

# Sistemas de Inteligencia Artificial

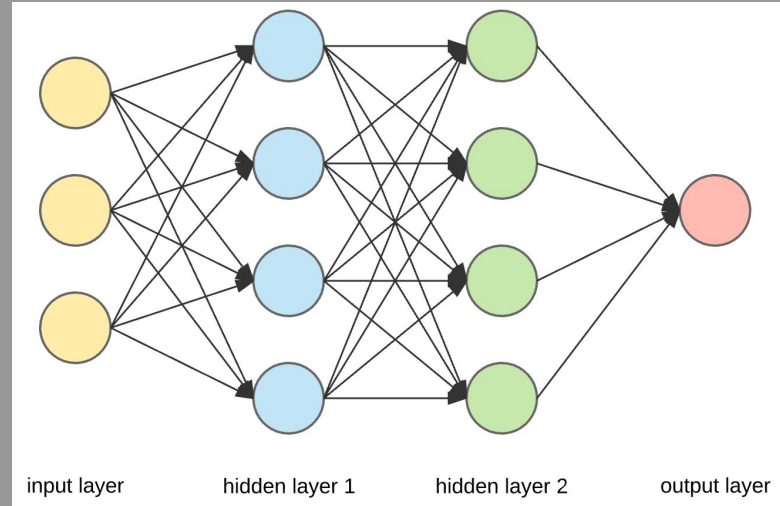
**Redes Neuronales**  
Grupo 5

# Objetivo

— — —

Resolver el problema de **aproximar un terreno** a partir de mediciones de altura, latitud y longitud utilizando **redes neuronales multicapa** y entrenandolas bajo el modelo de **aprendizaje supervisado**.

→ Antes  
→ Después  
→ Mejoras



# Análisis

Parámetros y decisiones

1. Implementación
  - a. Arquitecturas
  - b. Pesos iniciales
  - c. Funciones de Activación
2. Aprendizaje
  - a. Optimización

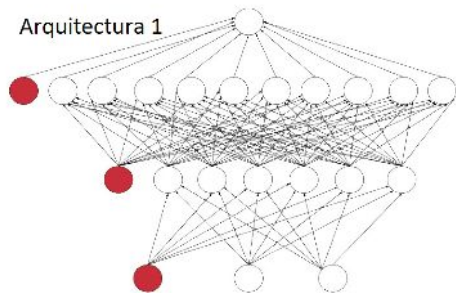
— — —

# Implementación

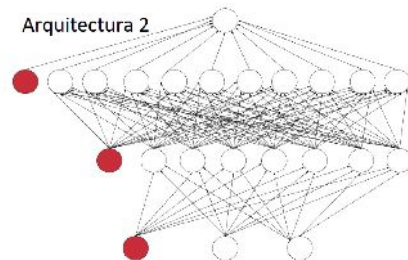
# Arquitecturas

— — —

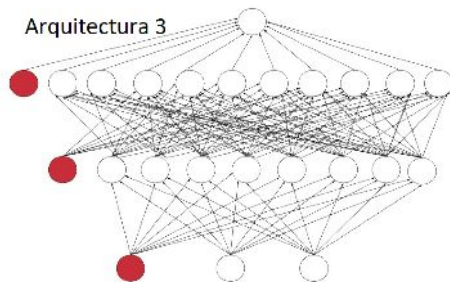
Arquitectura 1



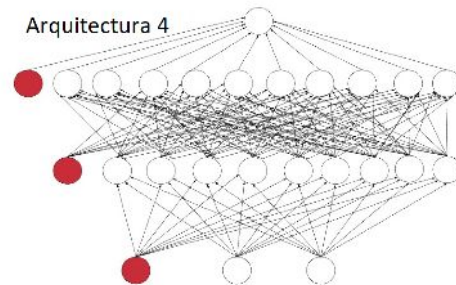
Arquitectura 2



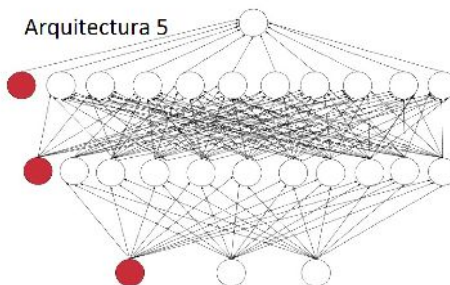
Arquitectura 3



Arquitectura 4



Arquitectura 5



# Pesos iniciales - Inicialización

---

## Aleatorio

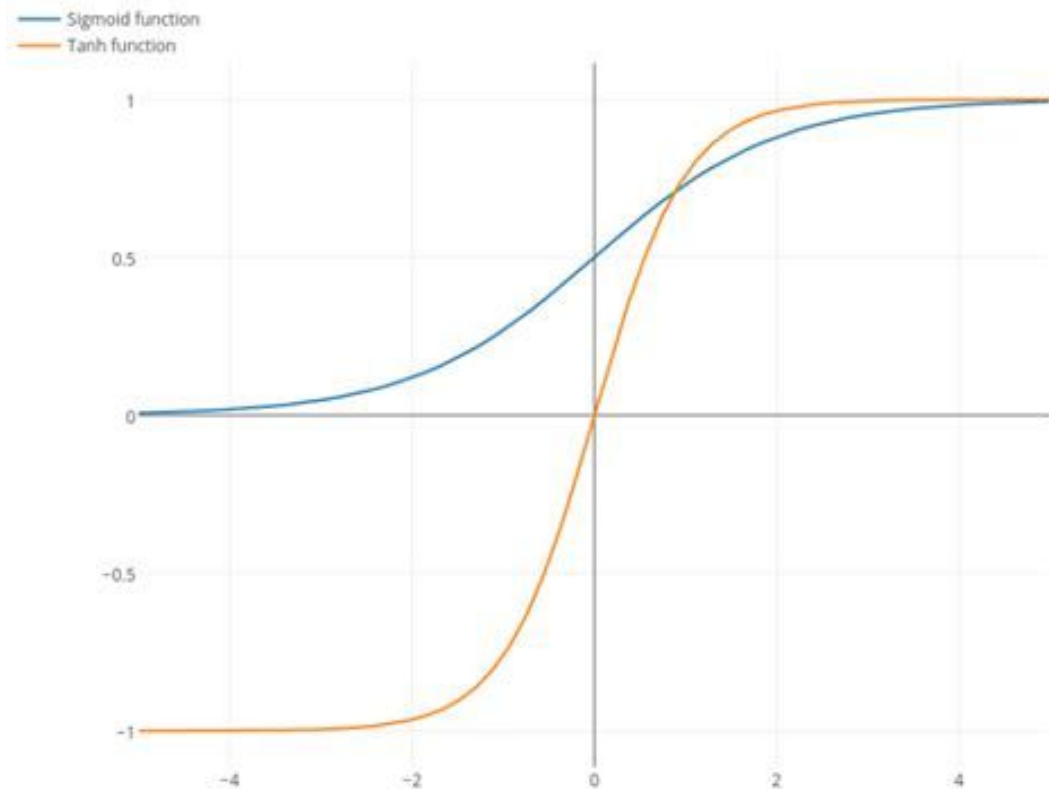
- Rompe con simetría
- Mayor precisión
- Cerca de cero, al azar

## He-at-el

- Inicialización del peso dependiendo de la capa anterior
- Alcanza un mínimo global mas rápido y eficientemente

# Funciones de activación - Sigmoidea y Tangente hiperbólica

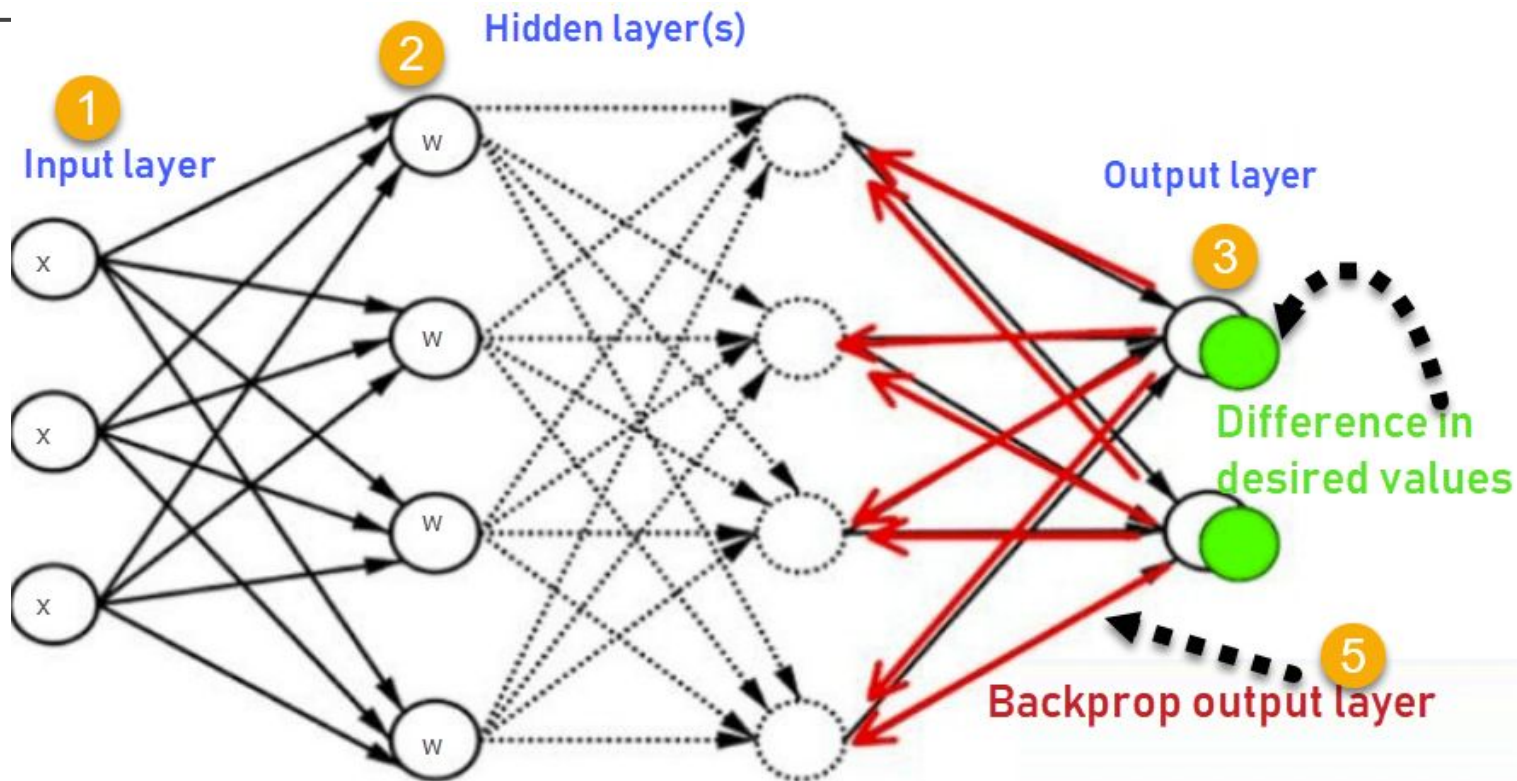
— — —

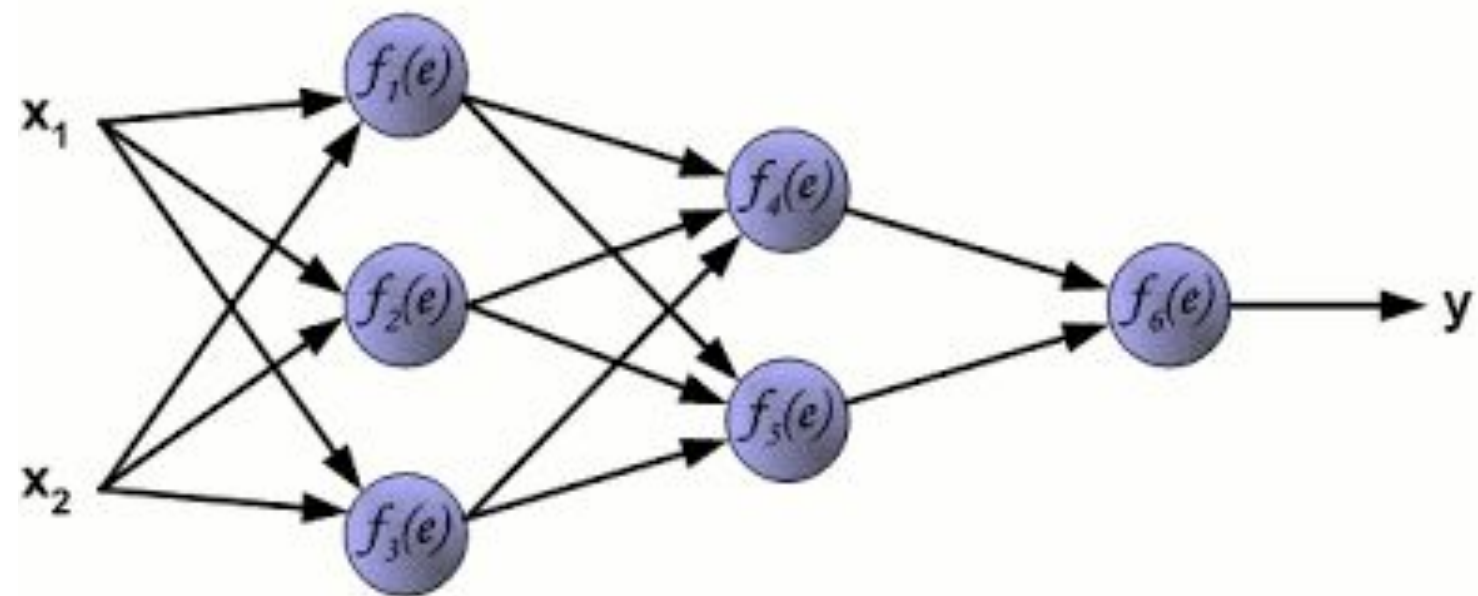




# Aprendizaje

# Backpropagation





# Backpropagation - Optimización

— — —

## Eta Adaptativo y Momentum



$W\{i\} += \text{delta\_}W\{i\} + \text{momentum} * \text{last\_delta\_}W\{i\}$

# Conclusiones

Comportamiento y  
resultados

1. Efecto del learning rate
2. Implicancias de las funciones de activación
  - a. Sigmoides vs. Tangente hiperbólica
3. Incremental vs. Batch
4. Eta vs Eta Adaptativo
5. Momentum vs !Momentum

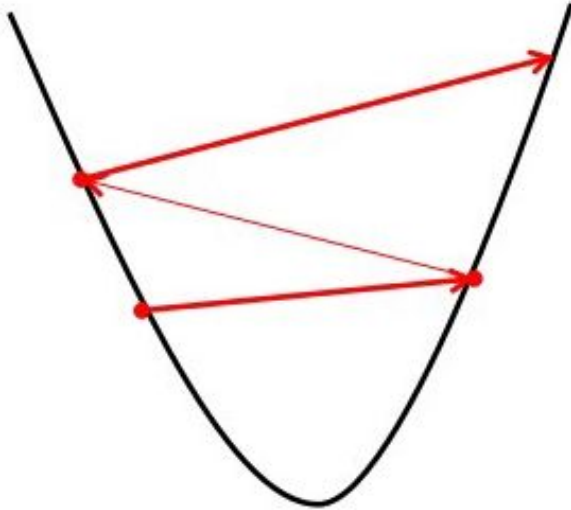
— — —

# Comportamiento

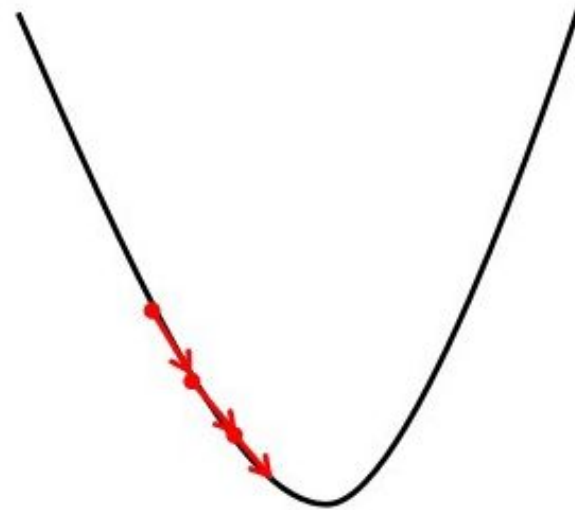
# Efecto del Learning Rate

---

Big learning rate

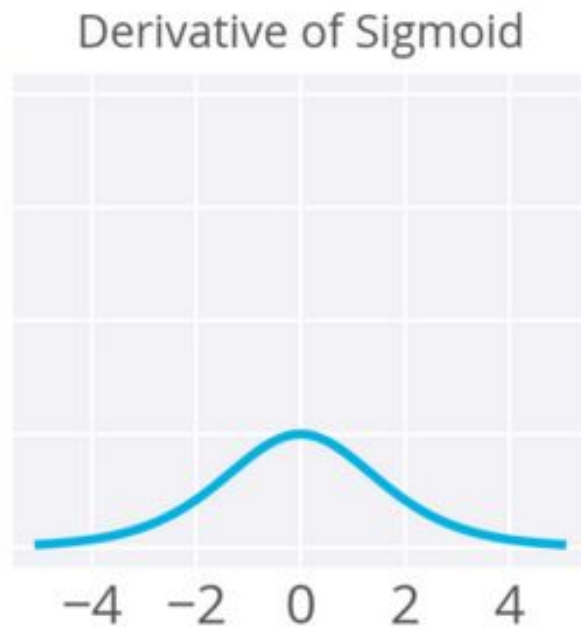
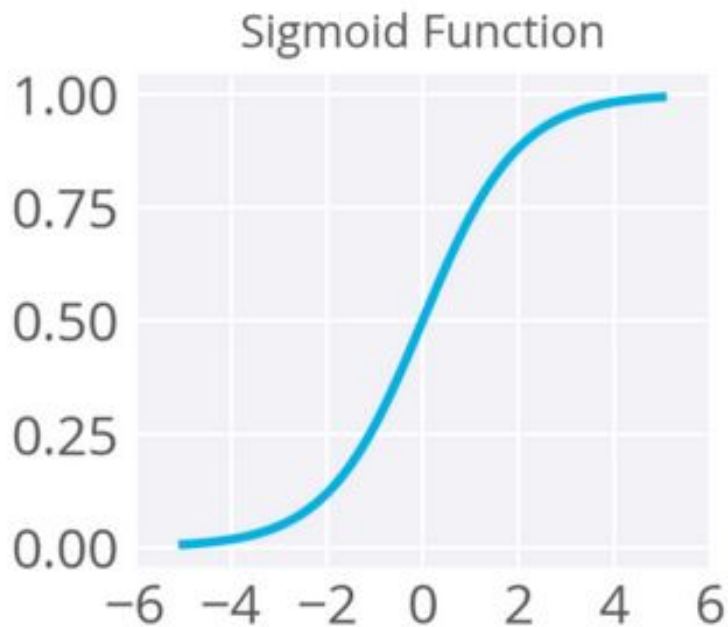


Small learning rate



# Implicancias de las funciones de activación y sus derivadas

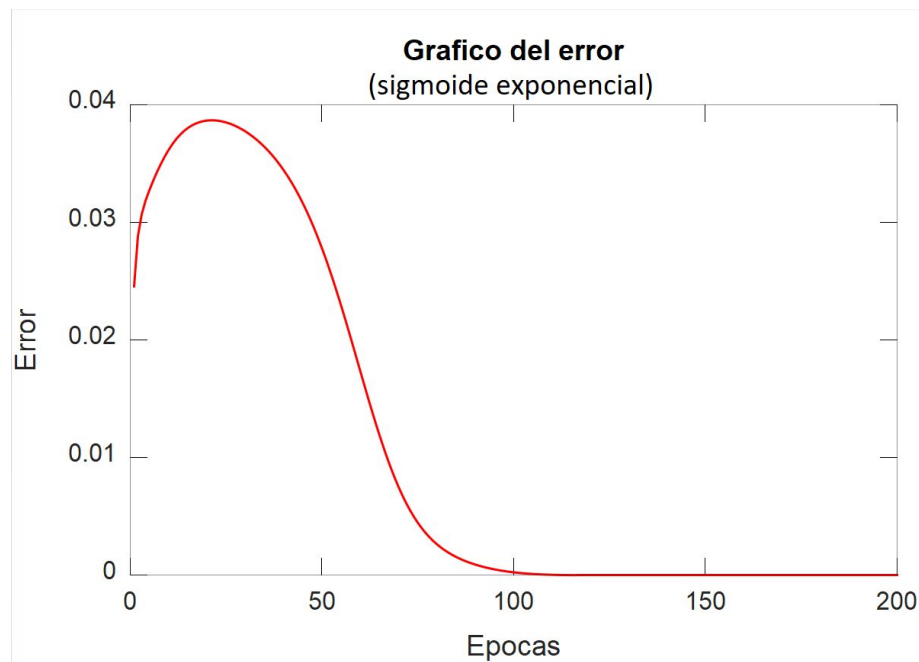
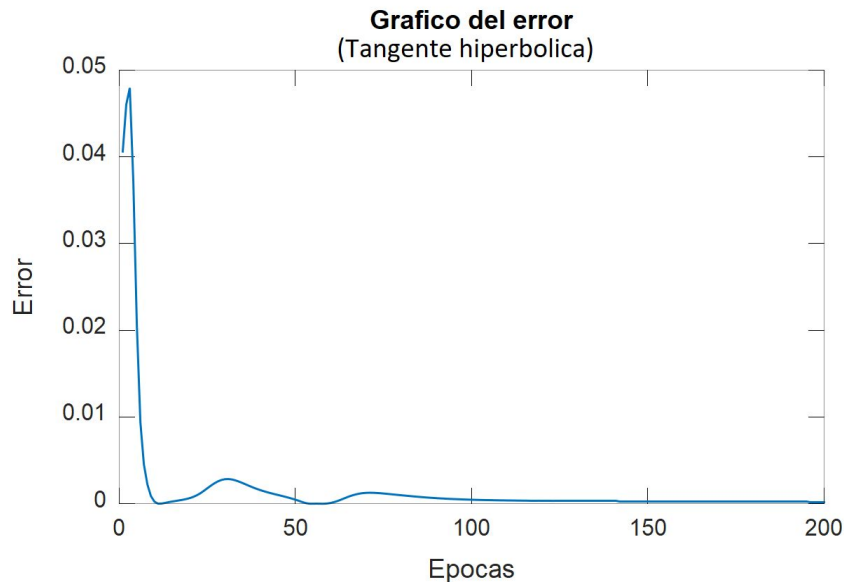
---





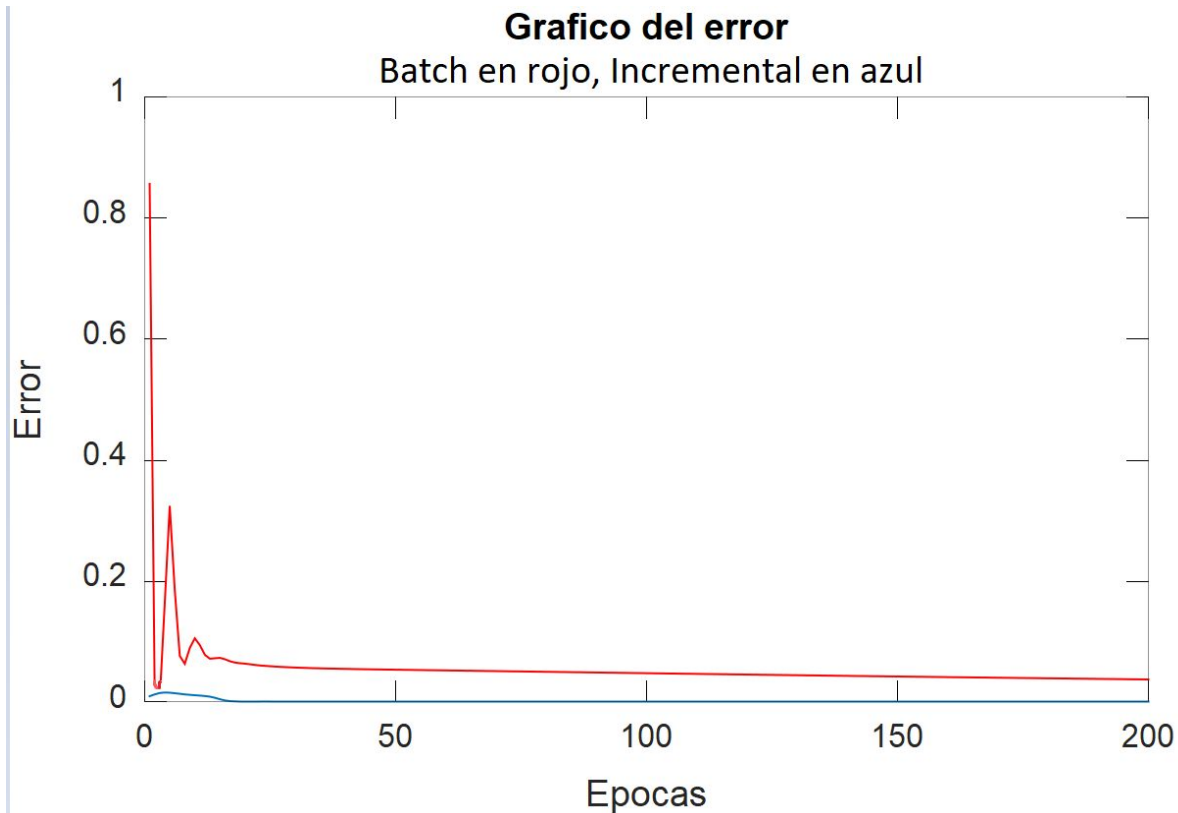
# Resultados

# Sigmoidea vs. Tangente hiperbólica



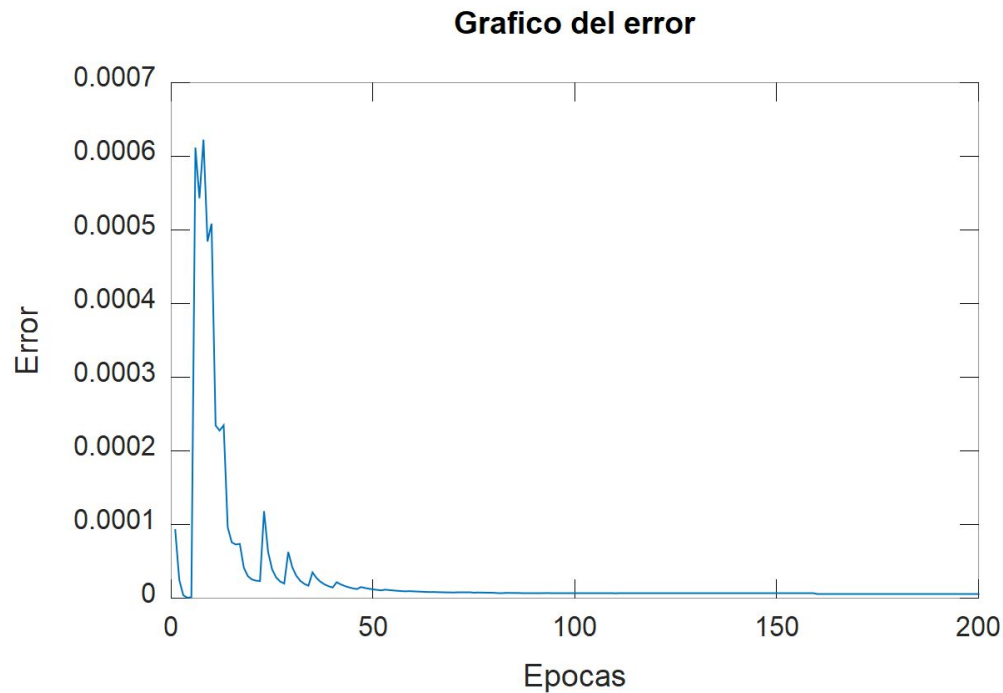
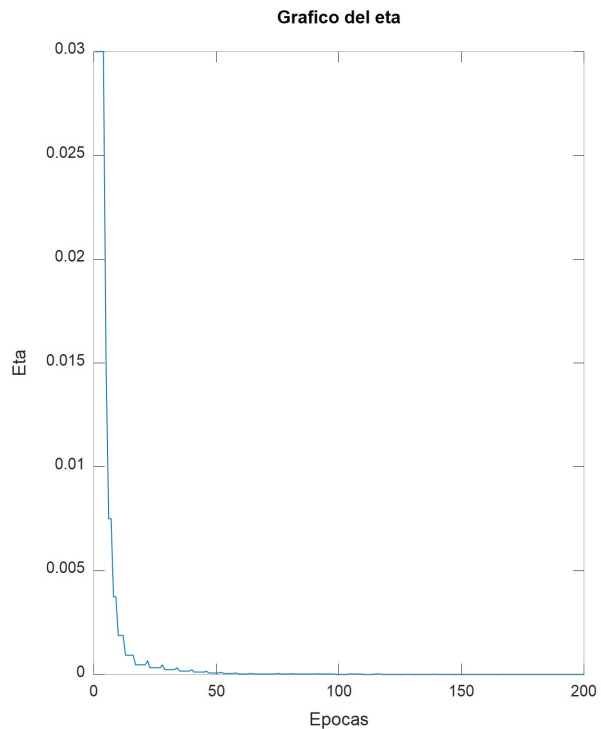
# Incremental vs. Batch

— — —



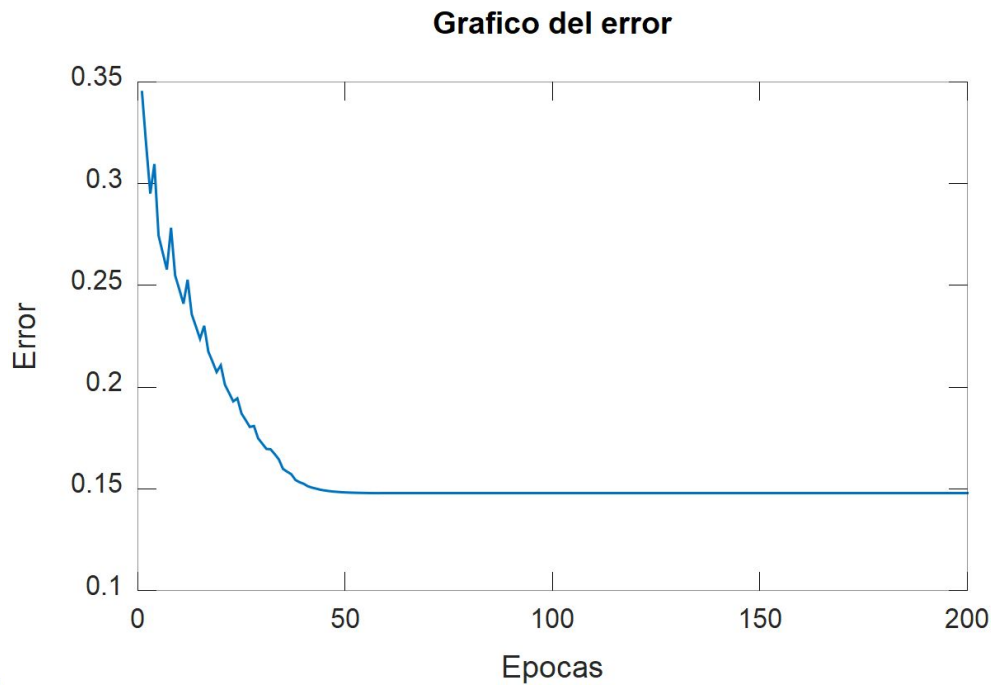
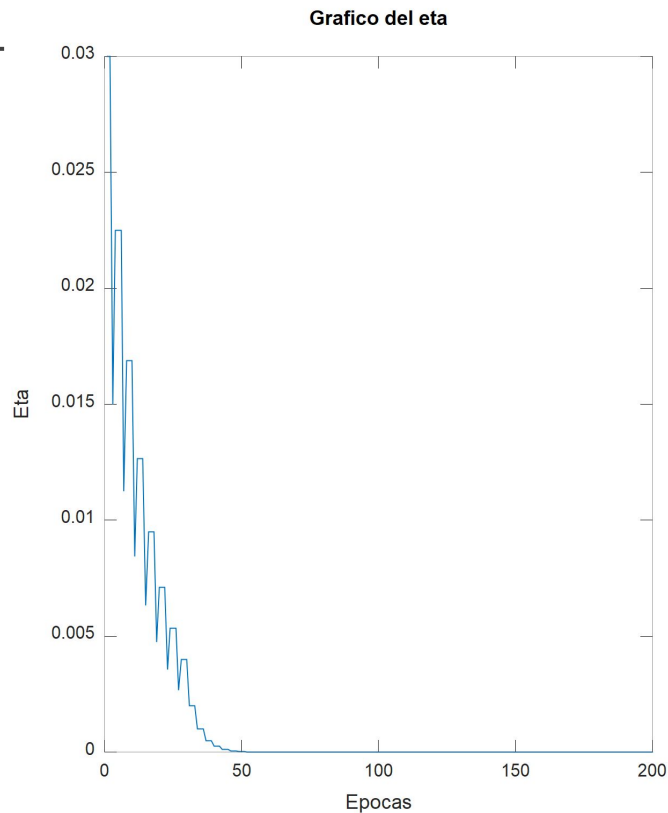
# Eta vs Eta adaptativo

Eta:  $\alpha=1.4$ ,  $\beta=0.5$ , Tercer parámetro=5

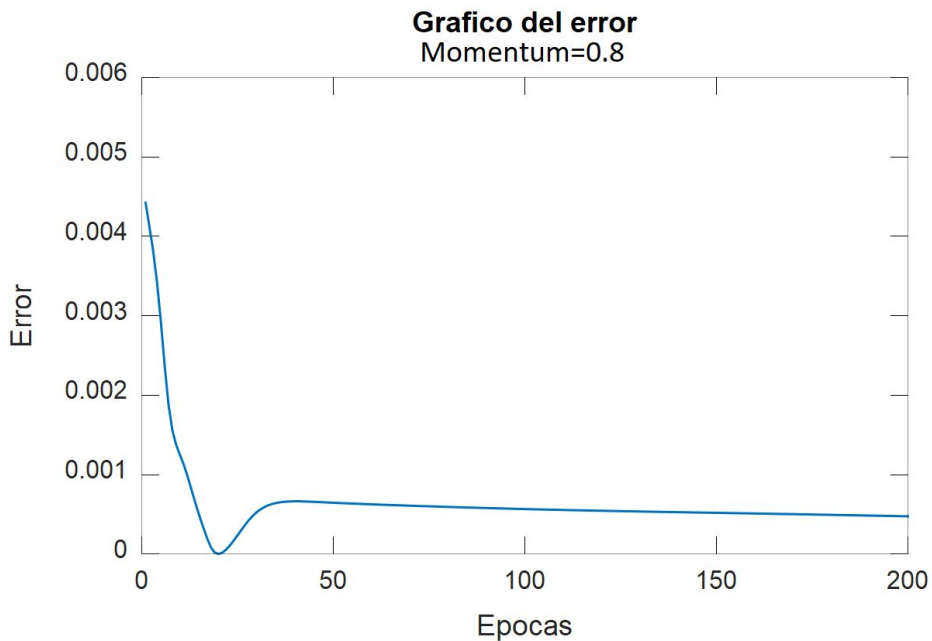
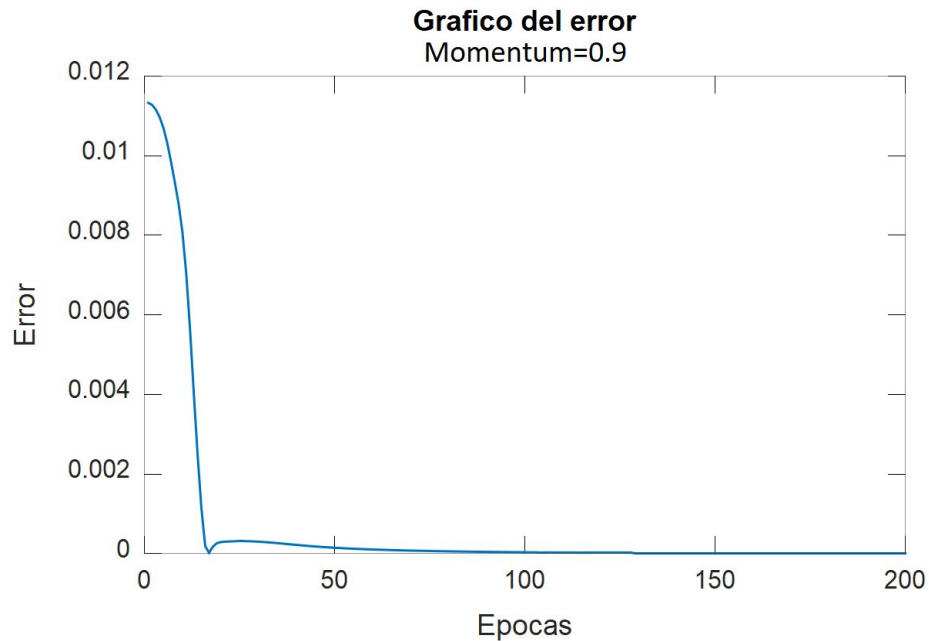


# Eta vs Eta adaptativo

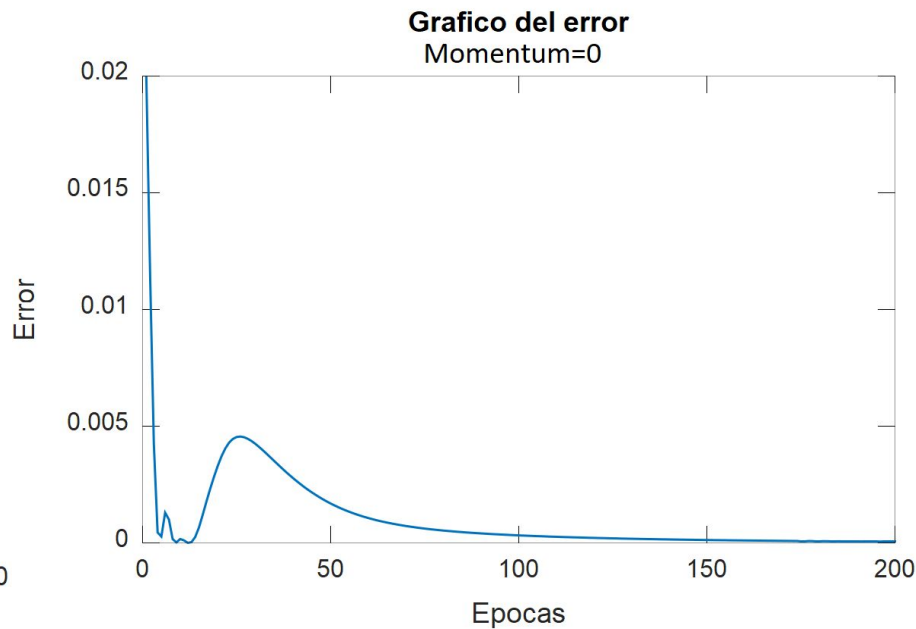
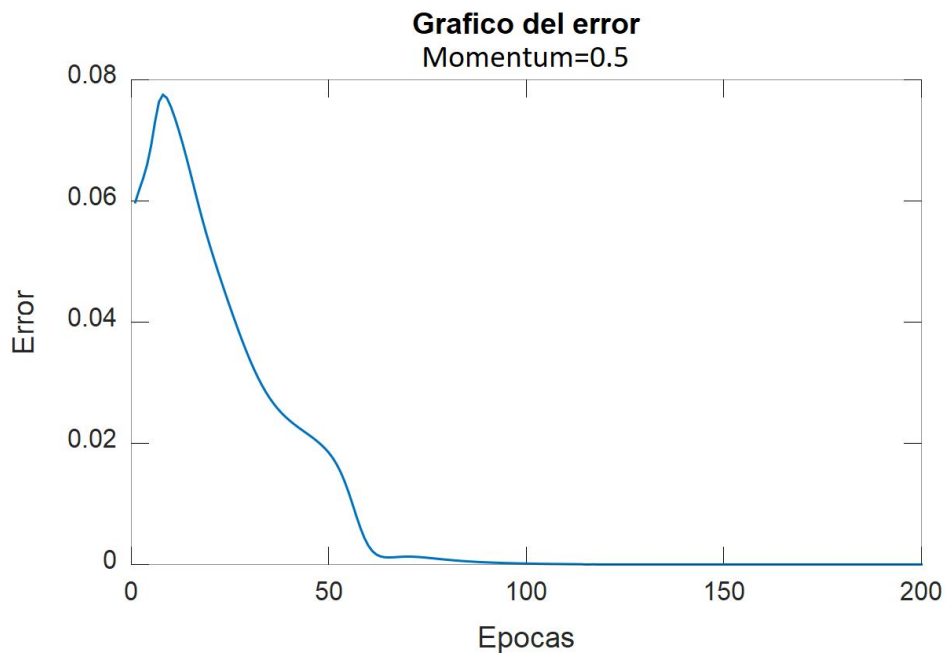
Eta:  $\alpha=0.5$ ,  $\beta= 1.5$ , Tercer parámetro= 3



# Momentum vs !Momentum



# Momentum vs !Momentum



# Resultados con diferentes arquitecturas

— — —

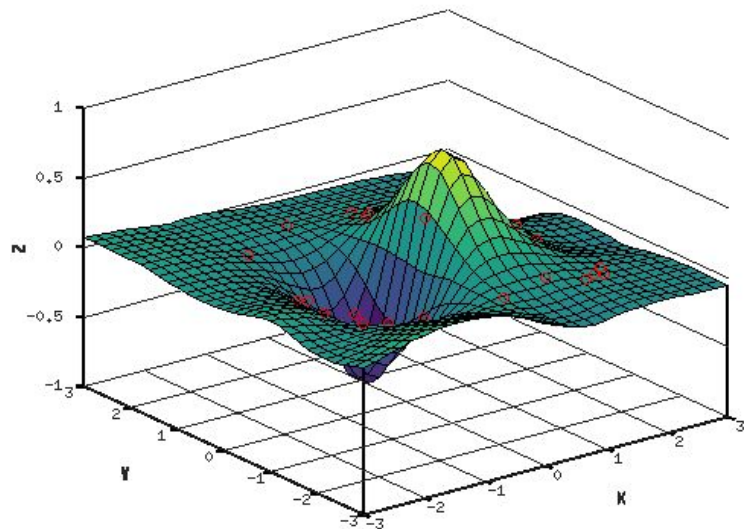
Terrain	Architecture	eta	momentum	Error (200 épocas)
Terrain05	2 6 10 1	0.08	0.9	1.6275e-4
Terrain05	2 7 10 1	0.08	0.9	6,12E-05
Terrain05	2 8 10 1	0.08	0.9	2,08E-06
Terrain05	2 9 10 1	0.08	0.9	3,34E-05
Terrain05	2 10 10 1	0.08	0.9	1,89E-05
Terrain05	2 6 10 1	1.4, 0.85, 3	0.9	2,42E-05
Terrain05	2 4 8 1	0.03	0.9	8,13E-06
Terrain07	2 6 10 1	0.08	0.9	2,50E-05
Terrain07	2 7 10 1	0.08	0.9	5,30E-04
Terrain07	2 8 10 1	0.08	0.9	0,038
Terrain07	2 9 10 1	0.08	0.9	1,40E-05
Terrain07	2 10 10 1	0.08	0.9	1,53E-05
Terrain14	2 6 10 1	0.08	0.9	0,0017409
Terrain14	2 7 10 1	0.08	0.9	0,0018283
Terrain14	2 8 10 1	0.08	0.9	0,0036958
Terrain14	2 9 10 1	0.08	0.9	0,0045899
Terrain14	2 10 10 1	0.08	0.9	8,31E-04



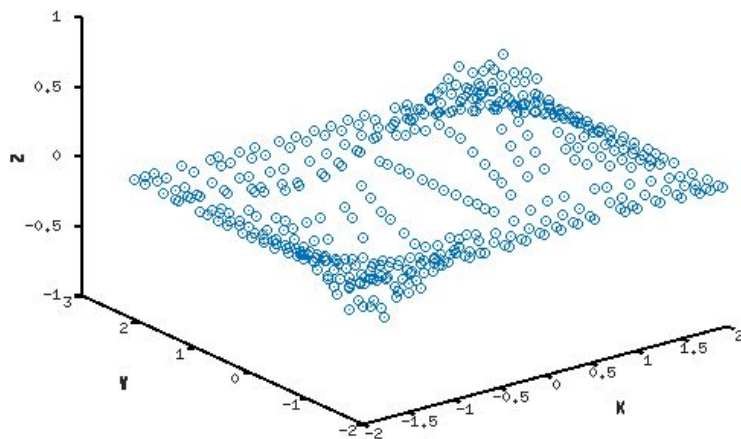
# Salida del programa con los mejores parámetros

Error de generalización:  $4.7799e-04$ .

Terreno realizado por la red



Terreno original



# Mejoras

- Familiarización con Octave para mejor manejo de recursos y tiempo de ejecución
- Valores de Eta adaptativo
- Más pruebas con mayor espectro de parámetros

¿Preguntas?

Fin