Artificial Intelligence: *Modeling Human Intelligence with Networks*

Jeová Farias Sales Rocha Neto
jeova_farias@brown.edu

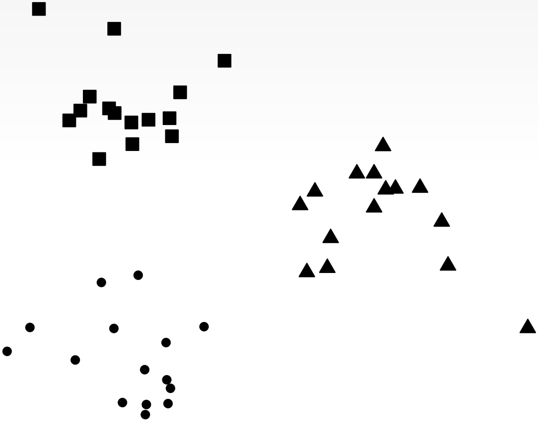
Multiclass Classification

Multiclass

- Our problem: m points in the plane.

Multiclass

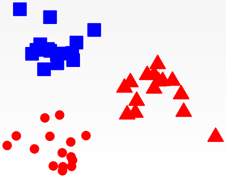
- Our problem: m points in the plane. Classify them in **three** classes:



- How to classify 3 classes if only have a binary classifier (can separate the dataset in two?).

Multiclass

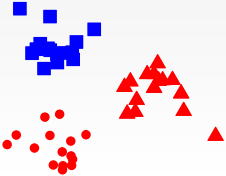
- We can use **three** classifiers!



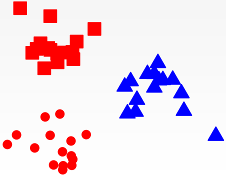
(a) Is this a square?

Multiclass

- We can use **three** classifiers!



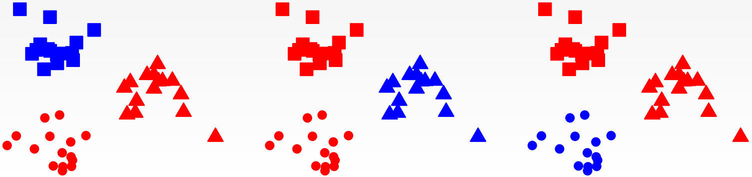
(a) Is this a square?



(b) Is this a triangle?

Multiclass

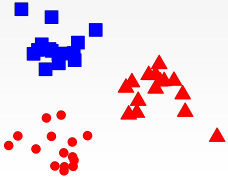
- We can use **three** classifiers!



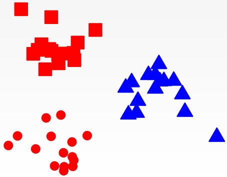
(a) Is this a square? *(b)* Is this a triangle? *(c)* Is this a ball?

Multiclass

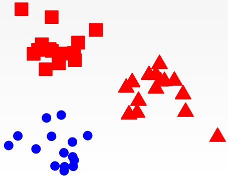
- We can use **three** classifiers!



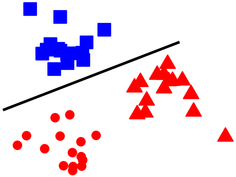
(a) Is this a square?



(b) Is this a triangle?

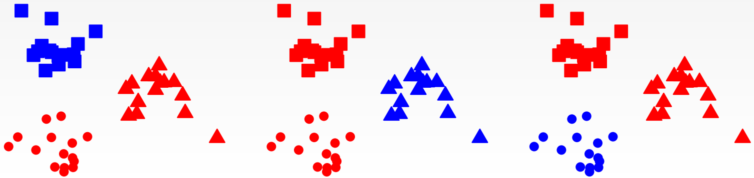


(c) Is this a ball?

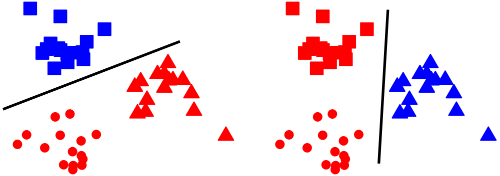


Multiclass

- We can use **three** classifiers!

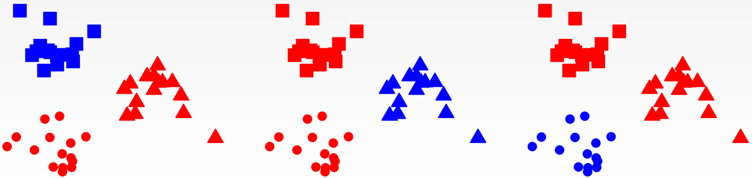


(a) Is this a square? *(b)* Is this a triangle? *(c)* Is this a ball?

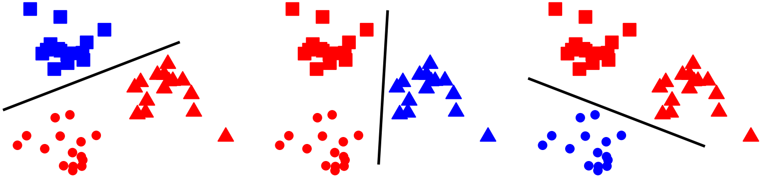


Multiclass

- We can use **three** classifiers!

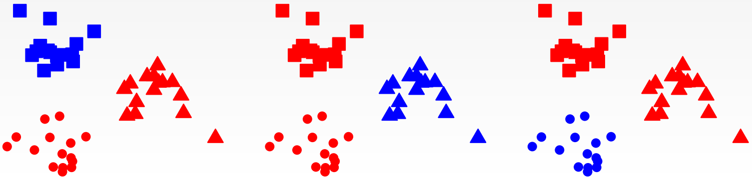


(a) Is this a square? *(b)* Is this a triangle? *(c)* Is this a ball?

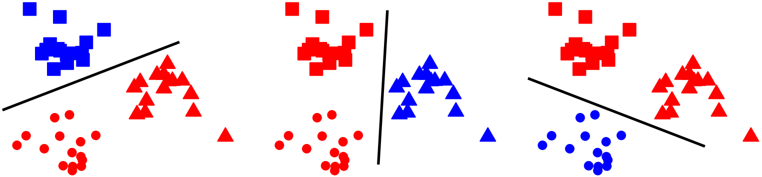


Multiclass

- We can use **three** classifiers!



(a) Is this a square? *(b)* Is this a triangle? *(c)* Is this a ball?



- This strategy is called *one-vs-rest* classification!

Matrix Multiplication

Matrix Multiplication

- Matrices are *not* just a set of vectors!

Matrix Multiplication

- Matrices are *not* just a set of vectors! They *transform* other vectors!!
- That happens through **Matrix Factorization**.

Matrix Multiplication

- Matrices are *not* just a set of vectors! They *transform* other vectors!!
- That happens through **Matrix Factorization**.
- The idea is simple: start with a matrix $M \in \mathbb{R}^{m \times n}$ (made of row vectors, $v^1, v^2, \ldots, v^m \in \mathbb{R}^n$).

$$M = \begin{bmatrix} — & v^1 & — \\ — & v^2 & — \\ — & \vdots & — \\ — & v^m & — \end{bmatrix}$$

Matrix Multiplication

- Matrices are *not* just a set of vectors! They *transform* other vectors!!
- That happens through **Matrix Factorization**.
- The idea is simple: start with a matrix $M \in \mathbb{R}^{m \times n}$ (made of row vectors, $v^1, v^2, \ldots, v^m \in \mathbb{R}^n$) and *column* vector $x \in \mathbb{R}^n$.

$$M = \begin{bmatrix} — & v^1 & — \\ — & v^2 & — \\ — & \vdots & — \\ — & v^m & — \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- Notice that $v^1, v^2, \ldots, v^m$ and x have the same size!
- The matrix multiplication Mx and is then done as:

$$Mx = \begin{bmatrix} — & v^1 & — \\ — & v^2 & — \\ — & \vdots & — \\ — & v^m & — \end{bmatrix} \begin{bmatrix} x1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} v^1 \cdot x \\ v^2 \cdot x \\ \vdots \\ v^m \cdot x \end{bmatrix}$$

Matrix Multiplication

- Example – let $M \in \mathbb{R}^{5\times 3}$ and $x \in \mathbb{R}^3$ be:

$$M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \end{bmatrix} \quad x = \begin{bmatrix} -1 \\ -2 \\ -3 \end{bmatrix}$$

Matrix Multiplication

- Example – let $M \in \mathbb{R}^{5\times 3}$ and $x \in \mathbb{R}^3$ be:

$$M = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \end{bmatrix} \quad x = \begin{bmatrix} -1 \\ -2 \\ -3 \end{bmatrix}$$

- Now, let $y = Mx$:

$$y = Mx = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 13 & 14 & 15 \end{bmatrix} \begin{bmatrix} -1 \\ -2 \\ -3 \end{bmatrix} = \begin{bmatrix} (1\times -1) + (2\times -2) + (3\times -3) \\ (4\times -1) + (5\times -2) + (6\times -3) \\ (7\times -1) + (8\times -2) + (9\times -3) \\ (10\times -1) + (11\times -2) + (12\times -3) \\ (13\times -1) + (14\times -2) + (15\times -3) \end{bmatrix} = \begin{bmatrix} -14 \\ -32 \\ -50 \\ -68 \\ -86 \end{bmatrix}$$

- Notice that $y \in \mathbb{R}^5$! x got *transformed* to y!

Matrix Multiplication

- How about multiplying two matrices?
- The idea is simple: start with a matrix $M \in \mathbb{R}^{m\times n}$ (made of *row* vectors, $v^1, v^2, \ldots, v^m \in \mathbb{R}^n$) a matrix $H \in \mathbb{R}^{n\times k}$ (made of *column* vectors, $u^1, u^2, \ldots, u^k \in \mathbb{R}^m$)

$$M = \begin{bmatrix} — & v^1 & — \\ — & v^2 & — \\ — & \vdots & — \\ — & v^m & — \end{bmatrix} \quad H = \begin{bmatrix} | & | & | & | \\ u^1 & u^2 & \cdots & u^k \\ | & | & | & | \end{bmatrix}$$

- Notice that M has as may columns as H has rows!
- The matrix multiplication Mx and is then done as:

$$MH = \begin{bmatrix} — & v^1 & — \\ — & v^2 & — \\ — & \vdots & — \\ — & v^m & — \end{bmatrix} \begin{bmatrix} | & | & | & | \\ u^1 & u^2 & \cdots & u^k \\ | & | & | & | \end{bmatrix} = \begin{bmatrix} v^1 \cdot u^1 & v^1 \cdot u^2 & \cdots & v^1 \cdot u^k \\ v^2 \cdot u^1 & v^2 \cdot u^2 & \cdots & v^2 \cdot u^k \\ \vdots & \vdots & \ddots & \vdots \\ v^m \cdot u^1 & v^m \cdot u^2 & \cdots & v^m \cdot u^k \end{bmatrix}$$

Matrix Multiplication

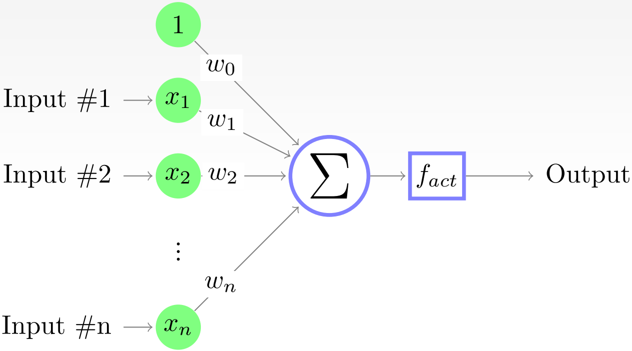
- The dimensions work this way:

$$(m \times n) \cdot (n \times k) = (m \times k)$$

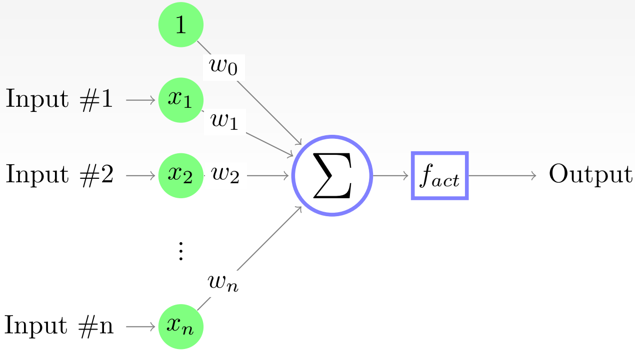
product is defined

Neuronal evolution

Our friend Perceptron

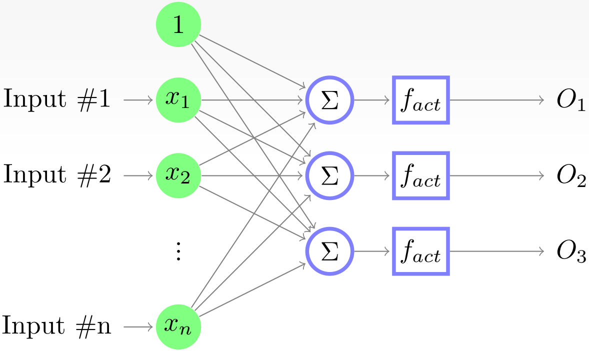


Our friend Perceptron

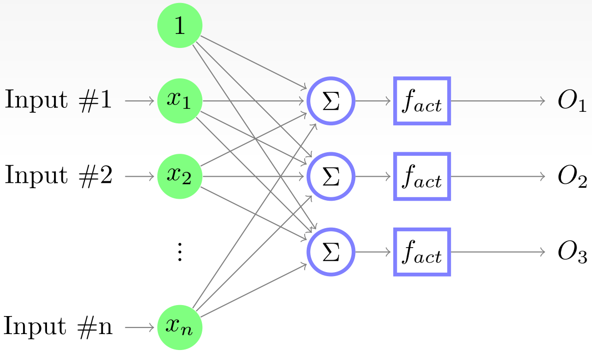


$$\text{Output} = f_{act}(w \cdot x) = \text{sign}(w \cdot x)$$

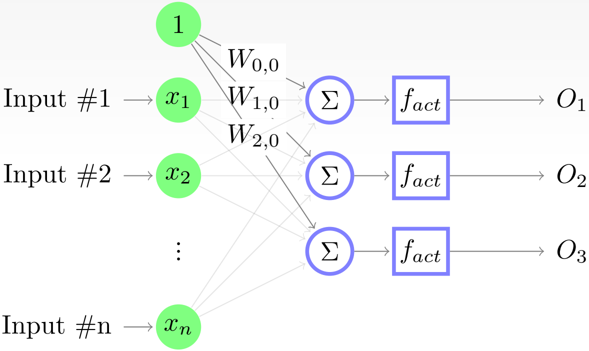
Multiclass Perceptron (a neural *network*)!



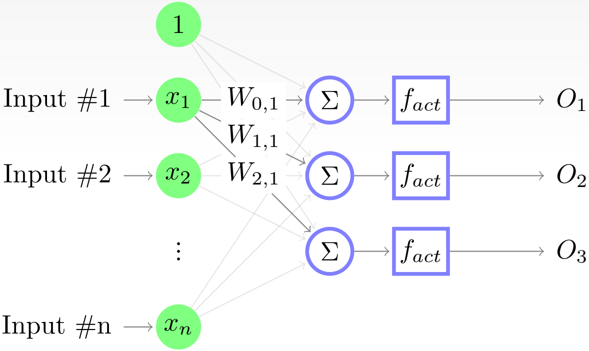
Multiclass Perceptron (a neural *network*)!



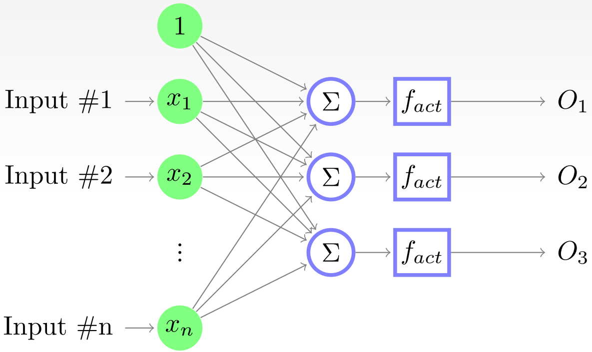
Multiclass Perceptron (a neural *network*)!



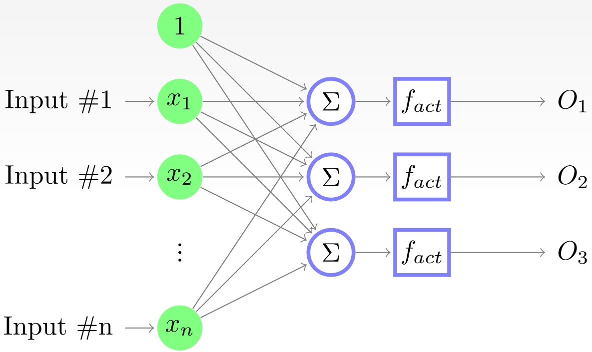
Multiclass Perceptron (a neural *network*)!



Multiclass Perceptron (a neural *network*)!

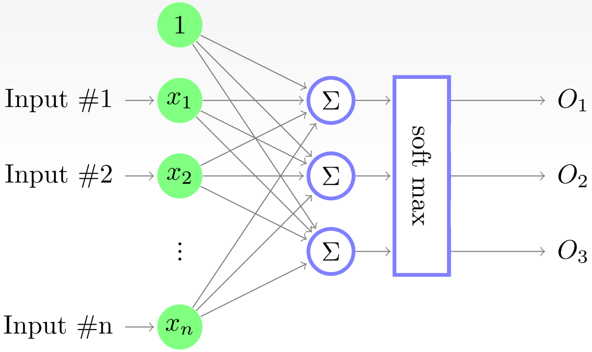


Multiclass Perceptron (a neural *network*)!

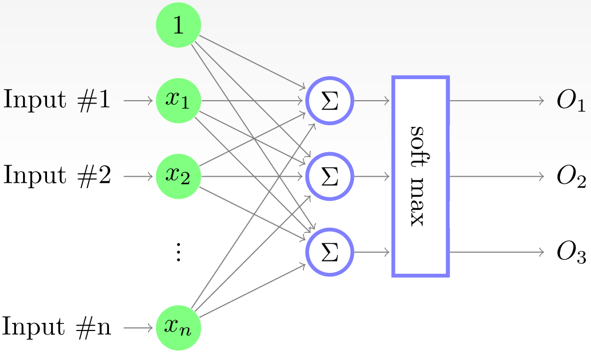


$$\text{Output} = [O_1, O_2, O_3] = f_{act}(Wx) = \text{sign}(Wx)$$

Multiclass Perceptron with softmax!



Multiclass Perceptron with softmax!



$$\text{Output} = [O_1, O_2, O_3] = \text{softmax}(Wx)$$