ML for Neuroimaging

St. line Fit - Linear Regression - Peceptron

Terminologies in ML

Task – ?	Model - ?	Distribution ?	Supervised
Datasets – ?	Training ?	Training Error ?	Unsupervised
Features - ?	Loss function ?	Testing Error ?	Self supervised
Training Data – ?	Cost function ?	Visualization ?	Semi supervised
Validation Data - ?	Ensemble ?	T-SNE Plot ?	Reinforcement learning
Testing Data – ?	Cross Entropy ?	ROC curves?	Data-driven
Performance ?	Accuracy ?	Gradient Descent ?	

Supervised Learning paradigm

Metaphor: Credit approval

Applicant information:

age	23 years
gender	male
annual salary	\$30,000
years in residence	1 year
years in job	1 year
current debt	\$15,000
...	...

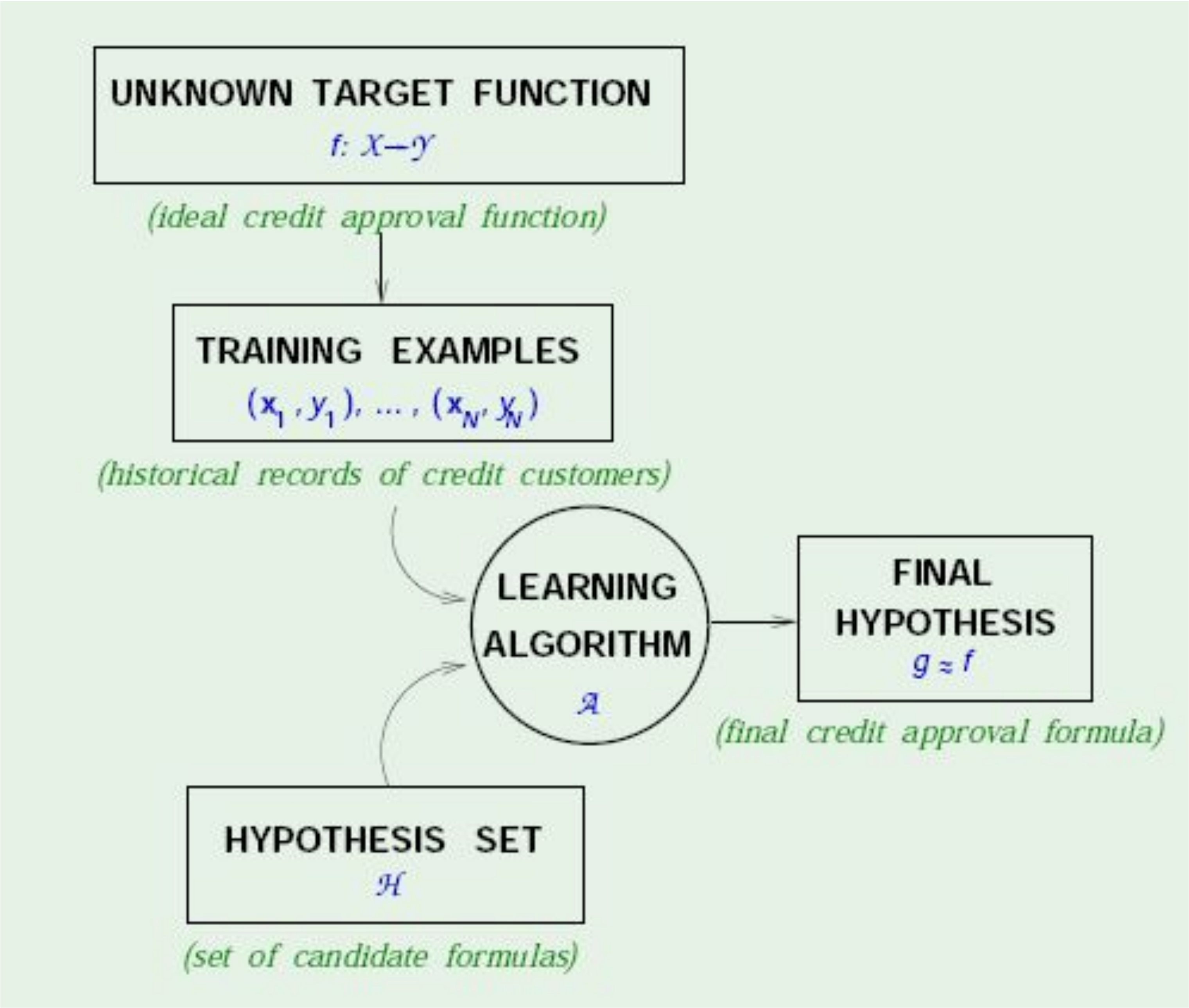
Approve credit?

Formalization:

- Input: $\mathbf{x}$ (*customer application*)
- Output: y (*good/bad customer?*)
- Target function: $f : \mathcal{X} \to \mathcal{Y}$ (*ideal credit approval formula*)
- Data: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$ (*historical records*)

$$\downarrow \quad \downarrow \quad \downarrow$$

- Hypothesis: $g : \mathcal{X} \to \mathcal{Y}$ (*formula to be used*)


UNKNOWN TARGET FUNCTION
$f: \mathcal{X} \to \mathcal{Y}$
(ideal credit approval function)
TRAINING EXAMPLES
$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
(historical records of credit customers)
LEARNING ALGORITHM
$\mathcal{A}$
FINAL HYPOTHESIS
$g \approx f$
(final credit approval formula)
HYPOTHESIS SET
$\mathcal{H}$
(set of candidate formulas)

Solution components

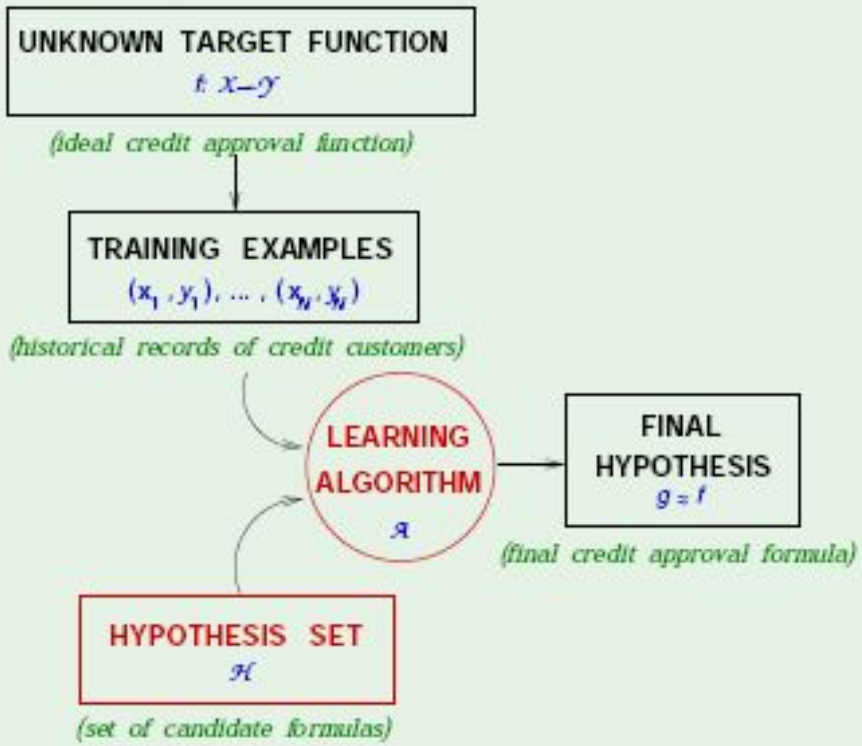
The 2 solution components of the learning problem:

- The Hypothesis Set

  $\mathcal{H} = \{h\} \qquad g \in \mathcal{H}$

- The Learning Algorithm

Together, they are referred to as the *learning model*.



Linear Models: Regression, Classification

Calculus Recap

Given a function, what does the derivative say ??

How to minimize a function ?
2 approaches : (i) Closed form (ii) Iterative

What is constrained ? Unconstrained minimization ?

Examples

- Say, we want to minimize

  $y = 4x^2 + 2$

  Plot the function :

  $dy/dx = 8x$

  Put $dy/dx = 0$, it implies $x = 0$

  So function is minimum at $x = 0$

  This is the closed form (analytical) way of solving

Examples

- Say, we want to minimize

  $y = 4(x-5)^2 + 2$

  Plot the function :

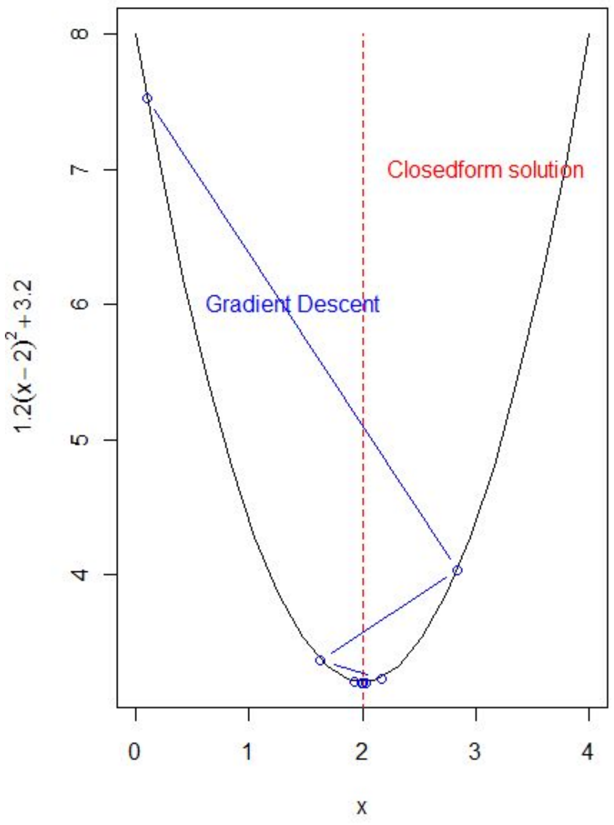
  $dy/dx = 8(x-5)$

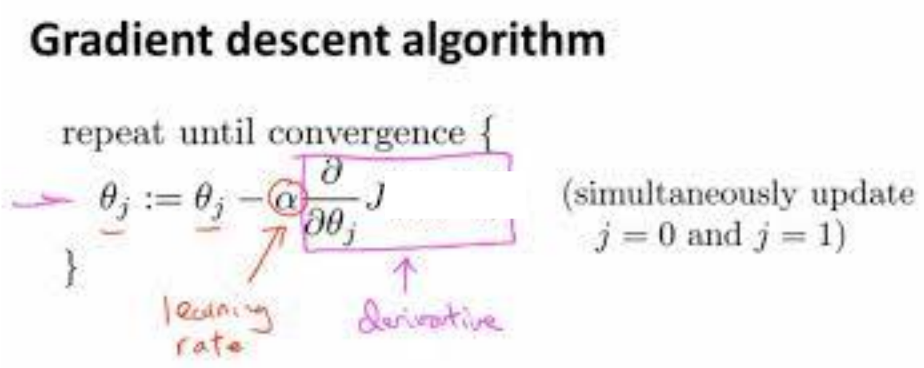
  Put $dy/dx = 0$, it implies $x = 5$

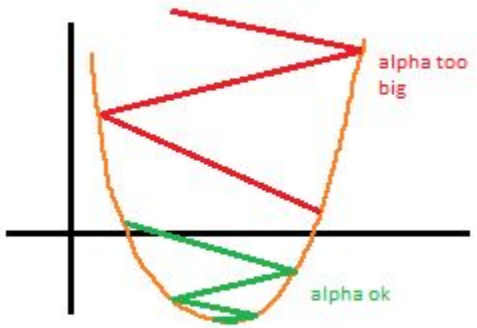
  So function is minimum at $x = 5$

  This is the closed form (analytical) way of solving

Numerical solution: Iterative minimization







Iterative Minimization

- $y = 4(x-5)^2 + 2$
- $dy/dx = 8(x-5)$

- Say x_old = 5.5
- X_new = x_old - alpha (dy/dx)|at x = x_old
- $X\_new^{(1)} = 5.5 - 0.1(8*(5.5-5)) = 5.5 – 0.4 = 5.1$

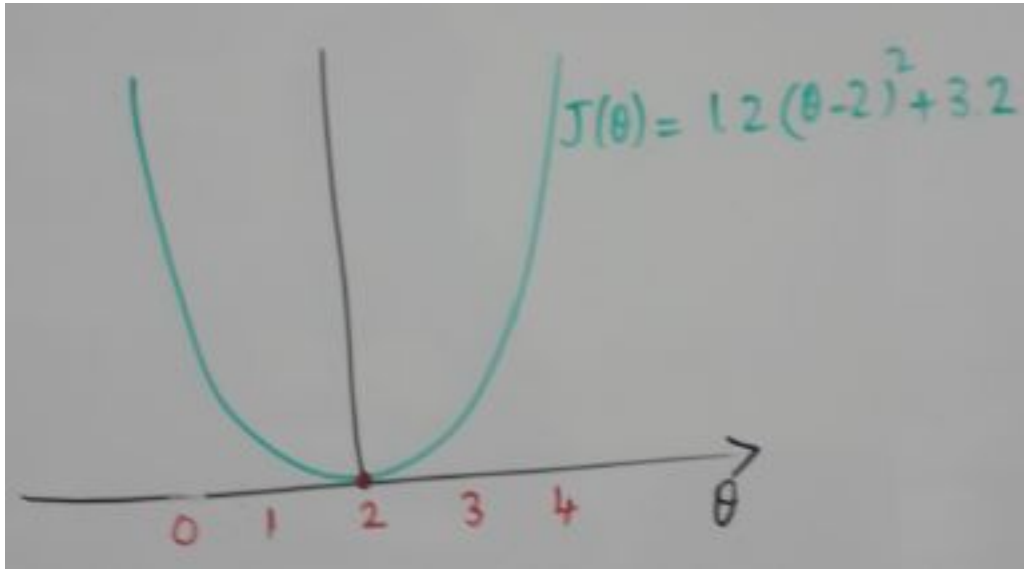
- $X\_new^{(2)} = 5.1 - 0.1(8*(5.1-5)) = 5.1 - 0.08 = 5.02$

Quick exercise

- $E = 1.2(x-2)^2 + 3.2$
- $x_{opt} = 2$

- Work out the closed-form solution
- Work out $x^{(1)}$, $x^{(2)}$ results of two steps of gradient descent

Example for Iterative method



$$J(\theta) = 1.2(\theta-2)^2 + 3.2$$

$$J'(\theta) = 2.4(\theta-2)$$

$$\theta(new) = \theta(old) + \Delta\theta$$

$$\Delta\theta = -\mu \left.\frac{\partial J(\theta)}{\partial \theta}\right|_{\theta = \theta_{old}}$$

GRADIENT DESCENT

Impact of Learning rate

$$\theta(\text{new}) = \theta(\text{old}) + \Delta\theta$$

$$\Delta\theta = -\mu \left.\frac{\partial J(\theta)}{\partial \theta}\right|_{\theta = \theta_{old}}$$

$$\theta^0 = 1 \;;\quad J(\theta^0) = 4.4$$

$$J'(\theta) = 2.4(\theta - 2)$$

Case 1 : $\mu = 0.1$

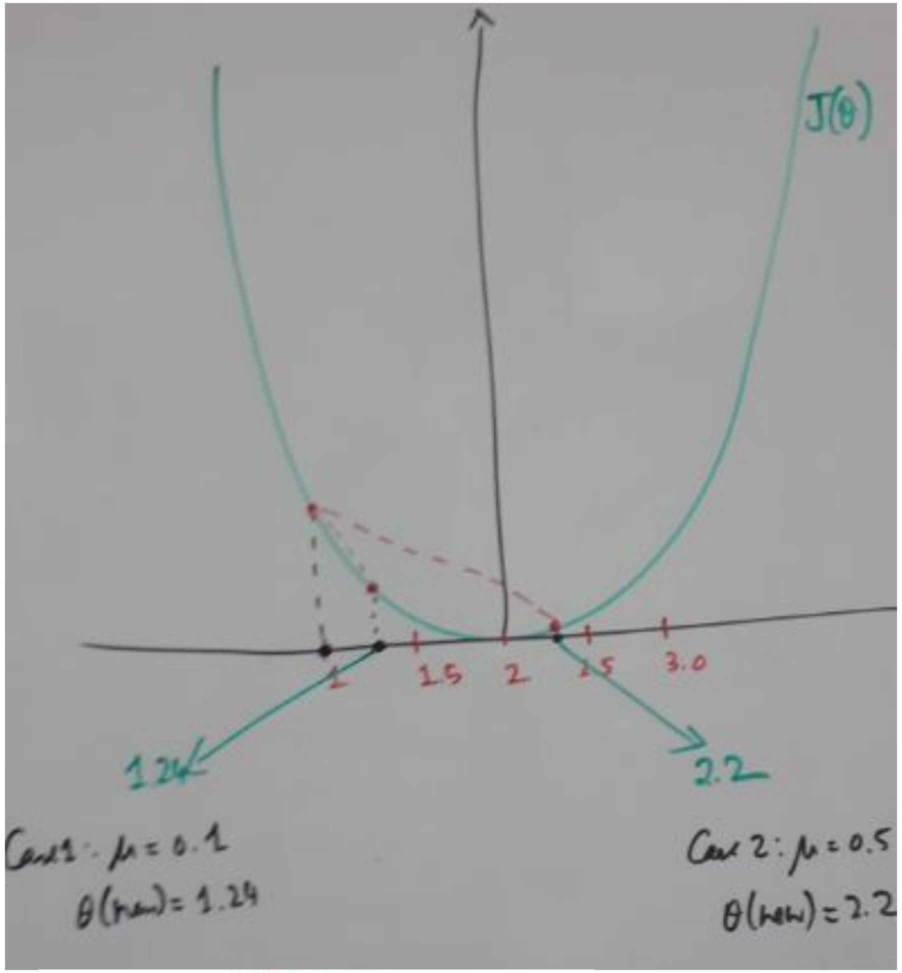
$$\theta(\text{new}) = 1 - 0.1(-2.4)$$

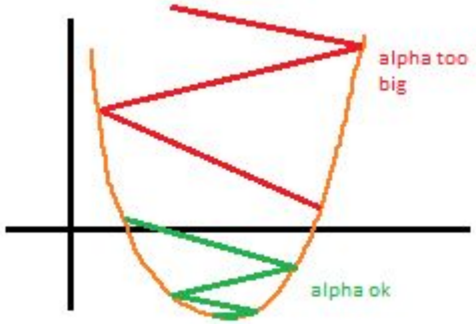
$$= 1.24$$

Case 2 : $\mu = 0.5$

$$\theta(\text{new}) = 1 - 0.5(-2.4)$$

$$= 2.2$$

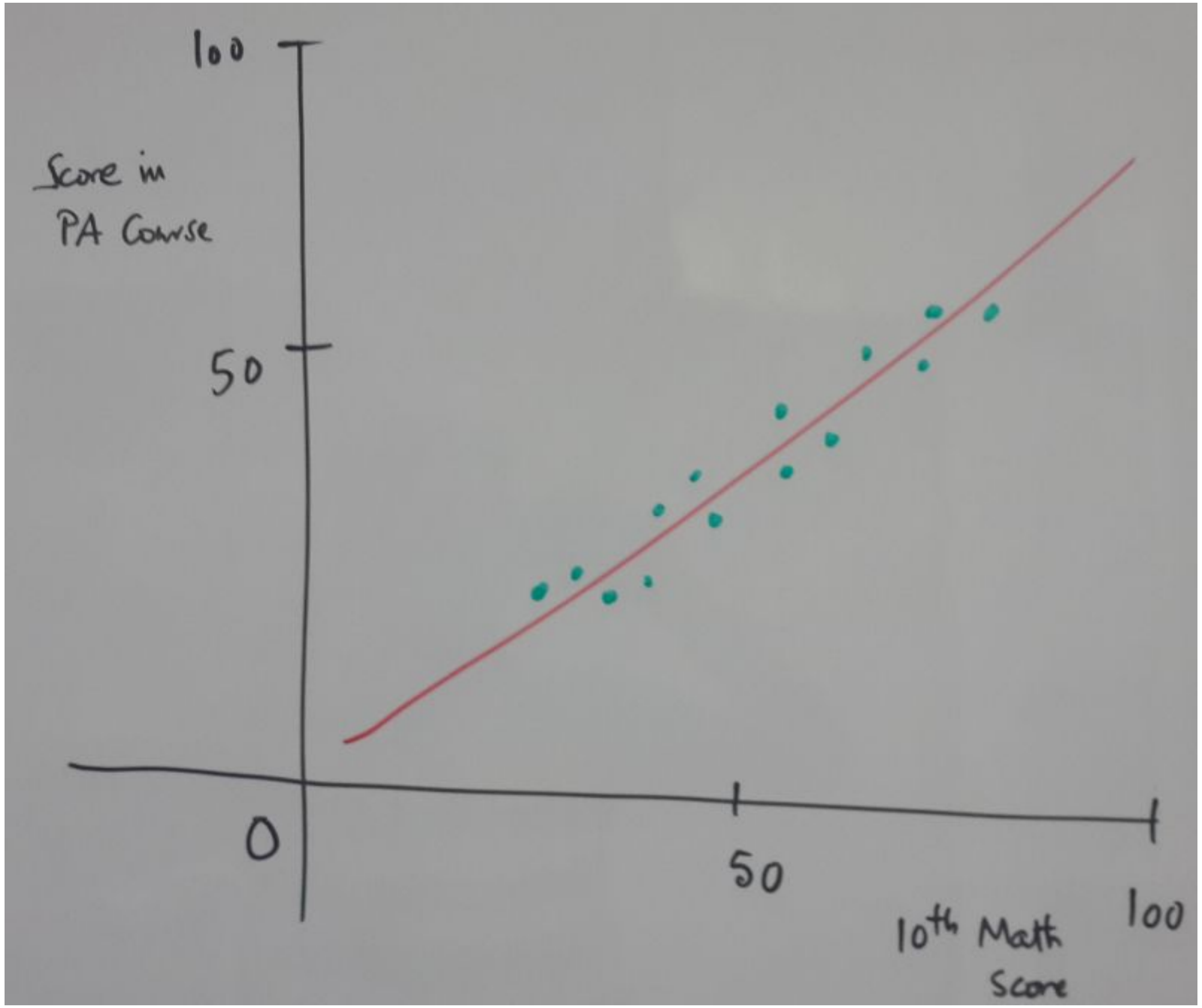




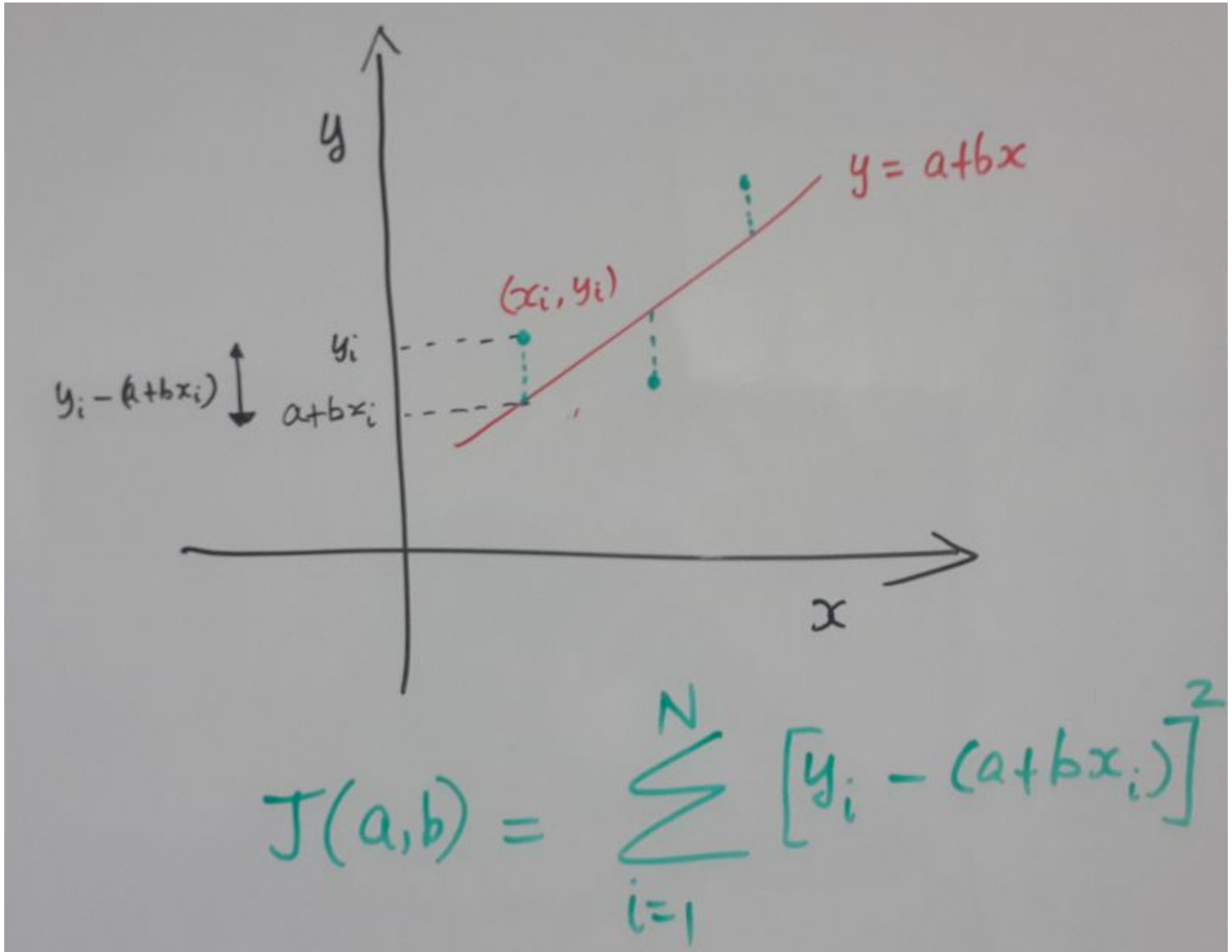
GRADIENT DESCENT

Fitting a straight line

Fitting a straight line



Fitting a straight line – Cost function



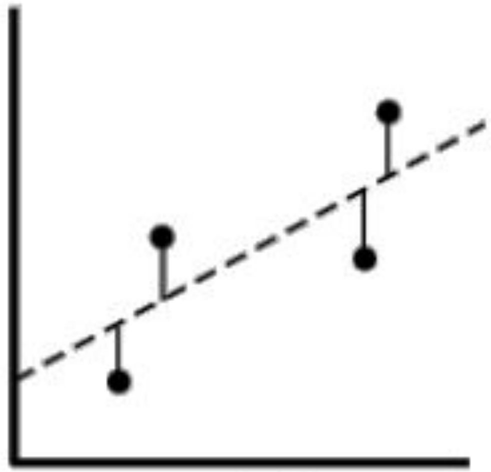
Closed form – Minimize sum of square error

$$\frac{\partial J(a,b)}{\partial a} = 0 \Rightarrow 2 \sum_{i=1}^{N} [y_i - (a+bx_i)](-1) = 0$$

$$\frac{\partial J(a,b)}{\partial b} = 0 \Rightarrow 2 \sum_{i=1}^{N} [y_i - (a+bx_i)](-x_i) = 0$$

$$n\,a + b \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

$$a \sum_{i=1}^{n} x_i + b \sum_{i=1}^{n} x_i^2 = \sum_{i=1}^{n} x_i\, y_i$$



vertical offsets

In matrix form,

$$\begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i\, y_i \end{bmatrix},$$

so

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} n & \sum_{i=1}^{n} x_i \\ \sum_{i=1}^{n} x_i & \sum_{i=1}^{n} x_i^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} x_i\, y_i \end{bmatrix}.$$

Gradient Descent – Minimize sum of square error

$$J(a,b) = \sum_{i=1}^{N} \left[y_i - (a+bx_i)\right]^2$$

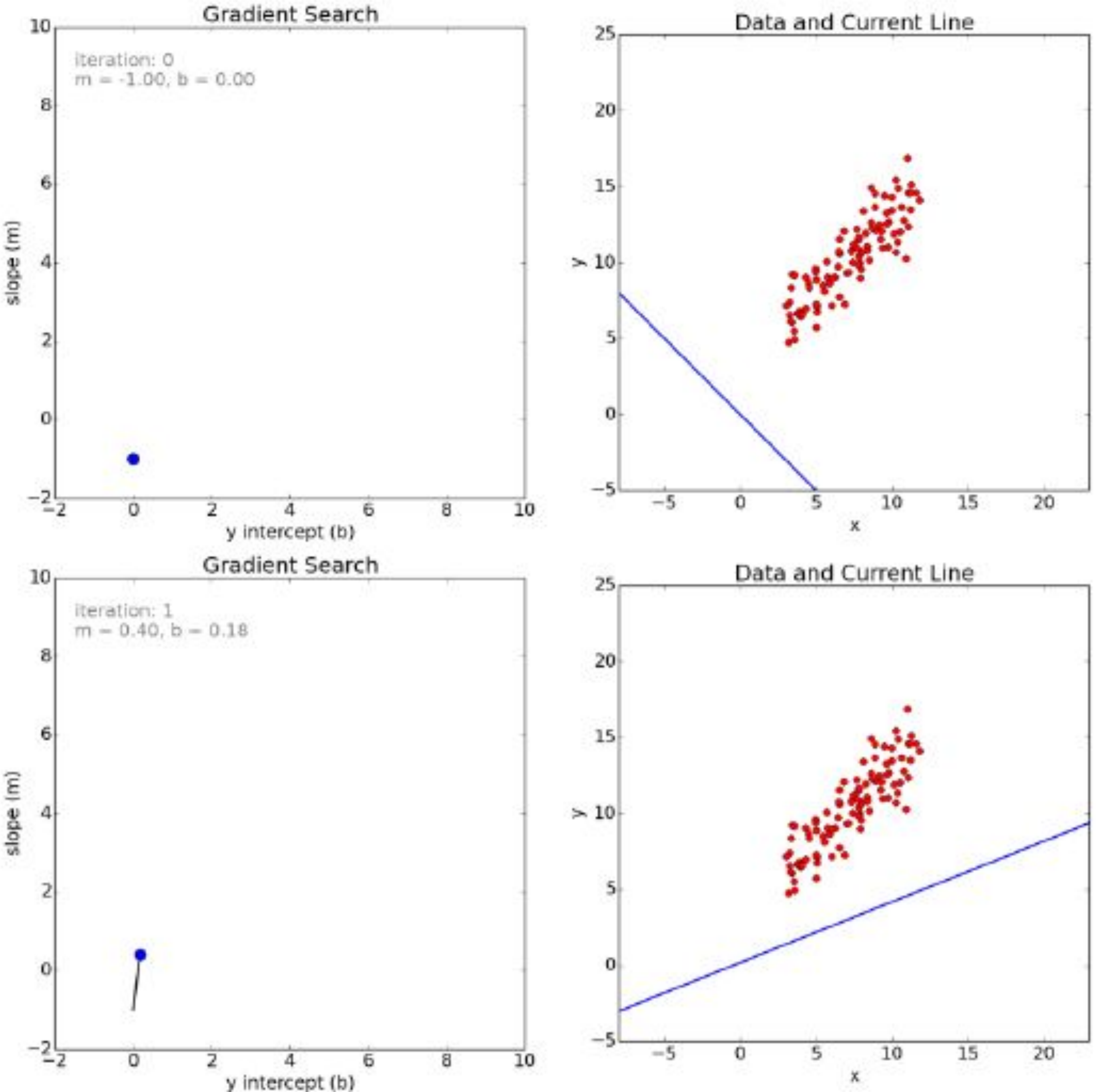
$$\theta(new) = \theta(old) + \Delta\theta$$

$$\Delta\theta = -\mu \frac{\partial J(\theta)}{\partial \theta}\bigg|_{\theta=\theta(old)}$$

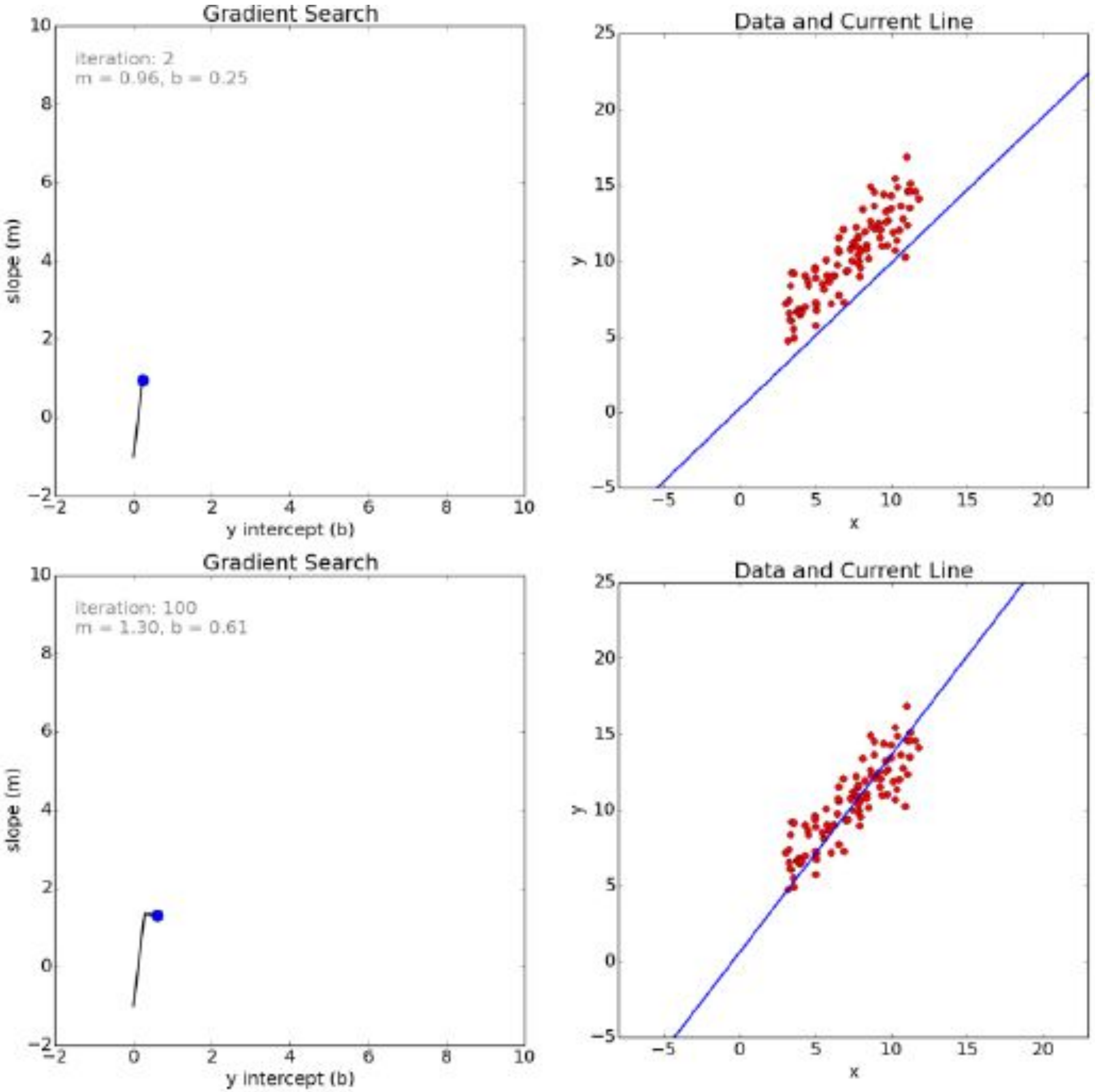
$$\frac{\partial J(a,b)}{\partial a} = 2 \sum_{i=1}^{N} \left[y_i - (a+bx_i)\right](-1)$$

$$\frac{\partial J(a,b)}{\partial b} = 2 \sum_{i=1}^{N} \left[y_i - (a+bx_i)\right](-x_i)$$

Grad


Gradient Search
iteration: 0
m = -1.00, b = 0.00
slope (m)
y intercept (b)
Data and Current Line
y
x
Gradient Search
iteration: 1
m = 0.40, b = 0.18
slope (m)
y intercept (b)
Data and Current Line
y
x

Grad



Gradient Search
iteration: 2
m = 0.96, b = 0.25
slope (m)
y intercept (b)
Data and Current Line
y
x
Gradient Search
iteration: 100
m = 1.30, b = 0.61
slope (m)
y intercept (b)
Data and Current Line
y
x

Convergence