Raport PS5 Piotr Szewczul

1.

```
import tensorflow as tf
import numpy as np
from tensorflow.keras import regularizers
```

```
def func(t):
 return np.sin(1 + t * np.pi * 2) / 2 + 1

def get_regression_dataset(start=0, end=3, seed=1, samples=80):
 t = np.linspace(start, end, samples)
 oryginal = func(t)
 np.random.seed(seed)
 noisy = oryginal + np.random.normal(0, 0.2, samples)
 return noisy, oryginal
```

```
#define dataset
x_train, y_train  = get_regression_dataset()
x_test, y_test = get_regression_dataset(seed=2)

x_train = x_train.reshape(80,1)
x_test = x_test.reshape(80,1)
y_train = y_train.reshape(80,1)
y_test = y_test.reshape(80,1)

x_train = tf.keras.utils.normalize(x_train, axis = 0) # scales data between 0 and 1
x_test = tf.keras.utils.normalize(x_test, axis = 0) # scales data between 0 and 1
```

```
model = tf.keras.Sequential(
 [
 tf.keras.layers.Dense(10,input_shape=(1,), activation="relu", kernel_regularizer=regularizers.l2(0.01)),
 tf.keras.layers.Dense(100, activation="relu", kernel_regularizer=regularizers.l2(0.01)),
 tf.keras.layers.Dense(1)
 ]
)
```

2. Droput regulalizer

```
model = tf.keras.Sequential(
 [
 tf.keras.layers.Dense(10,input_shape=(1,), activation="relu"),
 tf.keras.layers.Dropout(0.2),
 tf.keras.layers.Dense(100, activation="relu"),
 tf.keras.layers.Dropout(0.2),
 tf.keras.layers.Dense(1)
 ]
)
```

```
Epoch 1/15
80/80 [==============================] - 1s 2ms/step - loss: 0.6688 - MSE: 0.6262
Epoch 2/15
80/80 [==============================] - 0s 2ms/step - loss: 0.2927 - MSE: 0.1071
Epoch 3/15
80/80 [==============================] - 0s 2ms/step - loss: 0.2778 - MSE: 0.1060
Epoch 4/15
80/80 [==============================] - 0s 2ms/step - loss: 0.3158 - MSE: 0.1276
Epoch 5/15
80/80 [==============================] - 0s 2ms/step - loss: 0.2625 - MSE: 0.0960
Epoch 6/15
80/80 [==============================] - 0s 3ms/step - loss: 0.2513 - MSE: 0.0870
Epoch 7/15
80/80 [==============================] - 0s 2ms/step - loss: 0.2444 - MSE: 0.0839
Epoch 8/15
80/80 [==============================] - 0s 2ms/step - loss: 0.2383 - MSE: 0.0808
Epoch 9/15
80/80 [==============================] - 0s 2ms/step - loss: 0.2632 - MSE: 0.1014
Epoch 10/15
80/80 [==============================] - 0s 2ms/step - loss: 0.2652 - MSE: 0.0976
Epoch 11/15
80/80 [==============================] - 0s 2ms/step - loss: 0.2135 - MSE: 0.0641
Epoch 12/15
80/80 [==============================] - 0s 2ms/step - loss: 0.2298 - MSE: 0.0746
Epoch 13/15
80/80 [==============================] - 0s 2ms/step - loss: 0.2662 - MSE: 0.0986
Epoch 14/15
80/80 [==============================] - 0s 2ms/step - loss: 0.2353 - MSE: 0.0831
Epoch 15/15
80/80 [==============================] - 0s 2ms/step - loss: 0.2236 - MSE: 0.0802
3/3 [==============================] - 0s 3ms/step - loss: 0.1855 - MSE: 0.0485
0.18549910187721252
0.048502955585718155
```

L1 regulalizer

```
model = tf.keras.Sequential(
 [
 tf.keras.layers.Dense(10,input_shape=(1,), activation="relu", kernel_regularizer=regularizers.l1(0.01)),
 tf.keras.layers.Dense(100, activation="relu", kernel_regularizer=regularizers.l1(0.01)),
 tf.keras.layers.Dense(1)
 ]
)
```

```
Epoch 1/15
80/80 [==============================] - 1s 1ms/step - loss: 1.5580 - MSE: 0.5585
Epoch 2/15
80/80 [==============================] - 0s 1ms/step - loss: 0.7035 - MSE: 0.1060
Epoch 3/15
80/80 [==============================] - 0s 1ms/step - loss: 0.4384 - MSE: 0.1038
Epoch 4/15
80/80 [==============================] - 0s 2ms/step - loss: 0.3852 - MSE: 0.1024
Epoch 5/15
80/80 [==============================] - 0s 2ms/step - loss: 0.3758 - MSE: 0.1015
Epoch 6/15
80/80 [==============================] - 0s 2ms/step - loss: 0.3695 - MSE: 0.1024
Epoch 7/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3700 - MSE: 0.1053
Epoch 8/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3662 - MSE: 0.1068
Epoch 9/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3609 - MSE: 0.1046
Epoch 10/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3568 - MSE: 0.1031
Epoch 11/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3598 - MSE: 0.1069
Epoch 12/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3518 - MSE: 0.1059
Epoch 13/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3601 - MSE: 0.1073
Epoch 14/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3521 - MSE: 0.1032
Epoch 15/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3504 - MSE: 0.1040
3/3 [==============================] - 0s 3ms/step - loss: 0.3473 - MSE: 0.1021
0.34725329279899597
0.10210946947336197
```

L2 regulalizer

```
model = tf.keras.Sequential(
 [
 tf.keras.layers.Dense(10,input_shape=(1,), activation="relu", kernel_regularizer=regularizers.l2(0.01)),
 tf.keras.layers.Dense(100, activation="relu", kernel_regularizer=regularizers.l2(0.01)),
 tf.keras.layers.Dense(1)
 ]
)
```

```
Epoch 1/15
80/80 [==============================] - 1s 1ms/step - loss: 0.8617 - MSE: 0.7325
Epoch 2/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3759 - MSE: 0.1446
Epoch 3/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3314 - MSE: 0.1149
Epoch 4/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3200 - MSE: 0.1131
Epoch 5/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3187 - MSE: 0.1159
Epoch 6/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3168 - MSE: 0.1133
Epoch 7/15
80/80 [==============================] - 0s 2ms/step - loss: 0.3170 - MSE: 0.1131
Epoch 8/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3220 - MSE: 0.1157
Epoch 9/15
80/80 [==============================] - 0s 2ms/step - loss: 0.3129 - MSE: 0.1094
Epoch 10/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3122 - MSE: 0.1089
Epoch 11/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3104 - MSE: 0.1067
Epoch 12/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3107 - MSE: 0.1049
Epoch 13/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3070 - MSE: 0.1044
Epoch 14/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3023 - MSE: 0.1002
Epoch 15/15
80/80 [==============================] - 0s 1ms/step - loss: 0.3059 - MSE: 0.1021
3/3 [==============================] - 0s 2ms/step - loss: 0.3005 - MSE: 0.0988
0.30049461126327515
0.09883756190538406
```

Zastosowanie metody DROPOUT pozwala uzyskać wyniki najbardziej zbliżone do zakładanych. Regularyzacja technikami L1 oraz L2 daje podobne wyniki. Technika DROPUT w tym wypadku mogła okazać się najbardziej dostosowana ze względu na pominięcie obserwacji odstających. Metody L1 oraz L2 wykonane zostały z określonymi wartościami alfa, które mogą nie być najbardziej dostosowane to badanego przypadku.