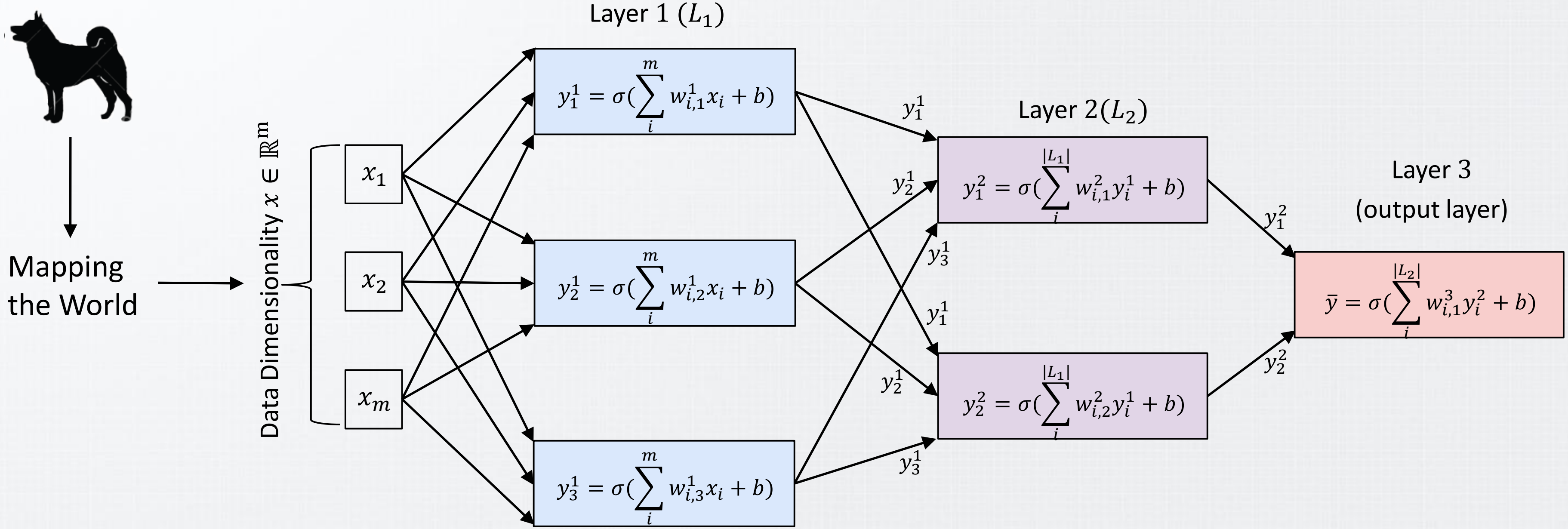
Universidade de São Paulo
Escola Politécnica - Engenharia de Computação e Sistemas Digitais

FeedForward Networks

Prof. Artur Jordão

Overview

FeedForward Networks



Components

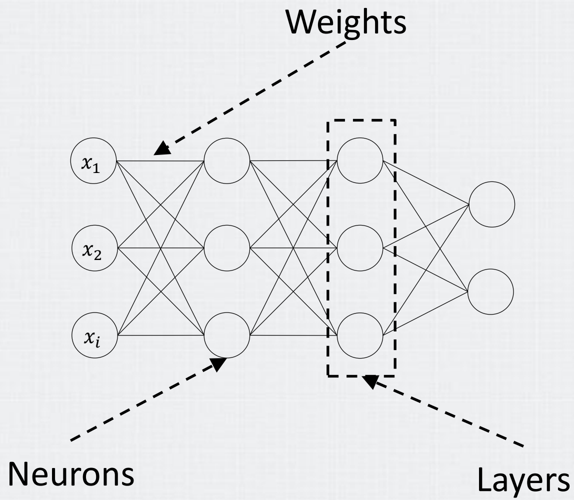
FeedForward Networks

- Weight vectors (weights for short)
  - Real values (randomly initialized)

- Neurons
  - Units composed of weights that receive a set of inputs and perform dot product
  - Each neuron has its own weights

- Layers
  - Neurons organized at the same level
  - The first layer is the data input (x)

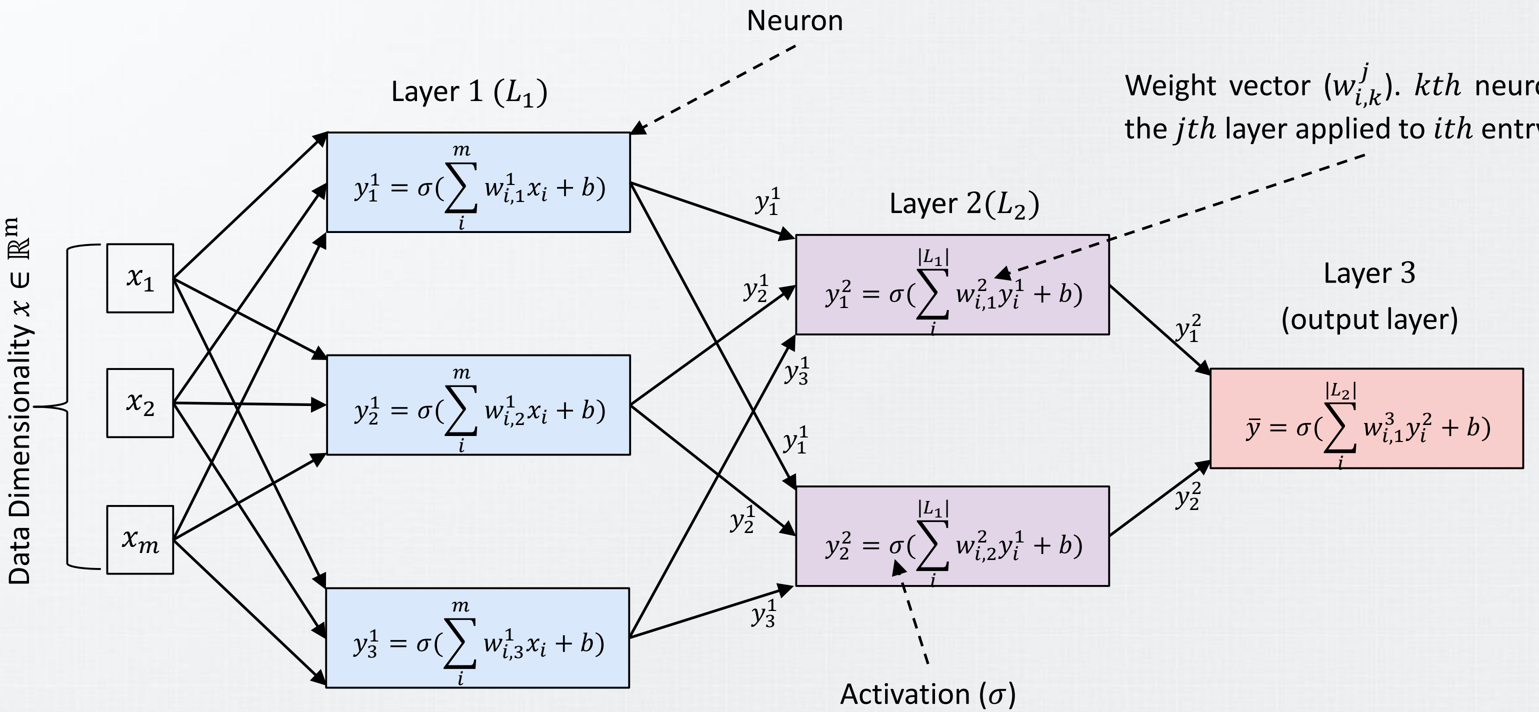
- Activations (σ)
  - Non-linear transformations



Components

FeedForward Networks

- In this architecture, we have 6 neurons (6 activations), 3 layers and $m \times 3 + 3 \times 2 + 2 \times 1$ weights (parameters)



Definitions

FeedForward Networks

- Let $\mathcal{F}(\cdot\,,\theta)$ be a neural network parametrized by a set of weights (parameters) θ

- Given an input x, the network predicts $\bar{y}$ according to its parameters θ
  - It means $\bar{y} = \mathcal{F}(x, \theta)$

- We can decompose $\mathcal{F}$ into a set of functions/transformations f – layers
  - $\mathcal{F}(x, \theta) = \bar{y} \Rightarrow f_L(\dots, f_2(f_1(x, \theta_1), \theta_2), \theta_L) = \bar{y}$
  - L defines the **depth** of the network

- Note that each f_i has its own set of parameters θ_i

Activations

FeedForward Networks

- An activation, $\sigma(\cdot)$, receives an input and applies a transformation to it
  - Such a transformation should be **non-linear**
- Most activations have no trainable parameters
- Despite simple, the activation plays an important role in the success of network
  - It injects nonlinearities into each $f(\cdot,\cdot)$; hence, into the full network

The Role of Activation

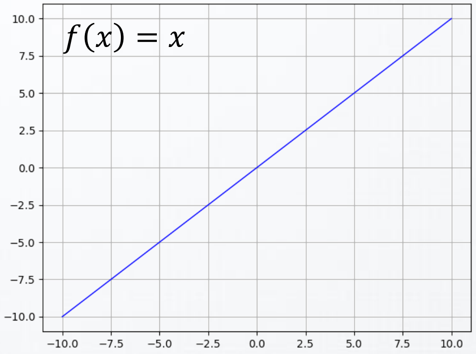
FeedForward Networks

- The composition of linear functions is indeed a linear function
- A linear function of a linear function is also a linear function
- Therefore, without nonlinearities, a neural network would be linear
    - No matter how many neurons/layers it has

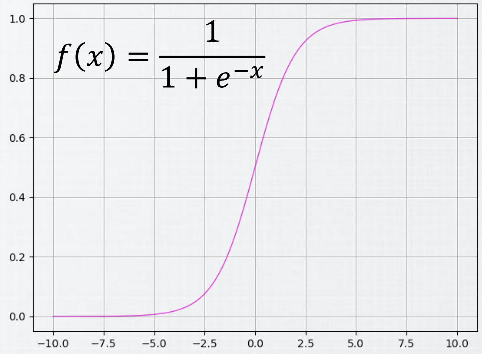
Popular Activations

FeedForward Networks

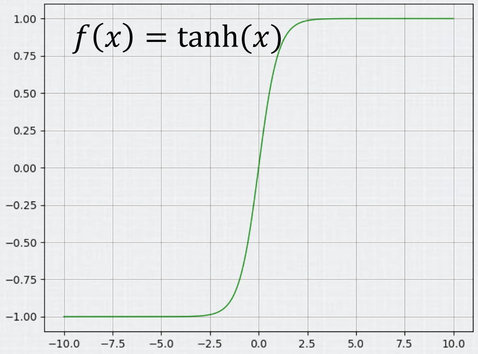
Linear



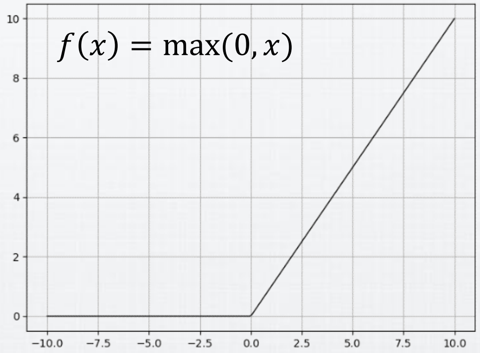
Sigmoid



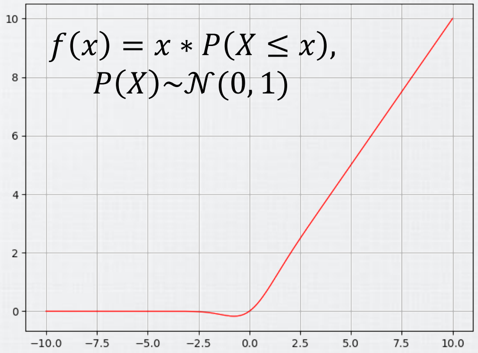
Hyperbolic Tangent



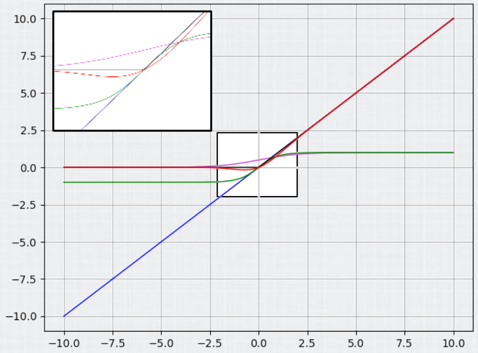
ReLU



GELU



Activation Zoo



Perceptron

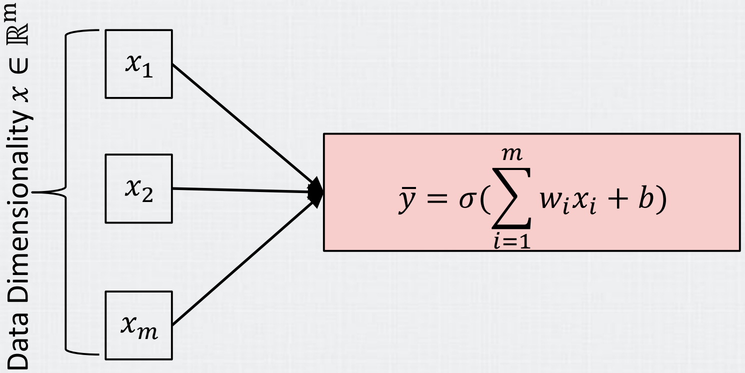
Introduction

Perceptron

- The first neural network
  - It occupies an important place in the history of pattern recognition algorithms

- Perceptron belongs to the family of linear models

- Perceptron components
  - Single layer and single neuron
  - Step Activation $\sigma = \begin{cases} 1, y \geq 0 \\ 0, otherwise \end{cases}$



The Perceptron Convergence Theorem

Perceptron

- The Perceptron Convergence Theorem states that if there exists an exact solution, then the perceptron is guaranteed to find an exact solution in a finite number of steps
    - Proof: Rosenblatt (1962); Block (1962); Minsky et al. (1969); Hornik et al. (1989); Barron et al. (1993)

- Exact solution means: if the training data set is **linearly separable**

Rosenblatt et al. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. 1962

Block et al. *Analysis of a four-layer series-coupled perceptron*. Reviews of Modern Physics, 1962

Minsky et al. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969

Hornik et al. *Multilayer feedforward networks are universal approximators.* Neural Networks, 1989

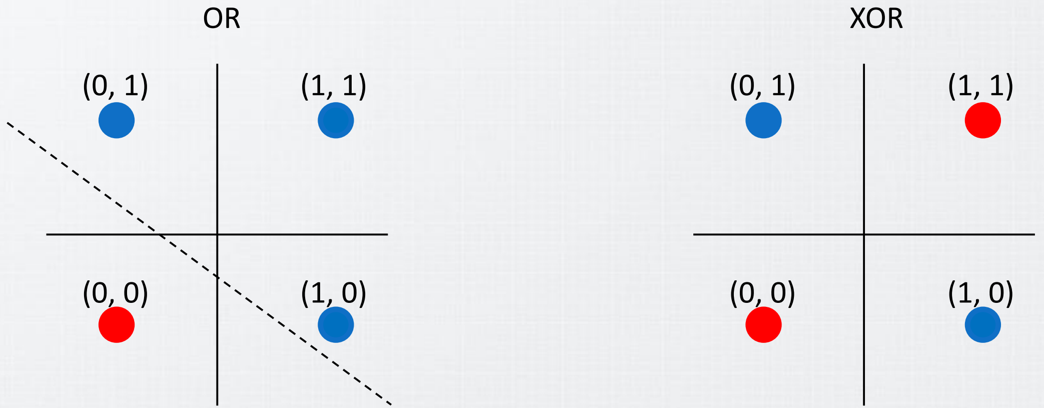
Barron et al. Universal approximation bounds for superpositions of a sigmoidal function. Transactions on Information Theory, *1993*

Limitations

Perceptron

- Perceptron is confined to solving linear problems
    - Remember that it belongs to the family of linear models

- The XOR problem
    - Formally: $\mathcal{F}([0,1],w)=1, \mathcal{F}([1,0],w)=1, \mathcal{F}([1,1],w)=0, \mathcal{F}([0,0],w)=0$
    - This was the first major dip in the popularity of neural networks



Final Note

Perceptron

- *On being asked, "How is Perceptron performing today?"* I am often tempted to respond, "Very well, thank you, and how are Neutron and Electron behaving?"

Frank Rosenblatt, inventor of the perceptron

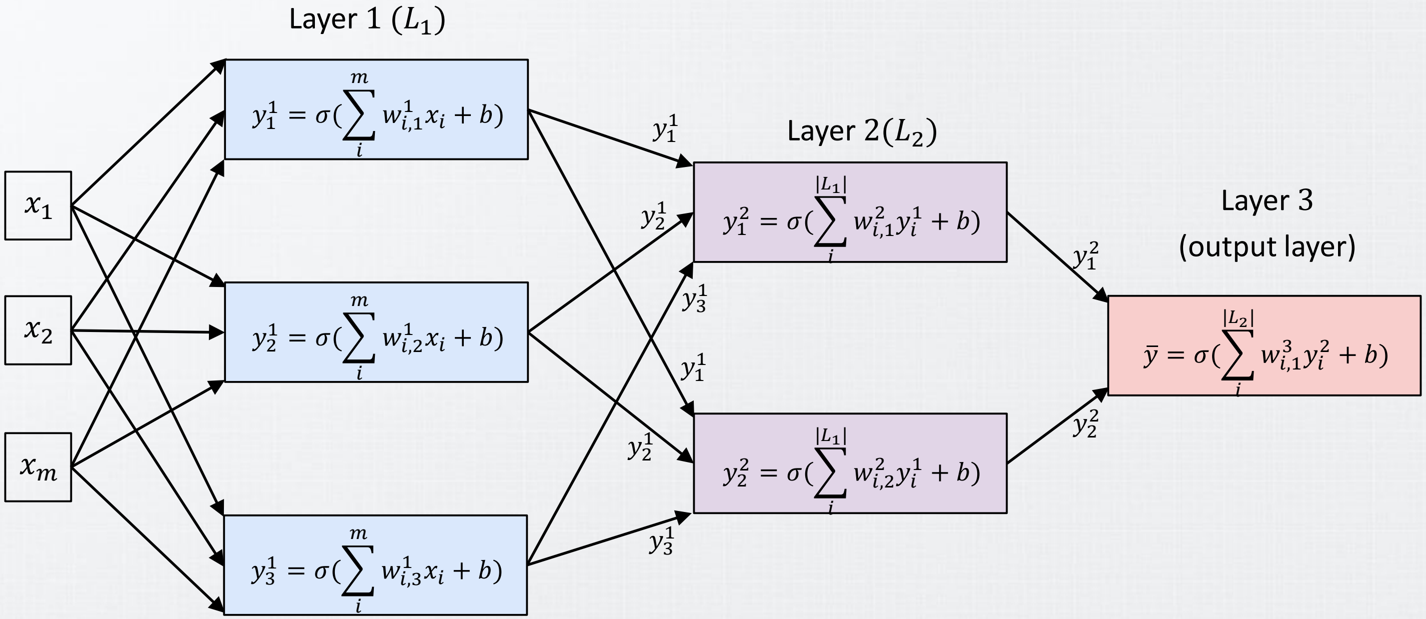
Multilayer Perceptron (MLP)

Introduction

Multilayer Perceptron

- Also known as Vanilla neural network and Feedforward network

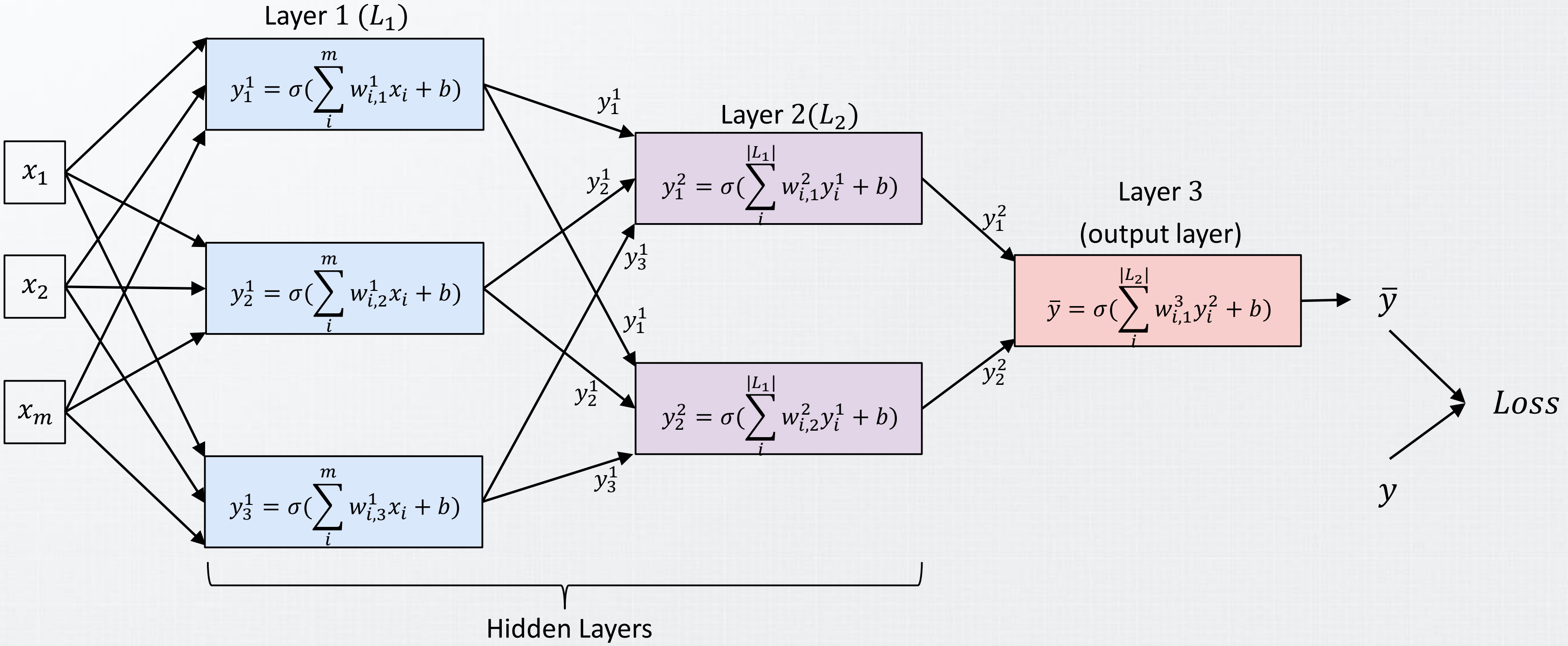
- Different from Perceptron, MLP has multiple layers and neurons (as its name suggests)



Hidden Layer

Multilayer Perceptron

- Layers for which the training data **does not show the desired output** (it is hidden)



The Universal Approximator Theorem

Perceptron

- Universal approximators
  - Models with the ability to approximate any continuous function

- An MLP with one hidden layer and a sufficient number of neurons is able to approximate **any function** [Hornik et al. (1989); Barron et al. (1993)]
  - Neural networks are therefore said to be universal approximators

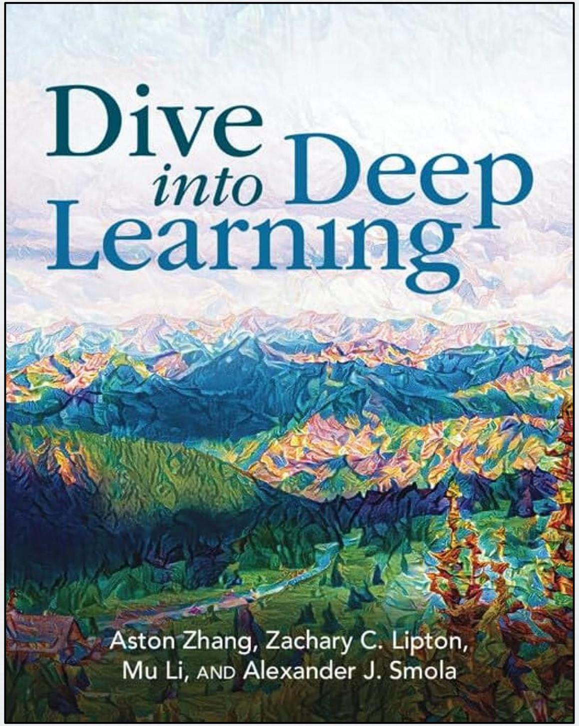
Hornik et al. *Multilayer feedforward networks are universal approximators.* Neural Networks, 1989

Barron et al. Universal approximation bounds for superpositions of a sigmoidal function. Transactions on Information Theory*, 1993*

Bibliography

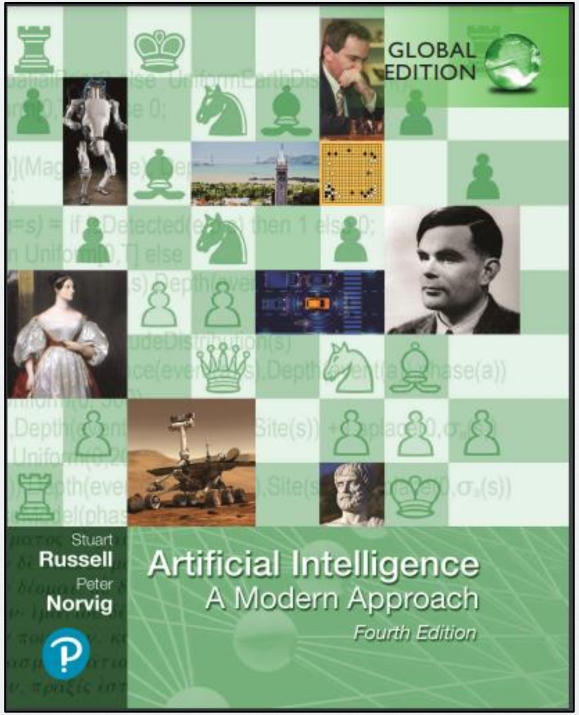
Bibliography

- Dive into Deep Learning
  - Chapter 5 – Multilayer Perceptrons
    - 5.1.1 – Hidden Layers
    - 5.1.2 – Activation Functions



Bibliography

- Artificial Intelligence A Modern Approach Fourth Edition
  - Chapter 22 – Deep Learning
    - 22.1 – Simple Feedforward Networks
      - 22.1.1 Networks as complex functions



Bibliography

- The Hundred-page Machine Learning Book
    - Chapter 6 – Neural Networks and Deep Learning
        - 6.1 Neural Networks