

```
# Fetch batch function:
```

```
def fetch_batch(epoch, batch_index, batch_size):  
    return X_batch, y_batch
```

```
# Set up computational graph:
```

```
import tensorflow as tf  
reset_graph ()
```

```
n_epochs = 1000  
learning_rate = 0.01
```

```
X = tf.constant(scaled_housing_data_plus_bias, dtype=tf.float32, name="X")  
y = tf.constant(housing_data_target, dtype=tf.float32, name="y")
```

```
theta = tf.Variable(tf.random_uniform([n + 1, 1], -1.0, 1.0), name="theta")  
y_pred = tf.matmul(X, theta, name="predictions")  
error = y_pred - y  
mse = tf.reduce_mean(tf.square(error), name="mse")  
optimizer = tf.train.GradientDescentOptimizer(learning_rate)  
training_op = optimizer.minimize(mse)
```

```
# Execute:
```

```
init = tf.global_variables_initializer()
```

```
with  
tf.Session() as sess:  
    sess.run(init)  
    for epoch in range(n_epochs):  
        if epoch % 100 == 0:  
            print("Epoch", epoch, "MSE=", mse.eval()) sess.run(training_op)  
    best_theta = theta.eval()
```