Neural Networks

Introduction

Dr. Petr Musilek

Department of Electrical and Computer Engineering
University of Alberta

Fall 2019

Neural Networks

- Information processing systems inspired both by
  - biological nervous systems, and
  - mathematical theories of learning.
- Massively parallel interconnected networks of simple (usually adaptive) elements and their hierarchical organizations which are intended to interact with objects of the real world in the same way as biological nervous systems do

[Kohonen, 1988]

The Brain

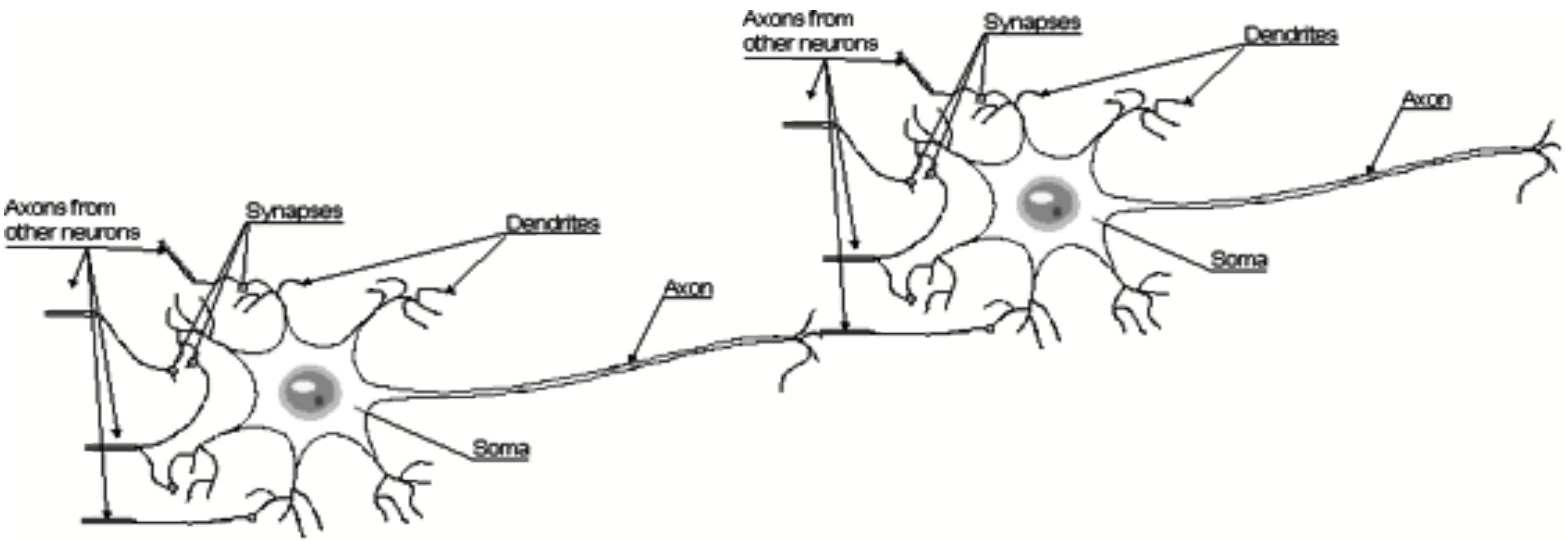
- The brain is a highly complex, non-linear, and parallel computer, composed of some 10^{11} neurons that are densely connected ($\sim 10^4$ connections per neuron)
- A neuron is much slower (10^{-3}sec) compared to a silicon logic gate (10^{-9}sec), however the massive interconnection between neurons make up for the comparably slow rate.

Information Processing in Brain

Hundred Steps rule: Complex perceptual decisions are arrived at quickly (within a few hundred ms)
Individual neurons operate slowly (10^{-3}sec), these calculations do not involve more than about 100 serial steps and the information sent from one neuron to another is very small (a few bits)

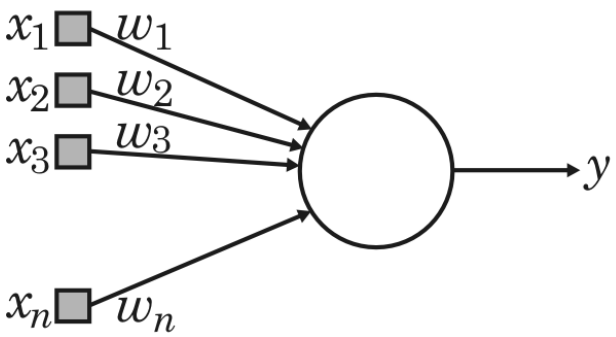
Neural Plasticity: Some of the neural structure of the brain is present at birth, while other parts are developed through learning, especially in early stages of life, to adapt to the environment (new inputs).

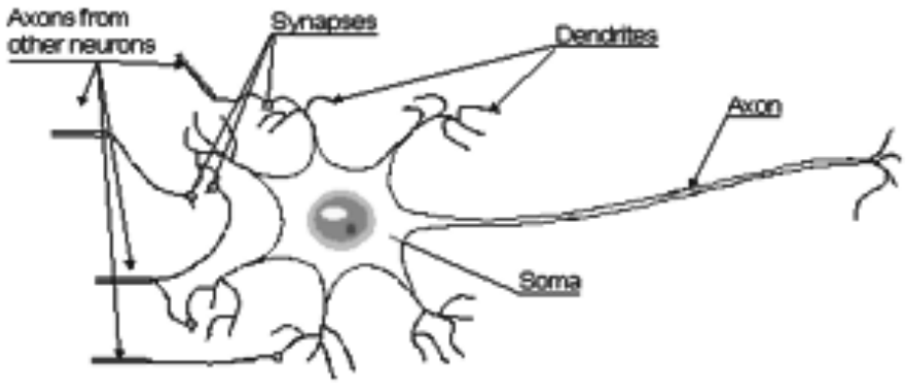
The Biological Neuron



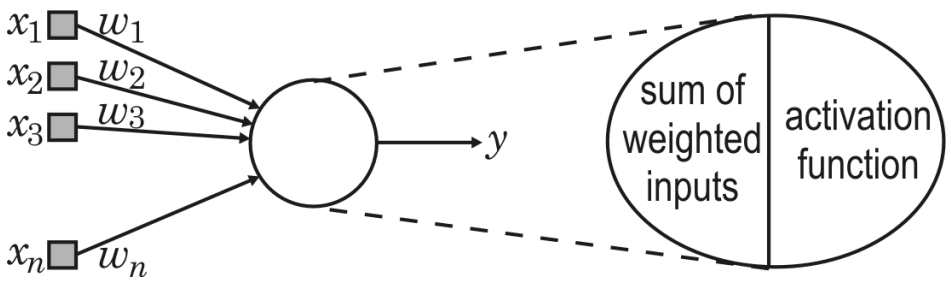
The Artificial Neuron

Model with components analogous to those of biological neurons





The Artificial Neuron



Sum of weighted inputs

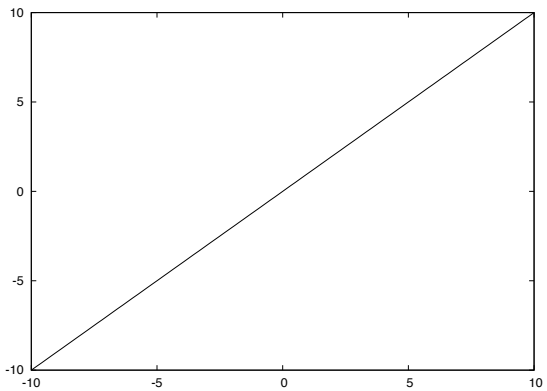
$$tot = \sum_{i=1}^{n} w_i x_i$$

Activation Function

$$o = f(tot)$$

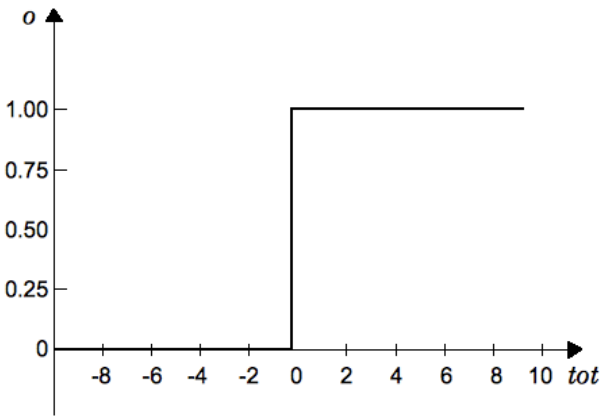
Note: there are other alternatives too.

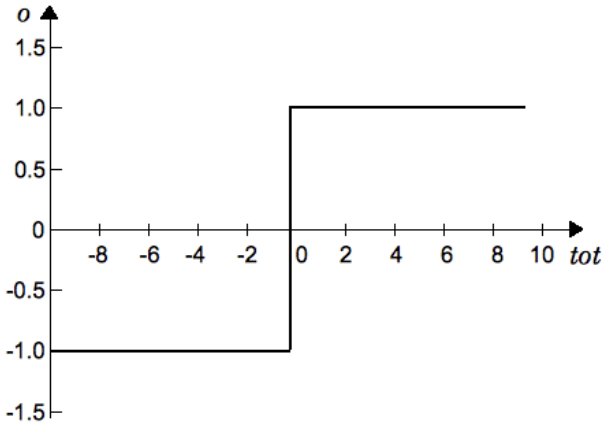
Activation Function Linear



$$o = f_{lin}(tot) = tot$$

Activation Function Hard Limiting

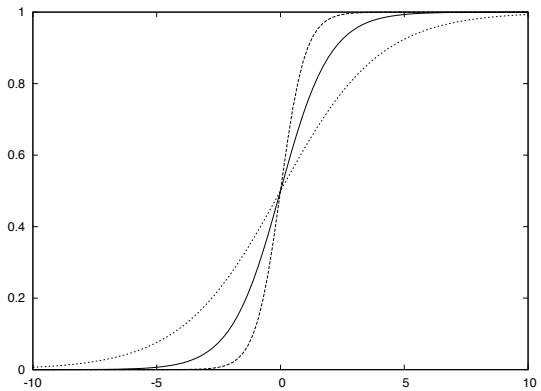


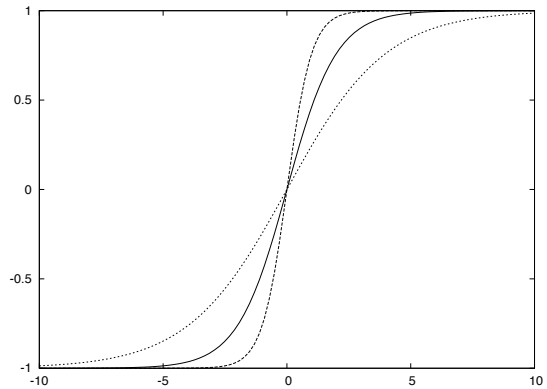


$$o = f_{hlu}(tot) = \begin{cases} 0 & \text{if } tot \leq 0, \\ 1 & \text{if } tot > 0. \end{cases}$$

$$o = f_{hlb}(tot) = \begin{cases} -1 & \text{if } tot \leq 0, \\ 1 & \text{if } tot > 0. \end{cases}$$

Activation Function Sigmoid





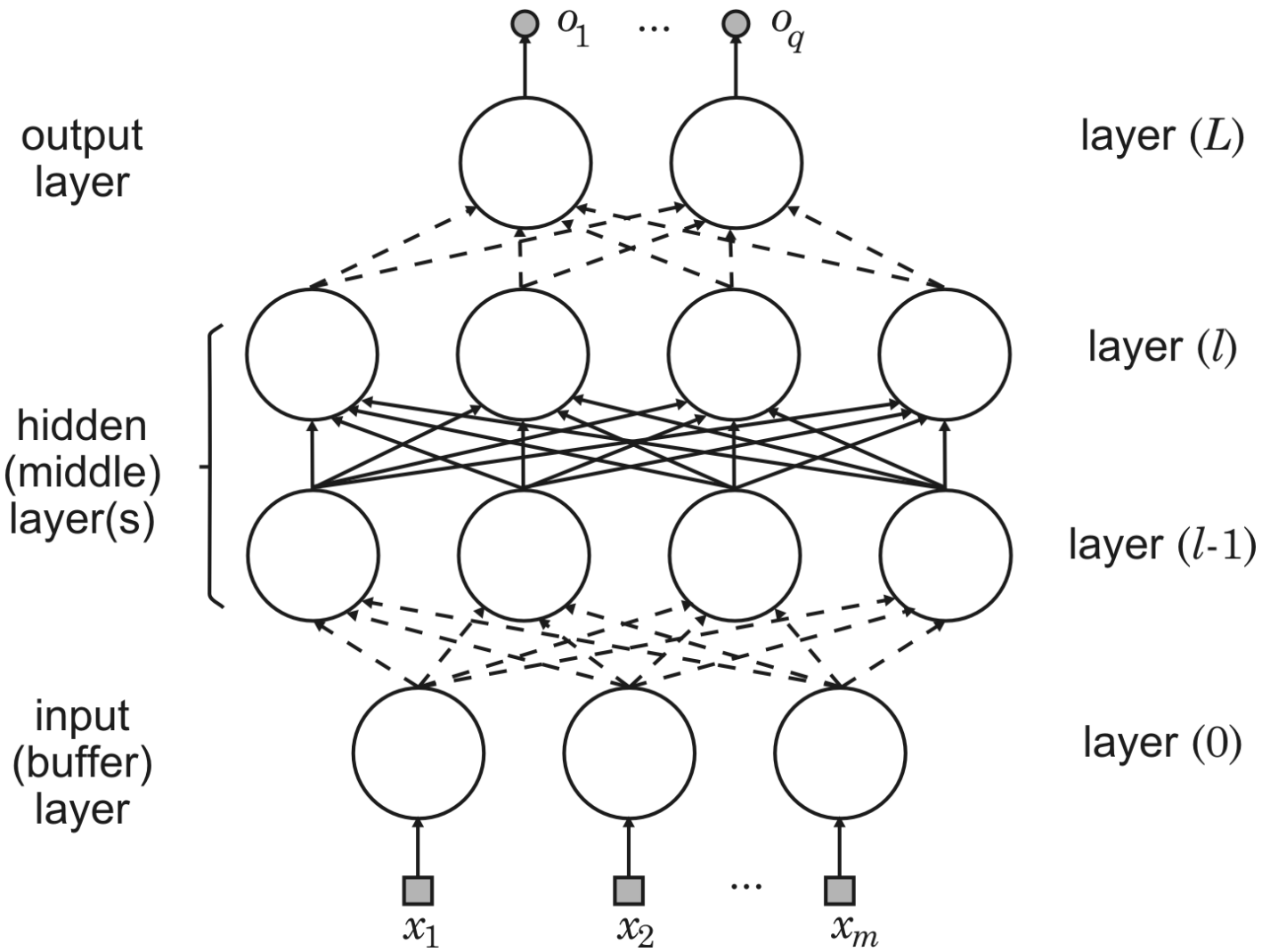
$$o = f_{sigu}(tot) = \frac{1}{1 + e^{-\alpha tot}}$$

$$o = f_{sigb}(tot) = \frac{1 - e^{-\alpha tot}}{1 + e^{-alphatot}}$$

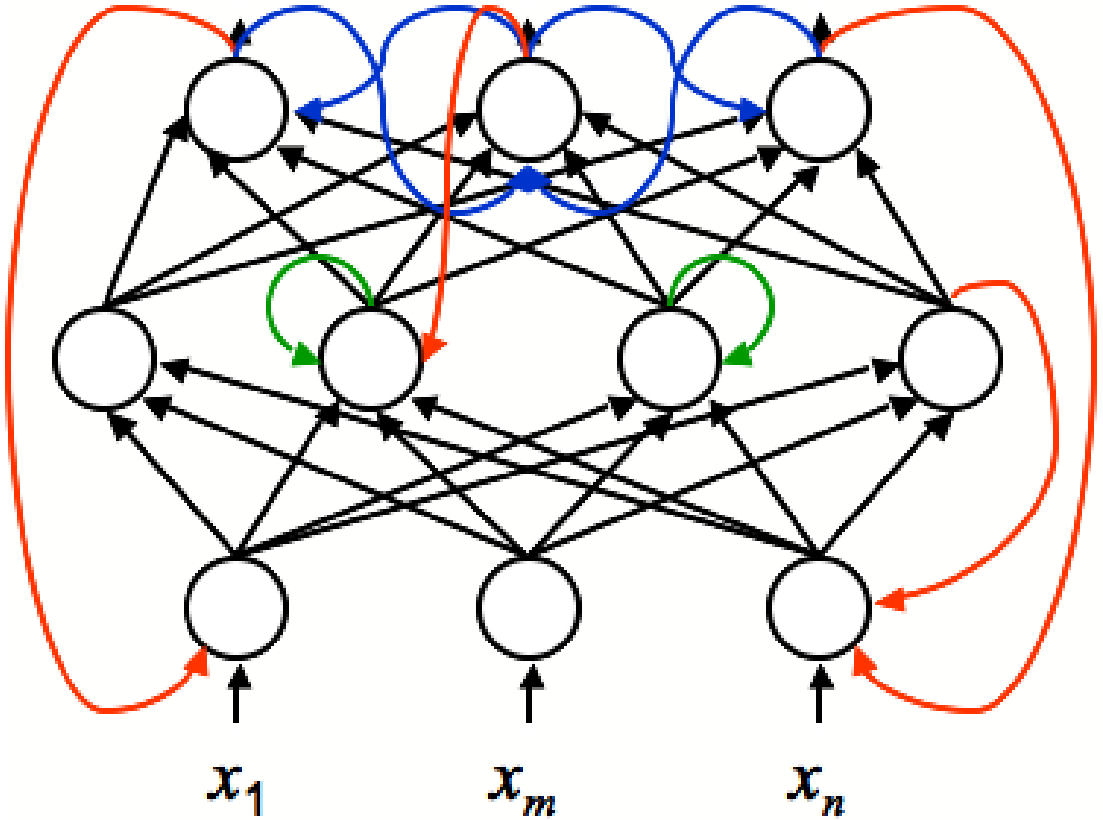
Artificial Neural Networks (ANN)

- An information processing system consisting of a large number of simple, highly interconnected processing elements (neurons) in an architecture inspired by the structure of the cerebral cortex of the brain.
- Usually organized into a sequence of layers with full connections between the layers.

Example architecture of ANN



Interconnection Variations



feedforward local feedback global feedback lateral

Impact of recurrent connections

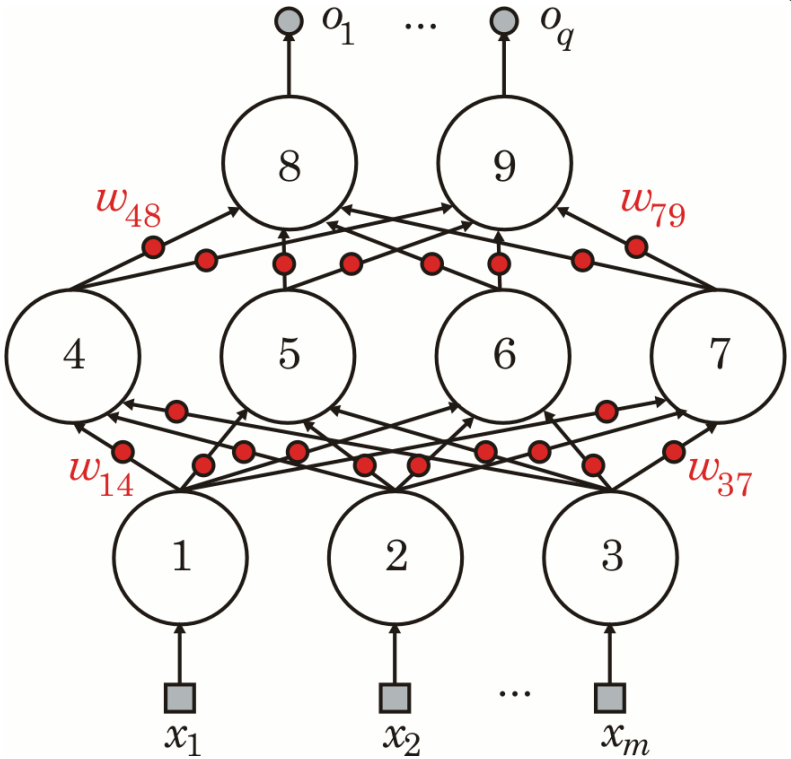
Fedforward	Recurrent
direct calculation	state-machine like
fixed calculation time	require “setting time”
direct I/O mapping	have memory
open loop	closed loop

Note

Feedforward NN with linear activation function can be described using *linear algebra*.

Weights

Each connection between neurons has an adjustable weight.

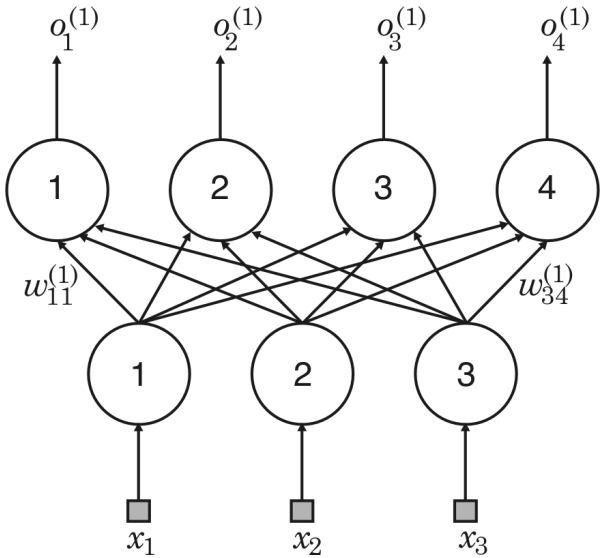


Notation

This notation is not very practical, let's develop a better one.

Notation (1/2)

$$\mathbf{o}^{(1)} = [o_1^{(1)}, o_2^{(1)}, o_3^{(1)}, o_4^{(1)}]$$

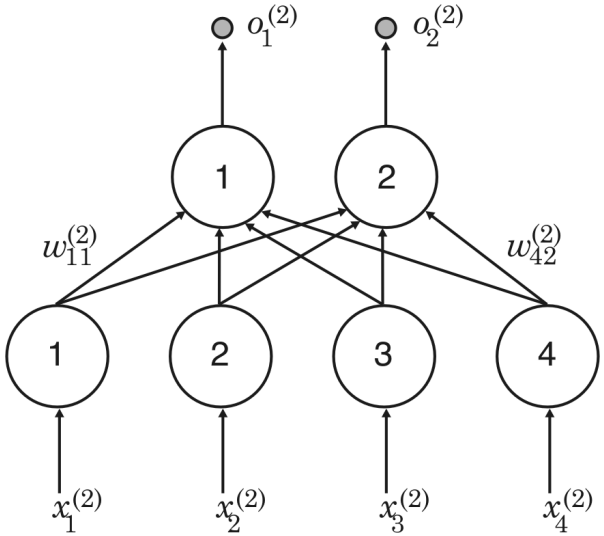


$$\mathbf{o}^{(1)} = \mathbf{W}^{(1)} \cdot \mathbf{x}^{\mathrm{T}} = \\ = \begin{bmatrix} w_{11}^{(1)} & w_{21}^{(1)} & w_{31}^{(1)} \\ w_{12}^{(1)} & w_{22}^{(1)} & w_{32}^{(1)} \\ w_{13}^{(1)} & w_{23}^{(1)} & w_{23}^{(1)} \\ w_{14}^{(1)} & w_{24}^{(1)} & w_{34}^{(1)} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\mathbf{x}^{(0)} = \mathbf{x} = [x_1, x_2, x_3]$$

Notation (2/2)

$$\mathbf{o} = \mathbf{o}^{(2)} = [o_1^{(2)}, o_2^{(2)}]$$



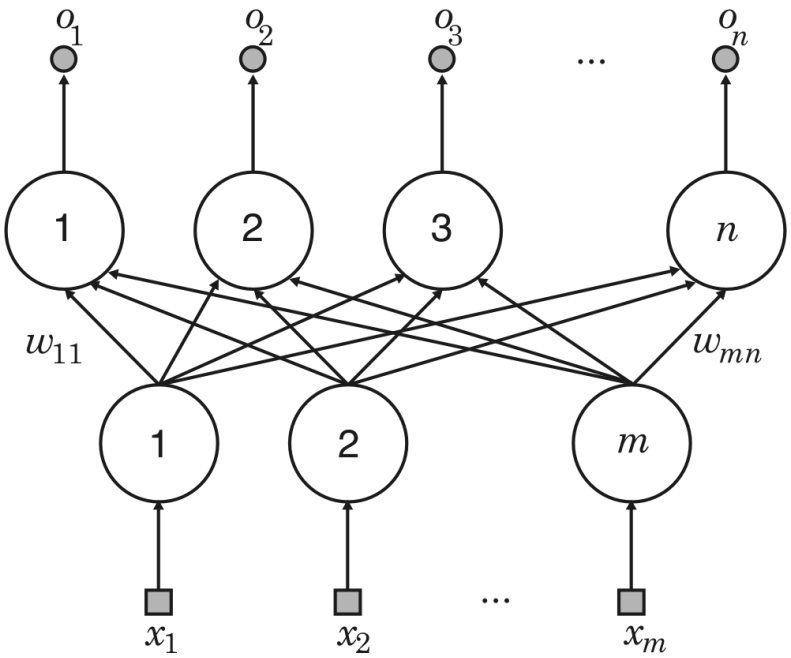
$$\mathbf{o}^{(2)} = \mathbf{W}^{(2)} \cdot \mathbf{x}^{(2)\mathrm{T}} =$$

$$= \begin{bmatrix} w_{11}^{(2)} & w_{21}^{(2)} & w_{31}^{(2)} & w_{41}^{(2)} \\ w_{12}^{(2)} & w_{22}^{(2)} & w_{32}^{(2)} & w_{42}^{(2)} \end{bmatrix} \cdot \begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \\ x_3^{(2)} \\ x_4^{(2)} \end{bmatrix}$$

$$\mathbf{x}^{(2)} = [x_1^{(2)}, x_2^{(2)}, x_3^{(2)}, x_4^{(2)}] =$$

$$= [o_1^{(1)}, o_2^{(1)}, o_3^{(1)}, o_4^{(1)}]$$

Example: Linear Associator



- Very simple ANN
- Capable to store more than one (linear) relationships simultaneously
- Not very accurate

History: 1940s and 1950s

1940 Similarity of the physiology of the brain and information processing

1943 First formal model of an elementary computing neuron, based on threshold logic (McCulloch and Pitts)

1949 Learning hypothesis: information is stored in connections; learning scheme for updating connections (Hebb)

1954 First computational machines used to simulate a Hebbian network (Farley and Clark)

1958 Perceptron - a trainable machine for pattern recognition (Rosenblatt)

1959 Biological model of simple and complex cells in the primary visual cortex (Nobel laureates Hubel and Wiesel)

1960 ADALINE - device for adaptive control and pattern recognition, trained by the least-mean-square (LMS) learning rule (Widrow)

History: 1960s and 1980s

1965	First functional networks with many layers (Ivakhnenko and Lapa, Group Method of Data Handling)
1965+	Weak learning theory and modest computational resources led to stagnation
1969	Minsky and Papert pointed out limitations of neural networks and questioned their usefulness. Only a handful of researchers (including those in Europe and Japan) continued NN research afterwards.
1975	Cognitron and neocognitron for pattern recognition (Fukushima)
1977	Mathematical theory of NNs (Amari)
1982	Unsupervised learning neural networks for feature mapping (Kohonen)
1982	Neural theories inspired by developmental physiology (Grossberg)
1985	Recurrent neural networks for information storage and retrieval and optimization (Hopfield)

History: 1980s and 1990s - turning point

1975 Backpropagation algorithm for training multi-layer feedforward percetrons was first developed, but received little attention (Werbos)

1986 The powerful gradient descend based backpropagation training algorithm received wide attention - parallel distributed processing, connectionism (Rumelhart and McClelland)

1991 The vanishing gradient (VG) problem affects many-layered feedforward networks that used backpropagation and also recurrent neural networks (RNNs)

1992 Multi-level hierarchy of pre-trained and then fine-tuned networks to overcome the VG proble (Schmidhuber); other aleternatives relied only on the gradient sign (Rprop, 2003)

History: 2000s

- Learning high-level representations using successive layers of binary or real-valued latent variables (Hinton)
- Earlier challenges in training deep neural networks were successfully addressed with methods such as unsupervised pre-training, while available computing power increased through the use of GPUs and distributed computing.
- Neural networks were deployed on a large scale, particularly in image and visual recognition problems. This became known as “deep learning”.

History: 2010s

- Networks learn to recognize higher-level concepts (such as cats) only from watching unlabeled images taken from YouTube (Ng and Dean).
- The state of the art in deep learning feedforward networks alternated between convolutional layers and max-pooling layers, topped by several fully or sparsely connected layers followed by a final classification layer.
- Supervised deep learning methods achieve human-competitive performance on certain tasks.