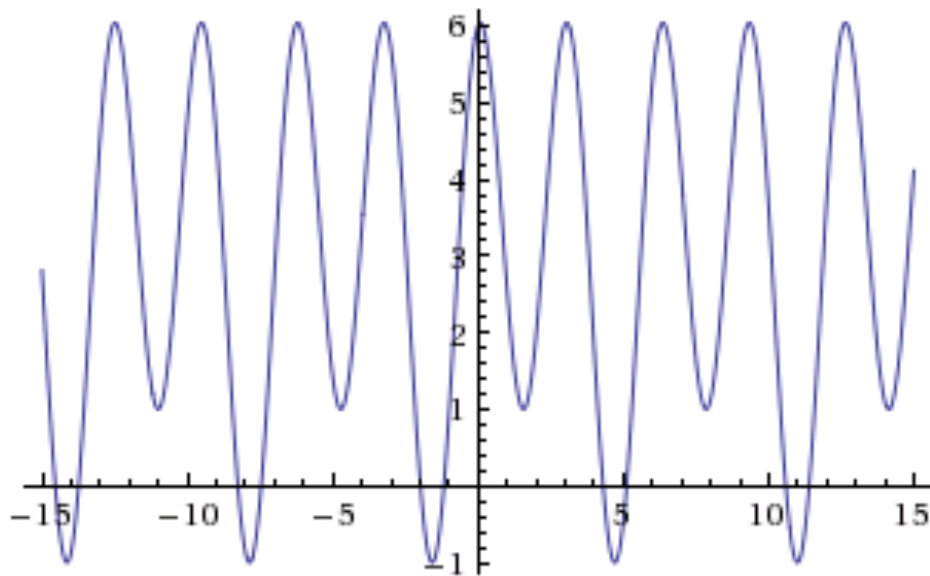


Redes Neuronales

Sistemas de Inteligencia Artificial
Grupo 7

Función propuesta

$$f(x) = \sin(x) + 6 (\cos(x))^2 \text{ con } x \in [-15, 15]$$

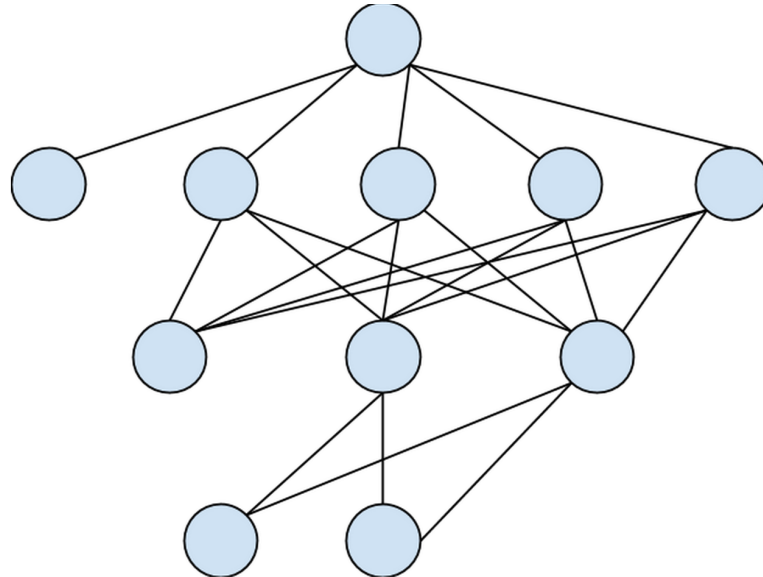


Objetivo

- Construir el perceptrón
- Enseñarle a partir de patrones de entrenamiento
- Testear el buen funcionamiento del aprendizaje
- ***Estimar la función propuesta por la cátedra***

Notación

- $[1\ 2\ 4\ 1]$ = representa la arquitectura que tiene dos capas ocultas de 2 y 4 neuronas respectivamente, más el umbral



Funciones de activación

a. La tangente hiperbólica (g1):

$$g(h) = \tanh(\beta h) \text{ para unidades entre } -1 \text{ y } 1.$$

b. La función exponencial (g2):

$$g(h) = 1/(1 + \exp(-2\beta h)) \text{ para unidades entre } 0 \text{ y } 1.$$

Implementación

Función: MultiLayerPerceptron

Parámetros: Values, layerSizes, eta, gValue, beta, error, momentum, etaAdaptativo, a, b.

Conjuntos de entrenamiento

Se entrenó al perceptrón con:

Puntos de la función con una separación de 0.1

Conjunto con más densidad en los extremos

Se utilizaron los patrones en orden aleatorio

Error

Error cuadrático medio

$$E(W) = \frac{1}{2} \sum_{i\mu} (S_i^\mu - o_i^\mu)^2$$

Mejoras

Momentum

- Se eligieron los valores 0.5 y 0.9 para los valores de alfa, utilizando la siguiente fórmula:

$$\Delta w_{pq}(t + 1) = -\eta \frac{\partial E}{\partial w_{pq}} + \alpha \Delta w_{pq}(t)$$

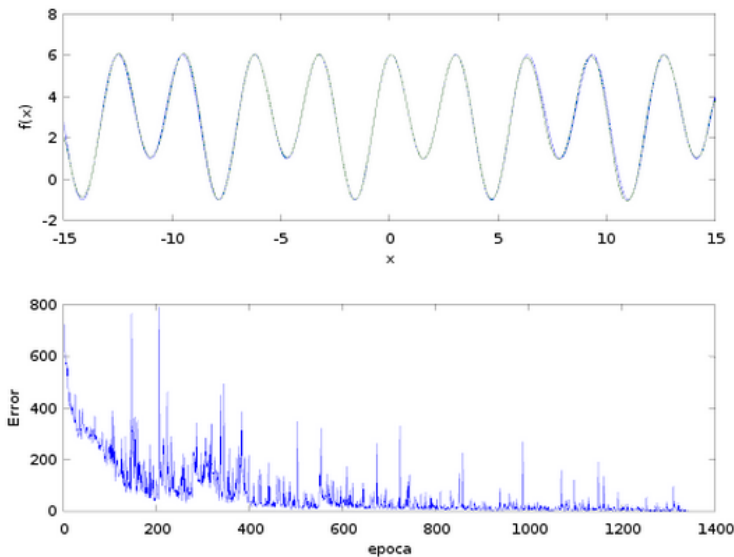
Eta adaptativo

- Eta adaptativo: busca corregir el factor de aprendizaje.

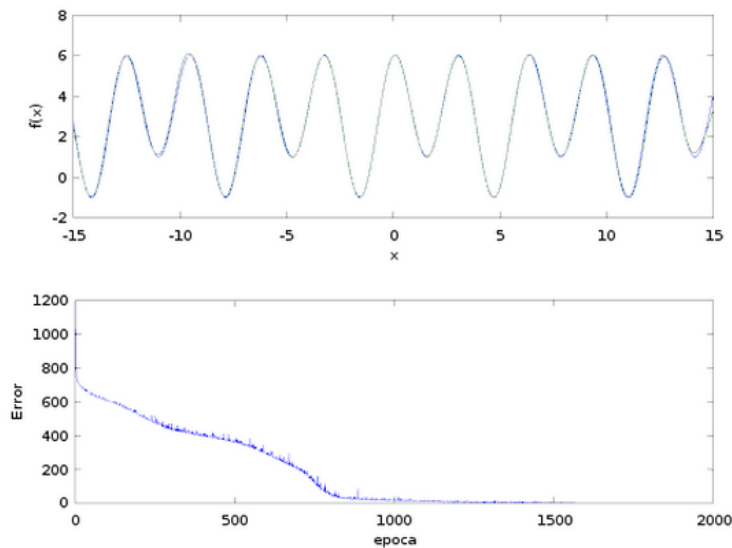
$$\Delta\eta = \begin{cases} +a & \text{si } \Delta E < 0 \text{ consistentemente} \\ -b\eta & \text{si } \Delta E > 0 \\ 0 & \text{en otro caso} \end{cases}$$

Resultados

Comparación de betas



$$\beta = 1.0$$



$$\beta = 0.5$$

Tabla comparativa: 1 capa oculta

#Capas Ocultas	Capas			Momentum	Adaptativo	a	b	Error	Épocas-g1	Épocas-g2
1	[1 5 1]	0.01	0.5	0	0	0	0	1	(+6500)	(+6500)
	[1 10 1]	0.01	0.5	0	0	0	0	1	1692	944
	[1 20 1]	0.01	0.5	0	0	0	0	1	2496	3249
	[1 100 1]	0.001	1	0.5	0	0	0	1	5241	5541

Tabla comparativa: 2 capas ocultas

2	[1 10 5 1]	0.01	0.5	0	0	0	0	1	719	3630
	[1 10 5 1]	0.01	0.5	0.5	0	0	0	1	347	2729
	[1 10 6 1]	0.01	0.5	0.5	0	0	0	1	283	565
	[1 10 8 1]	0.01	0.5	0.5	0	0	0	1	446	534
	[1 10 9 1]	0.01	0.5	0.5	0	0	0	1	459	1077
	[1 10 10 1]	0.01	0.5	0.5	0	0	0	1	467	1426
	[1 10 6 1]	0.01	0.5	0.5	0	0	0	1	298	1587
2	[1 15 5 1]	0.01	0.5	0.5	0	0	0	1	243	1124
	[1 15 6 1]	0.01	0.5	0.5	0	0	0	1	497	911
	[1 15 8 1]	0.01	0.5	0.5	0	0	0	1	423	480
	[1 15 10 1]	0.01	0.5	0.5	0	0	0	1	376	642
	[1 15 11 1]	0.01	0.5	0.5	0	0	0	1	299	858
	[1 15 12 1]	0.01	0.5	0.5	0	0	0	1	338	406
	[1 15 13 1]	0.01	0.5	0.5	0	0	0	1	297	714
	[1 15 14 1]	0.01	0.5	0.5	0	0	0	1	382	573

Tablas comparativas: múltiples capas ocultas

Capas	η	β	Momentum	η Adaptativo	a	b	Error	Epocas- g1	Tiempo(s)
[1 512 256 128 64 32 16 8 1]	0,0001	0.75	0.5	0	0	0	1	843	596,84
	0,0001	0.5	0.5	0	0	0	1	1180	1.612,75
	0,0001	1	0.5	0	0	0	1	1000	2.000,00
	0,0001	<u>0,75</u>	0.9	0	0	0	1	2785	5.369,93
	0,0002	<u>0,75</u>	0,5	0	0	0	1	1348	2.381,78
	0,00005	<u>0,75</u>	0.5	0	0	0	1	1110	1.675,20
[1 256 128 128 64 32 16 1]	0,0001	0.75	0.5	0	0	0	1	1248	454,18
	0,0001	0.5	0.5	0	0	0	1	3041	1.798,37
	0,0001	1	0.5	0	0	0	1	1687	1.649,02
	0,0001	<u>0,75</u>	0.9	0	0	0	1	3655	3.218,92
	0,0002	0.75	0.5	0	0	0	1	1569	1.562,52
	0,00005	0.75	0.5	0	0	0	1	1974	1.604,96
[1 512 256 128 64 32 16 1]	0,0001	<u>0,75</u>	0,5	0	0	0	1	810	914,504
	0,0001	0,5	0,5	0	0	0	1	1568	2982,73
	0,0001	1	0,5	0	0	0	1	1344	2955,16
	0,0001	<u>0,75</u>	0,9	0	0	0	1	1753	3767,68
	0,0002	<u>0,75</u>	0,5	0	0	0	1	958	1914,35
	0,00005	<u>0,75</u>	0,5	0	0	0	1	952	1290,56

Arquitectura: múltiples capas ocultas

Distribución de patrones de aprendizaje:

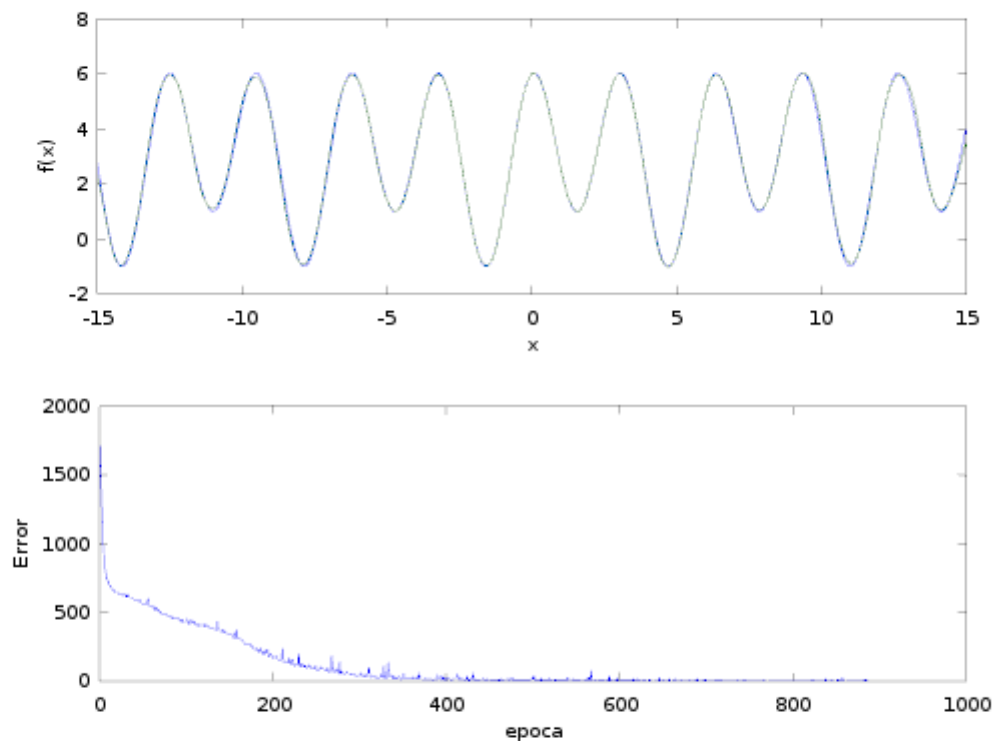
[[-15 : 0.05 : -12.5] [-12.5 : 0.25 : -5] [-5 : 0.5 : 5] [5 : 0.25 : 12.5] [12.5 : 0.05 : 15]]

Capas		Momentum	Adaptativo	a	b	Error	Épocas	Tiempo(segundos)
[1 256 128 128 64 32 16 1]	0,0001	0,75	0,5	0	0	1	1180	283,16
[1 512 256 128 64 32 16 8 1]	0,0001	0,75	0,5	0	0	1	852	722,40
[1 512 256 128 64 32 16 1]	0,0001	0,75	0,5	0	0	1	903	773,41

Arquitecturas óptimas: 1

- 6 capas ocultas ([1 256 128 128 64 32 16 1])
- Épocas: 810
- Rango -15:15
- Distribución de puntos: Mayor densidad en los extremos
- $\beta = 0.75$
- $\eta = 0.00005$

Gráfico arquitectura óptima 1



Arquitecturas óptimas: 2

- 6 capas ocultas ([1 256 128 128 64 32 16 1])
- Épocas: 593
- Rango -15:15
- Distribución de puntos: Mayor densidad en los extremos
- $\beta = 0.75$
- $\eta = 0.0001$

Arquitecturas óptimas: 2

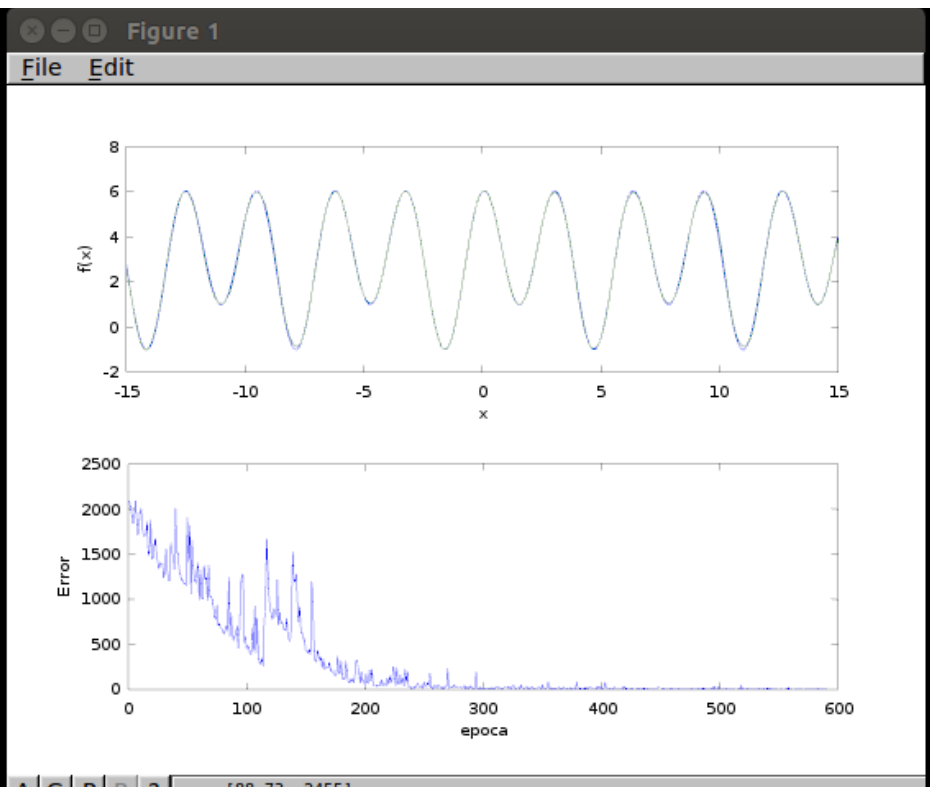
```
eta = 1.00000000000000e-04  
err = 1.86041339990870  
age = 580  
eta = 1.00000000000000e-04  
err = 3.50216618797937  
age = 585  
eta = 1.00000000000000e-04  
err = 2.69846622840732  
age = 590  
eta = 1.00000000000000e-04  
age = 593  
err = 0.789024895796158  
layerSizes =
```

```
1 256 128 128 64 32 16
```

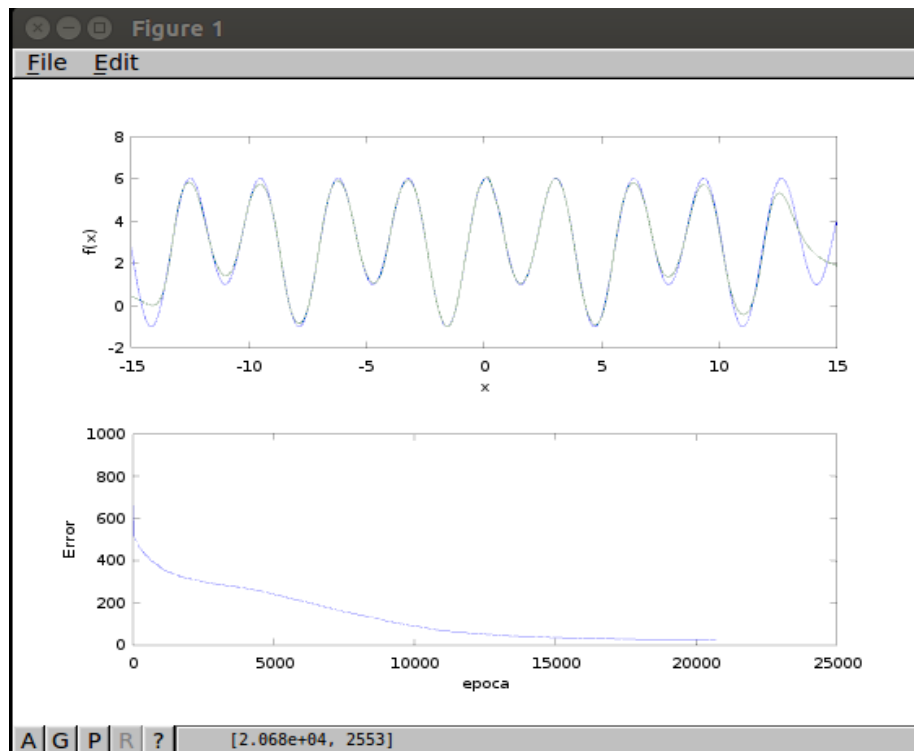
```
eta = 1.00000000000000e-04  
betaValue = 0.75000000000000  
etaAdaptativo = 0  
a = 5.00000000000000e-04  
b = 0.100000000000000  
time = -1430034561.52624
```

He Aprendido!!

Ingrese el valor a probar



Eta adaptativo



Conclusiones

- Más grande es la arquitectura, cantidad de neuronas \rightarrow mayor η
- Se deben utilizar varias capas para resolver funciones más complejas.
- Se deben modelar las capas ocultas del perceptrón como un triángulo
- η adaptativo hace al algoritmo más lento ya que descarta todas las iteraciones que aumentan el error.
- El momentum ayuda en la mayoría de los casos si se usa un valor de alpha de 0,5
- Error con la fórmula de error cuadrático medio de la clase, llegamos a la conclusión de que el error depende de la cantidad de patrones que se estén evaluando, por lo tanto resultaría más acorde dividir por n y no por 2.