



Simulación de Terrenos con Redes Neuronales Multicapa

Grupo 2

Barruffaldi, Carla
Bianchi, Luciano
Cerdá, Tomás
Lynch, Marcelo



Objetivo

- Crear y entrenar una red que simula la topología de un terreno de un videojuego
- Analizar cómo influyen distintas configuraciones en el funcionamiento y en el entrenamiento

Implementación



Función de Activación

Se aplica a toda la red

Escalón

Lineal

Tangente hiperbólica

Logística



Inicialización de Pesos

xavier: $W \sim N(\mu = 0, \sigma = 2/(ni + no))$

xavier_uniform: $W \sim U(-4\sqrt{6}/\sqrt{ni + no}, 4\sqrt{6}/\sqrt{ni + no})$

normal: $W \sim N(\mu = 0, \sigma = \sqrt{ni})$

uniform: $W \sim U(-1/\sqrt{ni}, 1/\sqrt{ni})$

uniform_one: $W \sim U(-1, 1)$



Entrenamiento

- Backpropagation tradicional
- Adaptativo
- Momentum

Tamaño de batch configurable en cualquiera de ellos.



Normalización

- De la entrada para evitar saturación
- De la salida para entrar en la imagen de la función de activación

Feature scaling:

$$c = a + \frac{z - x}{y - x} * (b - a)$$

Análisis



Consideraciones

Enorme cantidad de parámetros y de combinaciones de ellos

Se puede encontrar la configuración de red más adecuada sin importar el método de entrenamiento

Prioridades:

- Generalización. Cantidad de aciertos elevado en el conjunto de prueba a pesar de un costo alto en el conjunto de entrenamiento
- Red eficiente "en producción" (= pocas neuronas)



Criterios de corte

- Error de testeo menor a 3%
- Saturación de la red
 - Costo de entrenamiento no disminuye cierto valor reiteradas iteraciones
 - ¡Cuidado con los *plateaus*!

Selección de configuración

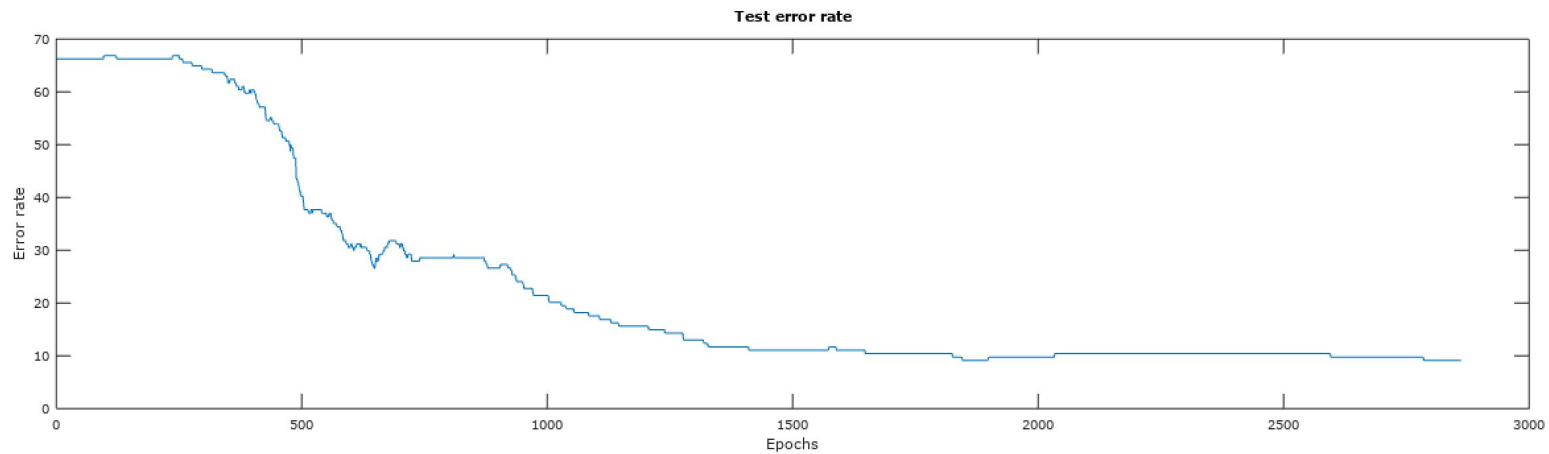
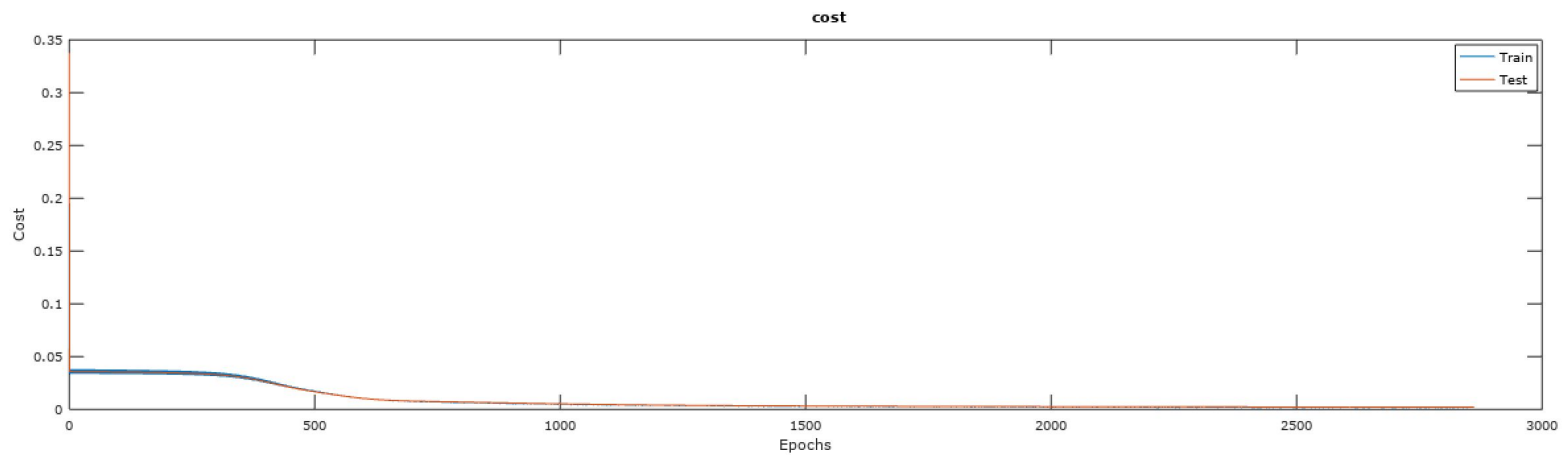
Elección de la arquitectura de la red y la función de activación



Una capa oculta

# Caso	Función	Neuronas	Error	Costo	Épocas	Tiempo (seg)
1	tanh	30	10.4%	17.519×10^{-4}	2618	338
2	logistic	30	10.4%	13.804×10^{-4}	2606	363
6	logistic	90	98%	1389.72×10^{-4}	1000	18

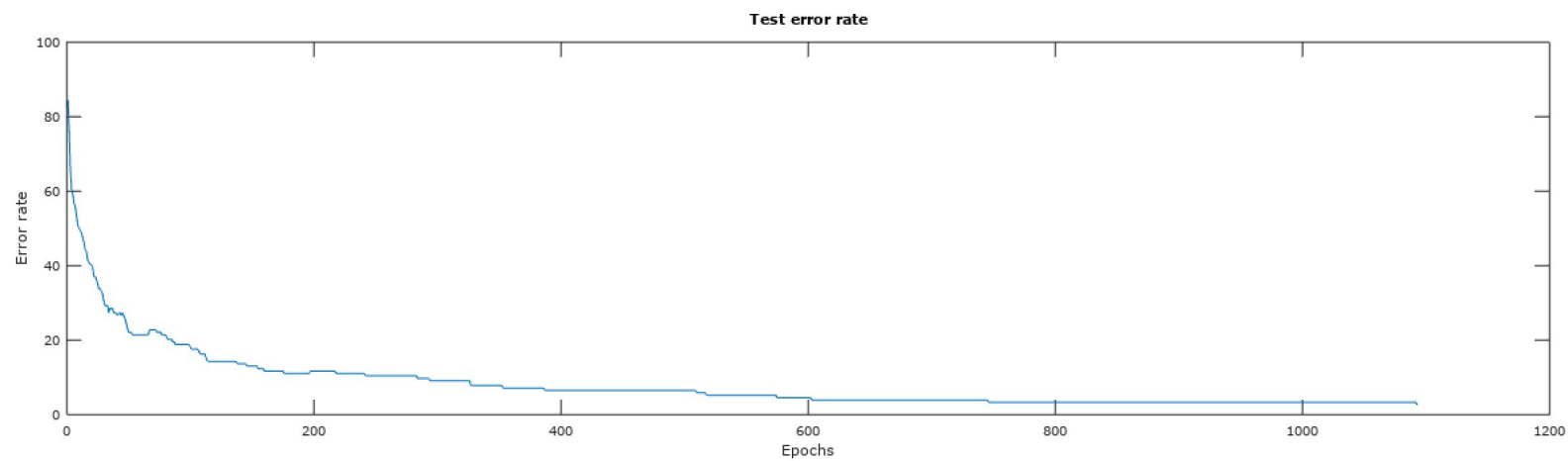
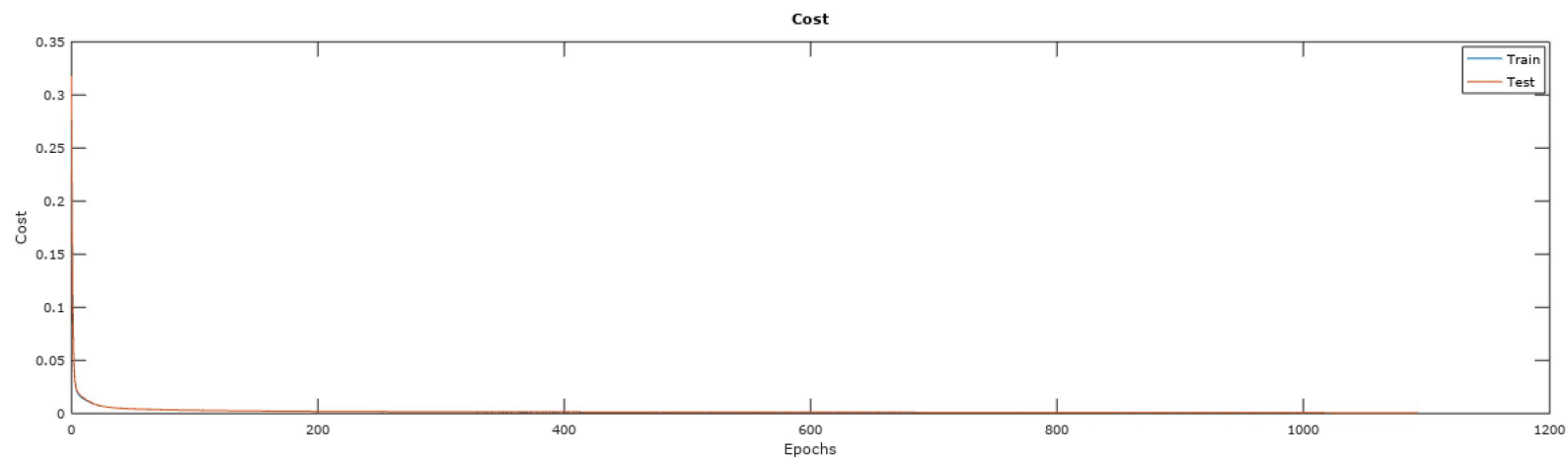
- Tendencia en tanh de mayor costo pero misma tasa de error
- Mayor generalización



Dos capas ocultas

# Caso	Función	Neuronas	Error	Costo	Épocas	Tiempo (seg)
1	tanh	[90 60]	<3%	12.198×10^{-4}	321	59
2	logistic	[90 60]	3.2%	5.2397×10^{-4}	1498	415
3	tanh	[60 30]	<3%	4.3811×10^{-4}	2492	544
5	tanh	[40 20]	<3%	6.5412×10^{-4}	1093	135

- Función logistic nunca alcanza el objetivo propuesto
- Se prioriza configuración con menor cantidad de neuronas



Entrenamiento

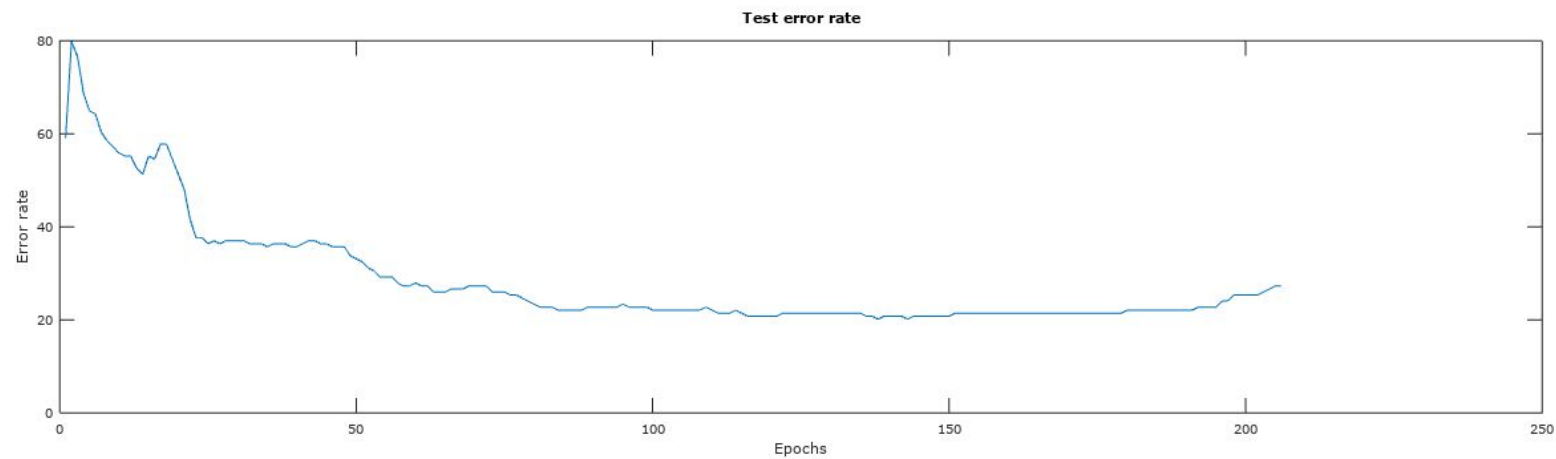
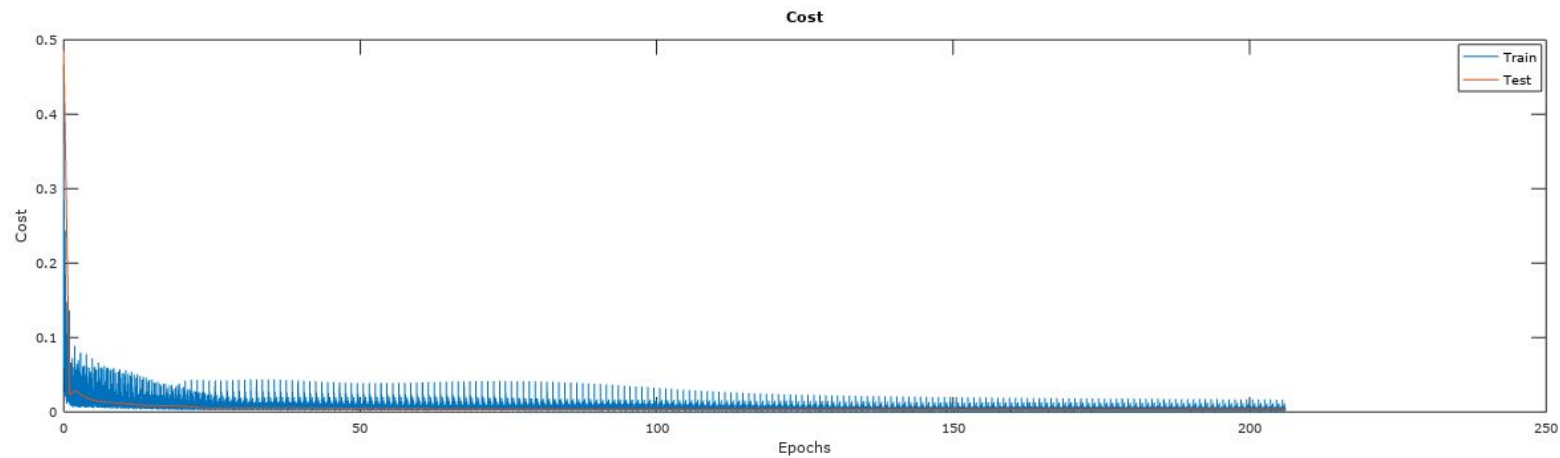
Elección del mejor método de entrenamiento junto con sus parámetros de funcionamiento



Tradicional

# Caso	learning rate	batch_size	Error	Costo	Épocas	Tiempo (seg)
1	0.1	1	27%	38.175×10^{-4}	210	212
2	0.1	5	<3%	8.1321×10^{-4}	386	71
3	0.1	<i>batch</i>	<3%	7.6641×10^{-4}	17935	135
4	0.01	1	<3%	8.3187×10^{-4}	459	450

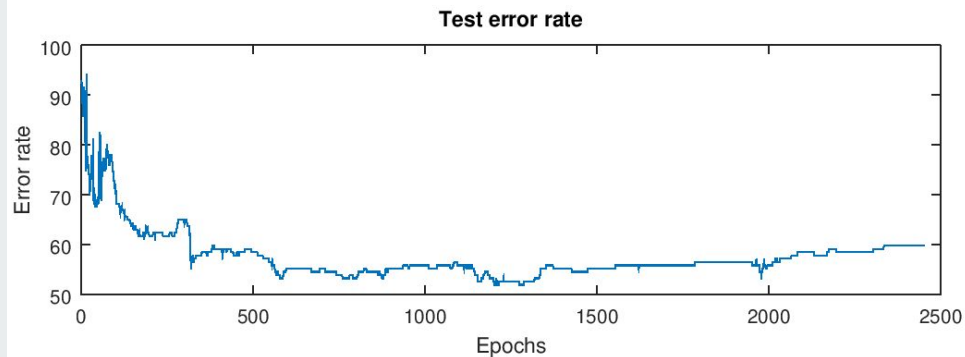
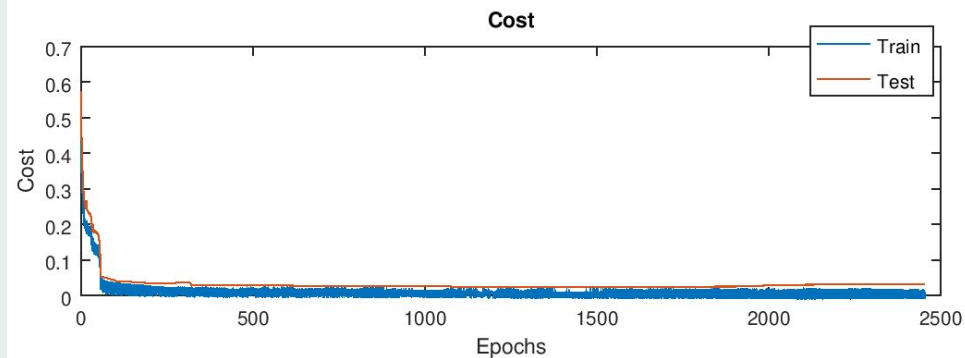
- Casi mayor generalización
- Tiempo considerablemente menor



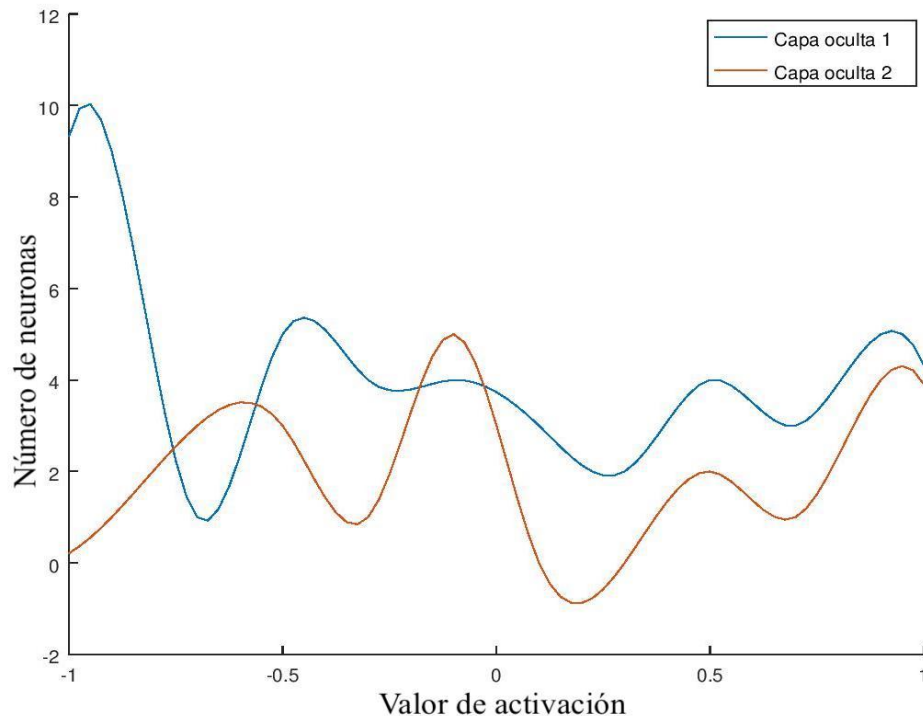
Inicialización normal

$$W \sim N(\mu = 0, \sigma = \sqrt{ni})$$

- Fracaso
- Estancamiento luego de ~800 épocas



Análisis de capas ocultas (800 épocas)



Saturación temprana

- Muchas neuronas saturadas en las capas ocultas **pero con el error (53%) y el costo aún muy altos**
- Coincide con que la red deja de aprender luego de 800 épocas
- Similar a las otras inicializaciones **salvo uniform_one: $U \sim (-1, 1)$**



Momentum

Se buscó la mejor configuración en tres principales pasos:

1. Selección del alfa
2. Selección del learning rate
3. Selección del algoritmo



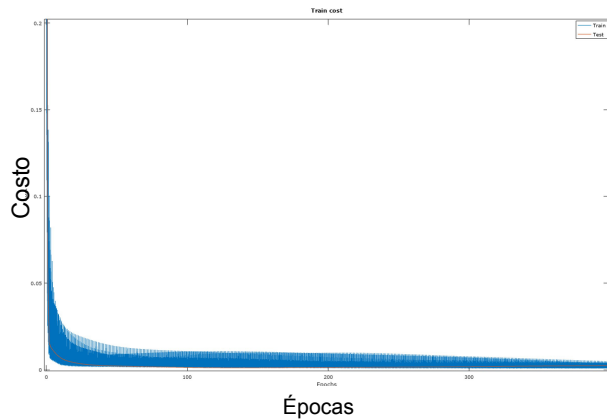
Paso 1 : Se eligió $\alpha = 0.9$

# Caso	alfa	batch_size	Error	Costo	Épocas	Tiempo (seg)
1	0.9	5	<3%	13.588×10^{-4}	362	48.6138
2	0.8	5	<3%	13.626×10^{-4}	384	49.7352
3	0.4	5	<3%	13.227×10^{-4}	521	67.4471

Paso 2: Se eligió learning rate = 0.05

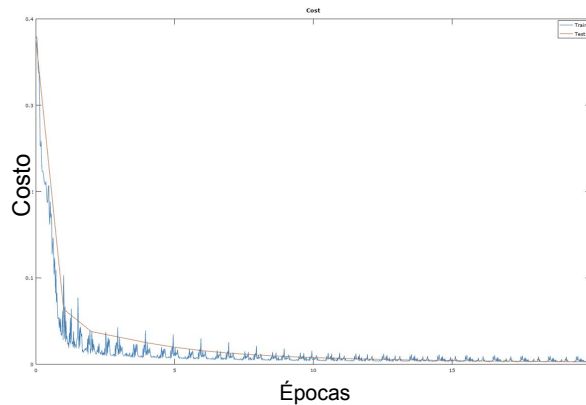
# Caso	learning rate	batch_size	Error	Costo	Épocas	Tiempo (seg)
1	0.01	5	<3%	13.588×10^{-4}	521	48
2	0.05	5	<3%	13.342×10^{-4}	93	12
3	0.09	5	<3%	8.8311×10^{-4}	689	67
4	0.1	5	3.9%	8.5387×10^{-4}	1588	324

Paso 3: El mejor algoritmo fue batch



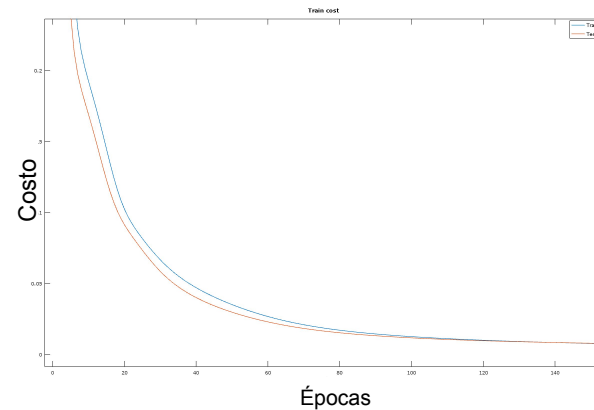
Incremental

Costo	Épocas	Tiempo (seg)
10.827×10^{-4}	625	368



Mini-batch

Costo	Épocas	Tiempo (seg)
13.342×10^{-4}	93	12



Batch

Costo	Épocas	Tiempo (seg)
10.086×10^{-4}	10436	95

Adaptive

Caso	batch_size	cost_interval	inc_steps	lr_increase	lr_decrease	Costo	Épocas	Tiempo (seg)
1	5	5	3	0.1	0.1	8.2649e-04	234	31
3	5	10	4	0.01	0.001	8.5815e-04	249	31
5	1	10	4	0.01	0.001	8.0584e-04	173	100
7	1	5	10	0.001	0.0001	12.726e-04	178	108

- Incremental
- Pequeños incrementos
- Pequeños decrementos
- Se compara el costo cada 5 iteraciones

Resultados



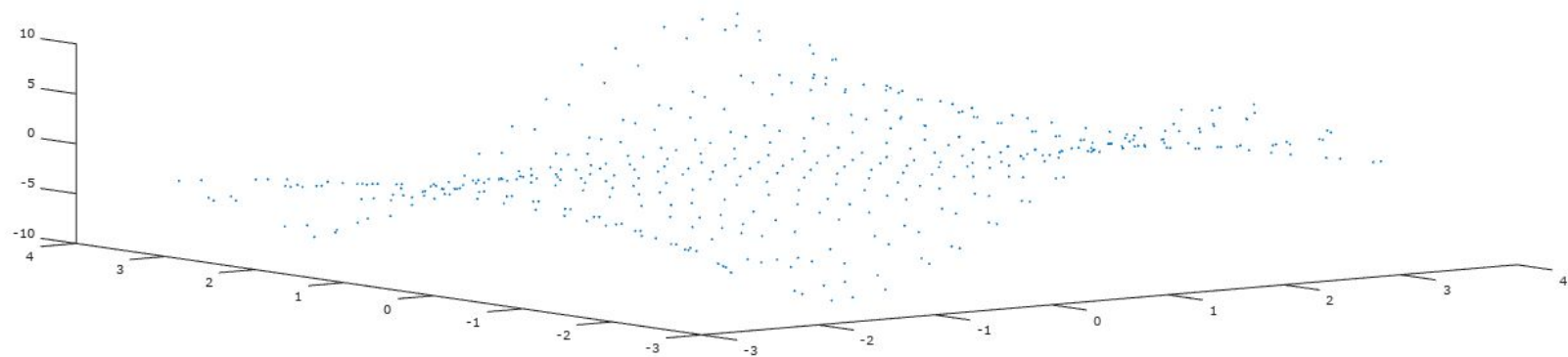
Configuración elegida

- 2 capas ocultas de 40 y 20 neuronas
- Función de activación: tangente hiperbólica

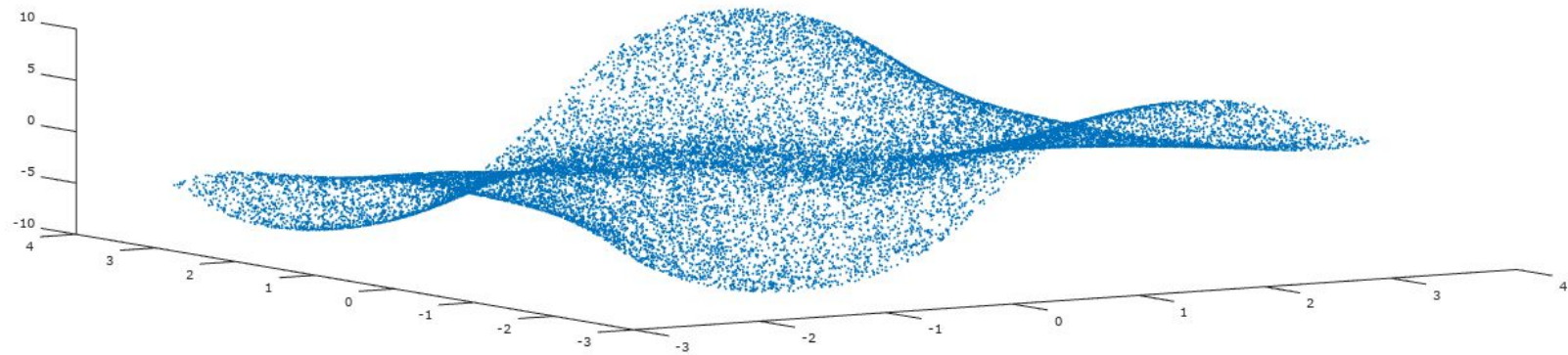
Entrenamiento

- Momentum
- `alfa: 0.9`
- `learning rate: 0.05`
- Mini-batch (size:5)

Original terrain



Neural net terrain



Conclusión



Conclusiones

- Se puede desacoplar la selección de la configuración de la red del entrenamiento sin afectar la calidad del análisis
- Las técnicas de inicialización de pesos no fueron efectivas para este problema
- La configuración de la red neuronal depende completamente del problema en cuestión
- Se debe tener en cuenta con qué criterios evaluar la correcta funcionalidad de la red en función de lo que se desea obtener

—

iGracias!