

Trabajo Práctico Especial 1

Redes Neuronales

Integrantes del Grupo 11:

Alonso, Juan Manuel

Cavallin, Florencia

Krammer, Esteban

Scomazzon, Martina

OBJETIVOS

- Crear una Red neuronal multicapa supervisada
- Simular terreno tal que parezca real
 - Altura [z]
 - Latitud
 - Longitud
- Distintas arquitecturas
- Obtener Arquitectura Óptima
- Variantes del algoritmo de backpropagation
- Red aprenda

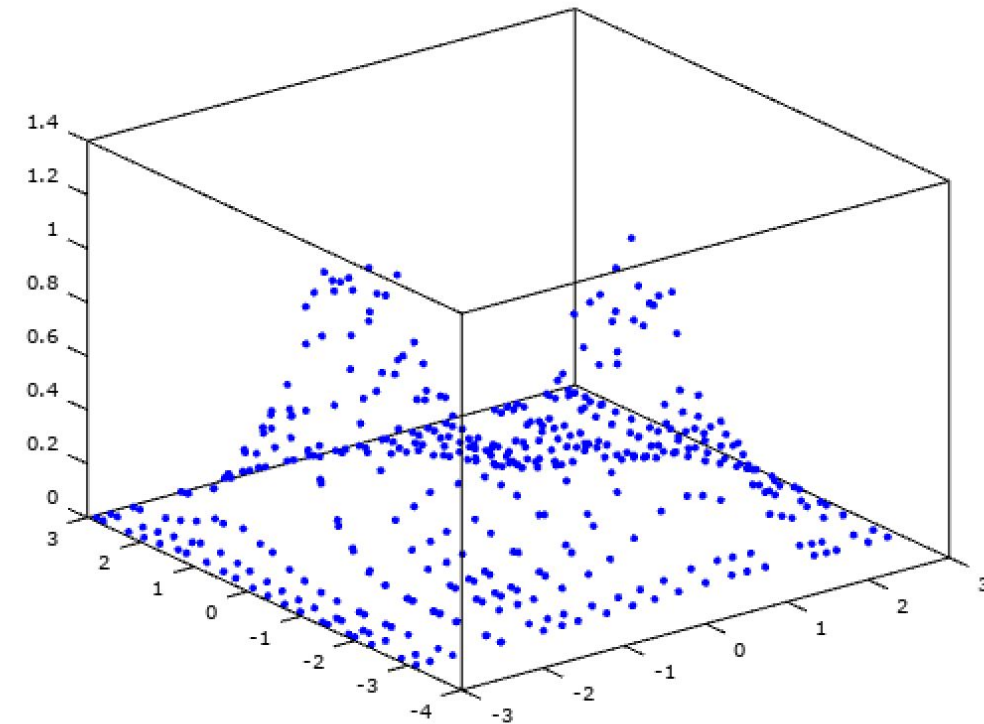


Figura 1. Terreno 11

ARQUITECTURA DE LA RED

- ¿Existe una **arquitectura óptima**?
 - o **Cantidad de capas ocultas**
 - o **Cantidad de neuronas**
 - o **Patrones**
 - Normalización
 - Selección random
 - Tamaño del conjunto de entrenamiento

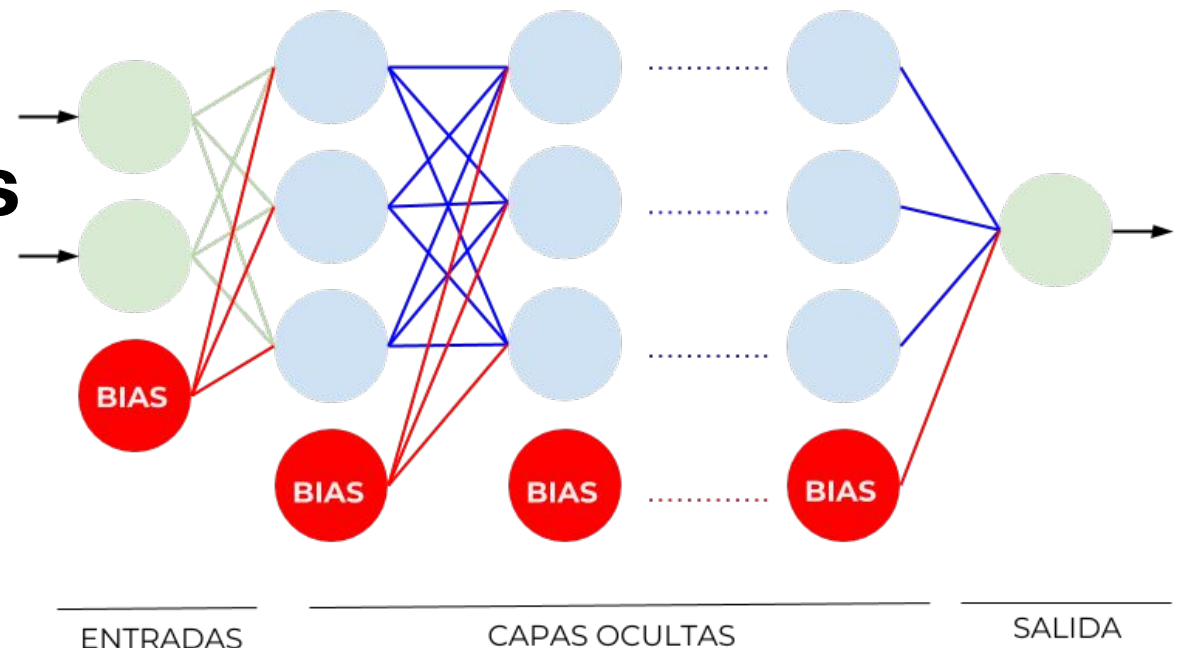


Figura 2. Arquitectura de la red implementada

ARQUITECTURA DE LA RED

Función de Activación

$\tanh(x)$

$\text{exponentialSigmoid}(x)$

Pesos

Cambio de pesos

Error

Mejoras a Backpropagation

Momentum

η adaptativo

MEJORAS ALGORITMO:

¿POR QUÉ? ¿TODO SIRVE?

- **η Adaptativo**
 - ¿Cuándo cambia?
 - Cambio por épocas ¿Por qué?
 - Error
- **Momentum**
 - Mínimos
 - Gradiente descendiente
- **Función de activación**
 - Salida
- **¿Mejoras?**

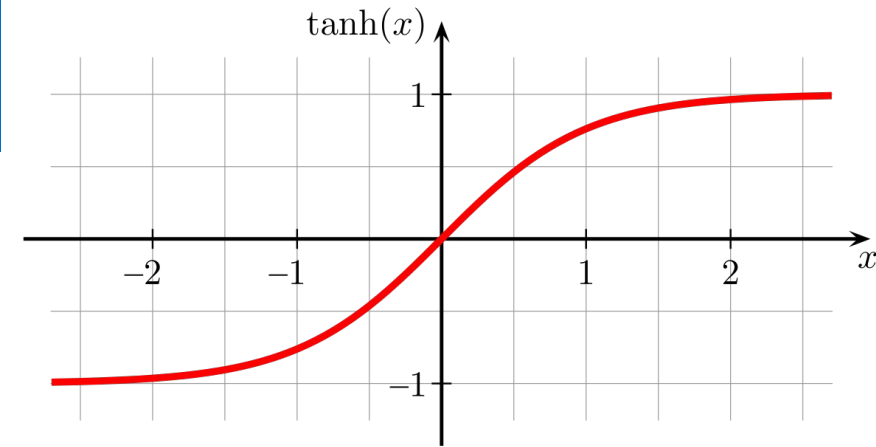


Figura 4. Función $\tanh(x)$

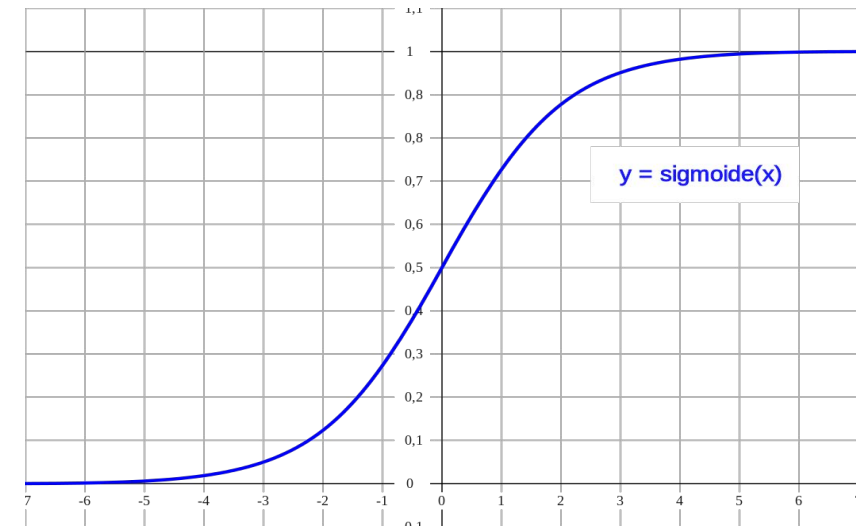
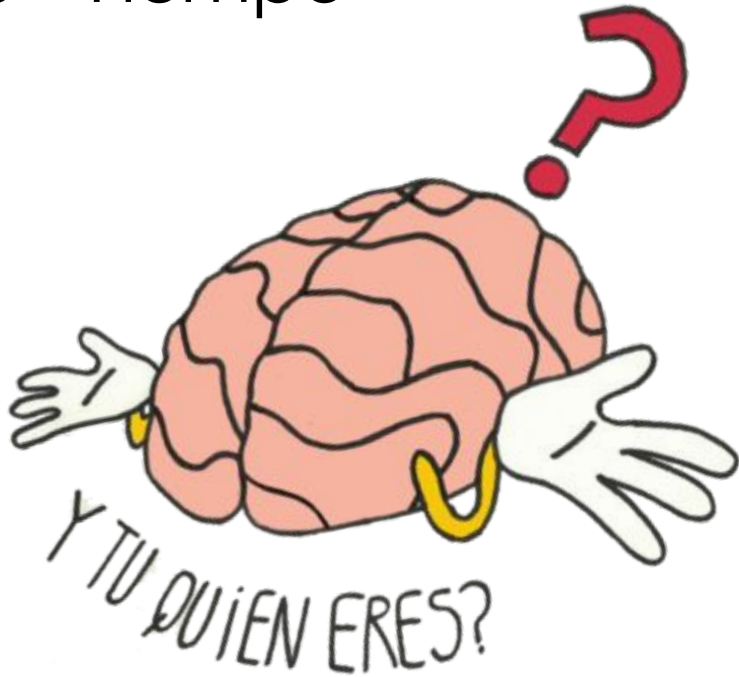


Figura 5. Función $\text{sigmoide}(x)$

MÉTRICAS

- Error
- Épocas
- Tiempo



ENTRENAMIENTO VS TESTEO

- Entreno con determinada cantidad de patrones
- Testeo contra patrones que no hayan estado involucrados en el entrenamiento
- Aproximación
- Success Rate

PRUEBAS

Efecto de los pesos random (var ctes)

Cantidad de capas ocultas: 3
Cantidad de neuronas en cada capa: [15 15 15]
Porcentaje de entrenamiento: 50%

Tiempo	Épocas	Mejoras	η_{inicial}	η_{final}	m	Error	Success Rate[1]
323.246	23	n + m	0.01	0.015604	0.9	0.01	62.8959 %
426.942	29	n + m	0.01	0.0196	0.9	0.01	65.6109 %
499.417	33	n + m	0.01	0.019604	0.9	0.01	61.991 %
478.083	36	n + m	0.01	0.021604	0.9	0.01	70.1357 %
1500.01	78	n + m	0.01	0.0187	0.9	0.01	70.1357%

Sin/Con η adapt. y momentum

Cantidad de capas ocultas: 2
Cantidad de neuronas en cada capa: [3 3]
Porcentaje de entrenamiento: 50%

Tiempo	Épocas	Mejoras	η_{inicial}	η_{final}	m	Error	Success Rate[1]
21.8814	26	No	0.1	0.1	0	0.01	74.2081 %
30.7502	32	No	0.1	0.1	0	0.01	64.2534 %
72.9128	84	No	0.2	0.2	0	0.01	75.5656 %
17.6731	21	n	0.1	0.104	0	0.01	67.4208 %
41.1221	50	n + m	0.1	0.116	0.5	0.01	74.2081 %
101.18	122	n + m	0.1	0.04960	0.9	0.01	65.6109 %

Incrementos en momentum y η adapt.

Cantidad de capas ocultas: 2
Cantidad de neuronas en cada capa: [10 10]
Porcentaje de entrenamiento: 60%

Tiempo	Épocas	Mejoras	η_{inicial}	η_{final}	m	Error	Success Rate[1]
261.242	48	n + m	0.01	0.02560	0.5	0.001	62.6415 %
121.769	22	n + m	0.05	0.05700	0.5	0.001	71.6981 %
67.9344	14	n + m	0.1	0.10200	0.5	0.001	73.962 %
247.885	50	n + m	0.01	0.02760	0.6	0.001	67.9245 %
109.695	22	n + m	0.05	0.05500	0.6	0.001	70.9434 %
90.2761	18	n + m	0.1	0.10400	0.6	0.001	60.3774 %
344.602	68	n + m	0.01	0.033604	0.7	0.001	74.3396 %
81.6888	16	n + m	0.05	0.10400	0.7	0.001	71.6981 %
102.798	21	n + m	0.1	0.10600	0.7	0.001	68.8113 %

Incrementos en momentum

Cantidad de capas ocultas: 2
Cantidad de neuronas en cada capa: [3 3]
Porcentaje de entrenamiento: 50%

Tiempo	Épocas	Mejoras	η_{inicial}	η_{final}	m	Error	Success Rate[1]
112.784	17	m	0.05	0.05	0.5	0.001	66.7925 %
100.564	13	m	0.05	0.05	0.6	0.001	75.0943 %
49.7843	10	m	0.05	0.05	0.7	0.001	72.4528 %
188.428	24	m	0.05	0.05	0.8	0.001	54.3396 %

Neuronas vs. Success Rate

Cantidad de capas ocultas: 2
Porcentaje de entrenamiento: 50%

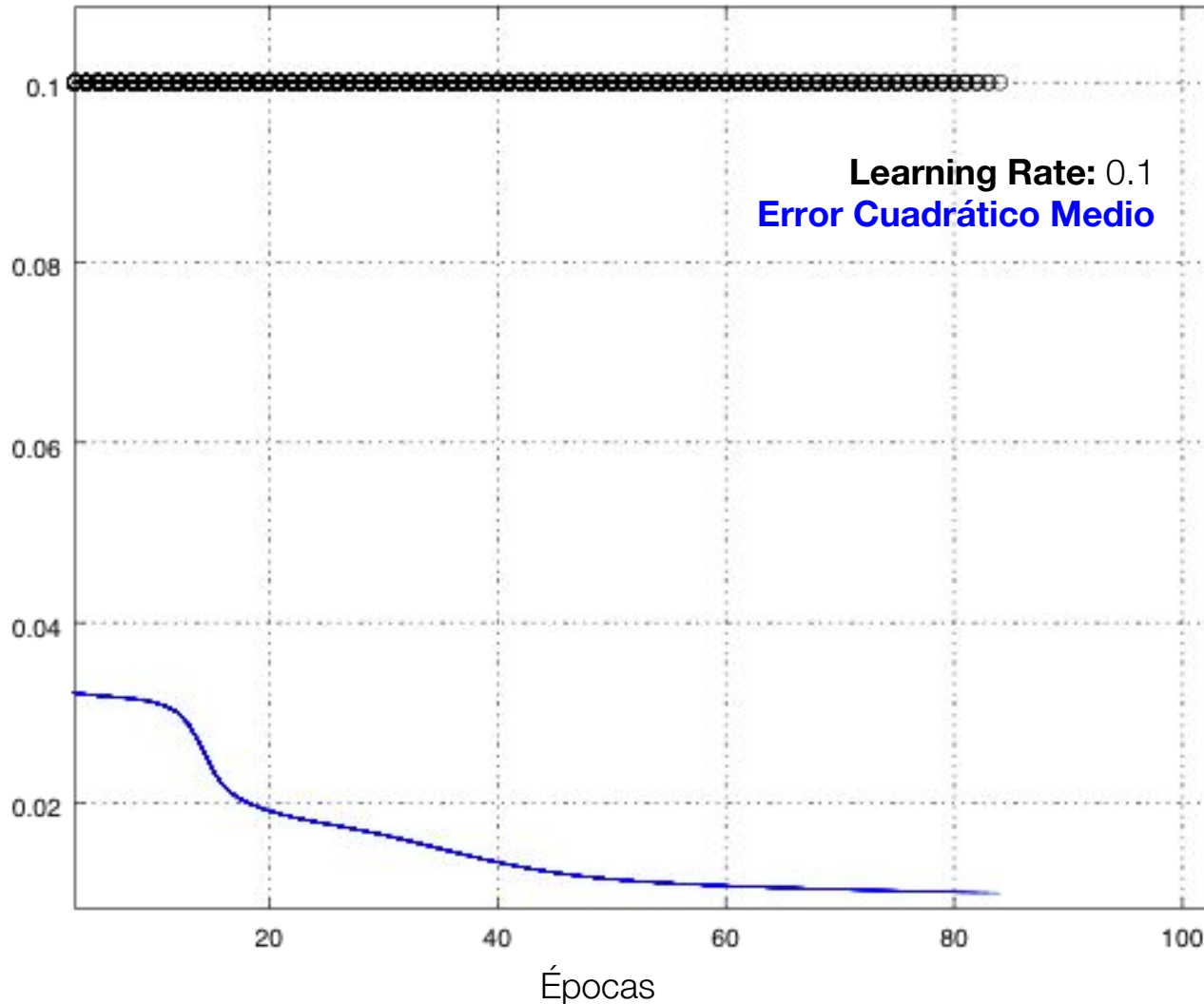
Tiempo	Épocas	Mejoras	η	Capas	Error	Success Rate
35.0707	39	No	0.1	[5 5]	0.01	61.6541 %
32.6875	26	No	0.1	[5 8]	0.01	51.8797 %
79.7794	42	No	0.1	[15 10]	0.01	74.4361 %
148.946	34	No	0.1	[15 15]	0.01	72.9323 %

% Entrenamiento vs. Success Rate

Cantidad de capas ocultas: 2
Cantidad de neuronas en cada capa: [10 10]

Tiempo	Épocas	Mejoras	η	% Entrenamiento	Error	Success Rate
19.9676	26	No	0.1	10	0.01	51.1111 %
67.4614	50	No	0.1	20	0.01	62.9213 %
79.7794	50	No	0.1	30	0.01	67.6692 %
128.794	50	No	0.1	40	0.01	69.9998 %

ALGUNOS GRÁFICOS



2 **Capas ocultas**

[3 3]

Algoritmo **Incremental**
tanh(x)

Cantidad de Patrones: 50%

Cota de error: 0.01

Sin eta adaptativo

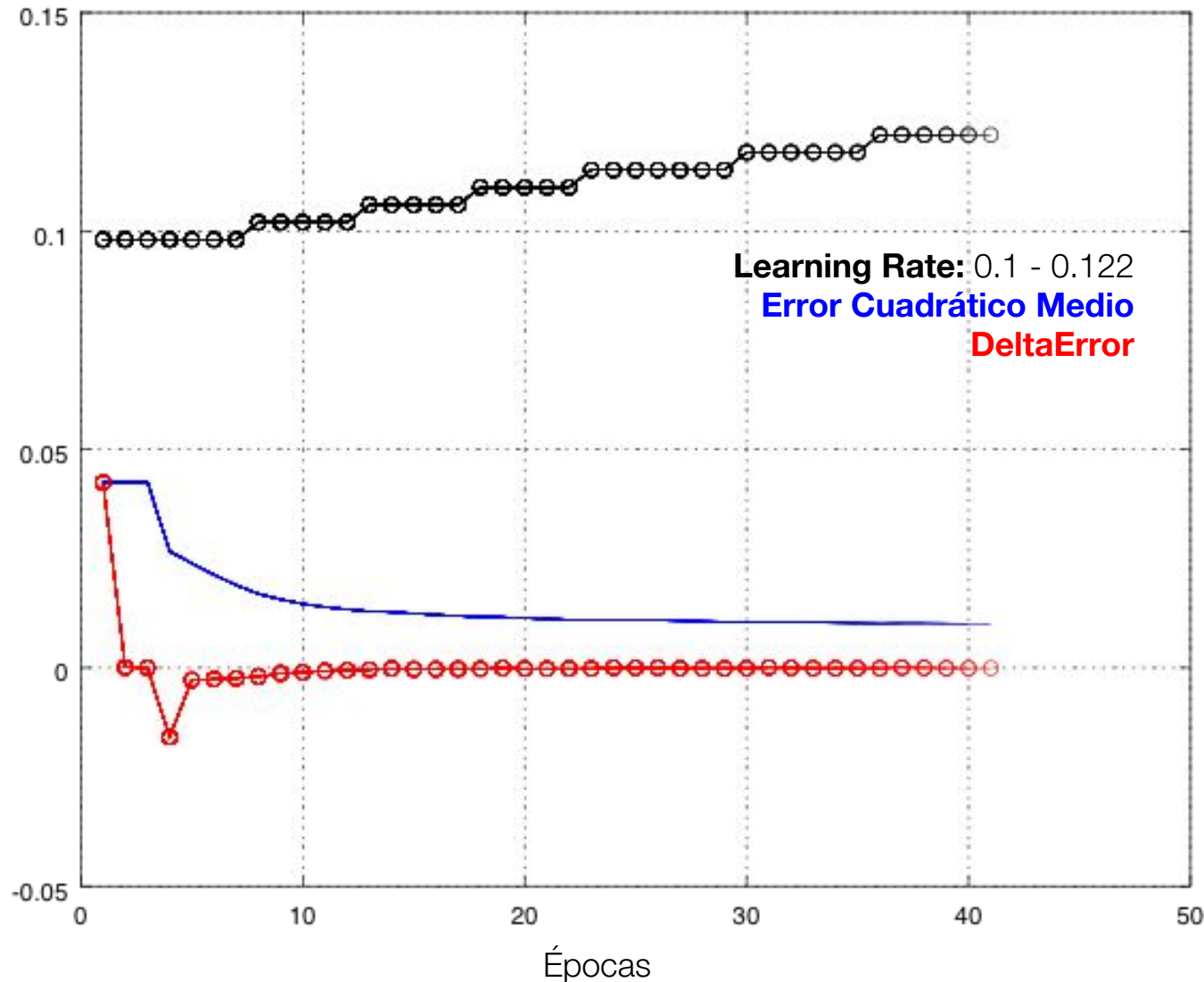
Sin momentum

Épocas: 84

Success Rate: 75.5656 %

Tiempo: 72.9 seg

ALGUNOS GRÁFICOS



2 Capas ocultas

[13 14]

Algoritmo **Incremental**
tanh(x)

Cantidad de Patrones: 50%

Cota de error: 0.01

Con eta adaptativo

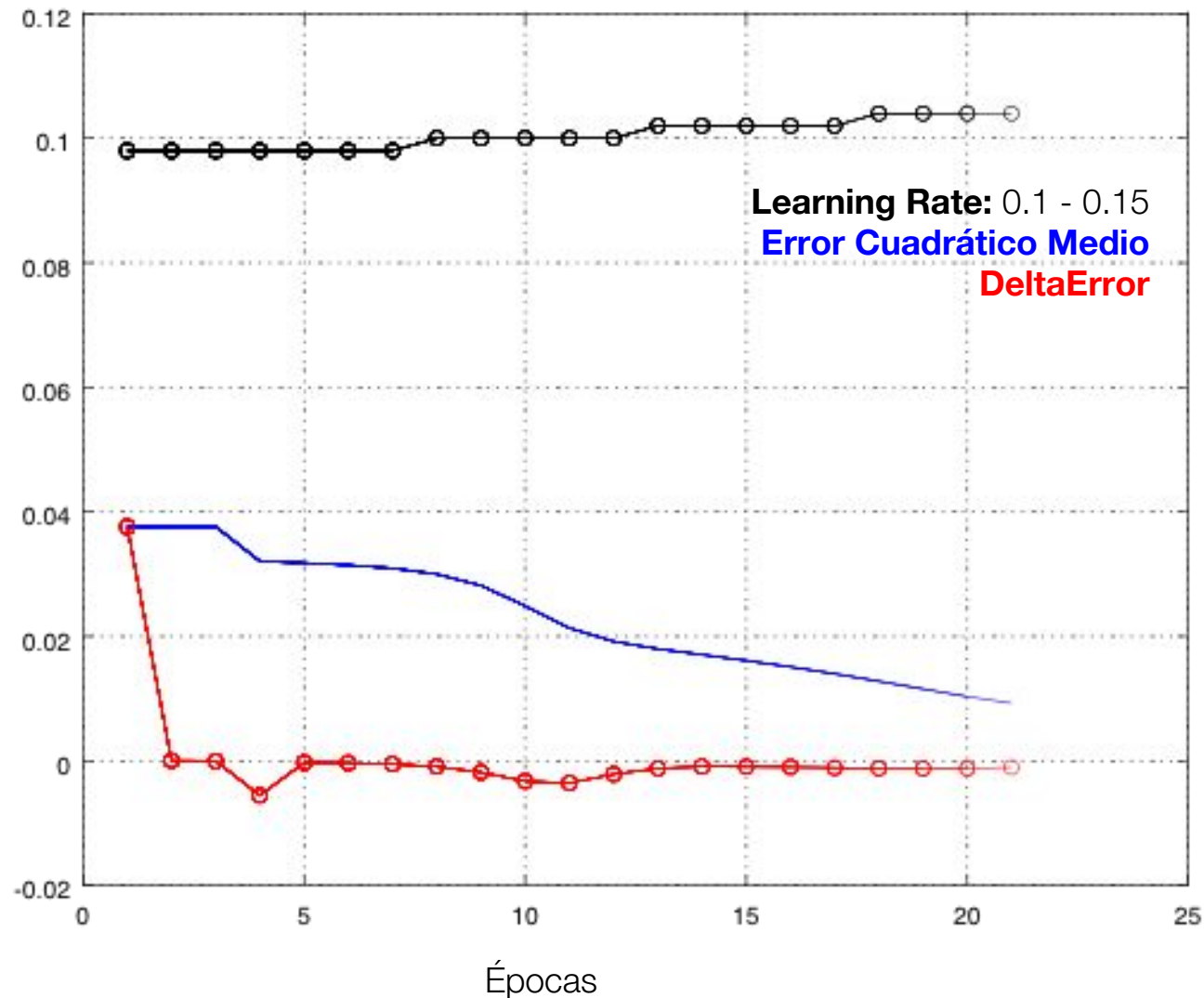
Sin momentum

Épocas: 41

Success Rate: 73.3032 %

Tiempo: 272 seg

ALGUNOS GRÁFICOS



3 Capas ocultas

[15 15 15]

Algoritmo **Incremental**
tanh(x)

Cantidad de Patrones: 50%

Cota de error: 0.01

Con eta adaptativo

Con momentum

Épocas: 23

Success Rate: 62.856 % (+)

Tiempo: 478 seg

ANÁLISIS DE RESULTADOS

- Importancia de la selección de patrones
- Importancia de la selección de pesos
- Mejores resultados cuando se combina alto momentum con un valor bajo de η inicial
- Mayor finalización del algoritmo por la cota de error cuadrático medio y no por la cantidad de épocas
- η adaptativo y momentum no siempre significa mejores resultados
- Vanishing Gradient con exponencial

ANÁLISIS DE RESULTADOS

- Más patrones de entrenamiento, mejor resultado
- No pudimos sacar una tendencia en cuanto a la cantidad de neuronas por capa

CONCLUSIONES

- No encontramos una arquitectura particular que satisfice nuestras expectativas.
- Existen varias arquitecturas que dan resultados similares
- No se requieren más de 2 capas ocultas
- Entre 3 y 10 neuronas por capa oculta
- Más patrones de entrenamiento, más neuronas
- Más grande el momentum, más pequeño el η
- No es recomendable el uso de la función exponencial