First example:

\[ f = \sigma(w^T \phi + b) \]

\[ \phi_k = \sigma(v^{(k)}^T x + b^{(k)}) \]

"unit" "neuron"

Fit $\xi \in \{v^{(k)}, b^{(k)}, w, b\}$ with a gradient-based optimizer. Match $f$ to training set, using some loss.
Why "Neural Network"? (non-examinable)

Neuron = Nerve cell

Dendrites → synapse → another neuron

"Inputs" \( \mathbf{x} \)

Nucleus

Axon

Output: \( g(\mathbf{w}^T \mathbf{x}) \)

Activation: \( g \)

Hard-step or Heaviside function.

Logistic sigmoid

Differentiable version.
Feed-forward Neural Networks

\[ f \circ f = g^{(3)}(W^{(3)}h^{(2)} + b^{(3)}) \]

\[ h^{(2)} = g^{(2)}(W^{(2)}h^{(1)} + b^{(2)}) \]

\[ h^{(1)} = g^{(1)}(W^{(1)}x + b^{(1)}) \]

- When \( f \) is a scalar, \( W^{(3)}h^{(2)} = w^{(2)T}h^{(2)} \)

- Other architectures possible:
  - "skip connections"
  - parameterize the \( g \)'s non-linearities

- Special layers for images/audio:
  - Conv Nets... and others.
Non-linearities

These are what we called basis functions:

Sigmoid $\sigma$: softly partitions space.
RBFs: is input near some point.
ReLU: Rectified Linear Units

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

Soft Plus

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

PReLU

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

Set initial weight matrix $W^{(k)}$

Must not set $W^{(k)}$ to be all zeros.

$\Rightarrow$ All hidden extract same features

$\Rightarrow$ Weights stay the same.

$\Rightarrow$ Randomly set each weight.