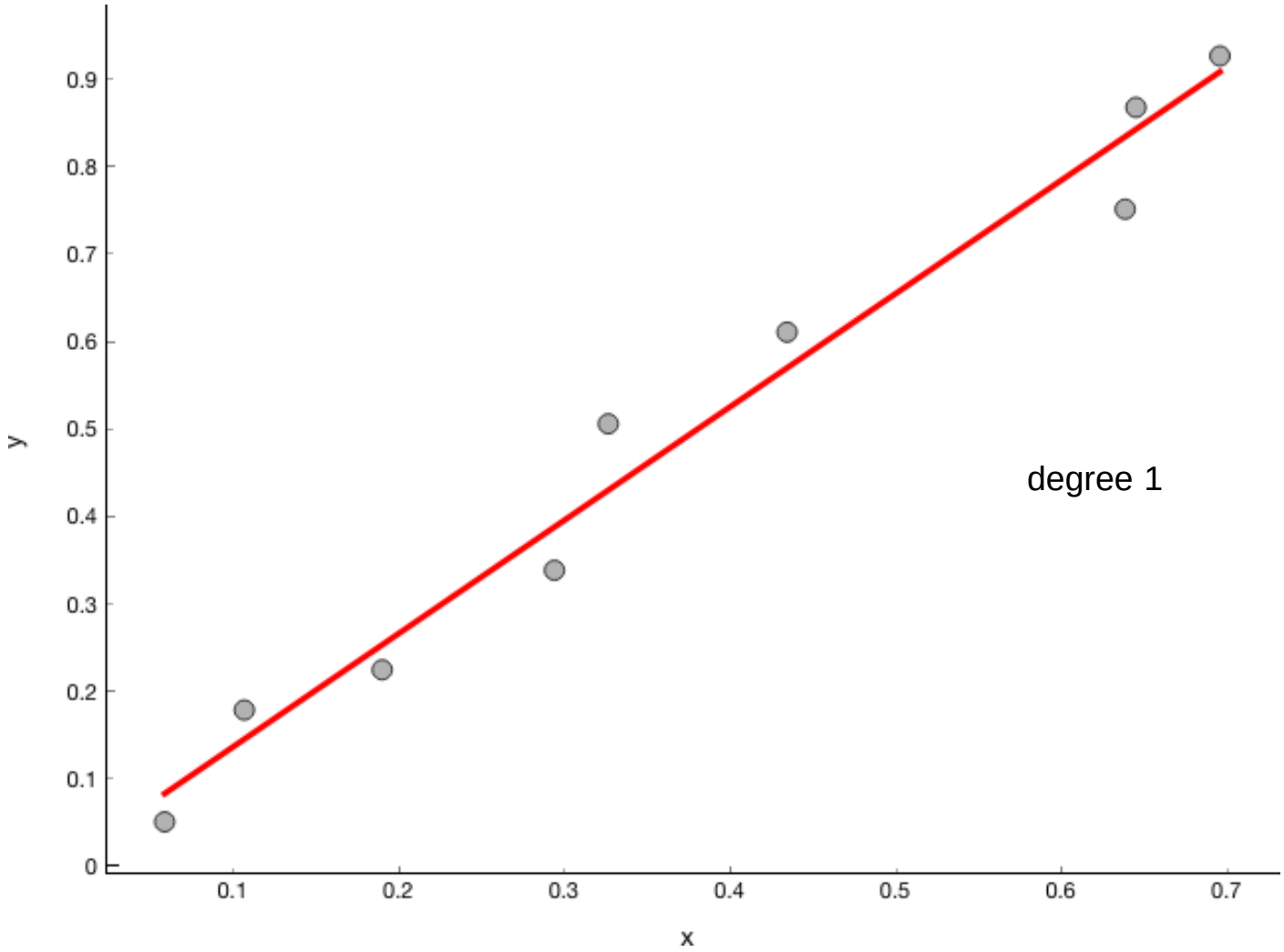
Linear regression

Regression is optimization

- Find a line of "best fit"
- Best fit implies optimization



Task in 2 dimensions

- Given data **D**, learn a function $h : X \rightarrow Y$
- Model form: $h(x) = mx+b$
  - Alternate form: $h_w(x) = w_1x+w_0$
- Determining m, b from data is called learning

Optimization

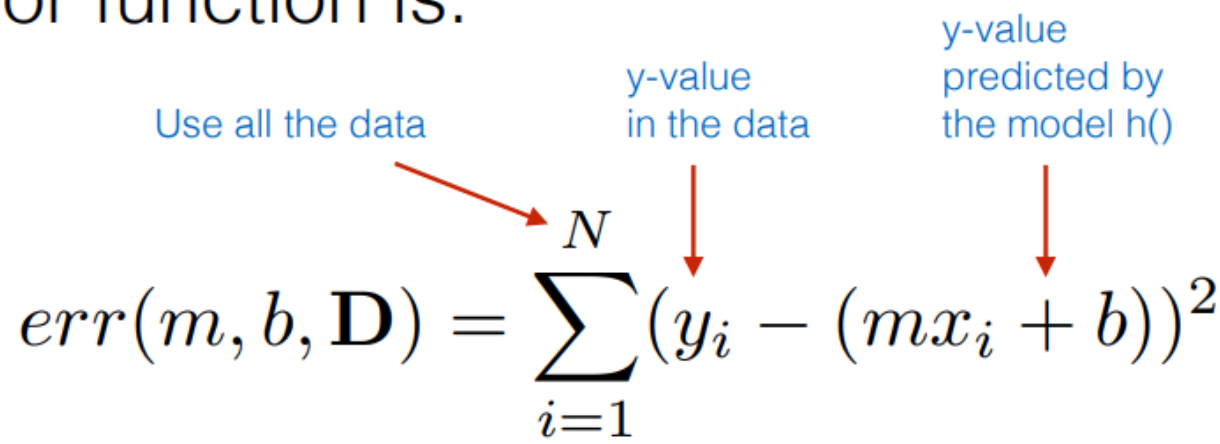
- The cost function is the error or difference between the predicted value and the true value.

- The error function is:

$$err(m, b, \mathbf{D}) = \sum_{i=1}^{N} (y_i - (mx_i + b))^2$$

Optimization by Calculus

- The error function is:



For a given data set **D**, find *m,b* to minimize *error*.

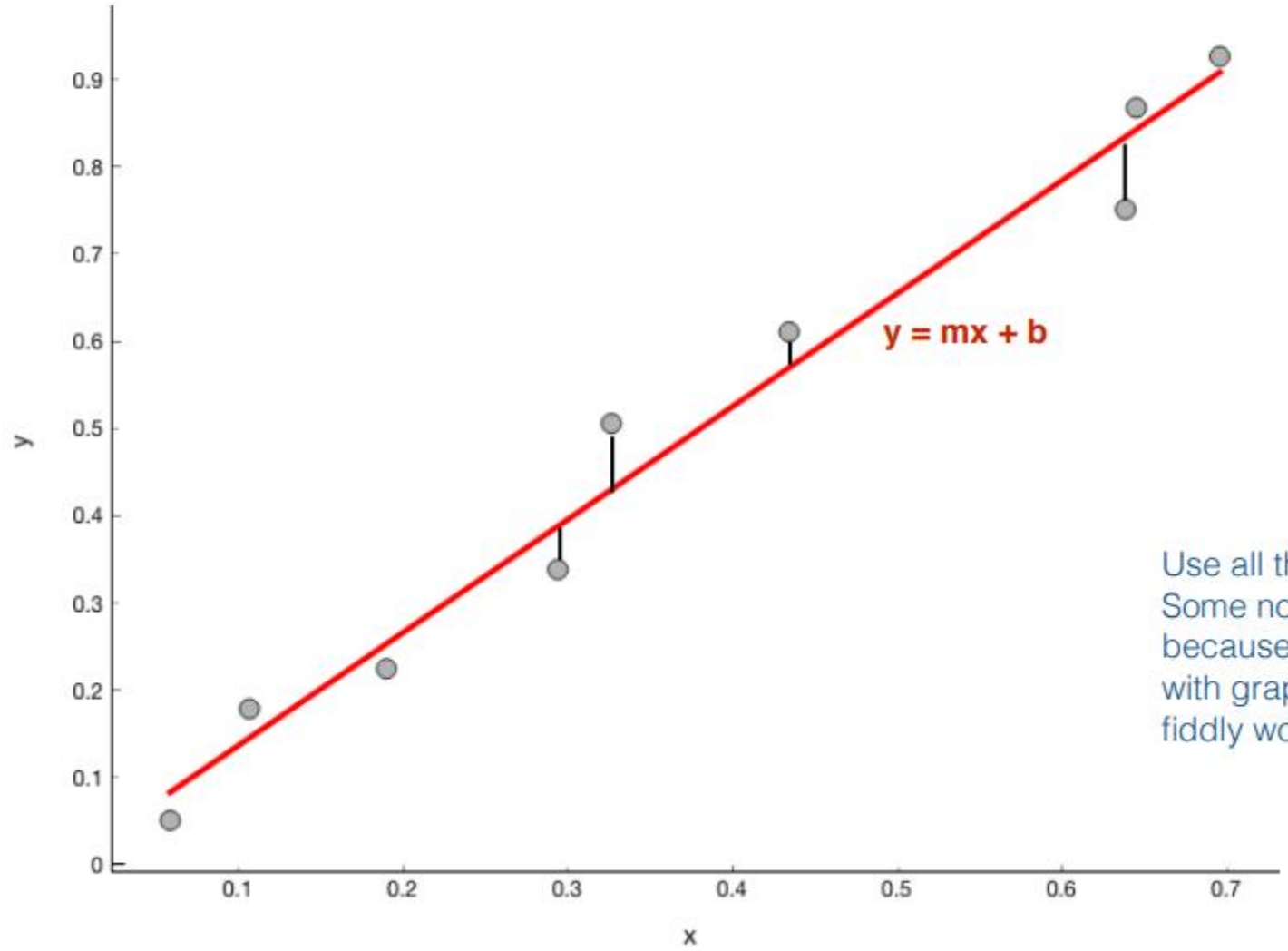
- Note: **D** = $\{(x_1, y_1), \ldots, (x_N, y_N)\}$

Gradient Descent

- A linear regression model can be trained using the optimization algorithm gradient descent by iteratively modifying the model's parameters to reduce the mean squared error (MSE) of the model on a training dataset.

- The idea is to start with random m and b values and then iteratively update the values, reaching minimum cost.

$$err(m, b, \mathbf{D}) = \sum_{i=1}^{N} (y_i - (mx_i + b))^2$$



Use all the data. Some not shown because playing with graphics is fiddly work.

Optimization by Calculus

- The error function is:

$$err(m, b, \mathbf{D}) = \sum_{i=1}^{N} (y_i - (mx_i + b))^2$$

For a given data set **D**, find *m,b* to minimize *error*.

- Very calculus friendly!
  - Solve: $\frac{\partial}{\partial m} err(m, b, \mathbf{D}) = 0$

$$\frac{\partial}{\partial b} err(m, b, \mathbf{D}) = 0$$

A gradient is nothing but a derivative that defines the effects on outputs of the function with a little bit of variation in inputs.

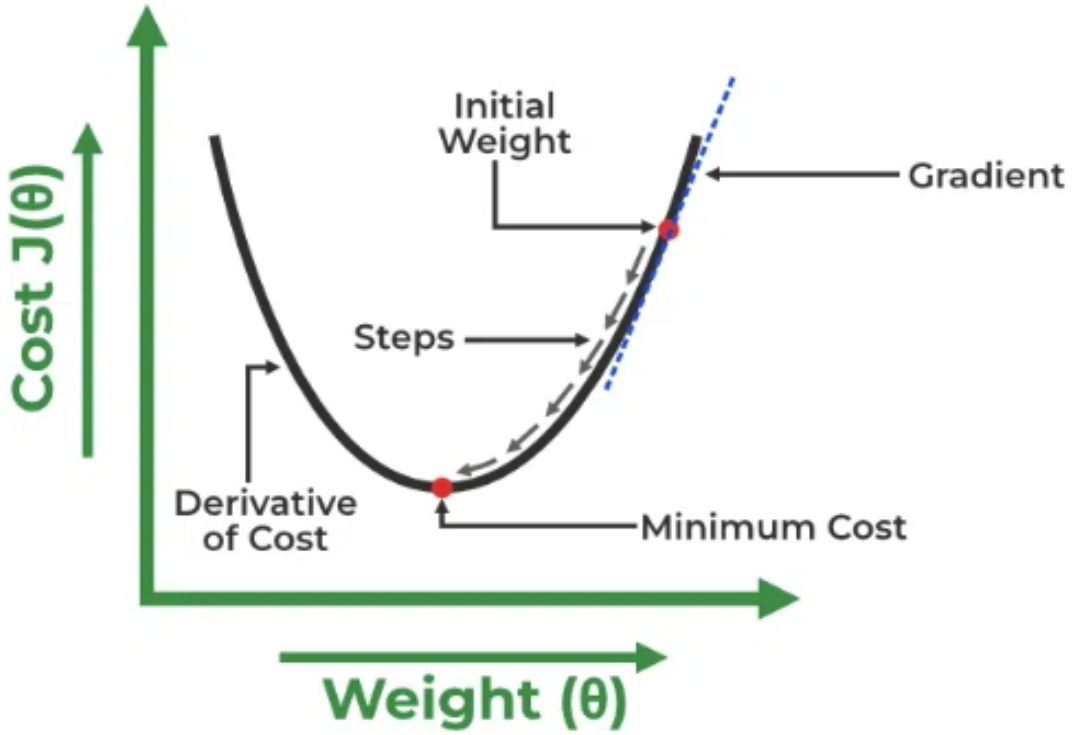
Let's differentiate the cost function(J) with respect to θ_1

$$
\begin{aligned}
J'_{\theta_1} &= \frac{\partial J(\theta_1, \theta_2)}{\partial \theta_1} \\
&= \frac{\partial}{\partial \theta_1}\left[\frac{1}{n}\left(\sum_{i=1}^{n}(\hat{y}_i - y_i)^2\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\left(\frac{\partial}{\partial \theta_1}(\hat{y}_i - y_i)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\left(\frac{\partial}{\partial \theta_1}(\theta_1 + \theta_2 x_i - y_i)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\,(1 + 0 - 0)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} (\hat{y}_i - y_i)\,(2)\right] \\
&= \frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)
\end{aligned}
$$

Let's differentiate the cost function(J) with respect to θ_2

$$
\begin{aligned}
J'_{\theta_2} &= \frac{\partial J(\theta_1, \theta_2)}{\partial \theta_2} \\
&= \frac{\partial}{\partial \theta_2}\left[\frac{1}{n}\left(\sum_{i=1}^{n}(\hat{y}_i - y_i)^2\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\left(\frac{\partial}{\partial \theta_2}(\hat{y}_i - y_i)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\left(\frac{\partial}{\partial \theta_2}(\theta_1 + \theta_2 x_i - y_i)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\,(0 + x_i - 0)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} (\hat{y}_i - y_i)\,(2x_i)\right] \\
&= \frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)\cdot x_i
\end{aligned}
$$

Finding the coefficients of a linear equation that best fits the training data is the objective of linear regression. By moving in the direction of the Mean Squared Error negative gradient with respect to the coefficients, the coefficients can be changed. And the respective intercept and coefficient of X will be if α is the learning rate.



Gradient Descent

Generalizing

- Assume **D** has n dimensional $\mathbf{x} = (x_i, \ldots, x_n)$
- Calculus & Linear Algebra can minimize squared error

$$\mathbf{w} = \mathbf{w} - \alpha \sum_{i=1}^{N} \nabla h(\mathbf{x})$$

Generalizing

- Assume **D** has n dimensional **x** = (x_i, …, x_n)
- Calculus & Linear Algebra can minimize squared error

$$\mathbf{w} = \mathbf{w} - \alpha \sum_{i=1}^{N} \nabla h(\mathbf{x})$$

$$\begin{aligned}
\theta_1 &= \theta_1 - \alpha \left(J'_{\theta_1}\right) \\
&= \theta_1 - \alpha \left(\frac{2}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)\right) \\
\theta_2 &= \theta_2 - \alpha \left(J'_{\theta_2}\right) \\
&= \theta_2 - \alpha \left(\frac{2}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i) \cdot x_i\right)
\end{aligned}$$