q3_tensorflow-2

February 27, 2024

[ ]:
```
import tensorflow as tf
from tensorflow.keras import layers
from sklearn.model_selection import train_test_split
import numpy as np
import matplotlib.pyplot as plt
```

[ ]:
```
# define the model
model = tf.keras.Sequential([
    layers.Dense(50, activation='relu', input_shape=(1,)),
    layers.Dense(50, activation='relu'),
    layers.Dense(1)
])

model.compile(optimizer=tf.keras.optimizers.Adam(0.01),
              loss=tf.keras.losses.MeanSquaredError())
```

[ ]:
```
# generate data for uniform sampling
num_samples = 1000
x = np.linspace(-1, 1, num_samples).reshape(-1, 1)
y = 1 / (1 + 25 * x**2)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2,␣
 ↪random_state=42)
```

[ ]:
```
# train the model
history = model.fit(x_train, y_train, epochs=1000, batch_size=32, verbose=1)
```

[ ]:
```
# evaluate the model
test_loss = model.evaluate(x_test, y_test, verbose=0)
print('Test Loss: %.6f' % test_loss)
```

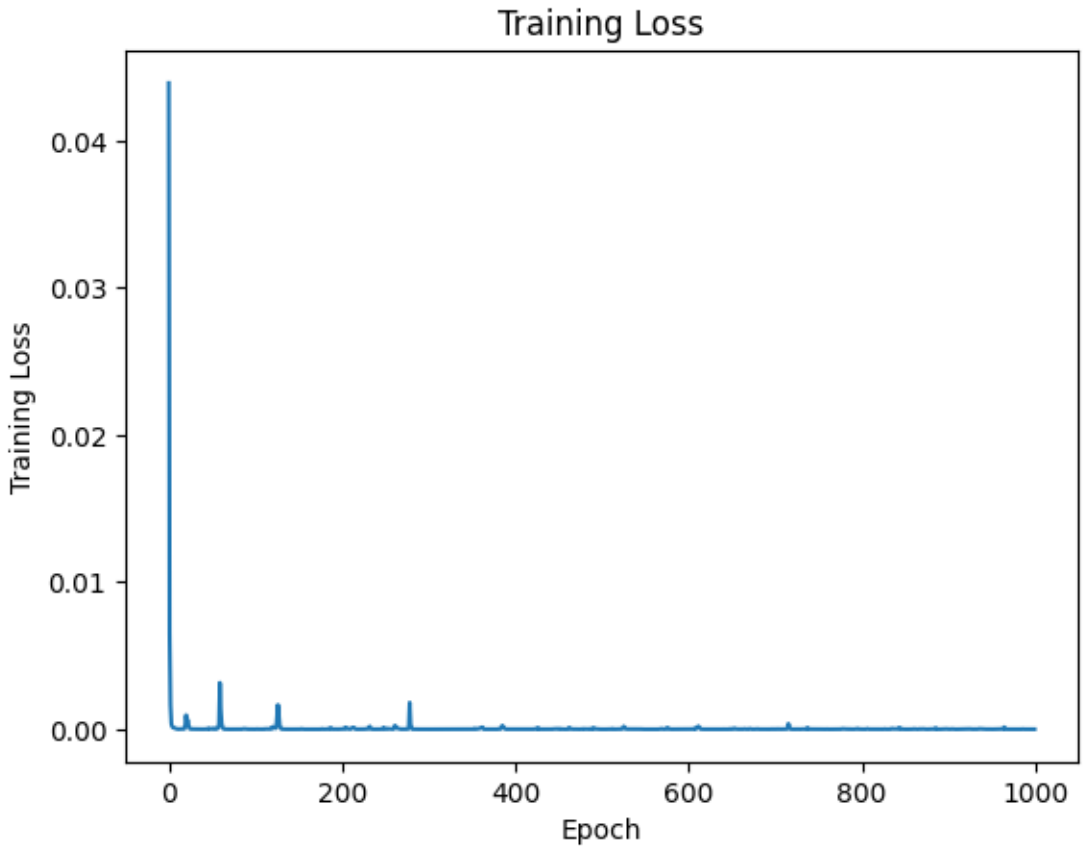
```
Test Loss: 0.000004
```

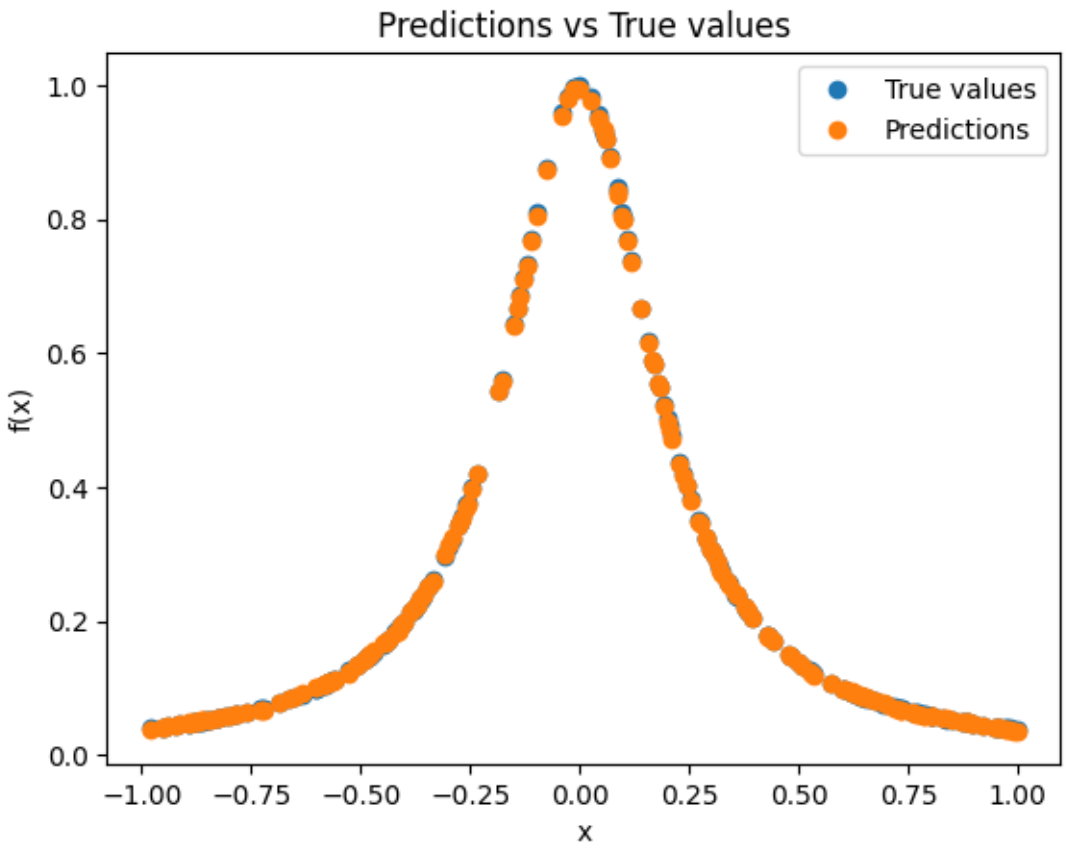
[ ]:
```
# predictions
outputs = model.predict(x_test)
```

```
7/7 [==============================] - 0s 1ms/step
```

[ ]:
```
plt.figure()
plt.plot(history.history['loss'])
plt.xlabel('Epoch')
plt.ylabel('Training Loss')
plt.title('Training Loss')
plt.show()

plt.figure()
plt.plot(x_test, y_test, 'o' ,label='True values')
plt.plot(x_test, outputs, 'o' ,label='Predictions')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title('Predictions vs True values')
plt.legend()
plt.show()
```




Predictions vs True values
True values
Predictions
f(x)
x
1.0
0.8
0.6
0.4
0.2
0.0
−1.00
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00