[AULA 01] Redes Neurais Artificiais

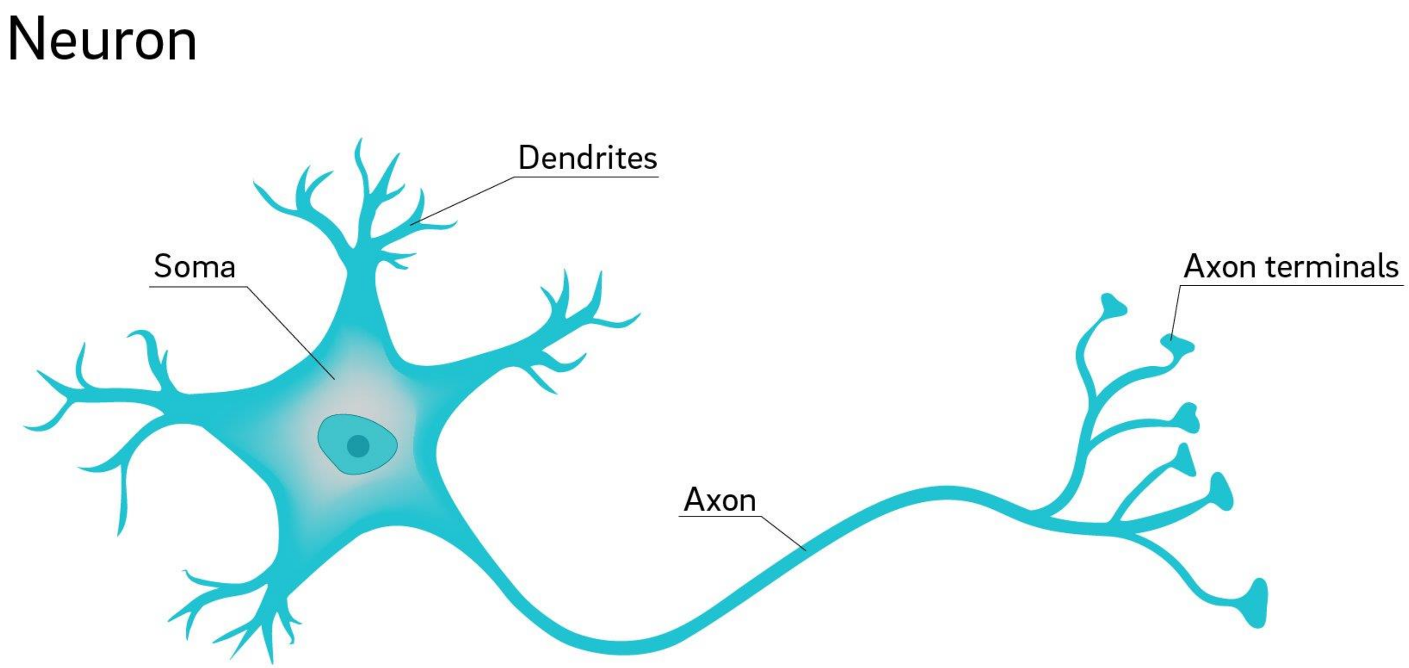
Prof. João F. Mari

joaof.mari@ufv.br

Roteiro

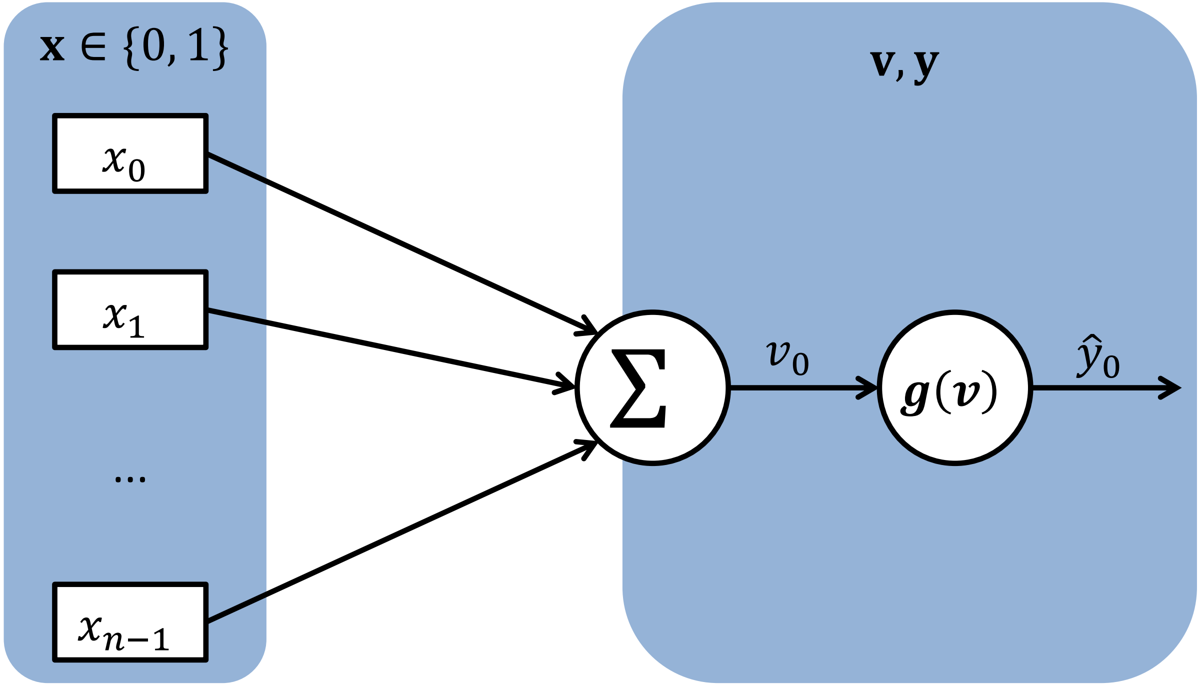
- Neurônio biológico
- Modelo de neurônio de McCulloch e Pitts
- O Perceptron
- Perceptron de camada simples
  - Algoritmo
  - Hiperplano de separação
  - Notação vetorial
  - Exemplo
  - Múltiplas classes
- Perceptron de múltiplas camadas
  - Estrutura da rede
  - Backpropagation - Algoritmo e equações
  - Exemplo
- Referencias

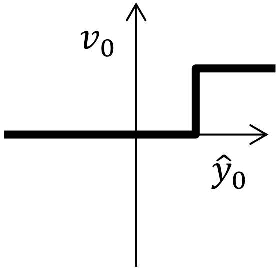
Neurônio biológico



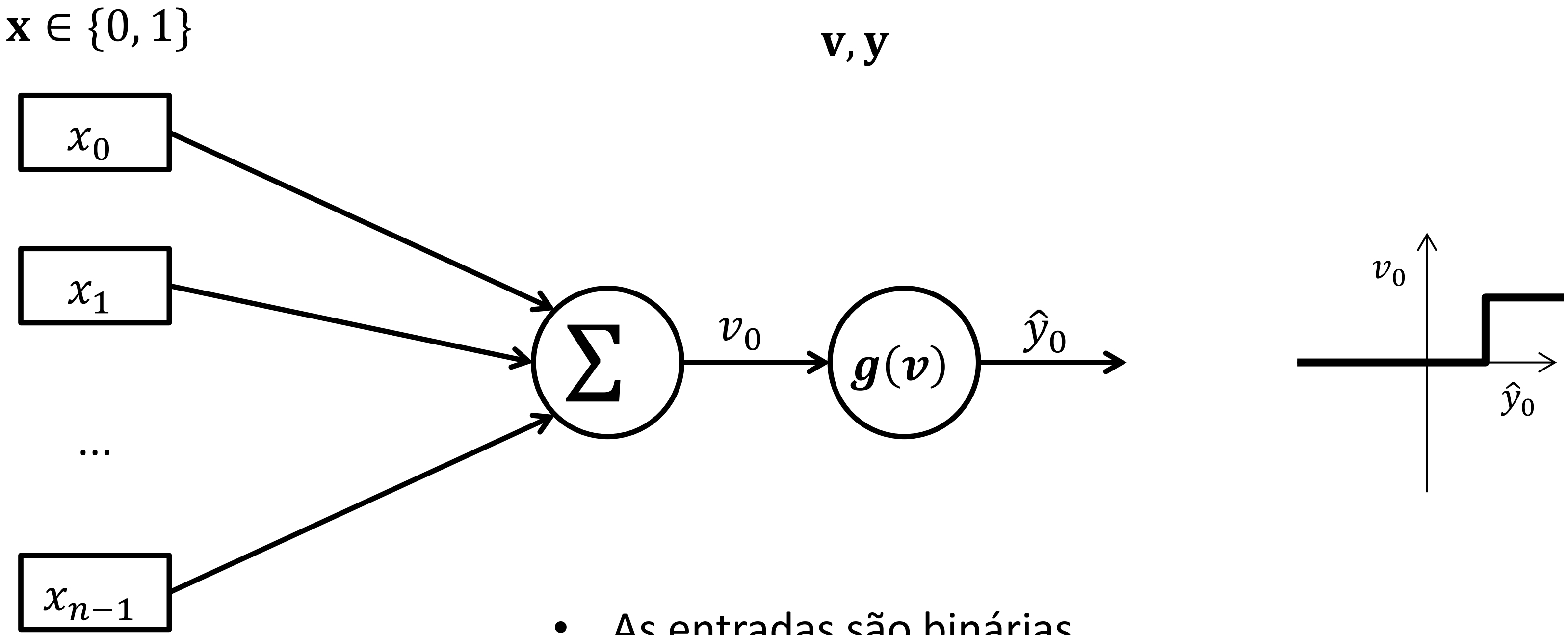
https://medicalxpress.com/news/2018-07-neuron-axons-spindly-theyre-optimizing.html

Modelo de neurônio de McCulloch e Pitts





Modelo de neurônio de McCulloch e Pitts



- As entradas são binárias
- Não possui pesos ajustáveis

Modelo de neurônio de McCulloch e Pitts

BULLETIN OF
MATHEMATICAL BIOPHYSICS
VOLUME 5, 1943

A LOGICAL CALCULUS OF THE
IDEAS IMMANENT IN NERVOUS ACTIVITY

WARREN S. MCCULLOCH AND WALTER PITTS

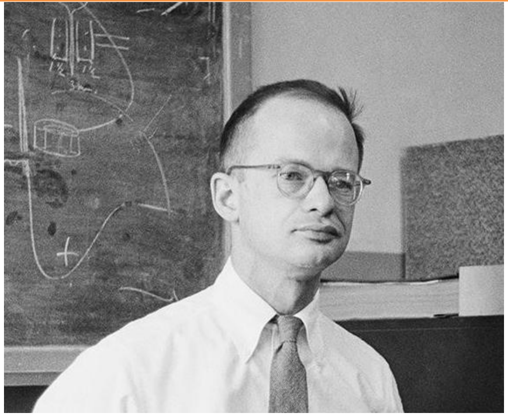
FROM THE UNIVERSITY OF ILLINOIS, COLLEGE OF MEDICINE,
DEPARTMENT OF PSYCHIATRY AT THE ILLINOIS NEUROPSYCHIATRIC INSTITUTE,
AND THE UNIVERSITY OF CHICAGO

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.

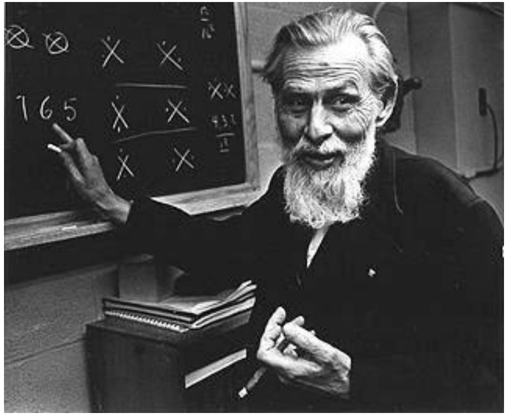
I. Introduction

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

115



Walter Pitts



Warren McCulloch

O 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.

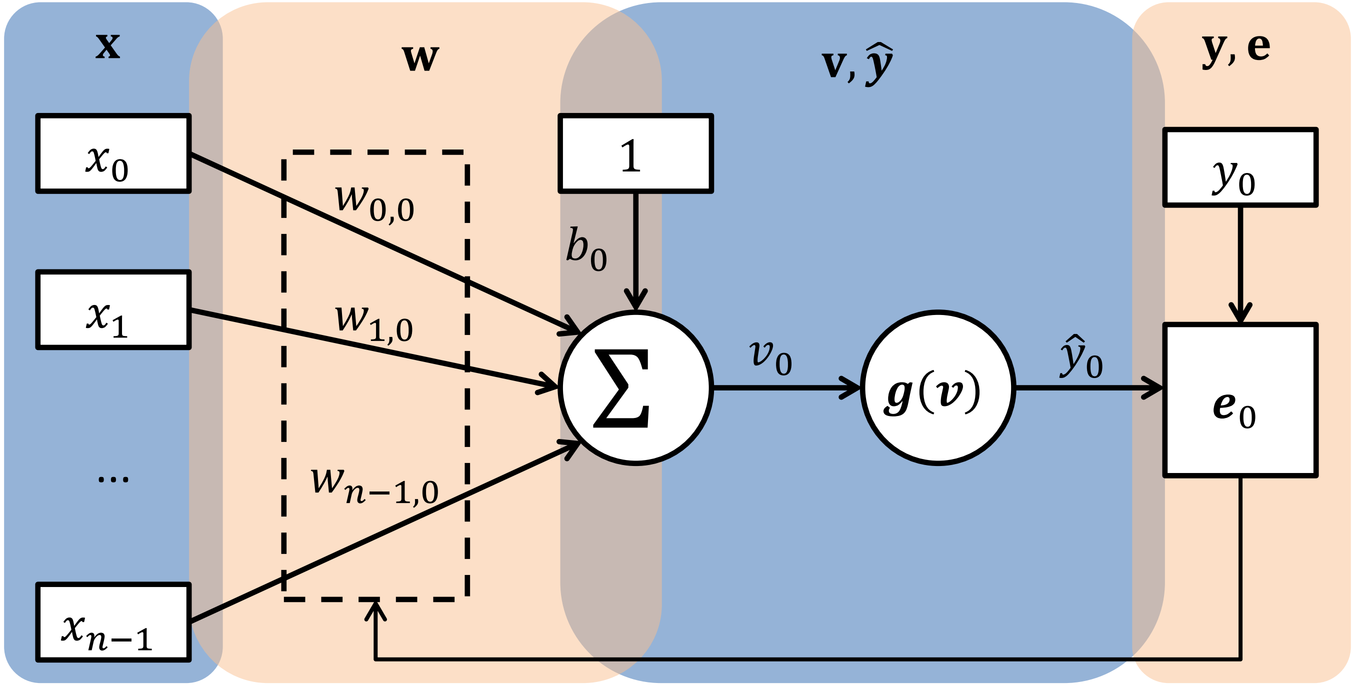
With regard to the second question, two alternative positions have been maintained. The first suggests that storage of sensory information is in the form of coded representations or images, with some sort of one-to-one mapping between the sensory stimulus and the stored pattern. According to this hypothesis, 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.

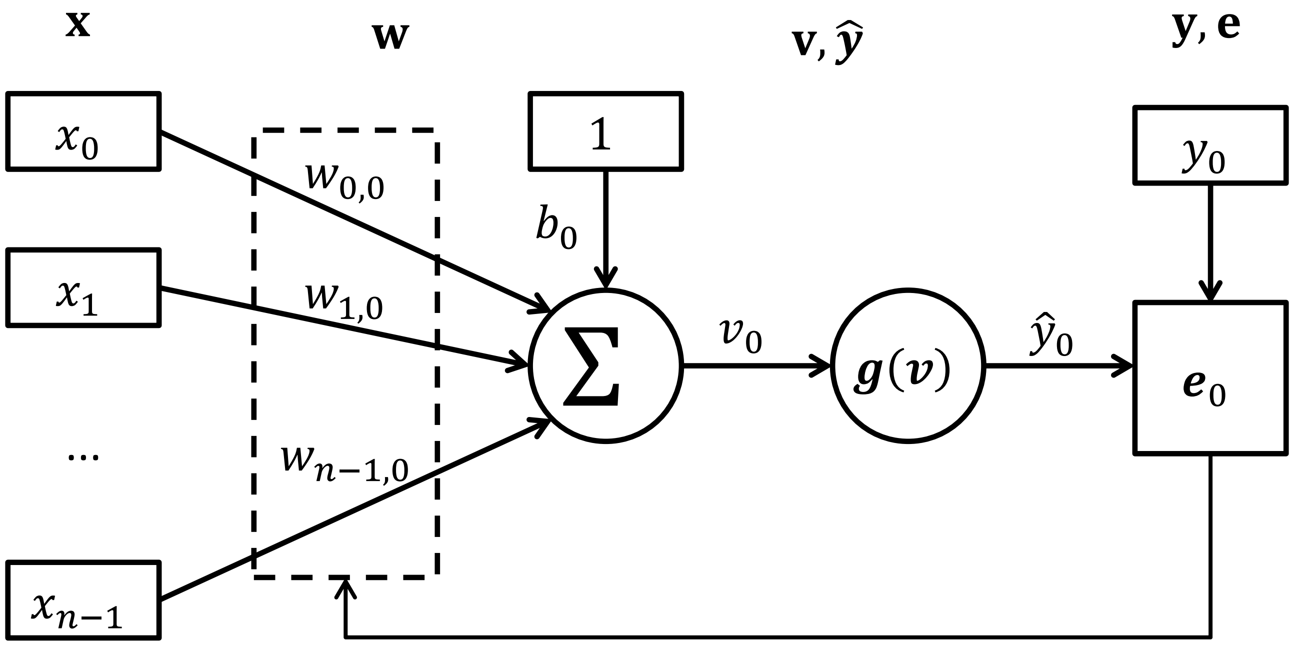
386

Frank Rosenblatt

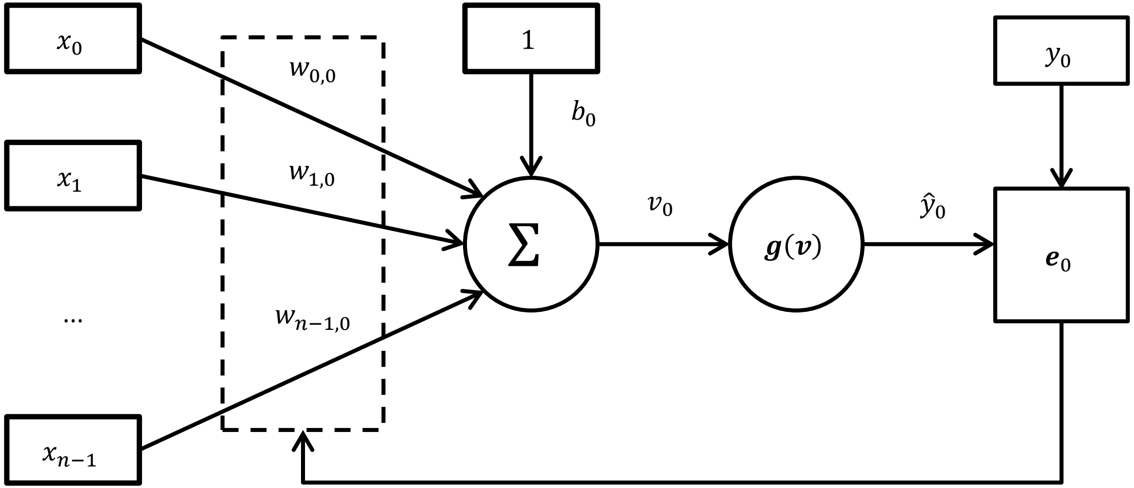
Perceptron de camada simples



Perceptron de camada simples



Perceptron de camada simples



Produto interno:

$$v_0 = \sum_{i=0}^{n-1} w_{i,0} x_i + b_0$$

Função de ativação:

$$\hat{y}_0 = g(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases}$$

Erro:

$$e_0 = y_0 - \hat{y}_0$$

Valor utilizado para atualizar os pesos:

$$\Delta w_{i,0} = e_0 x_i, \qquad para\ i = 0, 1, \dots, n-1$$

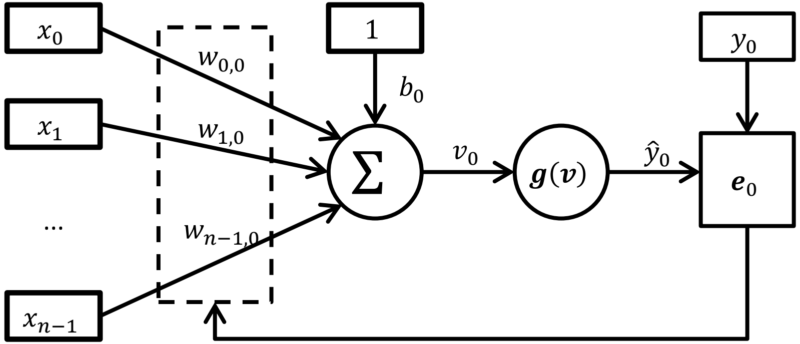
Atualização dos pesos:

$$w_{i,0}(t) = w_{i,0}(t-1) + \eta \Delta w_{i,0}, \qquad para\ i = 0, 1, \dots, n-1$$

Atualização do bias:

$$b_0(t) = b_0(t-1) + \eta e_0$$

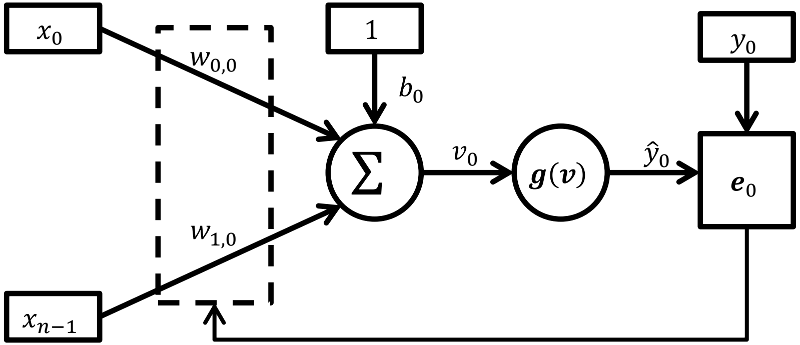
Perceptron de camada simples – algoritmo

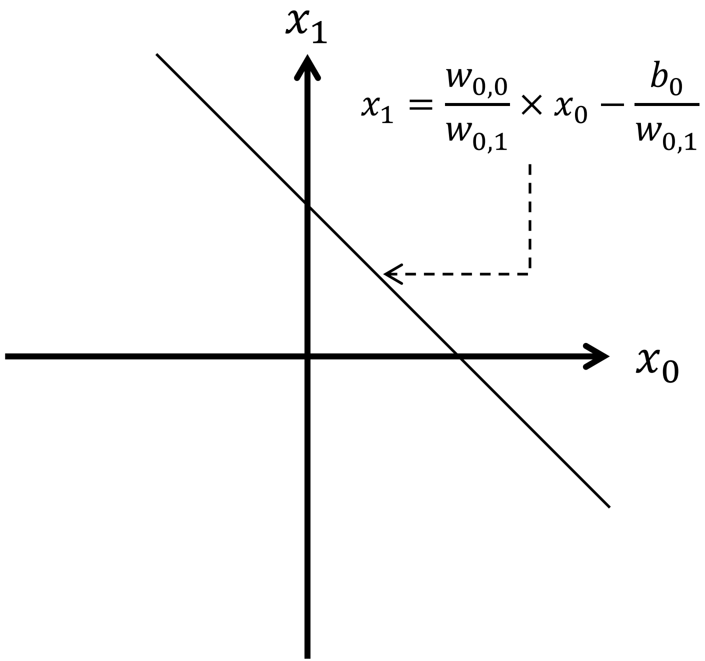


X : Matriz de dados de treinamento
y : vetor de rótulos do conjunto de treinamento
Definir a taxa de aprendizado: η
Definir o número máximo de épocas de treinamento: *max_epocas*
Inicializar os pesos e bias com valores aleatórios: **w** e b

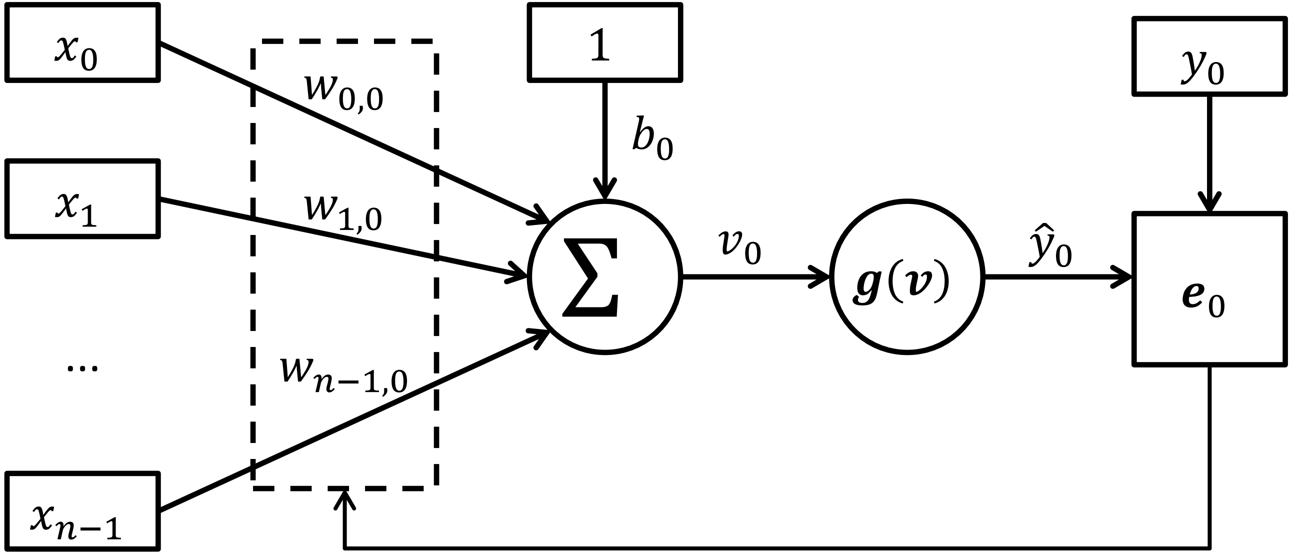
Para $\boldsymbol{t}$ de 1 até *max_epocas*:
- $e_{época} = 0$
- Para todo **x**, y em (**X**, **y**):
  - Calcular o produto inteiro do neurônio:
  - $v_0 = \sum_{i=0}^{n-1} w_{i,0} x_i + b_0$
  - Calcular a função de ativação:
  - $\hat{y}_0 = g(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases}$
  - Calcular o erro
  - $e_0 = y_0 - \hat{y}_0$
  - Calcular o valor de Δ para atualizar pesos **w**:
  - $\Delta w_{i,0} = e_0 x_i,\ para\ i = 0, 1, \dots, n-1$
  - Atualizar os pesos **w**:
  - $w_{i,0}(t) = w_{i,0}(t-1) + \eta \Delta w_{i,0},\ para\ i = 0, 1, \dots, n-1$
  - Atualizar o bias b:
  - $b_0(t) = b_0(t-1) + \eta e_0$
  - Atualizar o erro da época:
  - $e_{época} = e_{total} + (e_0)^2$
- $e_{época} = e_{época}/2$
- Se $e_{época} < e_{mínimo}$
  - Interromper o treinamento

Perceptron de camada simples – hiperplano de separação





Perceptron de camada simples – notação vetorial



Produto interno:

$$\mathbf{v} = \mathbf{x}\mathbf{w} + \mathbf{b}$$

Função de ativação:

$$\hat{\mathbf{y}} = g(\mathbf{v}) = \begin{cases} +1, & \mathbf{v} \geq 0 \\ -1, & \mathbf{v} < 0 \end{cases}$$

Erro:

$$\mathbf{e} = \mathbf{y}_0 - \hat{\mathbf{y}}_0$$

Valor utilizado para atualizar os pesos:

$$\Delta\mathbf{w} = \mathbf{e}\mathbf{x}$$

Atualização dos pesos:

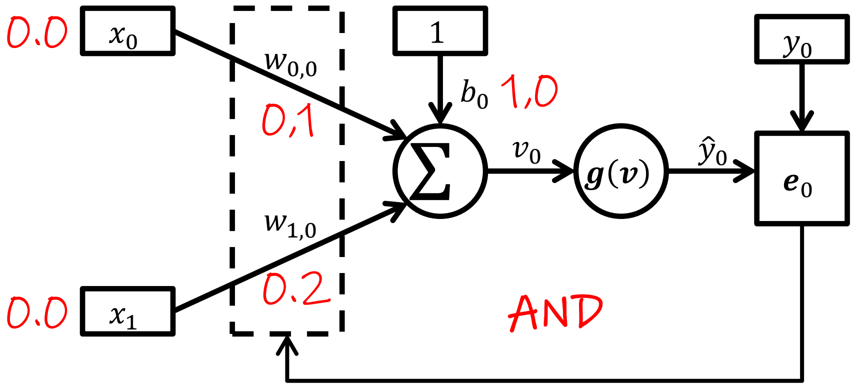
$$\mathbf{w}(t) = \mathbf{w}(t-1) + \eta\Delta\mathbf{w}$$

Atualização do bias:

$$\mathbf{b}(t) = \mathbf{b}(t-1) + \eta\mathbf{e}$$

Perceptron de camada simples - exemplo

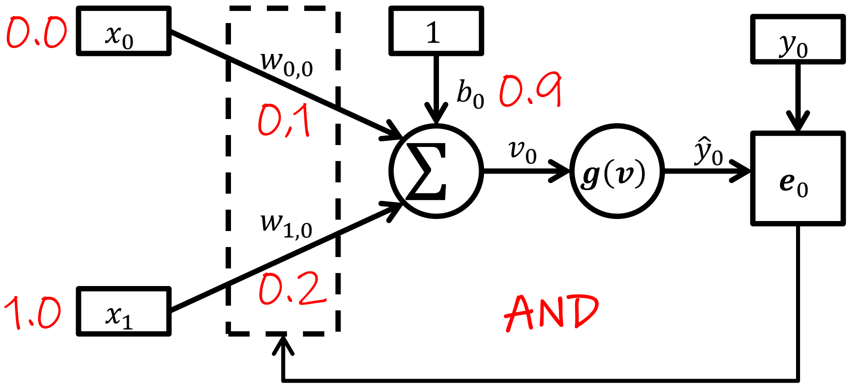
Perceptron de camada simples - exemplo



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

- Época 0:
  - $e_{época} = 0.0$
  - Iteração 0:
    - $\mathbf{x} = [x_0 \quad x_1] = [0 \quad 0]$
    - $\mathbf{w} = \begin{bmatrix} w_{0,0} \\ w_{1,0} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$, $\mathbf{b} = [b_0] = [1.0]$
    - $v_0 = \mathbf{xw} + b_0 = [0.0 \quad 0.0] \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} + 1.0 = 0.0 + 1.0 = 1.0$
    - $\hat{y}_0 = 1.0, \quad pois\ v > 0$
    - $e_0 = y - \hat{y} = 0.0 - 1.0 = -1.0$
    - $\Delta\mathbf{w} = \mathbf{ex} = -1.0[0.0 \quad 0.0] = [0.0 \quad 0.0]$
    - $\mathbf{w}(t) = \mathbf{w}(t-1) + \eta\Delta\mathbf{w} = [0.1 \quad 0.2] + 0.1[0.0 \quad 0.0]$
    - $\mathbf{w}(t) = [0.1 \quad 0.2]$
    - $b_0(t) = b_0(t-1) + \eta e_0 = 1.0 + 0.1 \times (-1.0) = 0.9$
    - $e_{época} = e_{época} + {e_0}^2 = 0.0 + (-1.0)^2 = 1.0$

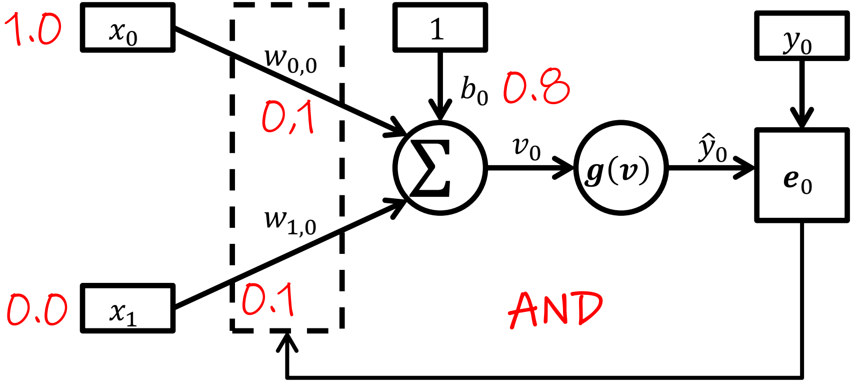
Perceptron de camada simples – exemplo



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

- Época 0:
  - Iteração 1:
    - $\mathbf{x} = [x_0 \quad x_1] = [0.0 \quad 1.0]$
    - $\mathbf{w} = \begin{bmatrix} w_{0,0} \\ w_{1,0} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$, $\mathbf{b} = [b_0] = [0.9]$
    - $v_0 = \mathbf{xw} + b_0 = [0.0 \quad 1.0] \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} + 0.9 = 0.2 + 0.9 = 1.1$
    - $\hat{y}_0 = 1.0, \quad pois\ v_0 > 0.0$
    - $e_0 = y - \hat{y} = 0.0 - 1.0 = -1.0$
    - $\Delta\mathbf{w} = \mathbf{ex} = -1.0[0.0 \quad 1.0] = [0.0 \quad -1.0]$
    - $\mathbf{w}(t) = \mathbf{w}(t-1) + \eta\Delta\mathbf{w} = [0.1 \quad 0.2] + 0.1[0.0 \quad -1.0]$
    - $\mathbf{w}(t) = [0.1 \quad 0.1]$
    - $b_0(t) = b_0(t-1) + \eta e_0 = 0.9 + 0.1 \times (-1.0) = 0.8$
    - $e_{época} = e_{época} + {e_0}^2 = 1.0 + (-1.0)^2 = 2.0$

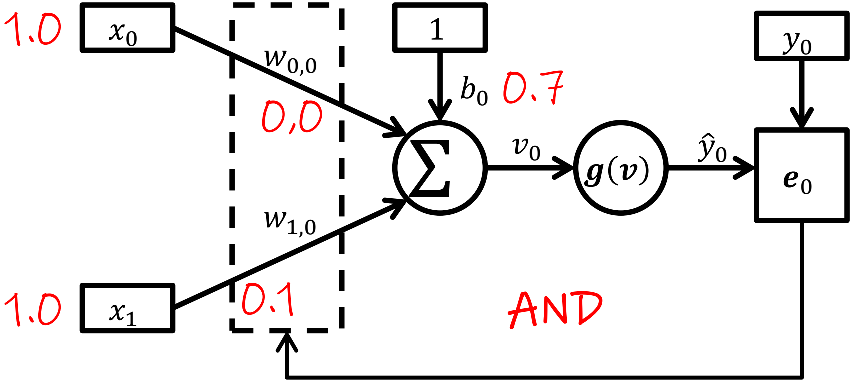
Perceptron de camada simples – exemplo



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

- Época 0:
  - Iteração 2:
    - $\mathbf{x} = [x_0 \quad x_1] = [1.0 \quad 0.0]$
    - $\mathbf{w} = \begin{bmatrix} w_{0,0} \\ w_{1,0} \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix}$, $\mathbf{b} = [b_0] = [0.8]$
    - $v_0 = \mathbf{xw} + b_0 = [1.0 \quad 0.0] \begin{bmatrix} 0.1 \\ 0.1 \end{bmatrix} + 0.8 = 0.2 + 0.8 = 1.1$
    - $\hat{y}_0 = 1.0, \quad pois\ v_0 > 0.0$
    - $e_0 = y - \hat{y} = 0.0 - 1.0 = -1.0$
    - $\Delta\mathbf{w} = \mathbf{ex} = -1.0[1.0 \quad 0.0] = [-1.0 \quad 0.0]$
    - $\mathbf{w}(t) = \mathbf{w}(t-1) + \eta\Delta\mathbf{w} = [0.1 \quad 0.1] + 0.1[-1.0 \quad 0.0]$
    - $\mathbf{w}(t) = [0.0 \quad 0.1]$
    - $b_0(t) = b_0(t-1) + \eta e_0 = 0.8 + 0.1 \times (-1.0) = 0.7$
    - $e_{época} = e_{época} + {e_0}^2 = 2.0 + (-1.0)^2 = 3.0$

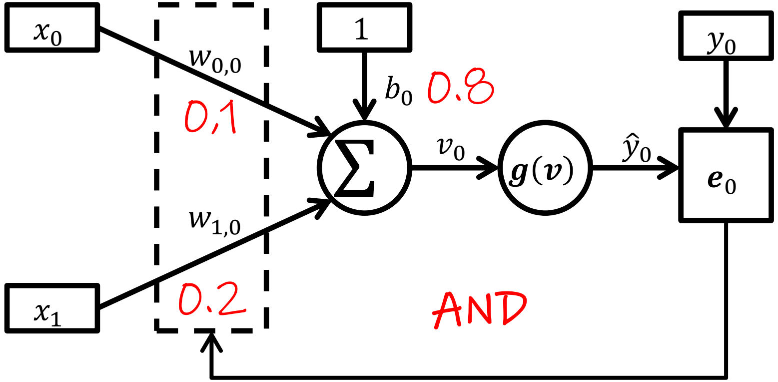
Perceptron de camada simples – exemplo



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

- Época 0:
  - Iteração 2:
    - $\mathbf{x} = [x_0 \quad x_1] = [1.0 \quad 1.0]$
    - $\mathbf{w} = \begin{bmatrix} w_{0,0} \\ w_{1,0} \end{bmatrix} = \begin{bmatrix} 0.0 \\ 0.1 \end{bmatrix}$, $\mathbf{b} = [b_0] = [0.7]$
    - $v_0 = \mathbf{xw} + b_0 = [1.0 \quad 1.0] \begin{bmatrix} 0.0 \\ 0.1 \end{bmatrix} + 0.7 = 0.1 + 0.8 = 0.9$
    - $\hat{y}_0 = 0.0, \quad pois\ v_0 < 0.0$
    - $e_0 = y - \hat{y} = 1.0 - 0.0 = 1.0$
    - $\Delta\mathbf{w} = \mathbf{ex} = 1.0[1.0 \quad 1.0] = [1.0 \quad 1.0]$
    - $\mathbf{w}(t) = \mathbf{w}(t-1) + \eta\Delta\mathbf{w} = [0.0 \quad 0.1] + 0.1[1.0 \quad 1.0]$
    - $\mathbf{w}(t) = [0.1 \quad 0.2]$
    - $b_0(t) = b_0(t-1) + \eta e_0 = 0.7 + 0.1 \times (1.0) = 0.8$
    - $e_{época} = e_{época} + {e_0}^2 = 3.0 + (1.0)^2 = 4.0$
  - $e_{época} = \frac{e_{época}}{2} = 2.0$

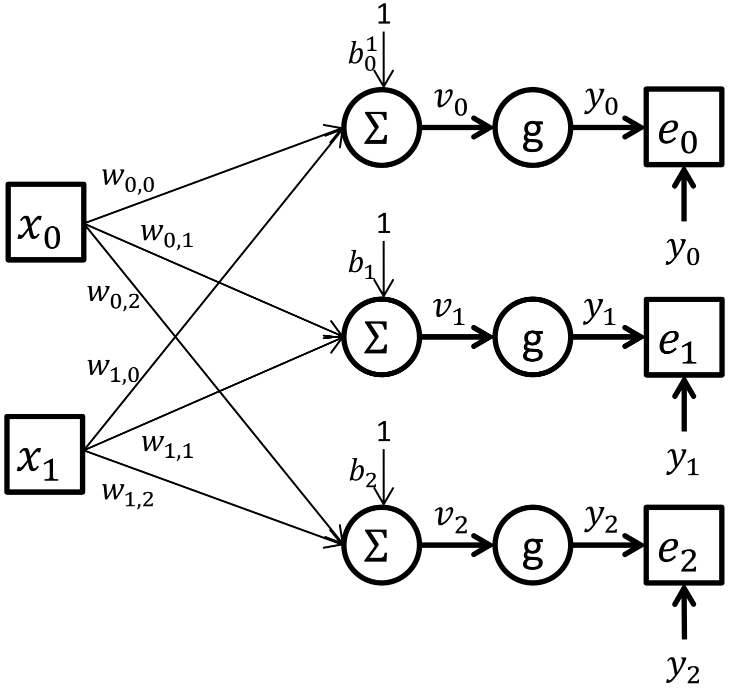
Perceptron de camada simples – exemplo



- Fim da época 0.
- Repetir até atingir o limite de épocas ou o erro da época ficar abaixo de um limiar pré-definido.

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

Perceptron de camada simples – múltiplas classes



C_0	C_1	C_2
1	0	0
0	1	0
0	0	1

Perceptron de múltiplas camadas

NATURE VOL. 323 9 OCTOBER 1986 — LETTERS TO NATURE — 533

Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure[1].

There have been many attempts to design self-organizing neural networks. The aim is to find a powerful synaptic modification rule that will allow an arbitrarily connected neural network to develop an internal structure that is appropriate for a particular task domain. The task is specified by giving the desired state vector of the output units for each state vector of the input units. If the input units are directly connected to the output units it is relatively easy to find learning rules that iteratively adjust the relative strengths of the connections so as to progressively reduce the difference between the actual and desired output vectors[2]. Learning becomes more interesting but more difficult when we introduce hidden units whose actual or desired states are not specified by the task. (In perceptrons, there are 'feature analysers' between the input and output that are not true hidden units because their input connections are fixed by hand, so their states are completely determined by the input vector: they do not learn representations.) The learning procedure must decide under what circumstances the hidden units should be active in order to help achieve the desired input–output behaviour. This amounts to deciding what these units should represent. We demonstrate that a general purpose and relatively simple procedure is powerful enough to construct appropriate internal representations.

The simplest form of the learning procedure is for layered networks which have a layer of input units at the bottom; any number of intermediate layers; and a layer of output units at the top. Connections within a layer or from higher to lower layers are forbidden, but connections can skip intermediate layers. An input vector is presented to the network by setting the states of the input units. Then the states of the units in each layer are determined by applying equations (1) and (2) to the connections coming from lower layers. All units within a layer have their states set in parallel, but different layers have their states set sequentially, starting at the bottom and working upwards until the states of the output units are determined.

The total input, x_j, to unit j is a linear function of the outputs, y_i, of the units that are connected to j and of the weights, w_{ji}, on these connections

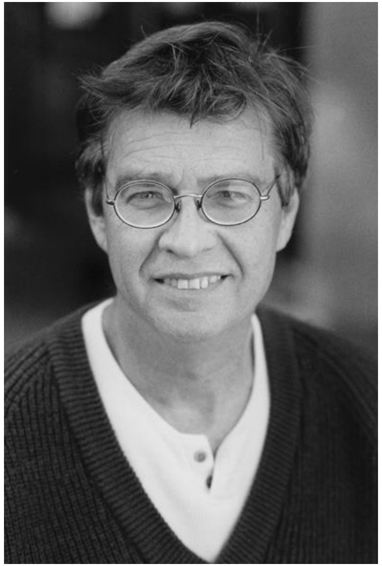
$$x_j = \sum_i y_i w_{ji} \qquad (1)$$

Units can be given biases by introducing an extra input to each unit which always has a value of 1. The weight on this extra input is called the bias and is equivalent to a threshold of the opposite sign. It can be treated just like the other weights.

A unit has a real-valued output, y_j, which is a non-linear function of its total input

$$y_j = \frac{1}{1+e^{-x_j}} \qquad (2)$$

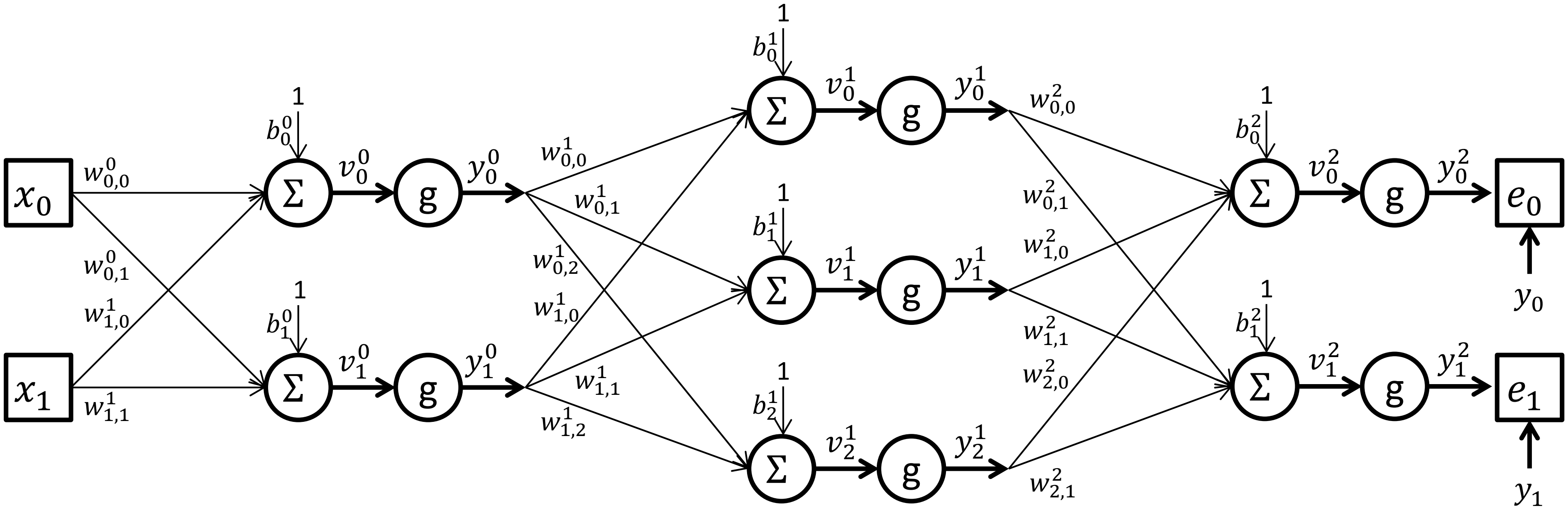
† To whom correspondence should be addressed



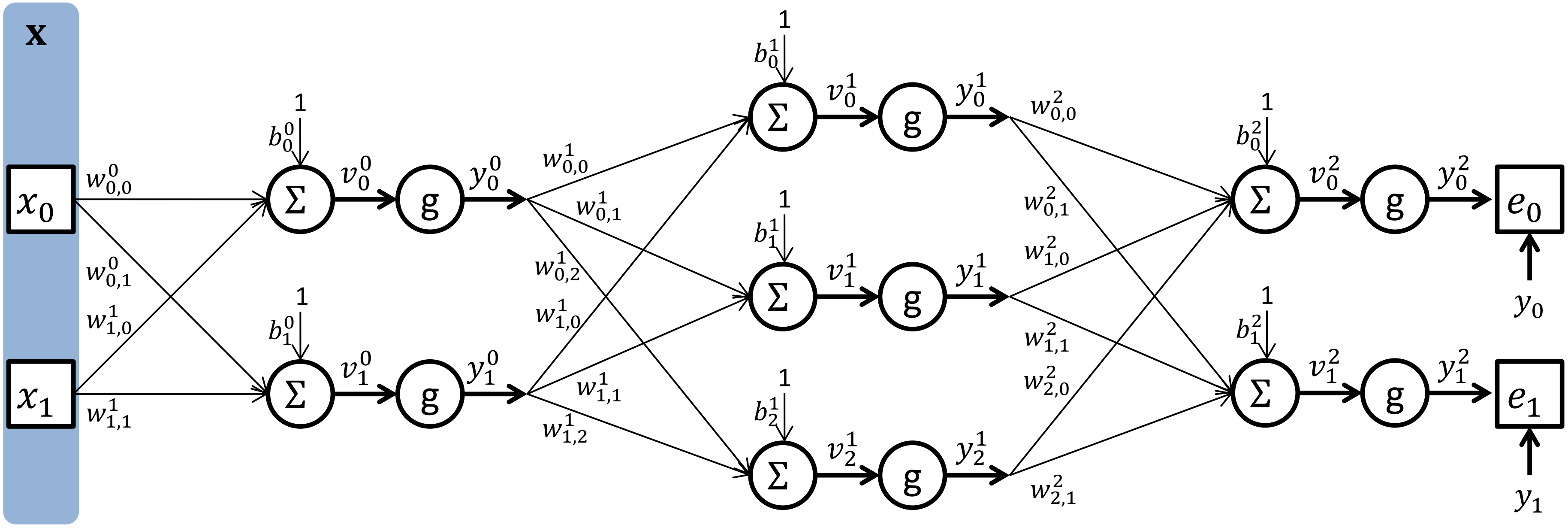
David E. Rumelhart

Perceptron de múltiplas camadas – Estrutura da rede

Perceptron de múltiplas camadas

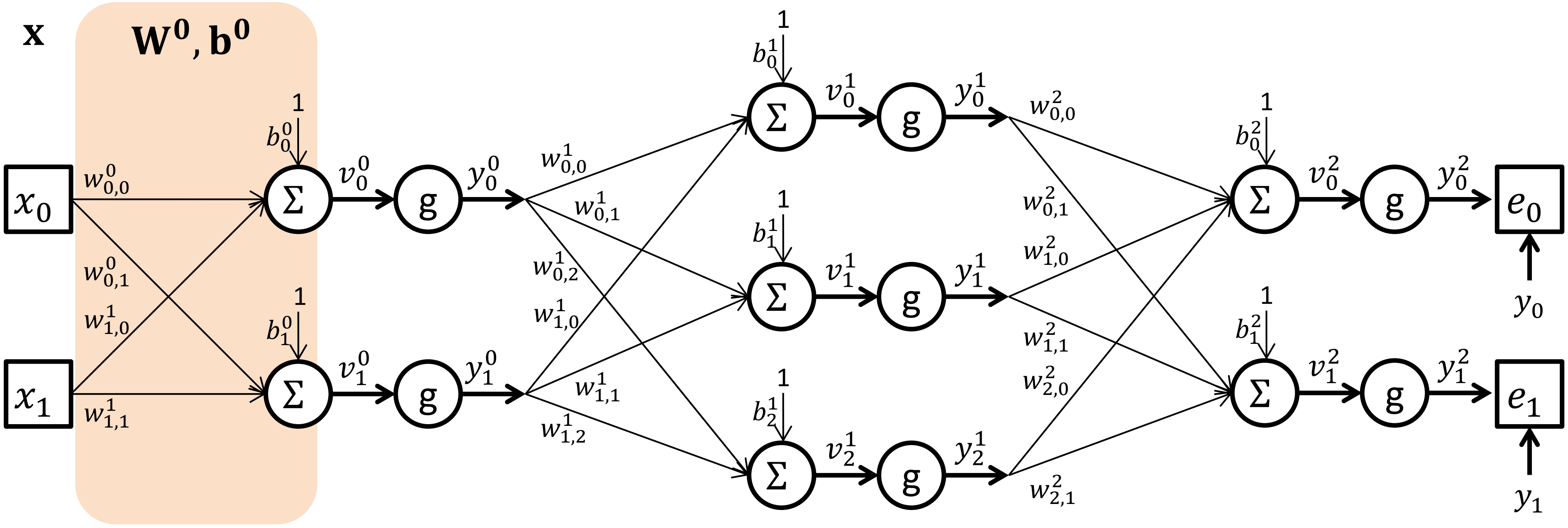


Perceptron de múltiplas camadas



$\mathbf{x} = [x_0 \quad x_1]$

Perceptron de múltiplas camadas

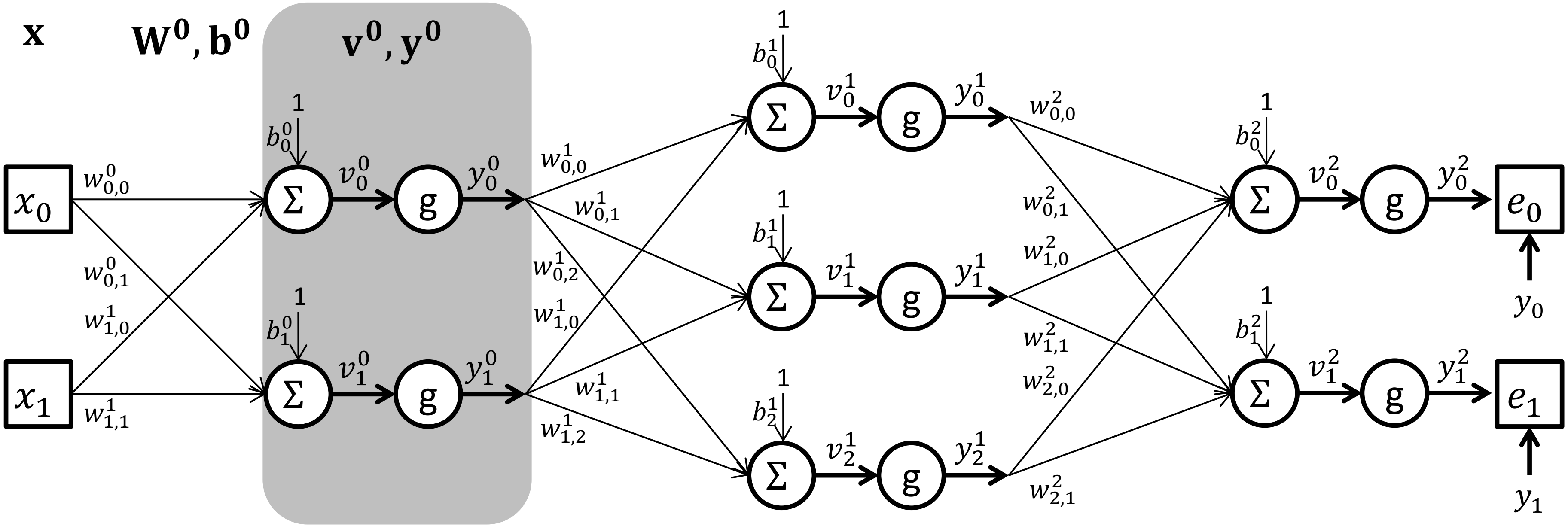


$\mathbf{x} = [x_0 \quad x_1]$

$$\mathbf{W^0} = \begin{bmatrix} w_{0,0}^0 & w_{0,1}^0 \\ w_{1,0}^0 & w_{1,1}^0 \end{bmatrix}$$

$$\mathbf{b^0} = [b_0^0 \quad b_1^0]$$

Perceptron de múltiplas camadas



$$\mathbf{x} = [x_0 \quad x_1]$$

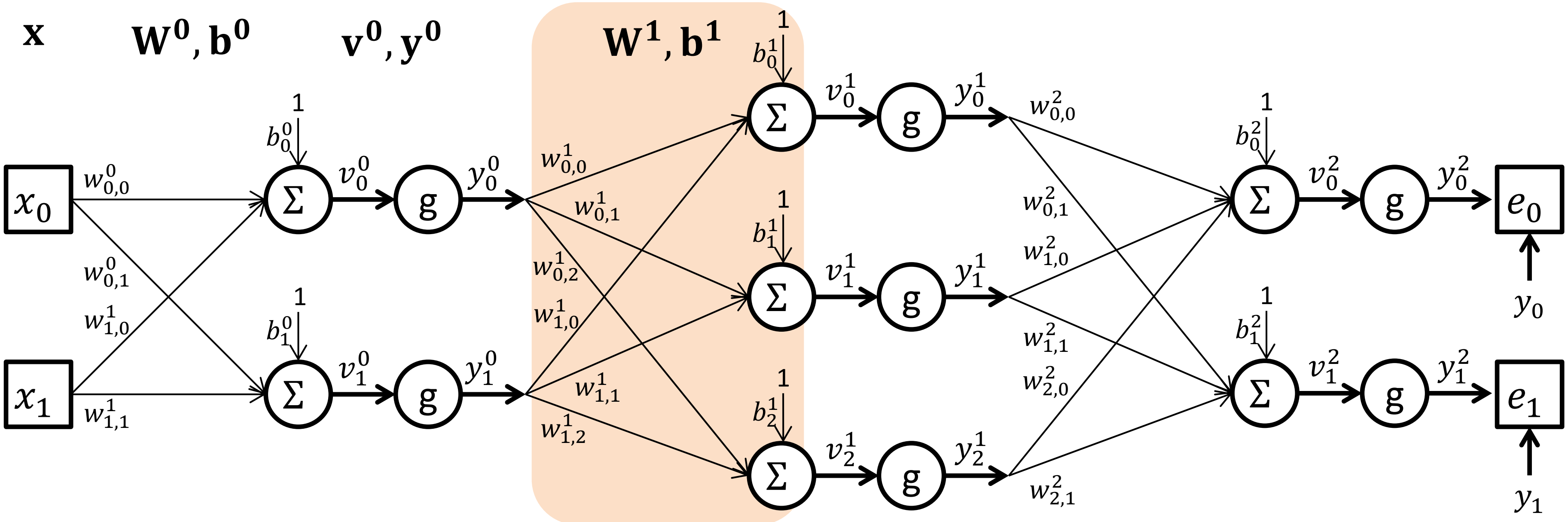
$$\mathbf{v^0} = [v_0^0 \quad v_1^0]$$

$$\mathbf{y^0} = [y_0^0 \quad y_1^0]$$

$$\mathbf{W^0} = \begin{bmatrix} w_{0,0}^0 & w_{0,1}^0 \\ w_{1,0}^0 & w_{1,1}^0 \end{bmatrix}$$

$$\mathbf{b^0} = [b_0^0 \quad b_1^0]$$

Perceptron de múltiplas camadas



$\mathbf{x} = [x_0 \quad x_1]$

$\mathbf{v^0} = [v_0^0 \quad v_1^0]$

$\mathbf{y^0} = [y_0^0 \quad y_1^0]$

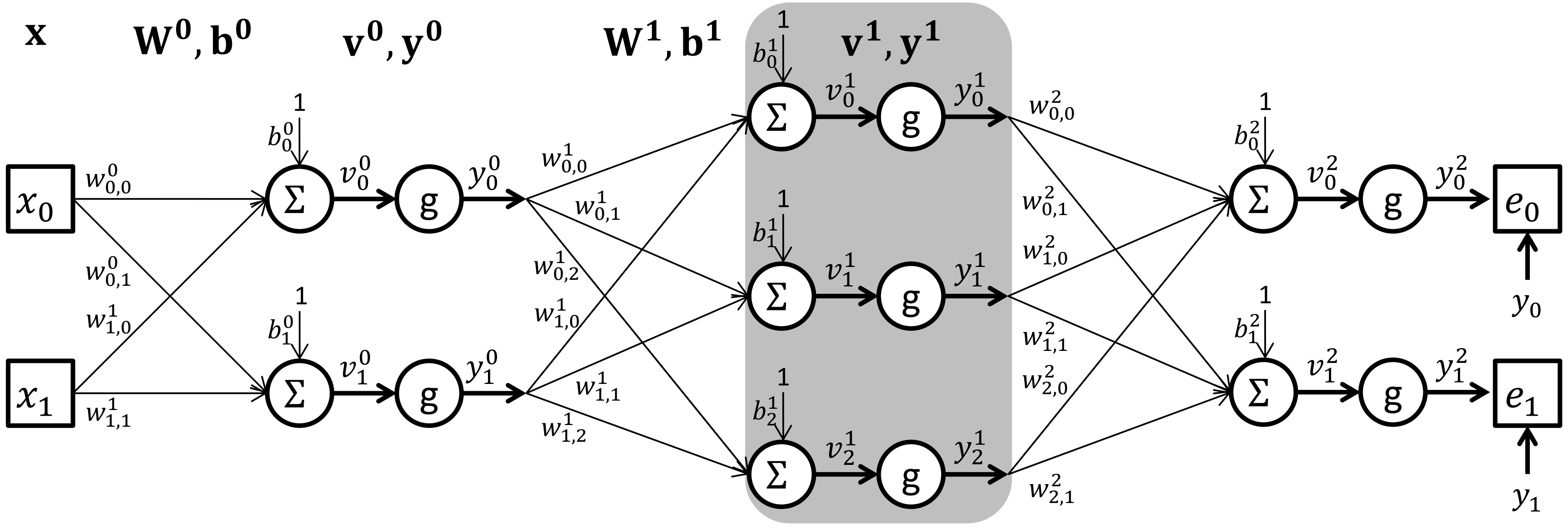
$$\mathbf{W^0} = \begin{bmatrix} w_{0,0}^0 & w_{0,1}^0 \\ w_{1,0}^0 & w_{1,1}^0 \end{bmatrix}$$

$$\mathbf{W^1} = \begin{bmatrix} w_{0,0}^1 & w_{0,1}^1 & w_{0,2}^1 \\ w_{1,0}^1 & w_{1,1}^1 & w_{1,2}^1 \end{bmatrix}$$

$\mathbf{b^0} = [b_0^0 \quad b_1^0]$

$\mathbf{b^1} = [b_0^1 \quad b_1^1 \quad b_2^1]$

Perceptron de múltiplas camadas



$\mathbf{x} = [x_0 \quad x_1]$

$\mathbf{v^0} = [v^0_0 \quad v^0_1]$

$\mathbf{y^0} = [y^0_0 \quad y^0_1]$

$\mathbf{v^1} = [v^1_0 \quad v^1_1 \quad v^1_2]$

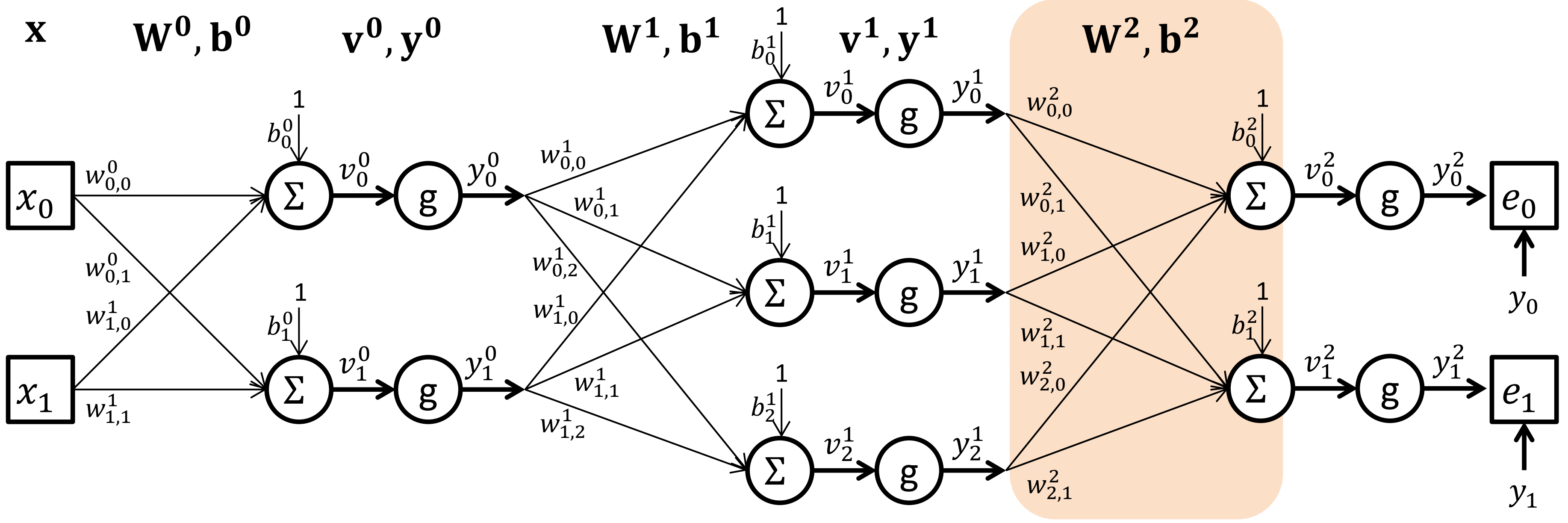
$\mathbf{y^1} = [y^1_0 \quad y^1_1 \quad y^1_2]$

$$\mathbf{W^0} = \begin{bmatrix} w^0_{0,0} & w^0_{0,1} \\ w^0_{1,0} & w^0_{1,1} \end{bmatrix} \qquad \mathbf{W^1} = \begin{bmatrix} w^1_{0,0} & w^1_{0,1} & w^1_{0,2} \\ w^1_{1,0} & w^1_{1,1} & w^1_{1,2} \end{bmatrix}$$

$\mathbf{b^0} = [b^0_0 \quad b^0_1]$

$\mathbf{b^1} = [b^1_0 \quad b^1_1 \quad b^1_2]$

Perceptron de múltiplas camadas



$$\mathbf{x} = [x_0 \quad x_1]$$

$$\mathbf{v^0} = [v^0_0 \quad v^0_1]$$

$$\mathbf{y^0} = [y^0_0 \quad y^0_1]$$

$$\mathbf{v^1} = [v^1_0 \quad v^1_1 \quad v^1_2]$$

$$\mathbf{y^1} = [y^1_0 \quad y^1_1 \quad y^1_2]$$

$$\mathbf{W^0} = \begin{bmatrix} w^0_{0,0} & w^0_{0,1} \\ w^0_{1,0} & w^0_{1,1} \end{bmatrix}$$

$$\mathbf{W^1} = \begin{bmatrix} w^1_{0,0} & w^1_{0,1} & w^1_{0,2} \\ w^1_{1,0} & w^1_{1,1} & w^1_{1,2} \end{bmatrix}$$

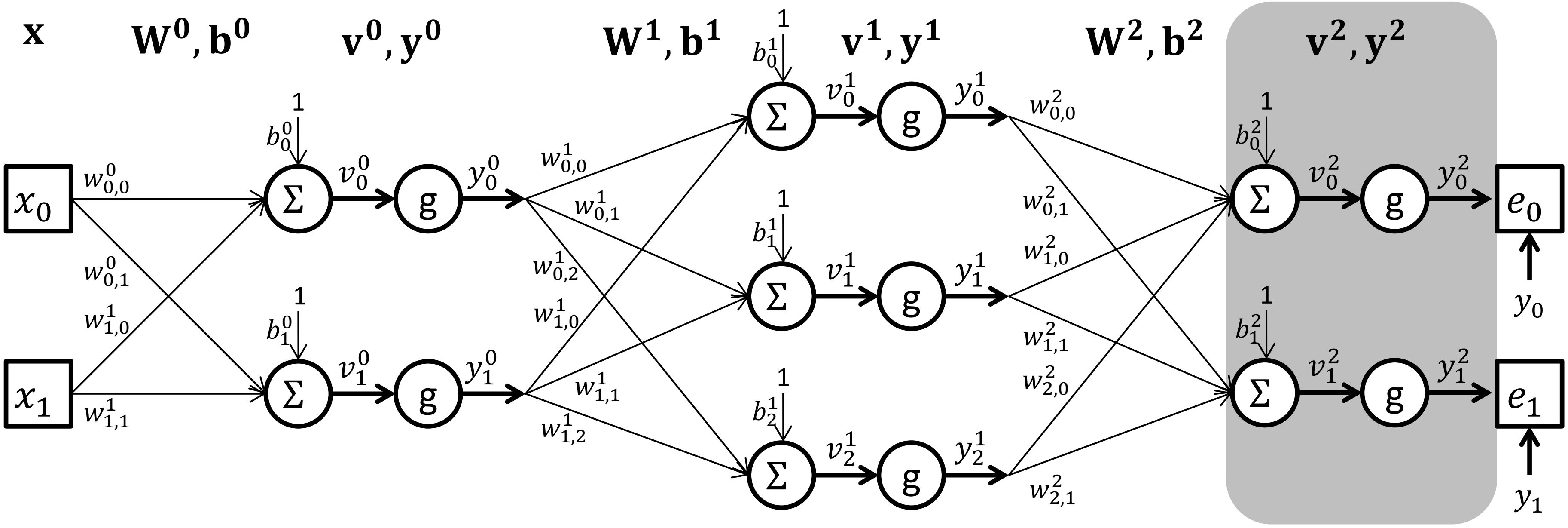
$$\mathbf{W^2} = \begin{bmatrix} w^2_{0,0} & w^2_{0,1} \\ w^2_{1,0} & w^2_{1,1} \\ w^2_{2,0} & w^2_{2,1} \end{bmatrix}$$

$$\mathbf{b^0} = [b^0_0 \quad b^0_1]$$

$$\mathbf{b^1} = [b^1_0 \quad b^1_1 \quad b^1_2]$$

$$\mathbf{b^2} = [b^2_0 \quad b^2_1]$$

Perceptron de múltiplas camadas



$\mathbf{x} = [x_0 \quad x_1]$

$\mathbf{v^0} = [v_0^0 \quad v_1^0]$

$\mathbf{y^0} = [y_0^0 \quad y_1^0]$

$\mathbf{v^1} = [v_0^1 \quad v_1^1 \quad v_2^1]$

$\mathbf{y^1} = [y_0^1 \quad y_1^1 \quad y_2^1]$

$\mathbf{v^2} = [v_0^2 \quad v_1^2]$

$\mathbf{y^2} = [y_0^2 \quad y_1^2]$

$$\mathbf{W^0} = \begin{bmatrix} w_{0,0}^0 & w_{0,1}^0 \\ w_{1,0}^0 & w_{1,1}^0 \end{bmatrix}$$

$$\mathbf{W^1} = \begin{bmatrix} w_{0,0}^1 & w_{0,1}^1 & w_{0,2}^1 \\ w_{1,0}^1 & w_{1,1}^1 & w_{1,2}^1 \end{bmatrix}$$

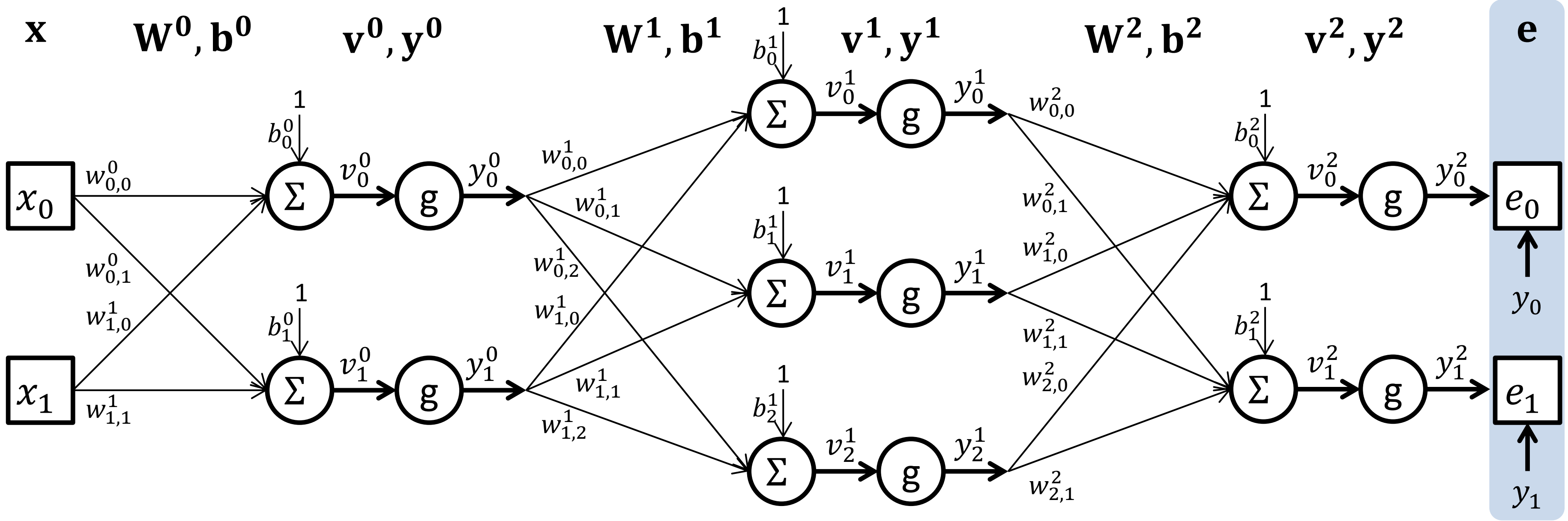
$$\mathbf{W^2} = \begin{bmatrix} w_{0,0}^2 & w_{0,1}^2 \\ w_{1,0}^2 & w_{1,1}^2 \\ w_{2,0}^2 & w_{2,1}^2 \end{bmatrix}$$

$\mathbf{b^0} = [b_0^0 \quad b_1^0]$

$\mathbf{b^1} = [b_0^1 \quad b_1^1 \quad b_2^1]$

$\mathbf{b^2} = [b_0^2 \quad b_1^2]$

Perceptron de múltiplas camadas



$\mathbf{x} = [x_0 \quad x_1]$

$\mathbf{v^0} = [v^0_0 \quad v^0_1]$

$\mathbf{y^0} = [y^0_0 \quad y^0_1]$

$\mathbf{v^1} = [v^1_0 \quad v^1_1 \quad v^1_2]$

$\mathbf{y^1} = [y^1_0 \quad y^1_1 \quad y^1_2]$

$\mathbf{v^2} = [v^2_0 \quad v^2_1]$

$\mathbf{y^2} = [y^2_0 \quad y^2_1]$

$$\mathbf{W^0} = \begin{bmatrix} w^0_{0,0} & w^0_{0,1} \\ w^0_{1,0} & w^0_{1,1} \end{bmatrix}$$

$$\mathbf{W^1} = \begin{bmatrix} w^1_{0,0} & w^1_{0,1} & w^1_{0,2} \\ w^1_{1,0} & w^1_{1,1} & w^1_{1,2} \end{bmatrix}$$

$$\mathbf{W^2} = \begin{bmatrix} w^2_{0,0} & w^2_{0,1} \\ w^2_{1,0} & w^2_{1,1} \\ w^2_{2,0} & w^2_{2,1} \end{bmatrix}$$

$\mathbf{y} = [y_0 \quad y_1]$

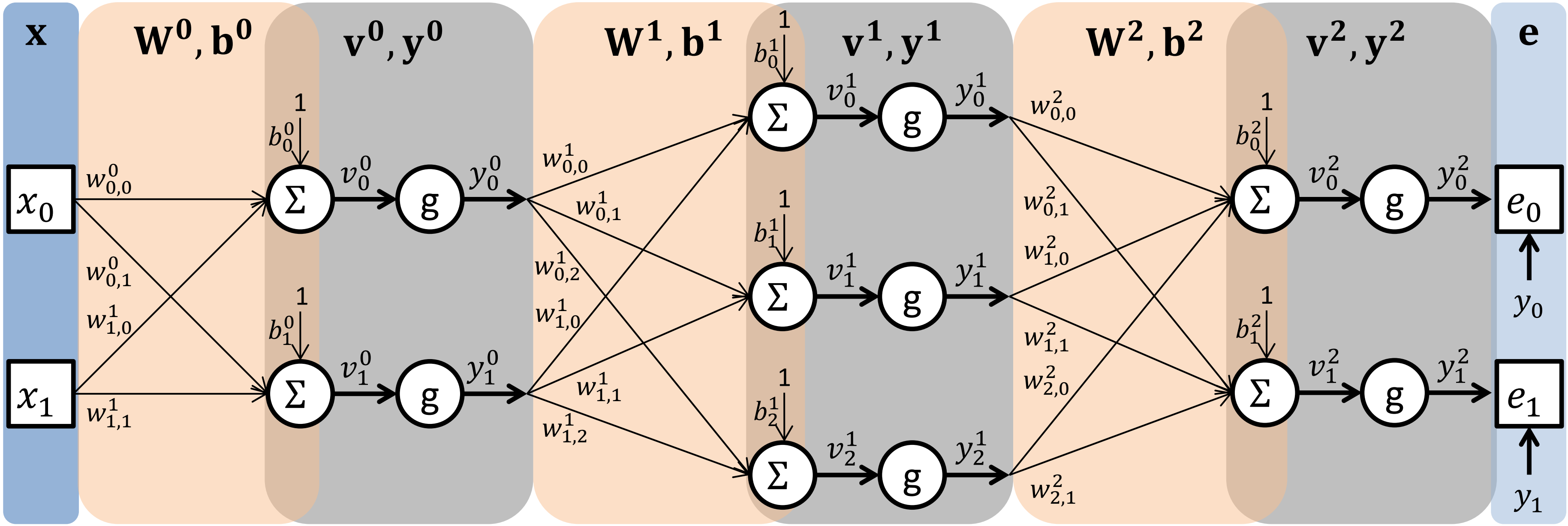
$\mathbf{b^0} = [b^0_0 \quad b^0_1]$

$\mathbf{b^1} = [b^1_0 \quad b^1_1 \quad b^1_2]$

$\mathbf{b^2} = [b^2_0 \quad b^2_1]$

$\mathbf{e} = [e_0 \quad e_1]$

Perceptron de múltiplas camadas



$\mathbf{x} = [x_0 \quad x_1]$

$\mathbf{v^0} = [v_0^0 \quad v_1^0]$

$\mathbf{y^0} = [y_0^0 \quad y_1^0]$

$\mathbf{v^1} = [v_0^1 \quad v_1^1 \quad v_2^1]$

$\mathbf{y^1} = [y_0^1 \quad y_1^1 \quad y_2^1]$

$\mathbf{v^2} = [v_0^2 \quad v_1^2]$

$\mathbf{y^2} = [y_0^2 \quad y_1^2]$

$$\mathbf{W^0} = \begin{bmatrix} w_{0,0}^0 & w_{0,1}^0 \\ w_{1,0}^0 & w_{1,1}^0 \end{bmatrix}$$

$$\mathbf{W^1} = \begin{bmatrix} w_{0,0}^1 & w_{0,1}^1 & w_{0,2}^1 \\ w_{1,0}^1 & w_{1,1}^1 & w_{1,2}^1 \end{bmatrix}$$

$$\mathbf{W^2} = \begin{bmatrix} w_{0,0}^2 & w_{0,1}^2 \\ w_{1,0}^2 & w_{1,1}^2 \\ w_{2,0}^2 & w_{2,1}^2 \end{bmatrix}$$

$\mathbf{y} = [y_0 \quad y_1]$

$\mathbf{b^0} = [b_0^0 \quad b_1^0]$

$\mathbf{b^1} = [b_0^1 \quad b_1^1 \quad b_2^1]$

$\mathbf{b^2} = [b_0^2 \quad b_1^2]$

$\mathbf{e} = [e_0 \quad e_1]$

Backpropagation – Algoritmo e equações

Backpropagation – equações

Função ReLu $\mathbf{g}^0(\mathbf{v}^0) = max(0, \mathbf{v}^0)$

Função Sigmoide $\mathbf{g}^1(\mathbf{v}^1) = \dfrac{1}{1 + e^{-\mathbf{v}^1}}$

Função Softmax $\mathbf{g}^2(\mathbf{v}^2) = \dfrac{e^{\mathbf{v}^2}}{\sum_{i=0}^{n} e^{\mathbf{v}_i^2}}$

$$\mathbf{v}^0 = \mathbf{x}\mathbf{W}^0 + \mathbf{b}^0 \rightarrow \mathbf{y}^0 = \mathbf{g}^0(\mathbf{v}^0) \rightarrow \mathbf{v}^1 = \mathbf{y}^0\mathbf{W}^1 + \mathbf{b}^1 \rightarrow \mathbf{y}^1 = \mathbf{g}^1(\mathbf{v}^1) \rightarrow \mathbf{v}^2 = \mathbf{y}^1\mathbf{W}^2 + \mathbf{b}^2 \rightarrow \mathbf{y}^2 = \mathbf{g}^2(\mathbf{v}^2)$$

Função entropia cruzada $\boldsymbol{e}(\mathbf{y}^2, \mathbf{y}) = -1 \times \left(\mathbf{y} \times log(\mathbf{y}^2) + (1 - \mathbf{y}) \times log(1 - \mathbf{y}^2)\right)$

$$\boldsymbol{e} = \boldsymbol{e}(\mathbf{y}^2, \mathbf{y})$$

$$\mathbf{W}'^2 = \mathbf{W}^2 - \eta \frac{\partial \mathbf{e}}{\partial \mathbf{W}^2} \leftarrow \mathbf{W}'^1 = \mathbf{W}^1 - \eta \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^1} \leftarrow \mathbf{W}'^0 = \mathbf{W}^0 - \eta \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^0}$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^0} = \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^0} \times \frac{\partial \mathbf{y}^0}{\partial \mathbf{v}^0} \times \frac{\partial \mathbf{v}^0}{\partial \mathbf{W}^0}$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^0} = \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^1} \times \frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} \times \frac{\partial \mathbf{v}^1}{\partial \mathbf{y}^0} \qquad \frac{\partial \mathbf{v}^1}{\partial \mathbf{y}^0} = \mathbf{W}^1$$

$$\frac{\partial \mathbf{y}^0}{\partial \mathbf{v}^0} = \begin{cases} 1, & se\ x > 0 \\ 0, & caso\ contrário \end{cases}$$

$$\frac{\partial \mathbf{v}^0}{\partial \mathbf{W}^0} = \frac{\partial(\mathbf{x}\mathbf{W}^0 + \mathbf{b}^0)}{\partial \mathbf{W}^0} = \mathbf{x}$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^1} = \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^1} \times \frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} \times \frac{\partial \mathbf{v}^1}{\partial \mathbf{W}^1}$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^1} = \frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} \times \frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} \times \frac{\partial \mathbf{v}^2}{\partial \mathbf{y}^1} \qquad \frac{\partial \mathbf{v}^2}{\partial \mathbf{y}^1} = \mathbf{W}^2$$

$$\frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} = \frac{1}{1 + e^{-\mathbf{v}^1}} \times 1 - \frac{1}{1 + e^{-\mathbf{v}^1}}$$

$$\frac{\partial \mathbf{v}^1}{\partial \mathbf{W}^1} = \frac{\partial(\mathbf{y}^0\mathbf{W}^1 + \mathbf{b}^1)}{\partial \mathbf{W}^1} = \mathbf{y}^0$$

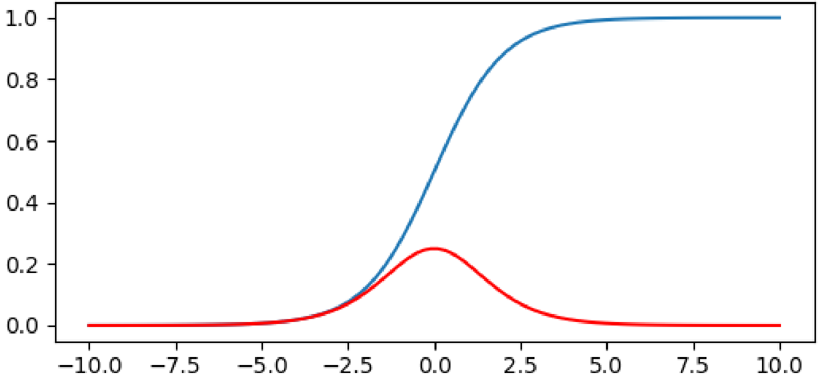
$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^2} = \frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} \times \frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} \times \frac{\partial \mathbf{v}^2}{\partial \mathbf{W}^2}$$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} = -1 \times \left(\mathbf{y}^2 \times \frac{1}{\mathbf{y}} + (1 - \mathbf{y}^2) \times \frac{1}{1 - \mathbf{y}^2}\right)$$

$$\frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} = \frac{e^{\mathbf{v}_i^2} \times \left(\sum_{j=0}^{n-1} e^{\mathbf{v}_j^2} - e^{\mathbf{v}_i^2}\right)}{\sum_{j=0}^{n-1} e^{\mathbf{v}_j^2}} \quad para\ i = 0,1, \dots, n-1$$

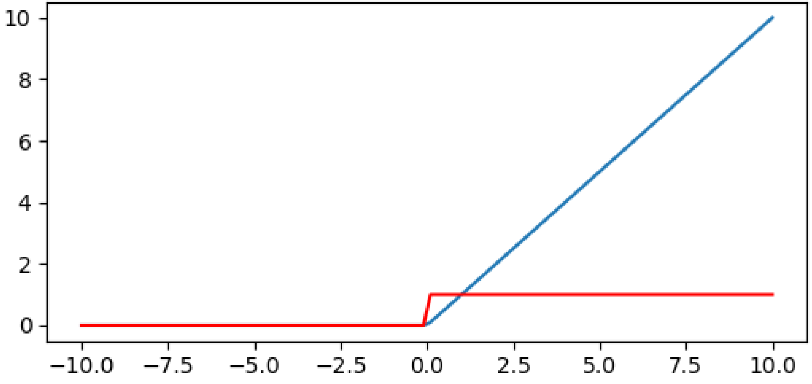
$$\frac{\partial \mathbf{v}^2}{\partial \mathbf{W}^2} = \frac{\partial(\mathbf{y}^1\mathbf{W}^2 + \mathbf{b}^2)}{\partial \mathbf{W}^2} = \mathbf{y}^1$$

Backpropagation – funções de ativação



$$\mathbf{g}(v) = \frac{1}{1 + e^{-\mathbf{v}^1}}$$

Função sigmoide (azul) e sua derivada (vermelho).

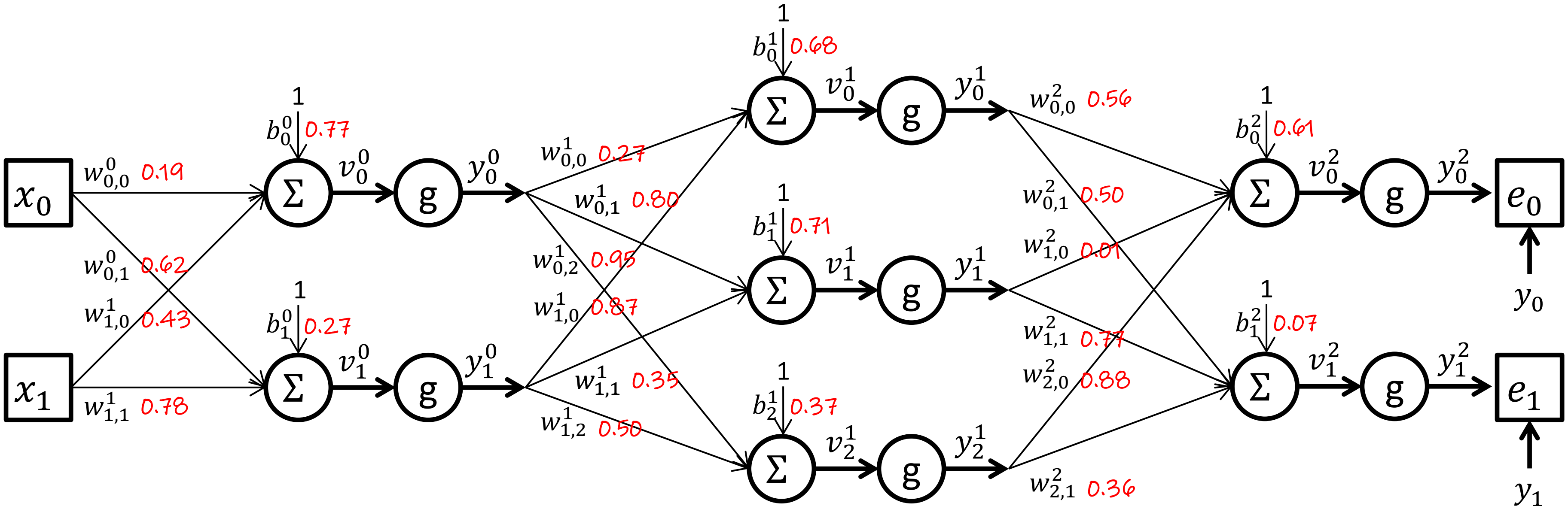


$$\mathbf{g}(\mathbf{v}) = max(0, \mathbf{v})$$

Função ReLu (azul) e sua derivada (vermelho).

Backpropagation – Exemplo

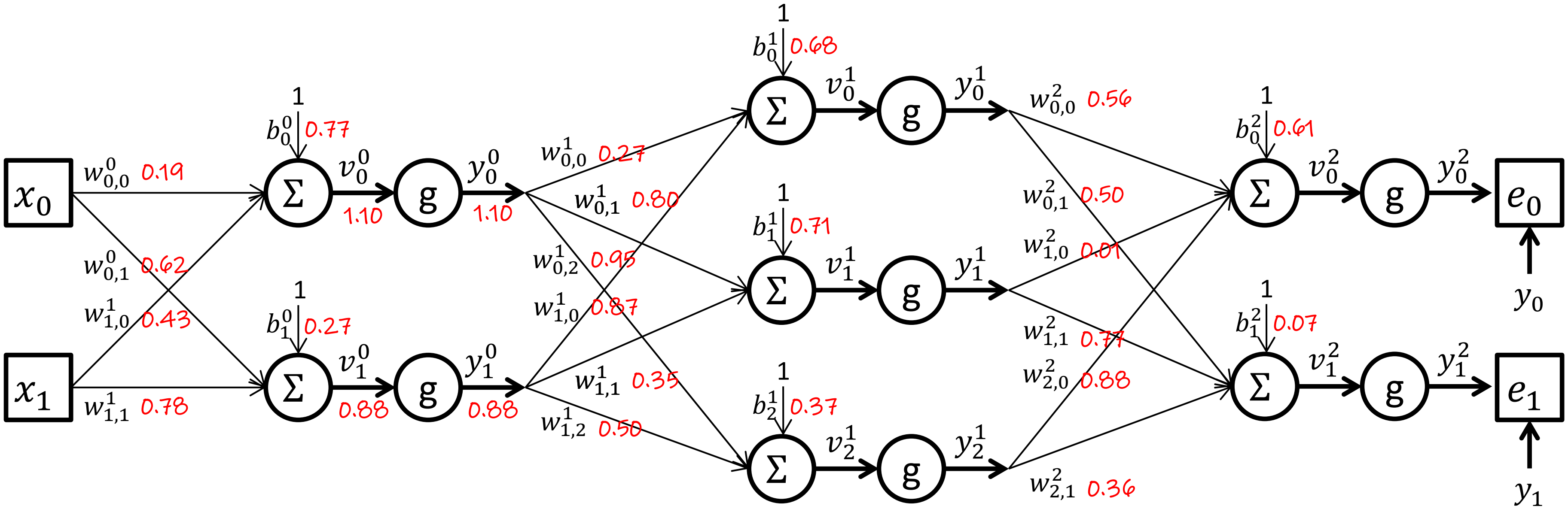
Inicialização dos pesos e bias



$$\mathbf{W^0} = \begin{bmatrix} w^0_{0,0} & w^0_{0,1} \\ w^0_{1,0} & w^0_{1,1} \end{bmatrix} = \begin{bmatrix} 0.19 & 0.62 \\ 0.43 & 0.78 \end{bmatrix} \qquad \mathbf{W^1} = \begin{bmatrix} w^1_{0,0} & w^1_{0,1} & w^1_{0,2} \\ w^1_{0,0} & w^1_{0,0} & w^1_{1,1} \end{bmatrix} = \begin{bmatrix} 0.27 & 0.80 & 0.95 \\ 0.87 & 0.35 & 0.50 \end{bmatrix} \qquad \mathbf{W^2} = \begin{bmatrix} w^2_{0,0} & w^2_{0,1} \\ w^2_{1,0} & w^2_{1,1} \\ w^2_{2,0} & w^2_{2,1} \end{bmatrix} = \begin{bmatrix} 0.56 & 0.50 \\ 0.01 & 0.77 \\ 0.88 & 0.36 \end{bmatrix}$$

$$\mathbf{b^0} = [b^0_0 \quad b^0_1] = [0.77 \quad 0.27] \qquad \mathbf{b^1} = [b^1_0 \quad b^1_1 \quad b^1_2] = [0.68 \quad 0.71 \quad 0.37] \qquad \mathbf{b^2} = [b^2_0 \quad b^2_1] = [0.61 \quad 0.07]$$

Propagação adiante (*forward*) – camada 0 (camada escondida)



$\mathbf{x} = [x_0 \quad x_1] = [0.10 \quad 0.70]$

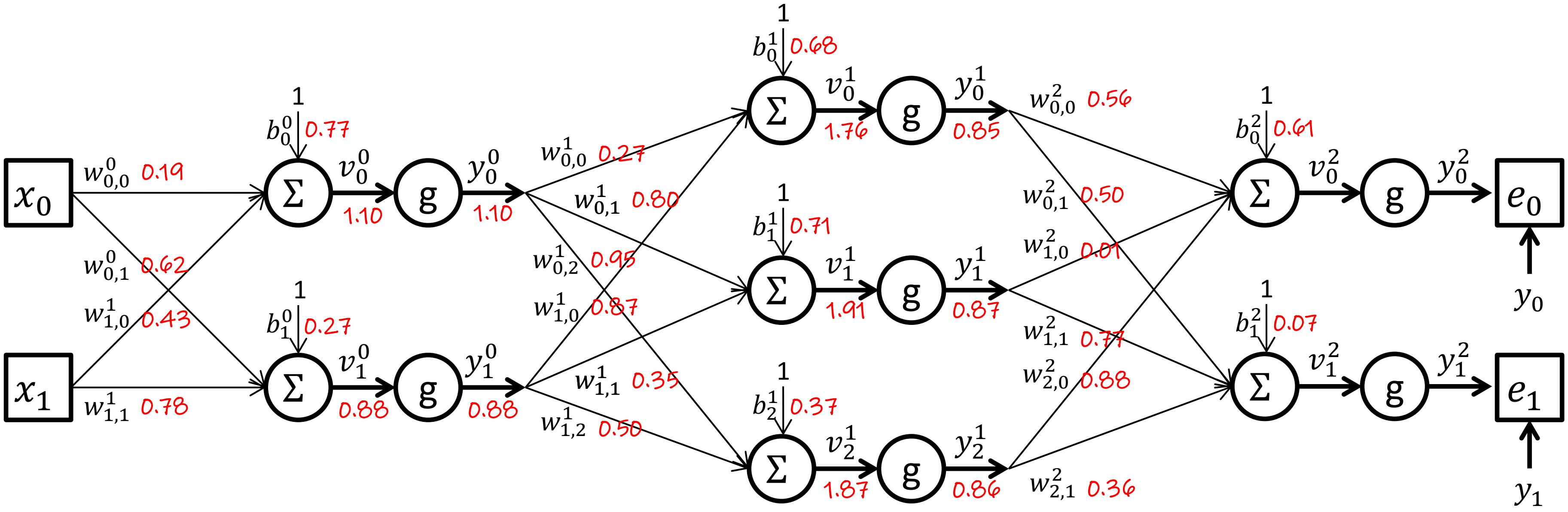
$\mathbf{v}^0 = \mathbf{x}\mathbf{W}^0 + \mathbf{b}^0$

$$\mathbf{v}^0 = [0.10 \quad 0.70]\begin{bmatrix} 0.19 & 0.62 \\ 0.43 & 0.78 \end{bmatrix} + [0.77 \quad 0.27] = [1.10 \quad 0.88]$$

$$\mathbf{y}^0 = \mathbf{g}^0(\mathbf{v}^0) = max(0, \mathbf{v}^0)$$

$$\mathbf{y}^0 = \mathbf{g}^0(\mathbf{v}^0) = [1.10 \quad 0.88]$$

Propagação adiante (*forward*) – camada 1 (camada escondida)



$$\mathbf{y}^0 = \mathbf{g}^0(\mathbf{v}^0) = [1.10 \quad 0.88]$$

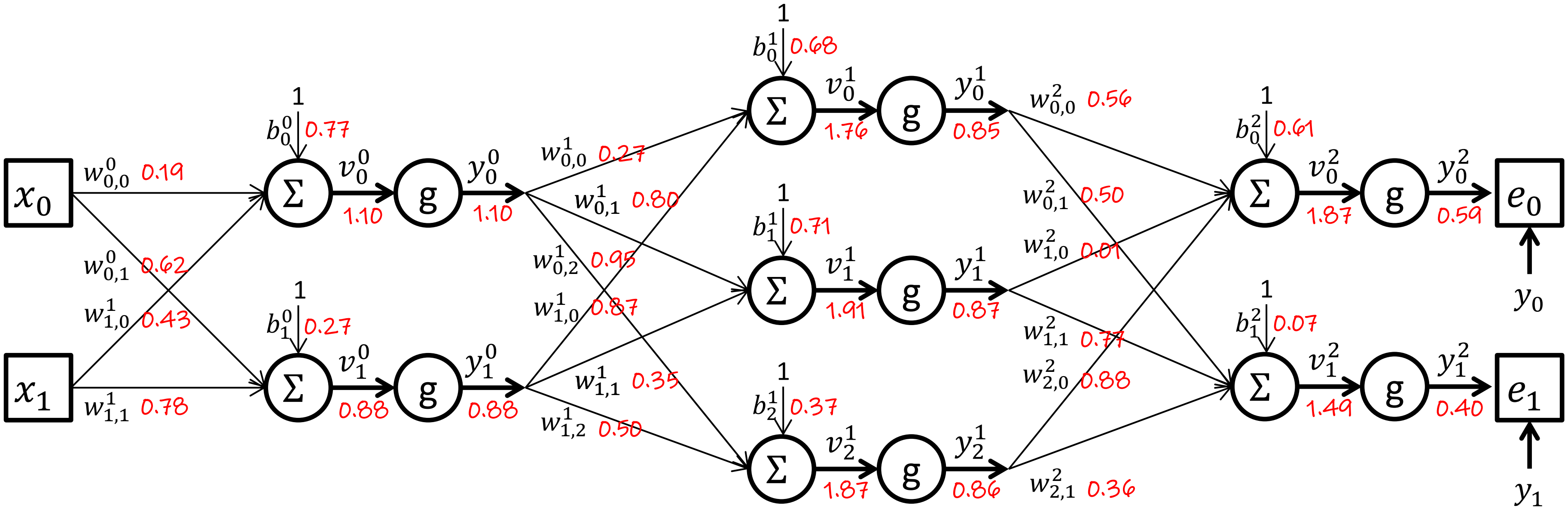
$$\mathbf{v}^1 = \mathbf{x}\mathbf{W}^1 + \mathbf{b}^1$$

$$\mathbf{v}^1 = [1.10 \quad 0.88]\begin{bmatrix} 0.27 & 0.80 & 0.95 \\ 0.87 & 0.35 & 0.50 \end{bmatrix} + [0.68 \quad 0.71 \quad 0.37] = [1.76 \quad 1.91 \quad 1.87]$$

$$\mathbf{y}^1 = \mathbf{g}^1(\mathbf{v}^1) = \frac{1}{1 + e^{-\mathbf{v}^1}}$$

$$\mathbf{y}^1 = \mathbf{g}^1(\mathbf{v}^1) = [0.85 \quad 0.87 \quad 0.86]$$

Propagação adiante (*forward*) – camada 2 (camada de saída)



$$\mathbf{y}^1 = \mathbf{g}^1(\mathbf{v}^1) = [0.85 \quad 0.87 \quad 0.86] \qquad \mathbf{v}^2 = \mathbf{x}\mathbf{W}^2 + \mathbf{b}^2$$

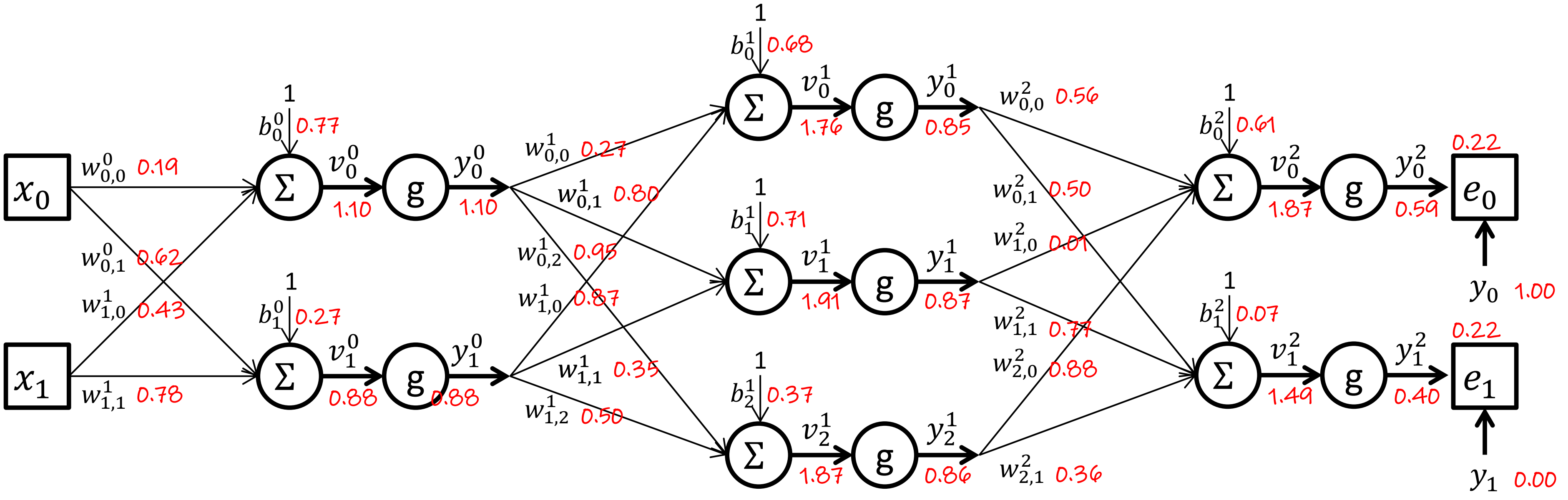
$$\mathbf{v}^2 = [0.85 \quad 0.87 \quad 0.86]\begin{bmatrix} 0.56 & 0.50 \\ 0.01 & 0.77 \\ 0.88 & 0.36 \end{bmatrix} + [0.61 \quad 0.07] = [1.87 \quad 1.49]$$

$$\mathbf{y}^2 = \mathbf{g}^2(\mathbf{v}^2) = \frac{e^{\mathbf{v}^2}}{\sum_{i=0}^{n} e^{\mathbf{v}_i^2}}$$

$$\mathbf{y}^2 = \mathbf{g}^2(\mathbf{v}^2) = [0.59 \quad 0.40]$$

Retropropagação (*backward*)
Erro

Propagação adiante (*forward*) – erro



$$\mathbf{y}^2 = \mathbf{g}^2(\mathbf{v}^2) = [0.59 \quad 0.40]$$

$$\mathbf{y} = [1.00 \quad 0.00]$$

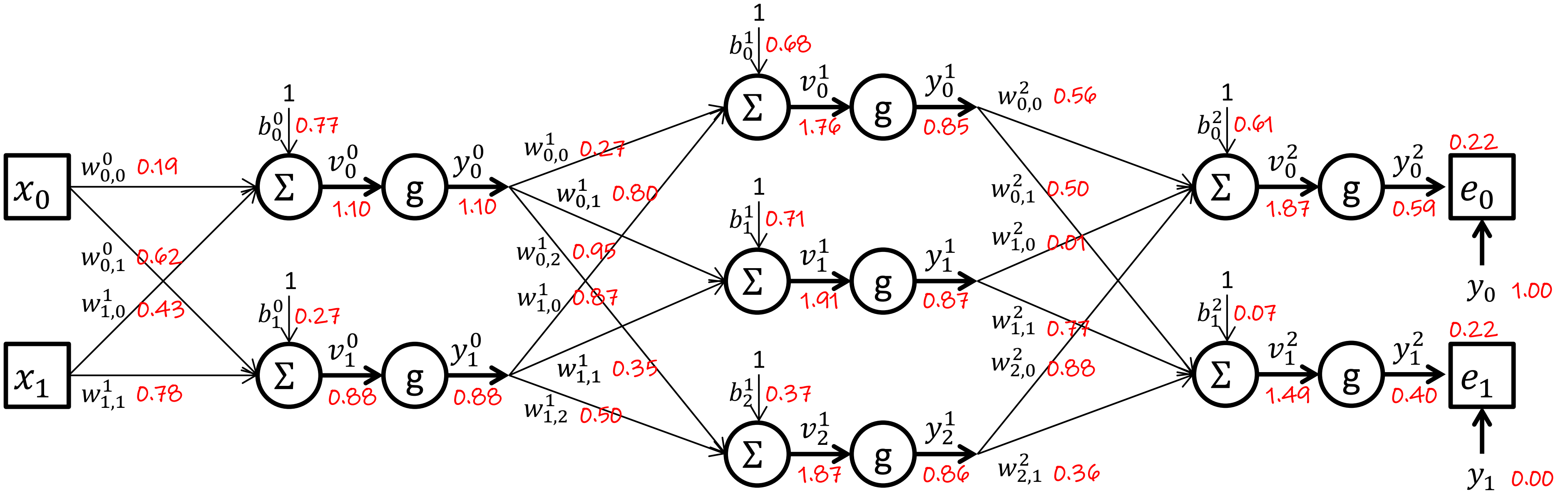
$$\boldsymbol{e} = \boldsymbol{e}(\mathbf{y}^2, \mathbf{y}) = -1 \times \left(\mathbf{y} \times log(\mathbf{y}^2) + (1 - \mathbf{y}) \times log(1 - \mathbf{y}^2)\right)$$

$$\boldsymbol{e} = \boldsymbol{e}(\mathbf{y}^2, \mathbf{y}) = -1 \times \left([1.00 \quad 0.00] \times log([0.59 \quad 0.40]) + (1 - [1.00 \quad 0.00]) \times log(1 - [0.59 \quad 0.40])\right)$$

$$\boldsymbol{e} = \boldsymbol{e}(\mathbf{y}^2, \mathbf{y}) = [0.22 \quad 0.22]$$

Retropropagação (*backward*) Camada 2 (camada de saída)

Retropropagação (*backward*) – camada 2 (camada de saída)



$$\mathbf{y}^2 = [0.59 \quad 0.40]$$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^2} = \frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} \times \frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} \times \frac{\partial \mathbf{v}^2}{\partial \mathbf{W}^2}$$

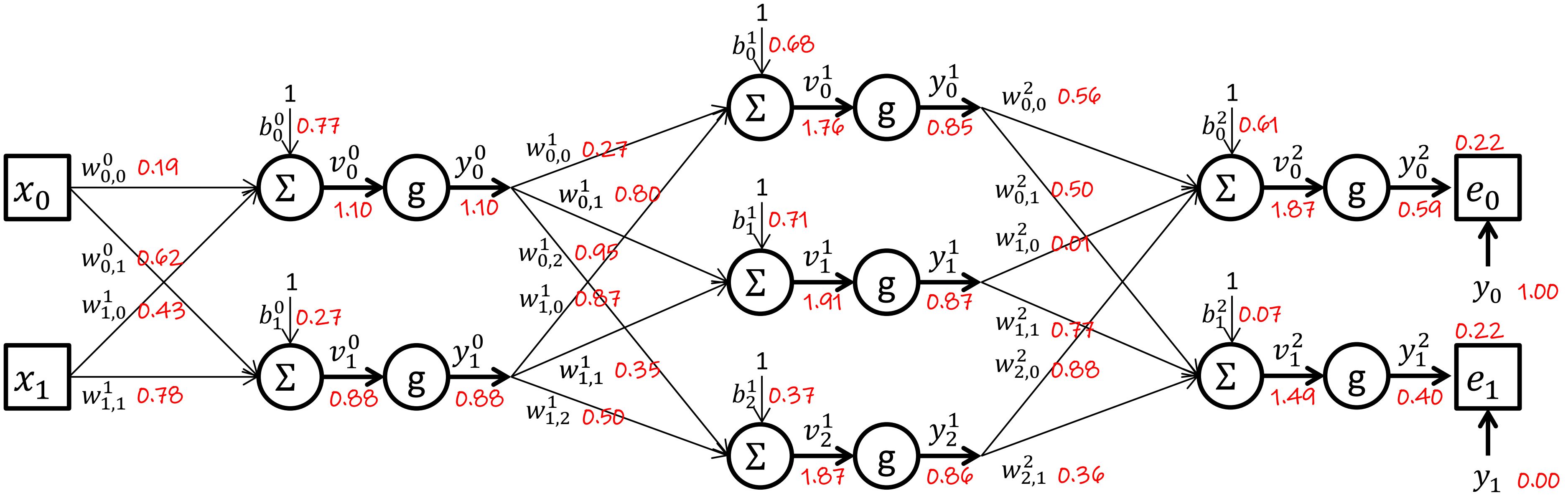
$$\mathbf{y} = [1.00 \quad 0.00]$$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} = -1 \times \left(\mathbf{y} \times \frac{1}{\mathbf{y}^2} + (1-\mathbf{y}) \times \frac{1}{1-\mathbf{y}^2}\right) = -1 \times \left(\frac{[1.00 \quad 0.00]}{[0.59 \quad 0.40]} + \frac{(1-[1.00 \quad 0.00])}{(1-[0.59 \quad 0.40])}\right)$$

$$\boldsymbol{e} = [0.22 \quad 0.22]$$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} = [-1.68 \quad -1.68]$$

Retropropagação (*backward*) – camada 2 (camada de saída)



$\mathbf{y}^2 = [0.59 \quad 0.40]$

$\mathbf{y} = [1.00 \quad 0.00]$

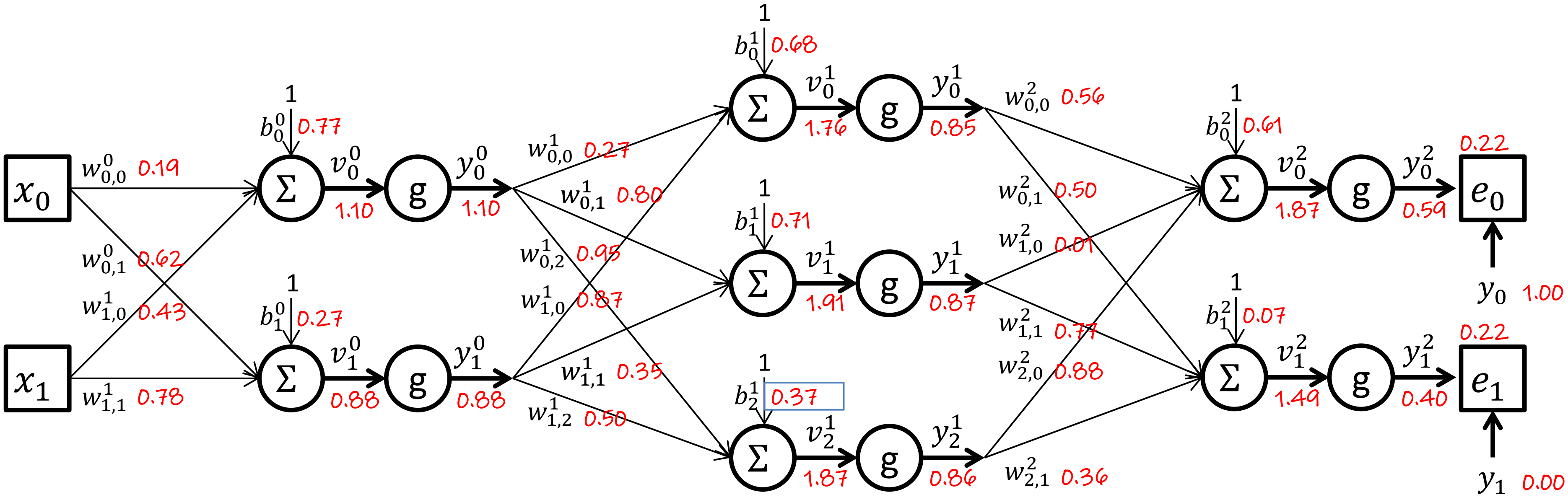
$\boldsymbol{e} = [0.22 \quad 0.22]$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^2} = \frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} \times \frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} \times \frac{\partial \mathbf{v}^2}{\partial \mathbf{W}^2}$$

$$\frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} = \frac{e^{\mathbf{v}_i^2} \times \left(\sum_{j=0}^{n-1} e^{\mathbf{v}_j^2} - e^{\mathbf{v}_i^2}\right)}{\sum_{j=0}^{n-1} e^{\mathbf{v}_j^2}} \; para\ i = 0,1,\dots,n-1 = \left[\frac{e^{v_0^2} \times \left(e^{v_1^2}\right)}{e^{v_0^2} + e^{v_1^2}} \quad \frac{e^{v_1^2} \times \left(e^{v_0^2}\right)}{e^{v_0^2} + e^{v_1^2}}\right]$$

$$\frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} = [0.24 \quad 0.24]$$

Retropropagação (*backward*) – camada 2 (camada de saída)



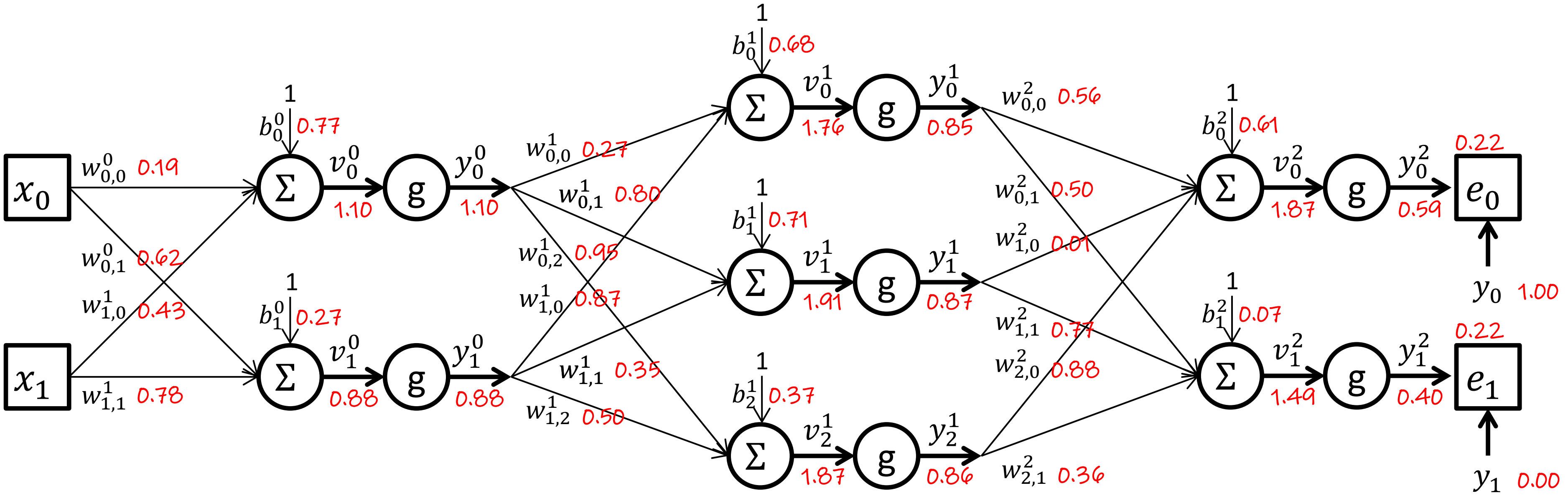
$$\mathbf{y}^1 = [0.85 \quad 0.87 \quad 0.86]$$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^2} = \frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} \times \frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} \times \frac{\partial \mathbf{v}^2}{\partial \mathbf{W}^2}$$

$$\frac{\partial \mathbf{v}^2}{\partial \mathbf{W}^2} = \frac{\partial (\mathbf{y}^1\mathbf{W}^2 + \mathbf{b}^2)}{\partial \mathbf{W}^2} = \left[{\mathbf{y}^1}^t \quad {\mathbf{y}^1}^t\right]$$

$$\frac{\partial \mathbf{v}^2}{\partial \mathbf{W}^2} = \begin{bmatrix} 0.85 & 0.85 \\ 0.87 & 0.87 \\ 0.86 & 0.86 \end{bmatrix}$$

Retropropagação (*backward*) – camada 2 (camada de saída)



$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} = \begin{bmatrix}-1.68 & -1.68\end{bmatrix}$$

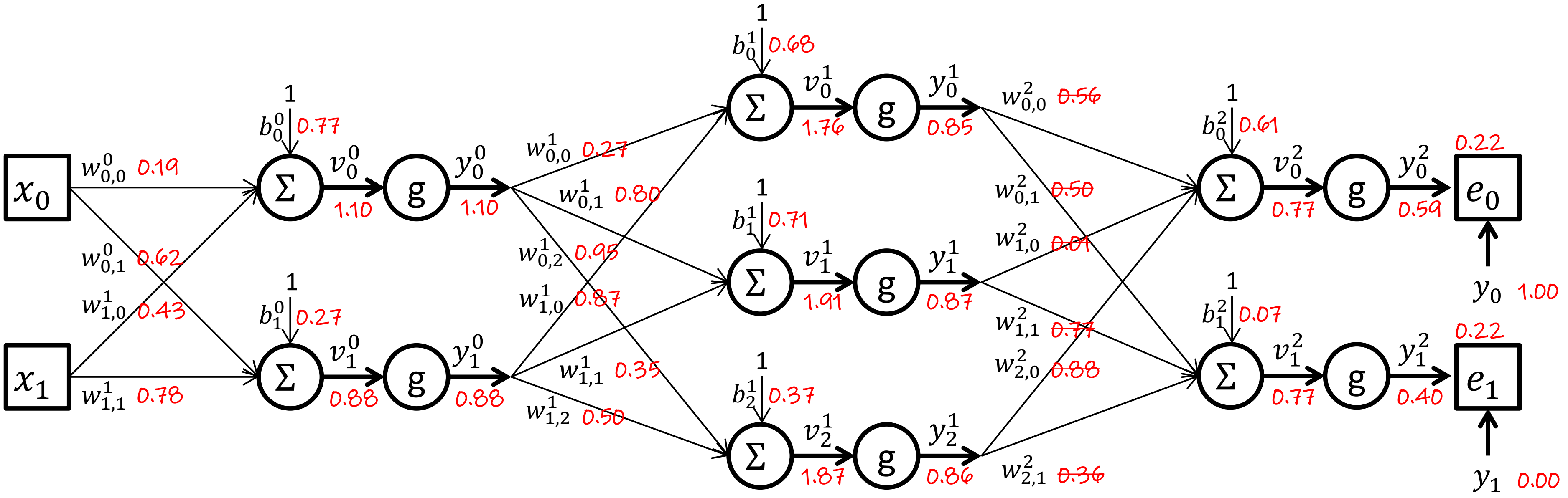
$$\frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} = \begin{bmatrix}0.24 & 0.24\end{bmatrix}$$

$$\frac{\partial \mathbf{v}^2}{\partial \mathbf{W}^2} = \begin{bmatrix}0.85 & 0.85\\ 0.87 & 0.87\\ 0.86 & 0.86\end{bmatrix}$$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^2} = \frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} \times \frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} \times \frac{\partial \mathbf{v}^2}{\partial \mathbf{W}^2} = \begin{bmatrix}-1.68 & -1.68\end{bmatrix} \times \begin{bmatrix}0.24 & 0.24\end{bmatrix} \times \begin{bmatrix}0.85 & 0.85\\ 0.87 & 0.87\\ 0.86 & 0.86\end{bmatrix}$$

$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^2} = \frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} \times \frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} \times \frac{\partial \mathbf{v}^2}{\partial \mathbf{W}^2} = \begin{bmatrix}-0.34 & -0.34\\ -0.35 & -0.35\\ -0.35 & -0.35\end{bmatrix}$$

Retropropagação (*backward*) – camada 2 (camada de saída)



$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^2} = \begin{bmatrix} -0.34 & -0.34 \\ -0.35 & -0.35 \\ -0.35 & -0.35 \end{bmatrix}$$

$$\eta = 0.1$$

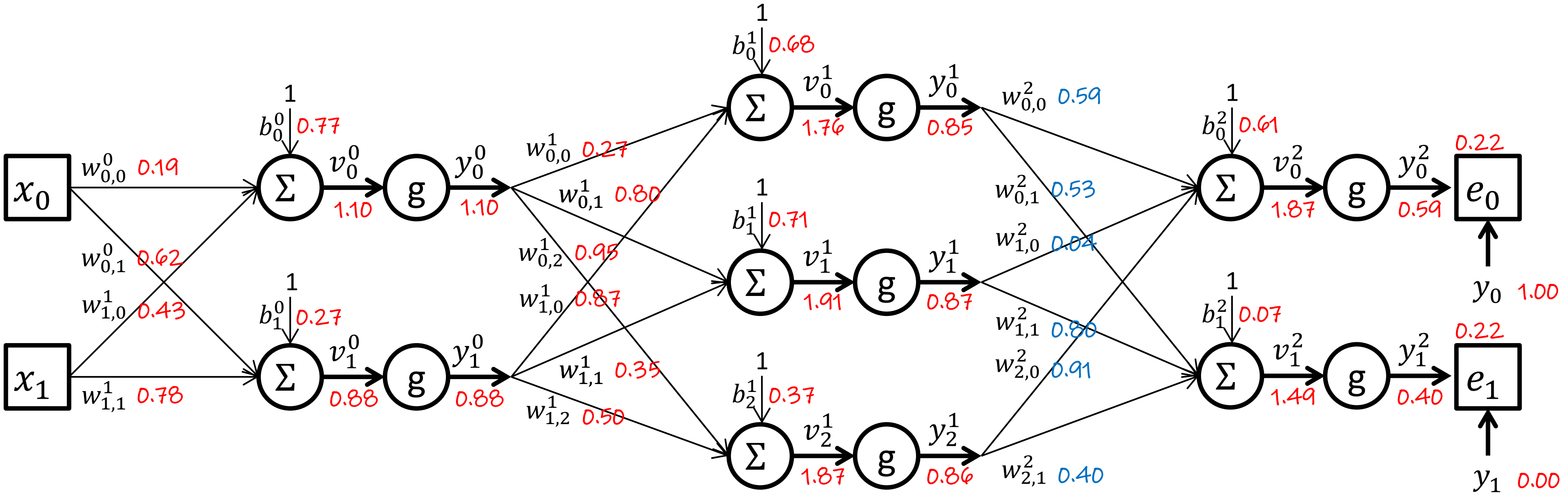
$$\mathbf{W}^2 = \begin{bmatrix} 0.56 & 0.50 \\ 0.01 & 0.77 \\ 0.88 & 0.36 \end{bmatrix}$$

$$\mathbf{W}'^2 = \mathbf{W}^2 - \eta \frac{\partial \mathbf{e}}{\partial \mathbf{W}^2}$$

$$\mathbf{W}'^2 = \begin{bmatrix} 0.56 & 0.50 \\ 0.01 & 0.77 \\ 0.88 & 0.36 \end{bmatrix} - 0.1 \begin{bmatrix} -0.34 & -0.34 \\ -0.35 & -0.35 \\ -0.35 & -0.35 \end{bmatrix}$$

$$\mathbf{W}'^2 = \begin{bmatrix} 0.59 & 0.53 \\ 0.04 & 0.80 \\ 0.91 & 0.40 \end{bmatrix}$$

Retropropagação (*backward*) – camada 2 (camada de saída)



$$\frac{\partial \mathbf{e}}{\partial \mathbf{W}^2} = \begin{bmatrix} -0.34 & -0.34 \\ -0.35 & -0.35 \\ -0.35 & -0.35 \end{bmatrix}$$

$$\eta = 0.1$$

$$\mathbf{W}^2 = \begin{bmatrix} 0.56 & 0.50 \\ 0.01 & 0.77 \\ 0.88 & 0.36 \end{bmatrix}$$

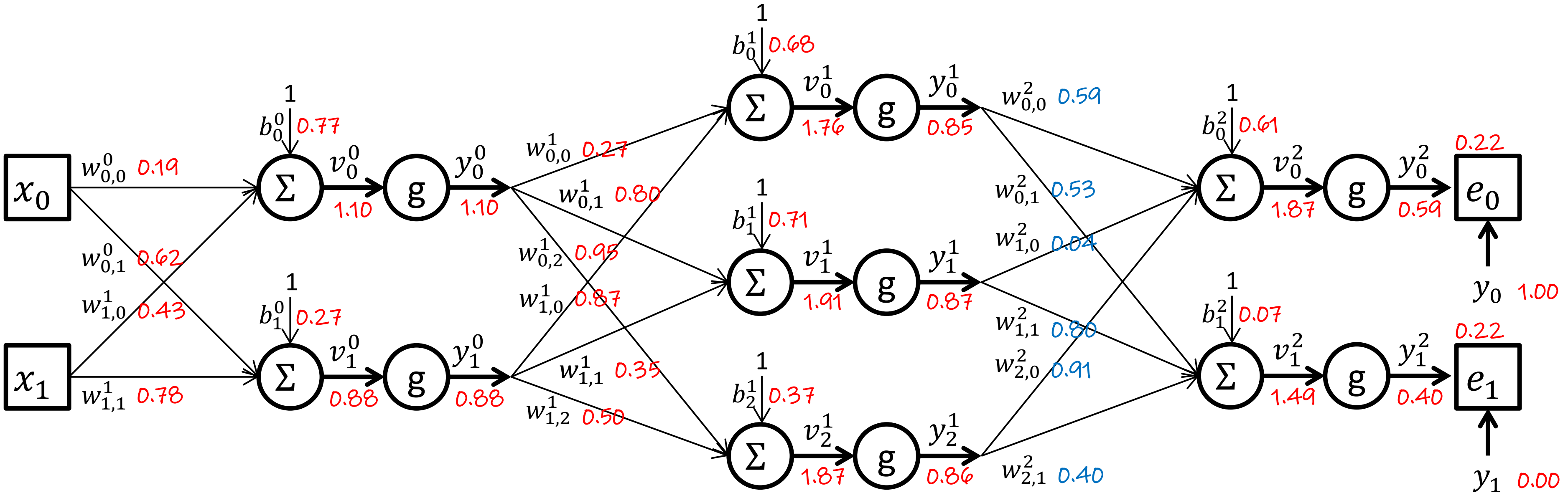
$$\mathbf{W}'^2 = \mathbf{W}^2 - \eta \frac{\partial \mathbf{e}}{\partial \mathbf{W}^2}$$

$$\mathbf{W}'^2 = \begin{bmatrix} 0.56 & 0.50 \\ 0.01 & 0.77 \\ 0.88 & 0.36 \end{bmatrix} - 0.1 \begin{bmatrix} -0.34 & -0.34 \\ -0.35 & -0.35 \\ -0.35 & -0.35 \end{bmatrix}$$

$$\mathbf{W}'^2 = \begin{bmatrix} 0.59 & 0.53 \\ 0.04 & 0.80 \\ 0.91 & 0.40 \end{bmatrix}$$

Retropropagação (*backward*)
Camada 1 (camada escondida)

Retropropagação (*backward*) – camada 1 (camada escondida)



$$\frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} = [-1.68 \quad -1.68]$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^1} = \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^1} \times \frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} \times \frac{\partial \mathbf{v}^1}{\partial \mathbf{W}^1}$$

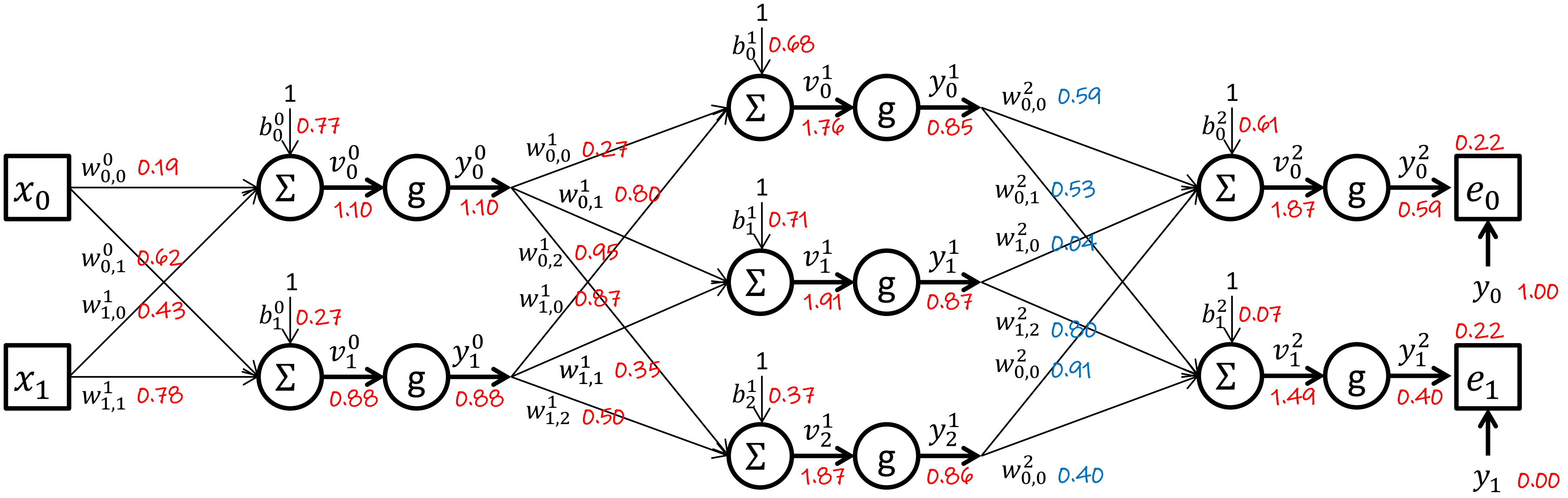
$$\frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} = [0.24 \quad 0.24]$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^1} = \frac{\partial \mathbf{e}}{\partial \mathbf{y}^2} \times \frac{\partial \mathbf{y}^2}{\partial \mathbf{v}^2} \times \frac{\partial \mathbf{v}^2}{\partial \mathbf{y}^1} = [-1.68 \quad -1.68][0.24 \quad 0.24]\begin{bmatrix} 0.59 & 0.53 \\ 0.04 & 0.80 \\ 0.91 & 0.40 \end{bmatrix}$$

$$\frac{\partial \mathbf{v}^2}{\partial \mathbf{y}^1} = \mathbf{W}^2 = \begin{bmatrix} 0.59 & 0.53 \\ 0.04 & 0.80 \\ 0.91 & 0.40 \end{bmatrix}$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^1} = \begin{bmatrix} -0.24 & -0.21 \\ -0.02 & -0.32 \\ -0.37 & -0.16 \end{bmatrix} \qquad (somando\ para\ \mathbf{y}^1) = [-0.46 \quad -0.34 \quad -0.53]$$

Retropropagação (*backward*) – camada 1 (camada escondida)



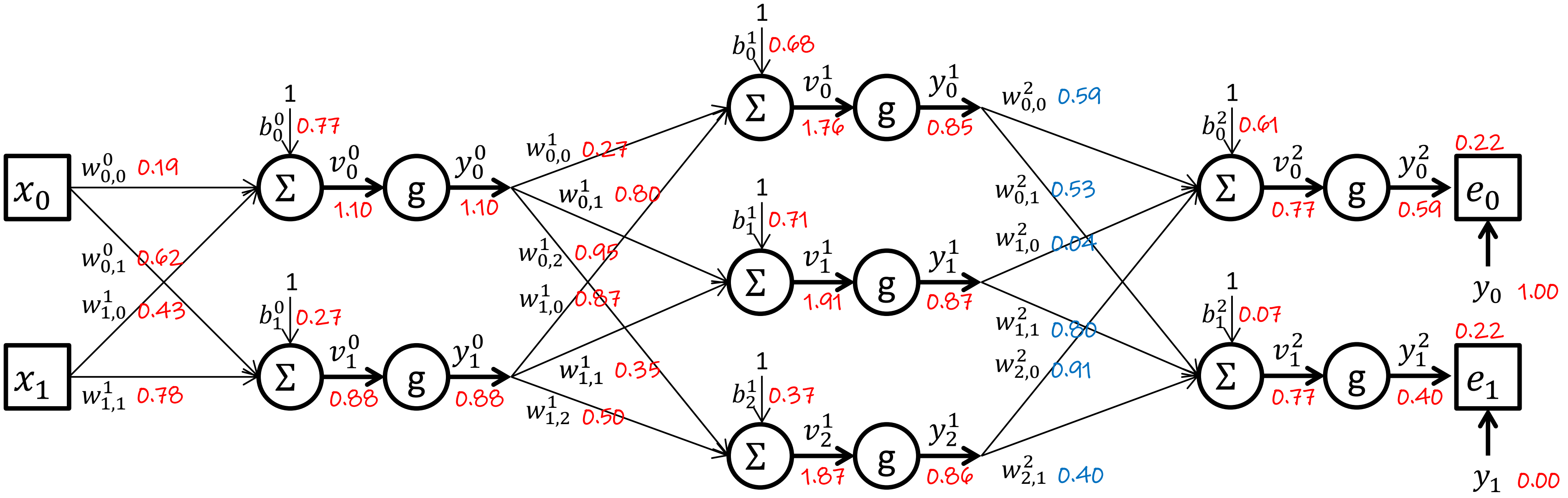
$$\mathbf{v}^1 = [1.76 \quad 1.91 \quad 1.87]$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^1} = \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^1} \times \frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} \times \frac{\partial \mathbf{v}^1}{\partial \mathbf{W}^1}$$

$$\frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} = \frac{1}{1+e^{-\mathbf{v}^1}} \times 1 - \frac{1}{1+e^{-\mathbf{v}^1}} = \frac{1}{1+e^{-[1.76 \quad 1.91 \quad 1.87]}} \times 1 - \frac{1}{1+e^{-[1.76 \quad 1.91 \quad 1.87]}}$$

$$\frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} = [0.12 \quad 0.11 \quad 0.11]$$

Retropropagação (*backward*) – camada 1 (camada escondida)



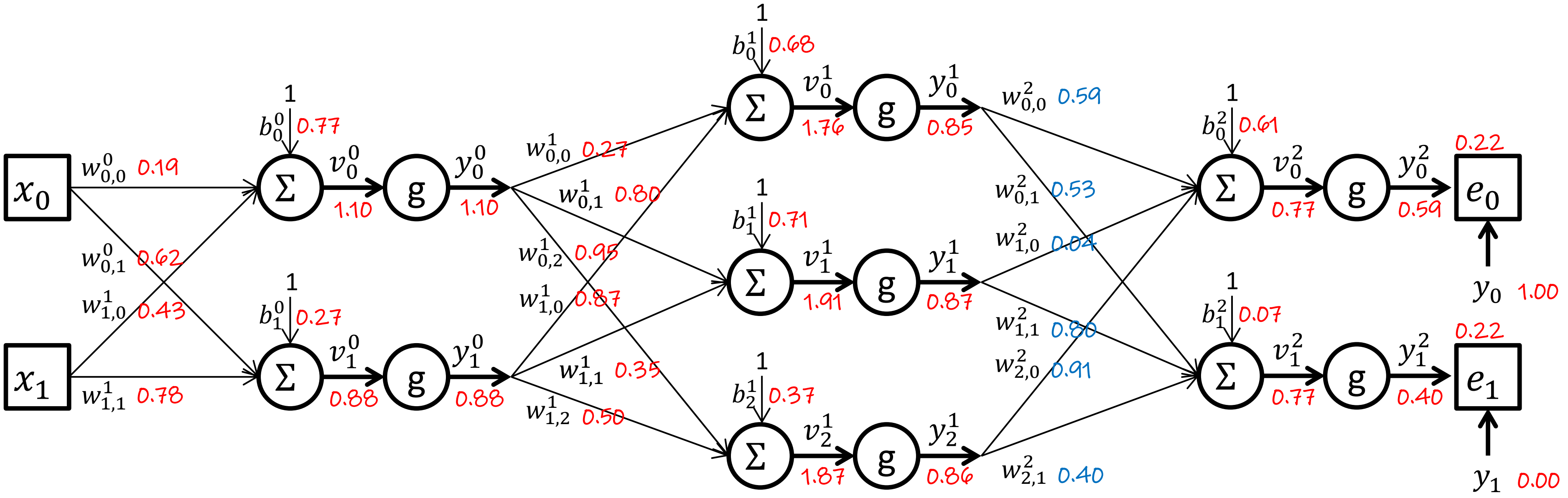
$$\mathbf{y}^0 = [1.10 \quad 0.88]$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^1} = \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^1} \times \frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} \times \frac{\partial \mathbf{v}^1}{\partial \mathbf{W}^1}$$

$$\frac{\partial \mathbf{v}^1}{\partial \mathbf{W}^1} = \frac{\partial(\mathbf{y}^0\mathbf{W}^1 + \mathbf{b}^1)}{\partial \mathbf{W}^1} = \left[\mathbf{y}^{0^T} \quad \mathbf{y}^{0^T} \quad \mathbf{y}^{0^T}\right]$$

$$\frac{\partial \mathbf{v}^1}{\partial \mathbf{W}^1} = \begin{bmatrix} 1.10 & 1.10 & 1.10 \\ 0.88 & 0.88 & 0.88 \end{bmatrix}$$

Retropropagação (*backward*) – camada 1 (camada escondida)



$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^1} = \begin{bmatrix} -0.46 & -0.34 & -0.53 \end{bmatrix}$$

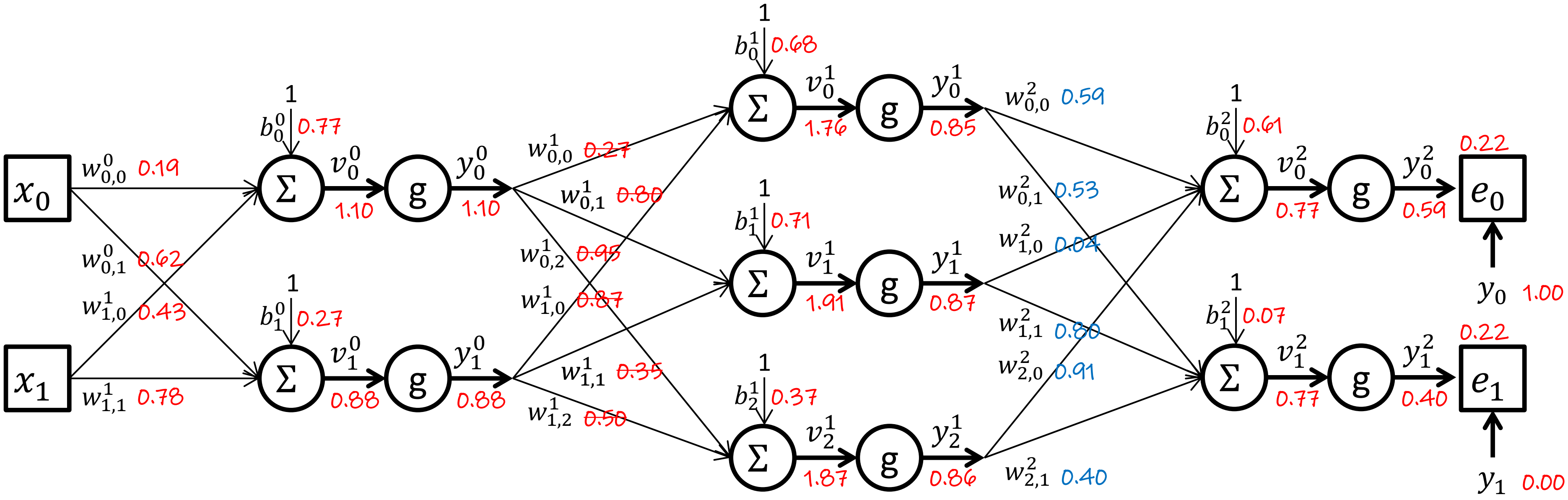
$$\frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} = \begin{bmatrix} 0.12 & 0.11 & 0.11 \end{bmatrix}$$

$$\frac{\partial \mathbf{v}^1}{\partial \mathbf{W}^1} = \begin{bmatrix} 1.10 & 1.10 & 1.10 \\ 0.88 & 0.88 & 0.88 \end{bmatrix}$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^1} = \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^1} \times \frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} \times \frac{\partial \mathbf{v}^1}{\partial \mathbf{W}^1} = \begin{bmatrix} -0.46 & -0.34 & -0.53 \end{bmatrix} \times \begin{bmatrix} 0.12 & 0.11 & 0.11 \end{bmatrix} \times \begin{bmatrix} 1.10 & 1.10 & 1.10 \\ 0.88 & 0.88 & 0.88 \end{bmatrix}$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^1} = \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^1} \times \frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} \times \frac{\partial \mathbf{v}^1}{\partial \mathbf{W}^1} = \begin{bmatrix} -0.06 & -0.04 & -0.06 \\ -0.05 & -0.03 & -0.05 \end{bmatrix}$$

Retropropagação (*backward*) – camada 1 (camada escondida)



$$\mathbf{W^1} = \begin{bmatrix} 0.27 & 0.80 & 0.95 \\ 0.87 & 0.35 & 0.50 \end{bmatrix}$$

$$\mathbf{W'^1} = \mathbf{W^1} - \eta \frac{\partial \mathbf{e}}{\partial \mathbf{W^1}}$$

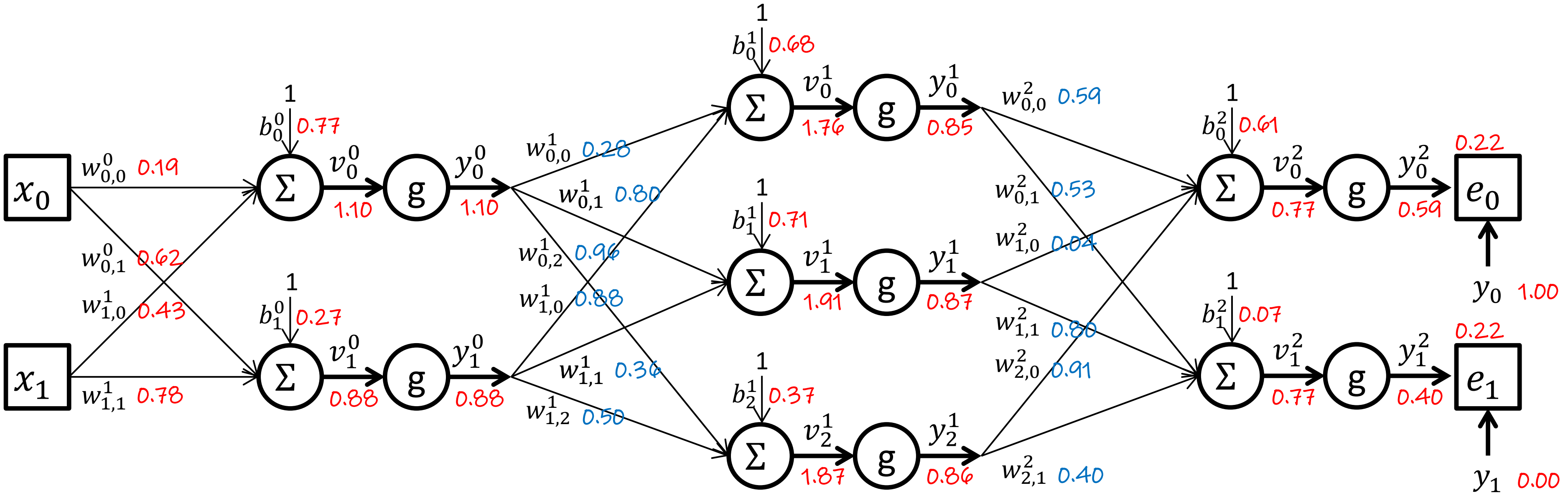
$$\eta = 0.1$$

$$\mathbf{W'^1} = \begin{bmatrix} 0.27 & 0.80 & 0.95 \\ 0.87 & 0.35 & 0.50 \end{bmatrix} - 0.1 \begin{bmatrix} -0.06 & -0.04 & -0.06 \\ -0.05 & -0.03 & -0.05 \end{bmatrix}$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W^1}} = \begin{bmatrix} -0.06 & -0.04 & -0.06 \\ -0.05 & -0.03 & -0.05 \end{bmatrix}$$

$$\mathbf{W'^1} = \begin{bmatrix} 0.28 & 0.80 & 0.96 \\ 0.88 & 0.36 & 0.50 \end{bmatrix}$$

Retropropagação (*backward*) – camada 1 (camada escondida)



$$\mathbf{W}^1 = \begin{bmatrix} 0.27 & 0.80 & 0.95 \\ 0.87 & 0.35 & 0.50 \end{bmatrix}$$

$$\eta = 0.1$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^1} = \begin{bmatrix} -0.06 & -0.04 & -0.06 \\ -0.05 & -0.03 & -0.05 \end{bmatrix}$$

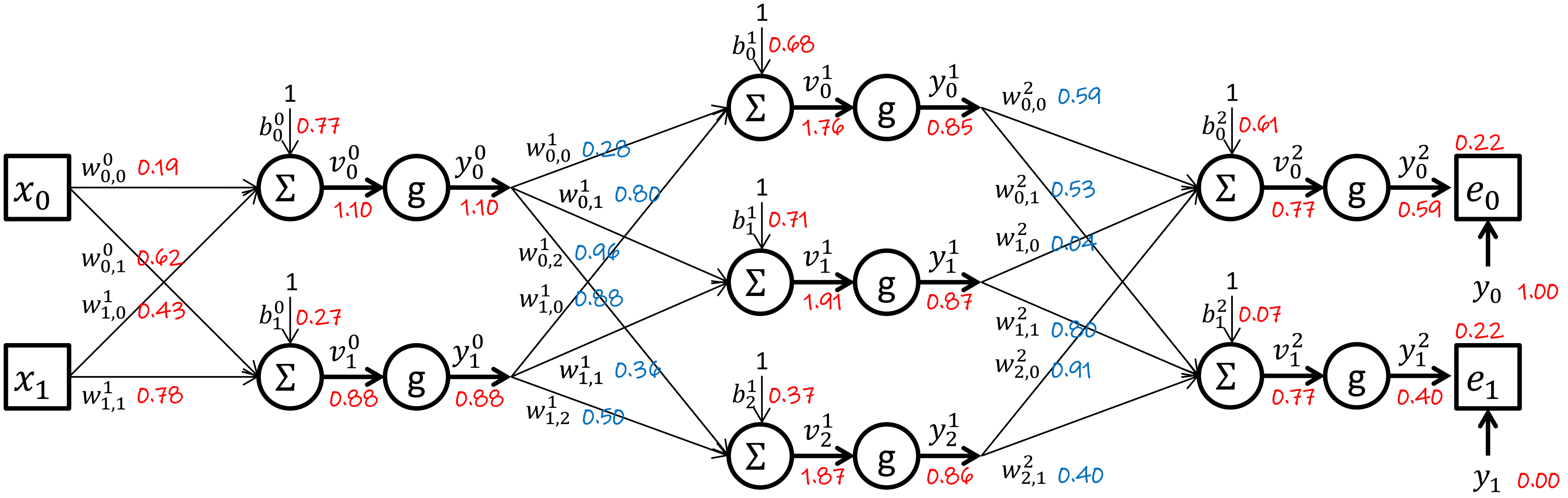
$$\mathbf{W}'^1 = \mathbf{W}^1 - \eta \frac{\partial \mathbf{e}}{\partial \mathbf{W}^1}$$

$$\mathbf{W}'^1 = \begin{bmatrix} 0.27 & 0.80 & 0.95 \\ 0.87 & 0.35 & 0.50 \end{bmatrix} - 0.1 \begin{bmatrix} -0.06 & -0.04 & -0.06 \\ -0.05 & -0.03 & -0.05 \end{bmatrix}$$

$$\mathbf{W}'^1 = \begin{bmatrix} 0.28 & 0.80 & 0.96 \\ 0.88 & 0.36 & 0.50 \end{bmatrix}$$

Retropropagação (*backward*)
Camada 0 (camada escondida)

Retropropagação (*backward*) – camada 0 (camada escondida)



$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^1} = [-0.46 \quad -0.34 \quad -0.53]$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^0} = \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^0} \times \frac{\partial \mathbf{y}^0}{\partial \mathbf{v}^0} \times \frac{\partial \mathbf{v}^0}{\partial \mathbf{W}^0}$$

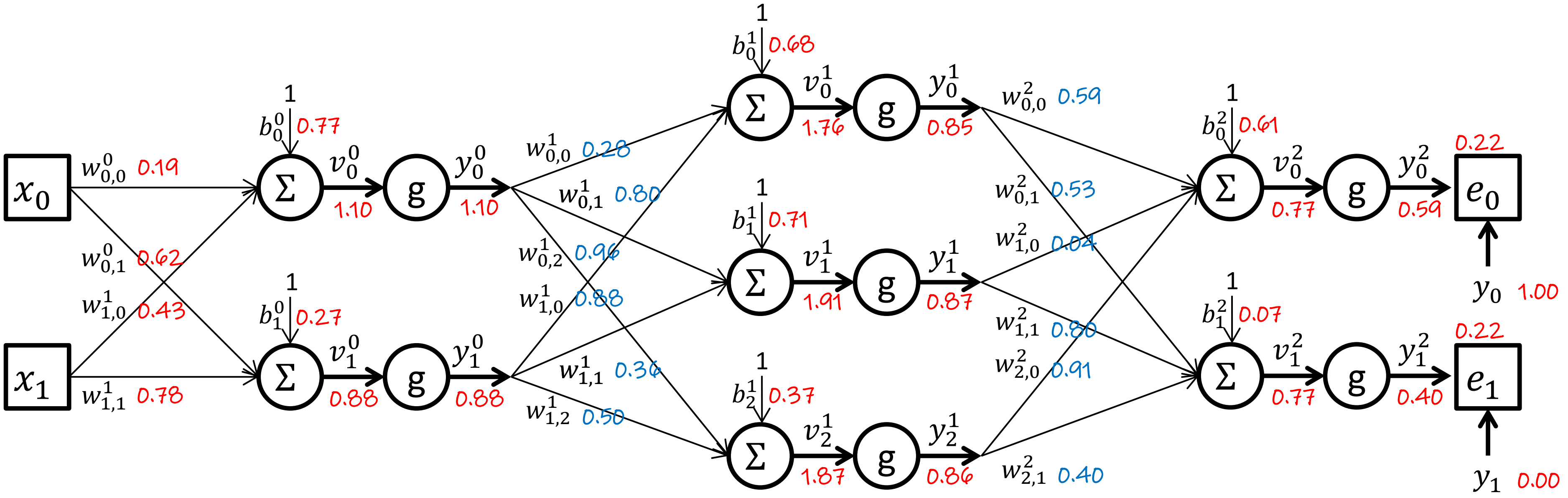
$$\frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} = [0.12 \quad 0.11 \quad 0.11]$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^0} = \frac{\partial \mathbf{e}}{\partial \mathbf{y}^1} \times \frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} \times \frac{\partial \mathbf{v}^1}{\partial \mathbf{y}^0} = [-0.46 \quad -0.34 \quad -0.53] \times [0.12 \quad 0.11 \quad 0.11] \times \begin{bmatrix} 0.27 & 0.80 & 0.95 \\ 0.87 & 0.35 & 0.50 \end{bmatrix}$$

$$\frac{\partial \mathbf{v}^1}{\partial \mathbf{y}^0} = \mathbf{W}^1 = \begin{bmatrix} 0.27 & 0.80 & 0.95 \\ 0.87 & 0.35 & 0.50 \end{bmatrix}$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^0} = \frac{\partial \mathbf{e}}{\partial \mathbf{y}^1} \times \frac{\partial \mathbf{y}^1}{\partial \mathbf{v}^1} \times \frac{\partial \mathbf{v}^1}{\partial \mathbf{y}^0} = \begin{bmatrix} -0.01 & -0.03 & -0.05 \\ -0.05 & -0.01 & -0.03 \end{bmatrix} \quad (somando\ para\ \mathbf{y}^0) = [-0.10 \quad -0.09]$$

Retropropagação (*backward*) – camada 1 (camada escondida)



$\mathbf{v}^0 = [1.10 \quad 0.88]$

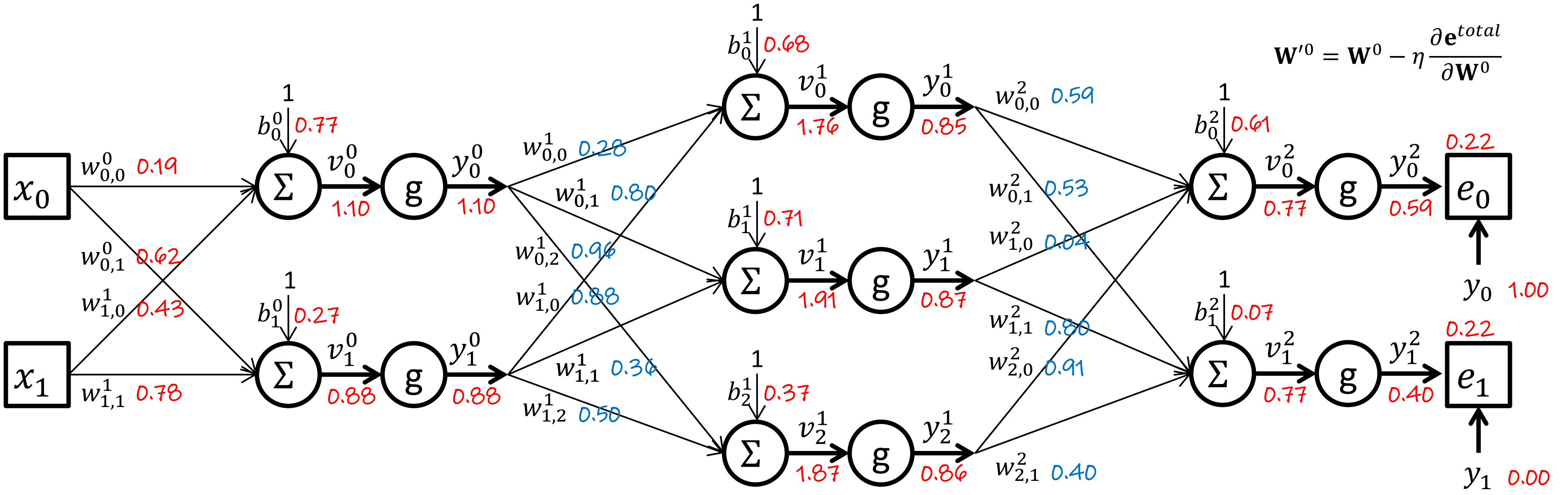
$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^0} = \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^0} \times \frac{\partial \mathbf{y}^0}{\partial \mathbf{v}^0} \times \frac{\partial \mathbf{v}^0}{\partial \mathbf{W}^0}$$

$$\frac{\partial \mathbf{y}^0}{\partial \mathbf{v}^0} = \begin{cases} 1, & se\ \mathbf{v}^0 > 0 \\ 0, & caso\ contrário \end{cases}$$

$$\frac{\partial \mathbf{y}^0}{\partial \mathbf{v}^0} = [1.0 \quad 1.0]$$

Retropropagação (*backward*) – camada 1 (camada escondida)

$$\mathbf{W}'^{0} = \mathbf{W}^{0} - \eta \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^{0}}$$



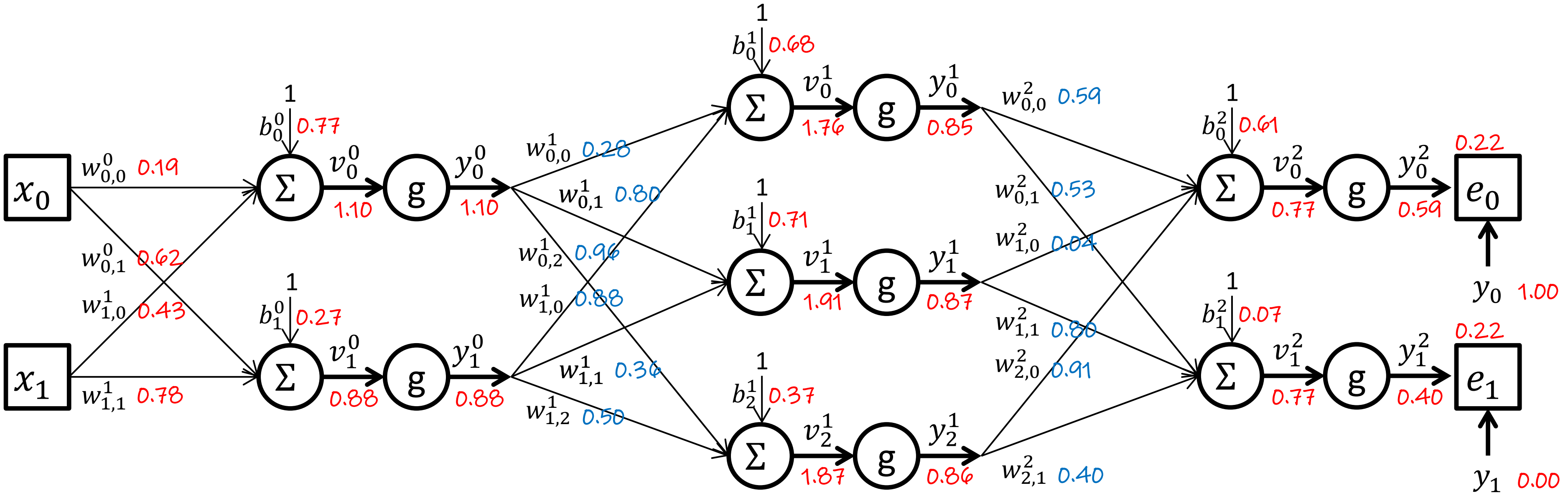
$\mathbf{x} = [0.10 \quad 0.70]$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^{0}} = \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^{0}} \times \frac{\partial \mathbf{y}^{0}}{\partial \mathbf{v}^{0}} \times \frac{\partial \mathbf{v}^{0}}{\partial \mathbf{W}^{0}}$$

$$\frac{\partial \mathbf{v}^{0}}{\partial \mathbf{W}^{0}} = \frac{\partial (\mathbf{x}\mathbf{W}^{0} + \mathbf{b}^{0})}{\partial \mathbf{W}^{0}} = [\mathbf{x}^T \quad \mathbf{x}^T]$$

$$\frac{\partial \mathbf{v}^{0}}{\partial \mathbf{W}^{0}} = \begin{bmatrix} 0.10 & 0.10 \\ 0.70 & 0.70 \end{bmatrix}$$

Retropropagação (*backward*) – camada 1 (camada escondida)



$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^0} = [-0.10 \quad -0.09]$$

$$\frac{\partial \mathbf{y}^0}{\partial \mathbf{v}^0} = [1.0 \quad 1.0]$$

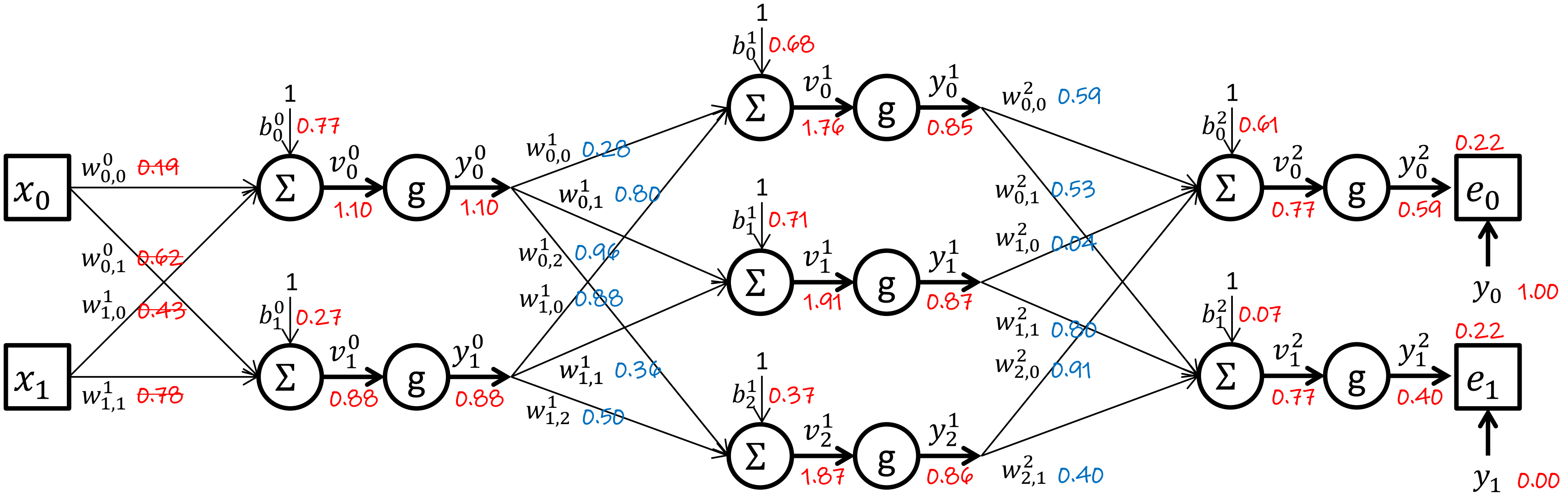
$$\frac{\partial \mathbf{v}^0}{\partial \mathbf{W}^0} = \begin{bmatrix} 0.10 & 0.10 \\ 0.70 & 0.70 \end{bmatrix}$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^0} = \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{y}^0} \times \frac{\partial \mathbf{y}^0}{\partial \mathbf{v}^0} \times \frac{\partial \mathbf{v}^0}{\partial \mathbf{W}^0}$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^0} = [-0.10 \quad -0.09] \times [1.0 \quad 1.0] \times \begin{bmatrix} 0.10 & 0.10 \\ 0.70 & 0.70 \end{bmatrix}$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^0} = \begin{bmatrix} -0.01 & -0.00 \\ -0.07 & -0.06 \end{bmatrix}$$

Retropropagação (*backward*) – camada 1 (camada escondida)



$$\mathbf{W^0} = \begin{bmatrix} 0.19 & 0.62 \\ 0.43 & 0.78 \end{bmatrix}$$

$$\eta = 0.1$$

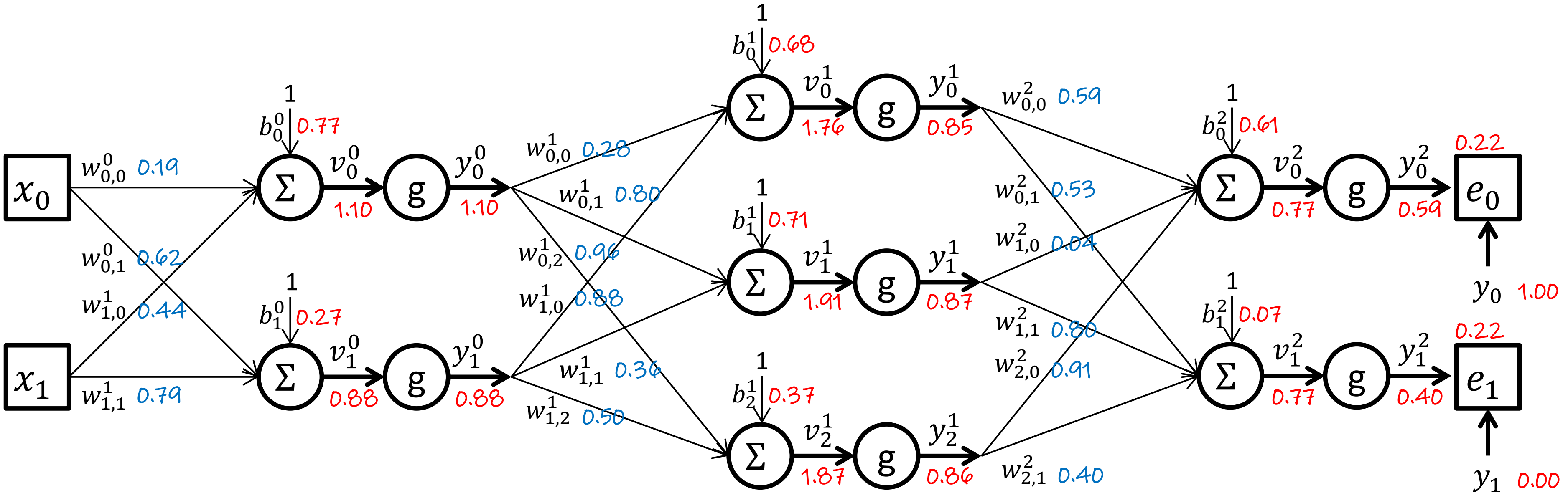
$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^0} = \begin{bmatrix} -0.01 & -0.00 \\ -0.07 & -0.06 \end{bmatrix}$$

$$\mathbf{W}'^0 = \mathbf{W}^0 - \eta \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^0}$$

$$\mathbf{W}'^0 = \begin{bmatrix} 0.19 & 0.62 \\ 0.43 & 0.78 \end{bmatrix} - 0.1 \times \begin{bmatrix} -0.01 & -0.00 \\ -0.07 & -0.06 \end{bmatrix}$$

$$\mathbf{W}'^0 = \begin{bmatrix} 0.19 & 0.62 \\ 0.44 & 0.79 \end{bmatrix}$$

Retropropagação (*backward*) – camada 1 (camada escondida)



$$\mathbf{W^0} = \begin{bmatrix} 0.19 & 0.62 \\ 0.43 & 0.78 \end{bmatrix}$$

$$\eta = 0.1$$

$$\frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^0} = \begin{bmatrix} -0.01 & -0.00 \\ -0.07 & -0.06 \end{bmatrix}$$

$$\mathbf{W}'^0 = \mathbf{W}^0 - \eta \frac{\partial \mathbf{e}^{total}}{\partial \mathbf{W}^0}$$

$$\mathbf{W}'^0 = \begin{bmatrix} 0.19 & 0.62 \\ 0.43 & 0.78 \end{bmatrix} - 0.1 \times \begin{bmatrix} -0.01 & -0.00 \\ -0.07 & -0.06 \end{bmatrix}$$

$$\mathbf{W}'^0 = \begin{bmatrix} 0.19 & 0.62 \\ 0.44 & 0.79 \end{bmatrix}$$

REFERENCIAS

- Referencia principal:
  - Back-Propagation is very simple. Who made it Complicated ?
    - https://medium.com/@14prakash/back-propagation-is-very-simple-who-made-it-complicated-97b794c97e5c

- Outras referencias:
  - Chapter 7 : Artificial neural networks with Math.
    - https://medium.com/deep-math-machine-learning-ai/chapter-7-artificial-neural-networks-with-math-bb711169481b
  - The Matrix Calculus You Need For Deep Learning
    - http://explained.ai/matrix-calculus/index.html
  - How backpropagation works, and how you can use Python to build a neural network
    - https://medium.freecodecamp.org/build-a-flexible-neural-network-with-backpropagation-in-python-acffeb7846d0
  - All the Backpropagation derivatives
    - https://medium.com/@pdquant/all-the-backpropagation-derivatives-d5275f727f60

REFERENCIAS

- Material do Prof. Moacir Ponti (ICMC-USP) sobre Deep Learning
  - https://github.com/maponti/deeplearning_intro_datascience
- 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 and Courville. Deep Learning. MIT Press, 2016
  - https://www.deeplearningbook.org/

FIM