

SIA TP3

TP Perceptron

Introducción

El trabajo práctico consiste de tres ejercicios.

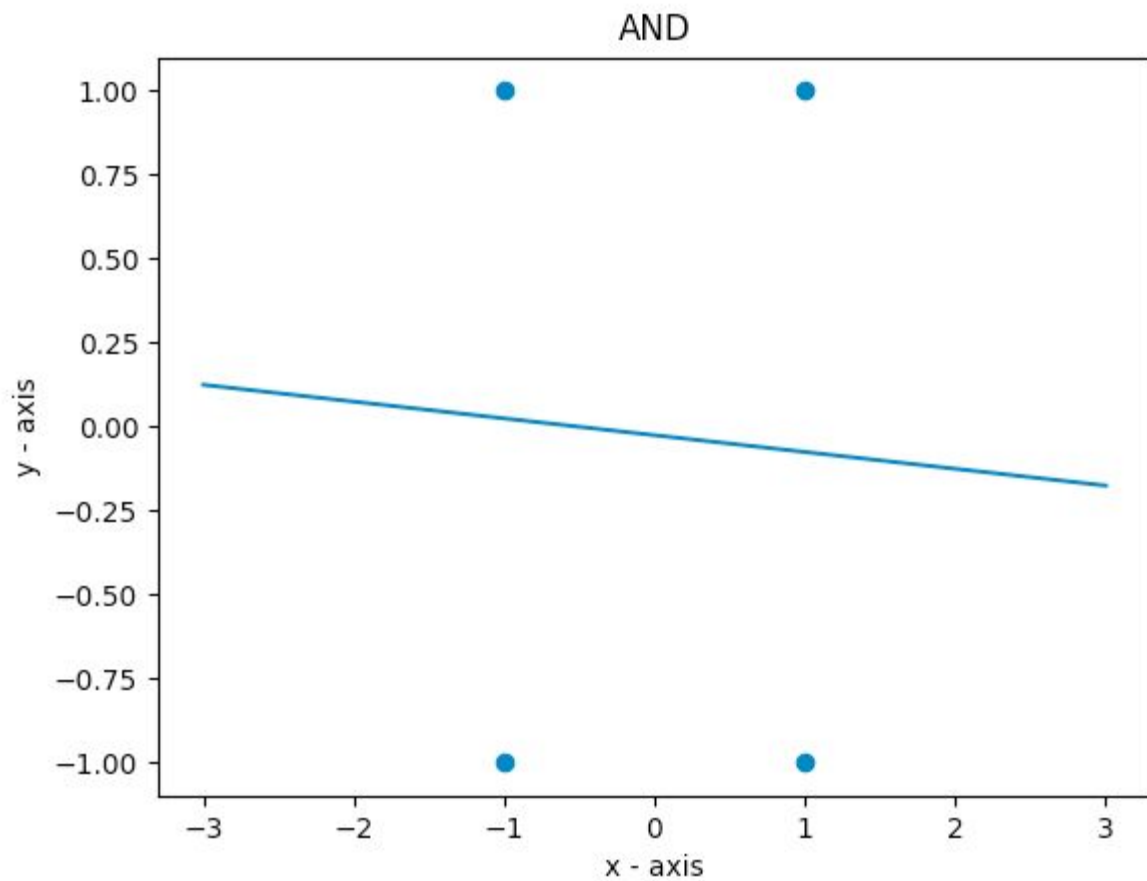
Los ejercicios 1.a y 1.b son un perceptrón simple sencillo con función escalón que aprende las funciones lógicas AND y OR. La función XOR no puede aprenderla ya que no existe un separador lineal.

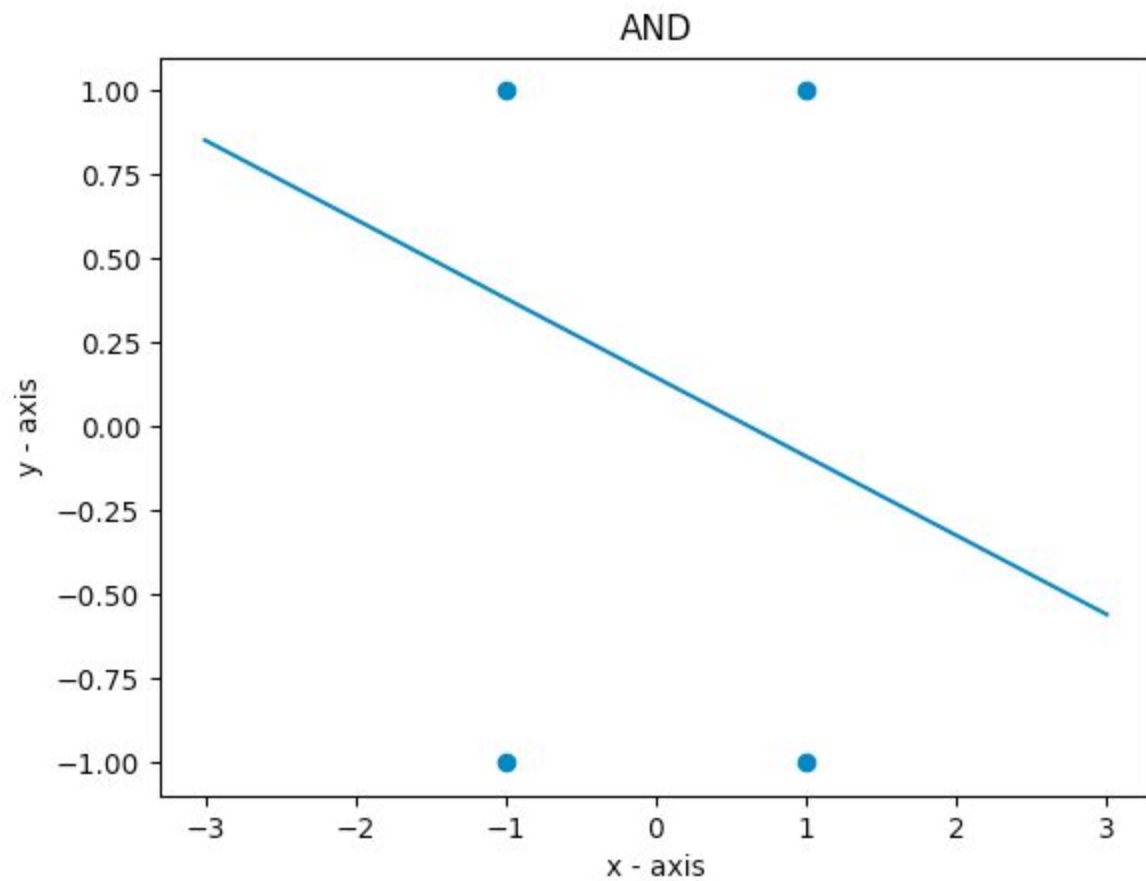
El ejercicio 2 es un perceptrón no lineal.

El ejercicio 3.a aprende la función lógica XOR. El ejercicio 3.b está relacionado con primalidad.

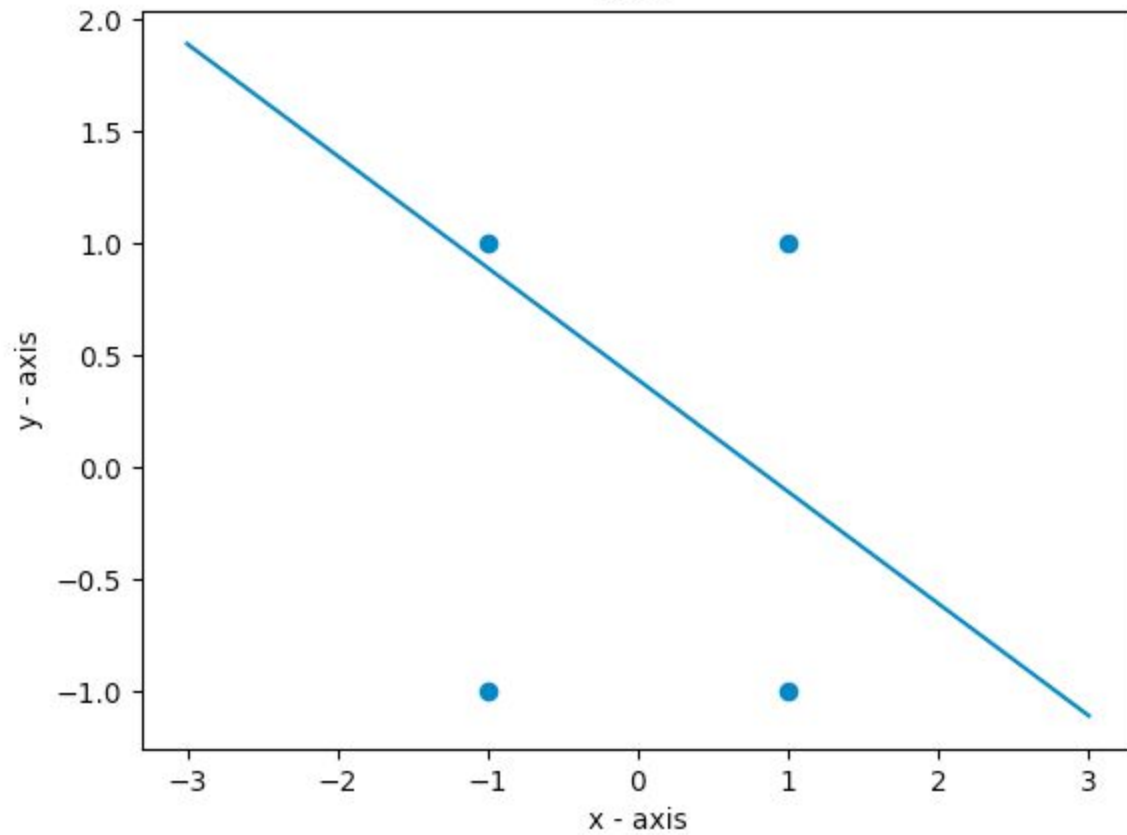
Resultados

```
ej1a=FALSE
ej1a_epochs=10
ej1a_learningrate=0.03
ej1a_ploteachepoch=TRUE
ej1a_stopearly=TRUE
ej1b=FALSE
ej1b_epochs=10
ej1b_learningrate=0.5
ej1b_ploteachepoch=TRUE
ej1b_stopearly=TRUE
ej2=TRUE
ej3a=FALSE
ej3b=FALSE
num_epochs=10000
learning_rate=0.5
beta=0.4
hidden_perceptrons=150
max_error_ej3=0.01
```

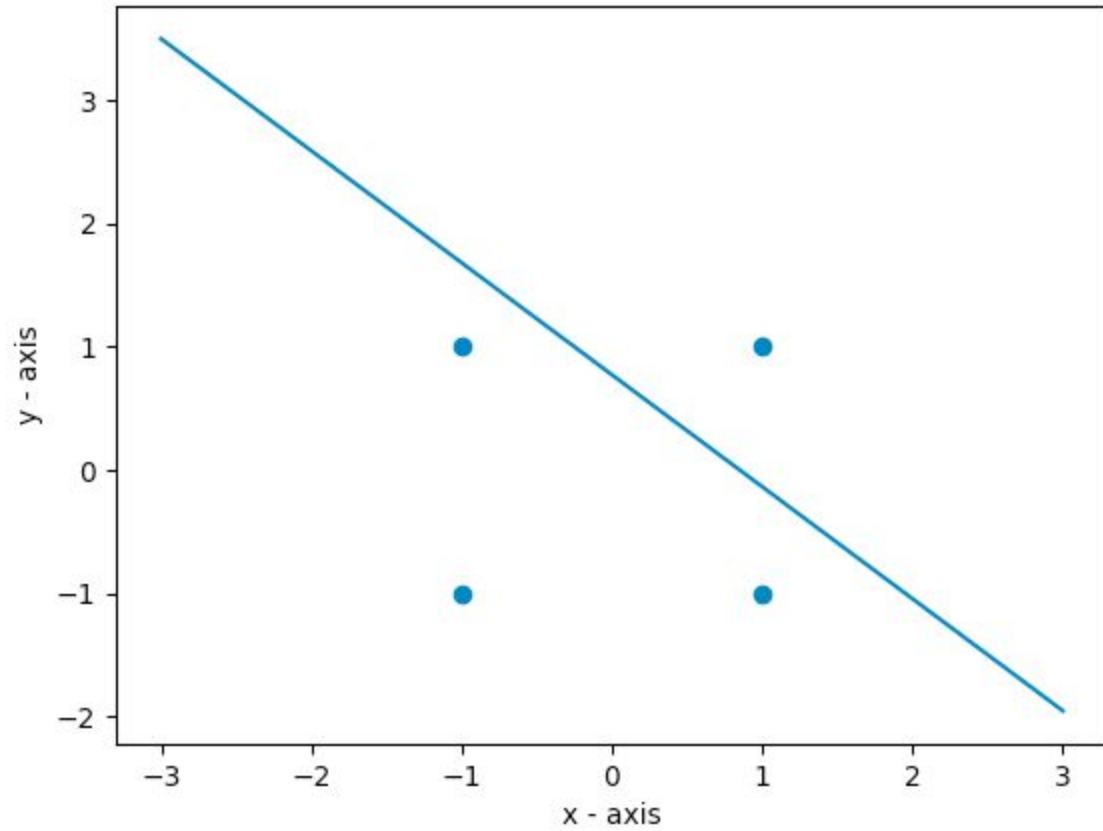




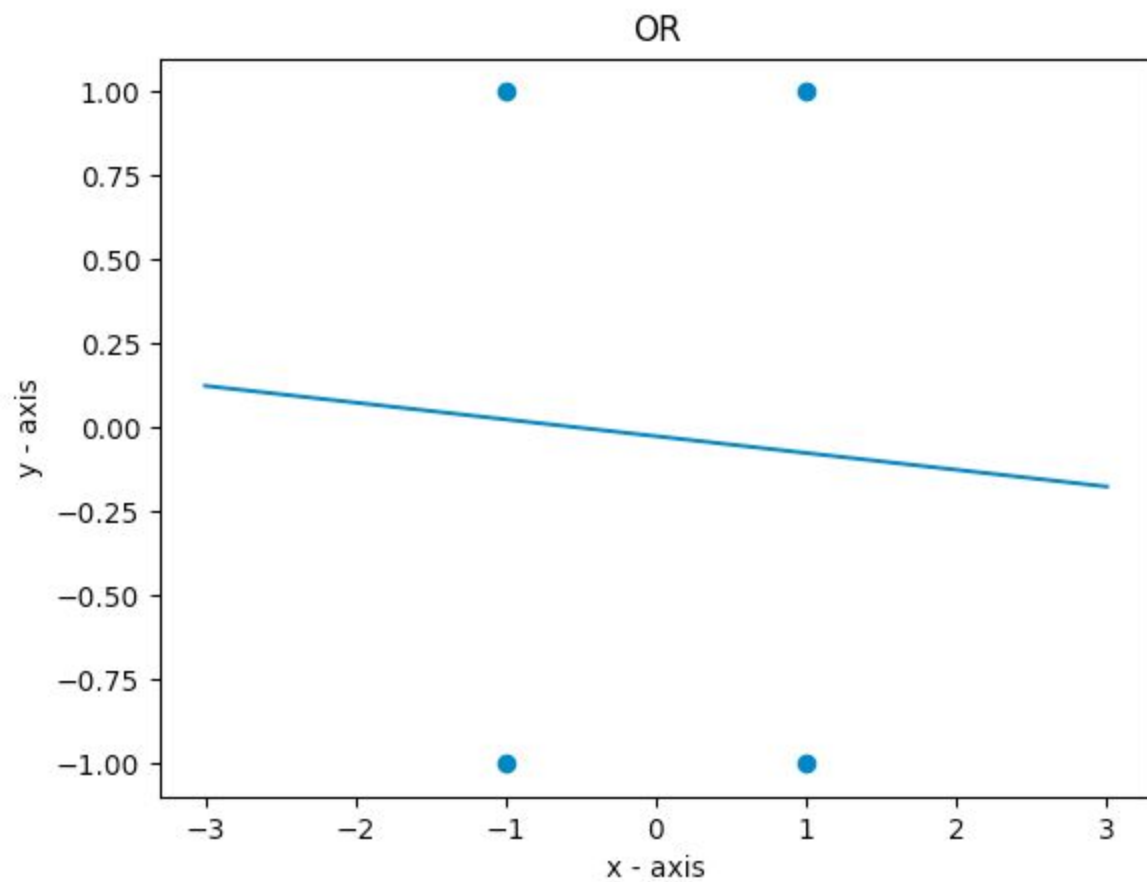
AND



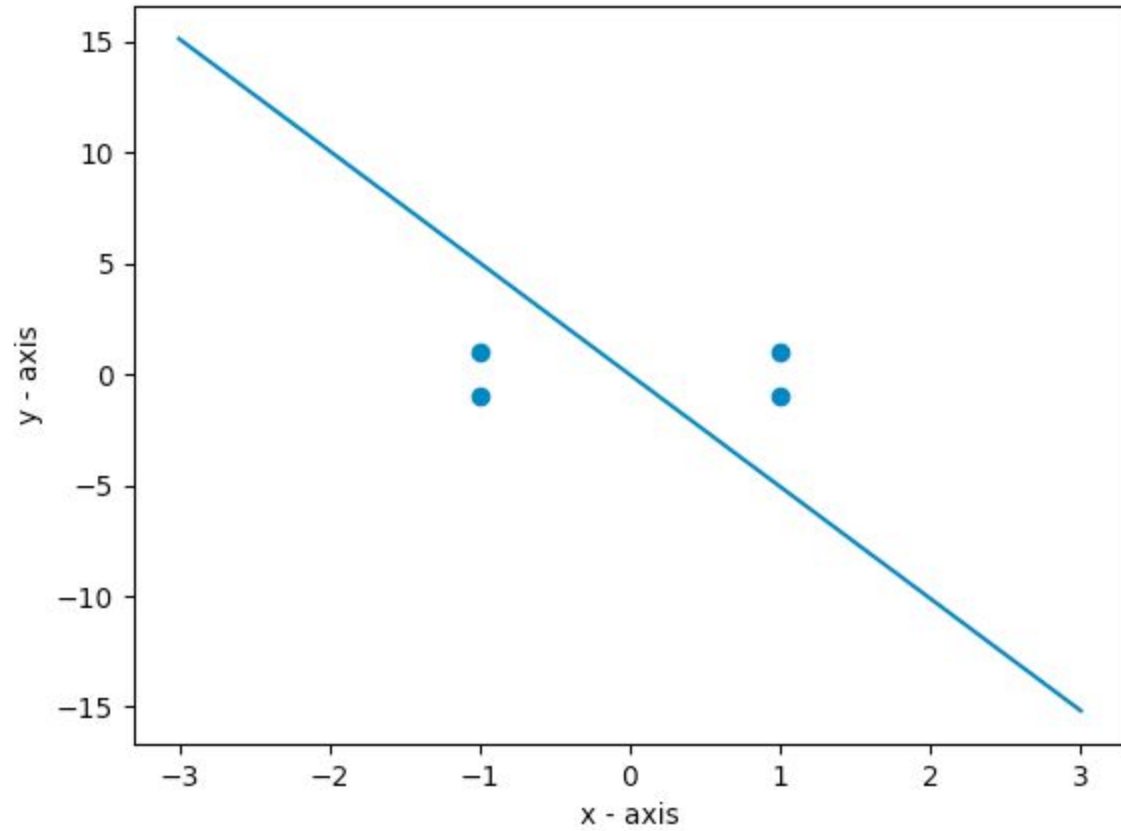
AND



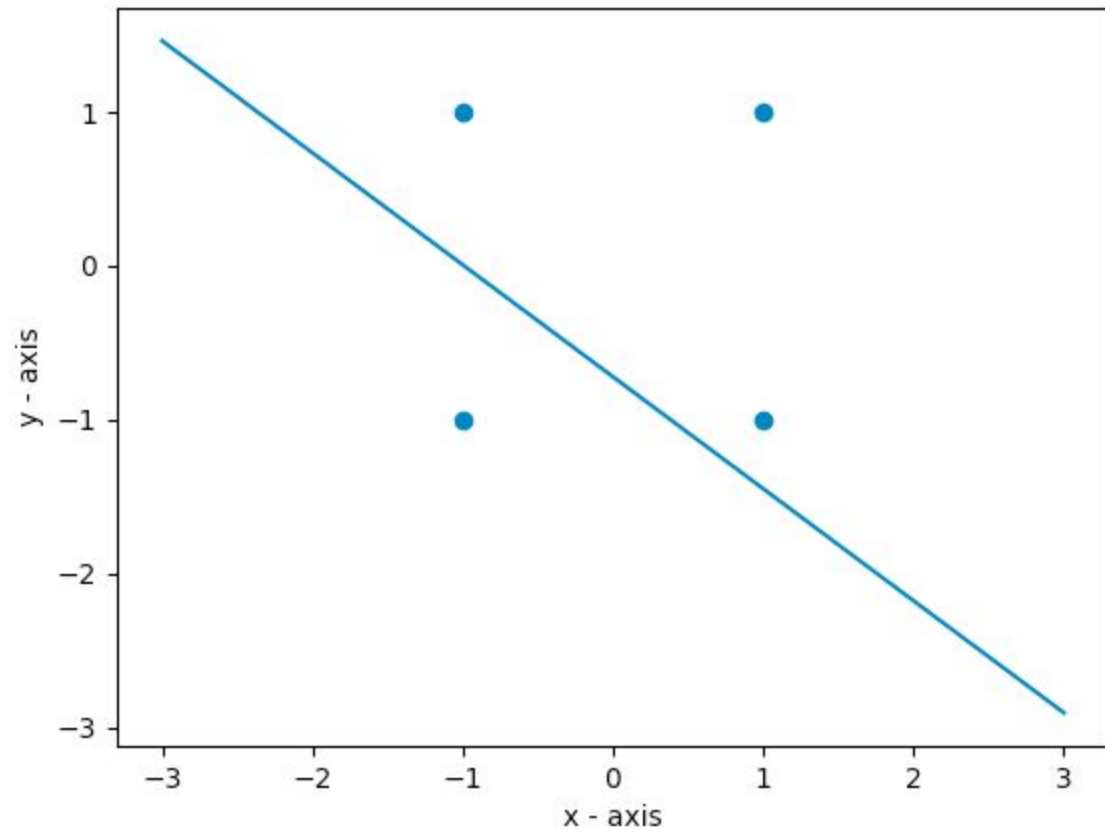
ej1b=TRUE
ej1b_epochs=10
ej1b_learningrate=0.5
ej1b_ploteachepoch=TRUE
ej1b_stopearly=TRUE



OR



OR



Qué puede decir acerca de los problemas que puede resolver el perceptrón simple escalón en relación a la resolución de los problemas que se le pidió que haga que el perceptrón aprenda?

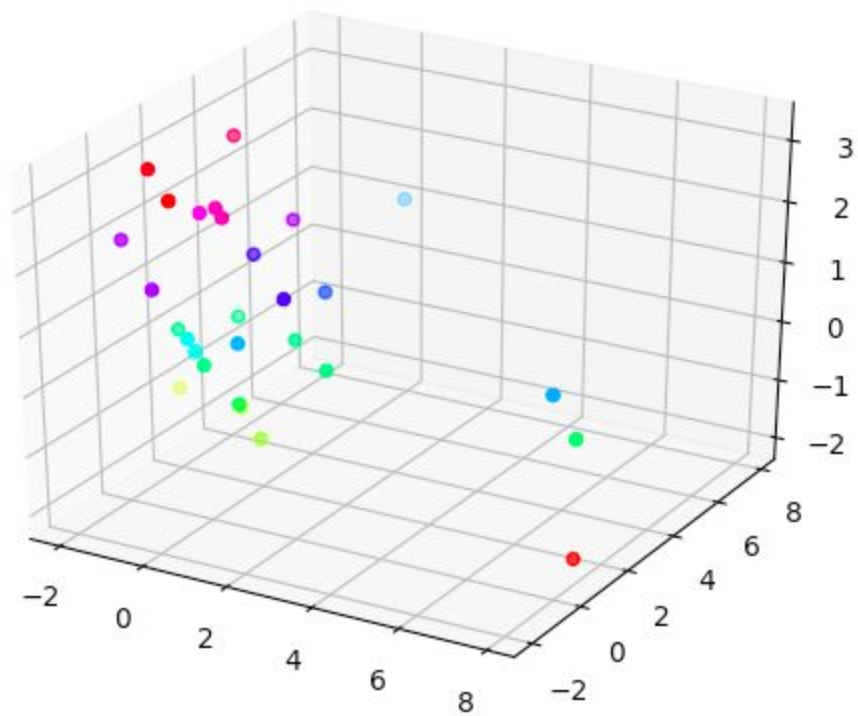
El AND y el OR son linealmente separables por lo tanto se obtiene una recta.

Cuando se intenta separar los datos del XOR con el perceptrón simple con función escalón nunca se obtiene un separador lineal ya que el problema no es linealmente separable.

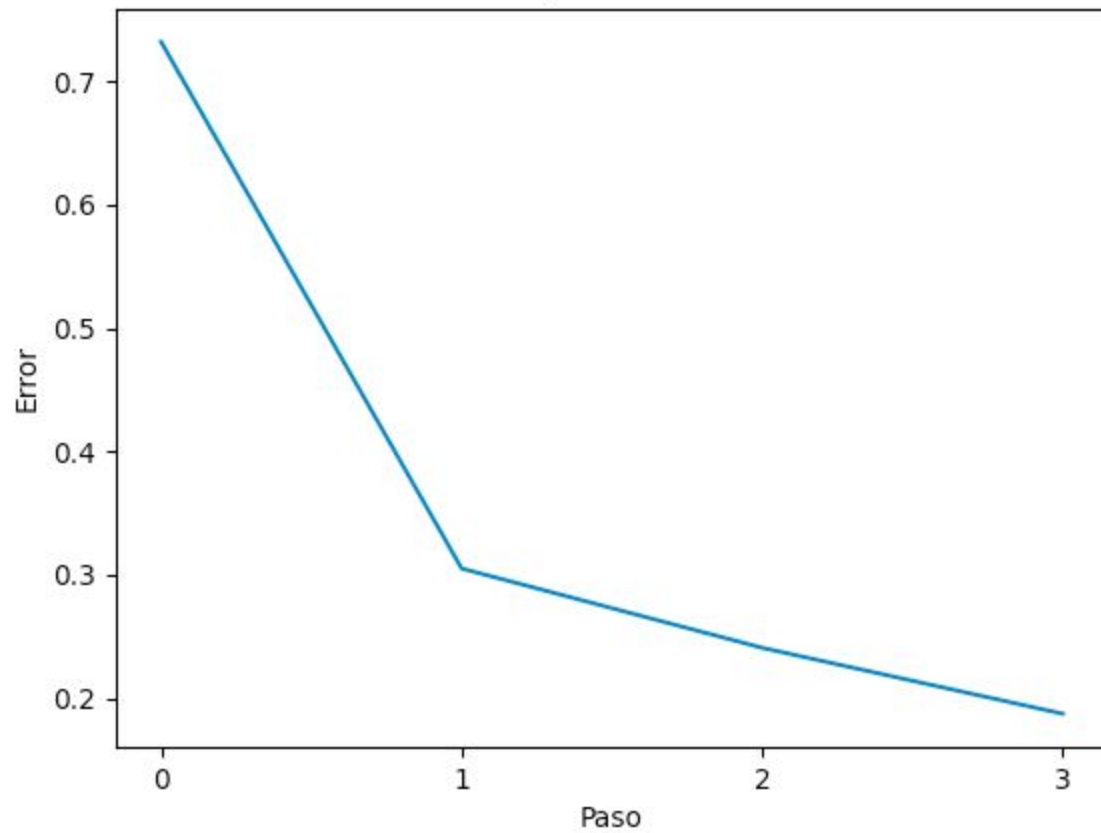
Lo que podemos decir es que el perceptrón puede aprender los problemas si son linealmente separables.

A continuación mostramos la disposición del set de datos de entrada para el ejercicio 2.

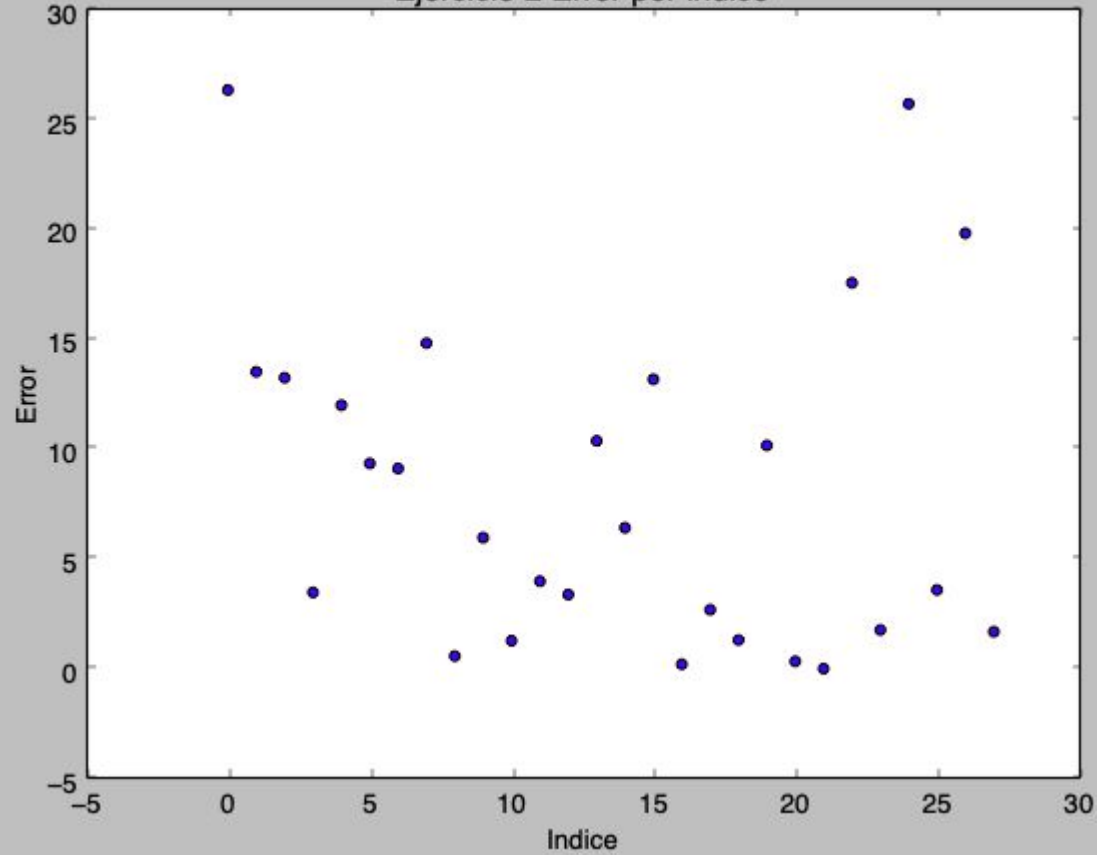
Para este ejercicio se normalizaron los datos de la entrada.

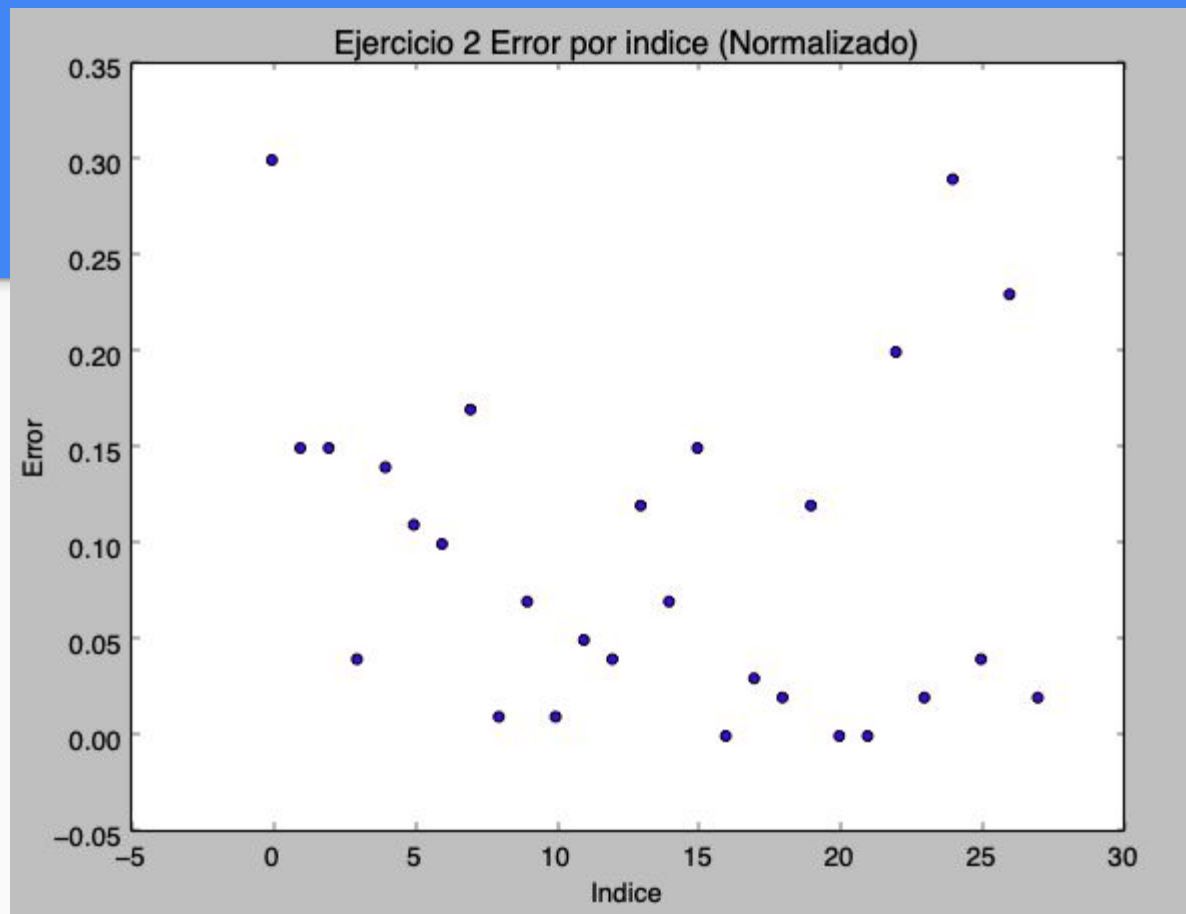


Ejercicio 2



Ejercicio 2 Error por indice



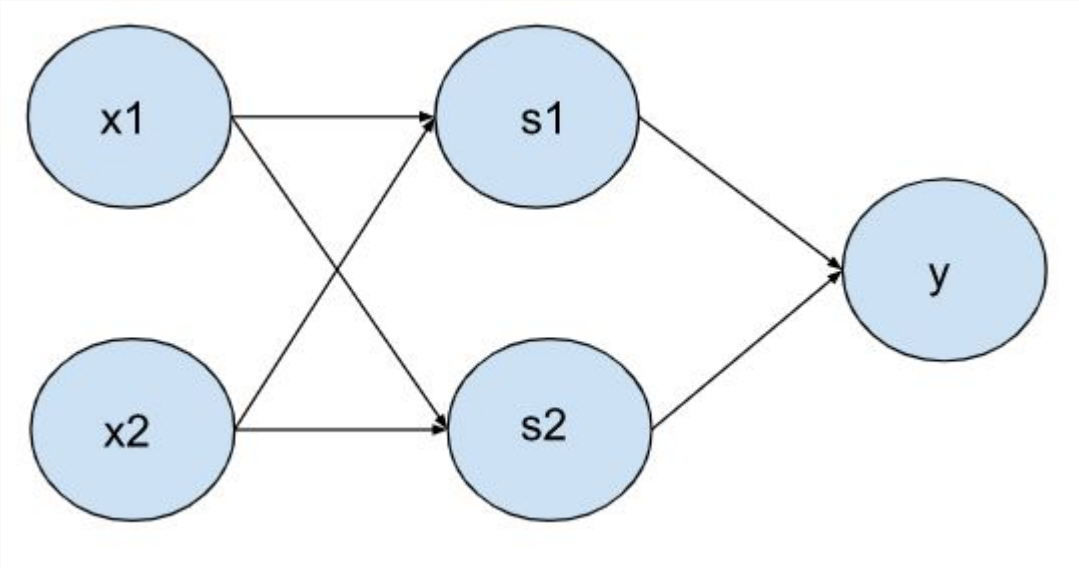


Cómo podría escoger el mejor conjunto de entrenamiento?
Cómo podría evaluar la máxima capacidad de generalización del
perceptrón para este conjunto de datos?

Para escoger el mejor conjunto de entrenamiento lo que podemos hacer es tomar posibles particiones del conjunto de entrenamiento y elegir la que nos da el menor error.

Lá máxima capacidad de generalización para este conjunto de datos podríamos obtenerla entrenando con la totalidad del conjunto de entrenamiento.

Perceptrón Multicapa



- Vamos a tener dos perceptrones en la capa oculta junto con dos perceptrones que van a funcionar como entrada y un ultimo que llamaremos "output"
- Resolver el problema de XOR

Clases utilizadas

- Utilizamos como función de activación la tangente hiperbólica.
- Lo hicimos tal que podemos meter N perceptrones en capa oculta (valor variable)

```
public class Neuron {
    private MultiLayerType type;
    private double[] weights;
    private double threshold = .1, error = 0, output = 0, beta;

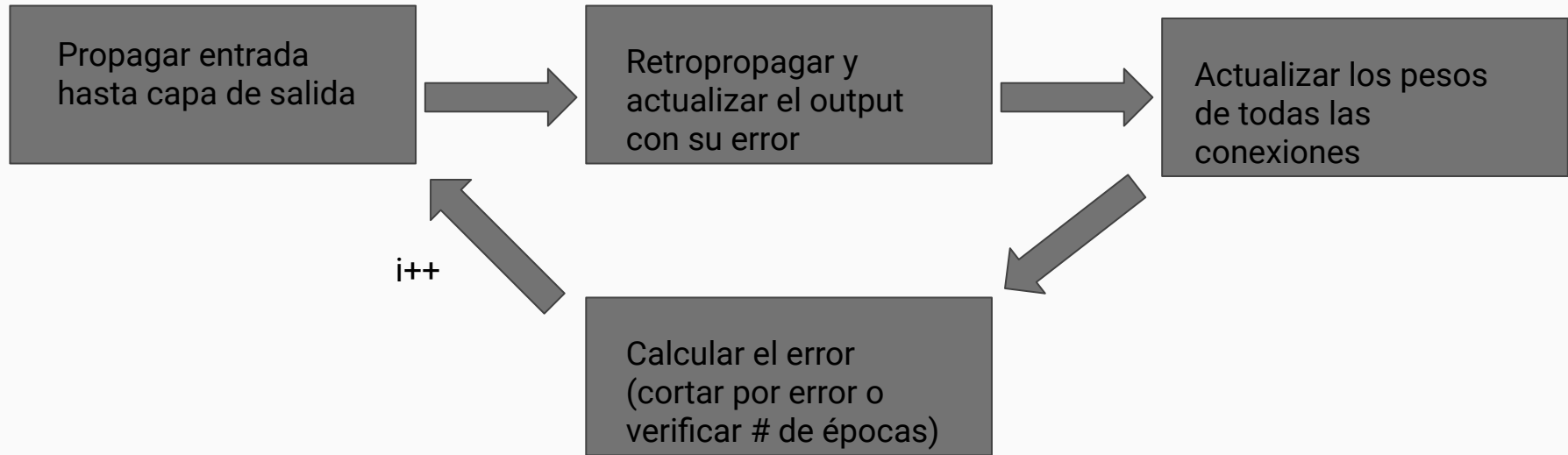
    public Neuron(MultiLayerType type, int nInput, int nHidden, double beta){...}
    public double derivative() { return (1 + output)*(1 - output); }
    public double[] getWeights() { return weights; }
    public double getThreshold() { return threshold; }
    public void setThreshold(double threshold) { this.threshold = threshold; }
    public double getError() { return error; }
    public void setError(double error) { this.error = error; }
    public double getOutput() { return output; }
    public void setOutput(double output) { this.output = output; }
    public double activationFunction(double sum){
        output = Math.tanh(beta * sum);
        return output;
    }
    public void updateWeights(int index, double deltaW) { weights[index] += deltaW; }
}
```

```
public class MultiLayerPerceptronGeneral {
    private int nInputs, nHidden, nEpochs;
    private Neuron[] hiddenPerceptrons, inputPerceptrons;
    private double learning_rate;
    private Neuron output;

    public Neuron getOutput() { return output; }
    public MultiLayerPerceptronGeneral(int nInputs, int nHidden, int nEpochs, double learning_rate, double beta){...}
    public void train(double[] input){...}

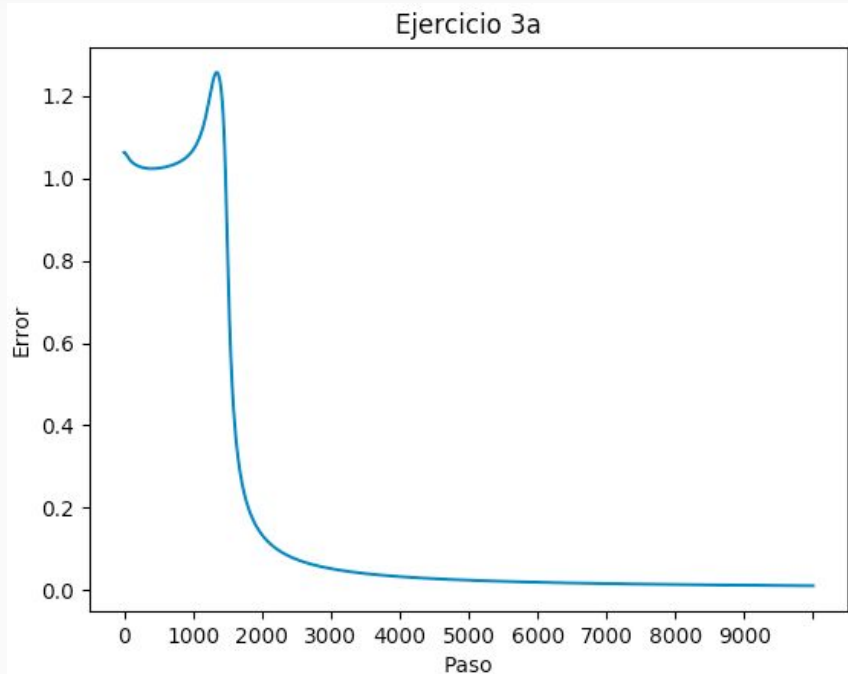
    private double dotProduct(double[] weights, Neuron[] perceptrons) {...}
    public void updateError(double expectedResult){...}
}
```

Consideración de la implementación

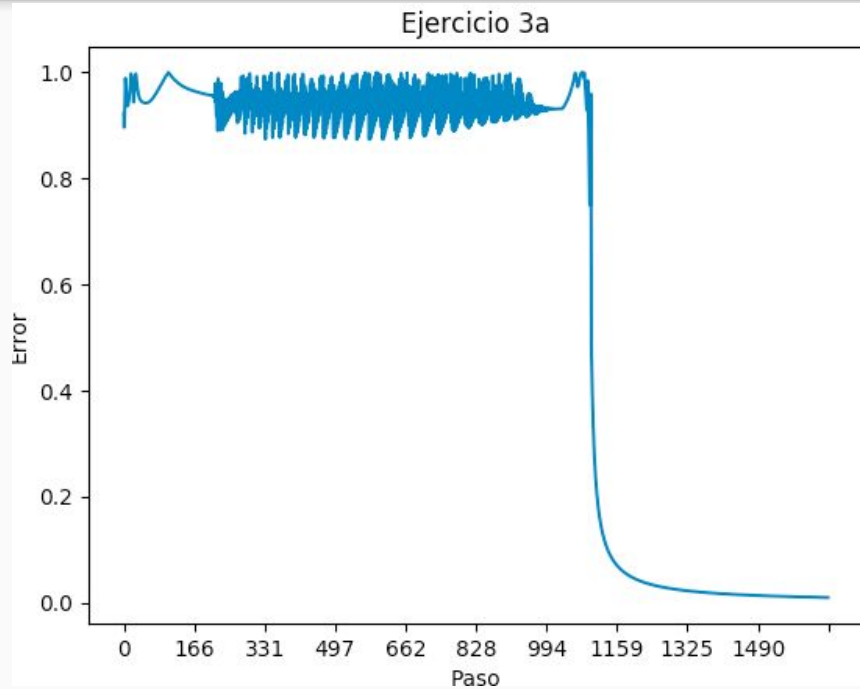


XOR Gráficos

beta = 0.1

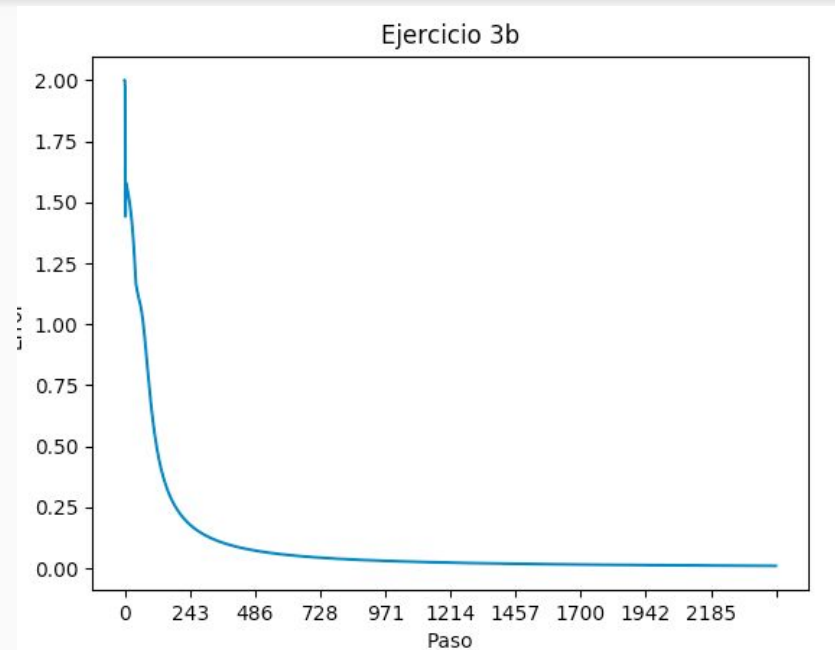
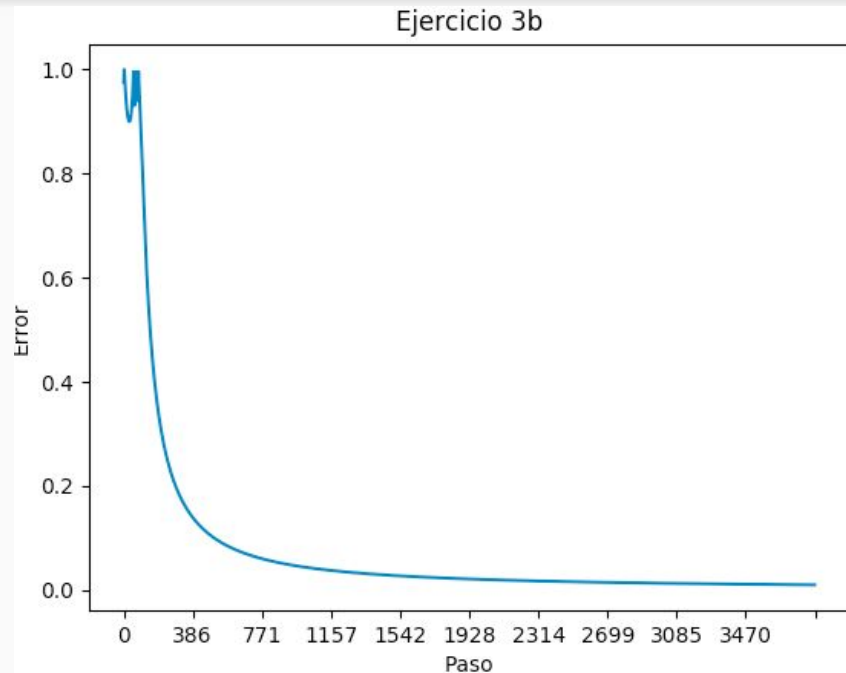


beta = 0.8

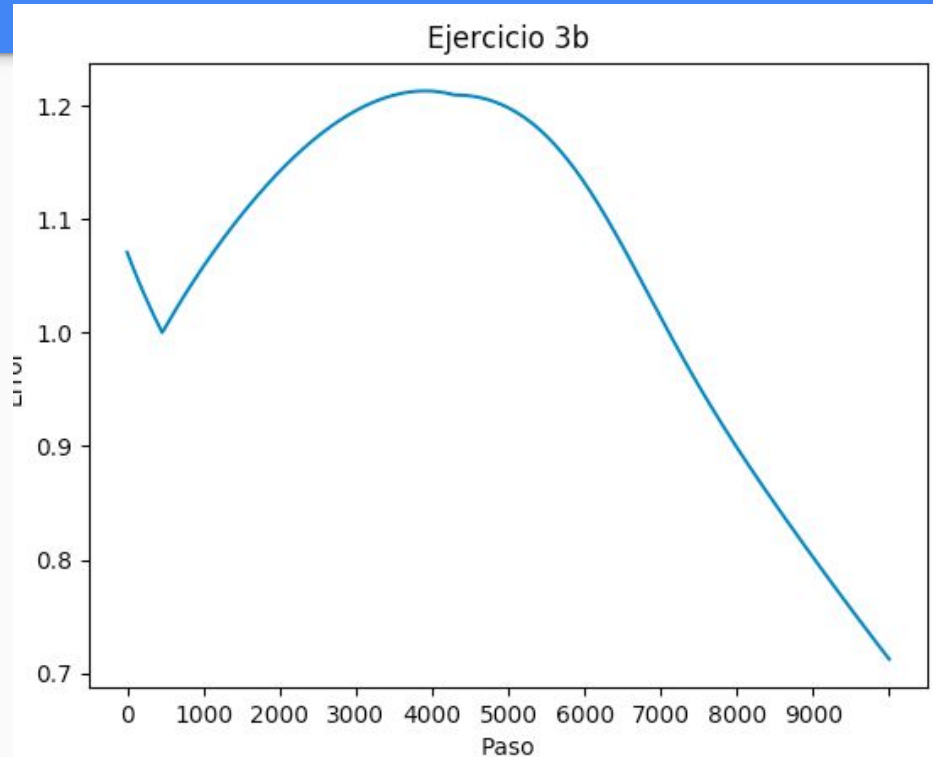


Imágenes gráficas

Con 2 y 150 perceptrones en capa oculta



Con un beta muuuuy bajo



Salida para las imágenes

Epoch 2427

For image 0 the output is -0.9959683442358654

For image 1 the output is -0.997144272440262

For image 2 the output is 0.998829616138576

For image 3 the output is 0.9900013404601621

For image 4 the output is -0.9999363600592163

For image 5 the output is 0.9966600185563725

For image 6 the output is -0.9999926717886188

For image 7 the output is 0.9984910861995345

For image 8 the output is -0.9933888626910639

For image 9 the output is -0.9979115131599824

- -1 Representa no primo
- 1 representa primo

Predecir un valor sin pasar por el proceso de entrenamiento

```
Epoch 44  
For image 0 the output is -0.9303737733403626  
For image 1 the output is -0.9961412328904683  
For image 2 the output is 0.9999430557107323  
For image 3 the output is 0.9916482070974112  
For image 4 the output is -0.9999999213173715  
For image 5 the output is 0.0  
For image 6 the output is -0.9999988843971059  
For image 7 the output is 0.9999907365535962  
For image 8 the output is -0.9741687790213929  
For image 9 the output is -0.9569516005880734  
  
Predict 5 without going through training process: -0.10955321599166574
```

Cada vez que pasamos el nuevo dato sin pasarlo por el entrenamiento y la retropropagación del error, este valor tomará nuevos números aleatorios. Esto pasa ya que la forma del número no dice si un número es o no es primo.

Conclusiones

El perceptrón escalón depende fuertemente de la tasa de aprendizaje para poder conseguir un separador lineal de manera exitosa.

El perceptrón simple con función escalón no puede separar el conjunto de datos del OR exclusivo.

En el ejercicio 3.b no podemos afirmar que la red aprendió ya que si bien el error tiende a cero, no se puede generalizar a partir de estos datos si un número es primo o no."La forma de los números no nos dice nada de si estos son primos o no"