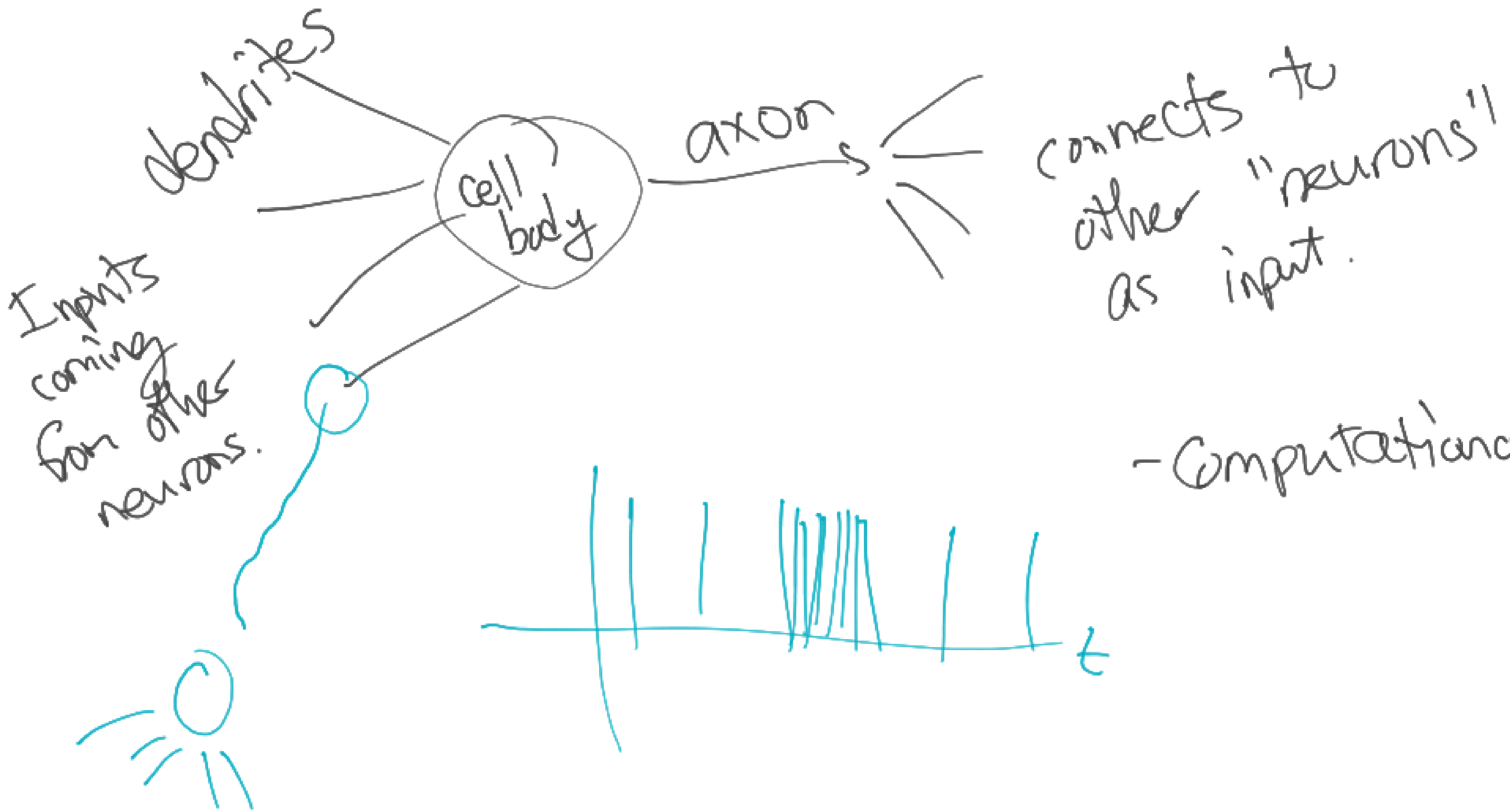
$(x_i, y_i)_{i=1}^{n}$ $\quad f_{w_k}(x_i) = \sum_{j=1}^{m'} w_{k,j}\,\underbrace{\phi_j(x_i)}$ $\quad \phi : \mathbb{R}^m \to \mathbb{R}^{m'}$

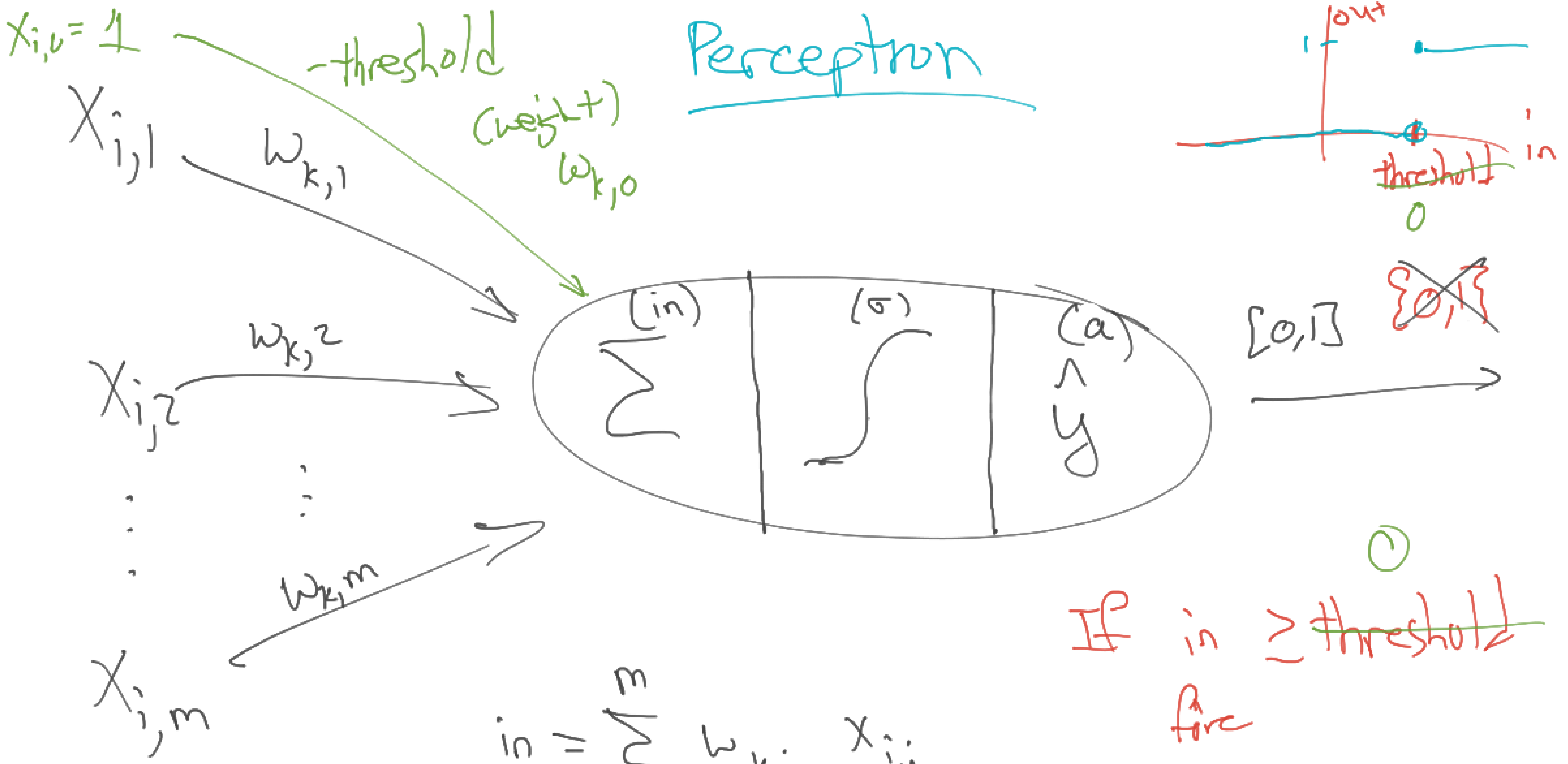
$\ell(w_k) = \sum_{i=1}^{n} \left(y_i - f_{w_k}(x_i)\right)^2$ $\quad \phi_j(x_i)$ or $\phi(x_i)_j$

Lecture 7:
- Perceptron
- ANN notation



- Computational neuroscience.

Perceptron



$$in = \sum_{j=1}^{m} w_{k,j} \; x_{i,j}$$

$$a = \sigma(in)$$

output

If $in \geq$ ~~threshold~~ 0
fire
else
don't fire.

"Activation function"

Sigmoid functions

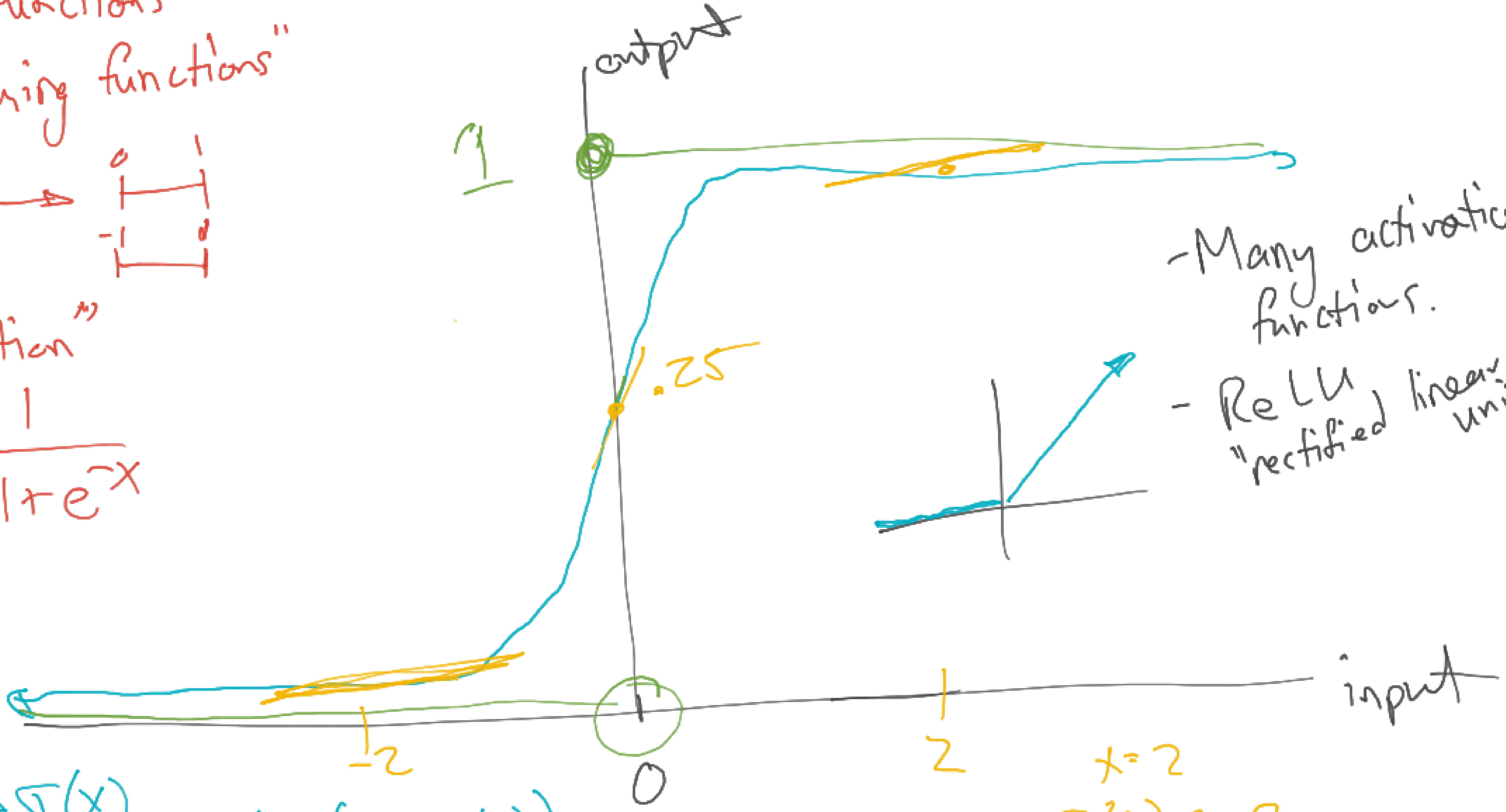
→ "squashing functions"

$\mathbb{R} \longleftrightarrow \;\to\; [0, 1]$, $[-1, 1]$

"Logistic function"

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

→ sigma



- Many activation functions.
- ReLU "rectified linear unit"

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

$x = -2$ $\sigma(x) \approx .1$

$.1(.9) = -.09$

$x = 0$

$\sigma(x) = 0.5$

$.5(.5) = .25$

$x = 2$

$\sigma(2) \approx .9$

$.9(1-.9) = .09$

$$f_{w_k}(x_i) = \sigma(in) \qquad in = \sum_{j=1}^{m} x_{i,j}\, w_{k,j}$$

$$f_{w_k}(x_i) = \frac{1}{1 + e^{-\sum_{j=1}^{m} x_{i,j} w_{k,j}}}$$

$$l(w_k) = \sum_{i=1}^{n} \underbrace{\left(y_i - \overbrace{\frac{1}{1 + e^{-\sum_{j=1}^{m} x_{i,j} w_{k,j}}}}^{g(i, w_k)} \right)}_{g(i, w_k)}{}^{2}$$

$$\frac{\partial\, l(w_k)}{\partial w_{k,j}} =$$

Chain rule.

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f}{\partial g} \frac{\partial g}{x}$$

$$\frac{\partial \ell(w_k)}{\partial w_{k,j}} = \sum_{i=1}^{n} \frac{\partial}{\partial w_{k,j}} \left(y_i - f_{w_k}(x_i)\right)^2$$

$$in_i = \sum_{\beta=1}^{m} x_{i\beta}\, w_{k\beta}$$

$$= \sum_{i=1}^{n} 2\left(y_i - f_{w_k}(x_i)\right) \frac{\partial f_{w_k}(x_i)}{\partial w_{k,j}}$$

$$f_{w_k}(x_i) = \sigma(in_i)$$

$$\frac{\partial \overbrace{f_{w_k}(x_i)}^{\sigma(in_i)}}{\partial w_{k,j}} = \boxed{\frac{\partial f_{w_k}(x_i)}{\partial in_i}} \boxed{\frac{\partial in_i}{\partial w_{k,j}}}$$

$$\frac{\partial f_{w_k}(x_i)}{\partial in_i} \rightarrow \sigma(in_i)\left(1 - \sigma(in_i)\right)$$

$$\frac{\partial in_i}{\partial w_{k,j}} \rightarrow \frac{\partial\, in_i}{\partial w_{k,j}} = \frac{\partial}{\partial w_{k,j}} \sum_{\beta=1}^{m} x_{i\beta}\, w_{k\beta}$$

$$= \sum_{\beta=1}^{n} \frac{\partial}{\partial w_{k,j}} x_{i\beta}\, w_{k\beta}$$

$$= \frac{\partial}{\partial w_{k,j}} x_{ij}\, w_{kj}$$

$$= x_{ij}$$

"Intuition"

$$\frac{\partial \ell(w_k)}{\partial w_{k,j}} = \frac{\partial \ell(w_k)}{\partial f_w} \frac{\partial f_w}{\partial in} \frac{\partial in}{\partial w_{kj}}$$

ish

$$\frac{\partial \ell(w_k)}{\partial w_{k,j}} = \underbrace{\sum_{i=1}^{n} 2\left(y_i - f_{w_k}(x_i)\right)}_{\frac{\partial \ell(w_k)}{\partial f_{w_k}(x_i)}\text{ -ish}} \underbrace{\sigma(in_i)\left(1-\sigma(in_i)\right)}_{\frac{\partial f_{w_k}(x_i)}{\partial in_i}} \underbrace{x_{i,j}}_{\frac{\partial in_i}{\partial w_{k,j}}}$$

ignoring $\sum_i$ different x_i.

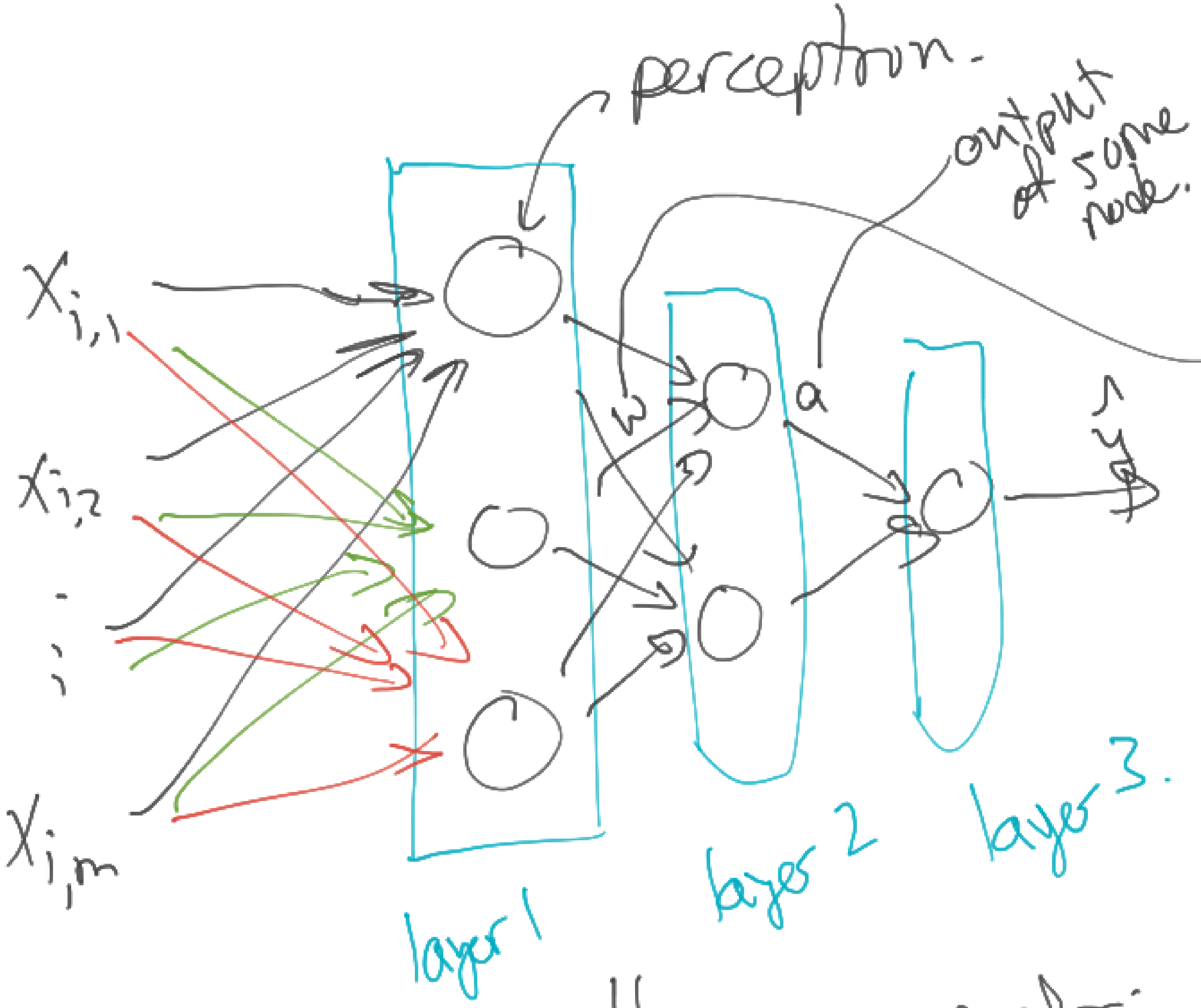
- Gradient descent on least squares objective using a perceptron for the parametric model.

Artificial Neural Networks (ANN).

(Deep learning)

- Fully connected
- Feed-forward

ANN.



W_{ij}

↳ from i^{th} neuron in prev layer to j^{th} in current layer.

Hyperparameter: structure of network.