

Sistemas de Inteligencia Artificial

Redes Neuronales

Grupo 1

Implementación

Parametrización

- Patrones
 - 70% para entrenamiento
 - 30% para testeo
 - Elegidos pseudo-aleatoriamente
- η (eta)
- ε (epsilon)
- α (alpha)
- Habilidad/deshabilidad de η adaptativo
- Habilidad/deshabilidad de variación de los deltas de cada capa

Parametrización

- Pesos iniciales
 - Obtenidos pseudo-aleatoriamente
 - $[-\sigma_m, \sigma_m]$
 - m la cantidad de conexiones entrantes a esa unidad
 - $\sigma_m = m^{-1/2}$
- Arquitectura de la red
 - Cantidad de capas ocultas
 - Cantidad de neuronas en cada capa oculta
 - Función de activación y su derivada de las capas ocultas
 - Función de activación y su derivada de la capa de salida

Durante el algoritmo de aprendizaje

- Por cada época
 - Se calcula el error global de entrenamiento y el de testeo
 - Opcionalmente: se aplica la optimización del eta adaptativo mediante el cálculo de la variación del error global de entrenamiento y aplicando rollback si necesario
- Guardado de red
 - Cada un cierto intervalo de épocas definido por el usuario
 - Al detectar una “intersección” entre el error global de entrenamiento y de testeo normalizados
- Gráficos
 - Errores
 - Evolución del aprendizaje de la función

Condiciones de corte

- Threshold ε de entrenamiento
 - Para todo patrón de entrenamiento, la diferencia entre la salida de la red neuronal y la esperada sea menor a ε
- Forzar la finalización del proceso de entrenamiento
 - Se puede acceder a las redes guardadas previo al corte

Variaciones/Optimizaciones

- $g' + 0.1$ en el cálculo de los delta
- Momentum (α)
- Eta (η) adaptativo

Criterios de análisis

Error de entrenamiento y testeo

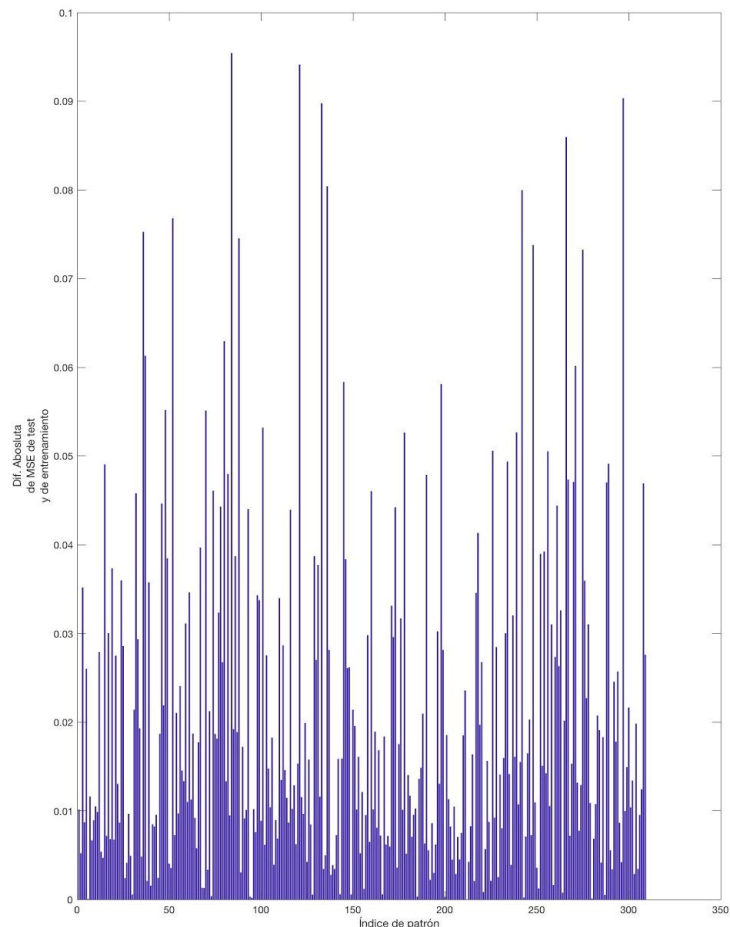
- Error cuadrático medio de entrenamiento y testeo en cada época
- Gráficos *en tiempo real* para poder visualizar el entrenamiento de la red neuronal
 - Diferencia en módulo del valor generado por la red neuronal y el esperado
 - Evolución de los errores a medida que pasan épocas

Ventajas:

- Detección de falta de entrenamiento observando la curva de la evolución del error cuadrático medio de entrenamiento
- Detección de sobre entrenamiento observando ambas curvas

Distancia lineal por patrón

- Se toma cada entrada del conjunto de patrones de entrenamiento
- Se grafica una barra cuya altura es la diferencia absoluta entre la salida esperada y la salida de la red neuronal para esa época



Terrenos generados

- Permiten visualizar cómo evoluciona la red como función aproximadora a través de las sucesivas épocas.
- Durante el entrenamiento de la red (cada una cierta cantidad de épocas)
 - Gráfico de la superficie del terreno generado por todos los patrones
 - Gráfico de los puntos del terreno generado por todos los patrones
- Al finalizar el entrenamiento
 - Gráfico de la superficie del terreno generado por un conjunto de puntos tomados pseudo-aleatoriamente dentro del dominio de los patrones dados
 - Gráfico de los puntos del terreno generado por el conjunto anterior y los puntos de la función original

Comparaciones al cambiar el número de patrones

Comparar el error cuadrático medio (normalizado) de testeo luego de entrenamiento de las redes

Condiciones:

- Misma arquitectura de red para ambas
- Diferente cantidad de patrones elegidos

Capacidad de generalización de la red

x = # de errores (diferencia entre salidas esperadas y salidas de la red) por debajo de ϵ

y = # total de patrones en el conjunto de patrones de testeo

$$\text{CGR} = x / y * 100$$

Problemas encontrados

Patrones

Salidas esperadas entre $[0, 1]$ (en general).

Problema:

- $\exists!$ salida esperada > 1 que vale **1.0553**

Posibles soluciones:

- **Función lineal identidad en la capa de salida**
- Siempre tomar ese patrón dentro del conjunto de patrones de testeo

Función de identidad lineal

Problema:

- Pesos de las conexiones tienden a ∞ cuando el valor de $\eta \geq 0.5$ (comprobado experimentalmente cambiando arbitrariamente la arquitectura de la red)

Posibles soluciones:

- Decrementar η
- No utilizar η adaptativo

Casos de análisis

Caso 1

- Objetivo: Analizar el comportamiento y aprendizaje de la red
- Configuración de parámetros que se encontró empíricamente pertinente
- Pasos
 - Configuración inicial de 4 capas y neuronas
 - Reducir en uno la cantidad de capas ocultas
 - Reducir la cantidad de neuronas en las capas ocultas
 - Reducir a 2 la cantidad de capas ocultas

Caso 2

- Objetivo: Comparar la performance de entrenamiento de la red
- Habilitar y deshabilitar las diferentes optimizaciones implementadas

Caso 3

- Objetivos:
 - Comprobar que la utilización de una función exponencial en las capas ocultas no es una buena decisión en cuanto al alcance de la condición de corte
 - Comprobar que la utilización de la tangente hiperbólica en todas las capas (incluyendo la de salida) resulta en el problema mencionado previamente

Caso 4

- Objetivos:
 - Comparar los errores cuadráticos medios normalizados obtenidos al entrenar una red con una arquitectura en particular, utilizando una cantidad diferente de patrones en cada caso

Resultados

Parámetros y aclaraciones

- A menos especificado la contrario, se utilizó
 - $\varepsilon = 0.1$
 - $\alpha = 0.9$
 - $\eta = 0.1$
 - $\beta = 1$
 - El conjunto de patrones provisto por la cátedra
- Notación de tablas
 - + .1: optimización en el cálculo de los delta
 - α : optimización de momentum
 - η : optimización del η adaptativo

Caso 1

Neuronas en capas ocultas	g salida	g ocultas	+ .1	α	η	# épocas entrenamiento	Capacidad de generalización de la red (%)
10, 5, 5, 3	lineal	tanh	Sí	Sí	No	263	96.21
10, 10, 5	lineal	tanh	Sí	Sí	No	251	98
5, 5, 2	lineal	tanh	Sí	Sí	No	871	96
12, 3	lineal	tanh	Sí	Sí	No	1382	99.24
15, 2	lineal	tanh	Sí	Sí	No	740	96.21

Caso 2

Neuronas en capas ocultas	g salida	g ocultas	+ .1	α	η	# épocas entrenamiento	Capacidad de generalización de la red (%)
10, 10, 15	lineal	tanh	Sí	Sí	Sí	-	-
			Sí	Sí	No	2730	94.70
			Sí	No	No	> 4515	-
			No	No	No	4261	93.18

$$\varepsilon = 0.05$$

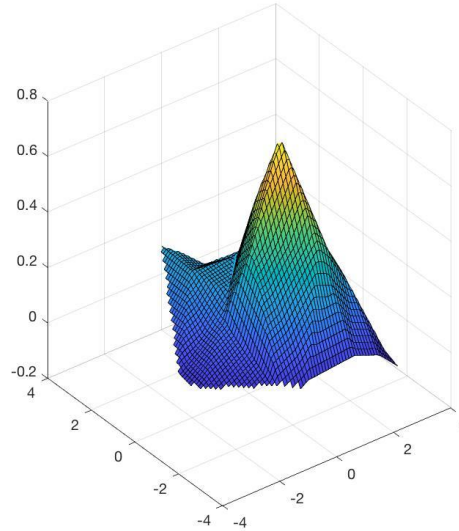
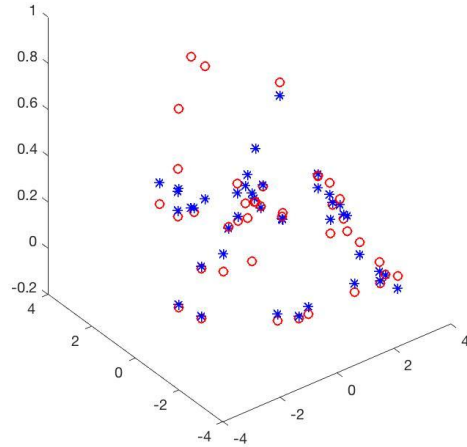
Caso 3

Neuronas en capas ocultas	g salida	g ocultas	+ .1	α	η	# épocas entrenamiento	Capacidad de generalización de la red (%)
12,3	lineal	exp	Sí	Sí	No	~3500	-
12,3	tanh	tanh	Sí	Sí	No	~3000	-

Caso 4

Neuronas en capas ocultas	g salida	g ocultas	+ .1	α	η	Cantidad de patrones	# épocas entrenamiento	MSE de testeo normalizado
10, 10, 5	lineal	tanh	Sí	Sí	No	441	251	7,88E-04
10, 10, 5	lineal	tanh	Sí	Sí	No	341	685	4.78e-4
10, 10, 5	lineal	tanh	Sí	Sí	No	241	430	0.0016
10, 10, 5	lineal	tanh	Sí	Sí	No	141	478	0.0070
10, 10, 5	lineal	tanh	Sí	Sí	No	41	662	0.0457

Gráfico de puntos / Mesh



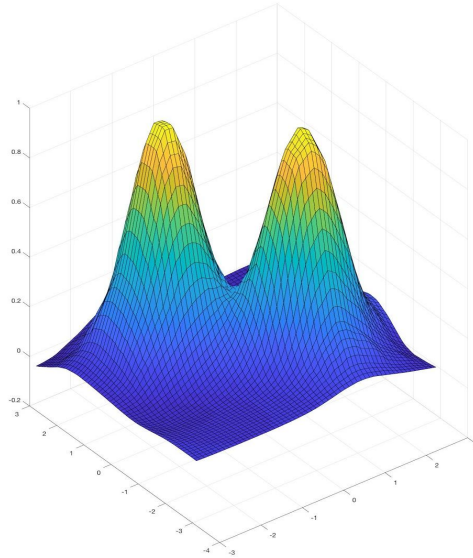
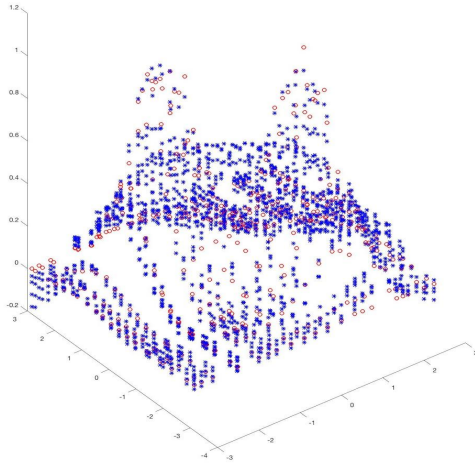
Red: [10,10,5]

Patrones = 41

g (capas ocultas) : tanh

g (capa de salida) : lineal

Gráfico de puntos / Mesh



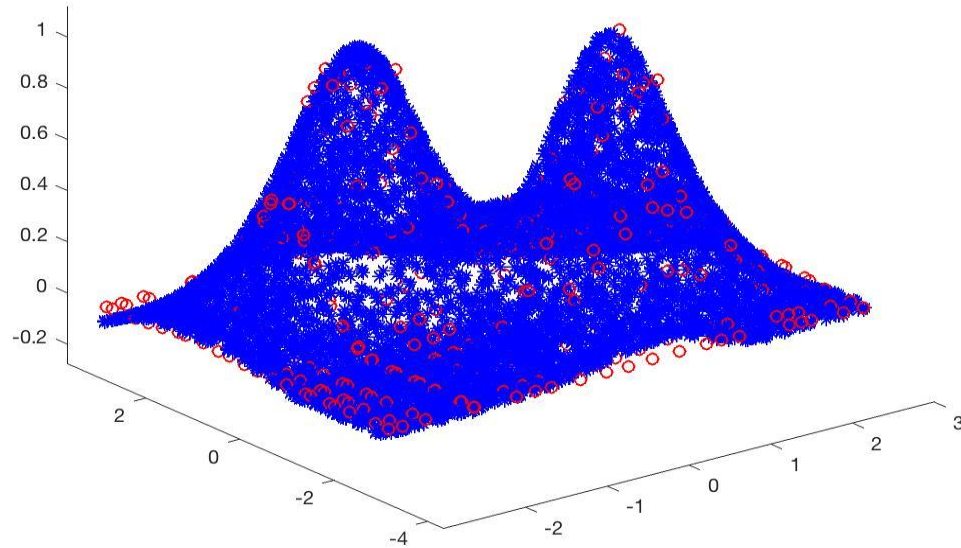
Red: [10,10,5]

Patrones = 441

g (capas ocultas) : tanh

g (capa de salida) : lineal

Terreno



Red: [15,2]

g (capas ocultas) : tanh

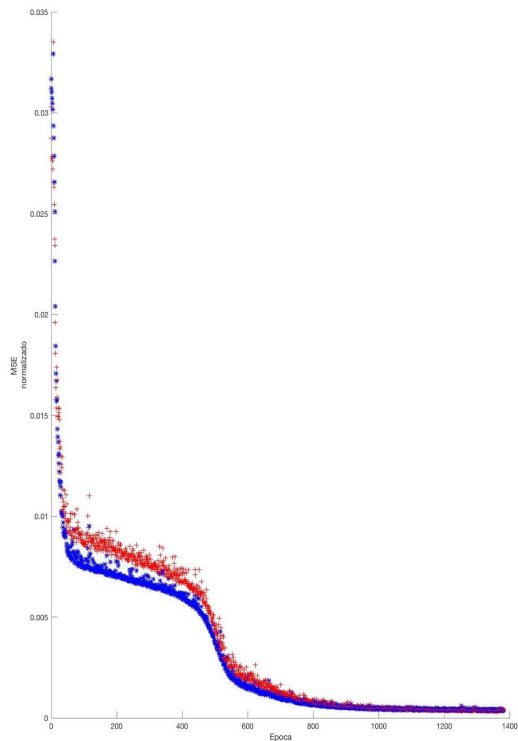
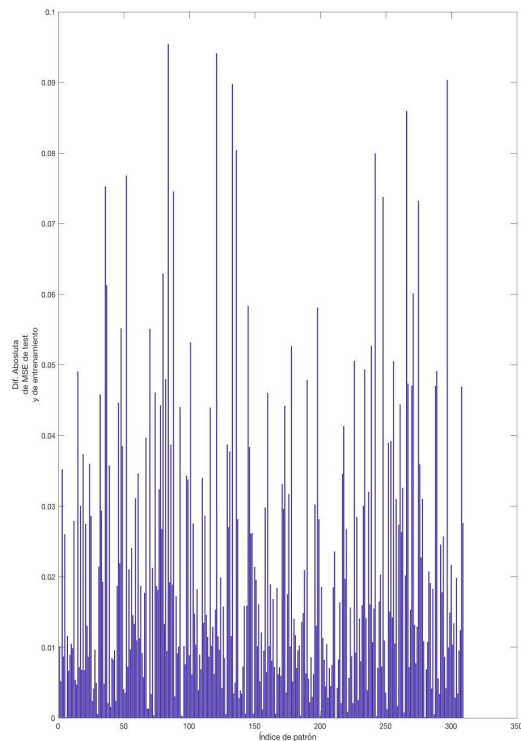
g (capa de salida) : lineal

Rango de valores:

$-3 \leq X \leq 3$

$-3 \leq Y \leq 3$

Dif. entre salidas / E. de entrenamiento y testeo

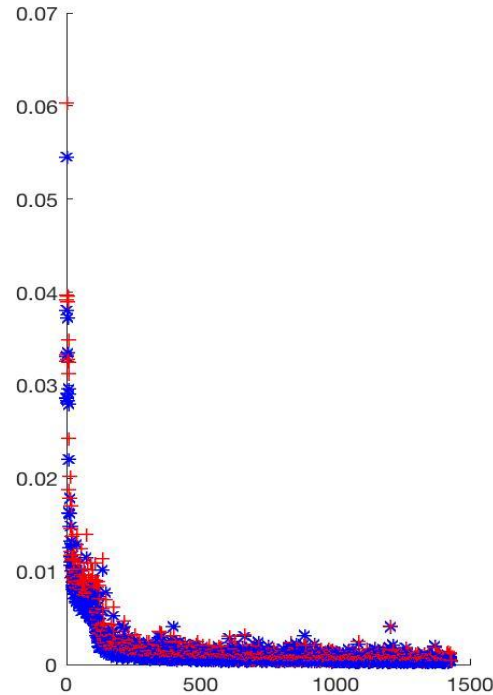
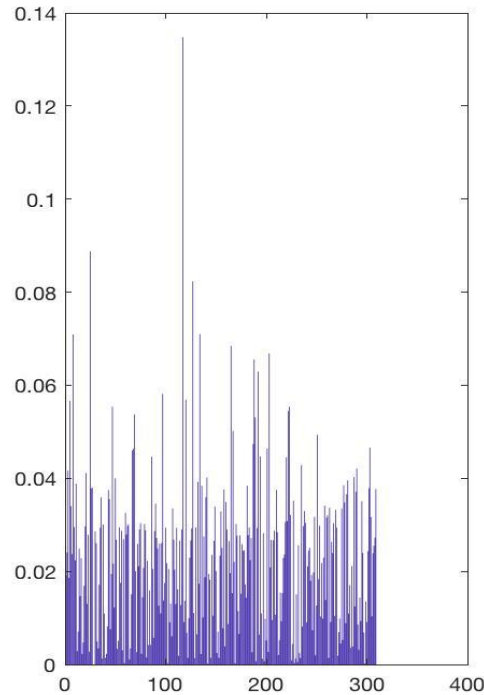


Red: [12,3]

g (capas ocultas) : tanh

g (capa de salida) : lineal

Caso del punto máximo



Red: [5,5,5]

g (capas ocultas) : tanh
g (capa de salida) : tanh

Conclusiones

Conclusiones

- Cantidad de capas y/o neuronas no proporcional a la cantidad de épocas para el entrenamiento
- Capacidad de generalización no está directamente asociada con la velocidad de entrenamiento, sino con la calidad del mismo
- No se pudo probar empíricamente que el η adaptativo fuese una optimización
- Para poder observar verdaderamente las optimizaciones, se debe forzar a la red a entrenar con el nivel de exigencia adecuado para ello
- La configuración de red óptima depende del problema, de la precisión de generalización deseada, del tiempo disponible para entrenar, de la cantidad de patrones disponibles, etc.
- Mejor red: Caso 2, segunda red

¿Preguntas?
