Mini Batch Gradient Descent

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

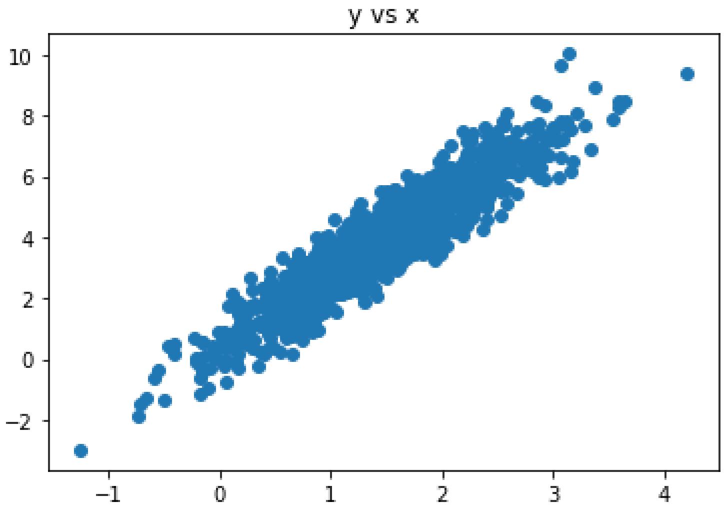
```
w1 = 2.4
x = np.random.normal(1.5, 0.8, 1000)
e = np.random.normal(0.4, 0.7, 1000)
y_actual = w1*x + e
```

In [3]:

```
plt.scatter(x, y_actual); plt.title('y vs x')
```

Out[3]:

```
Text(0.5, 1.0, 'y vs x')
```



In [4]:

```
index = np.random.randint(0, len(x), 10)
x[index]
```

Out[4]:

```
array([1.70063427, 1.18500634, 1.94343517, 1.54663912, 1.66992415,
       1.55011312, 1.9998682 , 1.36914797, 2.06099662, 0.89234036])
```

In [5]:

```
m = 0.2
c = 0.4
cost_list = []
para_list = []
epochs = 1000
n = len(x)
batch_size = 50
batches = n/batch_size
lr = 0.01
for epoch in range(epochs):
    cost = 0
    d_m  = 0
    d_c = 0
    for i in range(int(batches)):
        index = np.random.randint(0, len(x), batch_size)
        x_sub = x[index]
        y_actual_sub = y_actual[index]
        y_pred = m*x_sub + c

        cost = -1/n * sum(np.power(y_actual_sub - y_pred, 2))

        d_m += -2/n * (np.dot(x_sub, (y_actual_sub - y_pred)))
        d_c += -2/n * sum(y_actual_sub - y_pred)
    m = m - lr*d_m
    c = c - lr*d_c
    para_list.append((m,c))
    cost_list.append(cost)
```

In [6]:

```
pos = cost_list.index(min(cost_list, key = abs))
m = para_list[pos][0]
c = para_list[pos][1]
cost = cost_list[pos]
```

In [7]:

```
m,c, cost
```

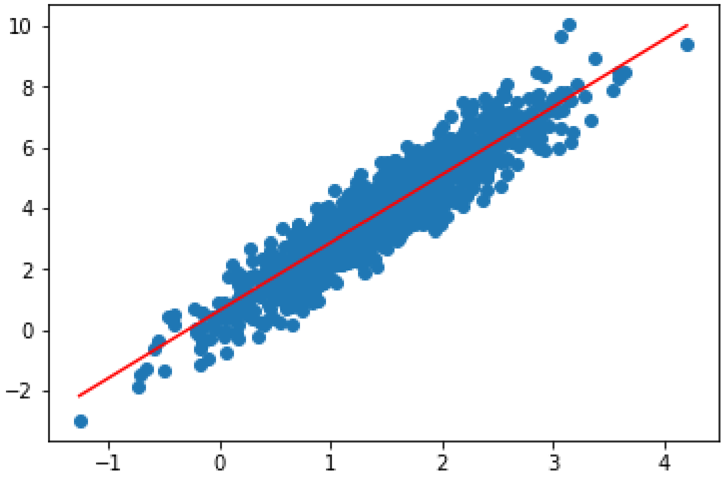
Out[7]:

```
(2.232881117425663, 0.6300818401604531, -0.01098220836404888)
```

In [8]:

```
Y_pred = m*x + c

plt.scatter(x, y_actual)
plt.plot([min(x), max(x)], [min(Y_pred), max(Y_pred)], color='red')  # regression line
plt.show()
```


10
8
6
4
2
0
−2
−1
0
1
2
3
4