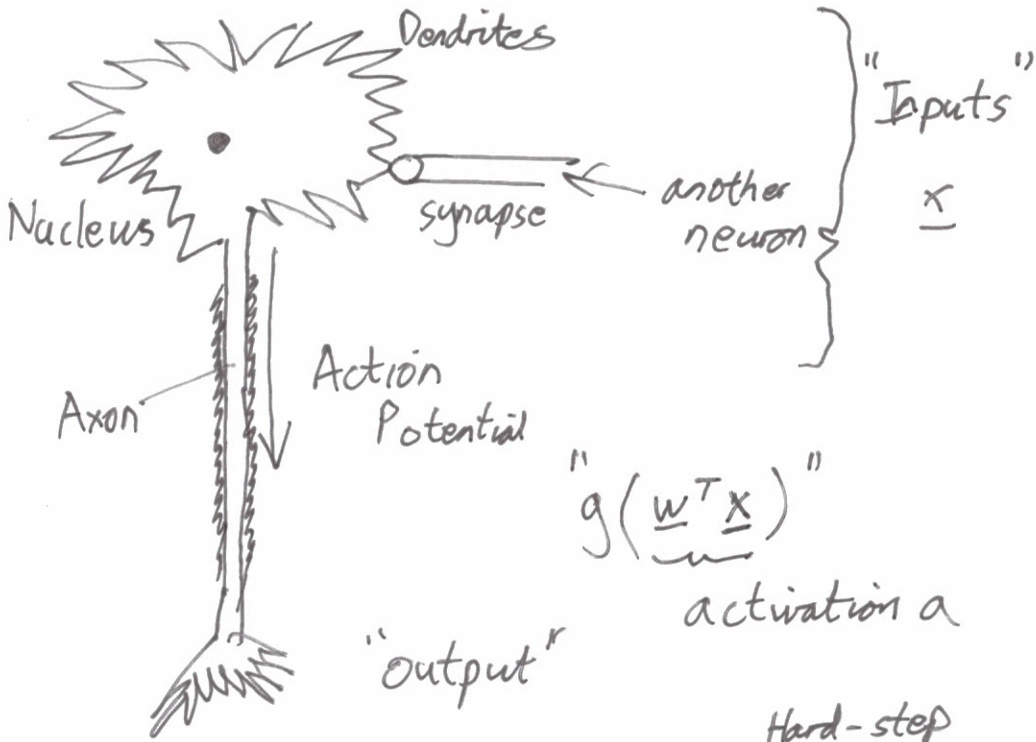


$$\phi_k = \sigma(\underline{v}^{(k)\top} \underline{x} + b^{(k)})$$

Fit $\{\{\underline{v}^{(k)}, b^{(k)}\}, \underline{w}, b\}$ with a gradient-based optimizer. Match f to training y's using some loss.

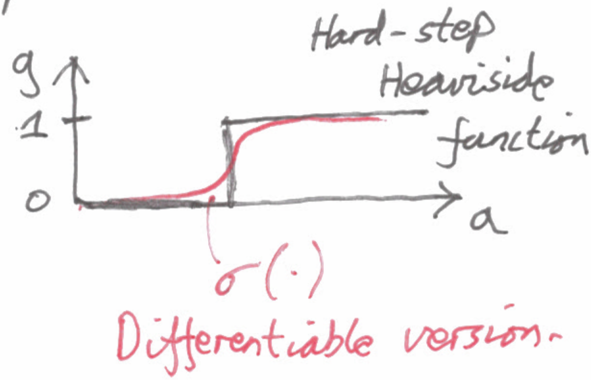
Why "Neural Network"? (non-examinable)

Neuron = Nerve cell

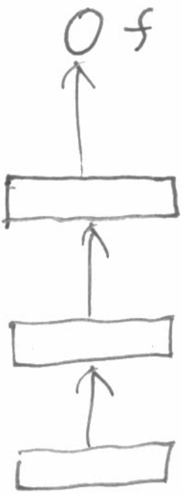


"$g(\underbrace{\underline{w}^T \underline{x}})$"

activation a



Feed-forward Neural Networks



$$\underline{f} = g^{(3)}\left(W^{(3)}\underline{h}^{(2)} + \underline{b}^{(3)}\right)$$

$$\underline{h}^{(2)} = g^{(2)}\left(W^{(2)}\underline{h}^{(1)} + \underline{b}^{(2)}\right)$$

$$\underline{h}^{(1)} = g^{(1)}\left(W^{(1)}\underline{x} + \underline{b}^{(1)}\right)$$

$\underline{x}$

If f is a scalar $W^{(3)}\underline{h}^{(2)} = \underline{w}^{(3)\,T}\underline{h}^{(2)}$

Other architectures possible:

"skip connections"

$$\underline{h}^{(1)} = g^{(1)}\left(W^{(1)}\underline{x} + b^{(1)}\right) + \underline{x}$$

"residual layer"

Parameterize the functions g.

For particular data: ConvNets for images and some audio.

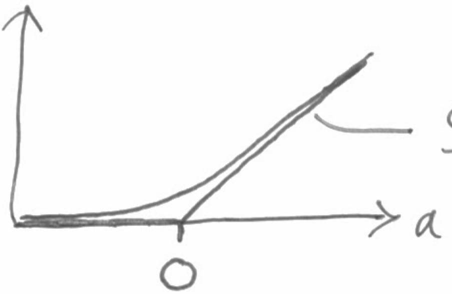
Sequence Models.

Non-linearity

Sigmoidal function σ:

ReLU: Rectified Linear Unit

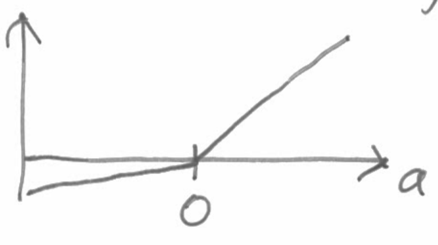
(2010 paper)



Soft plus $= \log(1+e^a)$

$$\text{ReLU}(a) = \max(a, 0)$$

PReLU



$$f(a) = \begin{cases} a & a > 0 \\ sa & a \leq 0 \end{cases}$$

s ← Parameter

Initialize the weights?

Mustn't set $W^{(l)}$ to zero.

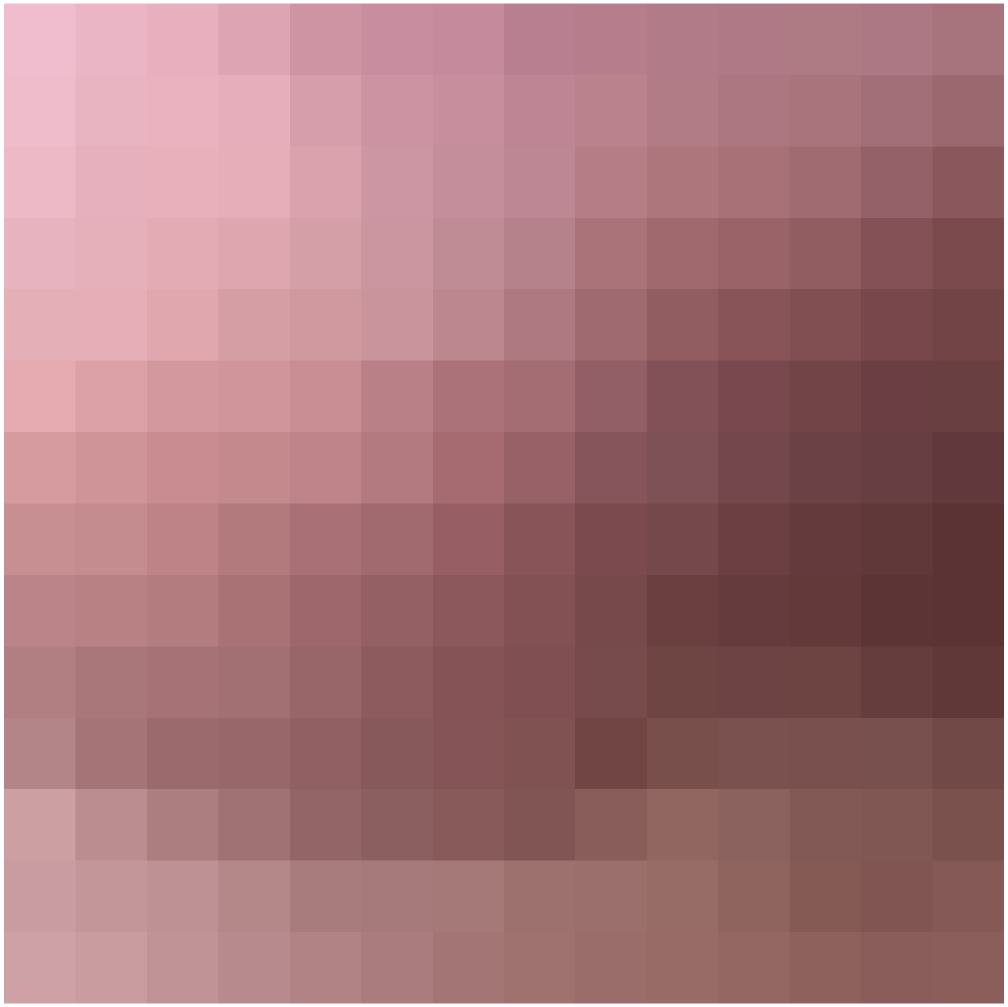
$\Rightarrow$ All hiddens extract same feature

$\Rightarrow$ Weights for each hidden stay the same.

$\Rightarrow$ Randomly set weights.

Vision in the brain

14×14 **patch of retinal pixels:** **(cartoon)**



How many V1 neurons connect to each patch?

Vision in the brain

$\sim 100,000$ V1 neurons connect to 14×14 pixels in macaques

via Bruno Olshausen `http://redwood.berkeley.edu/bruno/papers/CNS2010-chapter.pdf`