

# TECNOLOGICO NACIONAL DE MEXICO

INSTITUTO TECNOLOGICO DE MORELIA

“José María Morelos y Pavón”

Inteligencia Artificial

Profesor: Alcaraz Chávez Jesús Eduardo

## Proyecto Final Inteligencia Artificial

Carlos Jahir Castro Cázares

17120151

Ingeniería en Sistemas Computacionales

Semestre Marzo-Julio 2021

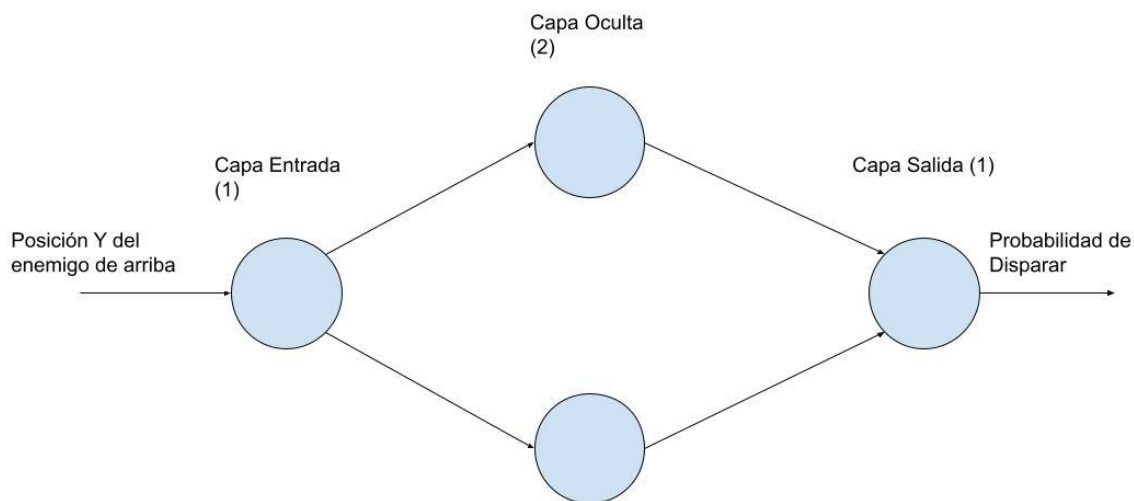
11 de Julio de 2021

## Videojuego 1 Space Invaders

### Red neuronal que dispara

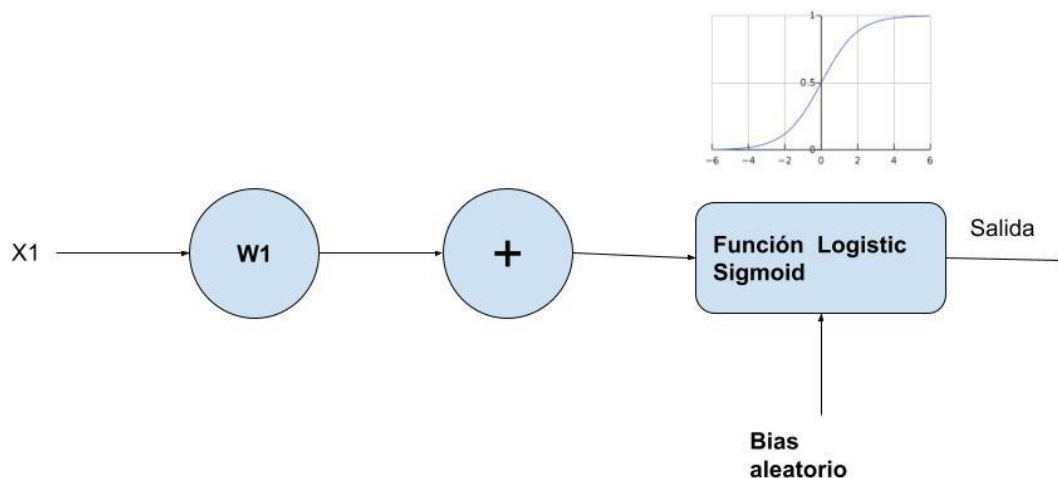
Diagrama del modelo que se utilizó para esta neurona es la siguiente 1 capa de entrada de una neurona, 1 capa oculta de 2 neuronas y 1 capa de salida de 1 neurona.

#### Modelo Red Neuronal Ataque Juego 1 (Space Invaders)

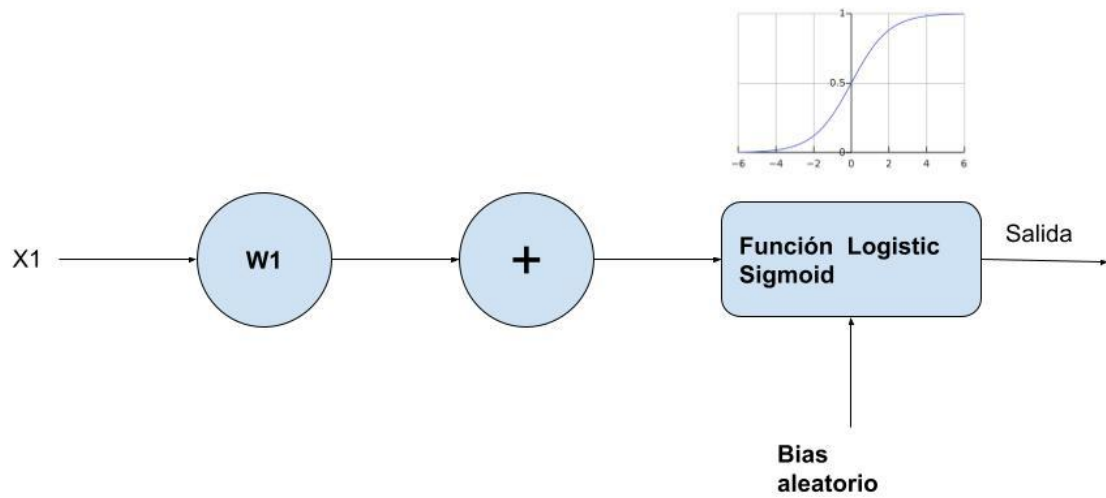


Dentro de las neuronas que conforman la red, se encuentran los **pesos sinápticos ( $w$ )**, el **bias (por default es random)** y la **función de activación (por default sigmoide)**, que corresponde a cada capa y su estructura según la capa es la siguiente:

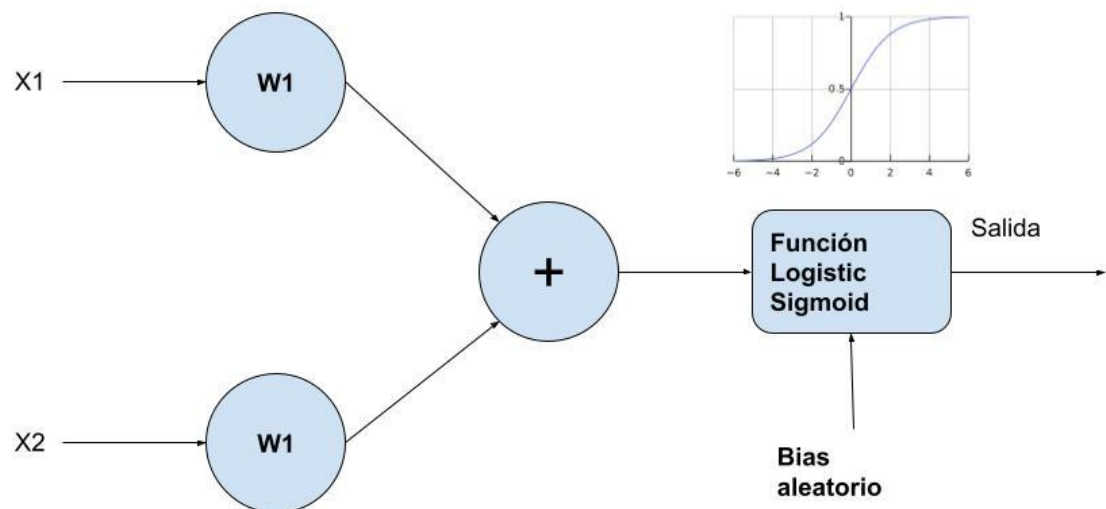
#### Neurona capa entrada



### Neurona capa oculta



### Neurona capa salida



La red neuronal corresponde a varios tipos, que estos corresponden a las siguientes según su número de capas es **multicapa** ya que usamos una capa de entrada, una de salida y una oculta; Según el tipo de conexiones es una **no recurrente** ya que la propagación se produce en un solo sentido; Finalmente por su grado de conexión es una **totalmente conectada** ya que por default sináptico proyecta todas las neuronas de una capa con la de la otra.

La estructura del dataset para esta neurona es el siguiente:

- **Input:** [posición en y del enemigo más cercano]
- **Ouput:** [0 o 1]

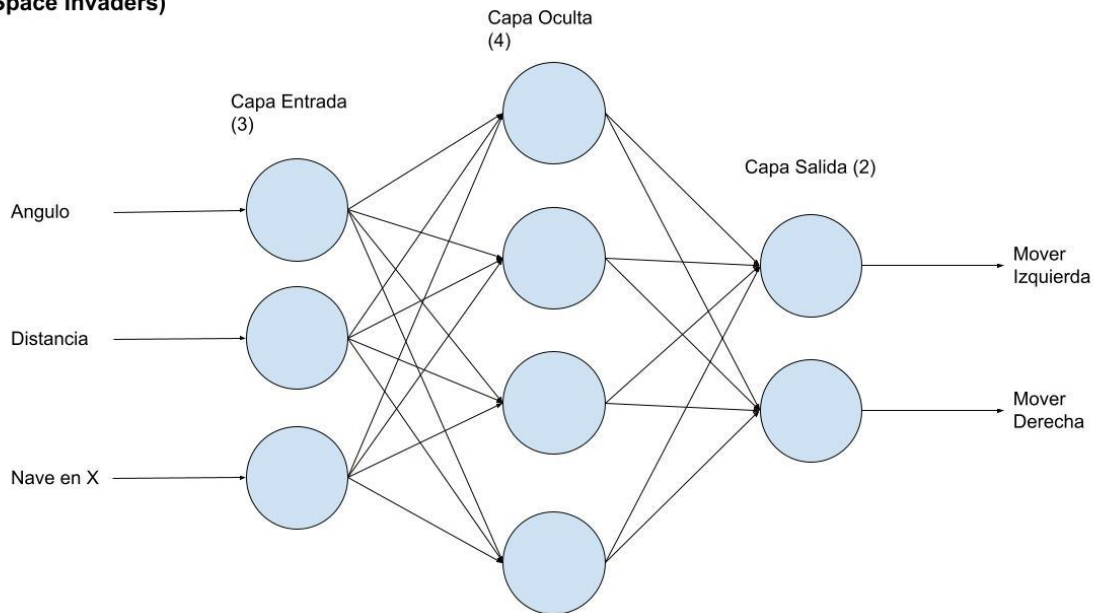
Como entrada uso la posición en y del enemigo mas cercano o el que esta justo encima del jugador ya que esta nos indicará si esta en la 1, 2 o 3 línea y si no encuentra un enemigo arrojará un -100; Como salida se dará un 0 si es que no se debe disparar y un 1 si se debe disparar.

Con estos datos aprenderá a diferenciar que los números iguales o mayores que cero siempre tendrá que disparar y con valores negativos no deberá de hacer nada.

## Red neuronal que esquiva

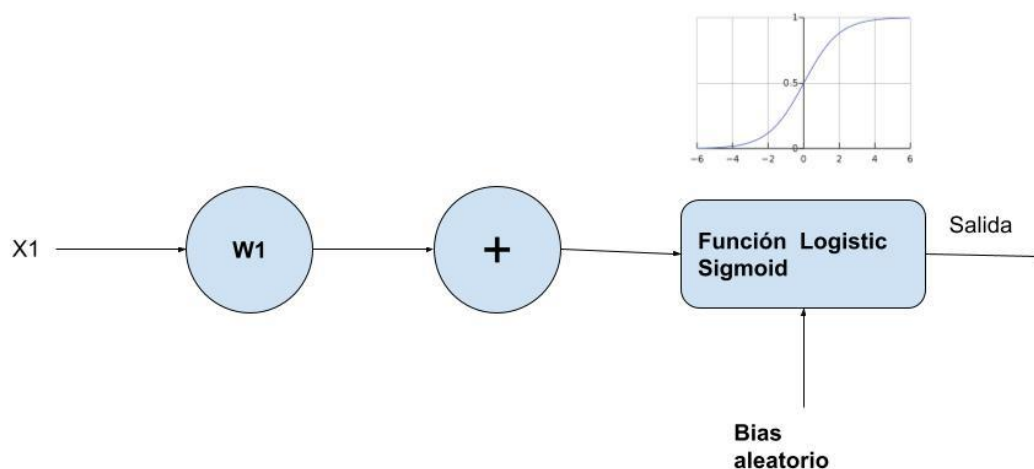
Diagrama del modelo que se utilizó para esta neurona es la siguiente 1 capa de entrada de 3 neuronas, 1 capa oculta de 4 neuronas y 1 capa de salida de 2 neuronas

### Modelo Red Neuronal Defensa Juego 1 (Space Invaders)

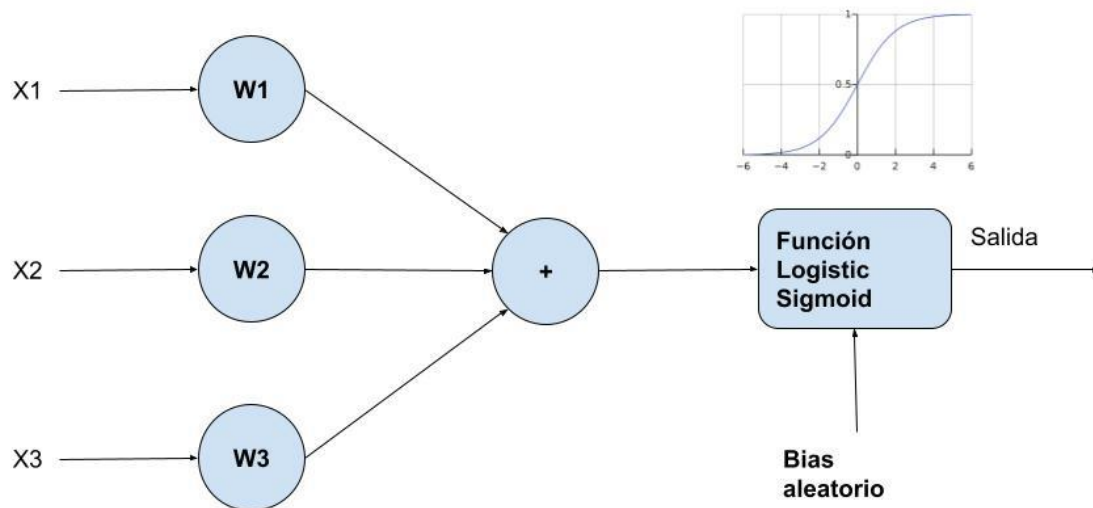


Dentro de las neuronas que conforman la red, se encuentran los **pesos sinápticos ( $w$ )**, el **bias (por default es random)** y la **función de activación (por default sigmoide)**, que corresponde a cada capa y su estructura según la capa es la siguiente:

### Neurona capa entrada

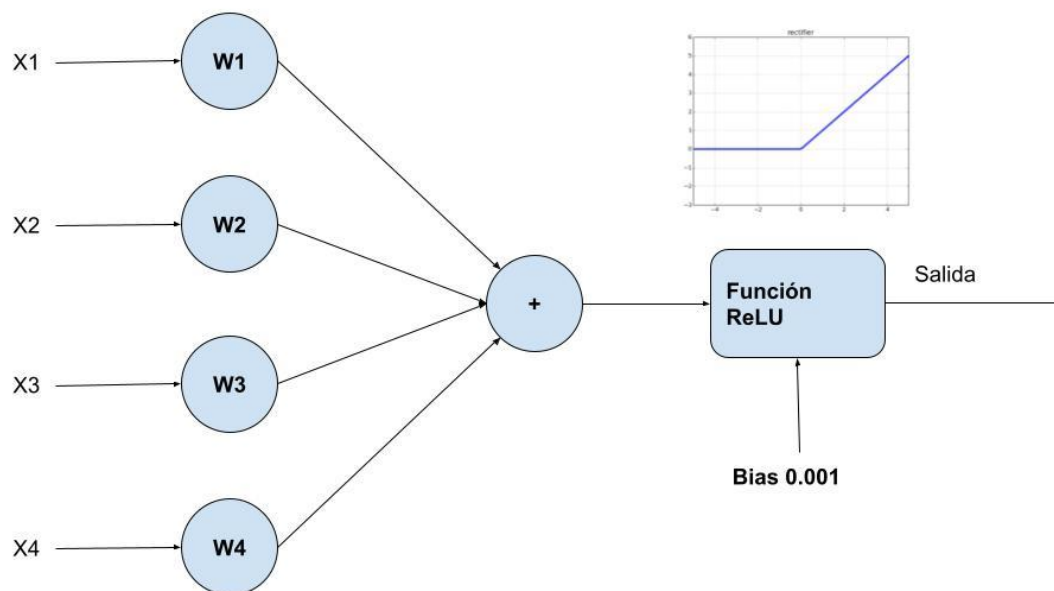


### Neurona capa oculta



En el ultimo caso para la capa de salida, la estructura de la neurona es la siguiente lo que difiere de las anteriores es que se cambió a una función de activación ReLU (ya que esta funciona mejor para las probabilidades) y movi el bias manual a 0.001

### Neurona capa salida



La red neuronal corresponde a varios tipos, que estos corresponden a las siguientes según su número de capas es **multicapa** ya que usamos una capa de entrada, una de salida y una oculta; Según el tipo de conexiones es una **no recurrente** ya que la propagación se produce en un solo sentido; Finalmente por su grado de conexión es una **totalmente conectada** ya que por default sinaptic proyecta todas las neuronas de una capa con la de la otra.

La estructura del dataset para esta neurona es el siguiente:

- **Input:** [Angulo, distancia, posición en x]
- **Ouput:** [0,0] o [1,0] o [0,1]

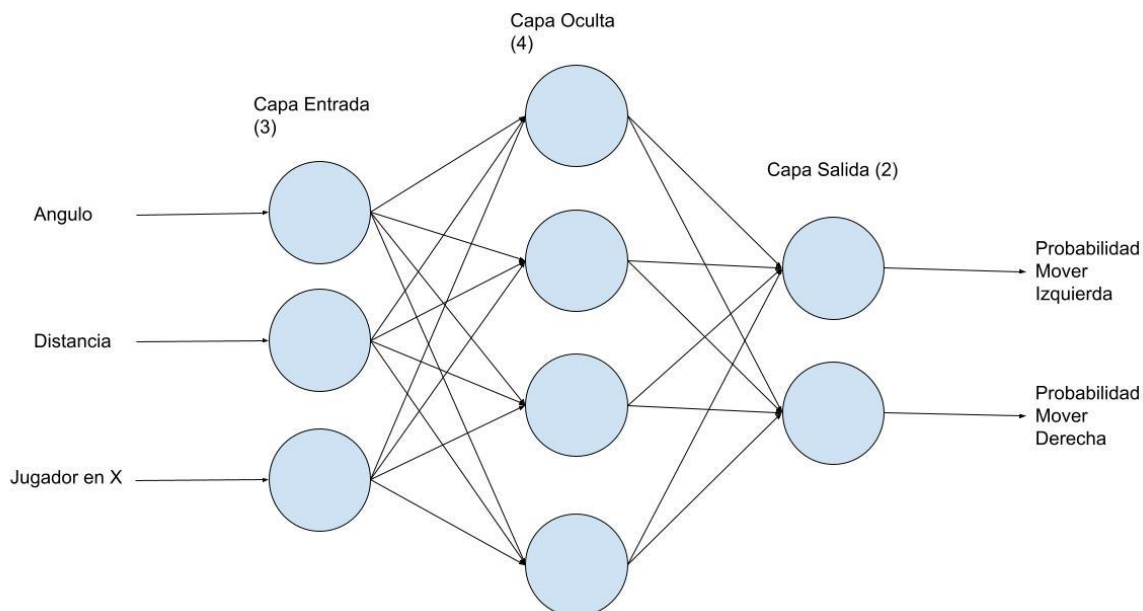
Para los datos de entrada use primeramente el **ángulo** que se forma entre la nave y la bala, esto nos indicará si la bala esta en la derecha, izquierda o arriba y en que inclinación estará según su grado de 0 – 180, seguidamente la **distancia** que se encuentra entre la nave y la bala esto nos indicara que tan lejos o cerca esta la bala al impactar con la nave, finalmente el dato de posición de **x de la nave**, que este nos indicara donde se encuentra la nave ya que en ocasiones aunque tengan el mismo ángulo y distancia se puede mover a otra dirección por las esquinas del mapa, este es un dato diferenciador que ayuda a separar más fácil los datos en las 2 categorías.

Como salidas usamos una por cada lado izquierda y derecha, lo cual cuando este detenido será 0,0, a la izquierda 1,0 y a la derecha 0,1. Lo cual nos dará como salida la probabilidad de que se mueva a un lado o a otro.

## Videojuego 2 Esquivar oponentes

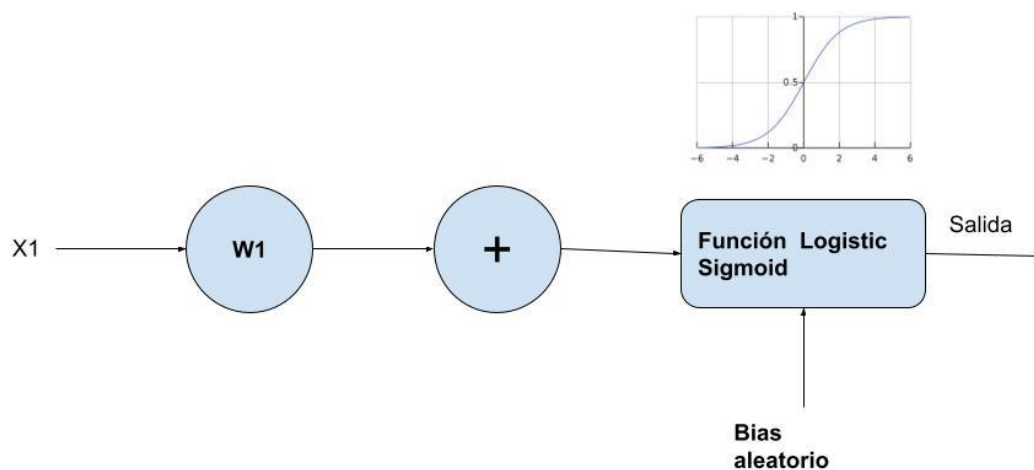
Diagrama del modelo que se utilizó para esta neurona es la siguiente 1 capa de entrada de 3 neuronas, 1 capa oculta de 4 neuronas y 1 capa de salida de 2 neuronas.

### Modelo Red Neuronal Juego 2 (Esquivar Oponentes)



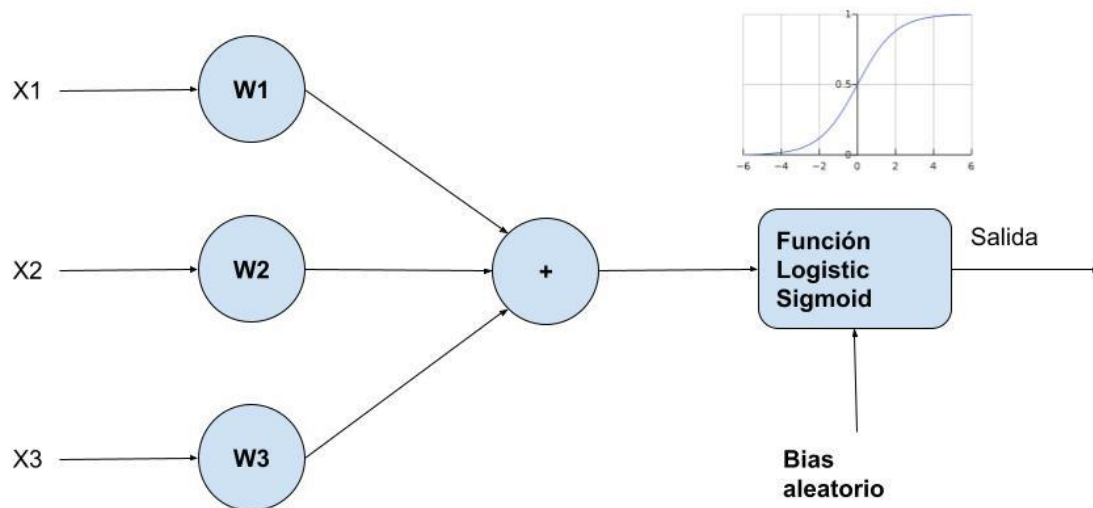
Dentro de las neuronas que conforman la red, se encuentran los **pesos sinápticos ( $w$ )**, el **bias (por default es random)** y la **función de activación (por default sigmoide)**, que corresponde a cada capa y su estructura según la capa es la siguiente:

### Neurona capa entrada



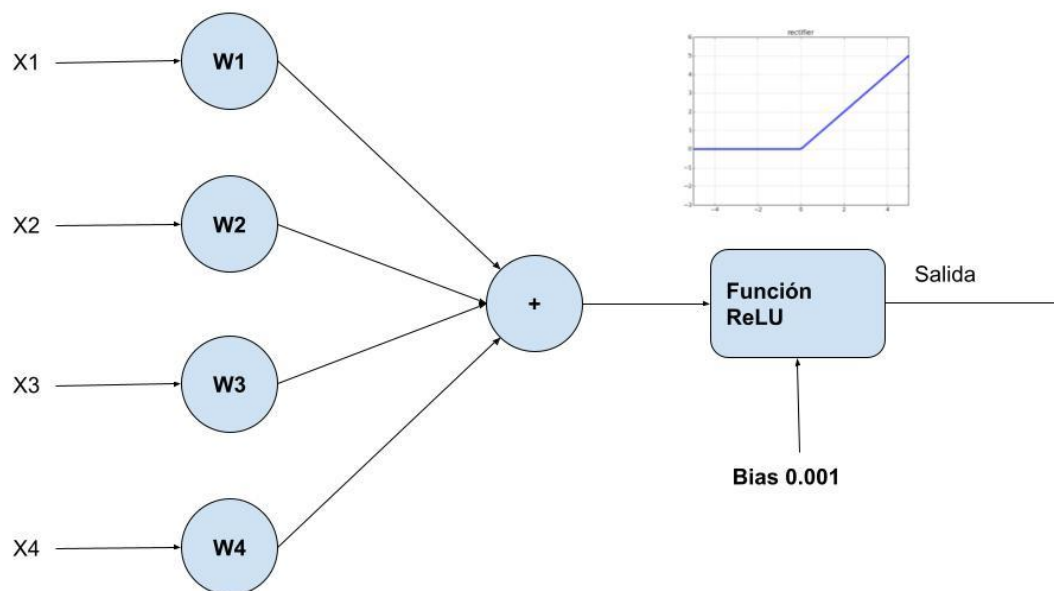


### Neurona capa oculta



En el último caso para la capa de salida, la estructura de la neurona es la siguiente lo que difiere de las anteriores es que se cambió a una función de activación ReLU (ya que esta funciona mejor para las probabilidades) y movi el bias manual a 0.001

### Neurona capa salida



La red neuronal corresponde a varios tipos, que estos corresponden a las siguientes según su número de capas es **multicapa** ya que usamos una capa de entrada, una de salida y una oculta; Según el tipo de conexiones es una **no recurrente** ya que la propagación se produce en un solo sentido; Finalmente por su grado de conexión es una **totalmente conectada** ya que por default sináptico proyecta todas las neuronas de una capa con la de la otra.

La estructura del dataset para esta neurona es el siguiente:

- **Input:** [Angulo, distancia, posición en x]
- **Ouput:** [0,0] o [1,0] o [0,1]

Para los datos de entrada use primeramente el **ángulo** que se forma entre la nave y la bala, esto nos indicará si la bala está en la derecha, izquierda o arriba y en que inclinación estará según su grado de 0 – 180, seguidamente la **distancia** que se encuentra entre la nave y la bala esto nos indicara que tan lejos o cerca esta la bala al impactar con la nave, finalmente el dato de posición de **x de la nave**, que este nos indicara donde se encuentra la nave ya que en ocasiones aunque tengan el mismo ángulo y distancia se puede mover a otra dirección por las esquinas del mapa, este es un dato diferenciador que ayuda a separar más fácil los datos en las 2 categorías.

Como salidas usamos una por cada lado izquierda y derecha, lo cual cuando este detenido será 0,0, a la izquierda 1,0 y a la derecha 0,1. Lo cual nos dará como salida la probabilidad de que se mueva a un lado o a otro.

***En este ultimo caso se uso la misma neurona del juego anterior, pero al tener mas enemigos en el campo, lo que se hace es hacer n entradas en el dataset de acuerdo al numero de enemigos que detecta, así que debes de tener una solo entrada con una salida, tendremos n entradas diferentes con la misma salida.***