

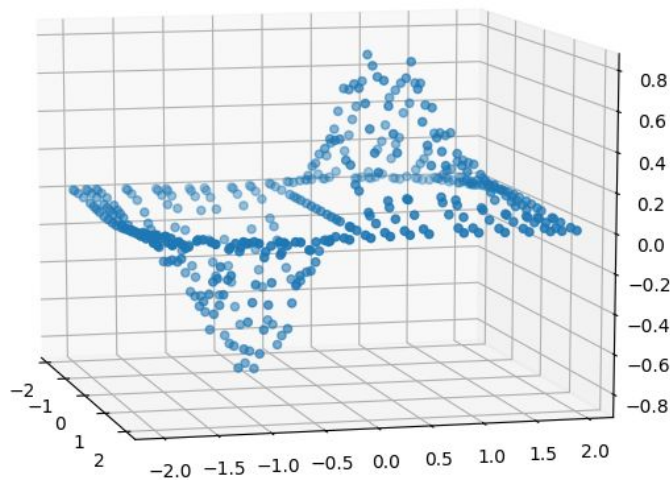


# TPE 2 - Redes Neuronales

Grupo 5

# Objetivo

Mediante una red neuronal  
aproximar la función altura  
de una simulación de terreno en base  
a la latitud y longitud.

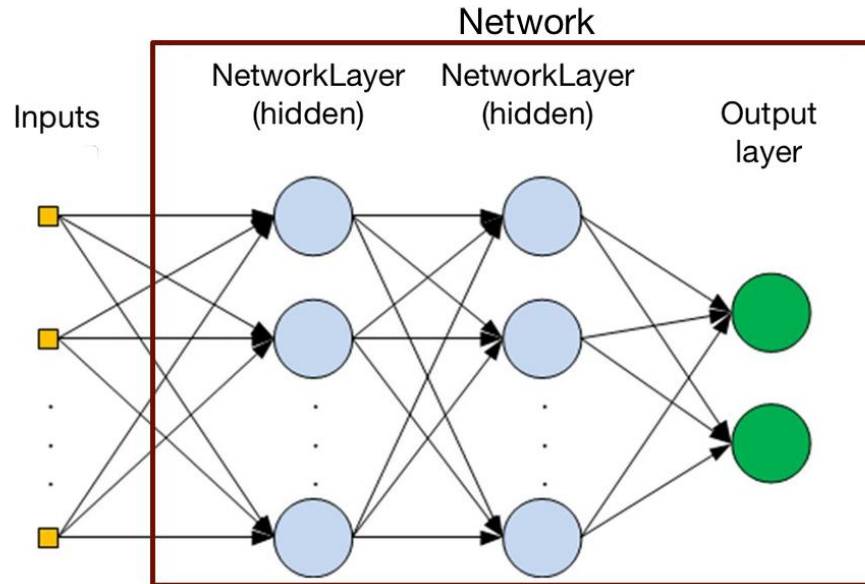


# Resolucion

Perceptrón multicapa con aprendizaje mediante backpropagation

## Modelos:

- Network
- Network Layer
- Neuron



# Selección de datos de aprendizaje

Red 2 capas, 4 neuronas usando tanh con  $\beta = \frac{2}{3}$ , 2000 epocas

n	Data entrenamiento	Data testeo	Error minimo
0.2	Primera mitad	Ultima mitad	0.00420552751 584
0.05	Ordenados por z, tomando pares	Ordenados por z, tomando impares	0.04759452548 23
0.05	Últimos 20 valores, como venían	Primeros 421 valores, como venían	0.02787457791 1

# Backpropagation - Variaciones

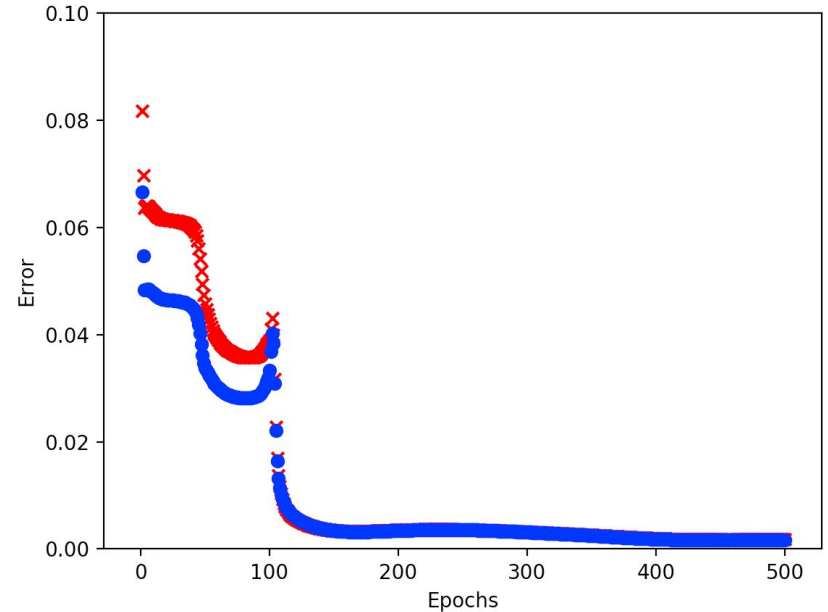
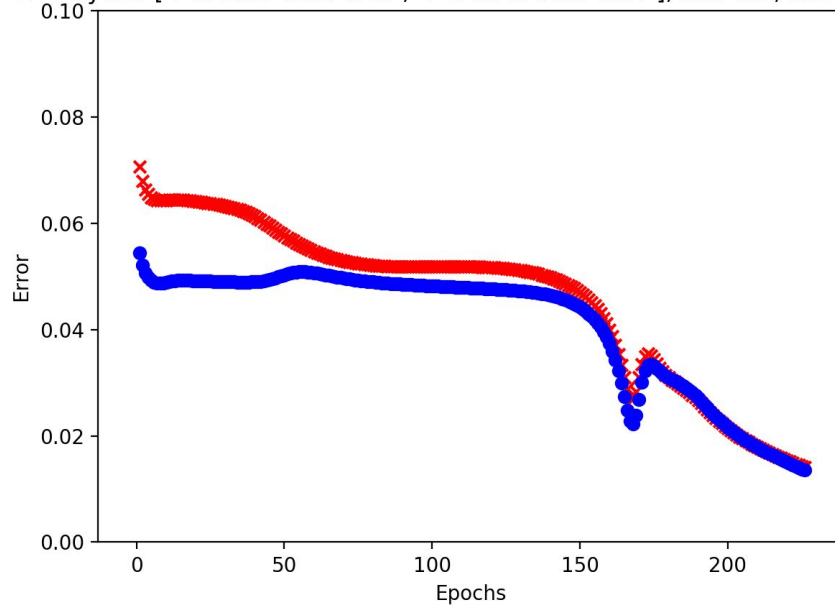
- Momentum

- Parametros Adaptativos

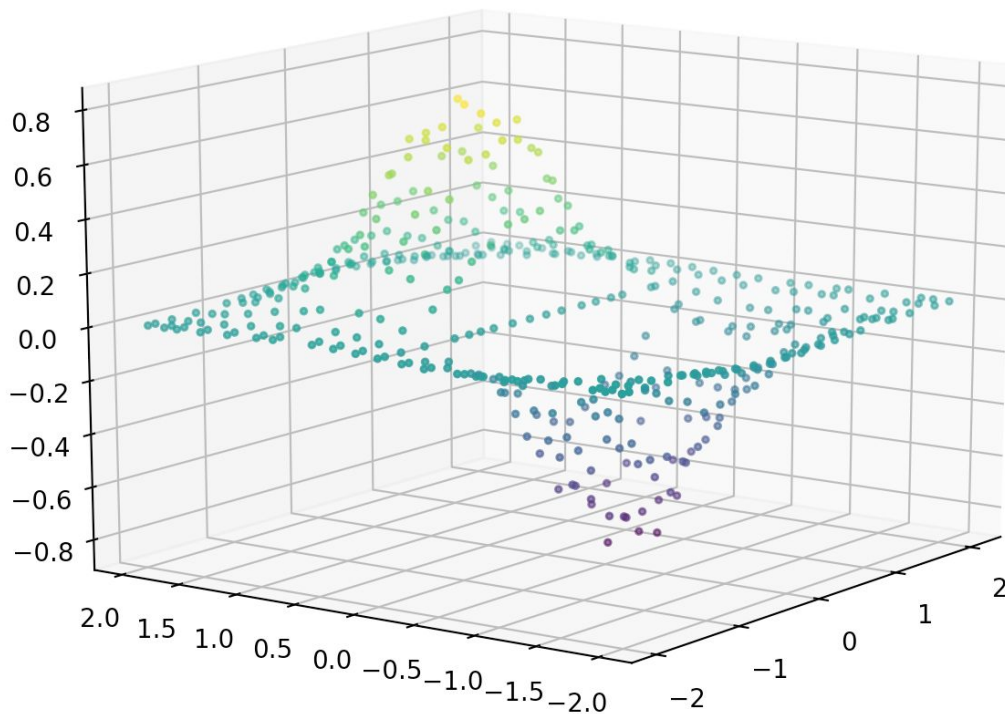
- Adaptative Annealing

# Backpropagation - Error

2 HLayers: ['7 N tanh beta 0.67', '7 N tanh beta 0.67'], eta: 0.1, err: 0.0001

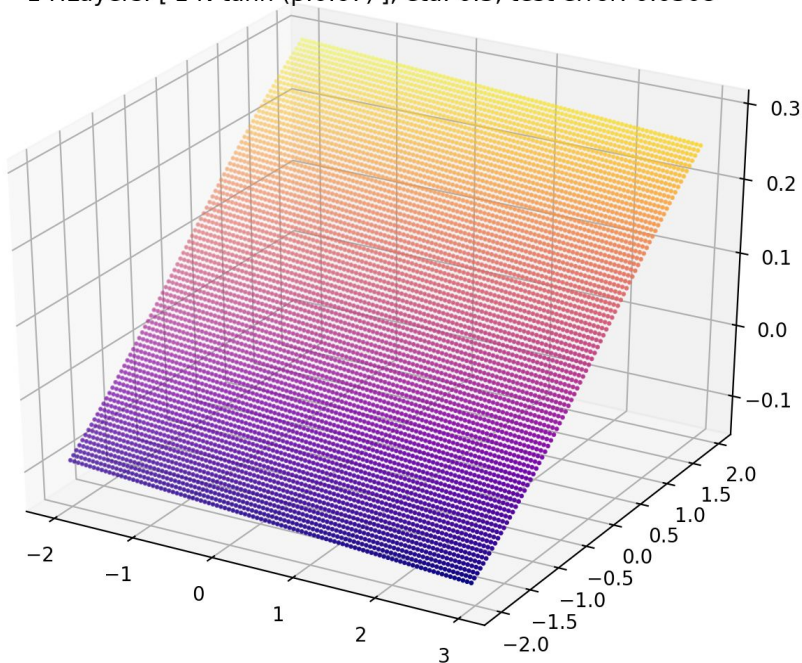


# Selección de arquitecturas - ¿qué buscamos?

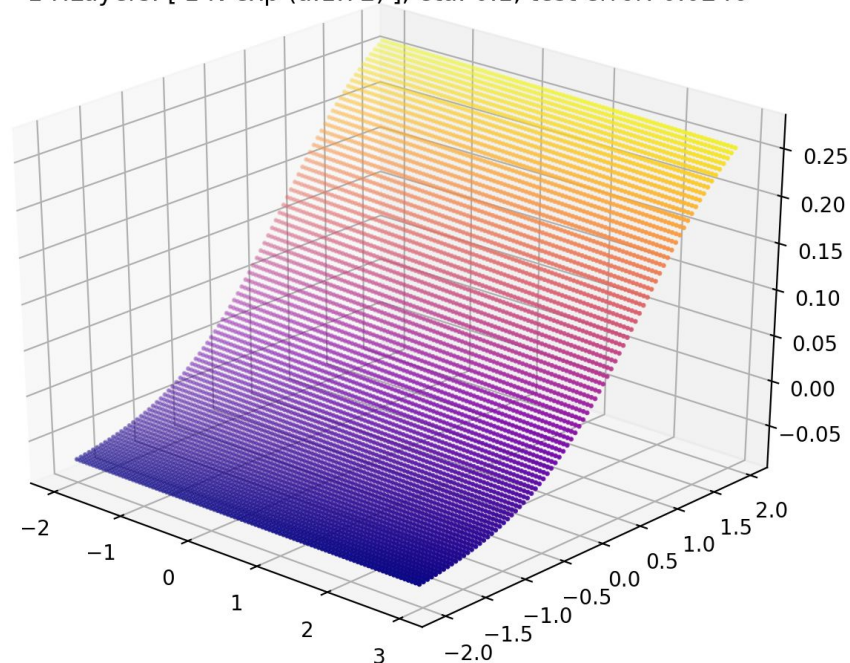


# Prueba de arquitecturas - 1 capa con 1 neurona

1 HLayers: ['1 N tanh ( $\beta$ :0.67)'], eta: 0.3, test error: 0.0308



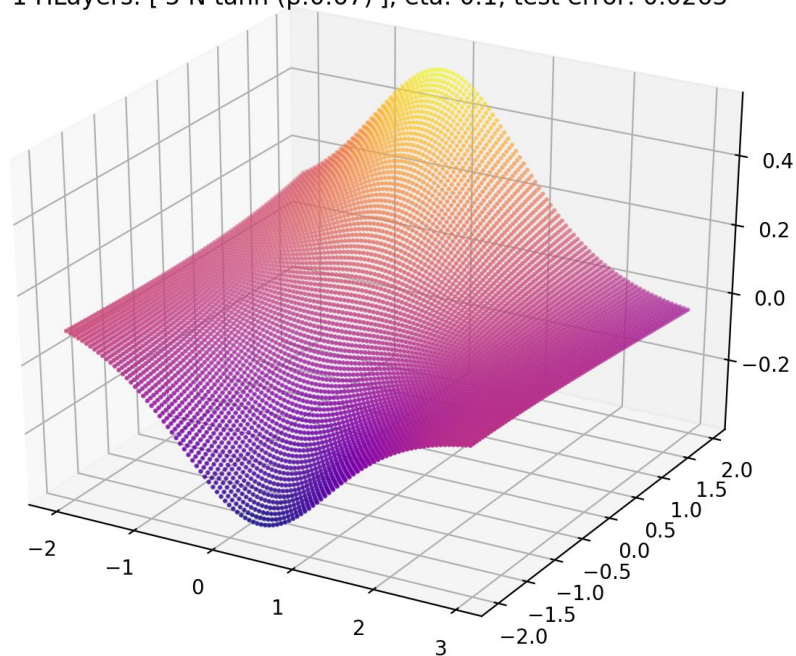
1 HLayers: ['1 N exp ( $a$ :1.72)'], eta: 0.1, test error: 0.0246



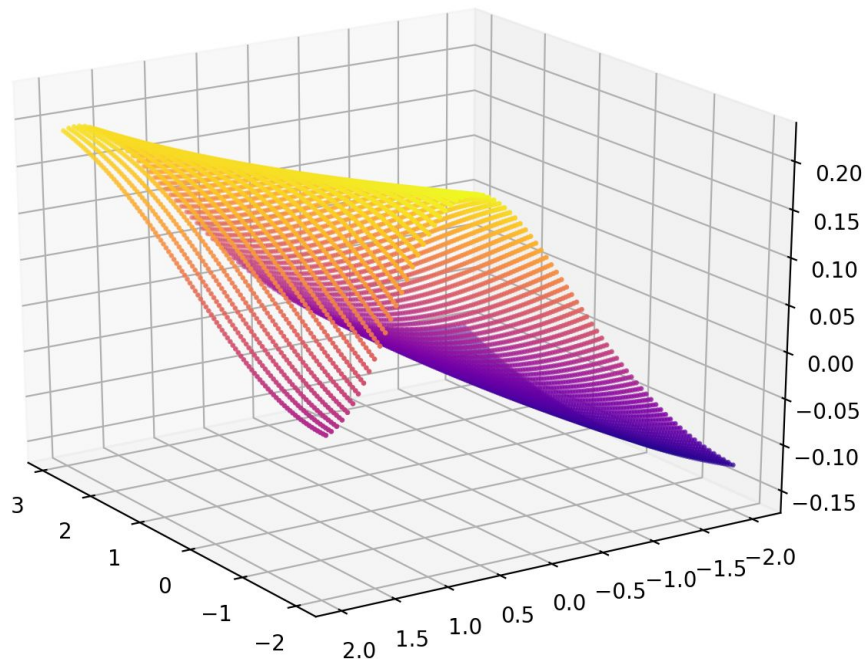


# Prueba de arquitecturas - 1 capa con 5 neuronas

1 HLayers: ['5 N tanh ( $\beta$ :0.67)'], eta: 0.1, test error: 0.0263

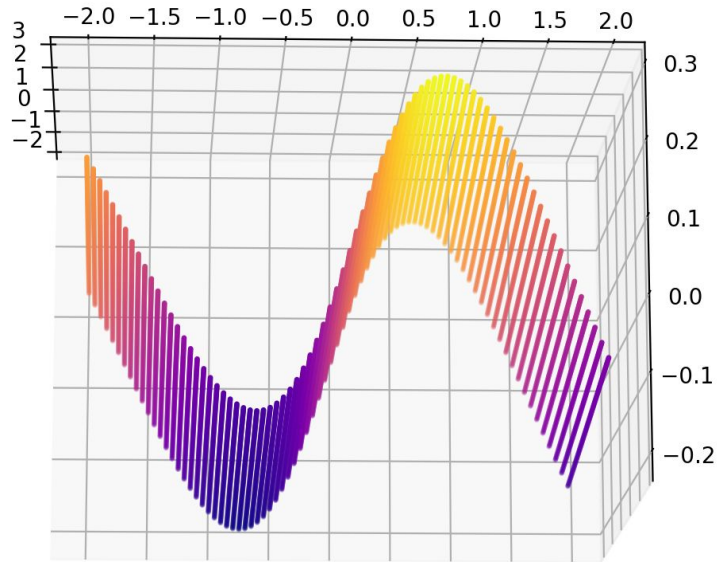


1 HLayers: ['5 N exp (a:1.72)'], eta: 0.1, test error: 0.0205

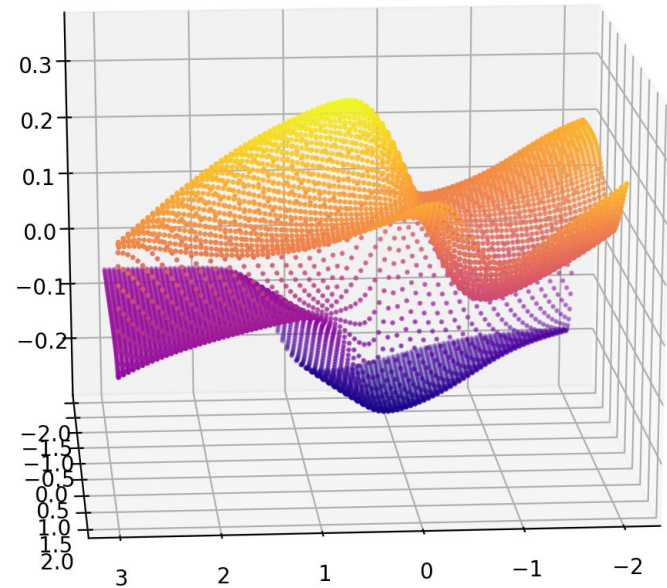


# Prueba de arquitecturas - 1 capa con 6 neuronas

1 HLayers: ['6 N tanh ( $\beta$ :0.67)'], eta: 0.1, test error: 0.0161

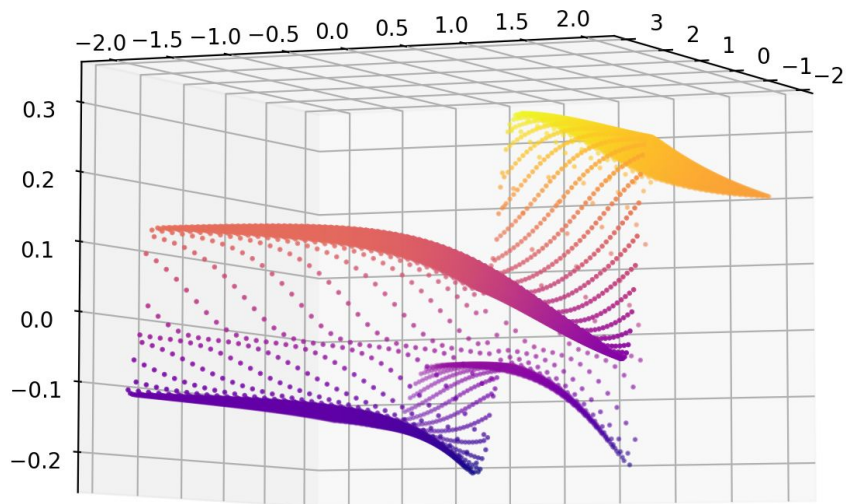


1 HLayers: ['6 N exp (a:1.72)'], eta: 0.1, test error: 0.0204

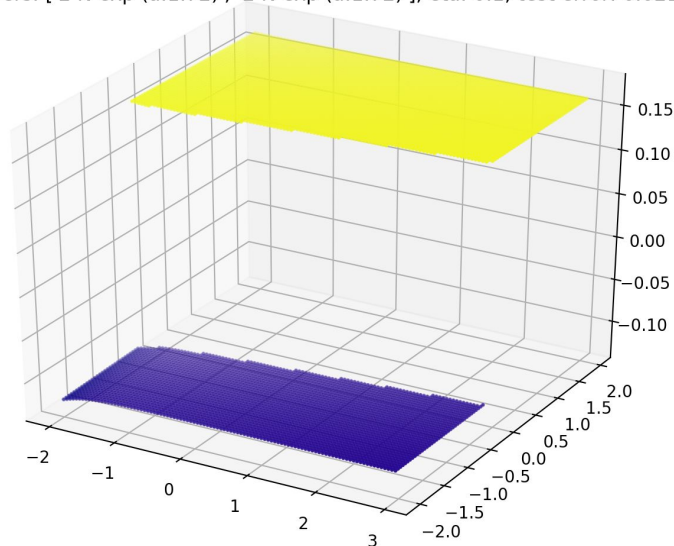


# Prueba de arquitecturas - 2 capas, 2 y 3 neuronas

2 HLayers: ['3 N exp (a:1.72)', '3 N exp (a:1.72)'], eta: 0.1, test error: 0.019

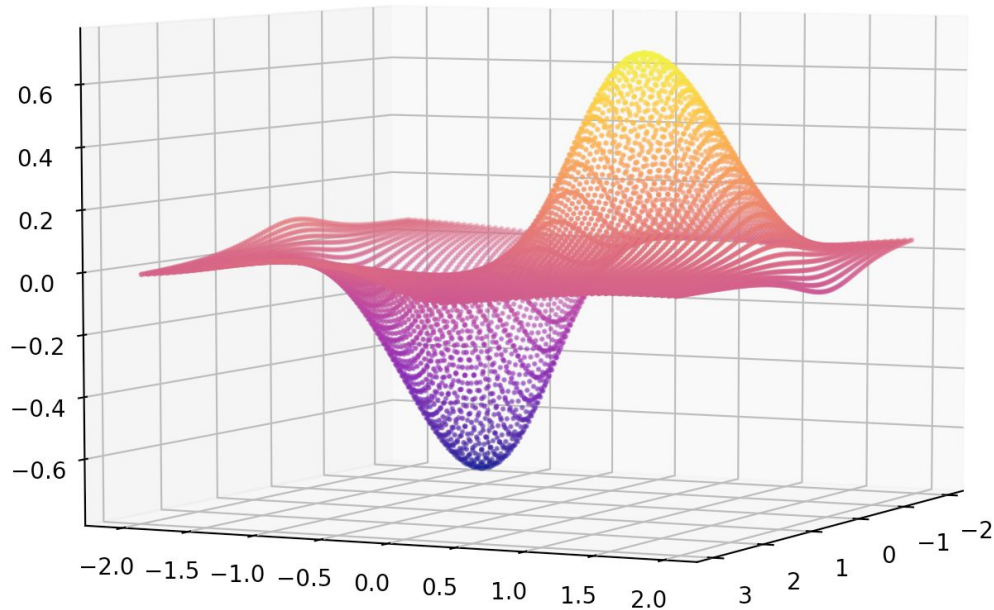


2 HLayers: ['2 N exp (a:1.72)', '2 N exp (a:1.72)'], eta: 0.1, test error: 0.0213



# Prueba de arquitecturas - 2 capas con 4 neuronas

2 HLayers: ['4 N tanh ( $\beta$ :0.67)', '4 N tanh ( $\beta$ :0.67)'], eta: 0.2, test error: 0.0013

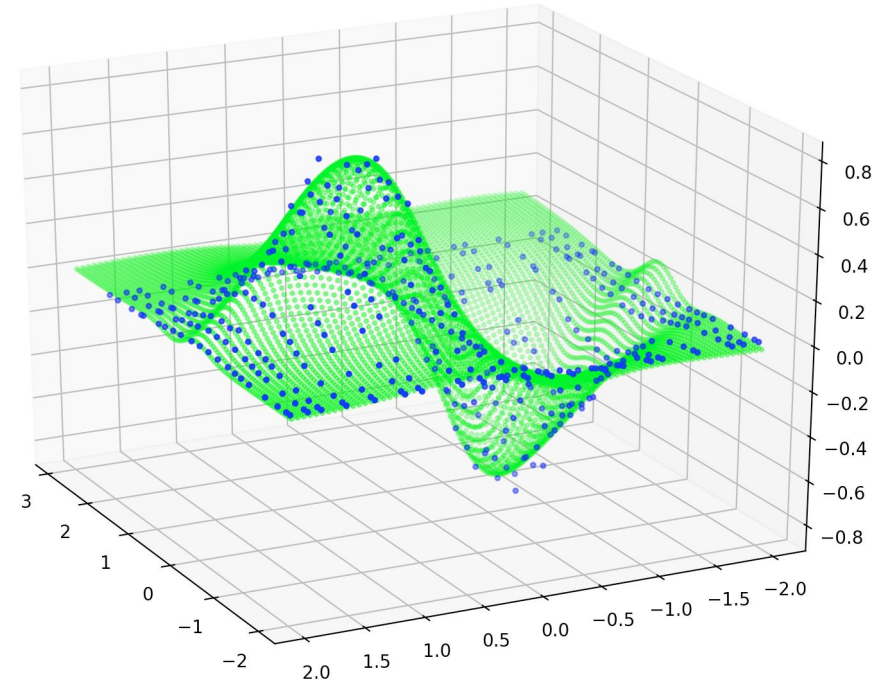
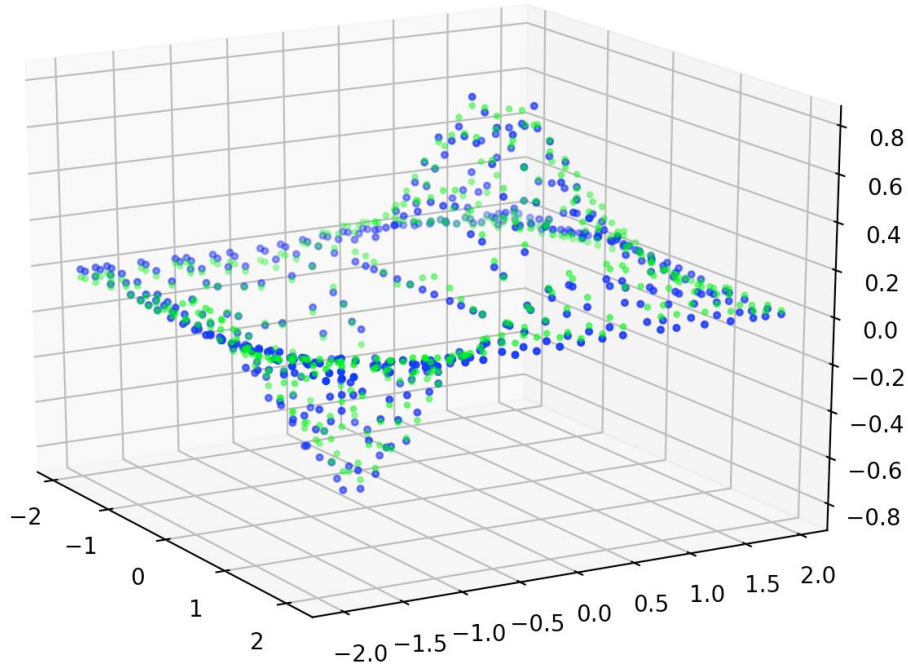


# Prueba de arquitecturas - Métricas

Arquitectura	Tiempo de entrenamiento promedio de una época [s]	Tiempo de predicción promedio [s]
1 capa 1 neurona c/ tanh	0.0097	1.4e-05
1 capa 1 neurona c/ exp	0.012	1.7e-05
1 capa 5 neurona c/ tanh	0.0277	4.2e-05
2 capas 4 neuronas c/ tanh	0.0448	7.2e-05

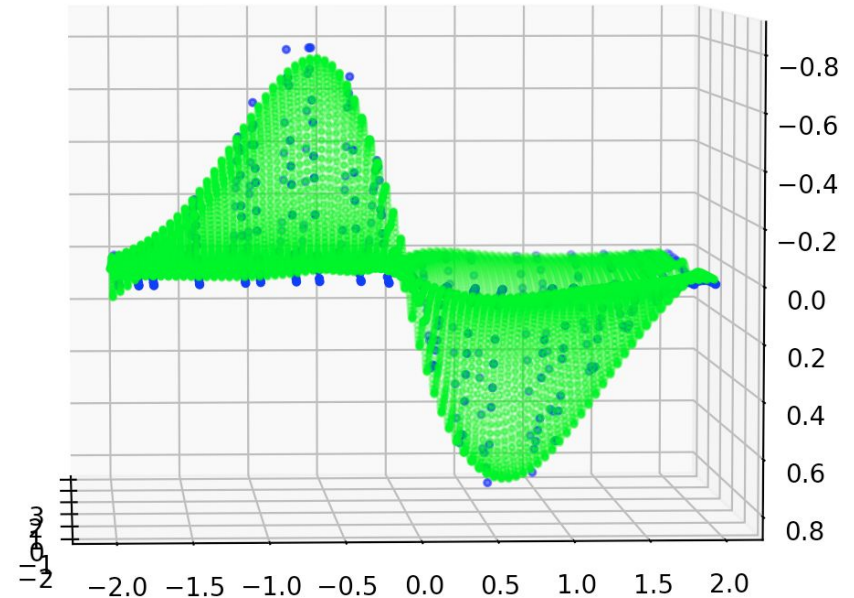
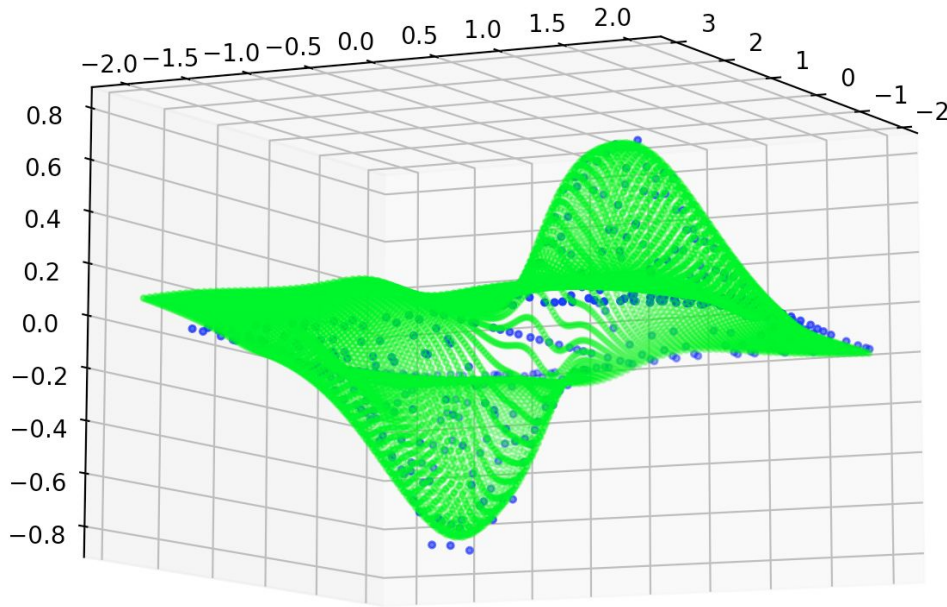
\* Pruebas realizadas para 15000 épocas y 15000 predicciones

# Comparando arquitectura con requerimiento





# Comparando arquitectura con requerimiento



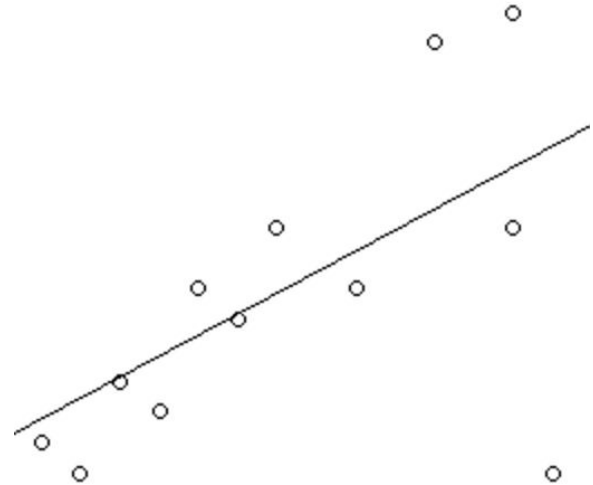
# Backpropagation - Parametros adaptativos

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

$$a = 0.01$$

$$b = 0.1$$

$$m = 1 \times 10^{-7}$$



$$m = \frac{y_2 - y_1}{x_2 - x_1}$$



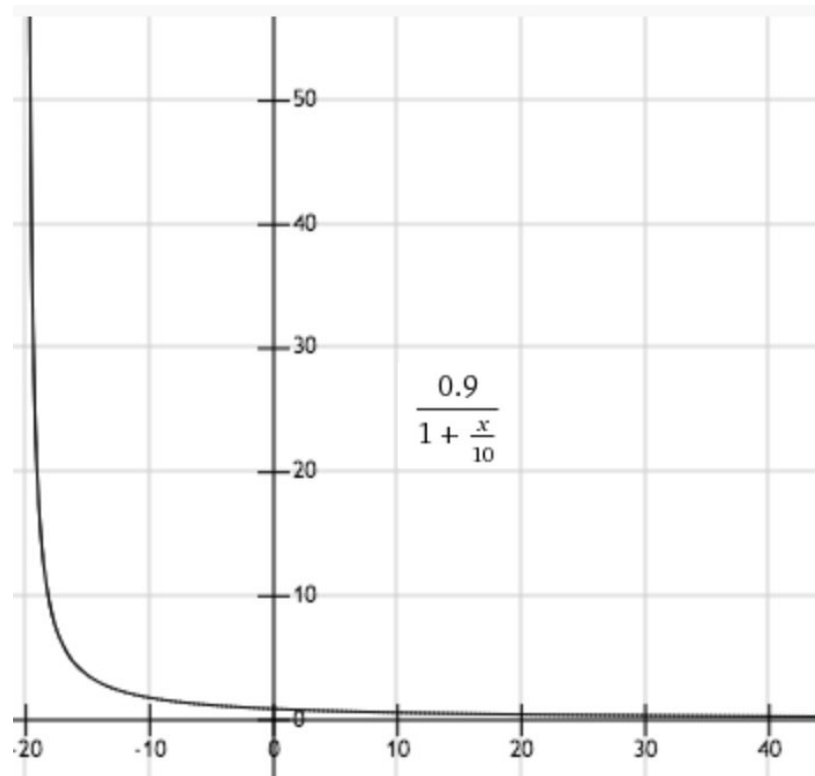
# Backpropagation - Adaptive annealing

- Tambien llamado temperatura
- El n se reduce a medida que avanzan las épocas
- $f(x) = n_0 / (1 + x/k)$

$n_0$ : eta original

$x$ : epoca actual

$k$ : parámetro



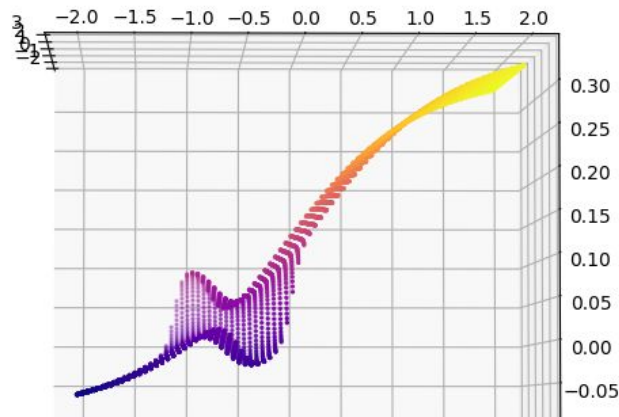
# Variaciones a Backpropagation - Error 2000 Épocas

Red: 2 capas con 4 neuronas usando tanh con  $\beta = \frac{2}{3}$

Variaciones	Test error
Standar	0.03434302922260327
Parametros Adaptativos	0.01929954857965098
Annealing	0.014793758422414646
Momentum + Parametros Adaptativos	0.004810595836781496
Momentum + Annealing	0.015952936858455963

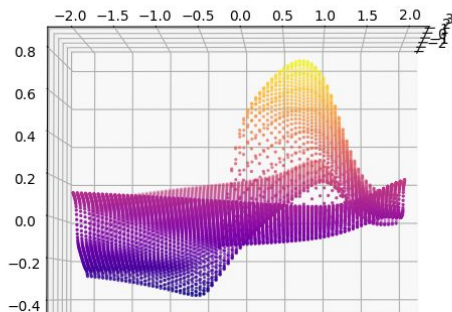
# Variaciones a Backpropagation - 2000 Épocas

Sin ninguna variación

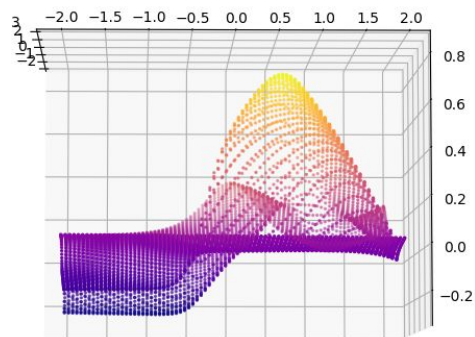


# Variaciones a Backpropagation - 2000 Épocas

Annealing

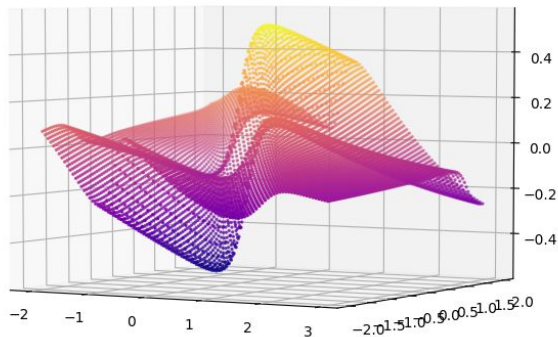


Parametros Adaptativos

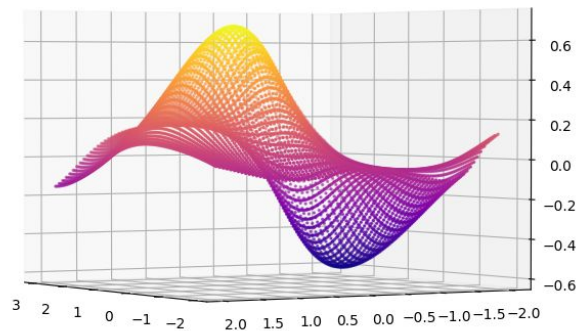


# Variaciones a Backpropagation - 2000 Épocas

Annealing + Momentum



Param. Adapt. + Momentum



# Conclusiones

- La red logra aprender con al menos un error de orden  $10^{-4}$
- La aleatoriedad en los datos de aprendizaje ayuda al aprendizaje
- Para el problema eran necesarias 2 capas ocultas
- El aprendizaje necesita mucha atención y alternar variantes para que sea efectivo