Universidad Nacional de San Martín

ICAS

Técnicas estadísticas para el análisis de datos astrofísicos

Rodrigo F. Díaz
International Center for Advanced Studies (ICAS)
University of San Martín (UNSAM), Argentina

rdiaz@unsam.edu.ar

Primera Escuela Latinoamericana de Relatividad y Asrofísica
Diciembre de 2021

Model extension

Multiple linear model

Simple Linear Regression

$$y(x, w_0, w_1) = w_0 + w_1 x \quad .$$

Multiple Linear Regression

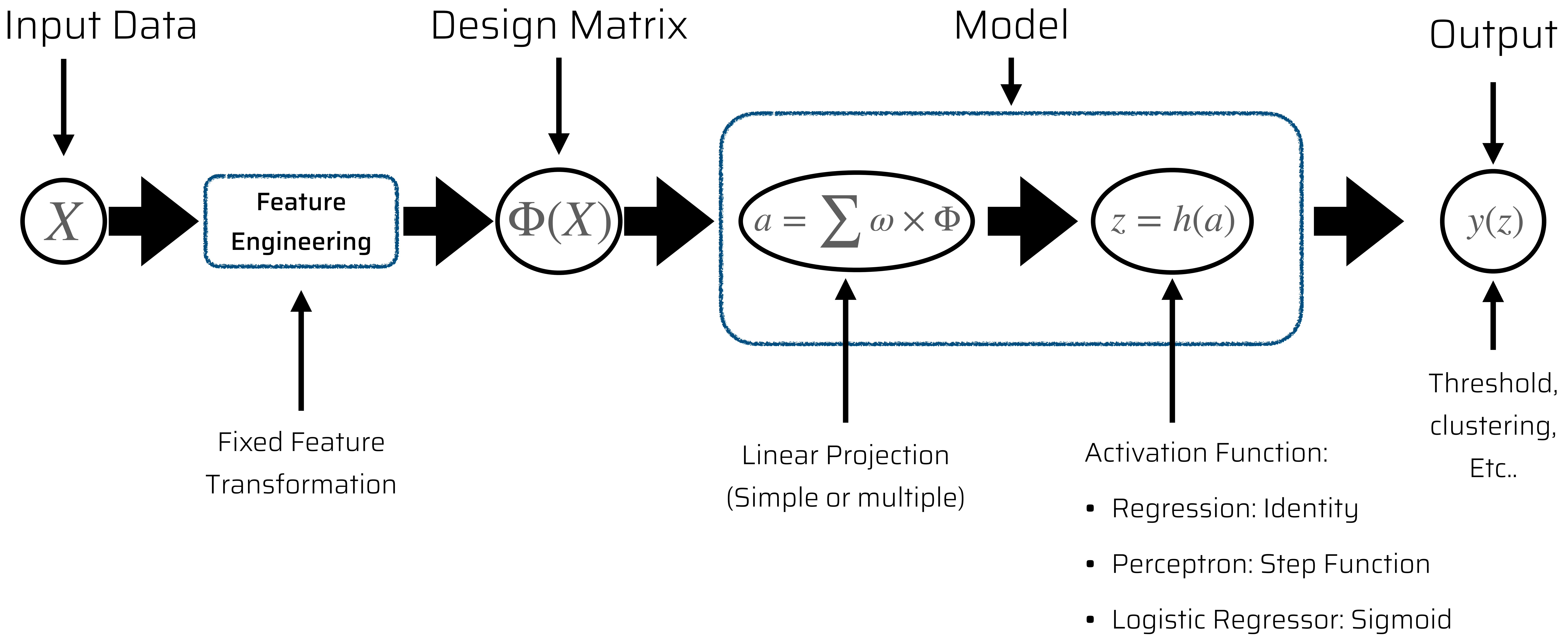
$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1 x_1 + \cdots + w_D x_D \quad .$$

More generally:

$$y_i(\boldsymbol{x}, \boldsymbol{w}) = w_0 + \sum_{i=1}^{D} w_j \phi_j(\boldsymbol{x_i}) = \sum_{j=0}^{D} w_j \phi_j(\boldsymbol{x_i}) = \boldsymbol{w}^T \boldsymbol{\phi}_i$$

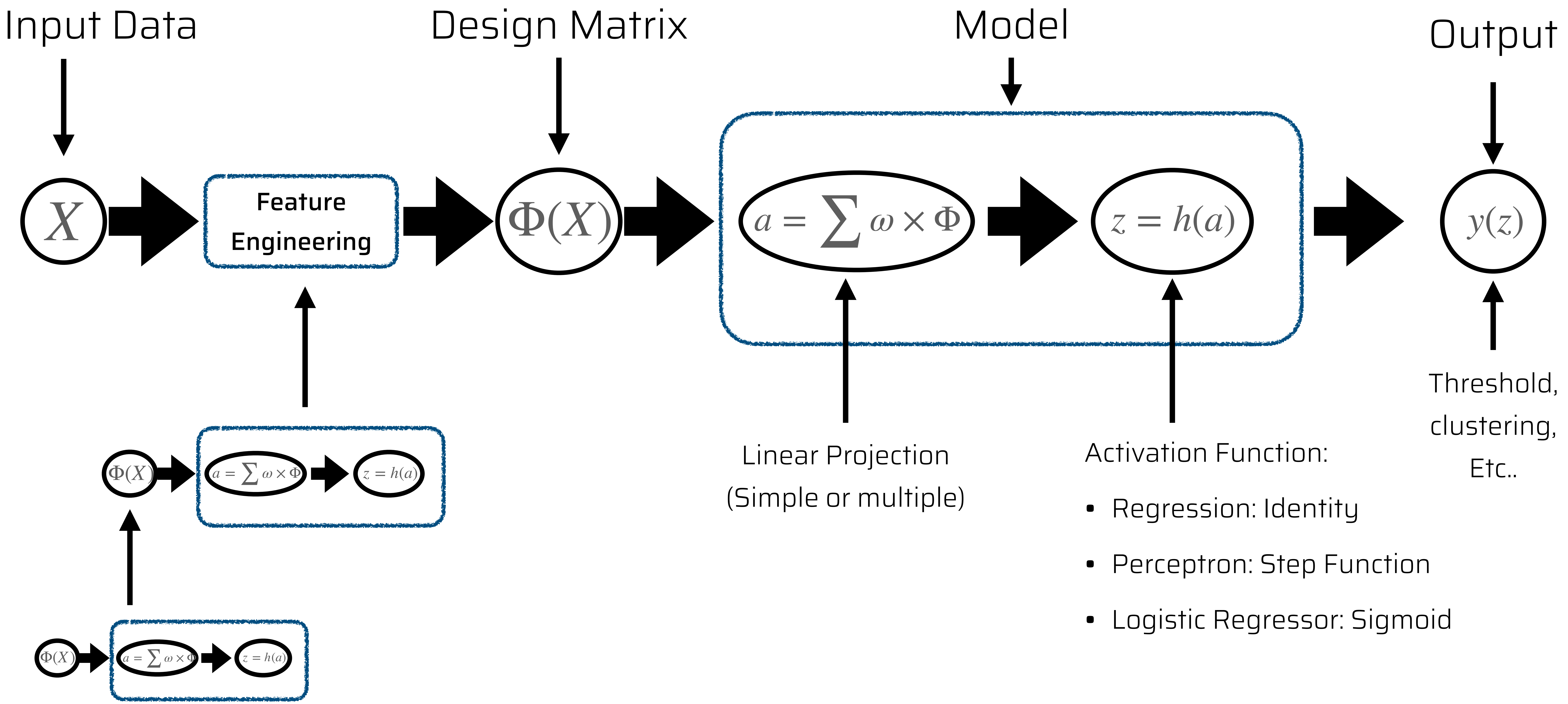
Models so far

(Most of them)



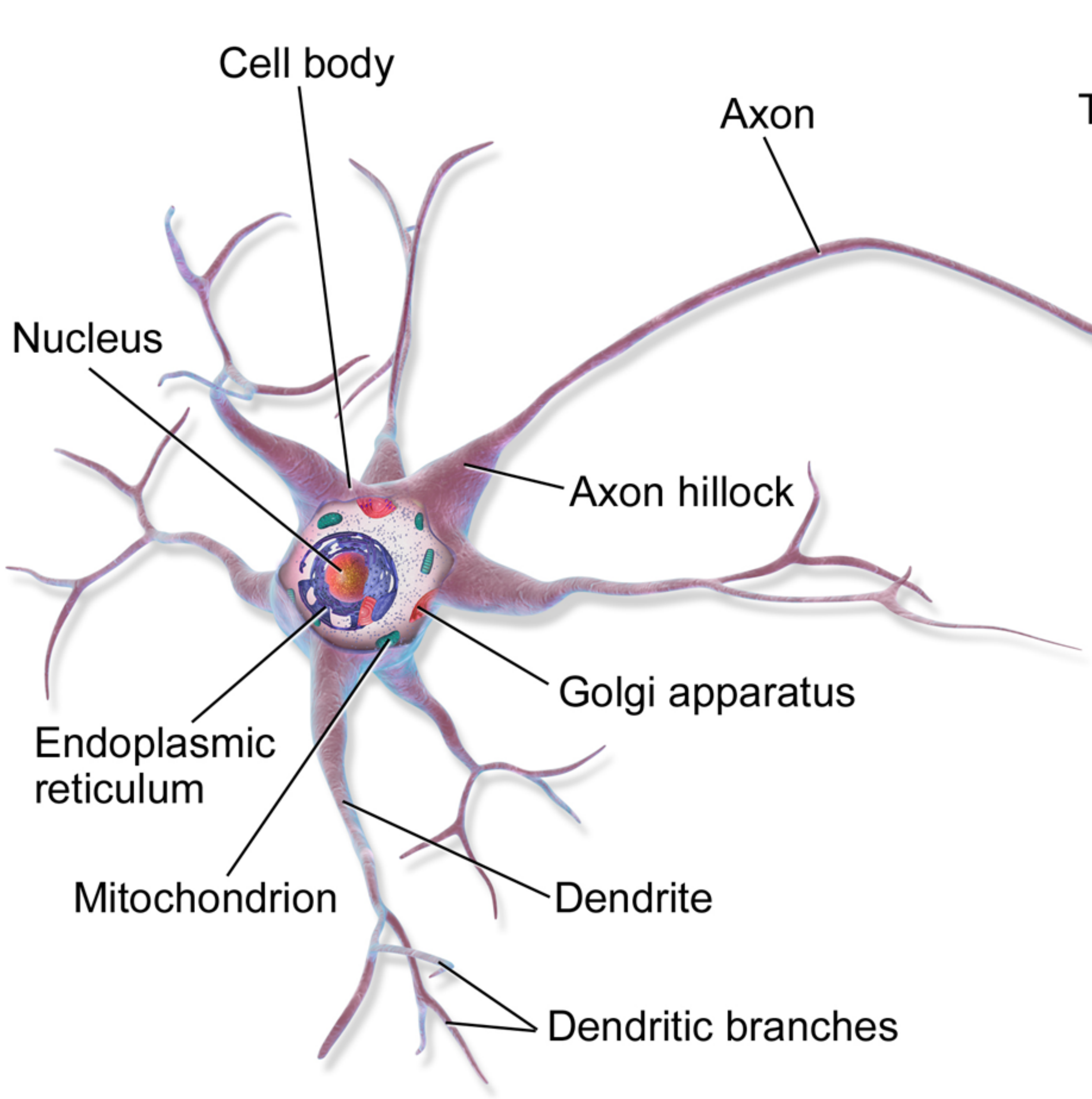
Motivation for Neural Networks

Fitting the Features

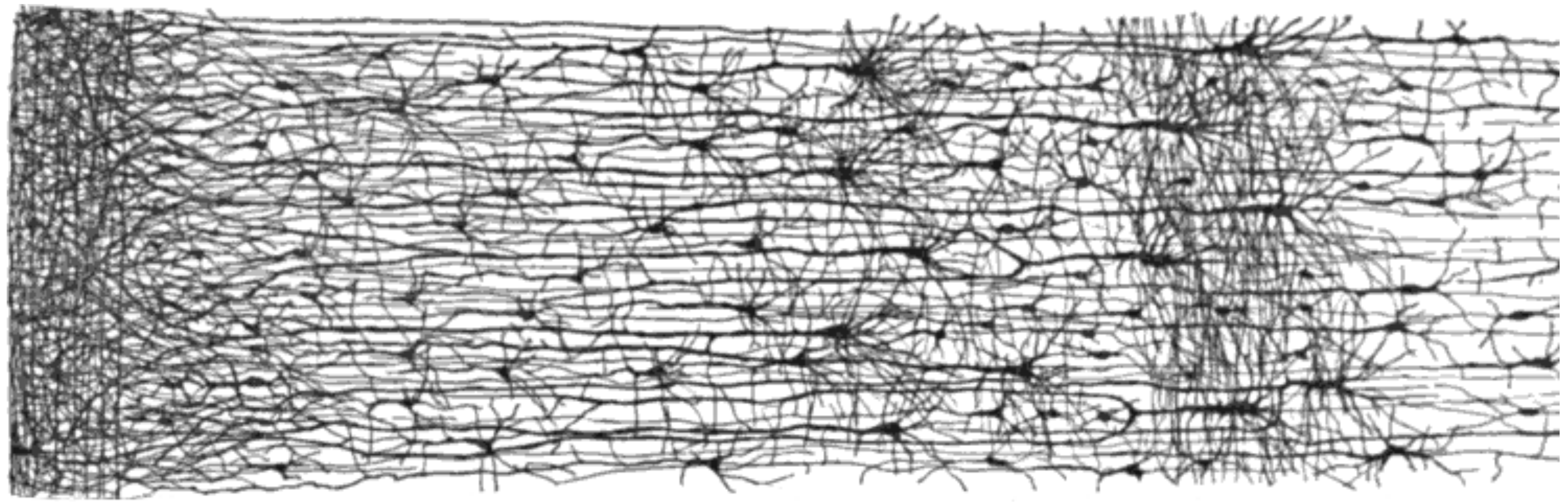


Biological Neural Networks

Where do you get your inspiration from?



Ramón y Cajal



- Image by Bruce Blaus (Creative Commons 3.0). Reproduced from *https://en.wikipedia.org/wiki/Neuron*.

Neural Networks

Some History

1943. McCulloch and Pitts. A simplified computational model of how biological neurons might work together in animal brains to perform complex computations using propositional logic. This was the first artificial neural network architecture.

1957. **Rosenblatt**. One of the first algorithms based on the behaviour of physical neurons. See *The Organization of Behavior*, by Donald Hebb.

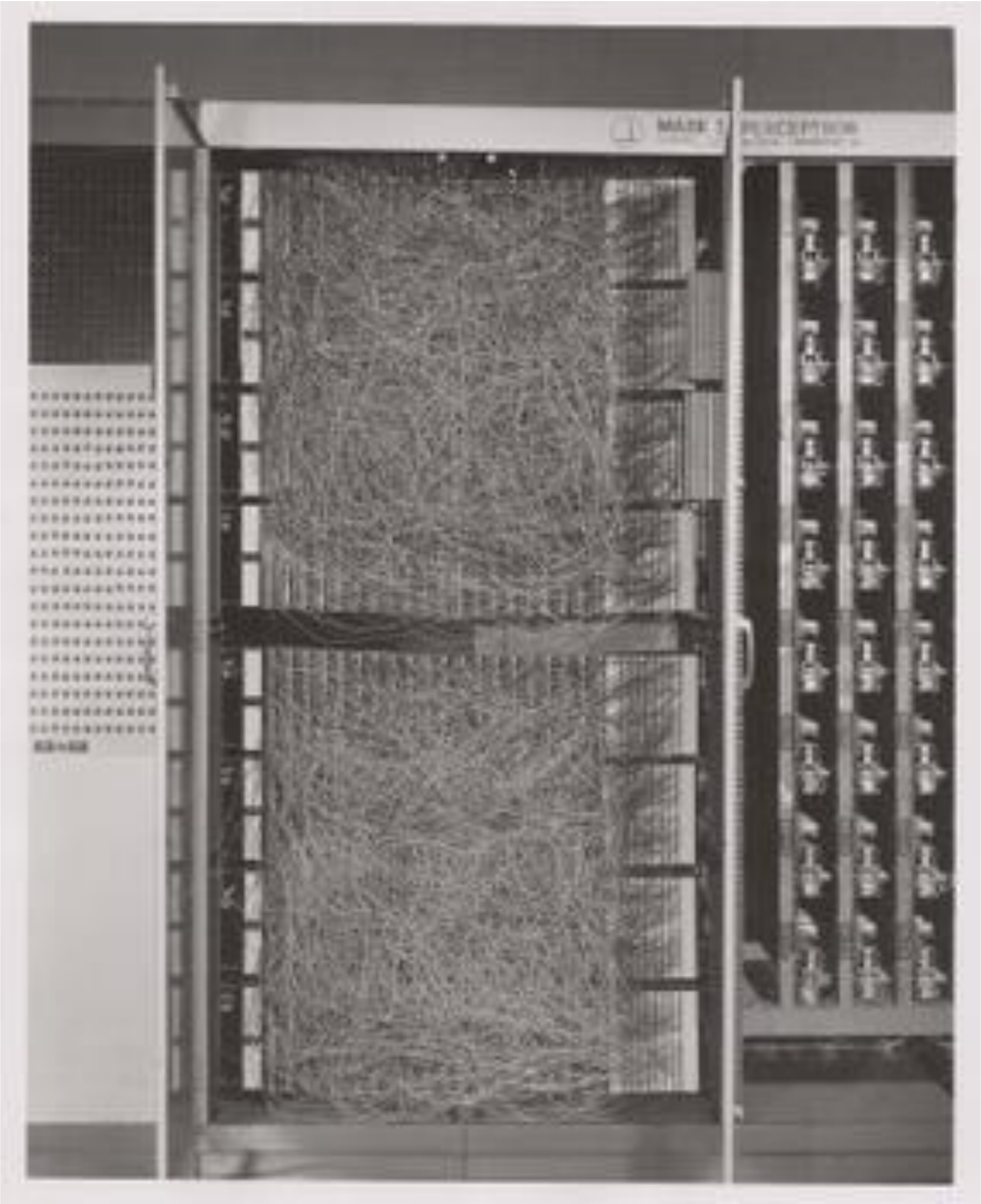
Implemented as a program, but later turned into a machine.

1958. NYT: "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence."

1969, **Minsky and Papert.** In their book *Perceptrons* they show that the perceptron cannot learn the XOR function. The dark age of neural networks begins.

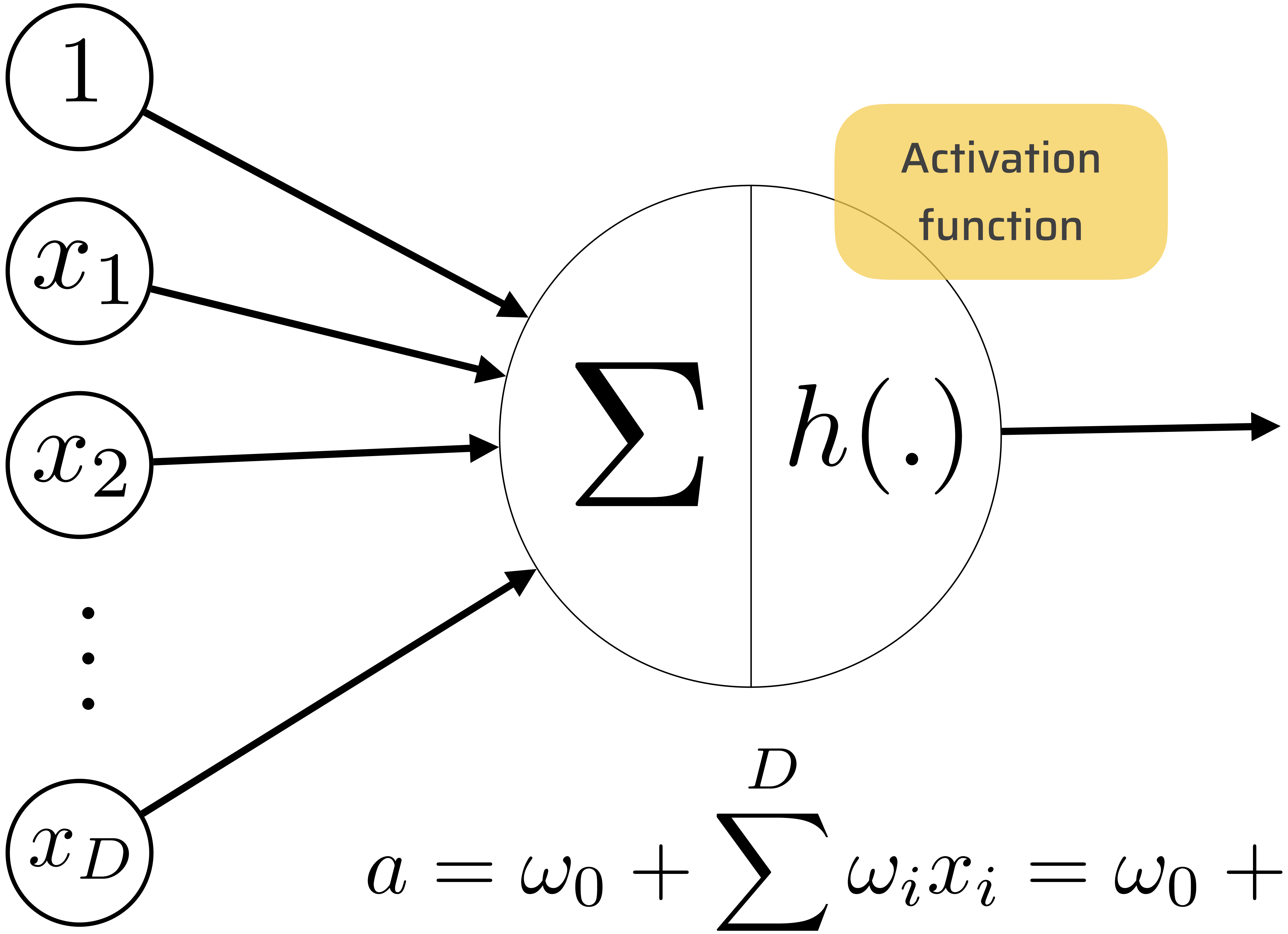
1980s. First comeback of Neural Networks, but quickly surpassed by other algorithms such as Support Vector Machines

2000s. Second comeback, fuelled by the increase in computational power and data availability, together with efficient training techniques.



Neurons / Units

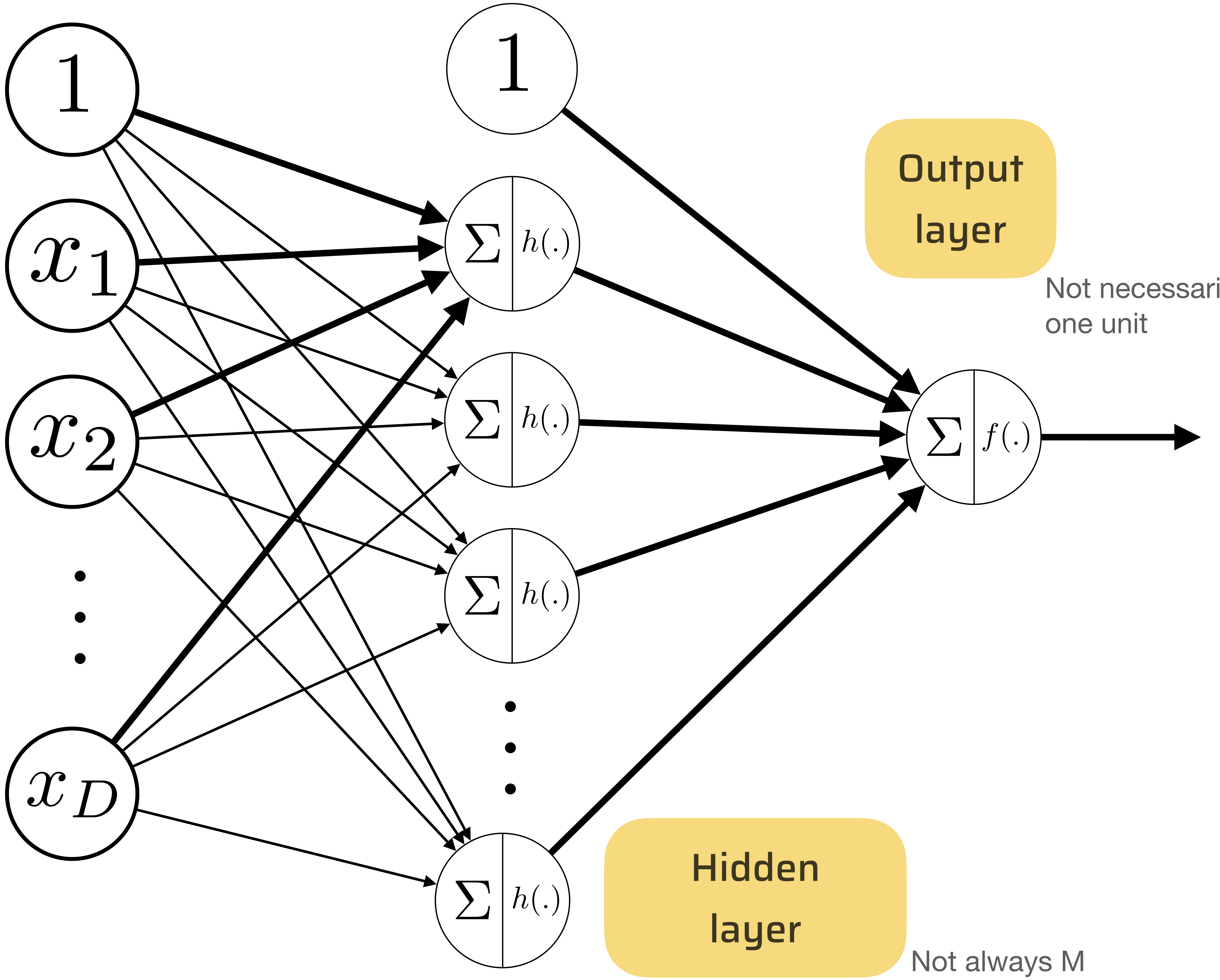
The basic component of ANNs



$$z = h(\omega_0 + \boldsymbol{\omega}^T \boldsymbol{x})$$

$$a = \omega_0 + \sum_{i=1}^{D} \omega_i x_i = \omega_0 + \boldsymbol{\omega}^T \boldsymbol{x}$$

Feed-forward, fully-connected ANN



Input layer

Hidden layer

Not always M units

Output layer

Not necessarily one unit

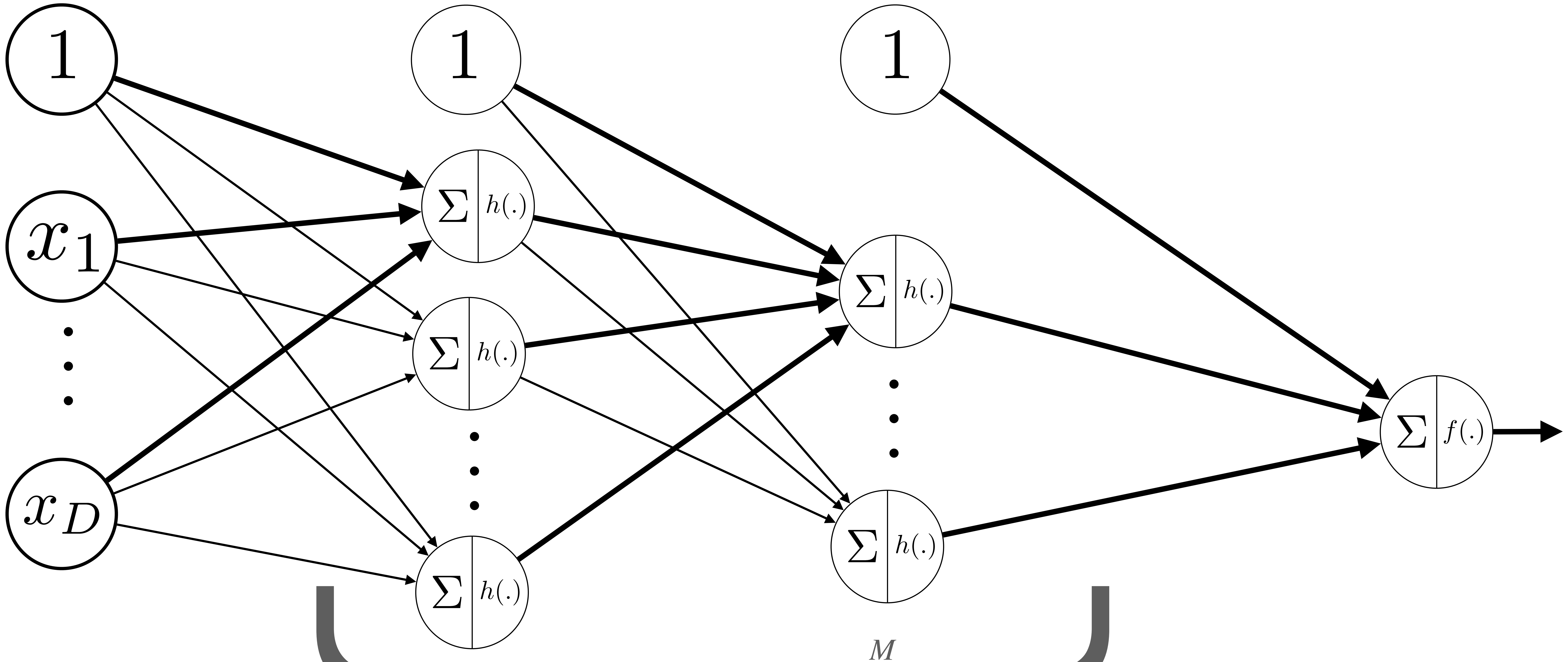
Feed-forward fully-connected

$$y = f\left(w_0^{(2)} + \sum_{j=1}^{M} w_j^{(2)} z_j \right)$$

where

$$z_j = h\left(w_{j0}^{(1)} + \sum_{i=1}^{D} w_{ji}^{(1)} x_i \right)$$

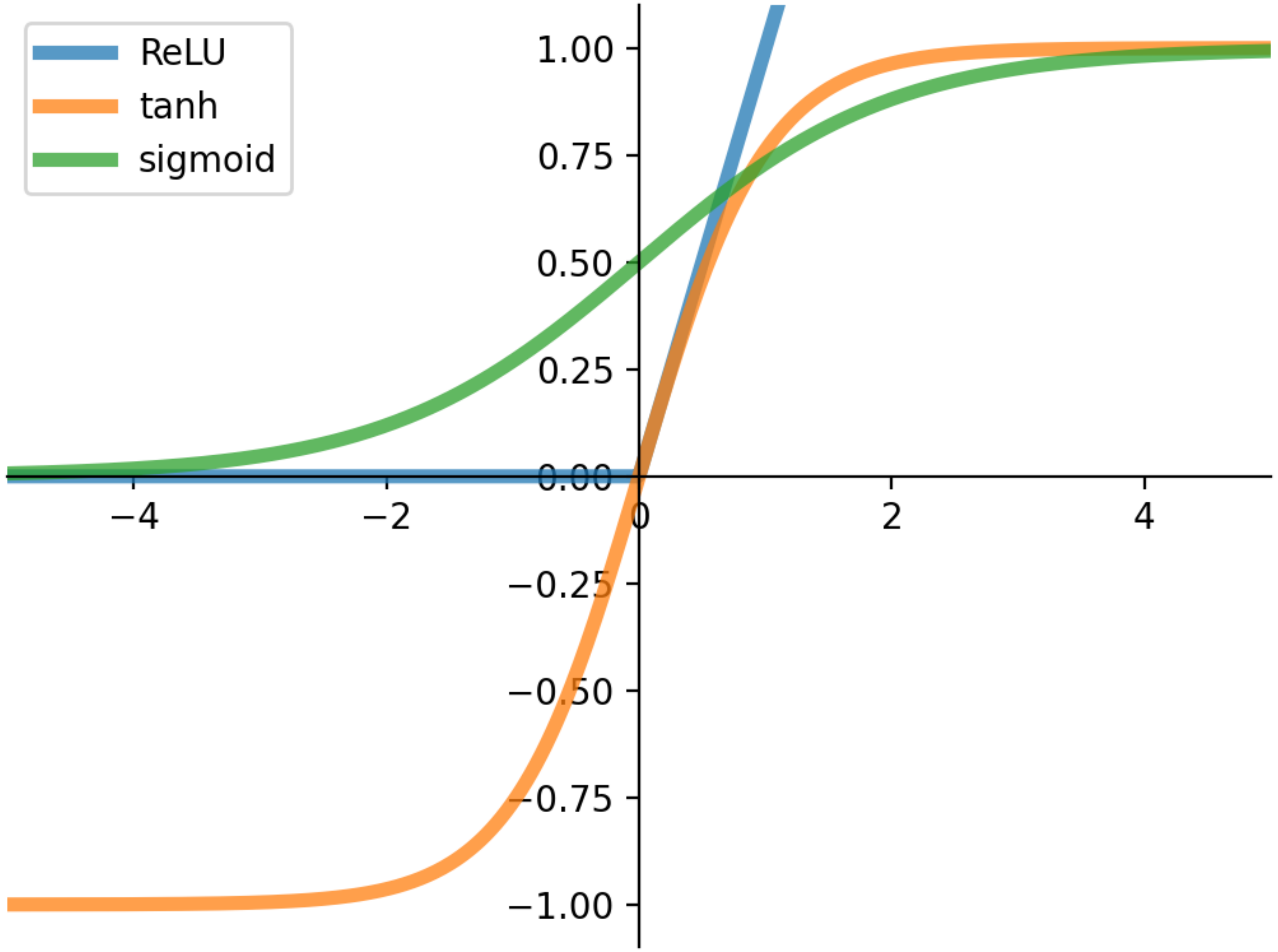
Parametrisation of the basis functions



$$\boldsymbol{\phi}(\boldsymbol{x})$$

Activation functions

Hidden Layers



Tanh

$$\tanh(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}}$$

Sigmoid

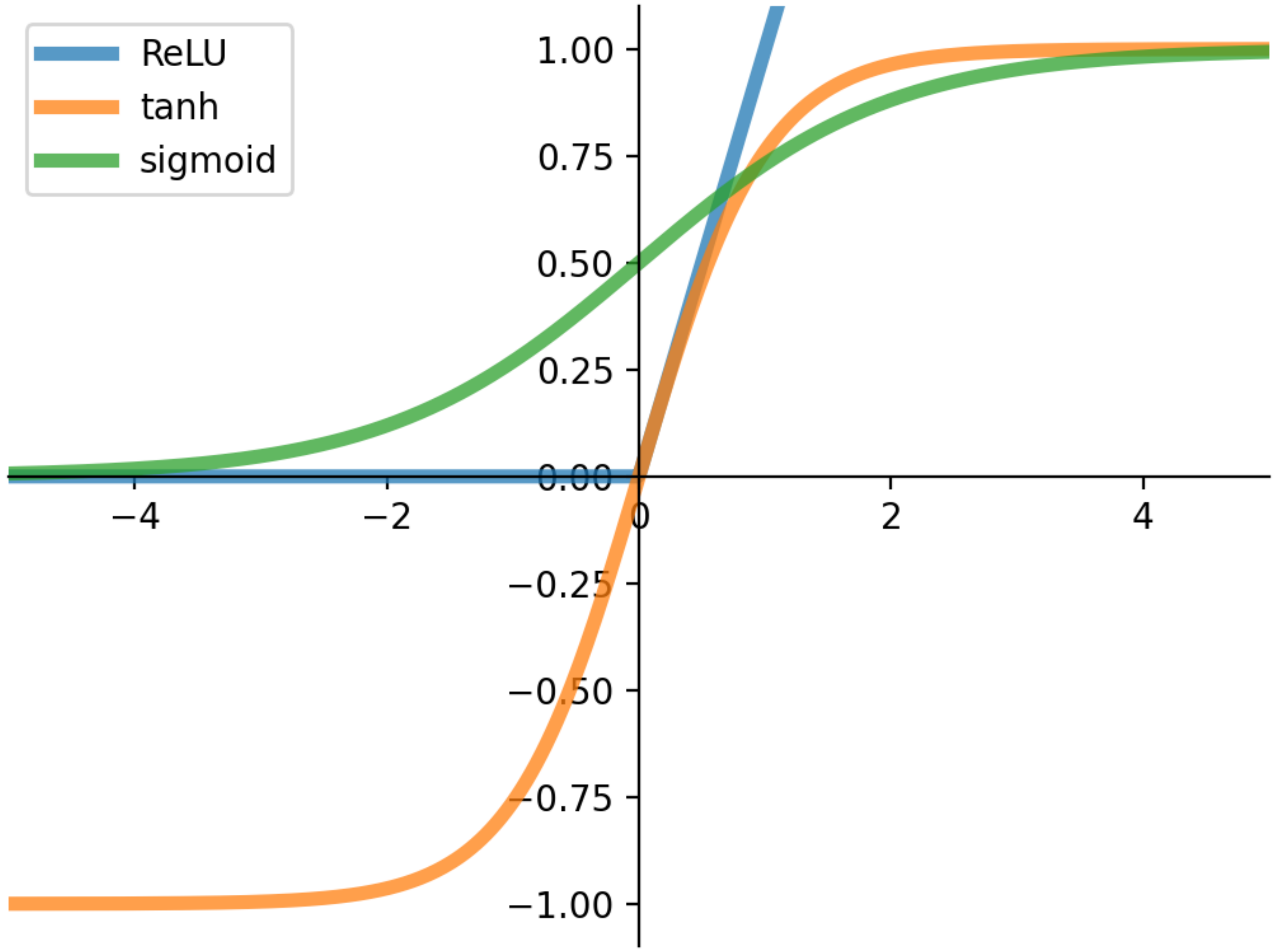
$$\sigma(a) = \frac{1}{1 - e^{-a}}$$

ReLU

$$r(a) = \max(0, a)$$

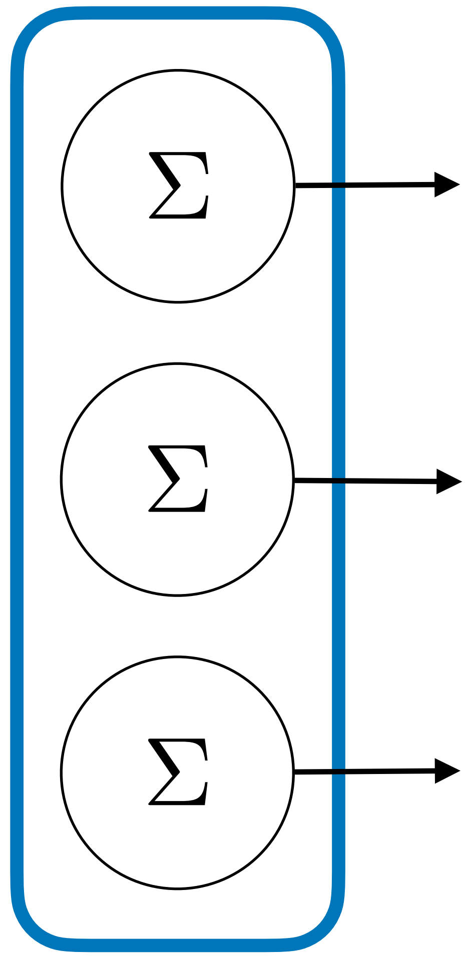
Activation functions

Output Layers



	Output Layer		
Problem	**Size**	**Activation**	**Error**
Regression	N	$f(x) = x$	MSE RMSE
Binary Classification	1	$f(x)$ = sigmoide	Cross-entropy
Multi-class classification	K	$f(x)$ = softmax	Multiclass Cross-entropy

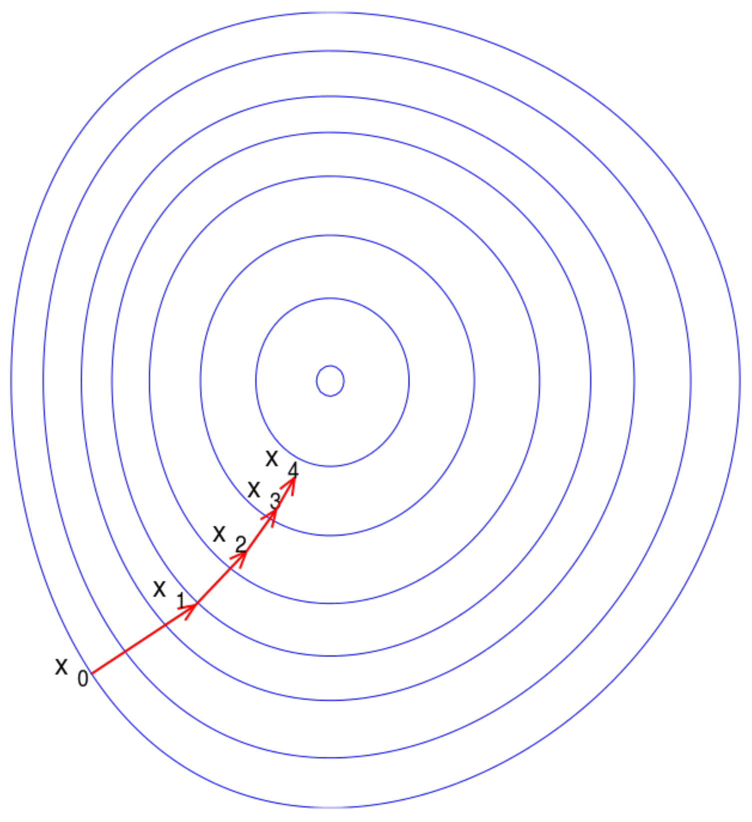
$$s(z_i) = \frac{\exp(z_i)}{\sum_{k=1}^{K} \exp(z_j)}$$



Training

	Output Layer		
Problem	**Size**	**Activation**	**Error**
Regression	N	$f(x) = x$	MSE RMSE
Binary Classification	1	$f(x) =$ sigmoide	Cross-entropy
Multi-class classification	K	$f(x) =$ softmax	Multiclass Cross-entropy

Gradient descent



- Training is equivalent to minimising the error function.
- Because of the large number of parameters in the model, regularisation is often used in the hidden layer weights (L2, L1 or ElasticNet).
- Complex, non-convex function.
  - Many local minima, saddle points, etc.
- The gradient can be efficiently computed using error back propagation (a.k.a. chain rule).
- Gradient methods are often used (see Gradient Descent example, or SGD).
- Complexity of function calls for additional tricks (e.g. scheduled learning rate) and specific algorithms.