

Redes Neuronales

Grupo 8 - Integrantes

52194 - Ducret, Argentino

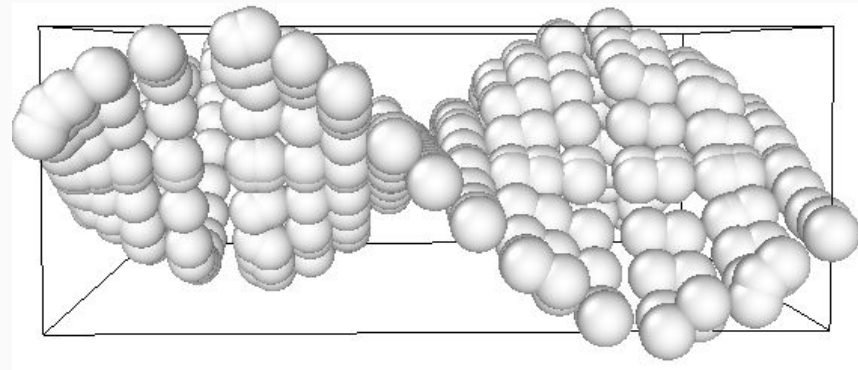
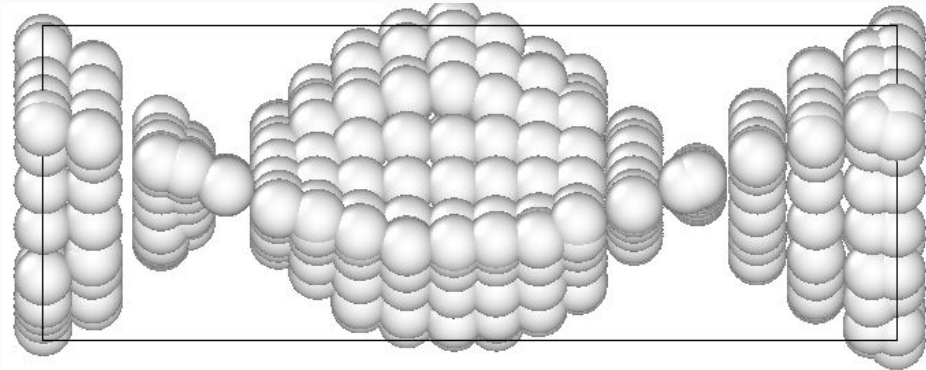
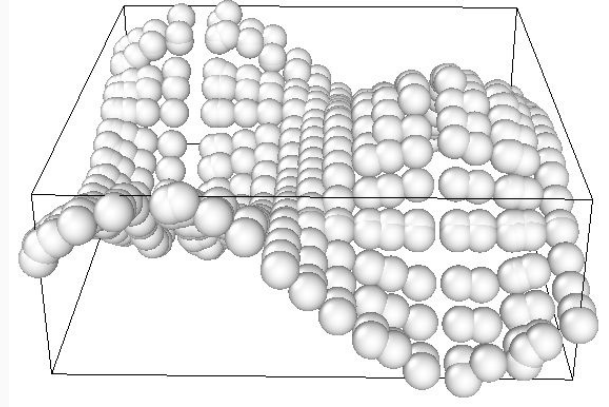
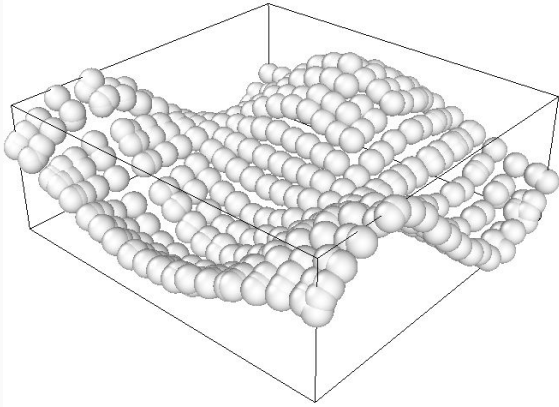
54263 - Prudhomme, Franco

54293 - Gutierrez, Ignacio

Objetivo

- Implementar una red neuronal multicapa con aprendizaje supervisado.
- Esta red deberá aprender un terreno (funcion en 3D).
- Utilizando las estrategias:
 - ◆ Multicapa incremental.
 - ◆ Multicapa incremental con momentum.
 - ◆ Multicapa incremental con etha adaptativo.
- Determinar una estructura capaz de aprender la función dada.

Terreno a estudiar



Implementación

Red neuronal representada de la siguiente forma:

- **cell array** conteniendo todos los inputs.
- **cell array** conteniendo en cada índice una matriz con los pesos de las conexiones.
- **cell array** conteniendo en cada índice una matriz con los resultados de cada neurona.

Funciones de activación

→ Tangente hiperbólica:

$$f(x) = \tanh(x * \text{betha})$$

$$f'(x) = \text{betha} * (1 - x^2)$$

→ Exponencial:

$$f(x) = (1 + e^{-2x * \text{betha}})^{-1}$$

$$f(x) = 2 * \text{betha} * x * (1 - x)$$

Conjuntos de entrenamiento y testeo

Dado un porcentaje se separaban los puntos de la muestra en dos subconjuntos:

- Entrenamiento
- Testeo

Se eligieron puntos al azar y no representativos.

Mejoras al algoritmo backpropagation

Fueron dos:

→ Momentum

→ Eta adaptativo

Momentum

La idea es dar a cada conexión W cierta inercia, de modo que tiende a cambiar en la dirección promedio de la fuerza que siente, en lugar oscilar violentamente con cada cambio.

El momentum (parámetro α) debe estar entre 0 y 1.

Eta adaptativo

No es fácil encontrar un α o η óptimo para un problema, por esto una técnica recomendada es ajustar los parámetros automáticamente.

- Se chequea si se redujo el error durante k épocas.
- Si esto ocurre entonces se debe aumentar η .
- En caso contrario se debe reducirlo.

Para esto se reciben 3 parámetros, cantidad de épocas a esperar, en cuanto incrementar η y en qué porcentaje disminuirlo.

Elección de arquitectura para la red

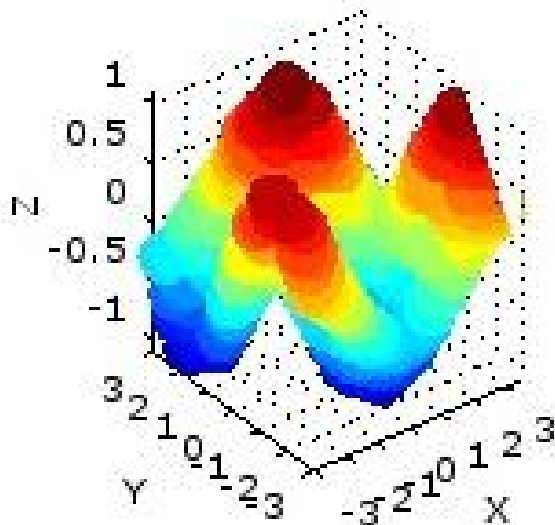
- Cantidad de capas ocultas
- Cantidad de neuronas por capas
- Eta
- Función de activación
- Beta

Incremental (sin mejoras)

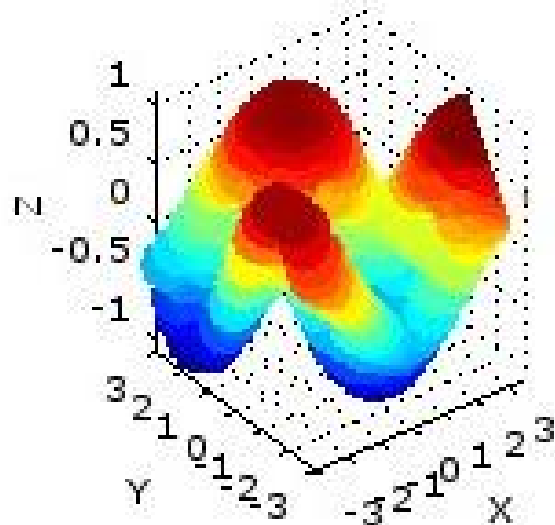
Estructura de la red	% de inputs	Función activación	etha	betha	alpha	épocas	Error mínimo de entrenamiento	% aprendido
[2 2 1]	100	tanh	0.1	0.5	0	750	0.0494377	23
[2 2 2 1]	100	tanh	0.1	0.5	0	900	0.0409222	31
[6 5 1]	100	tanh	0.1	0.5	0	3150	0.000938275	95
[6 5 4 1]	100	tanh	0.1	0.5	0	1649	0.000494431	98
[6 5 4 1]	100	exp	0.1	0.5	0	600	0.107839	20
[6 5 4 1]	100	tanh	0.1	0.75	0	690	0.000471692	99
[6 5 4 1]	100	tanh	0.1	0.25	0	900	0.00344382	76
[6 5 4 1]	100	tanh	0.1	0.875	0	767	0.000485247	99
[7 1]	100	tanh	0.1	0.75	0	1800	0.00134028	95
[6 5 4 1]	100	tanh	0.5	0.75	0	900	0.00172682	87
[6 5 4 1]	100	tanh	0.05	0.75	0	4244	0.000499474	99
[6 5 4 1]	75	tanh	0.1	0.75	0	1030	0.000486636	94
[6 5 4 1]	50	tanh	0.1	0.75	0	2185	0.000491268	90
[6 5 4 1]	25	tanh	0.1	0.75	0	1676	0.000491595	84
[6 5 1]	75	tanh	0.1	0.75	0	1200	0.00135565	85
[4 5 6 1]	100	tanh	0.1	0.75	0	900	0.00113088	96

Incremental (sin mejoras)

Original function



TrainingSet vs outputs



Momentum

Estructura de la red	% de inputs	Función activación	etha	betha	alpha	épocas	Min Training Error	% aprendido
[6 7 7 1]	100%	tanh	0.02	0.8	0.4	700	0.000660157	97%
[6 5 4 1]	100%	tanh	0.01	0.75	0.9	800	0.000538703	99%
[6 7 7 1]	50%	tanh	0.02	0.4	0.8	2200	0.00251108	68%
[6 5 4 1]	50%	tanh	0.01	0.75	0.9	4300	0.00032502	70%
[6 5 4 1]	50%	tanh	0.01	0.75	0.6	3300	0.000857664	86%
[6 5 4 1]	50%	tanh	0.01	0.75	0.5	3000	0.000890983	91%
[6 5 4 1]	30%	tanh	0.1	0.2	0.9	2100	0.00124841	75%
[6 5 4 1]	25%	tanh	0.01	0.75	0.35	1600	0.00193145	78%
[6 5 4 1]	25%	tanh	0.01	0.75	0.35	6300	0.000921665	90%

Eta adaptativo - a

Estructura de la red	% de inputs	Función activación	eta	beta	a	b	k épocas	épocas	Error mínimo	% aprendido
[6 5 4 1]	25	tanh	0.1	0.75	0.01	0.05	2	2300	0.00258682	78
[6 5 4 1]	25	tanh	0.1	0.75	0.1	0.05	2	2300	0.000465755	86,52
[6 5 4 1]	25	tanh	0.1	0.75	0.2	0.05	2	2300	0.000608095	87,07
[6 5 4 1]	25	tanh	0.1	0.75	0.3	0.05	2	2300	0.000598901	81,86
[6 5 4 1]	25	tanh	0.1	0.75	0.4	0.05	2	2300	0.00519381	75,96

Eta adaptativo - b

Estructura de la red	% de inputs	Función activación	eta	beta	a	b	k épocas	épocas	Error mínimo	% aprendido
[6 5 4 1]	100	tanh	0.1	0.75	0.2	0.05	2	1600	0.000331447	1
[6 5 4 1]	100	tanh	0.1	0.75	0.2	0.1	2	1400	0.000640667	98,18
[6 5 4 1]	100	tanh	0.1	0.75	0.2	0.2	2	1300	0.000996232	95,46
[6 5 4 1]	100	tanh	0.1	0.75	0.2	0.3	2	2100	0.00108314	90,92
[6 5 4 1]	100	tanh	0.1	0.75	0.2	0.4	2	2200	0.00169906	92,06

Eta adaptativo - k épocas

Estructura de la red	% de inputs	Función activación	eta	beta	a	b	k épocas	épocas	Error mínimo	% aprendido
[6 5 4 1]	100	tanh	0.1	0.75	0.01	0.01	2	1900	0.000341132	99,77
[6 5 4 1]	100	tanh	0.1	0.75	0.01	0.01	4	2400	0.000747129	97,73
[6 5 4 1]	100	tanh	0.1	0.75	0.01	0.01	5	2400	0.00074476	97,31
[6 5 4 1]	100	tanh	0.1	0.75	0.01	0.01	6	2400	0.000624499	98,41

Resultados de arquitecturas elegidas

- ❖ Capas ocultas: 6 5 4
- ❖ eta: 0.1
- ❖ beta: 0.75
- Incremental sin mejoras
- Incremental momentum
 - ◆ eta: 0.01
 - ◆ alpha: 0.35 - 0.6 - 0.9 (inputs: 25% - 50% - 100%)
- Incremental eta adaptativo
 - ◆ a: 0.1
 - ◆ b: 0.05
 - ◆ k: 2

Conclusiones

- La función de activación tangente hiperbólica dio mejores resultados que la exponencial.
- La configuración óptima de la red es particular para el problema a resolver.
- Para nuestra función en particular, es mejor que la cantidad de neuronas por capas esté distribuida de forma descendente.
- Momentum: Alpha y el porcentaje de patrones de entrada están directamente relacionados.
- Momentum: Funciona mejor con etas chicos.

¿Preguntas?