Mrunmayee Naik - 37 DL

EXPERIMENT - 04

Mini Batch Gradient Descent

Code:

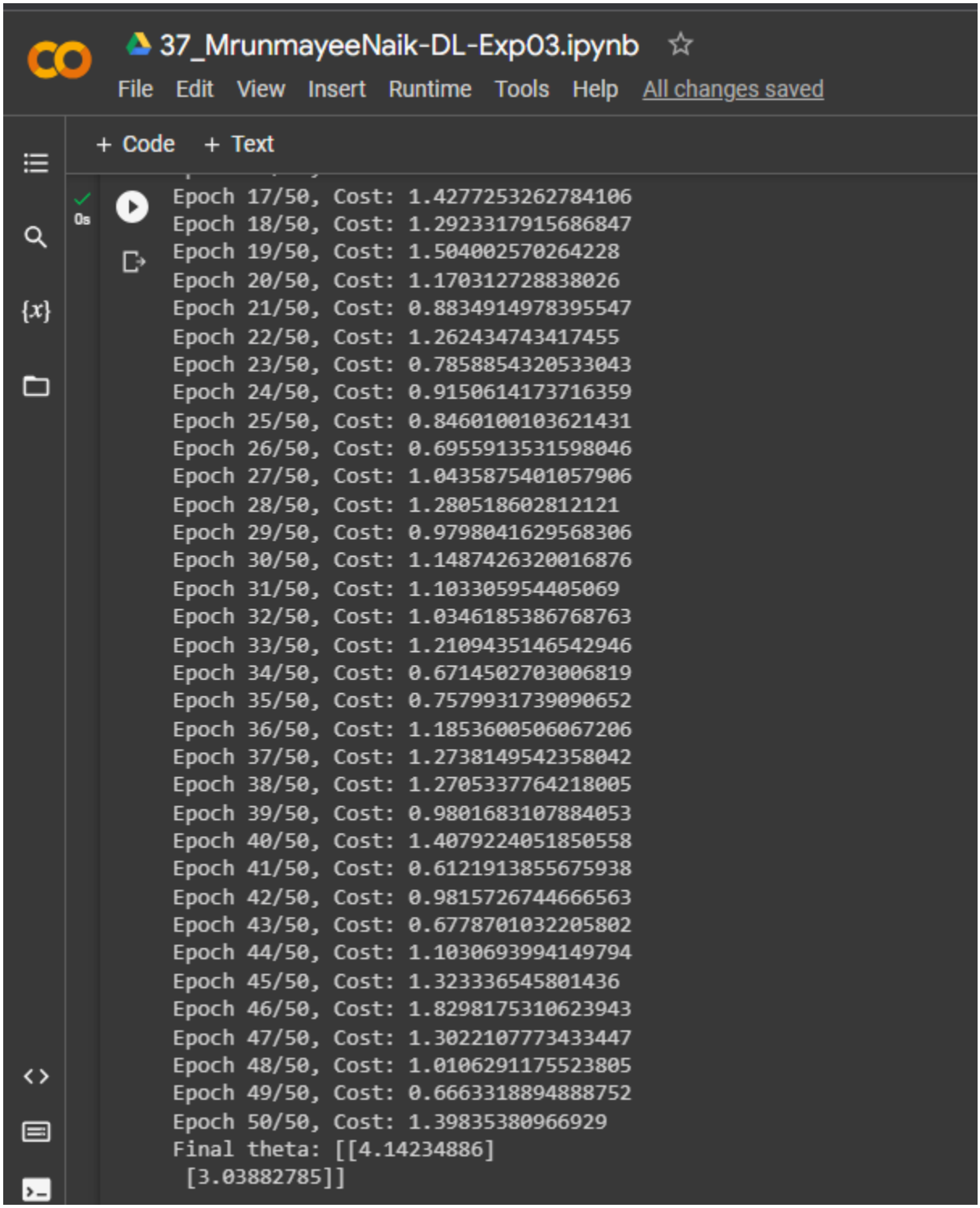
```
import numpy as np
np.random.seed(0)
X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X + np.random.randn(100, 1)
# Mini-Batch Gradient Descent parameters
learning_rate = 0.01
batch_size = 10
epochs = 50
theta = np.random.randn(2, 1)
# Mini-Batch Gradient Descent algorithm
for epoch in range(epochs):
    # Shuffle the data
    indices = np.random.permutation(len(X))
    X_shuffled = X[indices]
    y_shuffled = y[indices]

    for i in range(0, len(X), batch_size):
        X_batch = X_shuffled[i:i+batch_size]
        y_batch = y_shuffled[i:i+batch_size]
        # Add bias term to input features
        X_batch = np.c_[np.ones((X_batch.shape[0], 1)), X_batch]

        # Compute gradients
        gradients = -2 * X_batch.T.dot(y_batch - X_batch.dot(theta))

        # Update parameters
        theta -= learning_rate * gradients / batch_size
        cost = np.mean((X_batch.dot(theta) - y_batch) ** 2)
    print(f"Epoch {epoch+1}/{epochs}, Cost: {cost}")
print("Final theta:", theta)
```

Output:


37_MrunmayeeNaik-DL-Exp03.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
0s
Epoch 17/50, Cost: 1.4277253262784106
Epoch 18/50, Cost: 1.2923317915686847
Epoch 19/50, Cost: 1.504002570264228
Epoch 20/50, Cost: 1.170312728838026
Epoch 21/50, Cost: 0.8834914978395547
Epoch 22/50, Cost: 1.262434743417455
Epoch 23/50, Cost: 0.7858854320533043
Epoch 24/50, Cost: 0.9150614173716359
Epoch 25/50, Cost: 0.8460100103621431
Epoch 26/50, Cost: 0.6955913531598046
Epoch 27/50, Cost: 1.0435875401057906
Epoch 28/50, Cost: 1.280518602812121
Epoch 29/50, Cost: 0.9798041629568306
Epoch 30/50, Cost: 1.1487426320016876
Epoch 31/50, Cost: 1.103305954405069
Epoch 32/50, Cost: 1.0346185386768763
Epoch 33/50, Cost: 1.2109435146542946
Epoch 34/50, Cost: 0.6714502703006819
Epoch 35/50, Cost: 0.7579931739090652
Epoch 36/50, Cost: 1.1853600506067206
Epoch 37/50, Cost: 1.2738149542358042
Epoch 38/50, Cost: 1.2705337764218005
Epoch 39/50, Cost: 0.9801683107884053
Epoch 40/50, Cost: 1.4079224051850558
Epoch 41/50, Cost: 0.6121913855675938
Epoch 42/50, Cost: 0.9815726744666563
Epoch 43/50, Cost: 0.6778701032205802
Epoch 44/50, Cost: 1.1030693994149794
Epoch 45/50, Cost: 1.323336545801436
Epoch 46/50, Cost: 1.8298175310623943
Epoch 47/50, Cost: 1.3022107773433447
Epoch 48/50, Cost: 1.0106291175523805
Epoch 49/50, Cost: 0.6663318894888752
Epoch 50/50, Cost: 1.39835380966929
Final theta: [[4.14234886]
[3.03882785]]

Adam Learning Gradient Descent

Code:

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import train_test_split
np.random.seed(0)
X = 2 * np.random.rand(100, 1)
y = 4 + 3 * X + np.random.randn(100, 1)
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Create a sequential model
model = Sequential()
model.add(Dense(1, input_dim=1))  # Linear regression with one input and one
output

# Compile the model with Adam optimizer
optimizer = Adam(learning_rate=0.01)
model.compile(loss='mean_squared_error', optimizer=optimizer)

# Train the model
model.fit(X_train, y_train, epochs=100, batch_size=10, verbose=0)

# Evaluate the model on test data
loss = model.evaluate(X_test, y_test, verbose=0)
print("Test loss:", loss)
```

Output:

```
Test loss: 1.0365097522735596
```