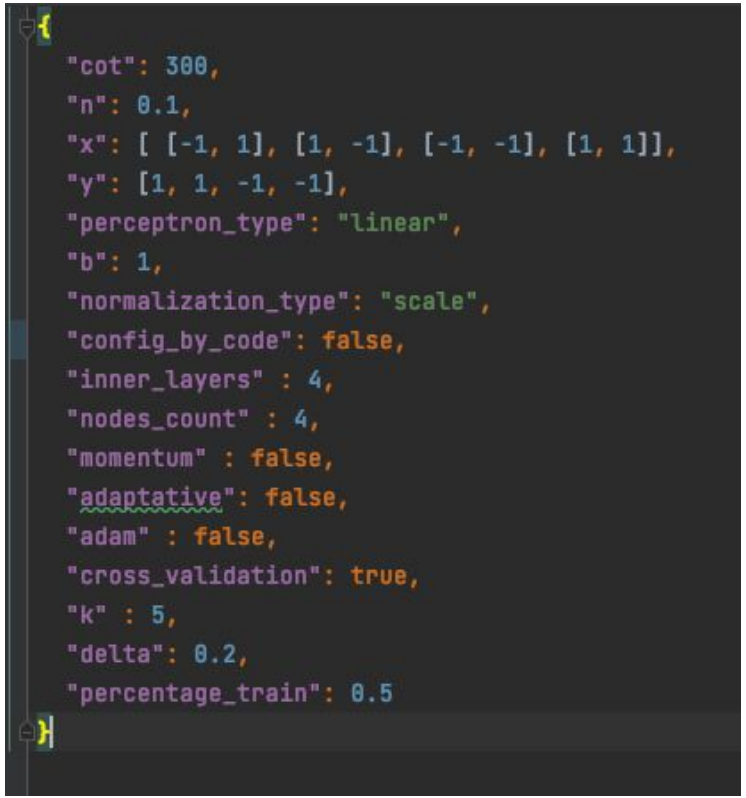
Sistemas de Inteligencia Artificial
TP2
Perceptrón Simple y Multicapa

Capart Micaela, Legajo Nº 60289
Ferraris Santiago, Legajo Nº 60129
Ruiz Mateo, Legajo Nº 60358

Consideraciones de Implementación

Configuración



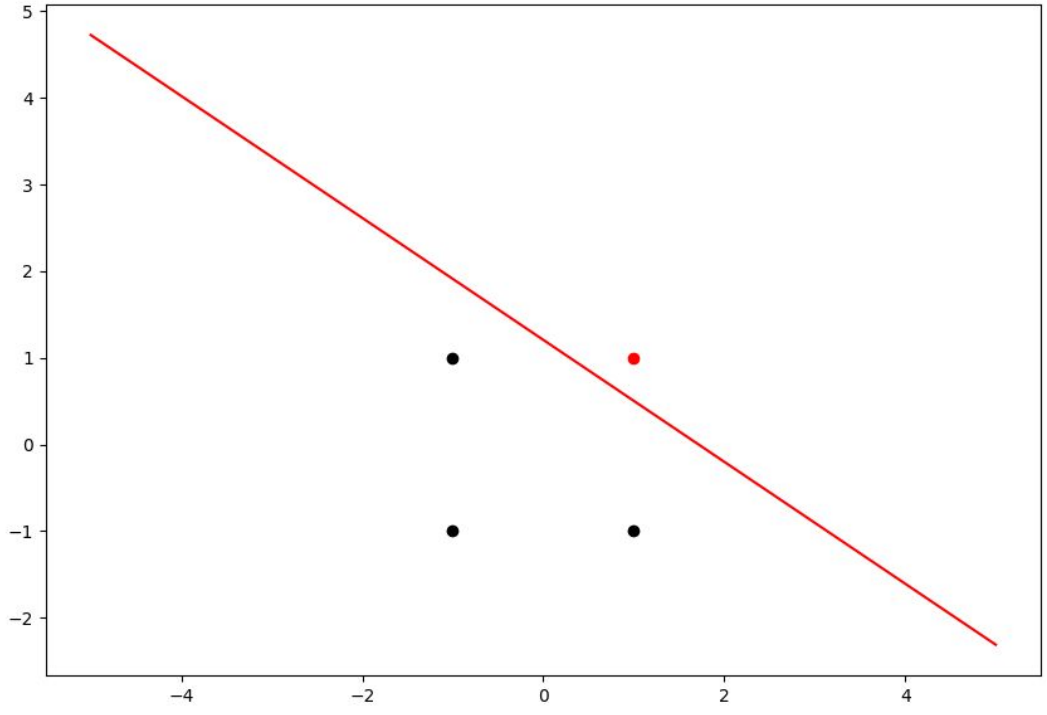
```
{
  "cot": 300,
  "n": 0.1,
  "x": [ [-1, 1], [1, -1], [-1, -1], [1, 1]],
  "y": [1, 1, -1, -1],
  "perceptron_type": "linear",
  "b": 1,
  "normalization_type": "scale",
  "config_by_code": false,
  "inner_layers" : 4,
  "nodes_count" : 4,
  "momentum" : false,
  "adaptative": false,
  "adam" : false,
  "cross_validation": true,
  "k" : 5,
  "delta": 0.2,
  "percentage_train": 0.5
}
```

- Perceptron_types:
  - step
  - linear
  - non-linear-tan
  - non-linear-logistic
  - multi-layer-xor
  - multi-layer-even
  - multi-layer-number
- Cross_validation solo aplica a linear, non-linear-tan, non-linear-logistic, multi-layer-even y multi-layer-number
- Config_by_code tendria que estar siempre en false

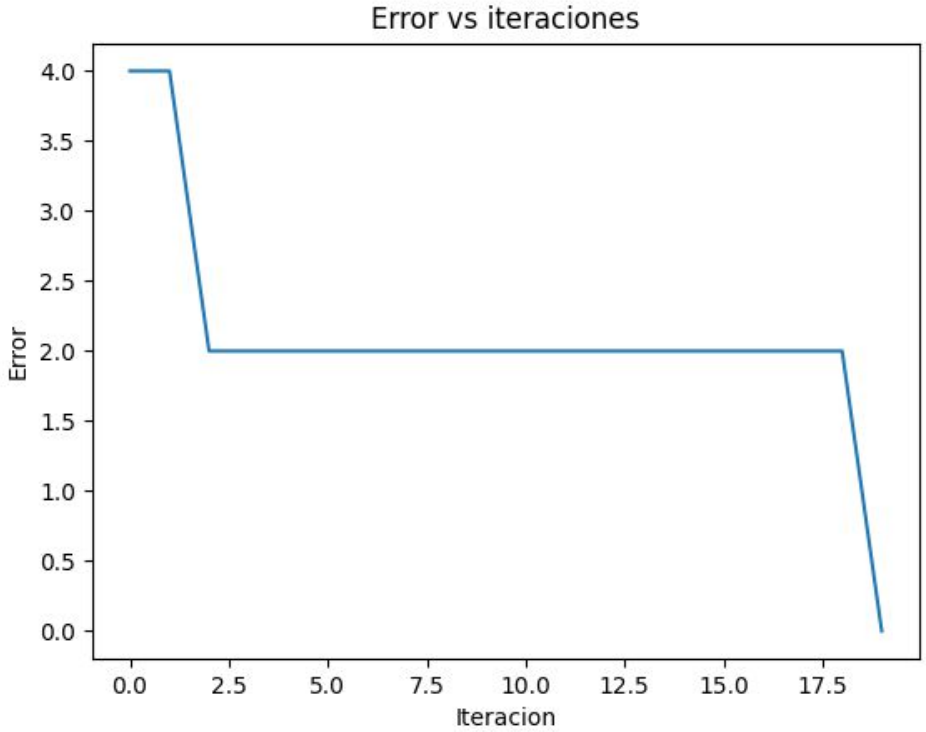
Ejercicio 1

Perceptrón Simple con función de activación escalón

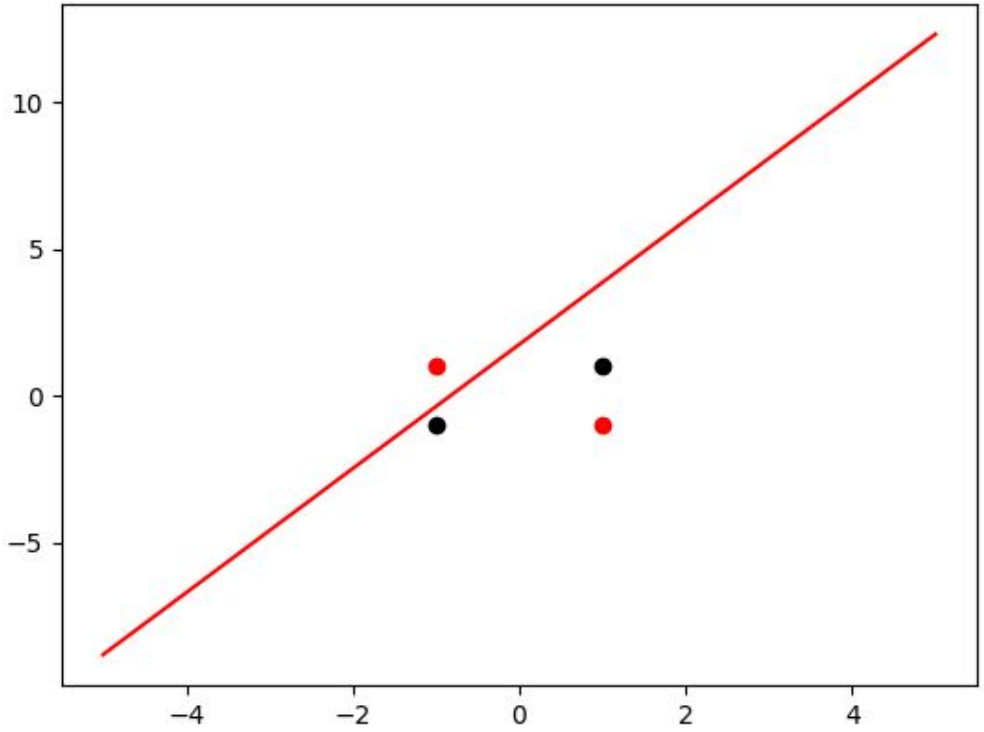
AND - Problema linealmente separable



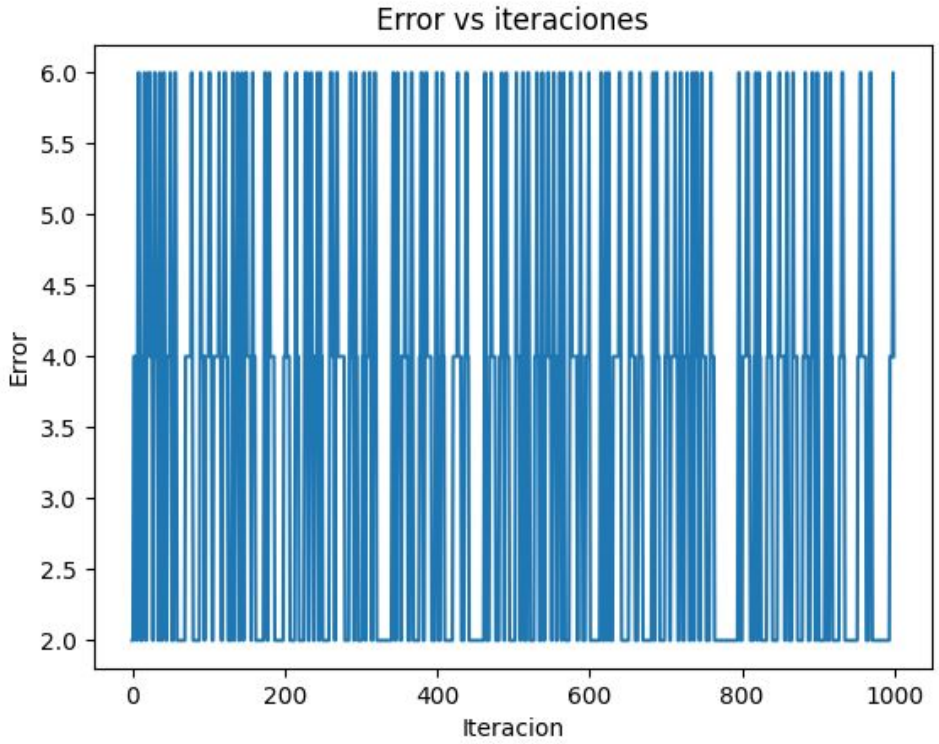
AND - Problema linealmente separable



XOR - Problema no linealmente separable



XOR - Problema no linealmente separable



No aprende

Resultados

AND

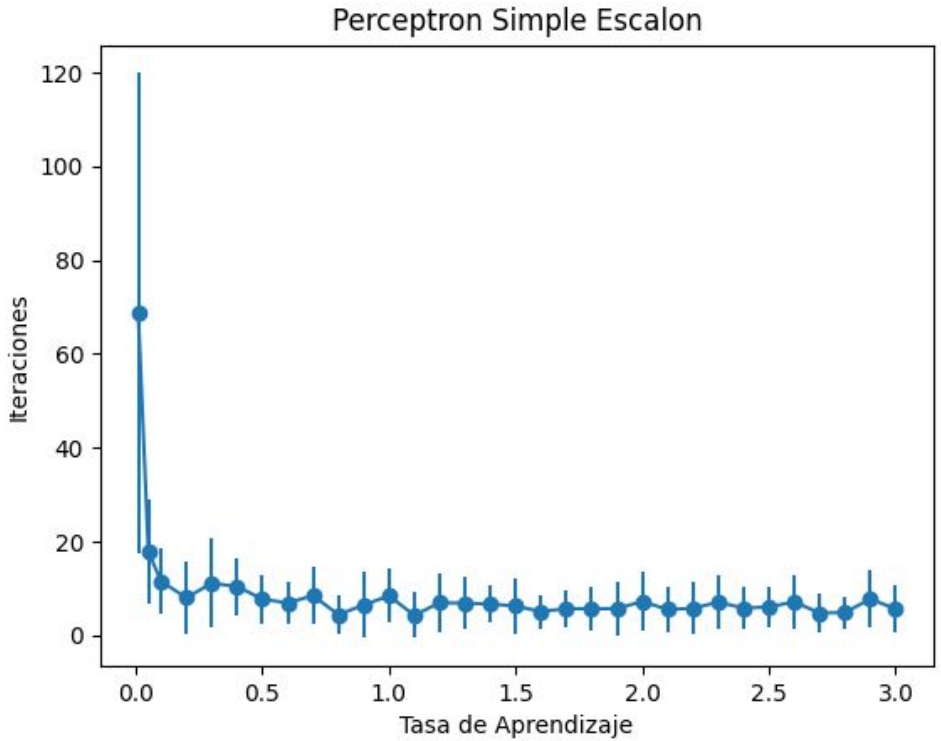
```
Iteraciones: 6
min training error: 0.0
evaluation error: 0.0
```

XOR

```
Iteraciones: 1000
min training error: 2.0
evaluation error: 2.0
```

Max-iteraciones = 1000, n=0.1

Tasa de Aprendizaje - **AND**

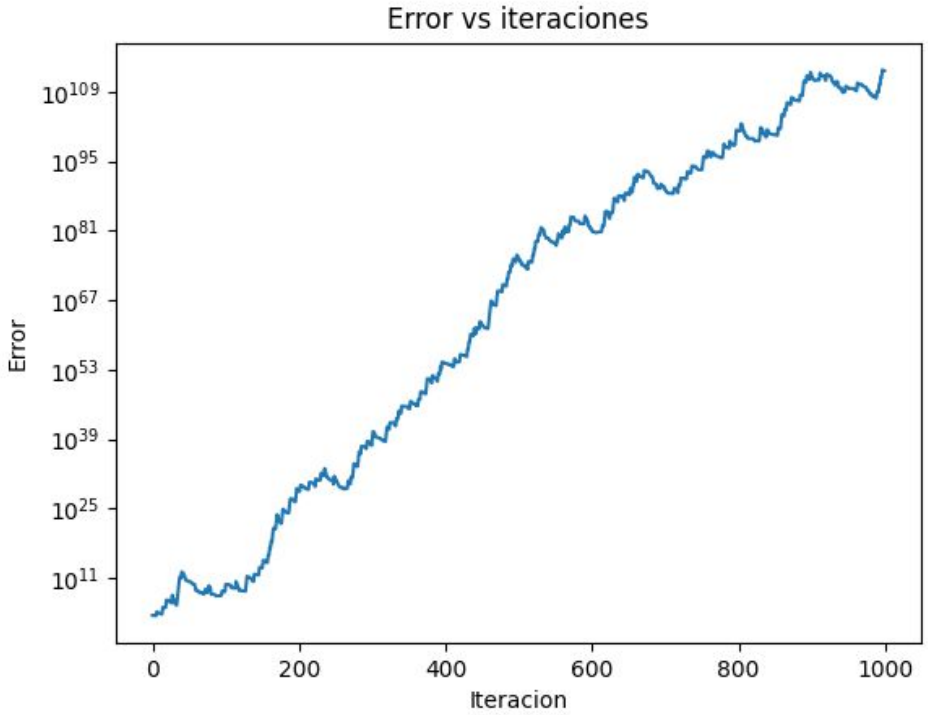


Promedio de 20 ejecuciones

Ejercicio 2

Perceptrón Simple Lineal y No Lineal

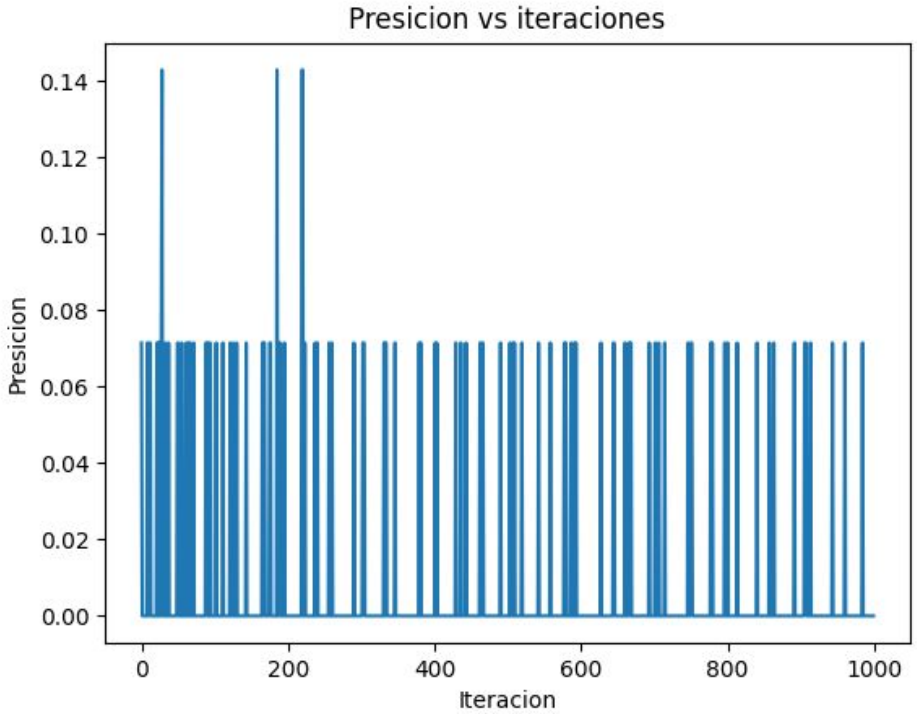
Perceptrón simple lineal



n = 0.1
Max iteraciones = 1000
Función de activación = Identidad
Normalización = Scale

No aprende

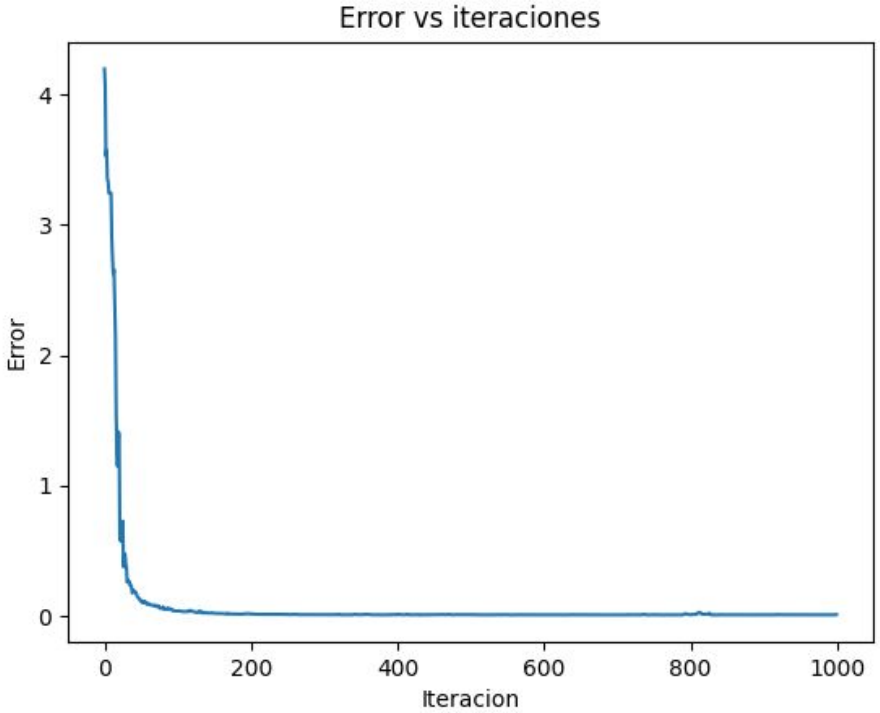
Perceptrón simple lineal



n = 0.1
Max iteraciones = 1000
Función de activación = Identidad
Normalización = Scale

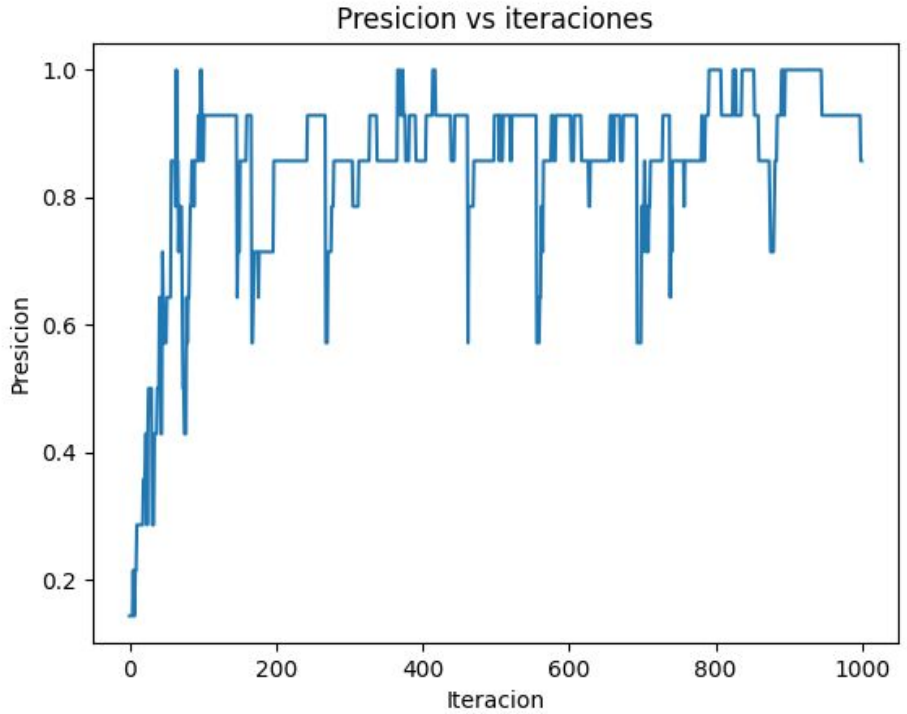
No aprende

Perceptrón simple no lineal (tan)



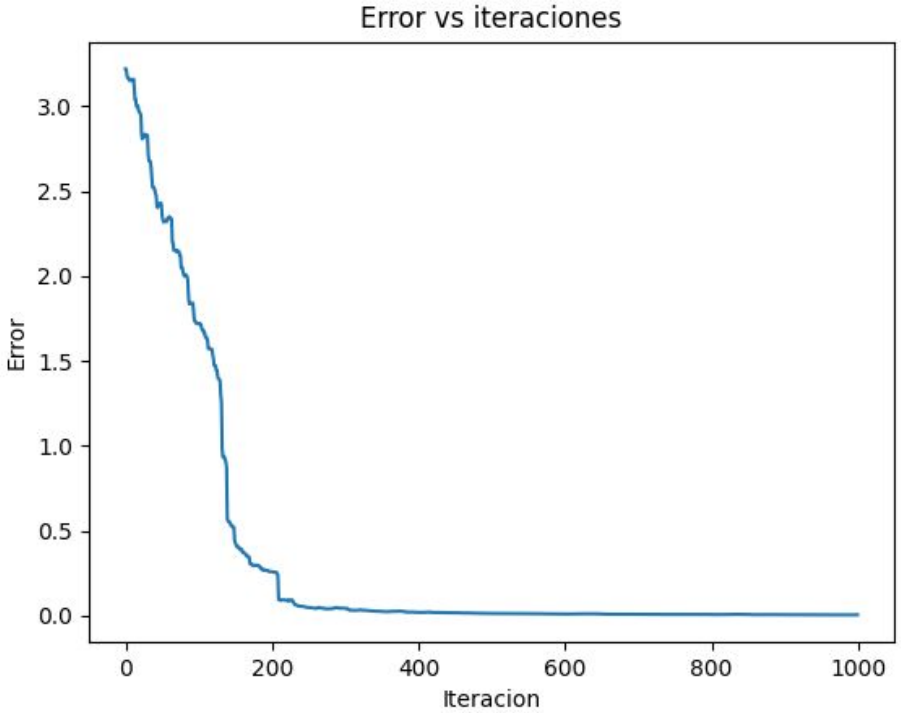
n = 0.1
Max iteraciones = 1000
Función de activación = Tanh
Normalización = Scale

Perceptrón simple no lineal (tan)



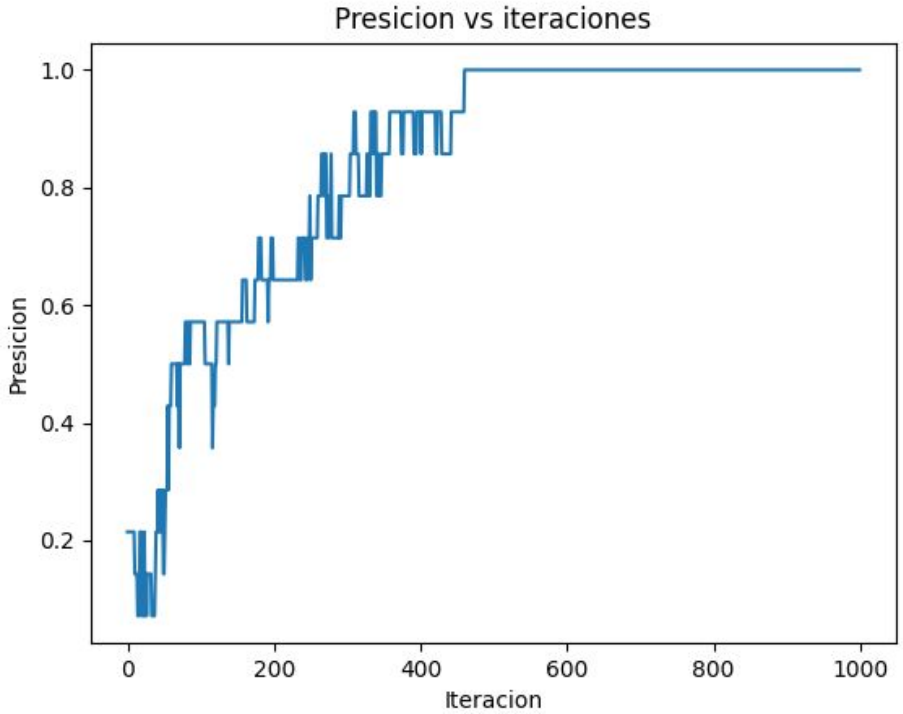
n = 0.1
Max iteraciones = 1000
Función de activación = Tanh
Normalización = Scale

Perceptrón simple no lineal (logistic)



n = 0.1
Max iteraciones = 1000
Función de activación = Logistic
Normalización = Scale

Perceptrón simple no lineal (logistic)



n = 0.1
Max iteraciones = 1000
Función de activación = Logistic
Normalización = Scale

Resultados

Linear

```
Iteraciones: 1000
min training error: 10435.779777997646
accuracy evaluation set: 0.0
accuracy train: 0.0
evaluation error: 11813.870711371212
```

Non Linear Logistic

```
Iteraciones: 1000
min training error: 0.004304929364354537
accuracy evaluation set: 1.0
accuracy train: 1.0
evaluation error: 0.005285477222773443
```

Non Linear Tanh

```
Iteraciones: 1000
min training error: 0.003071485504381398
accuracy evaluation set: 0.9285714285714286
accuracy train: 1.0
evaluation error: 0.011380567915570895
```

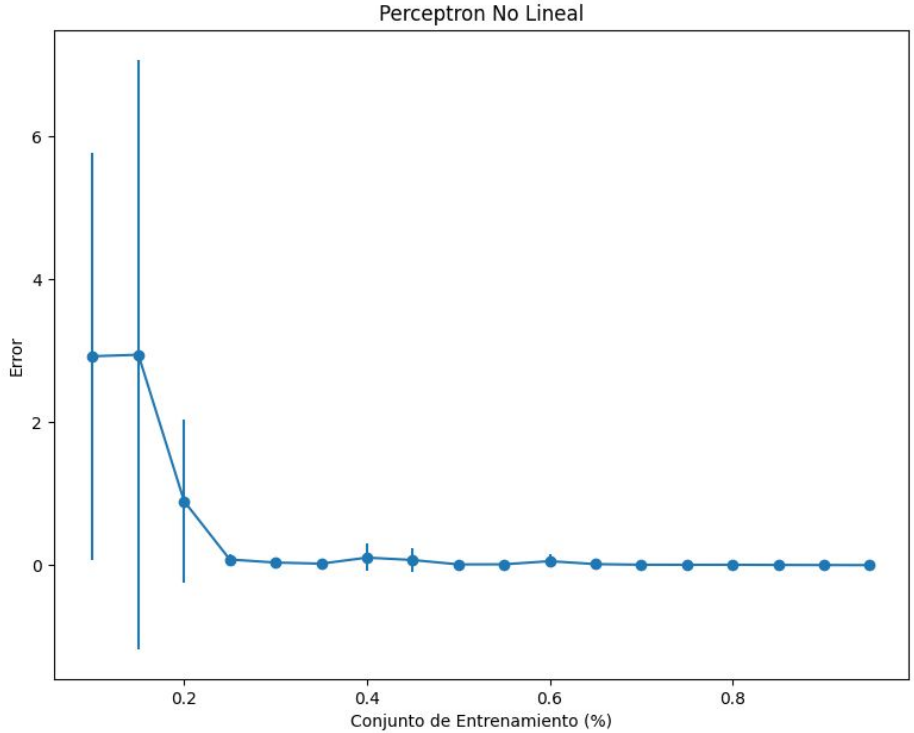
Max-iteraciones 1000
n=0.1
b =1
Elementos conjunto de entrenamiento: 50%

Elección conjunto de entrenamiento

2 formas

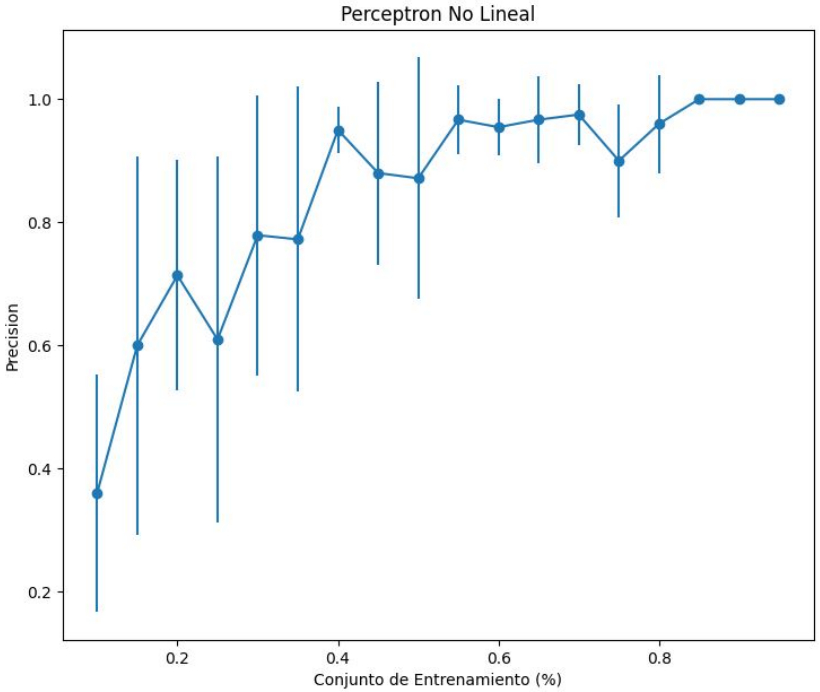
- Se toma la lista de puntos, se le hace un random shuffle y se agarra un porcentaje de los puntos para el entrenamiento y el resto para la evaluación
- Validación cruzada: se toma la lista de puntos, se le hace un random shuffle y se forman k conjuntos. Luego se hacen k entrenamientos, en donde se varía el conjunto de evaluación (tomando k-1 conjuntos de entrenamiento y 1 de evaluación). Se toman los w del conjunto que minimiza el error del conjunto de evaluación.

Elección conjunto de entrenamiento - Solución Propia



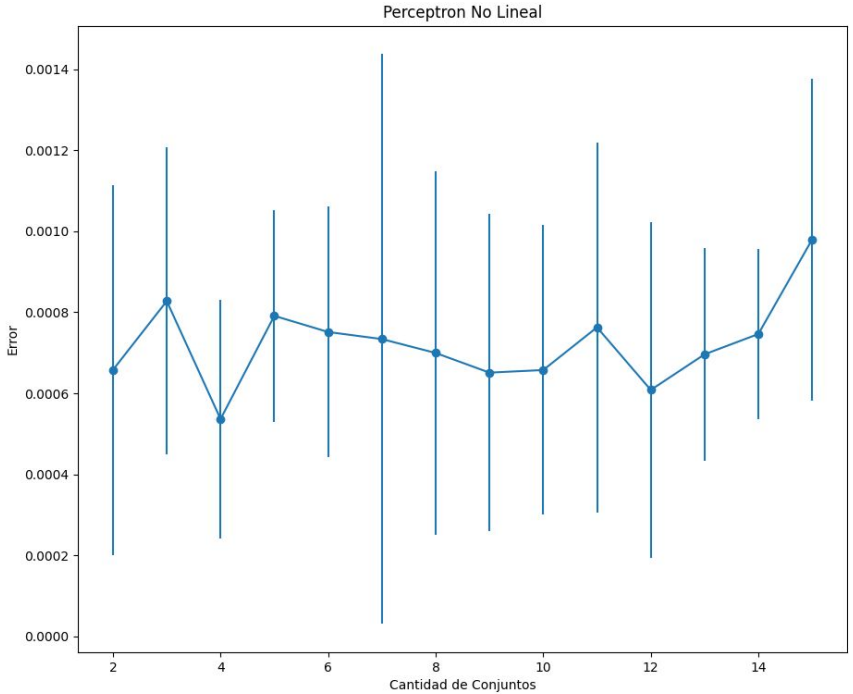
Promedio de 10 ejecuciones
n = 0.1
Max iteraciones = 1000
Función de activación = tanh
Normalización = Scale

Elección conjunto de entrenamiento - Solución Propia



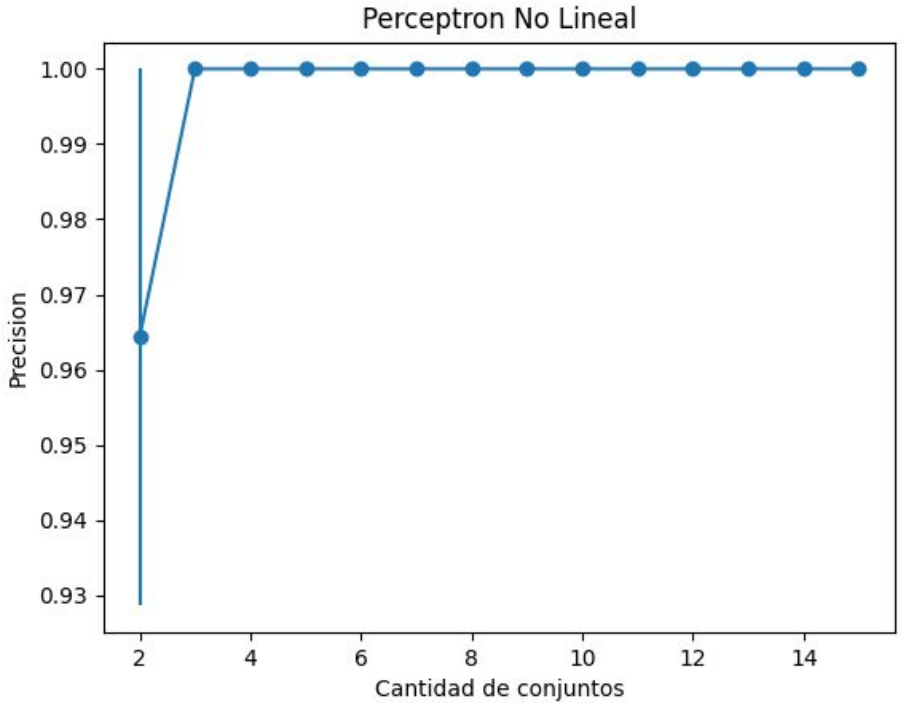
Promedio de 10 ejecuciones
n = 0.1
Max iteraciones = 1000
Función de activación = tanh
Normalización = Scale

Elección conjunto de entrenamiento - Validación Cruzada



Promedio de 10 ejecuciones
n = 0.1
Max iteraciones = 1000
Función de activación = tanh
Normalización = Scale

Elección conjunto de entrenamiento - Validación Cruzada



Promedio de 10 ejecuciones
n = 0.1
Max iteraciones = 1000
Función de activación = tanh
Normalización = Scale

Elección conjunto de entrenamiento

Para perceptrón no lineal con función de activación tanh, promedio de 10 ejecuciones, con 1000 iteraciones máximas y 0.1 de tasa de aprendizaje.

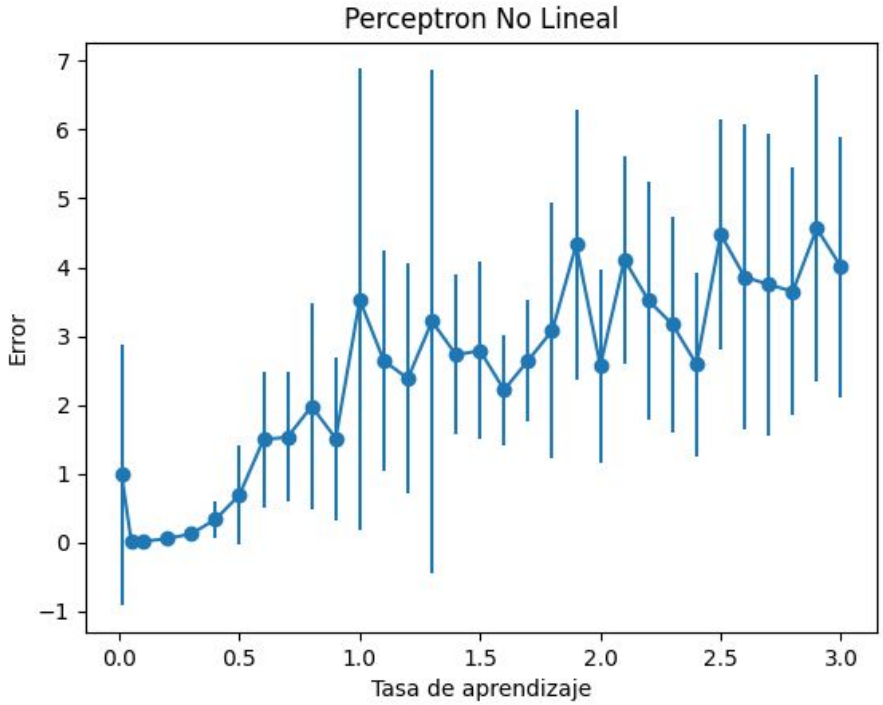
```
Media de error: 0.11620426075209385
Desviacion estandar de error: 0.3093404395507768
Media de accuracy entrenamiento: 0.9214285714285715
Desviacion estandar entrenamiento: 0.09819805060619656
Media de accuracy evaluacion: 0.8714285714285716
Desviacion estandar evaluacion: 0.22990681342044403
Media de cantidad iteraciones: 1000.0
Desviacion estandar de cantidad iteraciones: 0.0
```

Solución Propia. 50% para entrenamiento y 50% para evaluar

```
Media de error: 0.0005602639886921446
Desviacion estandar de error: 0.0004062851703617723
Media de accuracy entrenamiento: 0.9640000000000001
Desviacion estandar entrenamiento: 0.01200000000000001
Media de accuracy evaluacion: 1.0
Desviacion estandar evaluacion: 0.0
Media de cantidad iteraciones: 1000.0
Desviacion estandar de cantidad iteraciones: 0.0
```

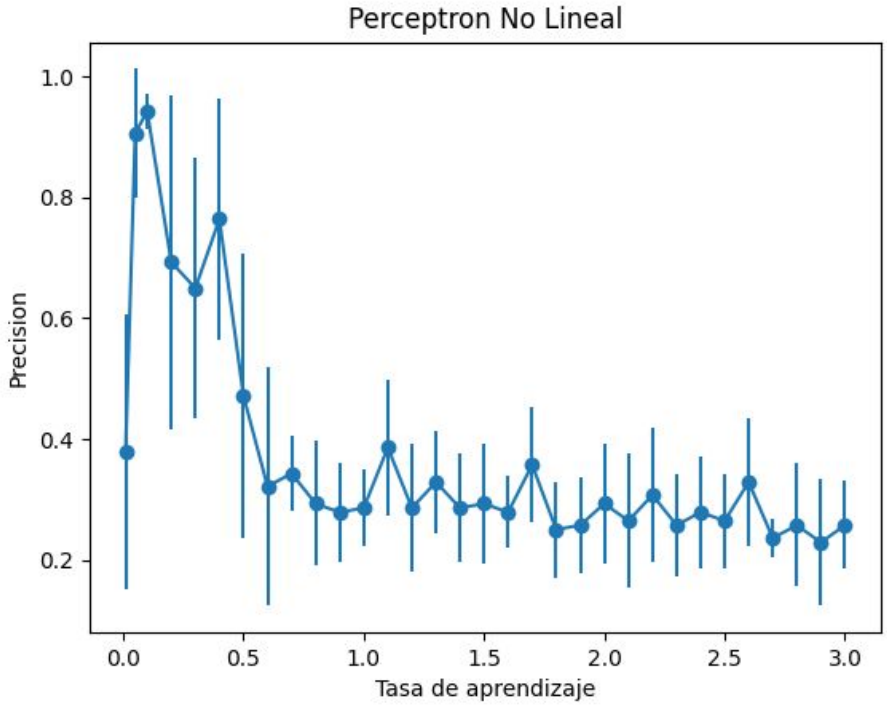
Validación Cruzada. 5 conjuntos

Evaluación variando tasa de aprendizaje



Promedio de 10 ejecuciones
n = 0.1
Max iteraciones = 1000
Función de activación = tanh
Normalización = Scale

Evaluación variando tasa de aprendizaje



Promedio de 10 ejecuciones
n = 0.1
Max iteraciones = 1000
Función de activación = tanh
Normalización = Scale

Ejercicio 3

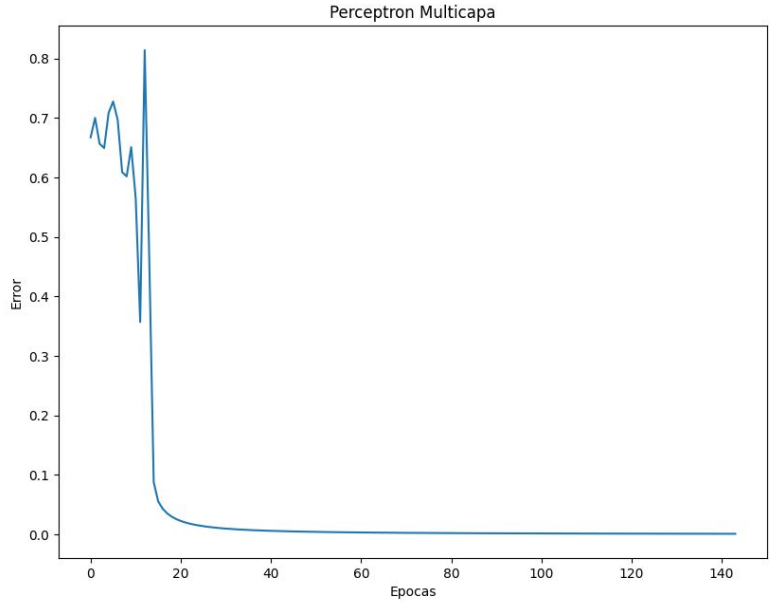
Perceptrón Multicapa

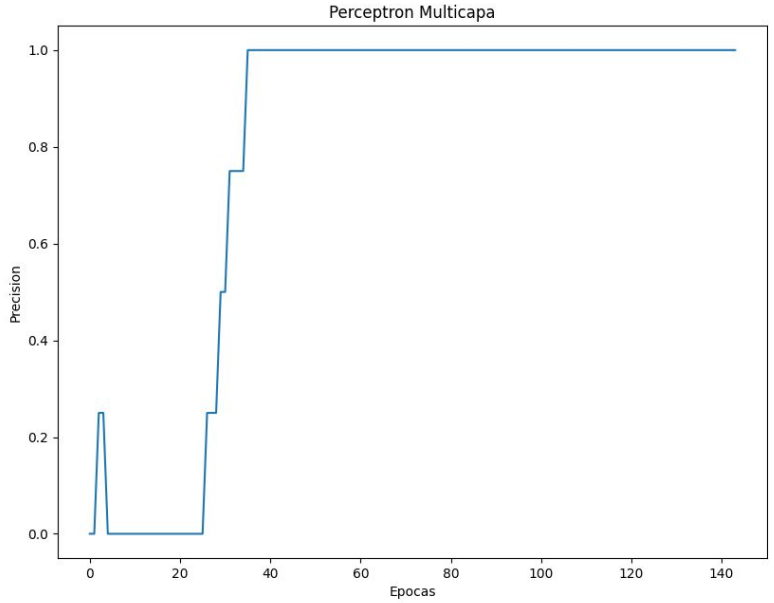
Perceptrón Multicapa (XOR)

Parámetros de los gráficos:

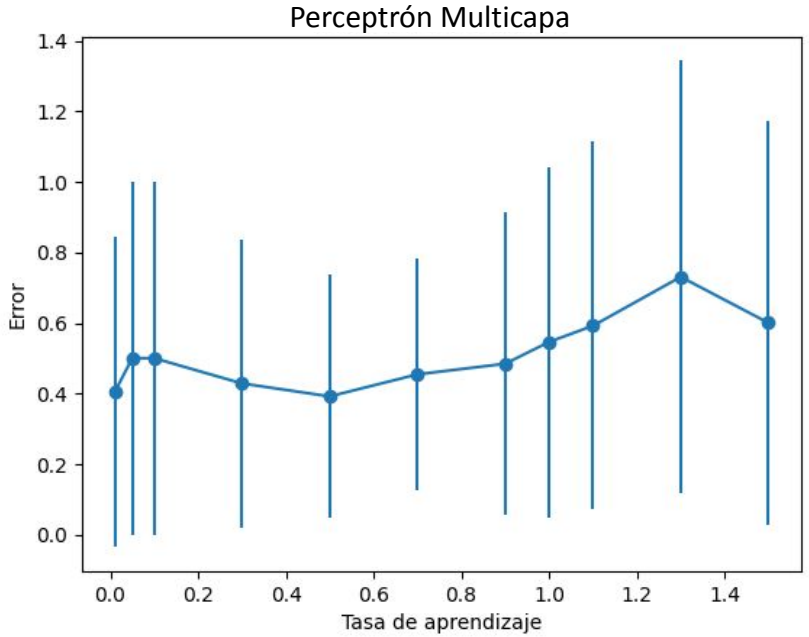
- Promedio de 10 ejecuciones
- n = 0.1
- Max iteraciones = 300
- Función de activación = tanh
- Normalización = Scale
- Error de corte = 1E-3
- Nodos internos = 6
- Capas = 6

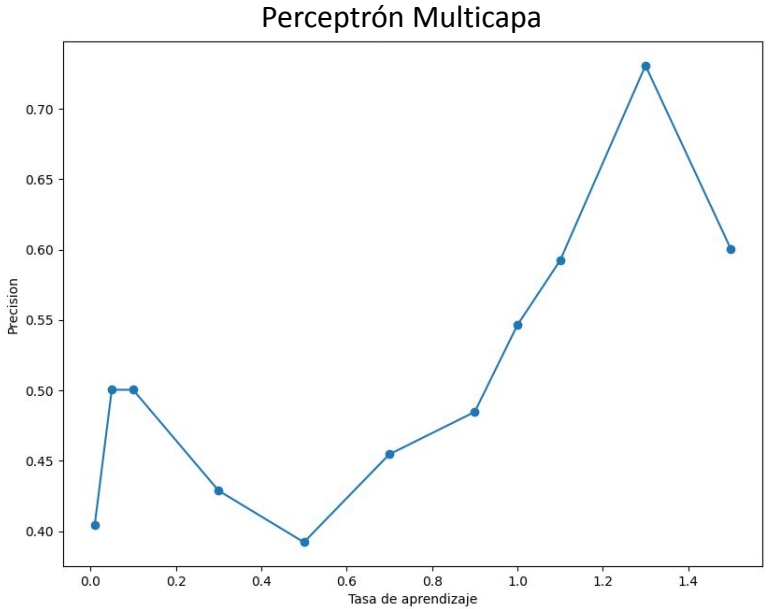
Perceptrón Multicapa (XOR) - Error/Precisión vs Épocas



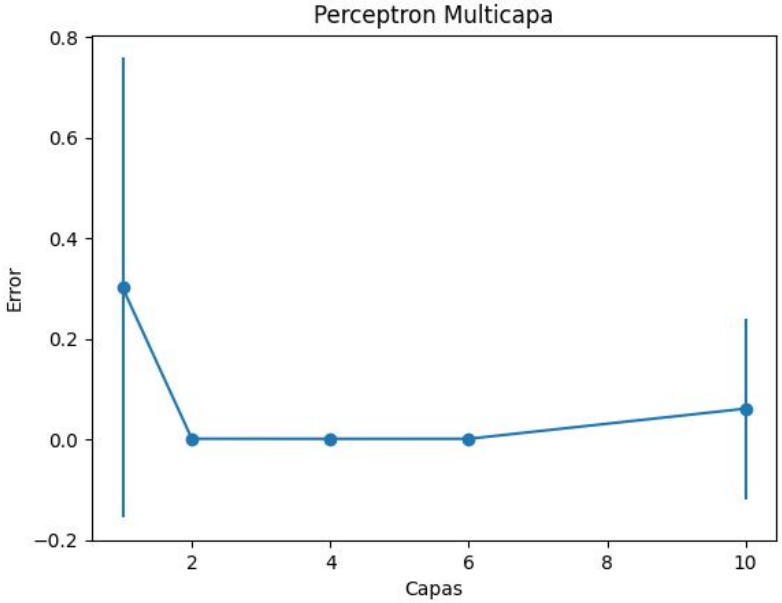


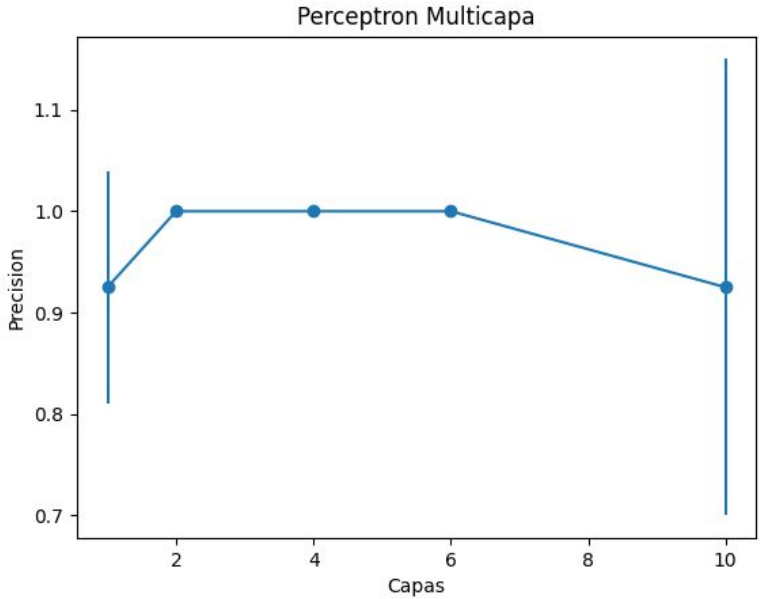
Perceptrón Multicapa (XOR) - Error/Precisión vs Tasa de aprendizaje



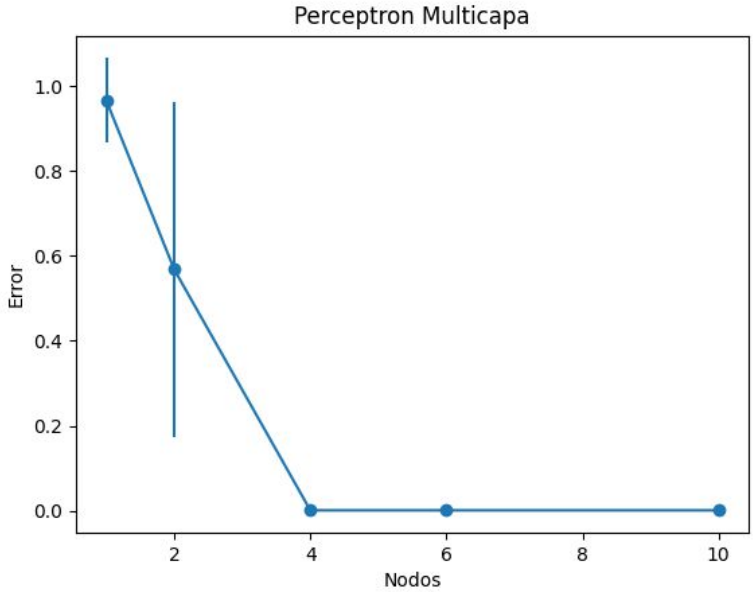


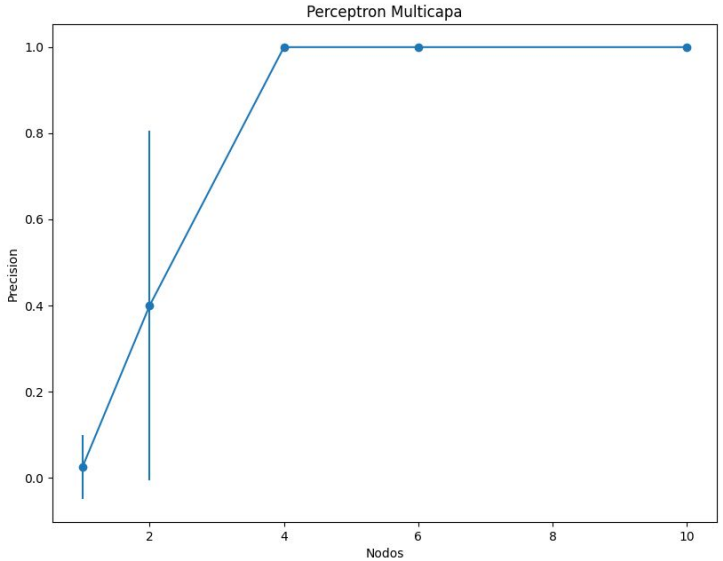
Perceptrón Multicapa (XOR) - Error/Precisión vs Capas



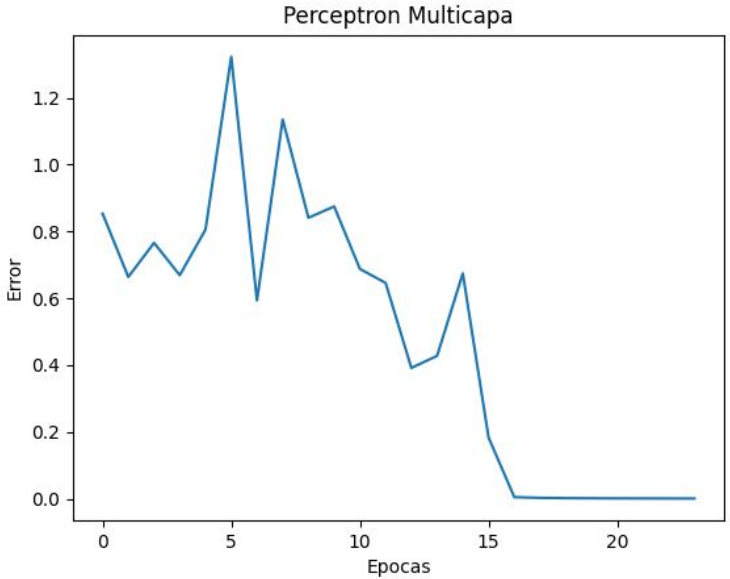


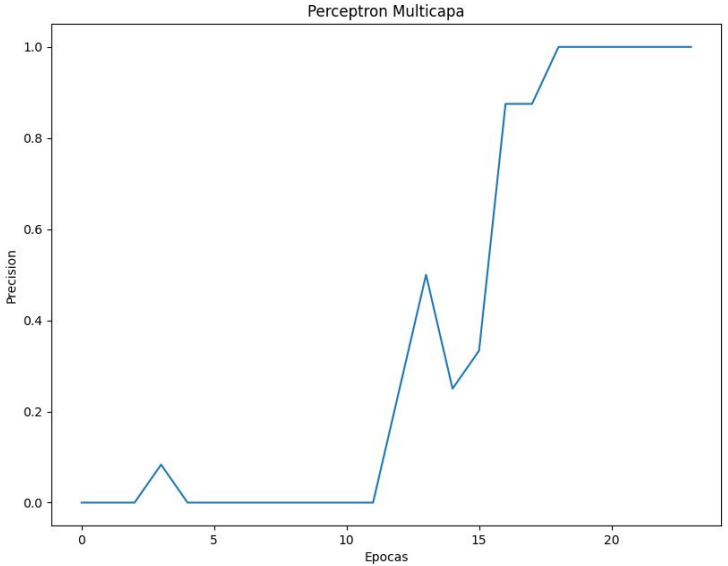
Perceptrón Multicapa (XOR) - Error/Precisión vs Nodos





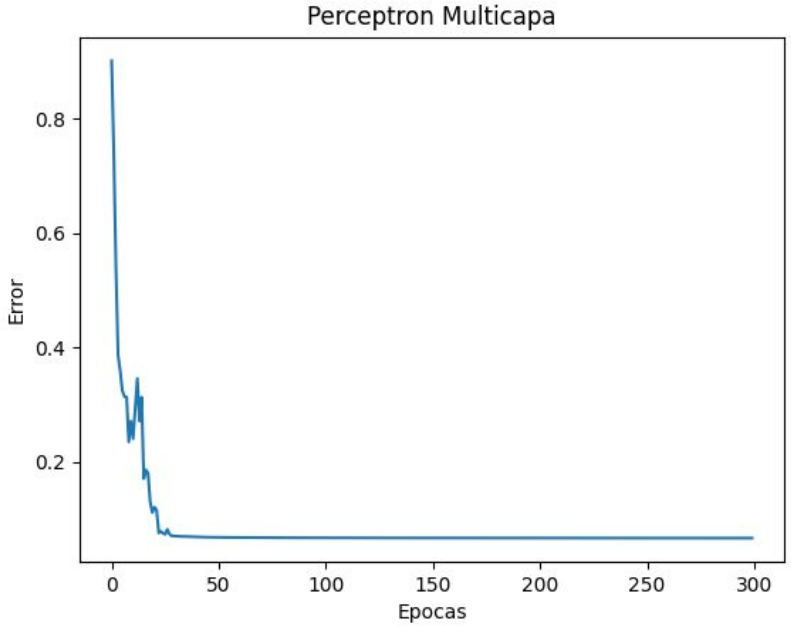
Perceptrón Multicapa (Par) - Error/Precisión vs Épocas

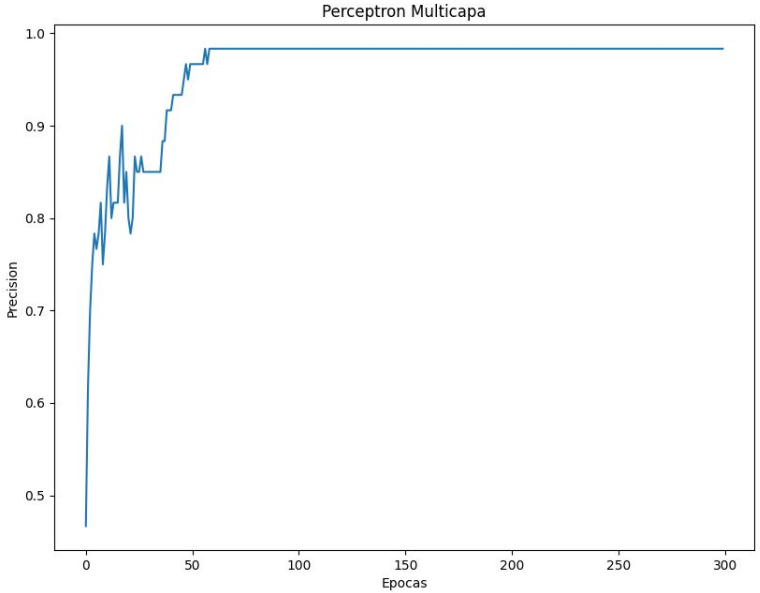




Perceptrón Multicapa (Number) - Error/Precisión vs Épocas

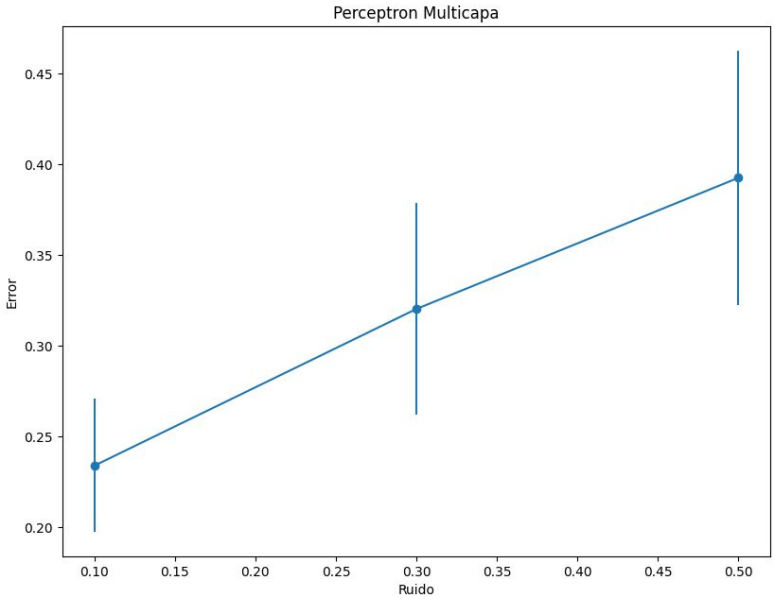
Cantidad de capas = 2
Cantidad de nodos = 20
Cantidad de nodos de salida = 10

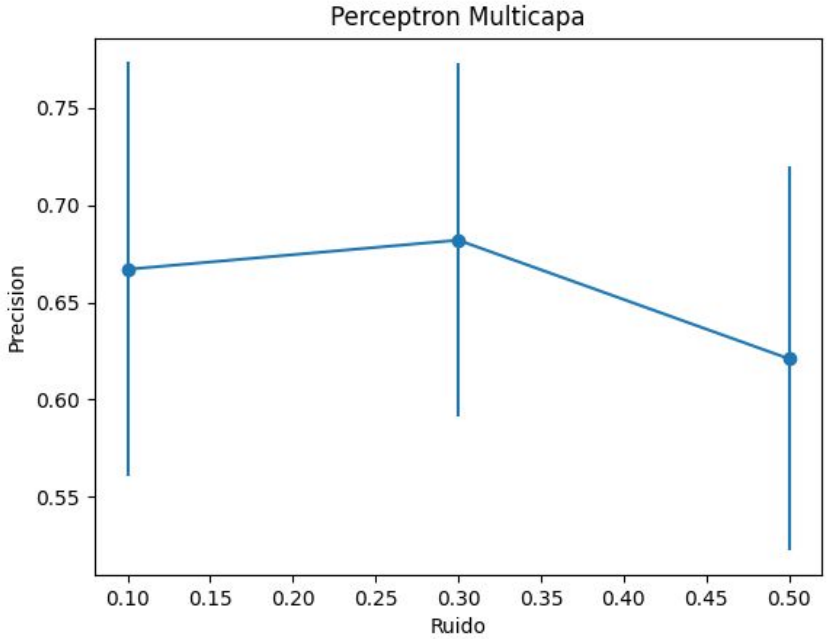




Perceptrón Multicapa (Number) - Error/Precisión vs Ruido

Cantidad de capas = 6
Cantidad de nodos = 6
Cantidad de nodos de salida = 10





Resultados

Multilayer xor
Capas Ocultas = 6 Nodos por capa = 6 Nodos
capa de salida = 1

```
epocas: 228
min training error: 0.0009981999298435986
accuracy evaluation set: 1.0
accuracy train: 1.0
evaluation error: 0.0009981999298435986
```

Multilayer even
Capas Ocultas = 6 Nodos por capa = 6 Nodos
capa de salida = 2

```
accuracy evaluation set: 1.0
accuracy train: 1.0
evaluation error: 0.00032703068810775927
```

Multilayer number
Capas Ocultas = 2 Nodos por capa = 35
Nodos capa de salida = 10

```
accuracy evaluation set: 0.9
accuracy train: 0.93
evaluation error: 0.40000442058490976

accuracy noise evaluation set: 0.93
noise error: 0.28008244414847205
```

Max-iteraciones 300
n=0.1
b=1
Cross validation, k=5 (no para xor)

Resultados

```
index: 0
epocas: 300
min training error: 0.20000724007382598

index: 1
epocas: 300
min training error: 0.12504378174382197

index: 2
epocas: 300
min training error: 0.283339091582605

index: 3
epocas: 300
min training error: 0.27499869135313093

index: 4
epocas: 300
min training error: 0.27999777855380353

accuracy evaluation set: 0.9
accuracy train: 0.93
evaluation error: 0.40000442058490976

accuracy noise evaluation set: 0.93
noise error: 0.28008244414847205
```

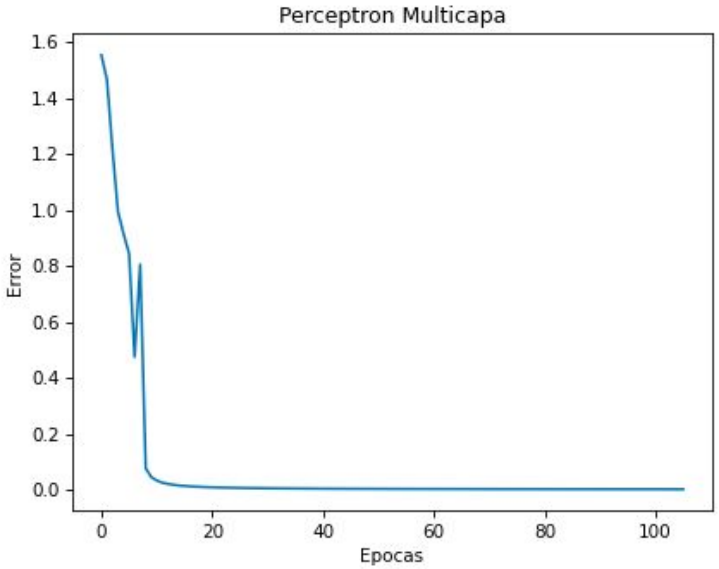
Non Linear Number
Nodos capa de salida = 10

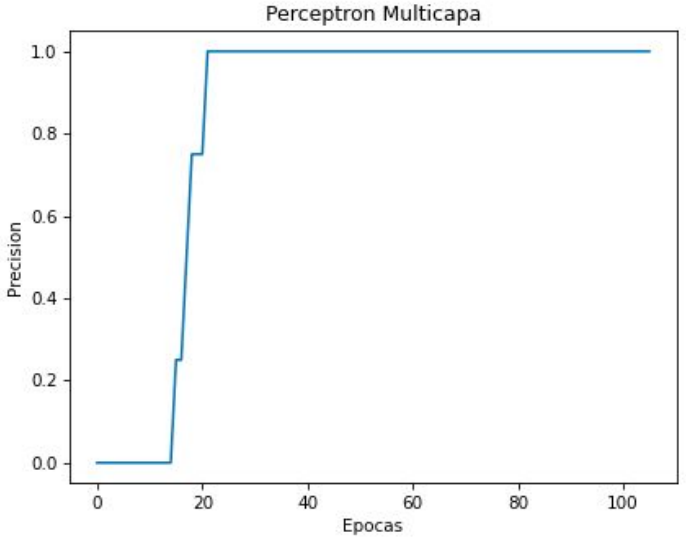
Max-iteraciones 300
n=0.1
b=1
Cross validation, k=5 (no para xor)
Capas ocultas = 6
Nodos capas ocultas = 6

Optimizaciones

- Momentum
- N adaptativo
- Adam

Optimizaciones

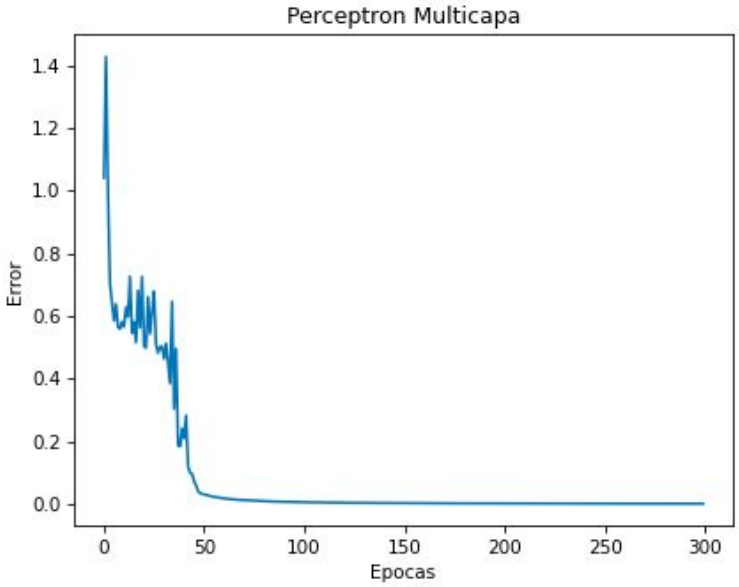


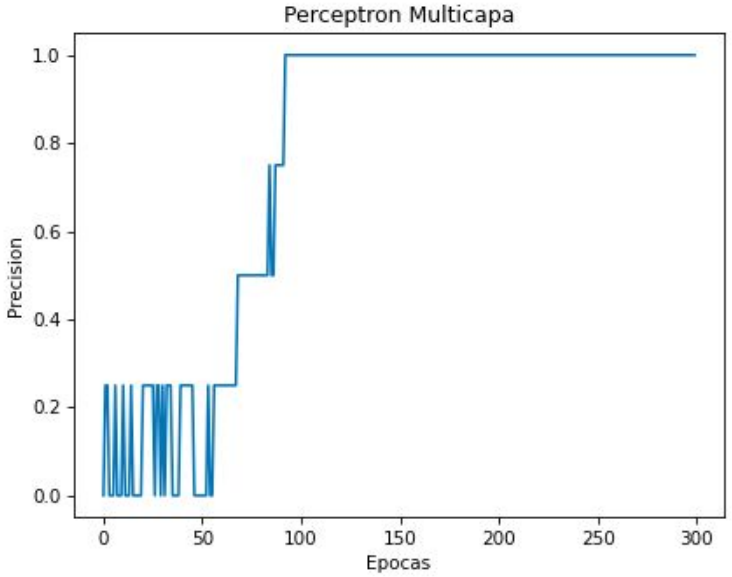


Max epocas = 300, n = 0.1, Capas ocultas = 6,Nodos por capa = 6, Nodos de salida = 1, Conjunto de entrenamiento: 0.6

Sin Optimizar

Optimizaciones

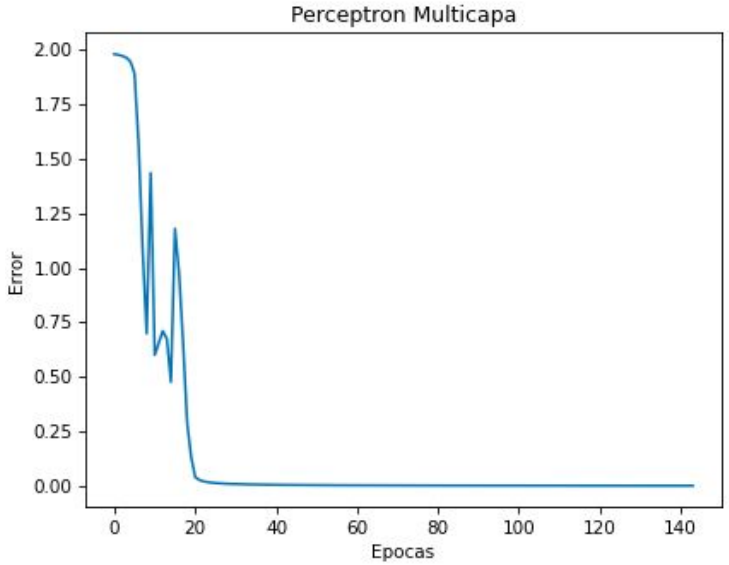


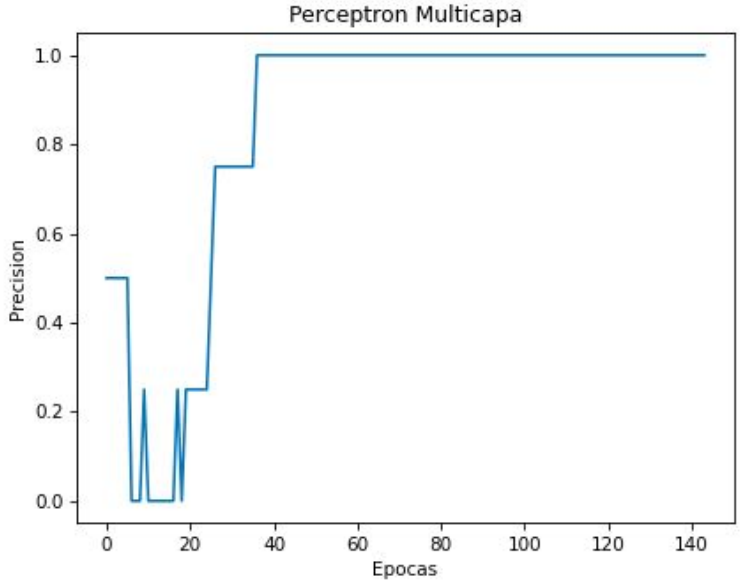


Max epocas = 300, n = 0.1, Capas ocultas = 6,Nodos por capa = 6, Nodos de salida = 1, Conjunto de entrenamiento: 0.6

Momentum

Optimizaciones

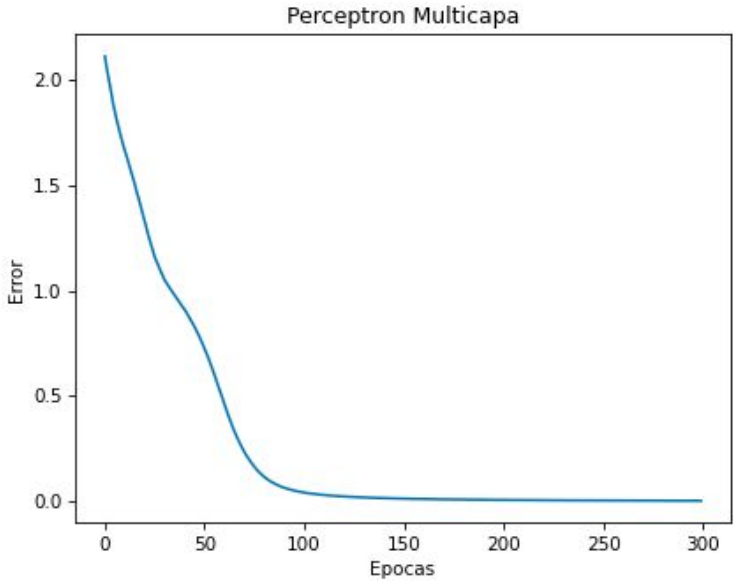


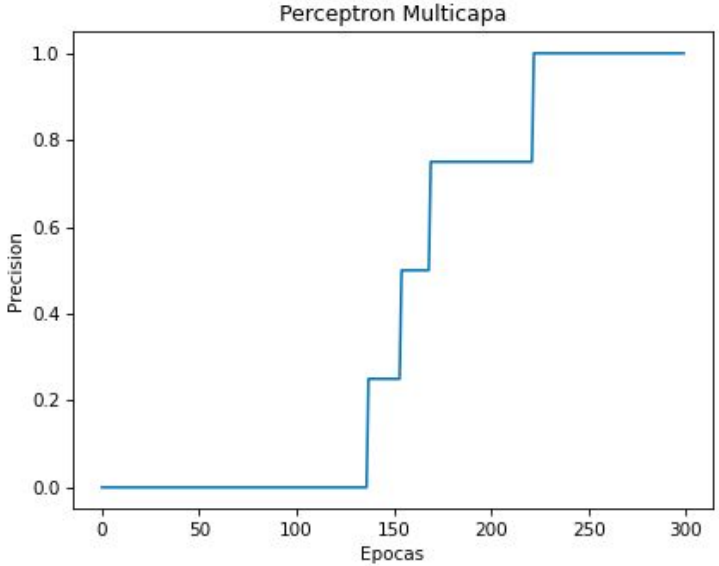


Max epocas = 300, n = 0.1, Capas ocultas = 6,Nodos por capa = 6, Nodos de salida = 1, Conjunto de entrenamiento: 0.6, Max-progress= 2, Incrementa en 0.01, Decrecemento en 0.03

N adaptativo

Optimizaciones





Max epocas = 300, n = 0.1, Capas ocultas = 6,Nodos por capa = 6, Nodos de salida = 1, Conjunto de entrenamiento: 0.6

Adam

Conclusiones

Conclusiones

- El perceptrón simple con función de activación escalón solo sirve para un problema que es linealmente separable.
- El perceptrón simple lineal sirve para problemas linealmente compatibles.
- El perceptrón simple no lineal sirve para resolver problemas que no son lineales.
- Utilizar cross validation para elegir el conjunto de entrenamiento da mejores resultados.
- Los perceptrones multicapa pueden resolver problemas no linealmente separables.