LABORATORIO 2

Semilleros
by DSRP

Ejercicios

En el tutorial, vimos cómo construir redes neuronales profundas apilando capas dentro de un **Sequential** model. Al agregar una función de activación después de las capas ocultas, le dimos a la red la capacidad de aprender relaciones más complejas (no lineales) en los datos.

En estos ejercicios, construirá una red neuronal con varias capas ocultas y luego explorará algunas funciones de activación más allá de ReLU.

¡Ejecute la siguiente celda para configurar todo!

[ ]:

```
import tensorflow as tf

# Setup plotting
import matplotlib.pyplot as plt

plt.style.use('seaborn-whitegrid')
# Set Matplotlib defaults
plt.rc('figure', autolayout=True)
plt.rc('axes', labelweight='bold', labelsize='large',
       titleweight='bold', titlesize=18, titlepad=10)

# Setup feedback system
from learntools.core import binder
binder.bind(globals())
from learntools.deep_learning_intro.ex2 import *
```

Semilleros by DSRP

Ejercicios

En el *Concrete* dataset, su tarea es predecir la resistencia a la compresión del concreto fabricado de acuerdo con varias recetas.

Ejecute la siguiente celda de código sin cambios para cargar el conjunto de datos.

[ ]:
```
import pandas as pd

concrete = pd.read_csv('../input/dl-course-data/concrete.csv')
concrete.head()
```

[5]:

	Cement	BlastFurnaceSlag	FlyAsh	Water	Superplasticizer	CoarseAggregate	FineAggregate	Age	CompressiveStrength
0	540.0	0.0	0.0	162.0	2.5	1040.0	676.0	28	79.99
1	540.0	0.0	0.0	162.0	2.5	1055.0	676.0	28	61.89
2	332.5	142.5	0.0	228.0	0.0	932.0	594.0	270	40.27
3	332.5	142.5	0.0	228.0	0.0	932.0	594.0	365	41.05
4	198.6	132.4	0.0	192.0	0.0	978.4	825.5	360	44.30

Semilleros by DSRP

Ejercicios

Paso1: Input shape

El objetivo de esta tarea es la columna '*CompressiveStrength'*. Las columnas restantes son las características que usaremos como entradas.

¿Cuál sería la forma de entrada para este conjunto de datos?

```
# YOUR CODE HERE
input_shape = [8]

# Check your answer
q_1.check()
```

Correct

Semilleros by DSRP

Ejercicios

Paso2: Define a model with Hidden Layers

Ahora cree un modelo con tres capas ocultas, cada una con 512 **units** y la activación de ReLU. Asegúrese de incluir una capa de salida de una unidad y sin activación, y también **input_shape** como argumento para la primera capa.

```
from tensorflow import keras
from tensorflow.keras import layers

# YOUR CODE HERE
model = keras.Sequential([
    layers.Dense(units=512, activation='relu', input_shape=[8]),
    layers.Dense(units=512, activation ='relu'),
    layers.Dense(units=512, activation ='relu'),
    layers.Dense(units=1)
])

# Check your answer
q_2.check()
```

Correct

Semilleros by DSRP

Ejercicios

Paso3: Activation Layers

Exploremos algunas funciones de activaciones.

La forma habitual de adjuntar una función de activación a una **Dense layer** es incluirla como parte de la definición con el argumento de **activation**. A veces, sin embargo, querrá poner alguna otra capa entre la **Dense layer** y su función de activación. En este caso, podemos definir la activación en su propia **Activation layer**, así:

Note:

```
layers.Dense(units=8),
layers.Activation('relu')
```

This is completely equivalent to the ordinary way: `layers.Dense(units=8, activation='relu')`.

Rewrite the following model so that each activation is in its own `Activation` layer.

```
### YOUR CODE HERE: rewrite this to use activation layers
model = keras.Sequential([

    layers.Dense(units=32, input_shape= [8]),
    layers.Activation('relu'),
    layers.Dense(units=32),
    layers.Activation('relu'),
    layers.Dense(1),
])

# Check your answer
q_3.check()
```

Correct

Semilleros by DSRP

Ejercicios

Optional: Alternatives to ReLU

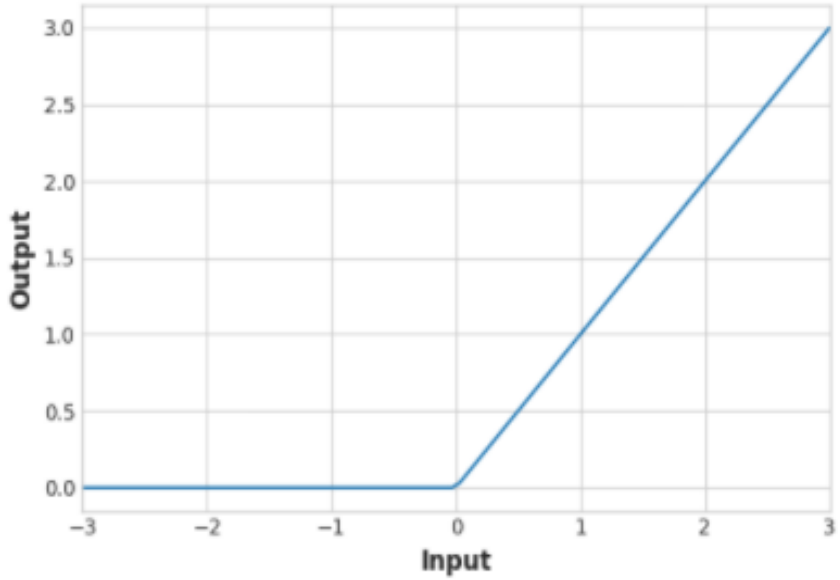
Hay toda una familia de variantes de la activación 'relu' - 'elu', 'selu' y 'swish', entre otras, todas las cuales puedes usar en Keras.

A veces, una activación funcionará mejor que otra en una tarea determinada, por lo que podría considerar experimentar con activaciones a medida que desarrolla un modelo. La activación de ReLU tiende a funcionar bien en la mayoría de los problemas, por lo que es una buena opción para empezar.

Veamos los gráficos de algunos de estos. Cambie la activación de 'relu' a una de las otras mencionadas anteriormente. Luego, ejecute la celda para ver el gráfico. (Consulte la documentación para obtener más ideas).

[42]:

```
# YOUR CODE HERE: Change 'relu' to 'elu', 'selu', 'swish'... or something else
activation_layer = layers.Activation('relu')

x = tf.linspace(-3.0, 3.0, 100)
y = activation_layer(x) # once created, a layer is callable just like a function

plt.figure(dpi=100)
plt.plot(x, y)
plt.xlim(-3, 3)
plt.xlabel("Input")
plt.ylabel("Output")
plt.show()
```

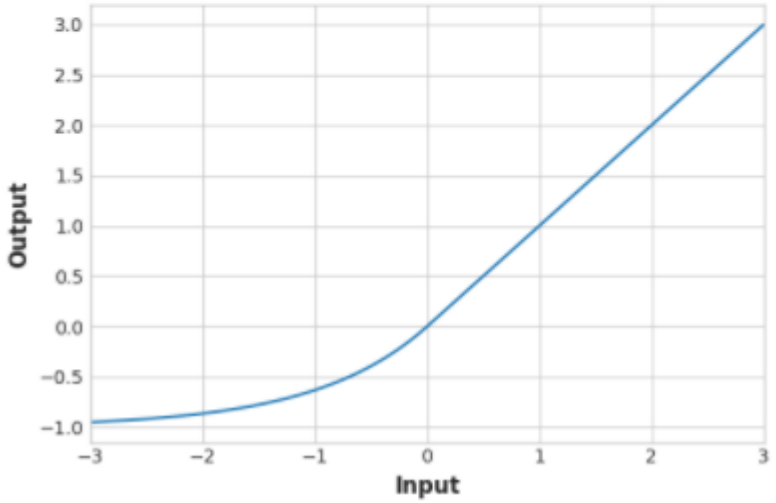


Semilleros by DSRP

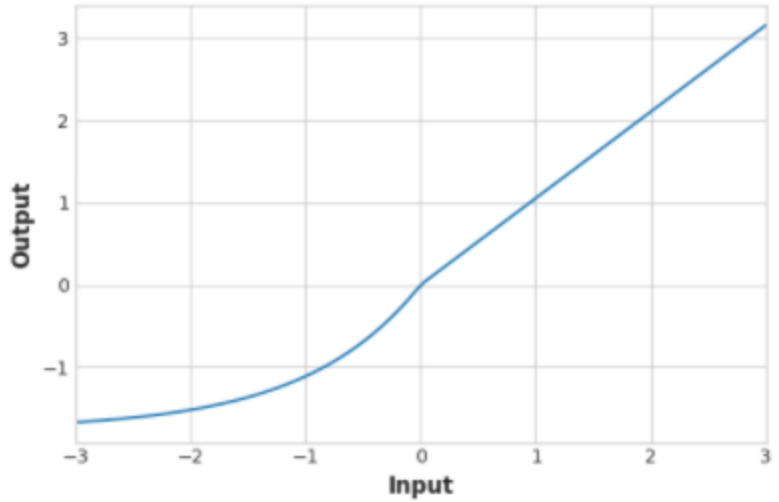
Ejercicios

Optional: Alternatives to ReLU

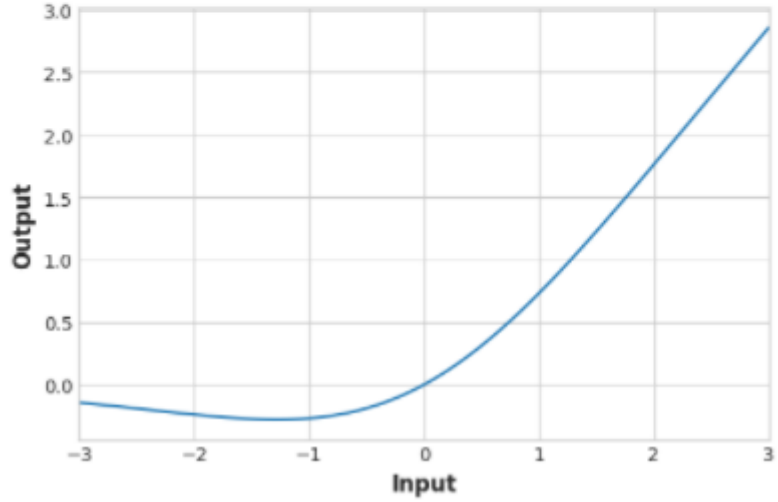
```
# YOUR CODE HERE: Change 'relu' to 'elu', 'selu', 'swish'... or somethi
activation_layer = layers.Activation('elu')

x = tf.linspace(-3.0, 3.0, 100)
y = activation_layer(x) # once created, a layer is callable just like a

plt.figure(dpi=100)
plt.plot(x, y)
plt.xlim(-3, 3)
plt.xlabel("Input")
plt.ylabel("Output")
plt.show()
```




```
# YOUR CODE HERE: Change 'relu' to 'elu', 'selu', 'swish'... or someth
activation_layer = layers.Activation('selu')

x = tf.linspace(-3.0, 3.0, 100)
y = activation_layer(x) # once created, a layer is callable just like

plt.figure(dpi=100)
plt.plot(x, y)
plt.xlim(-3, 3)
plt.xlabel("Input")
plt.ylabel("Output")
plt.show()
```




```
# YOUR CODE HERE: Change 'relu' to 'elu', 'selu', 'swish'... or somet
activation_layer = layers.Activation('swish')

x = tf.linspace(-3.0, 3.0, 100)
y = activation_layer(x) # once created, a layer is callable just like

plt.figure(dpi=100)
plt.plot(x, y)
plt.xlim(-3, 3)
plt.xlabel("Input")
plt.ylabel("Output")
plt.show()
```



Semilleros by DSRP

LABORATORIO 3

Semilleros
by DSRP

Ejercicio

En este ejercicio, entrenará una red neuronal en el *Fuel Economy* dataset y luego explorará el efecto de la tasa de aprendizaje y el tamaño del lote en SGD.

Cuando esté listo, ejecute la siguiente celda para configurar todo.

```
# Setup plotting
import matplotlib.pyplot as plt
from learntools.deep_learning_intro.dltools import animate_sgd
plt.style.use('seaborn-whitegrid')
# Set Matplotlib defaults
plt.rc('figure', autolayout=True)
plt.rc('axes', labelweight='bold', labelsize='large',
       titleweight='bold', titlesize=18, titlepad=10)
plt.rc('animation', html='html5')

# Setup feedback system
from learntools.core import binder
binder.bind(globals())
from learntools.deep_learning_intro.ex3 import *
```

Semilleros by DSRP

Ejercicio

En el *Fuel Economy* dataset, su tarea consiste en predecir la economía de combustible de un automóvil determinadas características como el tipo de motor o el año en que se fabricó.

Primero cargue el conjunto de datos ejecutando la celda a continuación.

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.compose import make_column_transformer, make_column_selector
from sklearn.model_selection import train_test_split

fuel = pd.read_csv('../input/dl-course-data/fuel.csv')

X = fuel.copy()
# Remove target
y = X.pop('FE')

preprocessor = make_column_transformer(
    (StandardScaler(),
     make_column_selector(dtype_include=np.number)),
    (OneHotEncoder(sparse=False),
     make_column_selector(dtype_include=object)),
)

X = preprocessor.fit_transform(X)
y = np.log(y) # log transform target instead of standardizing

input_shape = [X.shape[1]]
print("Input shape: {}".format(input_shape))
```

```
Input shape: [50]
```

Semilleros by DSRP

Ejercicio

Eche un vistazo a los datos si lo desea. Nuestro objetivo en este caso es la columna **'FE'** y las columnas restantes son las características.

```
# Uncomment to see original data
fuel.head()
```

	EngDispl	NumCyl	Transmission	FE	AirAspirationMethod	NumGears	TransLockup	TransCreeperGear	DriveDesc	IntakeValvePerCyl	ExhaustValvesPerCyl	CarlineClassDesc	VarValveTiming
0	4.7	8	AM6	28.0198	NaturallyAspirated	6	1	0	TwoWheelDriveRear	2	2	2Seaters	1
1	4.7	8	M6	25.6094	NaturallyAspirated	6	1	0	TwoWheelDriveRear	2	2	2Seaters	1
2	4.2	8	M6	26.8000	NaturallyAspirated	6	1	0	AllWheelDrive	2	2	2Seaters	1
3	4.2	8	AM6	25.0451	NaturallyAspirated	6	1	0	AllWheelDrive	2	2	2Seaters	1
4	5.2	10	AM6	24.8000	NaturallyAspirated	6	0	0	AllWheelDrive	2	2	2Seaters	1

Ejecute la siguiente celda para definir la red que usaremos para esta tarea.

[4]:

```
from tensorflow import keras
from tensorflow.keras import layers

model = keras.Sequential([
    layers.Dense(128, activation='relu', input_shape=input_shape),
    layers.Dense(128, activation='relu'),
    layers.Dense(64, activation='relu'),
    layers.Dense(1),
])
```

Semilleros by DSRP

Ejercicios

Paso1: Add Loss and Optimizer

Antes de entrenar la red, debemos definir la pérdida y el optimizador que usaremos. Usando el método de compilación del modelo, agregue el optimizador Adam y la pérdida MAE.

```
# YOUR CODE HERE
model.compile(
optimizer='adam',
loss='mae',
)

# Check your answer
q_1.check()
```

Correct

Semilleros by DSRP

Ejercicios

Paso2: Train model

Una vez que haya definido el modelo y lo haya compilado con una pérdida y un optimizador, estará listo para el entrenamiento. Entrene la red durante 200 épocas con un tamaño de lote de 128. Los datos de entrada son **X** con destino **y**

```
# YOUR CODE HERE
history = model.fit(
    X, y,
    batch_size=128,
    epochs=200,
)

# Check your answer
q_2.check()
```

```
Epoch 1/200
9/9 [==============================] - 1s 3ms/step - loss: 3.1927
Epoch 2/200
9/9 [==============================] - 0s 2ms/step - loss: 1.1906
Epoch 3/200
9/9 [==============================] - 0s 2ms/step - loss: 0.7161
Epoch 4/200
9/9 [==============================] - 0s 2ms/step - loss: 0.4400
Epoch 5/200
9/9 [==============================] - 0s 2ms/step - loss: 0.2911
Epoch 6/200
```

```
9/9 [==============================] - 0s 3ms/step - loss: 0.0268
Epoch 186/200
9/9 [==============================] - 0s 4ms/step - loss: 0.0276
Epoch 187/200
9/9 [==============================] - 0s 3ms/step - loss: 0.0274
Epoch 188/200
9/9 [==============================] - 0s 3ms/step - loss: 0.0233
Epoch 189/200
9/9 [==============================] - 0s 2ms/step - loss: 0.0262
Epoch 190/200
9/9 [==============================] - 0s 3ms/step - loss: 0.0279
Epoch 191/200
9/9 [==============================] - 0s 2ms/step - loss: 0.0289
Epoch 192/200
9/9 [==============================] - 0s 2ms/step - loss: 0.0414
Epoch 193/200
9/9 [==============================] - 0s 2ms/step - loss: 0.0366
Epoch 194/200
9/9 [==============================] - 0s 3ms/step - loss: 0.0322
Epoch 195/200
9/9 [==============================] - 0s 3ms/step - loss: 0.0265
Epoch 196/200
9/9 [==============================] - 0s 3ms/step - loss: 0.0285
Epoch 197/200
9/9 [==============================] - 0s 3ms/step - loss: 0.0280
Epoch 198/200
9/9 [==============================] - 0s 3ms/step - loss: 0.0357
Epoch 199/200
9/9 [==============================] - 0s 3ms/step - loss: 0.0373
Epoch 200/200
9/9 [==============================] - 0s 3ms/step - loss: 0.0388
```

Correct

Semilleros by DSRP

Ejercicios

Paso3: Evaluate training

```
# View the solution (Run this cell to receive credit!)
q_3.check()
```

Si entrenara el modelo por más tiempo, ¿esperaría que la pérdida disminuya aún más?

Esto depende de cómo haya evolucionado la pérdida durante el entrenamiento: si las curvas de aprendizaje se han estabilizado, normalmente no habrá ninguna ventaja para entrenar para épocas adicionales. Por el contrario, si la pérdida parece seguir disminuyendo, entonces entrenar durante más tiempo podría ser ventajoso.

Con la tasa de aprendizaje y el tamaño del lote, tiene cierto control sobre:
- Cuánto tiempo se tarda en entrenar a un modelo
- Qué ruidosas son las curvas de aprendizaje
- Qué pequeña se vuelve la pérdida

Para comprender mejor estos dos parámetros, veremos el modelo lineal, nuestra red neuronal más simple. Teniendo solo un peso y un sesgo, es más fácil ver qué efecto tiene un cambio de parámetro.
La siguiente celda generará una animación como la del tutorial. Cambie los valores **de learning_rate, batch_size** y **num_examples** (cuántos puntos de datos) y luego ejecute la celda. (Puede que tarde uno o dos minutos). Pruebe las siguientes combinaciones o pruebe algunas propias:

learning_rate	batch_size	num_examples
0.05	32	256
0.05	2	256
0.05	128	256
0.02	32	256
0.2	32	256
1.0	32	256
0.9	4096	8192
0.99	4096	8192

Semilleros by DSRP

Ejercicios

Paso3: Evaluate training

```
# YOUR CODE HERE: Experiment with different values for the learning rate, batch size, and number of ex
learning_rate = 0.05
batch_size = 128
num_examples = 256

animate_sgd(
    learning_rate=learning_rate,
    batch_size=batch_size,
    num_examples=num_examples,
    # You can also change these, if you like
    steps=50, # total training steps (batches seen)
    true_w=3.0, # the slope of the data
    true_b=2.0, # the bias of the data
)
```


```
# YOUR CODE HERE: Experiment with different values for the learning rate, batch size, and number of examples
learning_rate = 0.9
batch_size = 4096
num_examples = 8192

animate_sgd(
    learning_rate=learning_rate,
    batch_size=batch_size,
    num_examples=num_examples,
    # You can also change these, if you like
    steps=50, # total training steps (batches seen)
    true_w=3.0, # the slope of the data
    true_b=2.0, # the bias of the data
)
```

Semilleros by DSRP

Ejercicios

Paso4: Learning Rate and Batch Size

¿Qué efecto tuvo el cambio de estos parámetros? Una vez que lo haya pensado, ejecute la celda a continuación para discutir un poco.

```
# View the solution (Run this cell to receive credit!)
q_4.check()
```

Correct:

Probablemente haya visto que los tamaños de lote más pequeños daban actualizaciones de peso más ruidosas y curvas de pérdida. Esto se debe a que cada lote es una pequeña muestra de datos y las muestras más pequeñas tienden a dar estimaciones más ruidosas. Sin embargo, los lotes más pequeños pueden tener un efecto de "promediado" que puede ser beneficioso.

Las tasas de aprendizaje más pequeñas hacen que las actualizaciones sean más pequeñas y la formación tarda más en converger. Las tasas de aprendizaje elevadas pueden acelerar la capacitación, pero no "se adapten" al mínimo también. Cuando la tasa de aprendizaje es demasiado grande, la formación puede fallar por completo. (Intente establecer la tasa de aprendizaje en un valor grande como 0,99 para ver esto).

Semilleros by DSRP