Guided Tour of Machine Learning in Finance

Week 2 - Lesson 1 - part 4: Neural Networks

Igor Halperin

NYU Tandon School of Engineering, 2017

Linear Regression as a functional transform

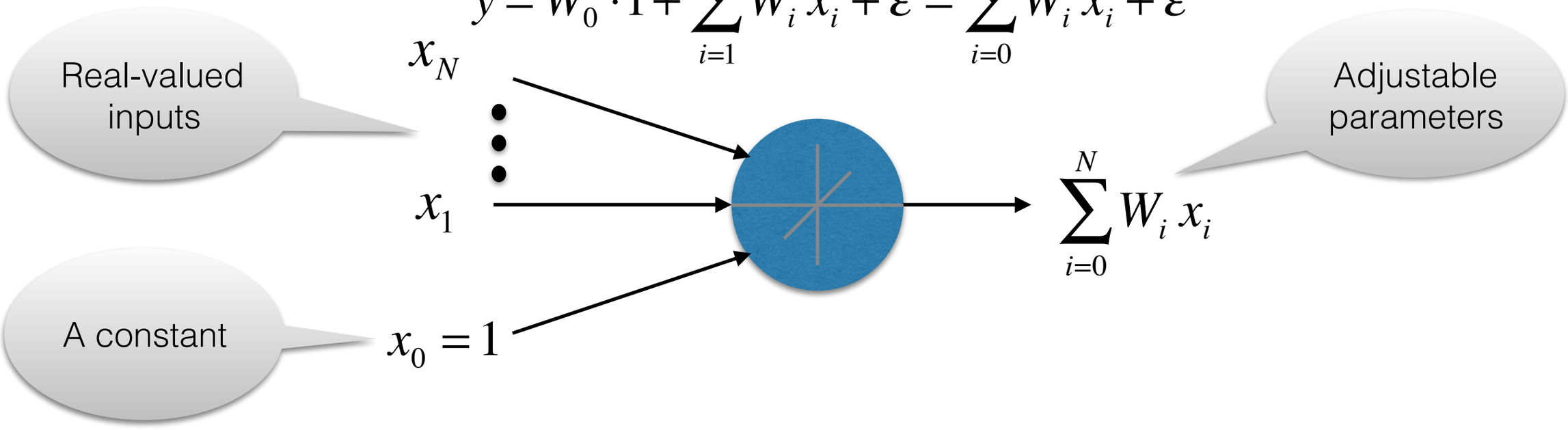
- **Inputs**: real-valued numbers
- **Output:** a real-valued number
- Can be thought of as a node computing a **linear function** of inputs
- Weights can be tuned to fit the data

$$y = W_0 \cdot 1 + \sum_{i=1}^{N} W_i x_i + \varepsilon = \sum_{i=0}^{N} W_i x_i + \varepsilon$$

Linear Regression as a functional transform

- **Inputs**: real-valued numbers
- **Output:** a real-valued number
- Can be thought of as a node computing a **linear function** of inputs
- Weights can be tuned to fit the data

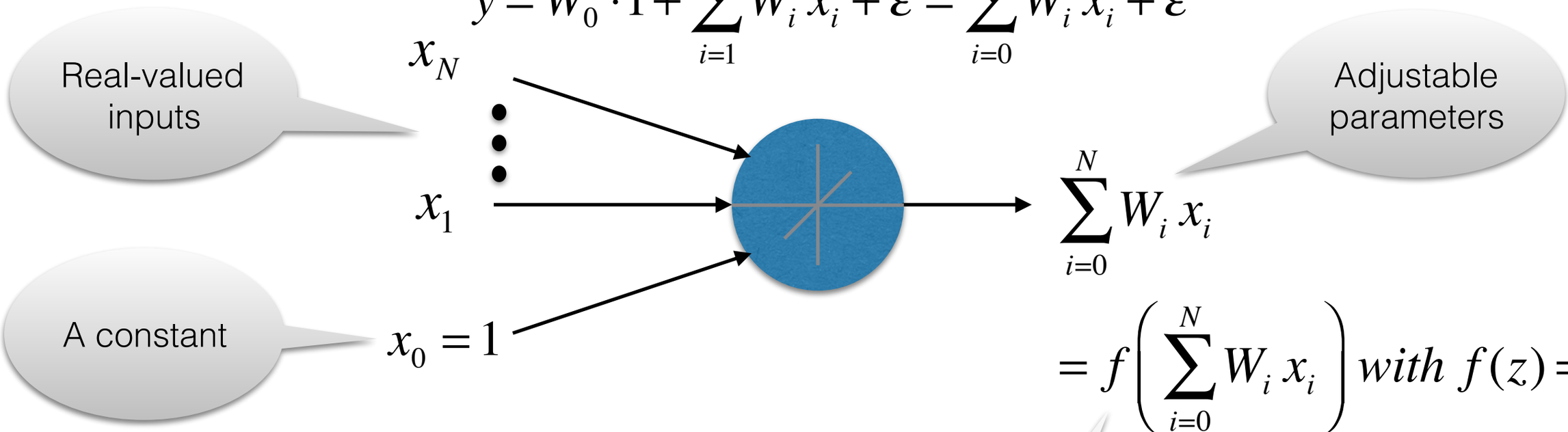
$$y = W_0 \cdot 1 + \sum_{i=1}^{N} W_i x_i + \varepsilon = \sum_{i=0}^{N} W_i x_i + \varepsilon$$



Linear Regression as a functional transform

- **Inputs**: real-valued numbers
- **Output:** a real-valued number
- Can be thought of as a node computing a **linear function** of inputs
- Weights can be tuned to fit the data

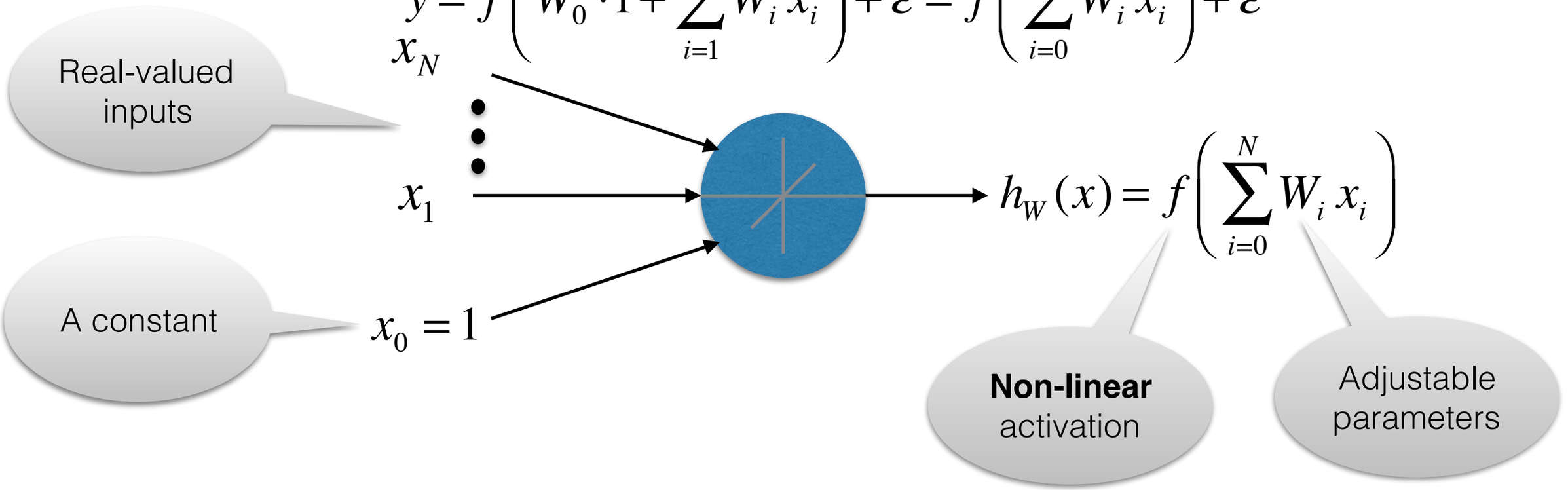
$$y = W_0 \cdot 1 + \sum_{i=1}^{N} W_i x_i + \varepsilon = \sum_{i=0}^{N} W_i x_i + \varepsilon$$



Non-linear Regression as a functional transform

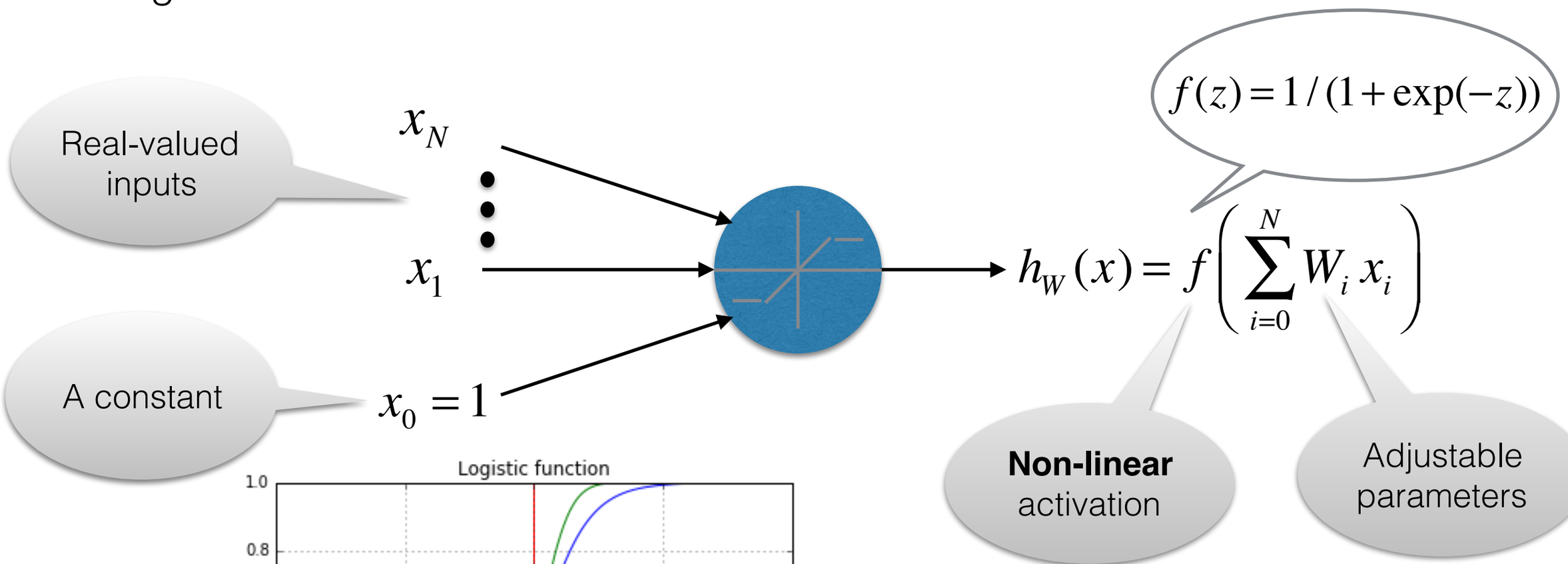
- **Inputs**: real-valued numbers
- **Output:** a real-valued number
- Can be thought of as a node computing a **non-linear function** of inputs
- Weights can be tuned to fit the data

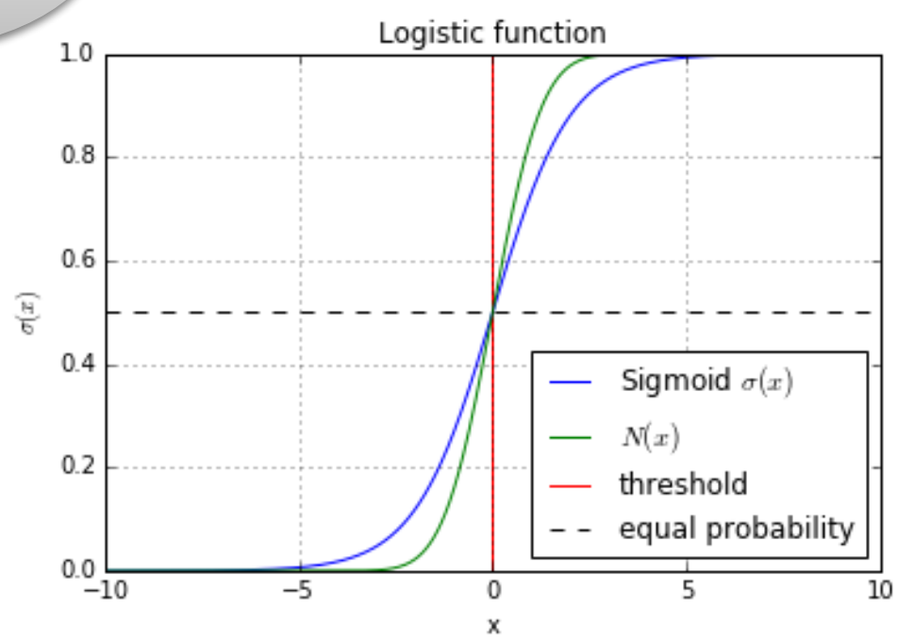
$$y = f\left(W_0 \cdot 1 + \sum_{i=1}^{N} W_i x_i \right) + \varepsilon = f\left(\sum_{i=0}^{N} W_i x_i \right) + \varepsilon$$



Logistic Regression as a functional transform

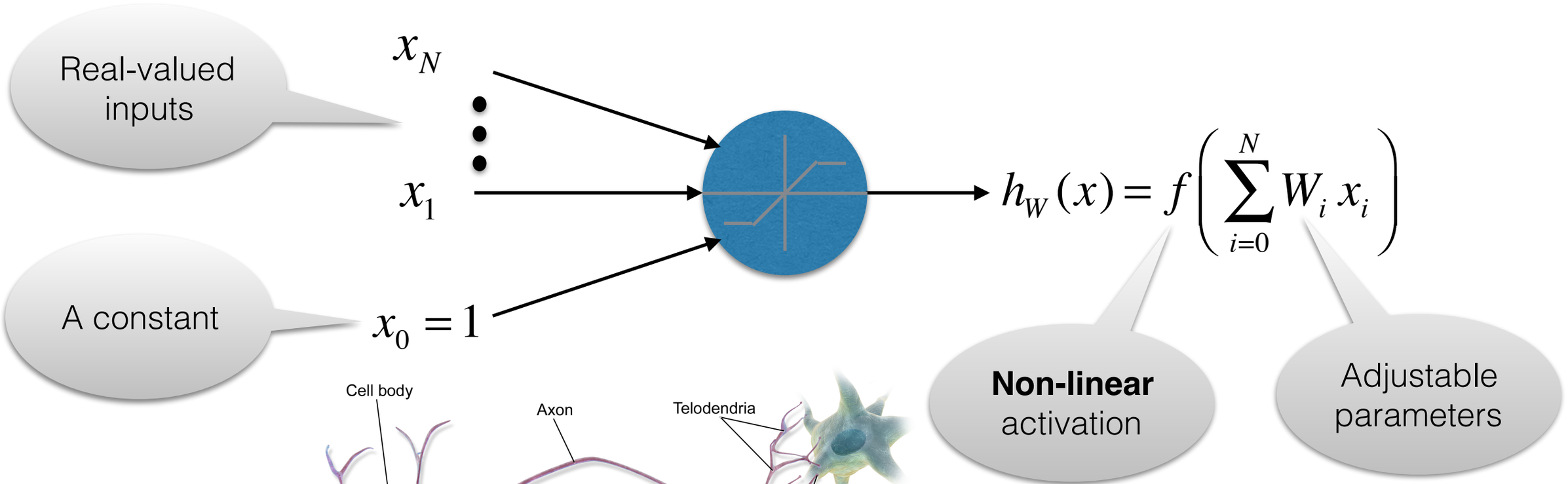
- **Inputs**: real-valued numbers
- **Output:** a real-valued number
- Can be thought of as a node computing a **non-linear function** of inputs
- Weights can be tuned to fit the data

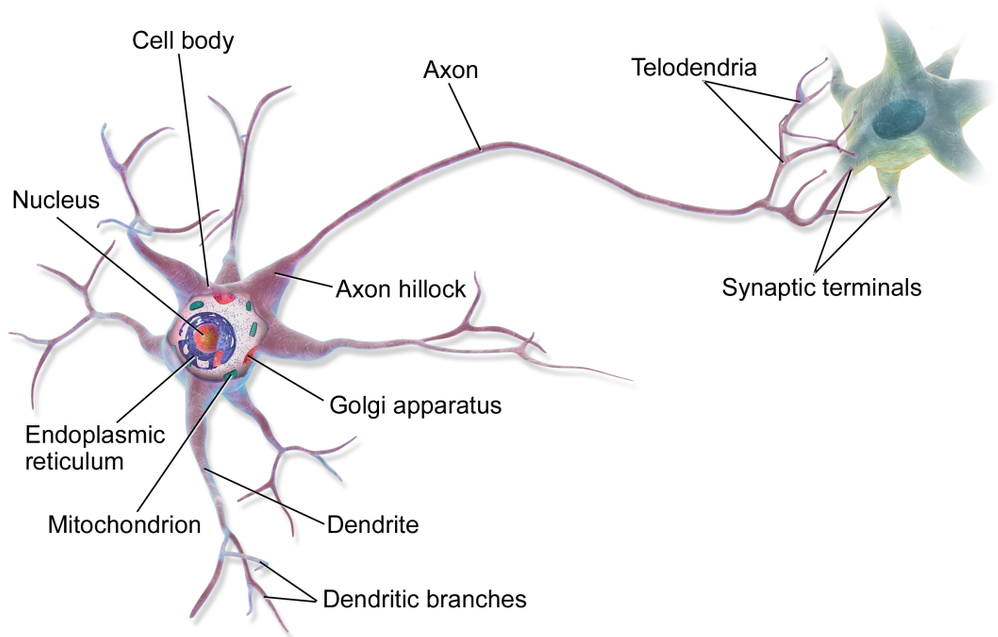




Artificial Neuron

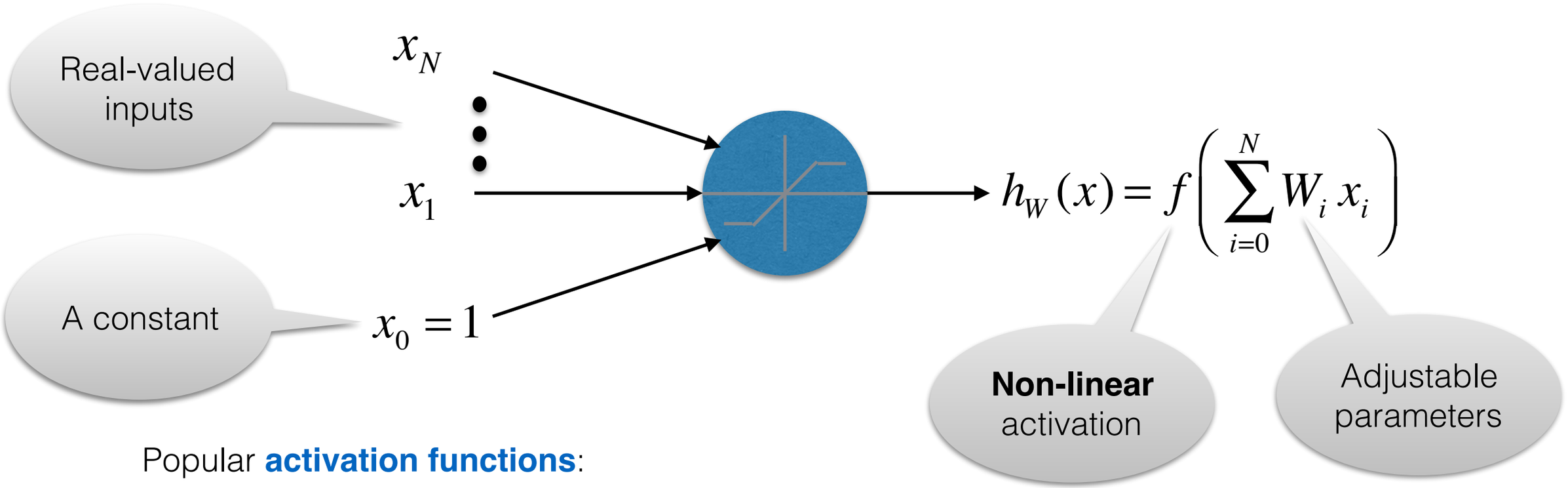
- **Artificial Neuron:** (perceptron): a function implementing a non-linear transformation of its input data (Rosenblatt, 1957)
- Can be viewed as a caricature of a physical neuron





Artificial Neuron

- **Artificial Neuron:** (perceptron): a function implementing a non-linear transformation of its input data (Rosenblatt, 1957)
- Can be viewed as a caricature of a physical neuron



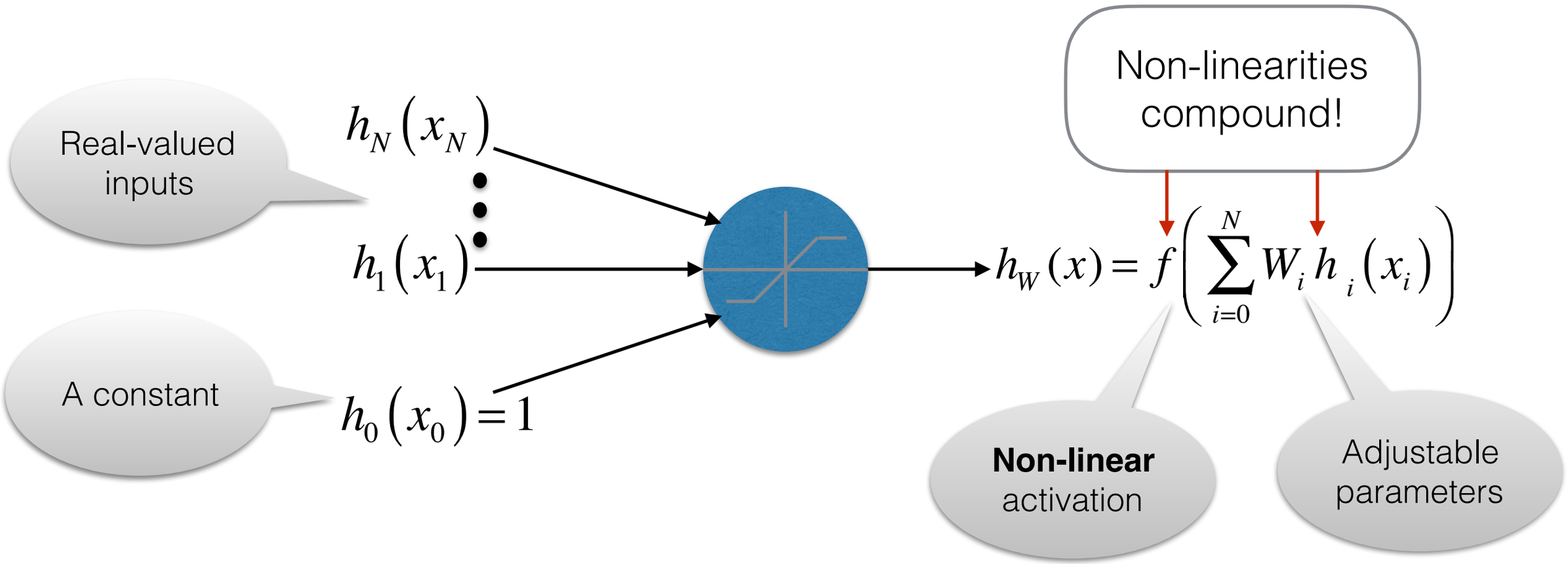
Popular **activation functions**:

- Sigmoid (logistic) function $f(z) = 1/(1+\exp(-z))$
- Tanh function $f(z) = (e^z - e^{-z})/(e^z + e^{-z})$
- Rectified linear neuron $f(z) = \max(z,0) \equiv z_+$

Logistic regression

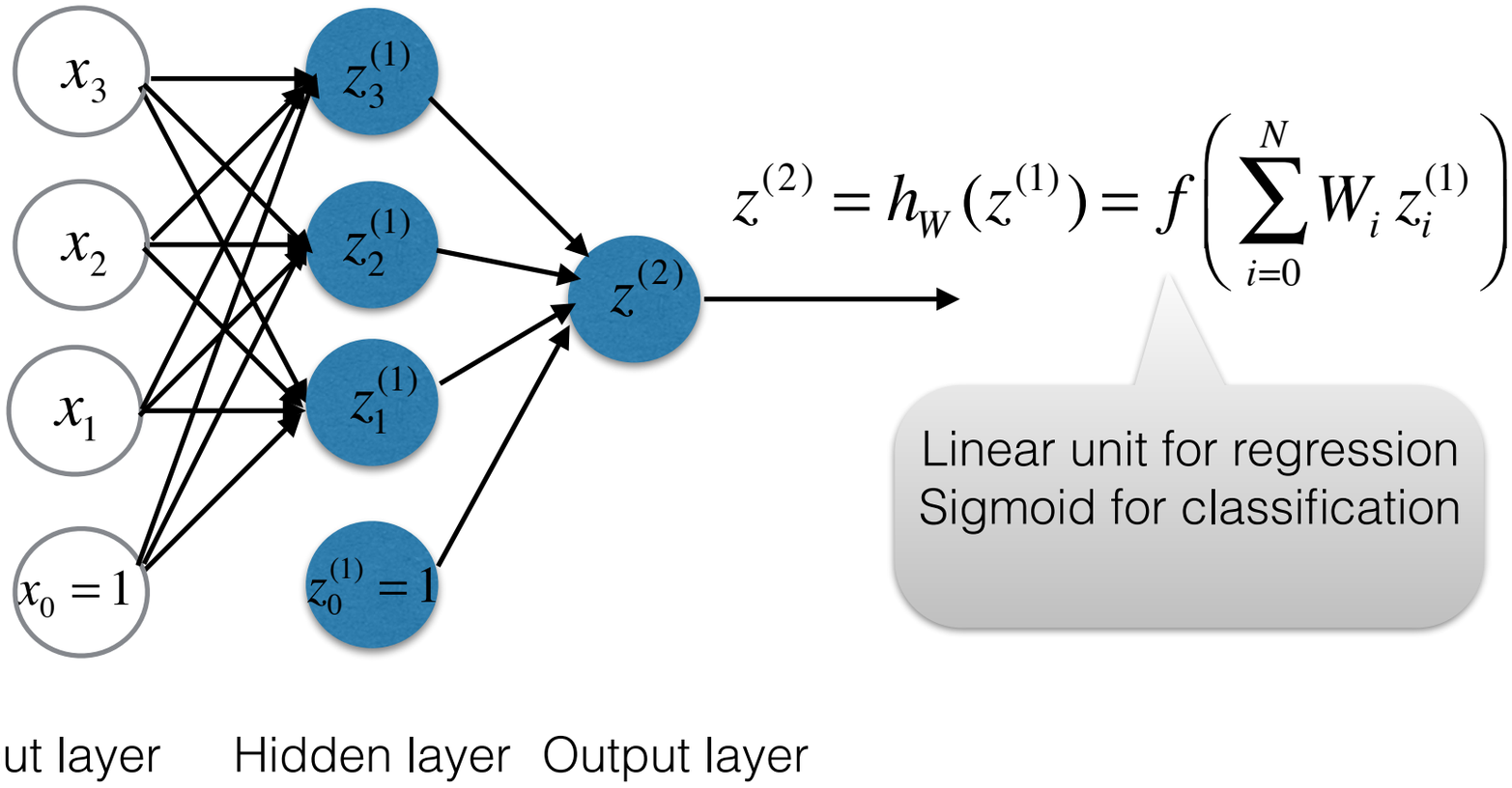
Composite function transform with a neuron

- Inputs to a neuron can themselves be non-linear transforms of raw inputs
- The output will have a composite non-linearity
- Can fit very complex functions



Artificial Neural Network

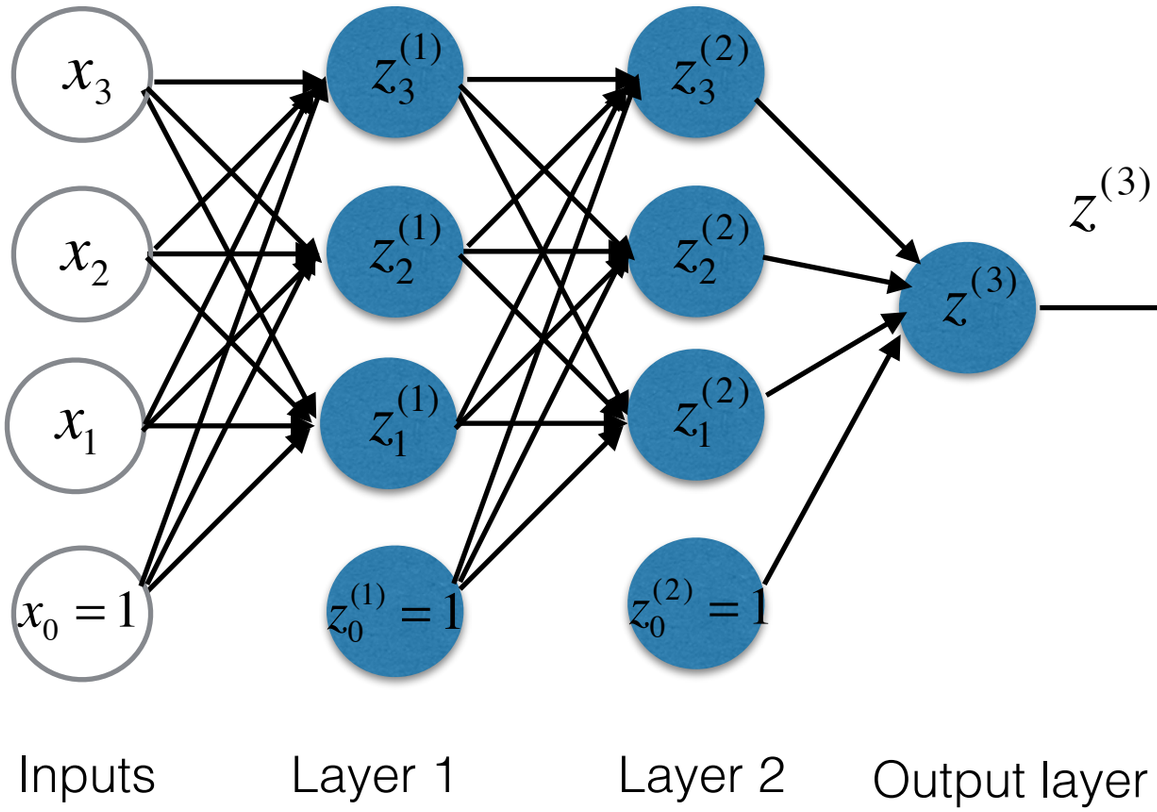
- **Feedforward Neural Network** is composed of layers of artificial neurons (perceptrons)



- This is a single (hidden) layer feedforward network
- Artificial Neural Network (ANN) is a highly stylized model of how the neocortex in mammals' brain processes visual and audio signals
- **Logistic regression = No hidden layer**

Neural Networks with more layers

- **Feedforward Neural Network** is composed of layers of artificial neurons (perceptrons)



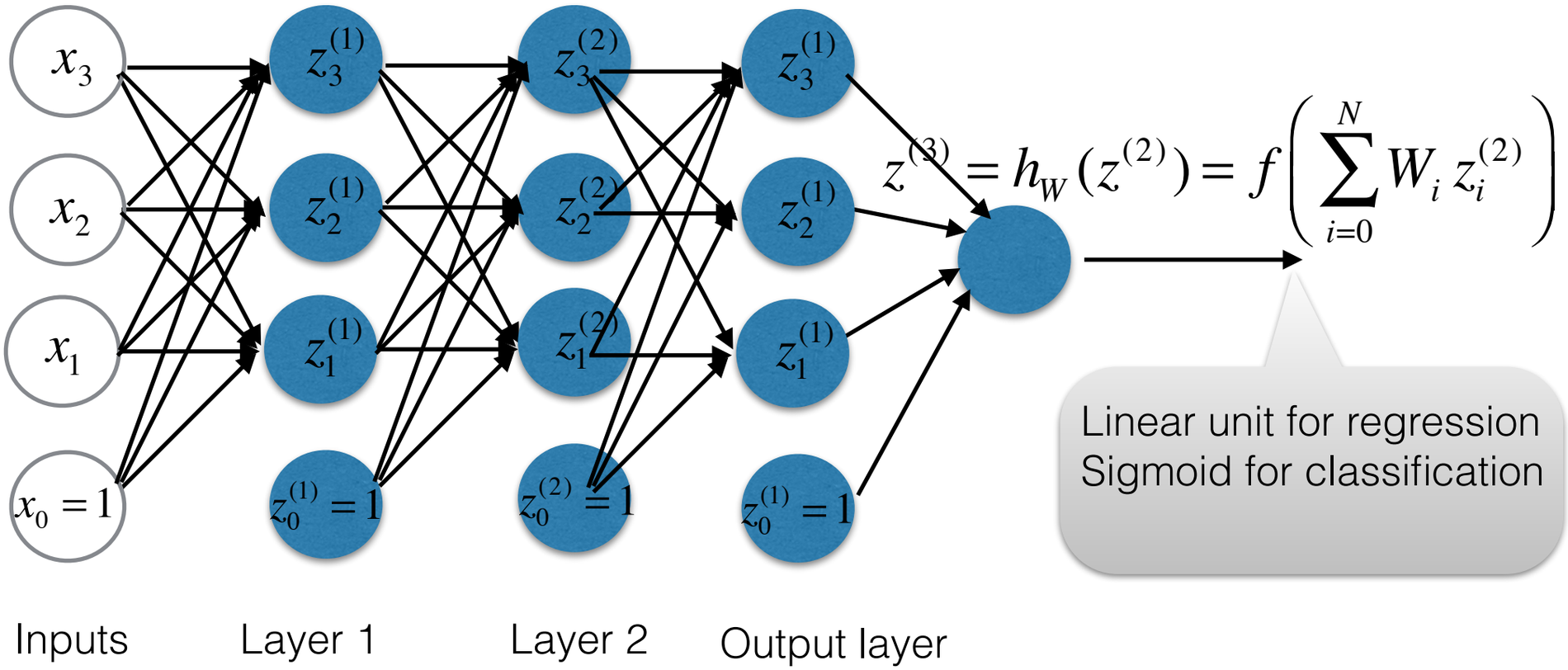
$$z^{(3)} = h_W(z^{(2)}) = f\left(\sum_{i=0}^{N} W_i \, z_i^{(2)}\right)$$

Linear unit for regression
Sigmoid for classification

- This is a two-layer feedforward network

Deep Neural Networks

- **Feedforward Neural Network** is composed of layers of artificial neurons (perceptrons)



- More than 2 hidden layers = **Deep Neural Network (Deep Learning)**
- Trained by **Gradient Descent**

Control question

Q: Select all correct statements:
1. Linear Regression can be viewed as a Neural Network with just one “linear neuron” (a node with a linear activation function).
2. Logistic regression can be viewed as a Neural Network with just one sigmoid neuron.
3. Deep Neural Networks are obtained when there are more than two hidden layers.
4. A Deep Linear Regression can be obtained if we put “linear neurons” in a hierarchical structure with at least three hidden layers.

Correct answer: 1,2 and 3.