Neural Networks - Prolog 1 - Linear Regression

September 13, 2016

Marcin Junczys-Dowmunt Machine Translation Marathon 2016

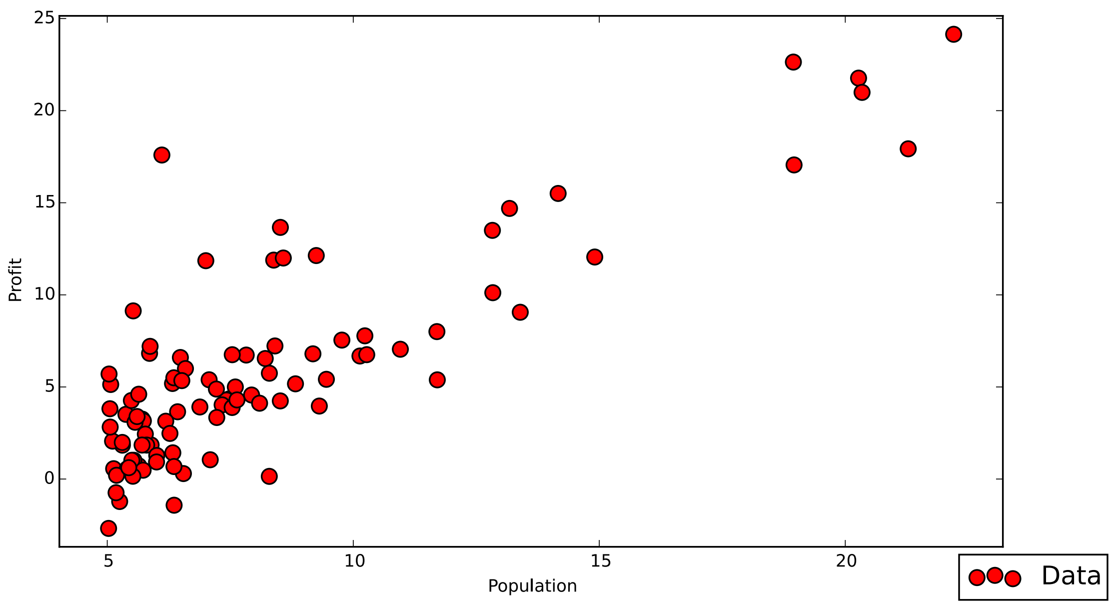
1 Introduction to Neural Networks

1.1 Prolog 1: Linear regression

1.1.1 Problem: Predict profit based on city's population size

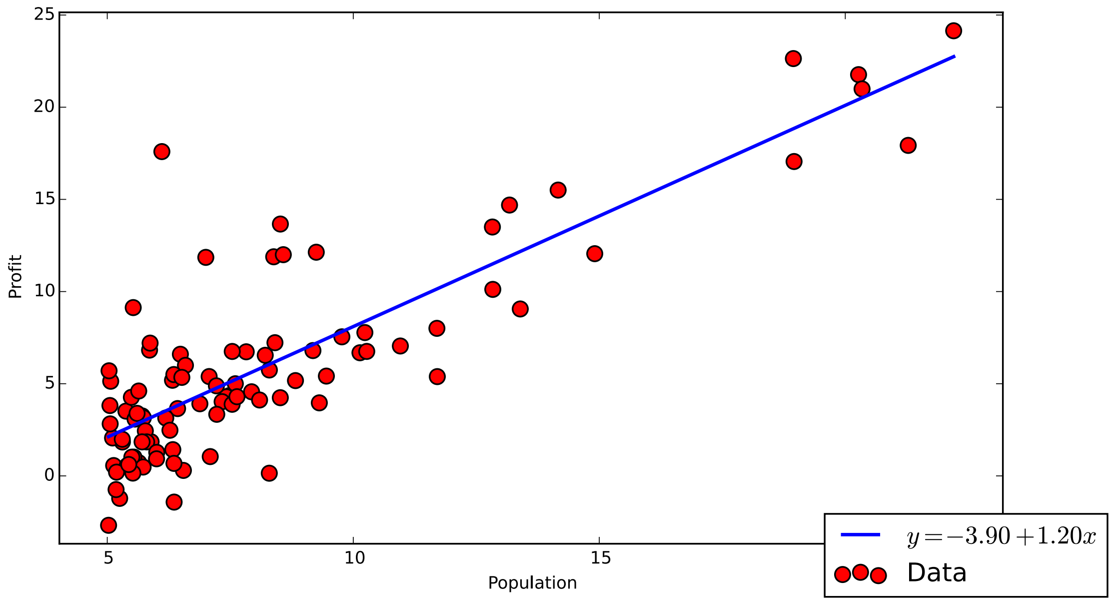
In [14]:
```
import csv
reader = csv.reader(open("ex1data1.txt"), delimiter=",")

x = list()
y = list()
for xi, yi in reader:
    x.append(float(xi))
    y.append(float(yi))

print("x = ", x[:10])
print("y = ", y[:10])
```

```
x =  [6.1101, 5.5277, 8.5186, 7.0032, 5.8598, 8.3829, 7.4764, 8.5781, 6.4862, 5.0546]
y =  [17.592, 9.1302, 13.662, 11.854, 6.8233, 11.886, 4.3483, 12.0, 6.5987, 3.8166]
```



$$\text{Parameters:} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} \quad \text{Model:} \quad h_\theta(x) = \theta_0 + \theta_1 x$$

```
In [16]: def h(theta, x):
             return theta[0] + theta[1]*x
```



1.2 How can this be done better?

1.3 Optimization Criterion: The Cost Function $J(\theta)$

We try to find $\hat{\theta}$ such that the cost function $J(\theta)$ is minimal:

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} \in \mathbb{R}^2 \quad J : \mathbb{R}^{\nvDash} \to \mathbb{R}$$

$$\hat{\theta} = \underset{\theta \in \mathbb{R}^2}{\arg\min}\, J(\theta)$$

1.4 Mean Square Error

$$\begin{aligned} J(\theta) &= \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2 \\ &= \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_0 + \theta_1 x^{(i)} - y^{(i)} \right)^2 \end{aligned}$$

where m is the number of data points in the training set.

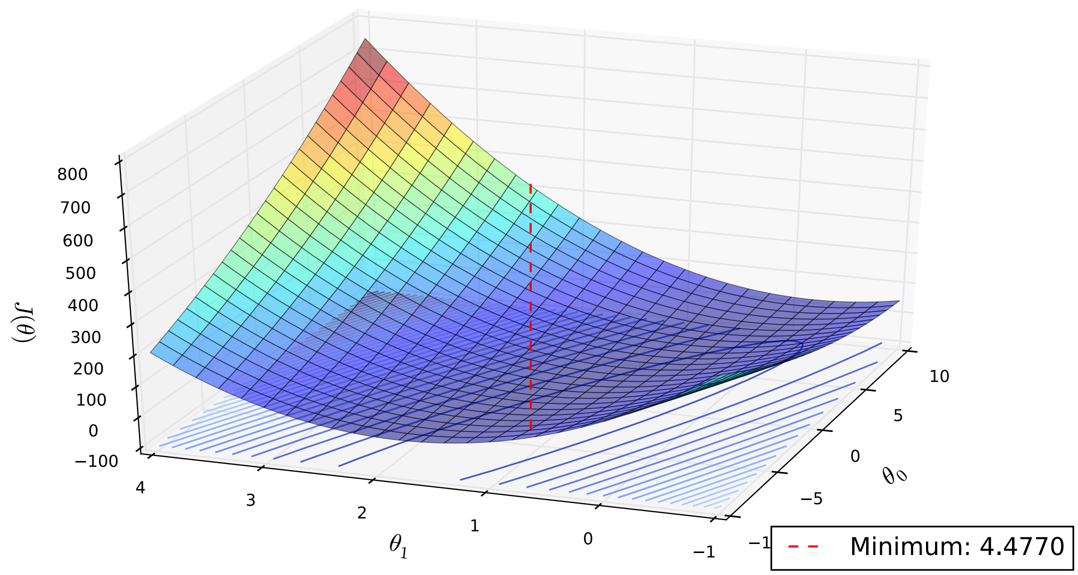
```
In [18]: def J(h, theta, x, y):
             m = len(y)
             return (1.0/(2*m) * sum((h(theta, x[i]) - y[i])**2
```

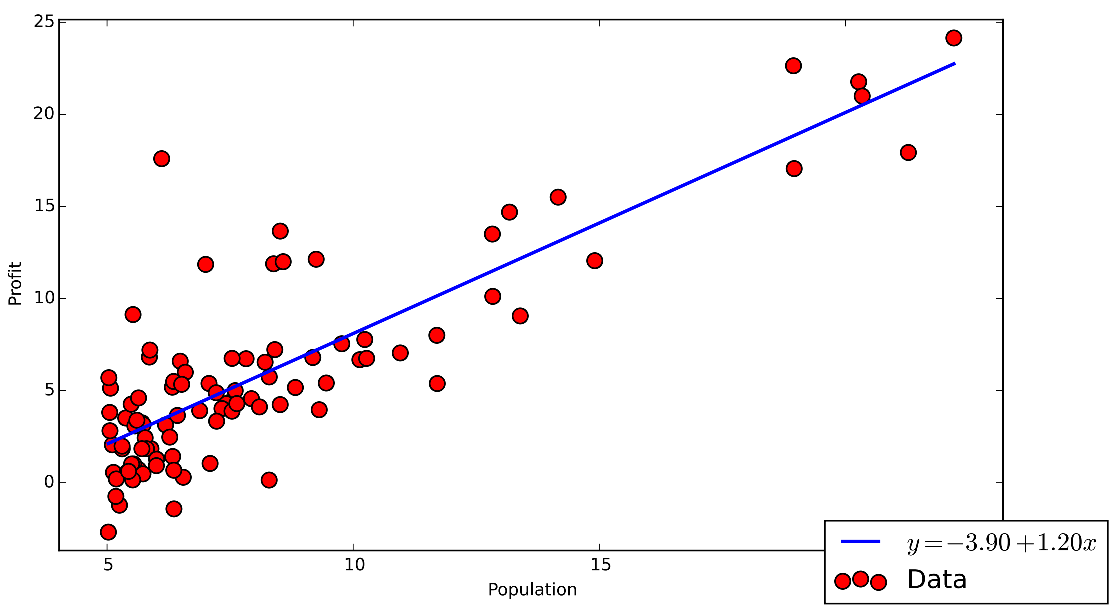
```
for i in range(m)))

display(Math(r"\Large J(\theta) = %.4f" % J(h,[ , 0],x,y)))
```

$$J(\theta) = 32.0727$$

1.5 The error surface $J(\theta)$

- $J(\theta)$ forms a convex surface with a single minimum.
- That means it is easy to find that minimum.





2 So, how do we find $\underset{\theta \in \mathbb{R}^2}{\arg\min}\, J(\theta)$ computationally?

2.1 Method 1: Normal Matrix

Linear regression has a closed form formula for calculating the optimal θ that minimizes $J(\theta)$:

$$\theta = (X^T X)^{-1} X^T \vec{y}$$

where

$$X = \begin{bmatrix} 1 & x_1^{(1)} & \cdots & x_n^{(1)} \\ 1 & x_1^{(2)} & \cdots & x_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \cdots & x_n^{(m)} \end{bmatrix}$$

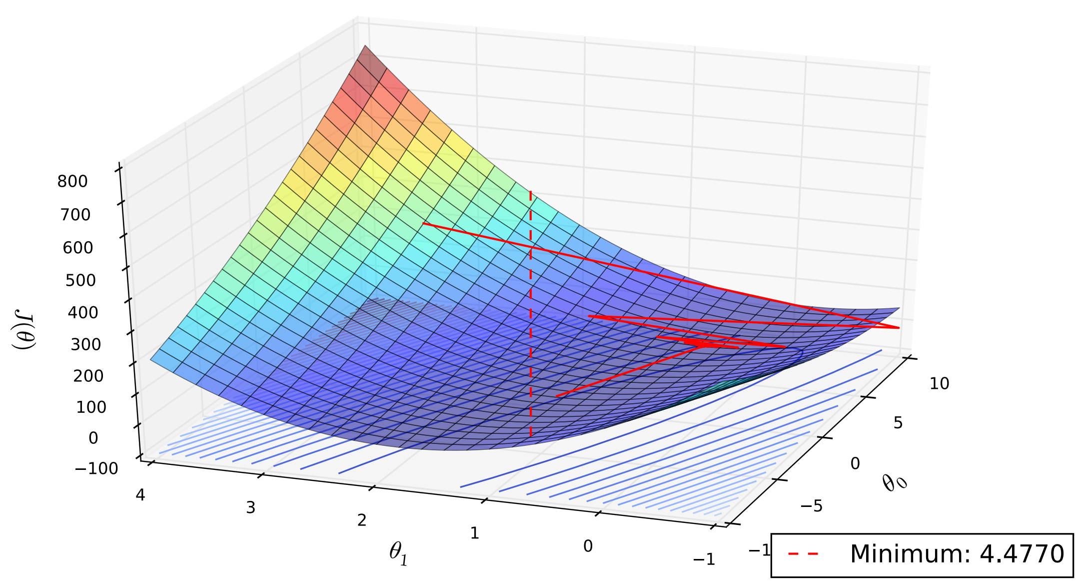
$$\theta = \begin{bmatrix} -3.8958 \\ 1.1930 \end{bmatrix}$$

2.2 Method 2: Gradient Descent

2.3 Gradient descent

2.3.1 Update rule for parameter θ_j

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \text{ for each } j$$



2.4 How do we calculate $\frac{\partial}{\partial \theta_j} J(\theta)$?

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \\
&= 2 \cdot \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x^{(i)}) - y^{(i)}) \\
&= \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot \frac{\partial}{\partial \theta_j} \sum_{i=0}^{n} \theta_i x_i^{(i)} \\
&= \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)}
\end{aligned}$$

2.5 The update rule once again

For linear regression we have the following model:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

and we repeat until convergence (θ_0 and θ_1 should be updated simultaneously):

$$\begin{aligned}
\theta_0 &:= \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \\
\theta_1 &:= \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x^{(i)}
\end{aligned}$$

$$\text{Result:} \quad \theta = \begin{bmatrix} -3.8958 \\ 1.1930 \end{bmatrix} \quad J(\theta) = 4.4770 \quad \text{after 7630 iterations}$$

```
In [58]: fig = regdots(x, y)
         thetaBest, errors = GD(h, J, [0,0], x, y, alpha=0.001, eps=0.0)
         regline(fig, h, thetaBest, x)
         legend(fig)
```