tf2-keras

August 21, 2019

1 Example of TF-2 keras model

[1]:
```
import tensorflow as tf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

2 Build training data

[2]:
```
n = 100
TRUE_W = 3.0
TRUE_b = 2.0

# random samples from normal distribution
np.random.seed(1)
r = np.random.normal(loc=0, scale=0.5, size=n)

# build data
inputs = np.random.normal(loc=0, scale=0.5, size=n)
outputs = TRUE_W * inputs + TRUE_b + r
```

3 Define model

[3]:
```
inputs = inputs.reshape((-1, 1))
outputs = outputs.reshape((-1, 1))
```

3.1 Construct model

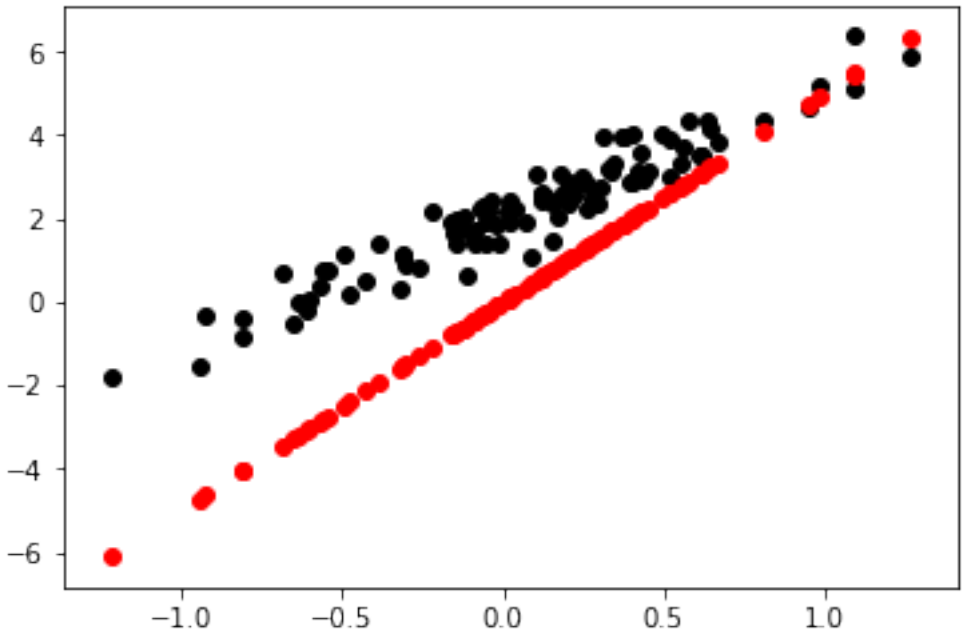
[4]:
```
model = tf.keras.Sequential([
    tf.keras.layers.Dense(
        units=1,
        input_shape=(1, ),
        use_bias=True,
        kernel_initializer=tf.initializers.constant(5.0), # W = 5.0
        bias_initializer=tf.initializers.constant(0.0)) # b = 0.0
```

```
])
```

3.2 Show model prediction before training

[5]:
```
def plot_model(pred):
    plt.scatter(inputs, outputs, c='black')
    plt.scatter(inputs, pred, c='r')
    plt.show()

predictions = model(inputs)
plot_model(predictions)
```



4 Train model

4.1 Configure training

[6]:
```
N_EPOCHS = 100
LEARNING_RATE = 0.1

model.compile(
    optimizer=tf.keras.optimizers.SGD(learning_rate=LEARNING_RATE),
    loss=tf.keras.losses.MeanSquaredError(),
    metrics=[tf.keras.metrics.MeanSquaredError()])
```

4.2 Execute training

[7]:
```
h = model.fit(inputs,
              outputs,
              verbose=0,
              batch_size=n,
              epochs=N_EPOCHS)
```

4.3 Show model prediction after training

[8]:
```
new_predictions = model(inputs)
plot_model(new_predictions)
```