Multilayer Perceptron

Multilayer Dense Feedforward Neural Networks

Vitor Greati[1]

[1]Federal University of Rio Grande do Norte

Table of Contents

Introduction

TODO.

Neurons

Basic elements

The neuron's basic task is to take an input, perform some computation and output a value.

Main elements

- An input vector $\mathbf{x} = \langle x_1, x_2, \ldots, x_n \rangle \in \mathbb{R}^n$.
- A vector of weights $\mathbf{w} = \langle w_1, w_2, \ldots, w_n \rangle \in \mathbb{R}^n$.
- A constant $b \in \mathbb{R}$, called *bias*.
- A computation between inputs and weights, like

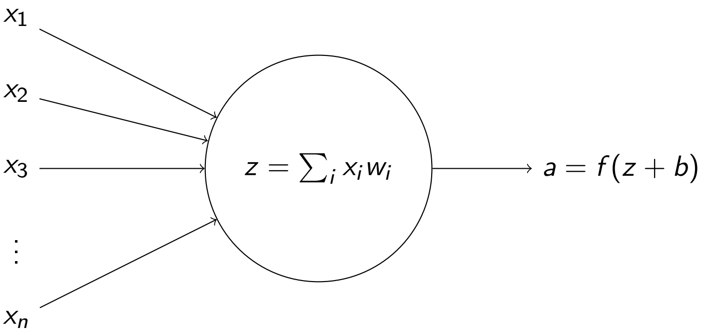
$$z = \mathbf{x} \cdot \mathbf{w} = \sum_i x_i w_i .$$

- An activation function f to produce an output $f(z + b)$.

Neurons

Graphical representation

A common graphical representation highlights those elements:



Neurons

Graphical representation

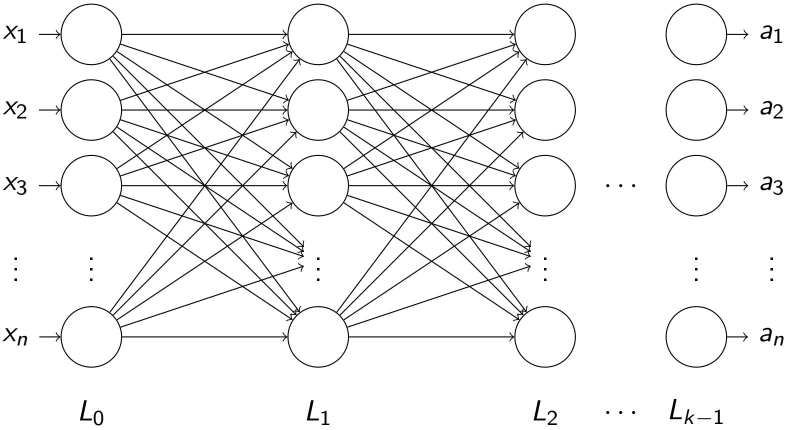
The same representation, but using only vectors:

$$\mathbf{x} \longrightarrow \boxed{z = \mathbf{x} \cdot \mathbf{w}} \longrightarrow a = f(z + b)$$

Multilayer Architecture

Fully connected feedforward architecture

An architecture with k layers, $L_0, L_1, \ldots, L_{k-1}$, is graphically represented as:



Multilayer Architecture

Fully connected feedforward architecture

Observations

- L_0 is the **input layer**.
  - Neurons in this layer are **input neurons**.
  - An input neuron j takes the j-th component of the input vector.
  - Input neurons do not perform any computation or activation: just output the input value.
  - The output of L_0 is $\mathbf{x}$, the input vector.
- L_{k-1} is the **output layer** and its output is the network output.
- This architecture is **fully connected** because each neuron in layer L_i is connected to every neuron in layer L_{i+1}.
- This architecture is **feedforward** because there is no back arrows forming cycles; otherwise, it would be *recurrent*.

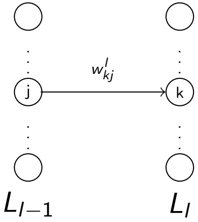
Multilayer Architecture

Fully connected feedforward architecture

How can we perform calculations and learning in this architecture? Mathematics, of course.

Representing connections between layers

Consider layer $L_l, l = 1, \ldots, C-1$, and neuron k of L_l. Since the architecture is fully connected, k is connected to all neurons j in L_{l-1}. Denote by w_{kj}^l the weight of the connection between neuron j of L_{l-1} and neuron k of L_l:



Multilayer Architecture

Fully connected feedforward architecture

Representing connections between layers

In this way, we can represent all weights between two layers using only one matrix $\mathbf{W}^l = (w_{kj}^l)$:

$$\mathbf{W}^l = \begin{bmatrix} w_{11}^l & w_{12}^l & \dots & w_{1|L_{l-1}|}^l \\ w_{21}^l & w_{22}^l & \dots & w_{2|L_{l-1}|}^l \\ \vdots & \vdots & \ddots & \vdots \\ w_{|L_l|1}^l & w_{|L_l|2}^l & \dots & w_{|L_l||L_{l-1}|}^l \end{bmatrix}.$$

Notice that:

- $\mathbf{W}^l$ has dimensions $|L_l| \times |L_{l-1}|$.
- The weight vector of connections to neuron k in L_l is the k-th row of $\mathbf{W}^l$, denoted by $\mathbf{w_k^l}$.
- All connections in the network are represented! They are all in matrices $\mathbf{W}^1, \mathbf{W}^2, \dots, \mathbf{W}^{C-1}$.

Representing the biases

Remember that each neuron has its own bias. So, let b^l_k denote the bias of neuron k in layer L_l. Then, the biases in layer l are represented as just a vector

$$\mathbf{b}^l = \langle b^l_1, b^l_2, \ldots, b^l_{|L_l|} \rangle.$$

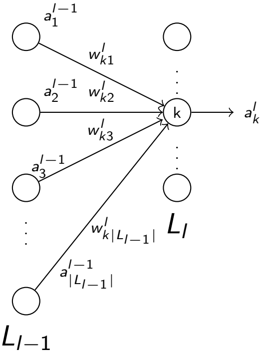
In this way, all biases are represented by vectors $\mathbf{b}^1, \mathbf{b}^2, \ldots, \mathbf{b}^{C-1}$.

Multilayer Architecture

Fully connected feedforward architecture

Representing layer outputs

Denote by a^l_k the output of neuron k in layer L_l, and by $\mathbf{a^l}$ the output vector for layer L_l. How to compute such output? Look:



Representing layer outputs

Denote by z_k^l the computation performed by the neuron, i.e:

$$z_k^l = \sum_j^{|L_{l-1}|} w_{kj}^l a_j^{l-1} = \mathbf{w_k^l a^{l-1}}.$$

Then, the neuron output is

$$a_k^l = f(z_k^l + b_k^l).$$

And the **layer output** is

$$\mathbf{a}^l = \mathbf{f}(\mathbf{W}^l \cdot \mathbf{a}^{l-1} + \mathbf{b}^l).$$

Multilayer Architecture

Fully connected feedforward architecture

Didn't get it? Look:

$$
\begin{aligned}
\mathbf{f}\left(\mathbf{W}^{l}\cdot\mathbf{a}^{\mathbf{l}-1}+\mathbf{b}^{\mathbf{l}}\right) &= \mathbf{f}\left(\begin{bmatrix} w^{l}_{11} & w^{l}_{12} & \cdots & w^{l}_{1|L_{l-1}|} \\ w^{l}_{21} & w^{l}_{22} & \cdots & w^{l}_{2|L_{l-1}|} \\ \vdots & \vdots & \ddots & \vdots \\ w^{l}_{|L_l|1} & w^{l}_{|L_l|2} & \cdots & w^{l}_{|L_l||L_{l-1}|} \end{bmatrix}\cdot\begin{bmatrix} a^{l-1}_{1} \\ a^{l-1}_{2} \\ \vdots \\ a^{l-1}_{|L_{l-1}|} \end{bmatrix}+\begin{bmatrix} b^{l}_{1} \\ b^{l}_{2} \\ \vdots \\ b^{l}_{|L_l|} \end{bmatrix}\right) \\
&= \mathbf{f}\left(\begin{bmatrix} \sum_j^{|L_{l-1}|} w_{1j}a^{l-1}_{j}+b^{l}_{1} \\ \sum_j^{|L_{l-1}|} w_{2j}a^{l-1}_{j}+b^{l}_{2} \\ \vdots \\ \sum_j^{|L_{l-1}|} w_{|L_l|j}a^{l-1}_{j}+b^{l}_{|L_l|} \end{bmatrix}\right) \\
&= \mathbf{f}\left(\begin{bmatrix} \mathbf{w}^{\mathbf{l}}_{\mathbf{1}}\mathbf{a}^{\mathbf{l}-\mathbf{1}}+b^{l}_{1} \\ \mathbf{w}^{\mathbf{l}}_{\mathbf{2}}\mathbf{a}^{\mathbf{l}-\mathbf{1}}+b^{l}_{2} \\ \vdots \\ \mathbf{w}^{\mathbf{l}}_{|\mathbf{L}_{\mathbf{l}}|}\mathbf{a}^{\mathbf{l}-\mathbf{1}}+b^{l}_{|L_l|} \end{bmatrix}\right) \\
&= \begin{bmatrix} f(z^{l}_{1}) & f(z^{l}_{2}) & \cdots & f(z^{l}_{|L_l|}) \end{bmatrix}^{T} \\
&= \begin{bmatrix} a^{l}_{1} & a^{l}_{2} & \cdots & a^{l}_{|L_l|} \end{bmatrix}^{T} \\
&= \mathbf{a}^{\mathbf{l}}
\end{aligned}
$$