UFV
Universidade Federal de Viçosa

Lecture 16 – Artificial neural networks

Prof. João Fernando Mari

joaofmari.github.io

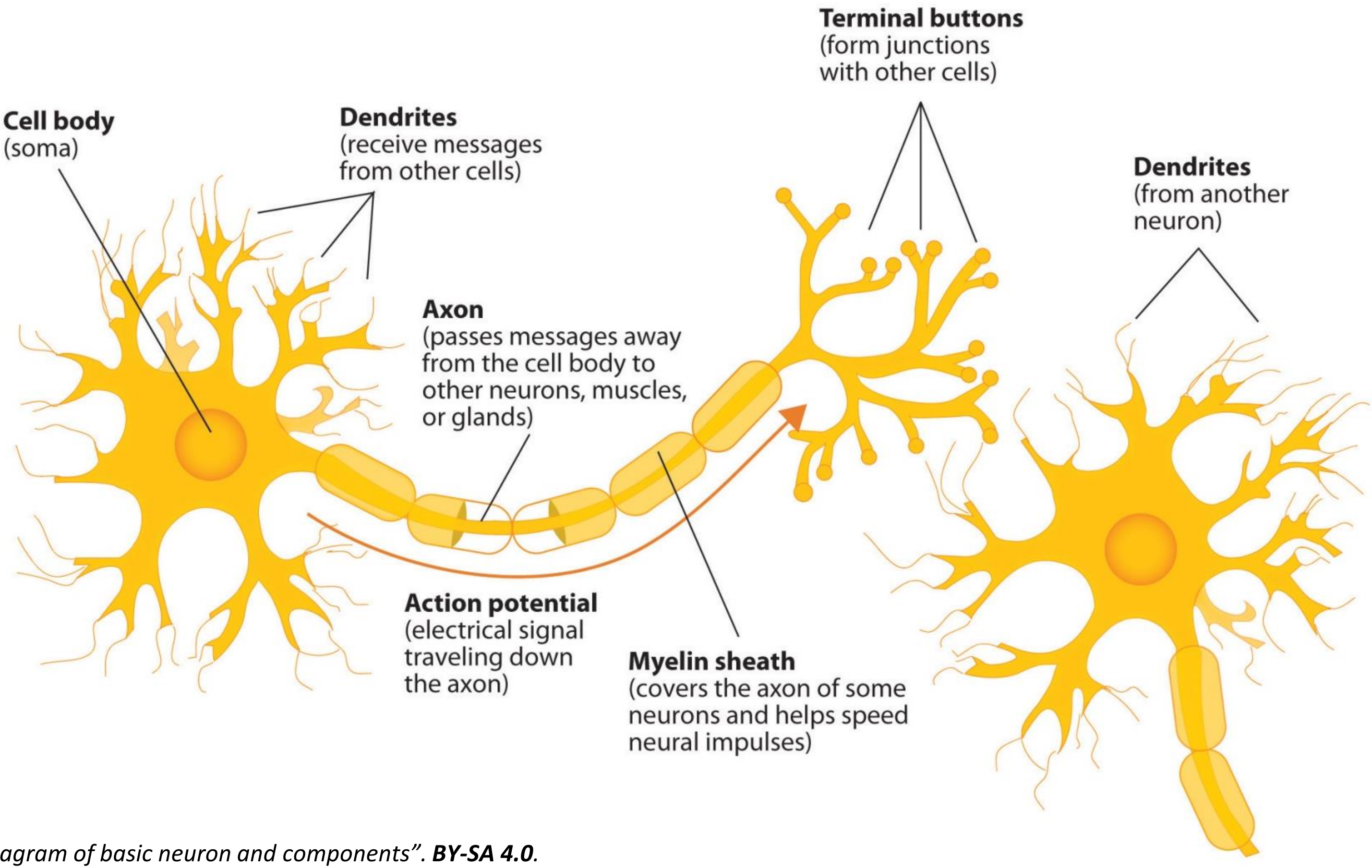
joaof.mari@ufv.br

Agenda

- The biological neuron
- The McCulloch and Pitts neuron
- The Perceptron
- Perceptron's learning algorithm
- Example: Single layer Perceptron

THE BIOLOGICAL NEURON

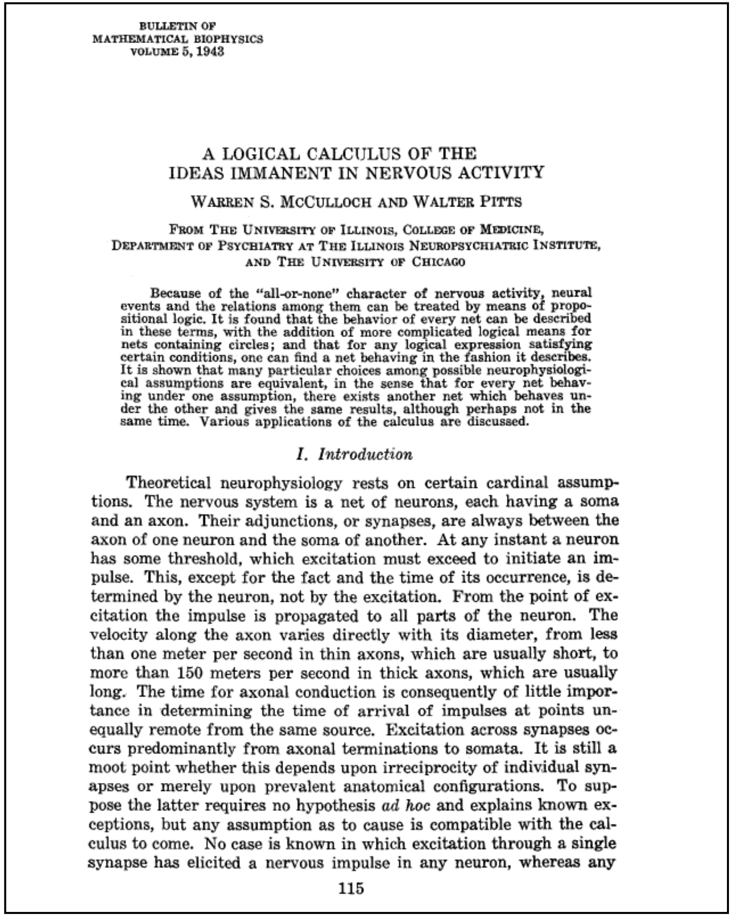
The biological neuron



Jennifer Walinga. " Diagram of basic neuron and components". ***BY-SA 4.0****.*

THE MCCULLOCH AND PITTS NEURON

The McCulloch and Pitts neuron



BULLETIN OF
MATHEMATICAL BIOPHYSICS
VOLUME 5, 1943

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY

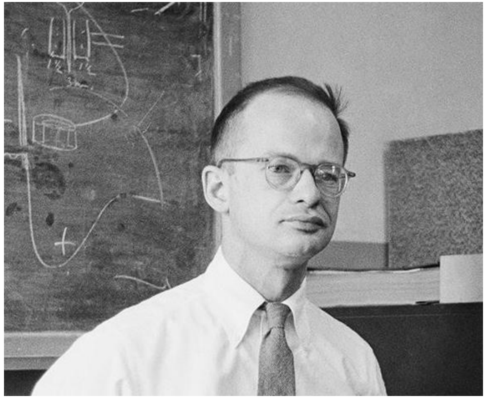
WARREN S. MCCULLOCH AND WALTER PITTS

FROM THE UNIVERSITY OF ILLINOIS, COLLEGE OF MEDICINE,
DEPARTMENT OF PSYCHIATRY AT THE ILLINOIS NEUROPSYCHIATRIC INSTITUTE,
AND THE UNIVERSITY OF CHICAGO

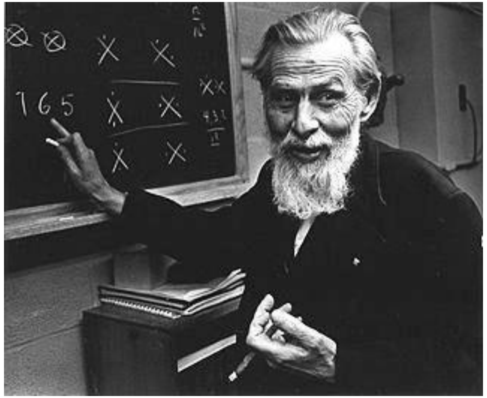
Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.

I. *Introduction*

Theoretical neurophysiology rests on certain cardinal assumptions. The nervous system is a net of neurons, each having a soma and an axon. Their adjunctions, or synapses, are always between the axon of one neuron and the soma of another. At any instant a neuron has some threshold, which excitation must exceed to initiate an impulse. This, except for the fact and the time of its occurrence, is determined by the neuron, not by the excitation. From the point of excitation the impulse is propagated to all parts of the neuron. The velocity along the axon varies directly with its diameter, from less than one meter per second in thin axons, which are usually short, to more than 150 meters per second in thick axons, which are usually long. The time for axonal conduction is consequently of little importance in determining the time of arrival of impulses at points unequally remote from the same source. Excitation across synapses occurs predominantly from axonal terminations to somata. It is still a moot point whether this depends upon irreciprocity of individual synapses or merely upon prevalent anatomical configurations. To suppose the latter requires no hypothesis *ad hoc* and explains known exceptions, but any assumption as to cause is compatible with the calculus to come. No case is known in which excitation through a single synapse has elicited a nervous impulse in any neuron, whereas any

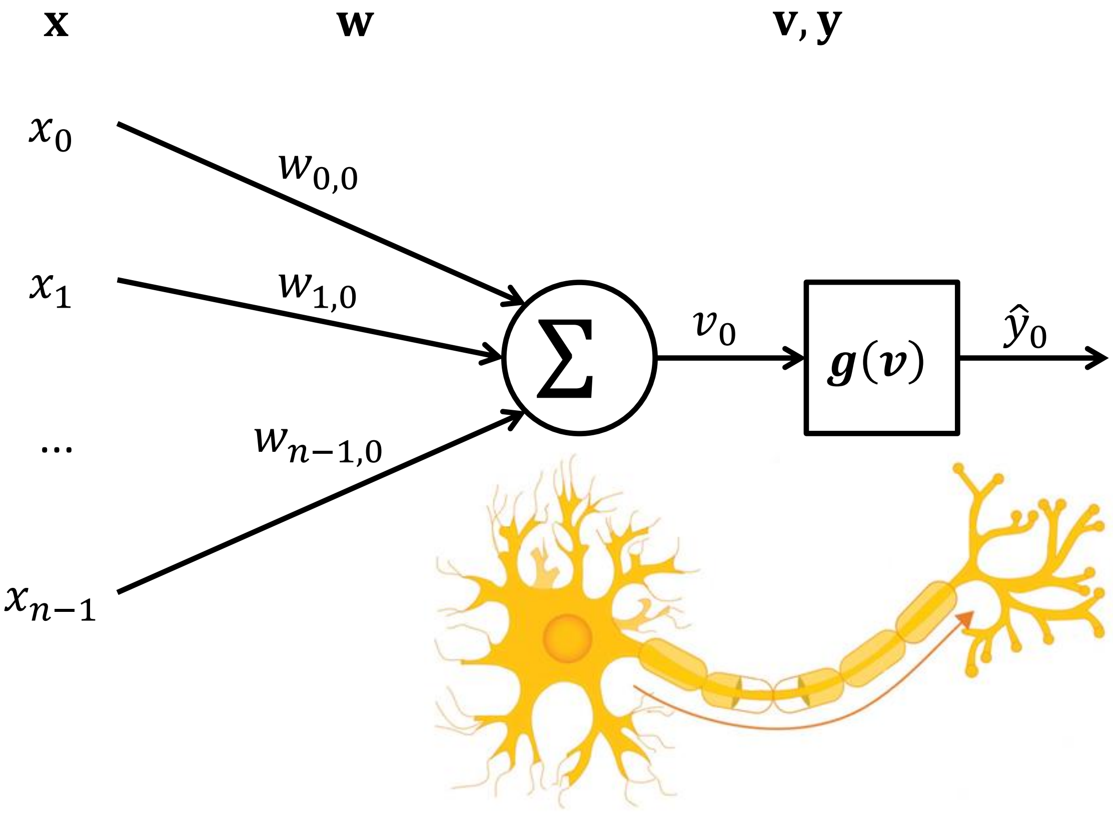


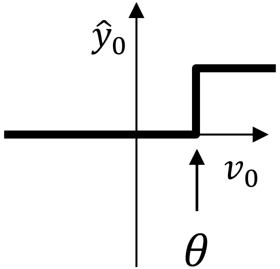
Walter Pitts
(1923 – 1969)
Logic and computational neuroscience



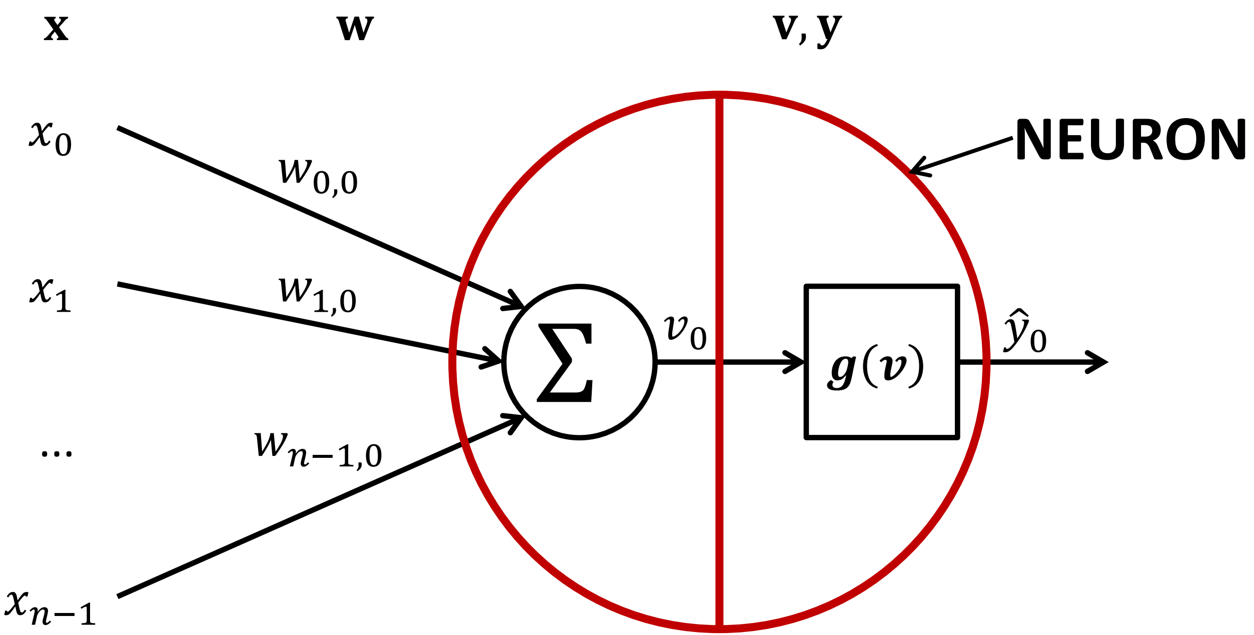
Warren McCulloch
(1898 – 1969)
Neurophysiology

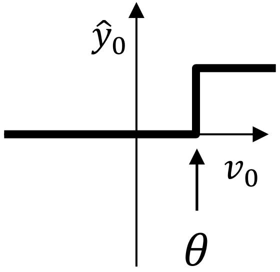
The McCulloch and Pitts neuron



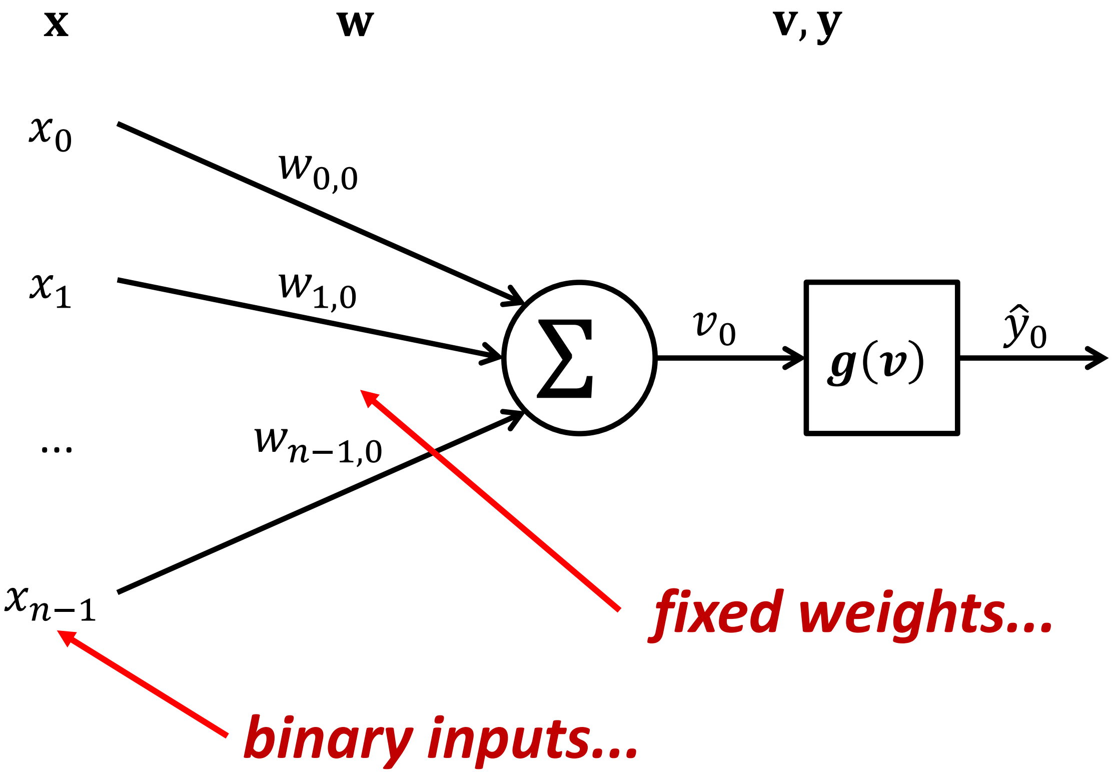


The McCulloch and Pitts neuron



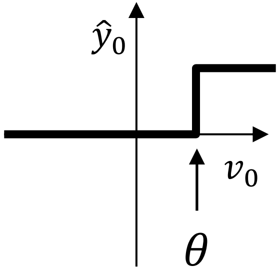


The McCulloch and Pitts neuron

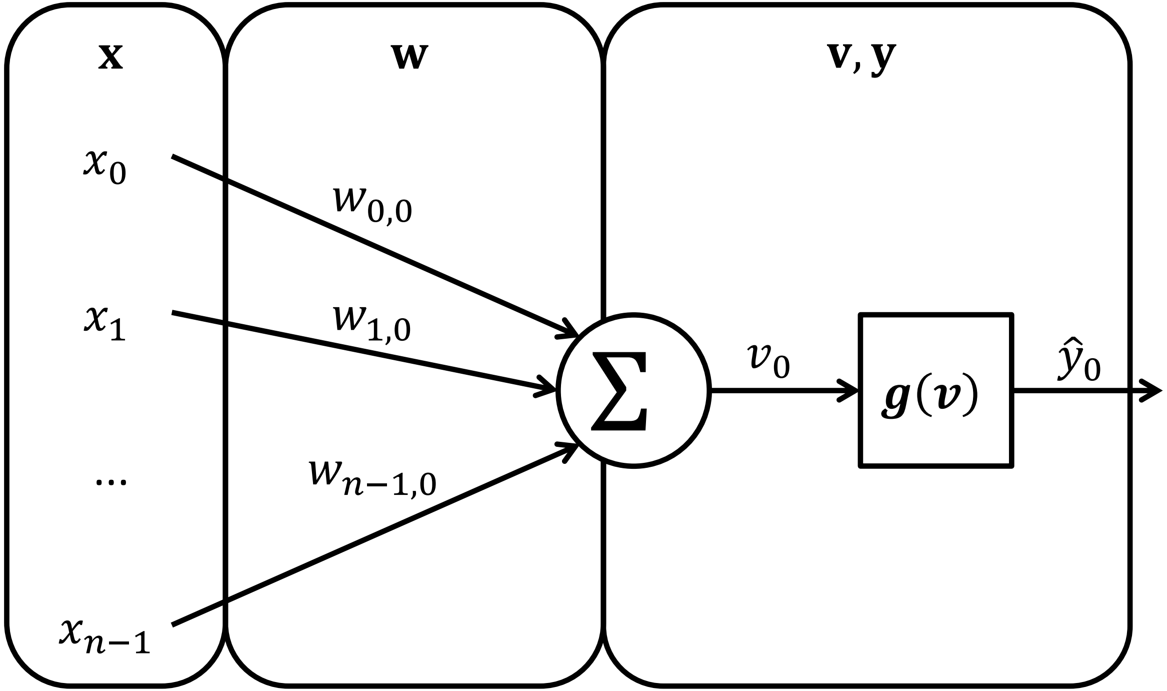


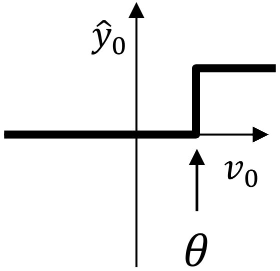
fixed weights...

binary inputs...

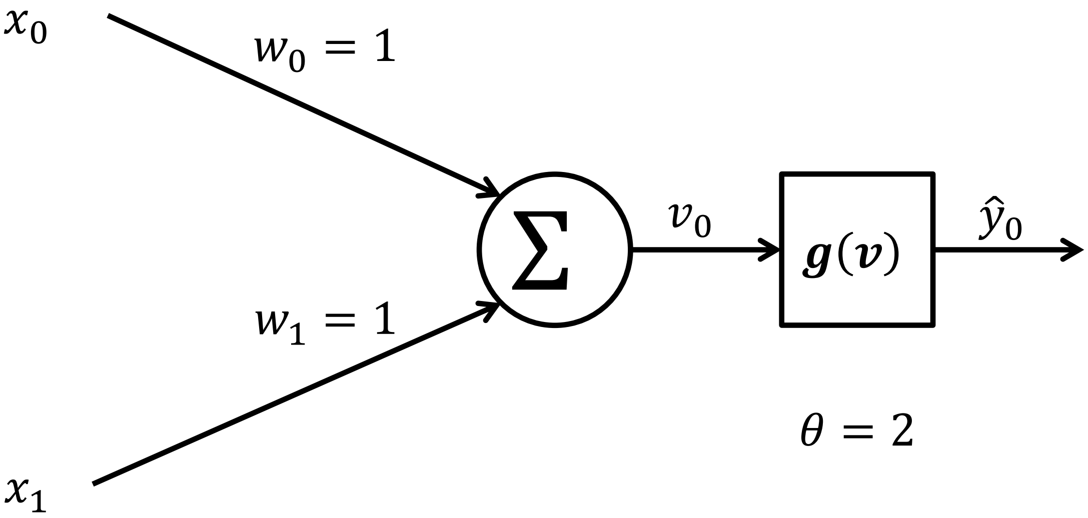


The McCulloch and Pitts neuron





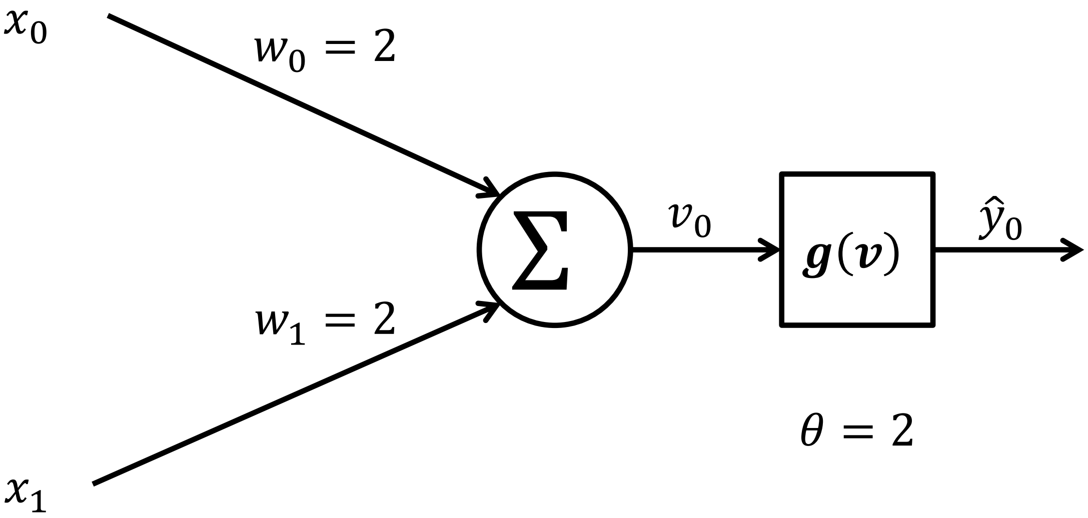
The McCulloch and Pitts neuron



AND

x_0	x_1	y	v	$\hat{y}$
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	1	2	1

The McCulloch and Pitts neuron

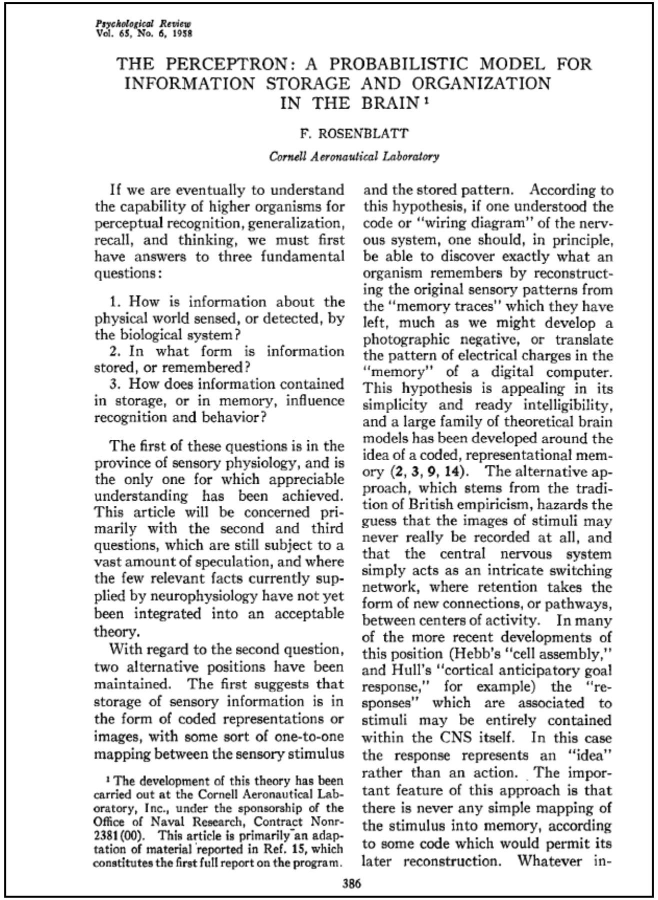


OR

x_0	x_1	y	v	$\hat{y}$
0	0	0	0	0
0	1	1	2	1
1	0	1	2	1
1	1	1	4	1

THE PERCEPTRON

The Perceptron



Psychological Review
Vol. 65, No. 6, 1958

THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN[1]

F. ROSENBLATT

Cornell Aeronautical Laboratory

If we are eventually to understand the capability of higher organisms for perceptual recognition, generalization, recall, and thinking, we must first have answers to three fundamental questions:

1. How is information about the physical world sensed, or detected, by the biological system?
2. In what form is information stored, or remembered?
3. How does information contained in storage, or in memory, influence recognition and behavior?

The first of these questions is in the province of sensory physiology, and is the only one for which appreciable understanding has been achieved. This article will be concerned primarily with the second and third questions, which are still subject to a vast amount of speculation, and where the few relevant facts currently supplied by neurophysiology have not yet been integrated into an acceptable theory.

With regard to the second question, two alternative positions have been maintained. The first suggests that storage of sensory information is in the form of coded representations or images, with some sort of one-to-one mapping between the sensory stimulus and the stored pattern. According to this hypothesis, if one understood the code or "wiring diagram" of the nervous system, one should, in principle, be able to discover exactly what an organism remembers by reconstructing the original sensory patterns from the "memory traces" which they have left, much as we might develop a photographic negative, or translate the pattern of electrical charges in the "memory" of a digital computer. This hypothesis is appealing in its simplicity and ready intelligibility, and a large family of theoretical brain models has been developed around the idea of a coded, representational memory (**2**, **3**, **9**, **14**). The alternative approach, which stems from the tradition of British empiricism, hazards the guess that the images of stimuli may never really be recorded at all, and that the central nervous system simply acts as an intricate switching network, where retention takes the form of new connections, or pathways, between centers of activity. In many of the more recent developments of this position (Hebb's "cell assembly," and Hull's "cortical anticipatory goal response," for example) the "responses" which are associated to stimuli may be entirely contained within the CNS itself. In this case the response represents an "idea" rather than an action. The important feature of this approach is that there is never any simple mapping of the stimulus into memory, according to some code which would permit its later reconstruction. Whatever in-

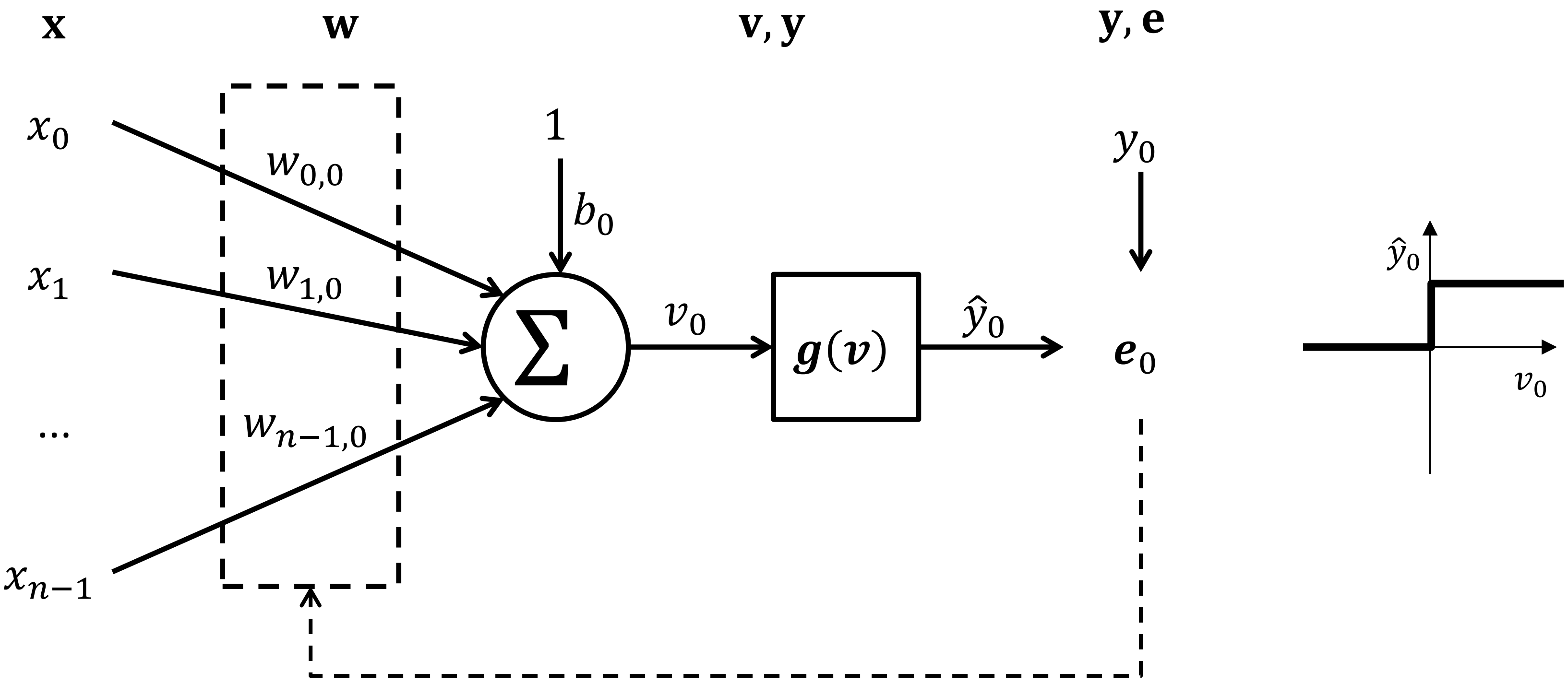
[1] The development of this theory has been carried out at the Cornell Aeronautical Laboratory, Inc., under the sponsorship of the Office of Naval Research, Contract Nonr-2381(00). This article is primarily an adaptation of material reported in Ref. 15, which constitutes the first full report on the program.

386

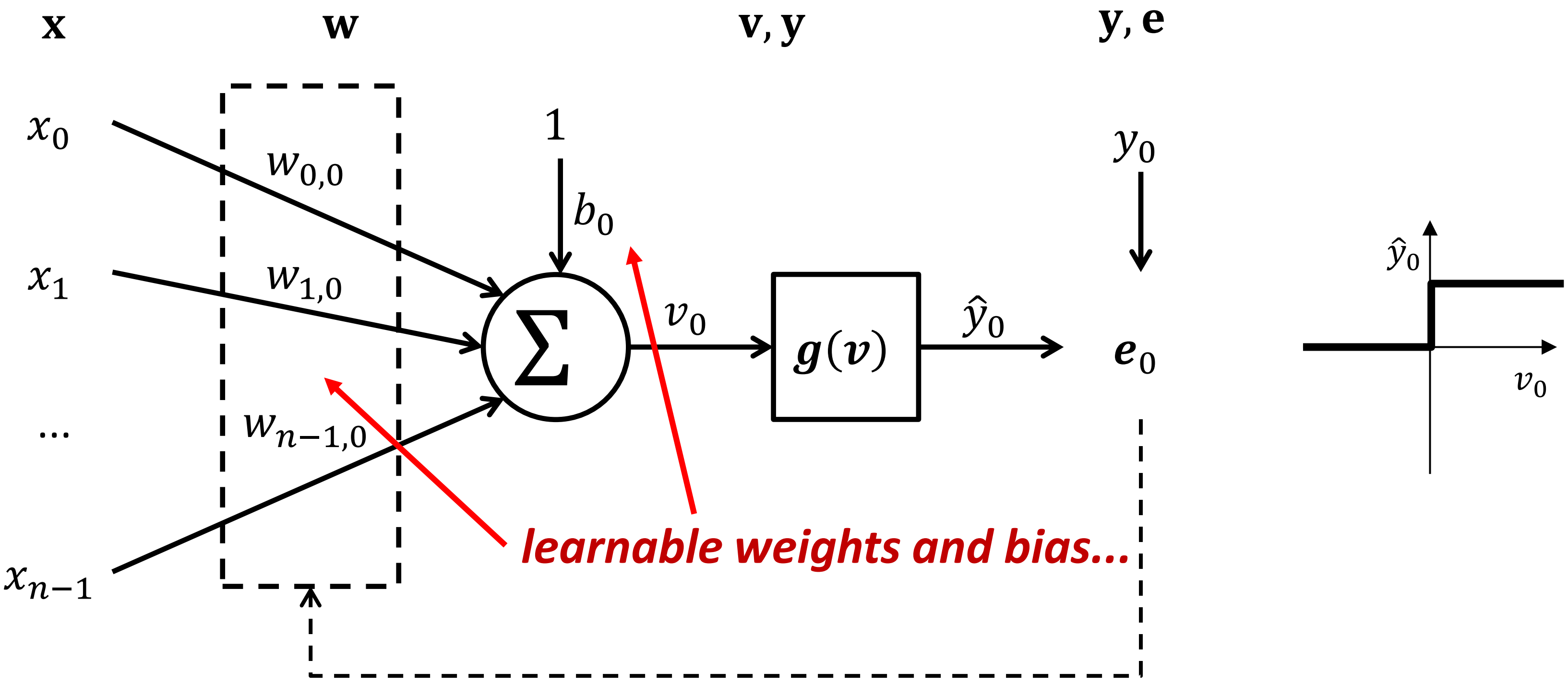
Frank Rosenblatt
(1928 – 1971)
Psycology

https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon

The Perceptron

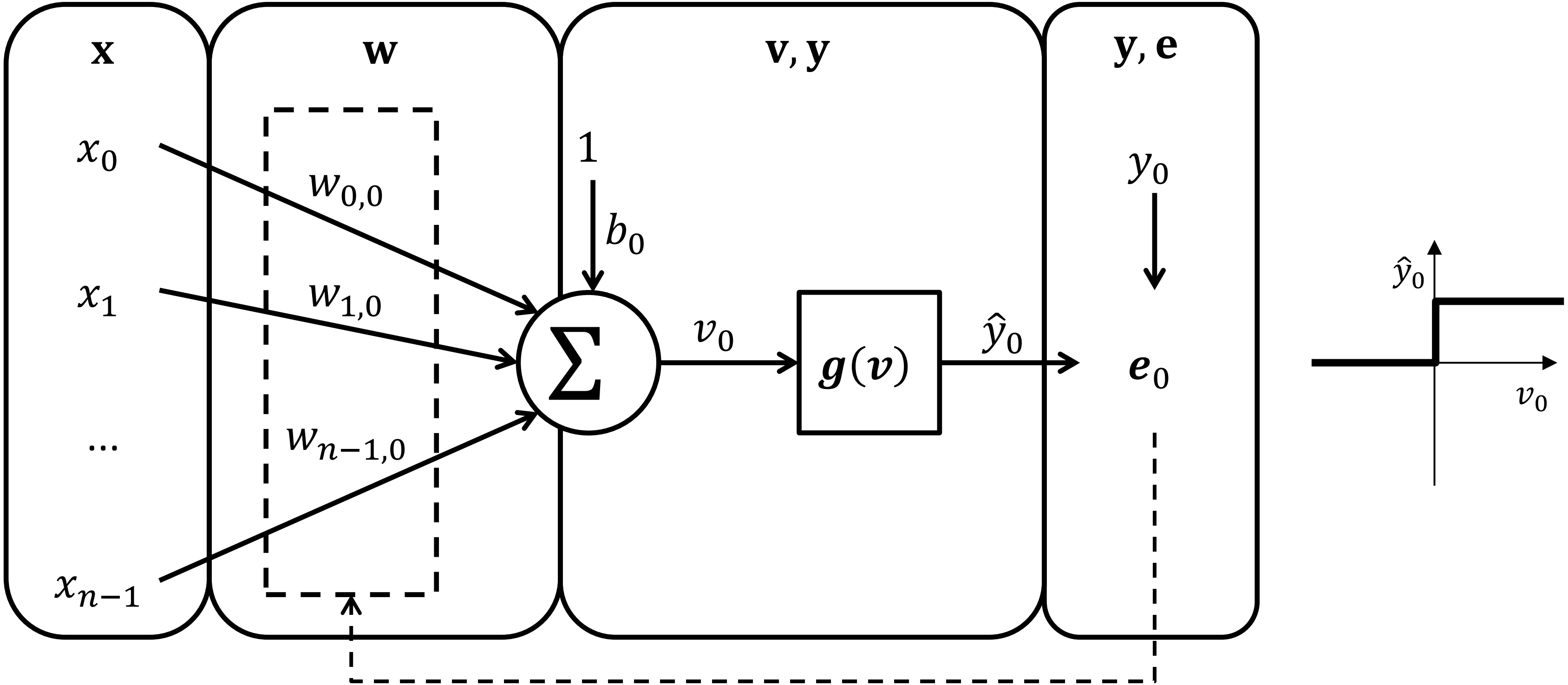


The Perceptron

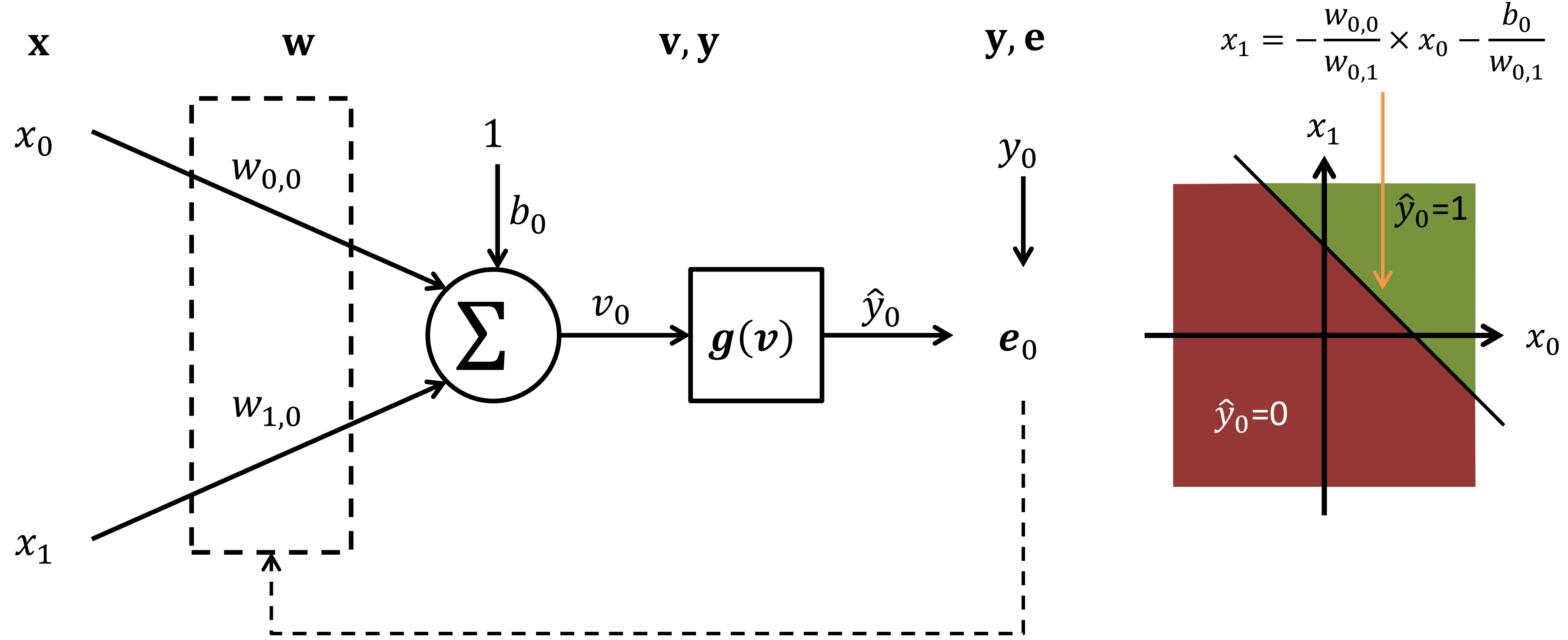


learnable weights and bias...

The Perceptron



The Perceptron

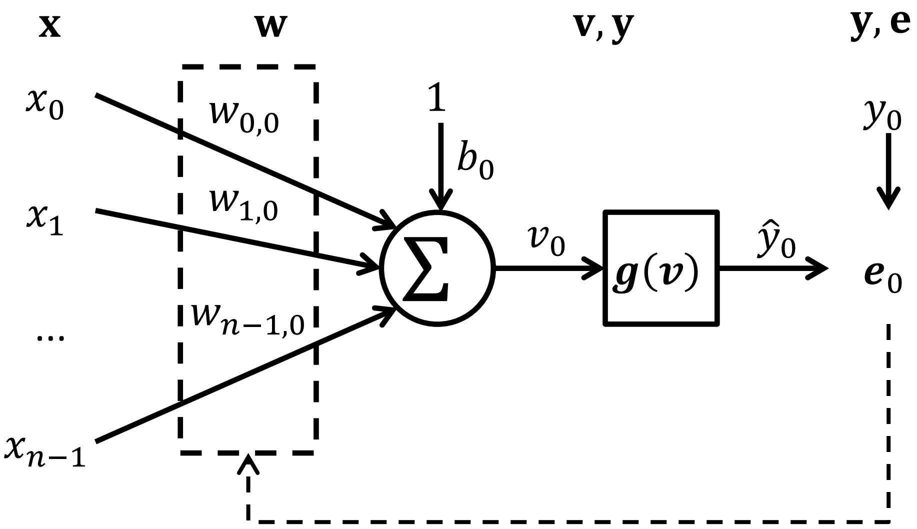


$$x_1 = -\frac{w_{0,0}}{w_{0,1}} \times x_0 - \frac{b_0}{w_{0,1}}$$

PERCEPTRON'S LEARNING ALGORITHM

Perceptron's learning algorithm

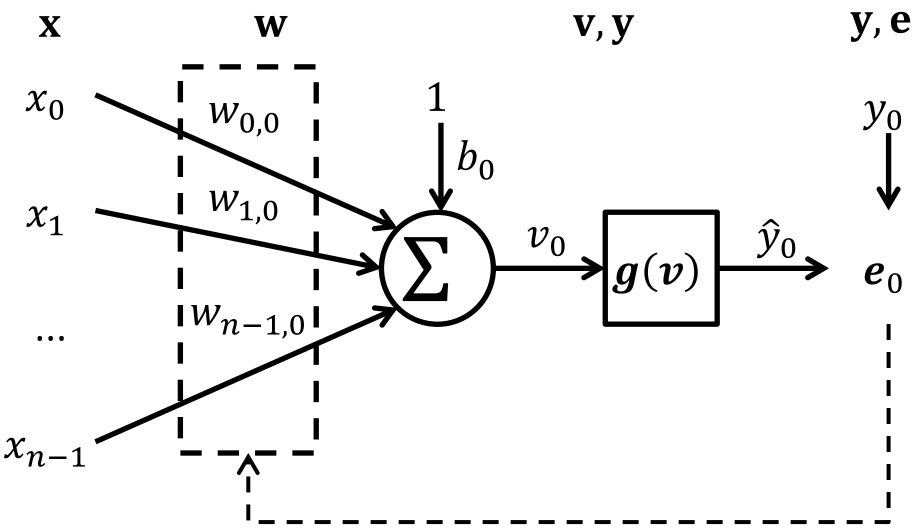
- For *t* from *1* to *max_epochs*:
  - $e_{epoch} = 0$
  - For each ***x***, *y* in (***X***, ***y***):
    - $v_0 = \sum_{i=0}^{n-1} w_{i,0} x_i + b_0$
    - $\hat{y}_0 = g(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases}$
    - $e_0 = y_0 - \hat{y}_0$
    - $\Delta w_{i,0} = e_0 x_i,$
      - $\text{for } i = 0, 1, \dots, n-1$
    - $w_{i,0}(t) = w_{i,0}(t-1) + \eta \Delta w_{i,0},$
      - $\text{for } i = 0, 1, \dots, n-1$
    - $b_0(t) = b_0(t-1) + \eta e_0$
    - $e_{epoch} = e_{epoch} + (e_0)^2$
  - $e_{epoch} = e_{epoch}/2$
  - If $e_{epoch} < e_{min}$, break the loop.



- **X** : training data
- **y** : training set labels
- η : learning rate
- *max_epochs* : Maximum number of epochs
- ***w*** e *b* : weights and bias – randomly initialized

Perceptron's learning algorithm

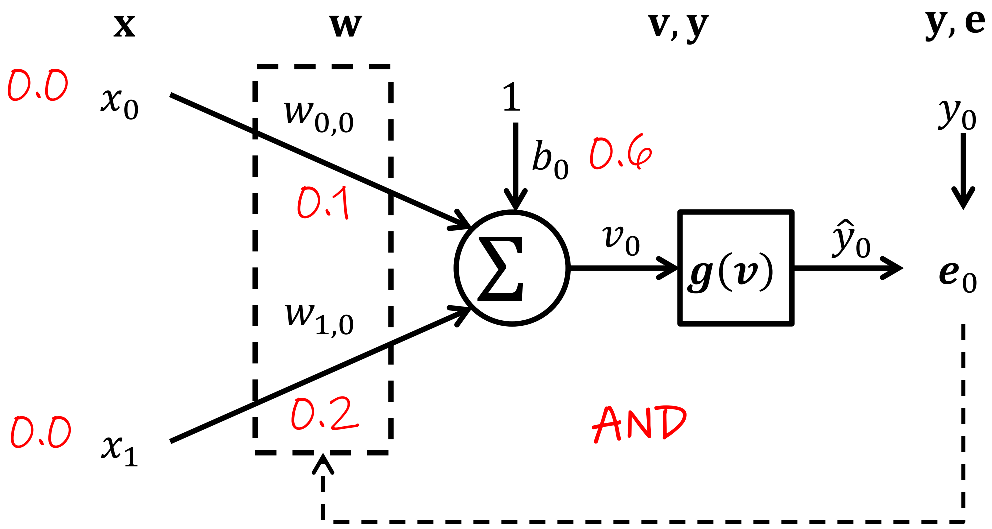
- Inner product :
  - $v_0 = \sum_{i=0}^{n-1} w_{i,0} x_i + b_0$
- Activation function:
  - $\hat{y}_0 = g(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases}$
- Error:
  - $e_0 = y_0 - \hat{y}_0$
- Value used to update the weights:
  - $\Delta w_{i,0} = e_0 x_i,\ for\ i = 0, 1, \dots, n-1$
- Weights update:
  - $w_{i,0}(t) = w_{i,0}(t-1) + \eta \Delta w_{i,0},\ for\ i = 0, 1, \dots, n-1$
- Bias update:
  - $b_0(t) = b_0(t-1) + \eta e_0$



EXAMPLE: SINGLE LAYER PERCEPTRON

Example: Single layer Perceptron

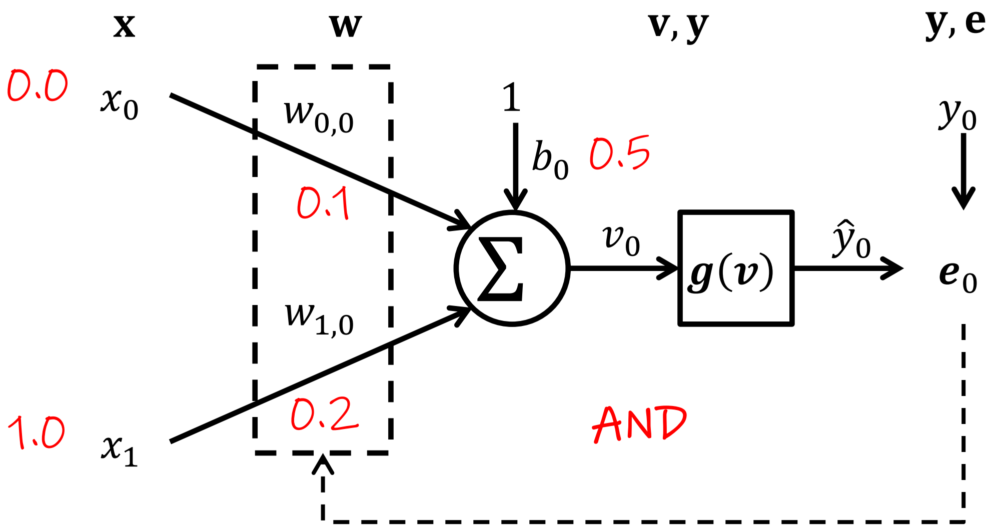
- *Epoch 0:*
  - *Iteration 0:*
  - $\mathbf{x} = [x_0 \quad x_1] = [0 \quad 0]$
  - $\mathbf{w} = \begin{bmatrix} w_{0,0} \\ w_{1,0} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}, \ \mathbf{b} = [b_0] = [0.6]$
  - $v_0 = \mathbf{x}\mathbf{w} + b_0 = [0.0 \quad 0.0] \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} + 0.6$
  - $v_0 = 0.0 + 0.6 = 0.6$
  - $\hat{y}_0 = 1.0, \quad because\ v_0 > 0.0$
  - $e_0 = y - \hat{y} = 0.0 - 1.0 = -1.0$
  - $\Delta\mathbf{w} = \mathbf{e}\mathbf{x} = -1.0[0.0 \quad 0.0] = [0.0 \quad 0.0]$
  - $\mathbf{w}(t) = \mathbf{w}(t-1) + \eta\Delta\mathbf{w} = [0.1 \quad 0.2] + 0.1[0.0 \quad 0.0]$
  - $\mathbf{w}(t) = [0.1 \quad 0.2]$
  - $b_0(t) = b_0(t-1) + \eta e_0 = 0.6 + 0.1 \times (-1.0) = 0.5$
  - $e_{epoch} = e_{epoch} + {e_0}^2 = 0.0 + (-1.0)^2 = 1.0$



$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \mathbf{y} = [0 \quad 0 \quad 0 \quad 1]$$

Example: Single layer Perceptron

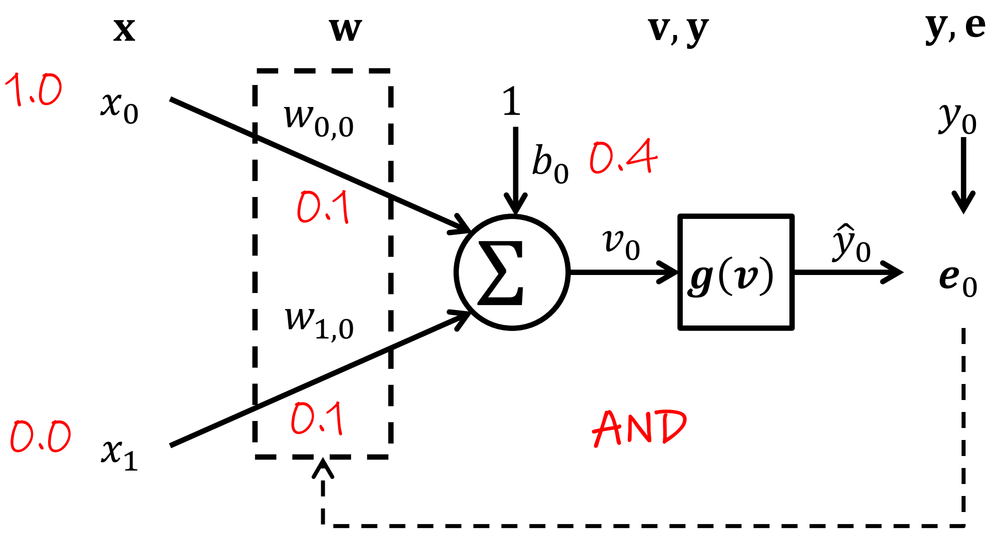
- *Epoch 0:*
  - *Iteration 1:*
  - $\mathbf{x} = [x_0 \quad x_1] = [0.0 \quad 1.0]$
  - $\mathbf{w} = \begin{bmatrix} w_{0,0} \\ w_{1,0} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}, \ \mathbf{b} = [b_0] = [0.5]$
  - $v_0 = \mathbf{xw} + b_0 = [0.0 \quad 1.0] \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} + 0.5$
  - $v_0 = 0.2 + 0.5 = 0.7$
  - $\hat{y}_0 = 1.0, \quad \textit{because } v_0 > 0.0$
  - $e_0 = y - \hat{y} = 0.0 - 1.0 = -1.0$
  - $\Delta\mathbf{w} = \mathbf{ex} = -1.0[0.0 \quad 1.0] = [0.0 \quad -1.0]$
  - $\mathbf{w}(t) = \mathbf{w}(t-1) + \eta\Delta\mathbf{w} = [0.1 \quad 0.2] + 0.1[0.0 \quad -1.0]$
  - $\mathbf{w}(t) = [0.1 \quad 0.1]$
  - $b_0(t) = b_0(t-1) + \eta e_0 = 0.9 + 0.1 \times (-1.0) = 0.4$
  - $e_{epoch} = e_{epoch} + {e_0}^2 = 1.0 + (-1.0)^2 = 2.0$



$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \mathbf{y} = [0 \quad 0 \quad 0 \quad 1]$$

Example: Single layer Perceptron

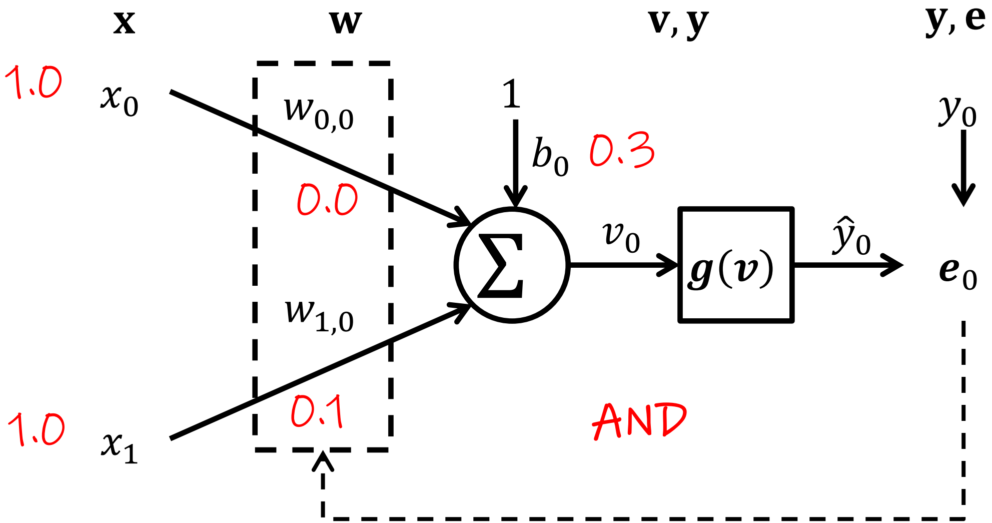
- *Epoch 0:*
  - *Iteration 2:*
  - $\mathbf{x} = [x_0 \quad x_1] = [1.0 \quad 0.0]$
  - $\mathbf{w} = \begin{bmatrix} w_{0,0} \\ w_{1,0} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$, $\mathbf{b} = [b_0] = [0.4]$
  - $v_0 = \mathbf{xw} + b_0 = [1.0 \quad 0.0] \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} + 0.4$
  - $v_0 = 0.1 + 0.4 = 0.5$
  - $\hat{y}_0 = 1.0, \quad because\ v_0 > 0.0$
  - $e_0 = y - \hat{y} = 0.0 - 1.0 = -1.0$
  - $\Delta\mathbf{w} = \mathbf{ex} = -1.0[1.0 \quad 0.0] = [-1.0 \quad 0.0]$
  - $\mathbf{w}(t) = \mathbf{w}(t-1) + \eta\Delta\mathbf{w} = [0.1 \quad 0.1] + 0.1[-1.0 \quad 0.0]$
  - $\mathbf{w}(t) = [0.0 \quad 0.1]$
  - $b_0(t) = b_0(t-1) + \eta e_0 = 0.4 + 0.1 \times (-1.0) = 0.3$
  - $e_{época} = e_{época} + {e_0}^2 = 2.0 + (-1.0)^2 = 3.0$



$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \mathbf{y} = [0 \quad 0 \quad 0 \quad 1]$$

Example: Single layer Perceptron

- *Epoch 0:*
  - *Iteration 3:*
  - $\mathbf{x} = [x_0 \quad x_1] = [1.0 \quad 1.0]$
  - $\mathbf{w} = \begin{bmatrix} w_{0,0} \\ w_{1,0} \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.1 \end{bmatrix}, \; \mathbf{b} = [b_0] = [0.3]$
  - $v_0 = \mathbf{xw} + b_0 = [1.0 \quad 1.0] \begin{bmatrix} 0.0 \\ 0.1 \end{bmatrix} + 0.3$
  - $v_0 = 0.1 + 0.3 = 0.4$
  - $\hat{y}_0 = 1.0, \quad because\ v_0 > 0.0$
  - $e_0 = y - \hat{y} = 1.0 - 1.0 = 0.0$
  - $\Delta\mathbf{w} = \mathbf{ex} = 0.0[1.0 \quad 1.0] = [0.0 \quad 0.0]$
  - $\mathbf{w}(t) = \mathbf{w}(t-1) + \eta\Delta\mathbf{w} = [0.0 \quad 0.1] + 0.1[0.0 \quad 0.0]$
  - $\mathbf{w}(t) = [0.0 \quad 0.1]$
  - $b_0(t) = b_0(t-1) + \eta e_0 = 0.3 + 0.1 \times (0.0) = 0.3$
  - $e_{epoch} = e_{epoch} + {e_0}^2 = 3.0 + (0.0)^2 = 3.0$

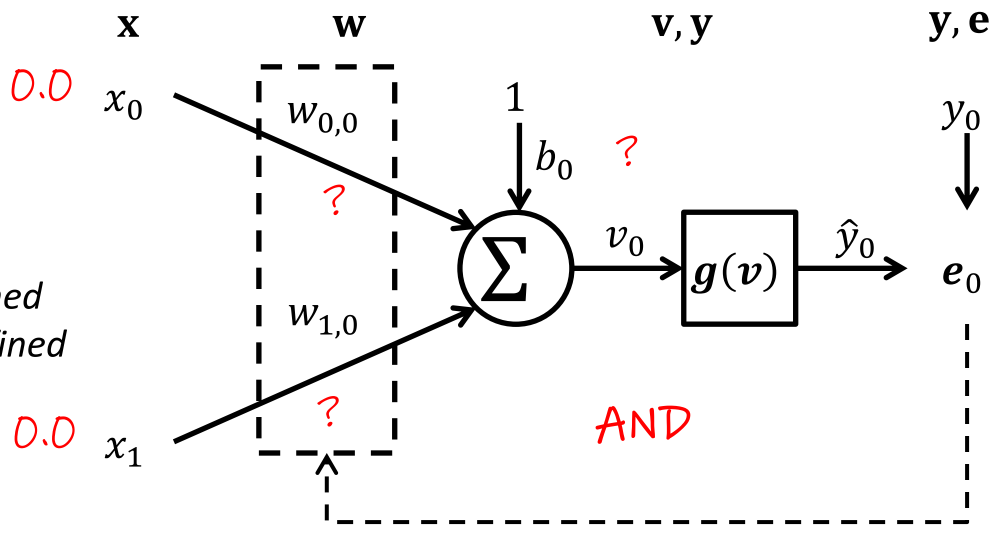


$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad \mathbf{y} = [0 \quad 0 \quad 0 \quad 1]$$

Example: Single layer Perceptron

- *End of Epoch 0.*
  - $e_{epoch} = \frac{e_{epoch}}{2} = \frac{3.0}{2} = 1.5$

- *Epoch 1:*
  - *Iteration 0:*
    - *Repeat until the limit of epochs is reached or the epoch error falls below a pre-defined threshold...*



$$\mathbf{X} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \quad \mathbf{y} = [0 \quad 0 \quad 0 \quad 1]$$

Bibliography

- Gonzalez, R.C.; Woods, R.E. **Digital Image Processing.**
  - Pearson. 3rd edition, 2007.
- Ponti et al. **Everything You Wanted to Know about Deep Learning for Computer Vision but Were Afraid to Ask**. Sibgrapi 2017.
  - https://sites.icmc.usp.br/moacir/p17sibgrapi-tutorial/
- Görner, M. **Learn TensorFlow and deep learning, without a Ph.D.**
  - https://cloud.google.com/blog/products/gcp/learn-tensorflow-and-deep-learning-without-a-phd
- CS231n: Convolutional Neural Networks for Visual Recognition
  - http://cs231n.github.io/
- Goodfellow, Bengio e Courville. **Deep Learning**. MIT Press, 2016
  - https://www.deeplearningbook.org/

```
@misc{mari_im_proc_2023,
       author = {João Fernando Mari},
       title = {Artificial neural networks},
       year = {2023},
       publisher = {GitHub},
       journal = {Introduction to digital image processing - UFV},
       howpublished = {\url{https://github.com/joaofmari/SIN392_Introduction-to-digital-image-processing_2023}}
}
```

THE END