

Laboratorio No. 1



Galileo
UNIVERSIDAD
La Revolución en la Educación

Osiel Gutierrez Herrera
Carne: 21000958

LABORATORIO No. 1

Instrucciones

Del problema del Titanic participar en la competencia del Titanic •

- Probar al menos 5 arquitecturas y configuraciones distintas de redes
- Entregar todo el código:

- **5 arquitecturas**

- <https://github.com/OsielGH/StaticallLearning2.git>

- **Métricas que hayan realizado para clasificación**

1. Arquitectura No. 1

6 Neuronas 2 capas Activacion: sigmoid, Relu

```
#base secuencial para definir La estructura de La Red (SIEMPRE INICIAR CON SEQUENTIAL)
clasificador = Sequential()

#primera capa oculta solo La primera necesita (input_dim = inputs
clasificador.add(Dense(input_dim = inputs, units=6, activation='relu', kernel_initializer='uniform'))

#segunda capa oculta ya en La segunda ya no se necesita Los input
clasificador.add(Dense(units=6, activation='relu', kernel_initializer='uniform'))

#capa de salida
clasificador.add(Dense(units=1, activation='sigmoid', kernel_initializer='uniform'))

#parametros de optimización
clasificador.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

#parametros de entrenamiento.
clasificador.fit(X_train, y_train, batch_size=10, epochs=50)
```

2. Arquitectura No. 2

10 Neuronas 2 capas Activacion: sigmoid, Relu, than

```
#base secuencial para definir la estructura de la Red (SIEMPRE INICIAR CON SQUENTIAL)
clasificador = Sequential()

#primera capa oculta solo la primera necesita (input_dim = inputs
clasificador.add(Dense(input_dim = inputs, units=10, activation='relu', kernel_initializer='uniform'))

#segunda capa oculta ya en la segunda ya no se necesita los input
clasificador.add(Dense(units=10, activation='tanh', kernel_initializer='uniform'))

#capa de salida
clasificador.add(Dense(units=1, activation='sigmoid', kernel_initializer='uniform'))

#parametros de optimización
clasificador.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

#parametros de entrenamiento.
clasificador.fit(X_train, y_train, batch_size=10, epochs=75)
```

3. Arquitectura No. 3

15 Neuronas 3 capas Activacion: sigmoid, Relu, than, selu

```
#base secuencial para definir la estructura de la Red (SIEMPRE INICIAR CON SQUENTIAL)
clasificador = Sequential()

#primera capa oculta solo la primera necesita (input_dim = inputs
clasificador.add(Dense(input_dim = inputs, units=15, activation='relu', kernel_initializer='uniform'))

#segunda capa oculta ya en la segunda ya no se necesita los input
clasificador.add(Dense(units=15, activation='tanh', kernel_initializer='uniform'))

#tercera capa oculta ya en la segunda ya no se necesita los input
clasificador.add(Dense(units=15, activation='selu', kernel_initializer='uniform'))

#capa de salida
clasificador.add(Dense(units=1, activation='sigmoid', kernel_initializer='uniform'))

#parametros de optimización
clasificador.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

#parametros de entrenamiento.
clasificador.fit(X_train, y_train, batch_size=10, epochs=75)
```

4. Arquitectura No. 4

15 Neuronas 3 capas Activacion: sigmoid, softsign, tanh, selu

```
1 #base secuencial para definir la estructura de la Red (SIEMPRE INICIAR CON SQUENTIAL)
2 clasificador = Sequential()
3
4 #primera capa oculta solo la primera necesita (input_dim = inputs
5 clasificador.add(Dense(input_dim = inputs, units=15, activation='softsign', kernel_initializer='uniform'))
6
7 #segunda capa oculta ya en la segunda ya no se necesita los input
8 clasificador.add(Dense(units=15, activation='tanh', kernel_initializer='uniform'))
9
10 #tercera capa oculta ya en la segunda ya no se necesita los input
11 clasificador.add(Dense(units=15, activation='selu', kernel_initializer='uniform'))
12
13 #capa de salida
14 clasificador.add(Dense(units=1, activation='sigmoid', kernel_initializer='uniform'))

1 #parametros de optimización
2 clasificador.compile(optimizer='SGD', loss='binary_crossentropy', metrics=['accuracy'])
3
4 #parametros de entrenamiento.
5 clasificador.fit(X_train, y_train, batch_size=10, epochs=75)
```

5. Arquitectura No. 5

30 Neuronas 4 capas Activacion: sigmoid, softsign, tanh, selu, softmax

```
1 #base secuencial para definir la estructura de la Red (SIEMPRE INICIAR CON SQUENTIAL)
2 clasificador = Sequential()
3
4 #primera capa oculta solo la primera necesita (input_dim = inputs
5 clasificador.add(Dense(input_dim = inputs, units=30, activation='softsign', kernel_initializer='uniform'))
6
7 #segunda capa oculta ya en la segunda ya no se necesita los input
8 clasificador.add(Dense(units=30, activation='tanh', kernel_initializer='uniform'))
9
10 #tercera capa oculta ya en la segunda ya no se necesita los input
11 clasificador.add(Dense(units=30, activation='selu', kernel_initializer='uniform'))
12
13 #cuarta capa oculta ya en la segunda ya no se necesita los input
14 clasificador.add(Dense(units=30, activation='softmax', kernel_initializer='uniform'))
15
16 #capa de salida
17 clasificador.add(Dense(units=1, activation='sigmoid', kernel_initializer='uniform'))



1 #parametros de optimización
2 clasificador.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
3
4 #parametros de entrenamiento.
5 clasificador.fit(X_train, y_train, batch_size=10, epochs=200)
```

○ **Posición en leaderboard de la competencia (por cada configuración)**



1. Arquitectura No. 1

14923	Osiel Gutierrez Herrera		0.51674	1	3d
-------	--------------------------------	---	---------	---	----



2. Arquitectura No. 2

14882	Osiel Gutierrez Herrera		0.57416	3	1s
	Your Best Entry! Your most recent submission scored 0.57416, which is an improvement of your previous score of 0.51674. Great job!				Tweet this



3. Arquitectura No. 3

14884	Osiel Gutierrez Herrera		0.57655	4	1s
	Your Best Entry! Your most recent submission scored 0.57655, which is an improvement of your previous score of 0.57416. Great job!				Tweet this

4. Arquitectura No. 4

14802	Osiel Gutierrez Herrera		0.62200	5	1s
	Your Best Entry! Your most recent submission scored 0.62200, which is an improvement of your previous score of 0.57655. Great job!				Tweet this

5. Arquitectura No. 5

14804	Osiel Gutierrez Herrera		0.62200	6	1s
	Your Best Entry! Your submission scored 0.59808, which is not an improvement of your previous score. Keep trying!				

• **Concluir cual es el mejor resultado y explicar por qué**

El mejor resultado es la arquitectura 4, ya que cuenta 15 neuronas, 3 capas y 4 modos de activación siendo softsing, tanh, selu, sigmoid, y realizando 75 epocas para un resultado de accuracy de 0.6047

```
Epoch 75/75
48/48 [=====] - 0s 1ms/step - loss: 0.6712 - accuracy: 0.6047
<keras.callbacks.History at 0x7f8d24be1690>
```