Artificial Neural Networks
Lecture Notes

Stephen Lucci, PhD

Lecture 1

About this file:

- If you have trouble reading the contents of this file, or in case of transcription errors, email gi0062@bcmail.brooklyn.cuny.edu
- Acknowledgments:
  Background image is from http://www.anatomy.usyd.edu.au/online/neuroanatomy/tutorial1/tutorial1.html (edited) at the University of Sydney Neuroanatomy web page. Mathematics symbols images are from metamath.org's GIF images for Math Symbols web page. Other image credits are given where noted, the remainder are native to this file.

Contents

1. Overview
2. Computational Models
3. Artifical Neural Networks
   - McCulloh-Pitts Networks

Brief Overview

- **Historical Note** McCullloch and Pitts (1943) - developed the first model of artificial neurons.
- When we study artificial neural networks, we need to clarify their relationship to the biological paradigm.

  "Artifical neural networks are an attempt at modeling the information processing capabilities of the nervous system" - *from the textbook*.

- Animal nervous systems are composed of thousands or millions of interconnected cells. In the case of humans, it is billions of cells.
- Neurons are slow when compared to electronic logic gates.
  Neuronal delays are on the order of milli-seconds, vs. fractions of a nanosecond in electronic gates.
- Biologists and neurologists understand the mechanisms whereby individual neurons communicate with one another.
  However, the mechanism whereby massively parallel and hierarchical collections of neurons form functional units eludes us.
- We study the information processing capabilities of complex arrangements of simple computing units (i.e., artificial neurons.)
- **The networks will be adaptive** - Adjustment of parameters is done through a learning algorithm (i.e., there will be no explicit programming.)

Models of Computation

Artificial neural networks can be considered as just another approach to the problem of computation.

Formal Definitions of Computability (1930's & 1940's)
The following lists 5 classic approaches to the study of computability.

- The Mathematical Model
- The Logic Operational Model (Turing Machines)
- Cellular Automata
- The Computer Model
- Biological Model (Neural Networks)

I. The Mathematical Model

- **Notable Contributors:** Hilbert, Ackerman, Church and Kleene.
- **Set of Primitive Functions**

Zero-Function	$Z(x)=0 \quad \forall x \in \mathbb{N}$
Successor Function	$S(x)=x+1$ Example S(1) = 2 S(2) = 3, &c.
Projection Function	$U^i_n(x_1, x_2, x_3, \ldots x_i, \ldots x_n) = x_i$ Example $U^2_3(x_1, x_2, x_3) = x_2$.
Set of Operations	Composition and Recursion u - minimization operator.

- **A simple Example: Showing that Addition is Recursive**
  Let us represent the addition of two numbers, m and n, by the function f(m,n). Thus, we have

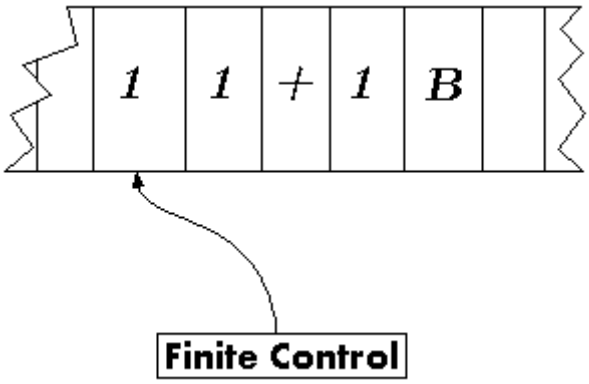
f(x,0) = x	(i.e., x+0 = x - Identity axiom for addition)
f(x,y+1) = S(F(x,y);	(i.e., x + (y+1) = succesor function of (x+y)

- For example, to add 3+2, we have

f(3,2) = S(f(3,1))
= S(S(f(3,0)))
= S(S(3))
= S(4)
= 5.

II. The Logical/Operational Model: The Turing Machine

- **Notable Contributors:** Alan Turing 1936, inventor of the Turing Machine; Church.



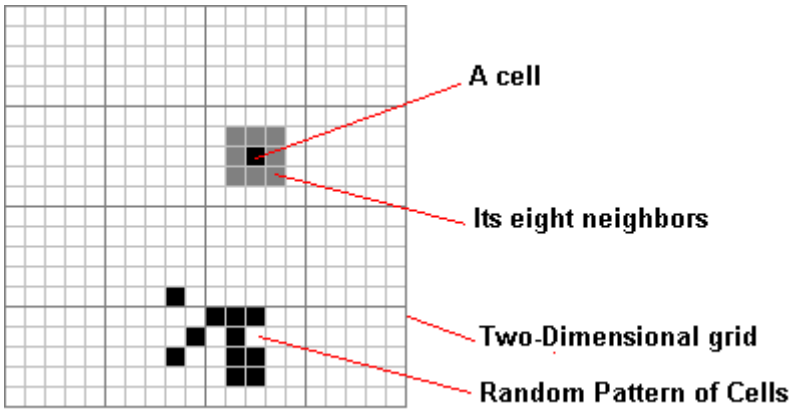
A Turing Machine "tape" illustrating the unary addition 2+1

- The Turing machine consists of a tape serving as the internal memory of the machine, of unlimited size, and a read/write head which moves along the tape.
- The Turing machine is described by the state, output and direction functions.
  We can write (state, input) $\longrightarrow$(state, write, {L, R, N})
  where L,R,N mean Left, Right and No movement, respectively.
- In the above figure, we can describe the process of addition in 5 steps. The first step looks like this
  1. (q_0, 1, q_0, 1, R)
  where the the first q_0 is the current state, followed by the current symbol scanned, followed by the next state, and in the end, the direction to move (in this case Rightward). Write the remaining 4 steps.
- The operation of this addition may be described or traced as follows:
  (q_0,11+1)$\longrightarrow$
  1. (1q_0,1+1)
  1. (11q_0,+1)
  2. (111q_1,1)
  3. (111q_1,B)
  4. (111$q_2$1)
  5. (111q_2B)
- **Unsolvability:** Some problems are undecidable. They are not solvable by the Turing Machine and therefore and not solvable by any other computing machine.

  An example of undecideable problems is the **Halting Problem**.

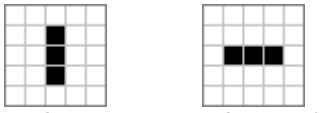
III. The Cellular Automata Model

- **Notable Contributors:** Von Neumann; Conway; Wolfram (1D Cellular automata)



A Two-Dimensional Cellular Automaton

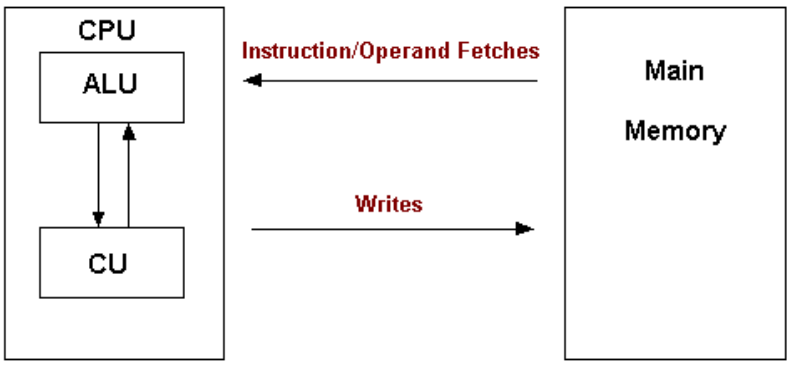
- **The Game of life** The game of Life is a well-known example of cellular automata, developed by Conway.
  It is comprised of a two-dimensional grid of cells. Each cell can be in one of two states, on or off, or alive or dead.
  Cells may transition from one state to the other, and become dead or alive based on **a set of rules.**.
- **Rules of the Game of Life**
  Let N be the number of neighbors of a given cell.
  If N = 0 or 1, cell dies
  If N = 2, the cell maintains its current state (status quo)
  If N = 3, the cell becomes alive
  If N = 4,5,6,7 or 8, the cell dies.
- An Example



- Wolfram studied one-dimensional CA.
  He enumerated 256 possible rules and 4 complexity classes for such automata.

IV. The Computer Model (Z1, ENIAC, Mark I)

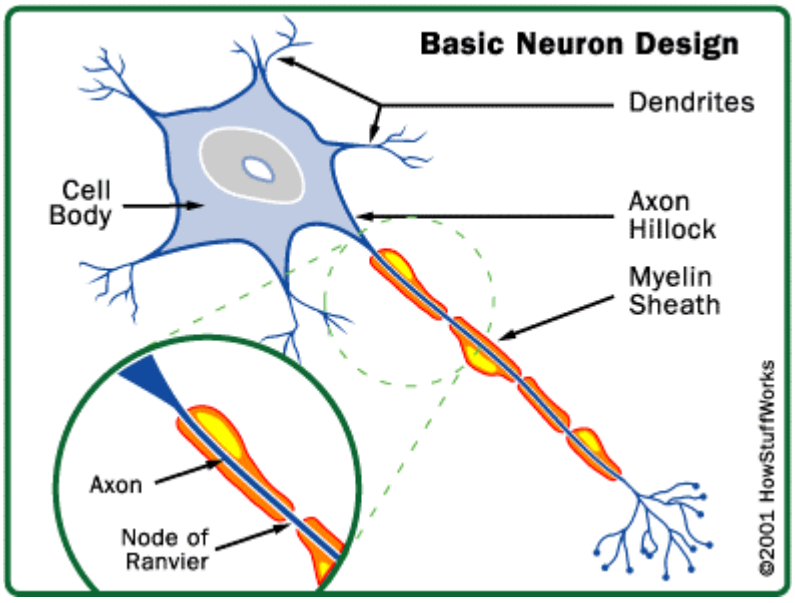
- **Notable Contributors:** Von Neumann; Post; &c



A Von Neumann Machine Schematic

V. The Biological Model (Neural Networks)

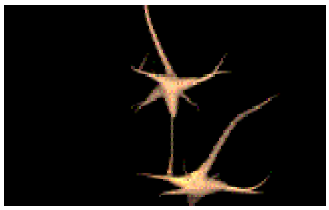
- **Notable Contributors:** McCulloh; Pitts; Wiener; Minsky; et al.
- Neural networks as a computing model possess the following **properties:**
  - **Operate in parallel**
    Computing elements (neurons) work in parallel as opposed to the sequential action of Turing machines.
  - **Hierarchical Multilayered Structure**
    Information is transmitted not only to immediate neighbors, but also to more distant units, as opposed to CA.
  - **No Program is handed over to the hardware**
    The free parameters of the network have to be found adaptively - as opposed to conventional computers (which require pre-programmed algorithms to execute).
- **Structure of the Neuron**



Schematic of a Brain Neuron

(Image from How Stuff Works ©2002. Full article may be found at "How Your Brain Works" by Craig C. Freudenrich, Ph.D.)

· The cell body (soma)	"Processing" occurs here.
· Dendrites	Protrude from soma. They conduct signals to the cell body.
· Axon Hillock	Extends from cell body - initial portion of the axon.
· Axon	A long fibre - generally splits into smaller branches.
· Synapse	The axon-dendrite (axon-soma; axon-axon) contact between an endbulb and the cell it impinges upon is called a synapse. (End bulbs can be seen at the bottom right corner of the above image.)



Animated close-up of a neuron and the synaptic cleft
(Image from "Neurons: Our Internal Galaxy" by Silvia Helena Cardoso, PhD
http://www.epub.org.br/cm/n07/fundamentos/neuron/rosto_i.htm)

- **The signal flow in the neuron** is from the dendrites through the soma converging at the axon hillock and down the axon to the endbulbs.
- **A neuron typically has many dendrites** but only a single axon.
- The four elements
  - dendrites
  - synapses
  - cell body
  - axon

  are the **minimal structure** will will adopt from the biological model.

- Artificial neurons will have
  - input channels
  - cell body
  - output channel
  - synapses

  where the synapses will be simulated by contact points between the cell body and input or output connection.

- A **weight** will be associated with these points.

- **Organizational and Computational Principles of the Brain**

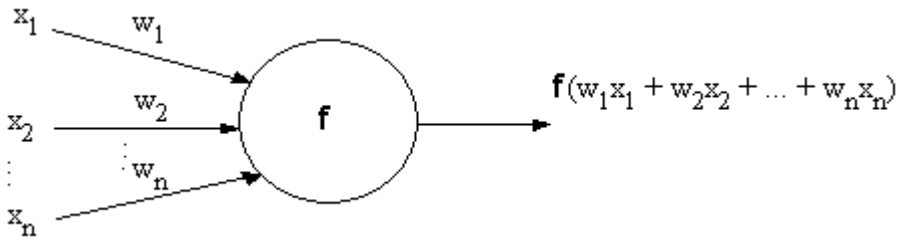
  Massive parallelism
  A large number of simple, slow units are organized to solve problems independently but collectively.

  High degree of connection complexity
  Neurons have a large number of connections to other neurons and have complex interconnection patterns.

- Trainability (Inter-neuron interaction parameters)
  Connection patterns and connection strengths are changeable as a result of accumulated sensory experience.
- Binary states and continuous variables
  Each neuron has only two states: resting and depolarization.
  However, the variables of the brain are continuous (potentials, synaptic areas, ion and chemical density, &c. ...) and vary continuously in time and space.
- There are many types of neurons and signals.
- Intricate signal interaction
  The interaction of impulses received at a single neuron is highly nonlinear and depends on many factors.
- Physical decomposition (Nature's way of dealing with the complexities of the brain itself)
  The brain is organized as a mosaic of subnetworks.
  Each subnetwork consists of several thousand densely connected neurons.
  These subnetworks are the basic processing modules of the brain.
  Connections to distant neurons are sparser and with less feedback...
  Autonomous local collective processing in parallel, followed by a more serial and integrative processing of those local collective outcomes.
- Functional decomposition
  Each area, or subnetwork is responsible for specific functions.

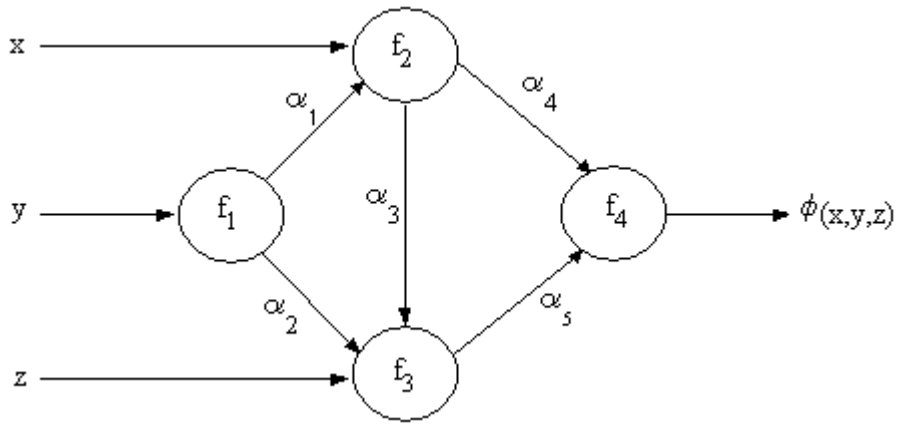
Artificial Neural Networks

- We may think of artificial neural networks as networks of primitive functions:



Primitive function f computed in the body of the abstract neuron

- Usually, the input channels have an associated weight
- Different models of ANN's (Artificial Neural Networks) will differ in:
  - Primitive function used
  - Interconnection pattern (topology)
  - Timing of transmission

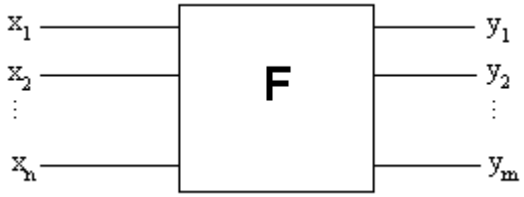


Function model of ANN

- Function ϕ evaluated at the point (x,y,z).
- The nodes implement the primitive functions, f_1, f_2, f_3, f_4, which are combined to form ϕ.
- The function ϕ - is a network function.
- Different weights α_i... will produce different functions.
- **The key Elements**
  - Structure of the nodes
  - Topology of the network
  - Learning algorithm to find the weights

Threshold Logic

- F: $R^n \longrightarrow R^m$
- A neural net as a black box:

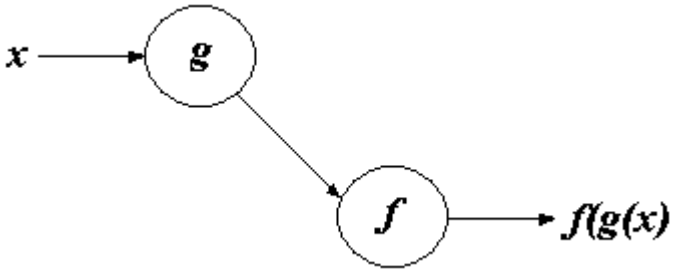


A neural net as a black box

- certain inputs should produce specific outputs.
- How is this done? Through a self-organizing process.

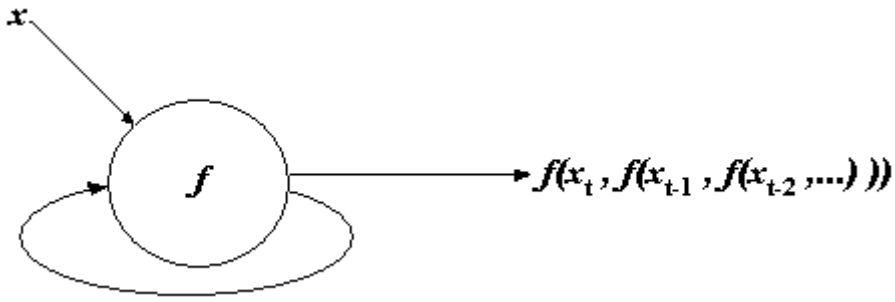
Function Composition Vs. Recursion

If no loops, then synchronization is not a problem - Assume no delay.



Function Composition (see fig 2.2 in textbook)

When loops are present, synchronization is required(Δt per step)
$f(x_0) = x_1$
$f(x_1) = x_2 = f(f(x_0))$
$f(x_2) = x_3 = f(f(x_1)) = f(f(f(x_0)))$
...



Recursion

- **Evaluating functions of n arguments** (see figure 2.4 in the textbook, Rojas, p31)
  - f is a function of n arguments, which thus has **unlimited fan-in**.
- **Simplifying matters** (Figure 2.5, Rojas, p.35):
  - Here, g is an integration function which reduces the inputs x_1, x_2, ..., x_n to a single argument.
  - f is the **output or activation function** - it produces the output of this node.

McCulloch-Pitts Networks

- See figure 2.6, Rojas p.32.
- Binary Signals - input/output
- Directed, unweighted edges of either an **excitatory** or an **inhibitory** type. (Inhibitory edges are marked with a small circle at the end of the edge that is incident on the unit.)

- Threshold value is θ
- inputs $x_1, x_2, ..., x_n$ come in through n **excitatory** edges.
- Suppose there are also inputs $y_1, y_2, ..., y_m$ coming in through m **inhibitory** edges.

If m >= 1, and at least one of the signals $y_1, y_2, ..., y_m$ is 1, the unit is **inhibited** and thus output = 0.

Otherwise, the total excitation

$$\overline{X} = x_1 + x_2 + ... + x_n$$

is computed and compared with the threshold θ.

If

$\overline{X}$ >= θ, the unit fires a 1.

- The **Activation Function** for McCulloh-Pitts units, is a *step function*.
See figure 2.7 p.33 in Rojas, for the graph of the activation function.

- **McCulloh-Pitts Units As Boolean Functions**

  - see figures 2.8 and 2.9 in Rojas, pp.33-34, illustrating an AND and an OR gate, the tables for which are shown here:

x_1	x_2	$F(x_1,x_2)$
0	0	0
0	1	0
1	0	0
1	1	1

x_1 AND x_2

x_1	x_2	x * w	$F(x_1,x_2)$
0	0	0	1
0	1	1	1
1	0	1	1
1	1	2	1

x_1 OR x_2

  - What would this picture look like if n=3 in each case?
  - **Monotonic Logic Functions**
  A monotonic logic function f of n arguments is one whose value at two given n-dimensional points X bar = $(x_1, x_2, ..., x_n)$ and Y bar = $(y_1, y_2, ..., y_n)$ is such that f(x) >= f(y), whenever the number of ones in the input y is a subset of the number of ones in the input x.
  - An example of a non-monotonic logic function of one argument is ...(?)
  - **Proposition 2.2.1** (p.34)
  Uninhibited threshold logic elements of the McCulloh-Pitts type can only implement monotonic logic functions.
  - Implementation of some non-monotonic logic functions: (see figure 2.10, p.34)

x_1	x_2	Σx_i	NOR
0	0	0	1
0	1	-	1
1	0	-	0
1	1	-	0

x_1	x_2	Σx_i	NAND
0	0		1
0	1		1
1	0		1
1	1		0

- **Geometric Interpretation** (see figure 2.12, p.35)

  A McCulloh-Pitts unit divides the input space into two half-spaces.

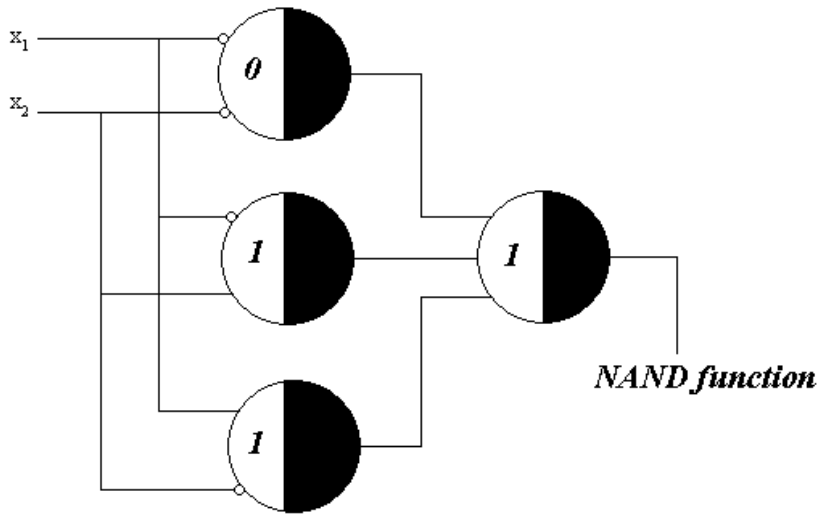
  For a given input (x_1, x_2, x_3), and threshold , $x_1 + x_2 + x_3$ is tested *true* for all points to one side of the plane with equatoin $x_1 + x_2 + x_3 =$.

- **Decoder** for the vector (1,0,1) (see figure 2.14)

  Only this input will cause the neuron to fire.

- **Constructing a circuit for the NAND function**

Input vectors	F
(0,0)	1
(0,1)	1
(1,0)	1
(1,1)	0



- **Proposition 2.2.2**

  Any logic function F:$\{0,1\}^n \longrightarrow \{0,1\}$ can be computed with a McCulloh-Pitts network of two layers.

- McCulloh-Pitts networks do *not* use weighted edges.

- **Weighted Edges vs. more Complex Topology**

  Compare figures 2.17 (a unit with weighted edges) and 2.18 (an equivalent unit with a more complex topology), p.39.
  Fig 2.17: $0.2x_1 + 0.4x_2 + 0.3x_3 >= 0.7$
  Fig 2.18: $2x_1 + 4x_2 + 3x_3 >= 7$.

- Networks with unweighted edges are equivalent to networks with weighted edges.

- **McCulloh-Pitts Networks with Weighted Edges**

  An example of a network for XOR:

x_1	x_2	z_1	z_2	**XOR**
0	0	0	0	0
0	1	0	1	1
1	0	1	0	1
1	1	1	1	0