

**Práctica 4. Redes Neuronales**

**Daniel Parra Segovia**

**02591500K**

**Curso 2023/2024**

**ÍNDICE**

Ejercicio 1 Matlab 3

Ejercicio 2 Matlab 4

Ejercicio 3 Matlab 5

Ejercicio 1 Python 6

Ejercicio 2 Python 7

Ejercicio 3 Python 8

**Ejercicio 1 Python.**

**Regresión**

Utilizamos una red neuronal 8:4:1

MSE en entrenamiento = 0.6073

MSE en datos de prueba = 0.6233

No hay sobreajuste.

Utilizamos una red neuronal 8:8:4:1

MSE en entrenamiento = 0.5121

MSE en datos de prueba = 0.5309

No hay sobreajuste.

Utilizamos una red neuronal 8:64:64:1

MSE en entrenamiento = 0.4864

MSE en datos de prueba = 0.5095

No hay sobreajuste

Al aumentar la profundidad tarda más en entrenarse, con un poco más de precisión

**Ejercicio 2 Python.**

**Clasificación binaria**

Utilizamos una red neuronal 60:10:1

BCE en entrenamiento = 0.4119

BCE en datos de prueba = 0.5191

No hay sobreajuste

Utilizamos una red neuronal 60:60:30:1

BCE en entrenamiento = 0.0037

BCE en datos de prueba = 1.0933

Hay sobreajuste

Utilizamos una red neuronal 60:200:200:1

BCE en entrenamiento = 0.0048

BCE en datos de prueba = 1.1894

Hay sobreajuste

Al aumentar la profundidad tardan más en entrenarse y se produce sobreajuste.

**Ejercicio 3 Python.**

**Clasificación multiclase**

Utilizamos una red neuronal 4:20:3

CEL en entrenamiento = 0.4593

CEL en datos de prueba = 0.4543

Utilizamos una red neuronal 4:20:20:3

CEL en entrenamiento = 0.1847

CEL en datos de prueba = 0.2192

Utilizamos una red neuronal 4:200:200:3

CEL en entrenamiento = 0.0630

CEL en datos de prueba = 0.0860

Utilizamos una red neuronal 4:200:2000:200:3

CEL en entrenamiento = 0.0503

CEL en datos de prueba = 0.0495

Al aumentar la profundidad tardan más en entrenarse, pero hay un aumento significativo de la precisión

­