MANCHESTER 1824
The University of Manchester

COMP24112: Machine Learning

Chapter 6: Training and Optimisation, III

Dr. Tingting Mu

Email: tingting.mu@manchester.ac.uk

Content

- Iterative optimisation approaches for training
  - Gradient descent
  - Stochastic gradient descent
  - Mini-batch gradient descent

- Often, it is impossible to solve a set of nonlinear equations!

$$\begin{cases} \dfrac{\partial O}{\partial \theta_1} = 0, \\ \dfrac{\partial O}{\partial \theta_2} = 0, \\ \vdots \\ \dfrac{\partial O}{\partial \theta_m} = 0 \end{cases}$$

What else can you do?

Gradient Descent

- An alternative approach is gradient descent (GD).
- To minimise the objective function,
  - We start from a random guess (called an initial state) of Θ, and apply a change to the guess.
  - The changed Θ aims at leading to a reduced value of the objective function.

The updating process:

$$\theta^{(1)} = \theta^{(0)} + \text{change}\left(\theta^{(0)}\right),$$
$$\theta^{(2)} = \theta^{(1)} + \text{change}\left(\theta^{(1)}\right),$$
$$\theta^{(3)} = \theta^{(2)} + \text{change}\left(\theta^{(2)}\right),$$
$$\vdots$$

$$\theta^{(t+1)} = \theta^{(t)} + \text{change}\left(\theta^{(t)}\right)$$

How to decide the change?

Gradient Descent

MANCHESTER 1824
The University of Manchester

- *According to the definition of the derivative,* ***gradient of a function indicates the direction in which the function ascends the fastest***.

- An effective way to set the change is

$$\text{change}\left(\theta\right) = -\eta \nabla O\left(\theta\right)$$ η >0: learning rate

  - Align the change with the function descending direction, which is the opposite of the gradient.
  - Set an appropriate learning rate.

Gradient Descent

The University of Manchester

- We start from a random guess (called an initial state) of Θ, and move along the direction $-\nabla O\left(\theta\right)$.

The updating process:

$$\theta^{(1)} = \theta^{(0)} - \eta \nabla O\left(\theta^{(0)}\right),$$
$$\theta^{(2)} = \theta^{(1)} - \eta \nabla O\left(\theta^{(1)}\right),$$
$$\theta^{(3)} = \theta^{(2)} - \eta \nabla O\left(\theta^{(2)}\right),$$
$$\vdots$$

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla O\left(\theta^{(t)}\right)$$

Try a simple example:

MANCHESTER 1824
The University of Manchester

- Given a simple function ***O(x) = (x-1)^2 +2***, find its minimum by applying a gradient descent process.
  - Gradient: $\nabla O\left(x\right) = 2\left(x-1\right)$
  - Updating equation:

$$x^{(t+1)} = x^{(t)} - \eta \nabla O\left(x^{(t)}\right) = x^{(t)} - 2\eta\left(x^{(t)} - 1\right) = \left(1-2\eta\right)x^{(t)} + 2\eta$$

  - Set starting state (initialisation) and learning rate:

What happens with $x^{(0)} = 3,\ \eta = 0.1$?

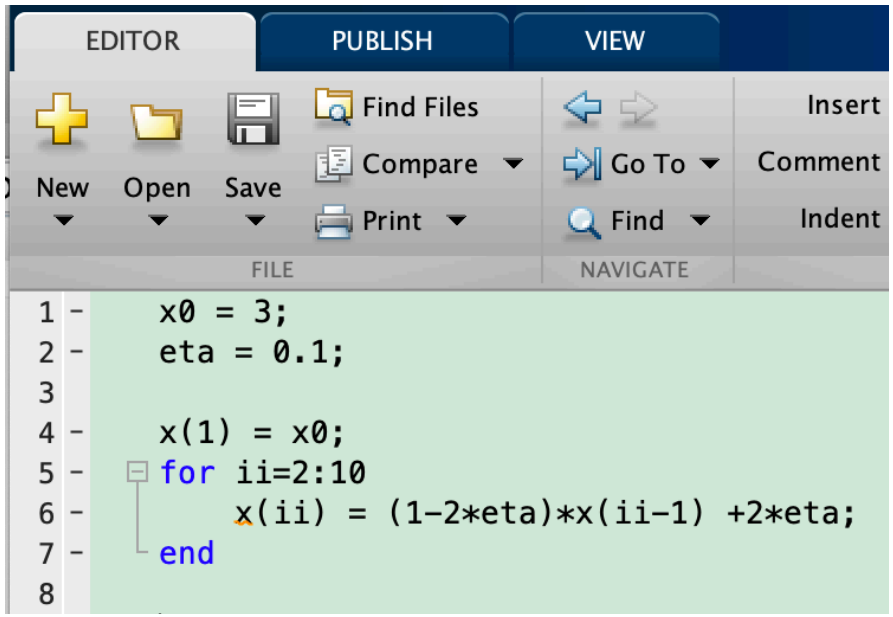
What happens with $x^{(0)} = 3,\ \eta = 0.5$?

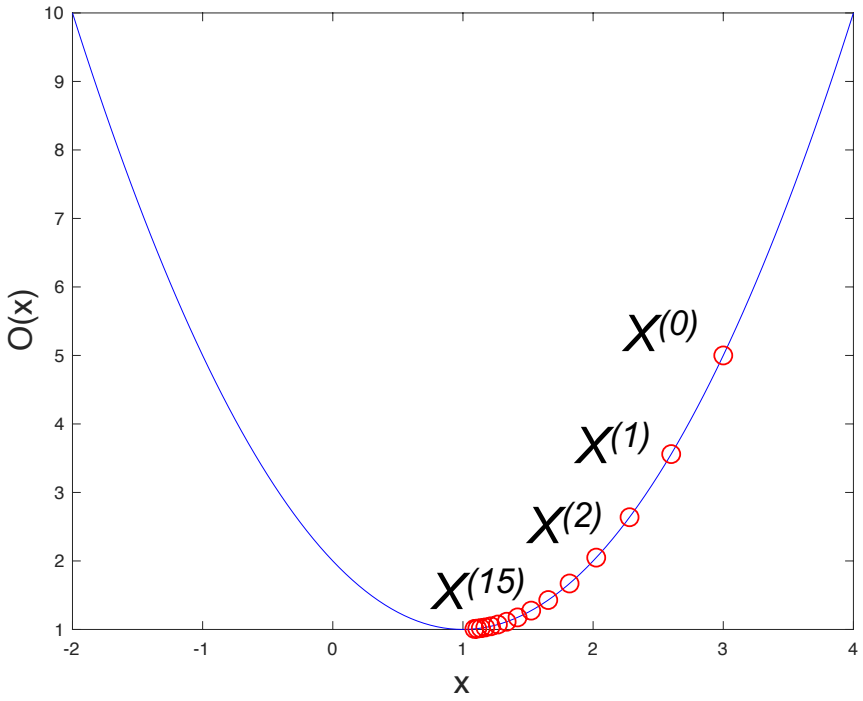
What happens with $x^{(0)} = 3,\ \eta = 0.95$?

Try a simple example:

- Given a simple function ***O(x) = (x-1)² +2***, find its minimum by applying a gradient descent process.

What happens with $x^{(0)} = 3,\ \eta = 0.1$?



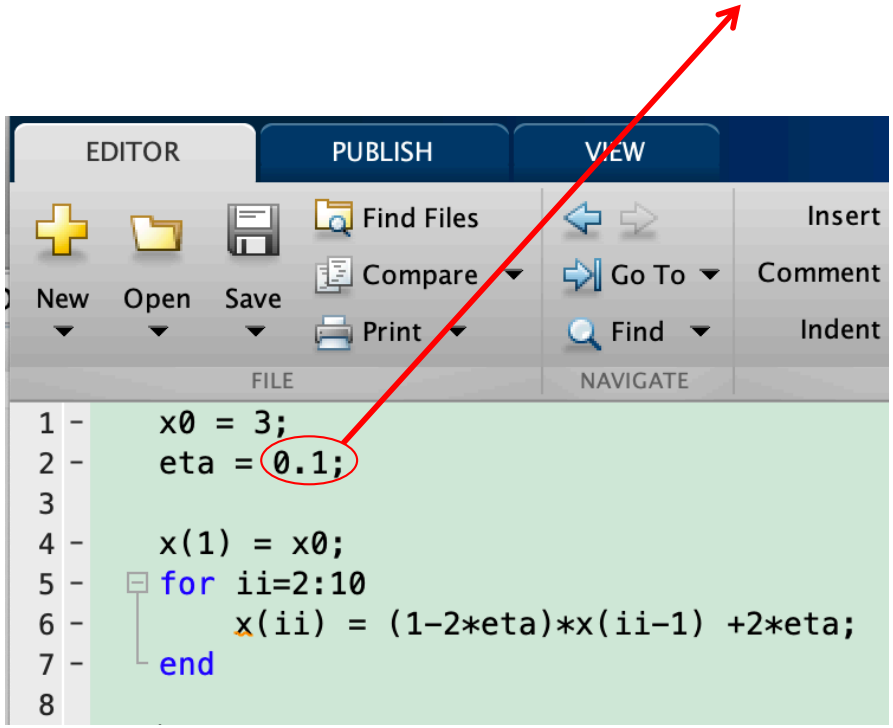


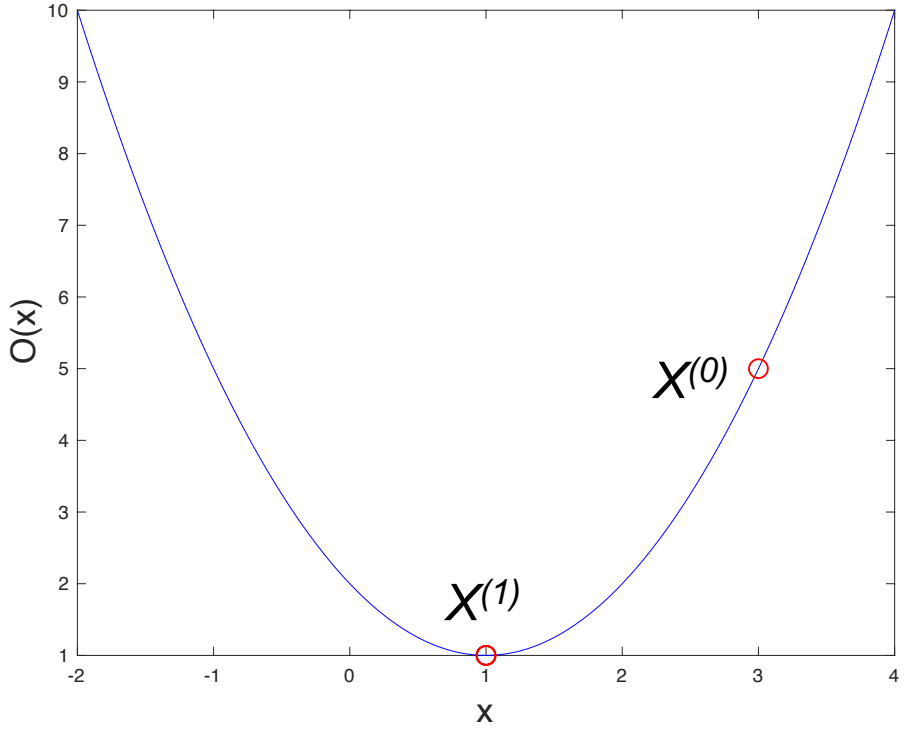
15 iterations

Try a simple example:

- Given a simple function ***O(x) = (x-1)² +2***, find its minimum by applying a gradient descent process.

What happens with $x^{(0)} = 3,\ \eta = 0.5$?



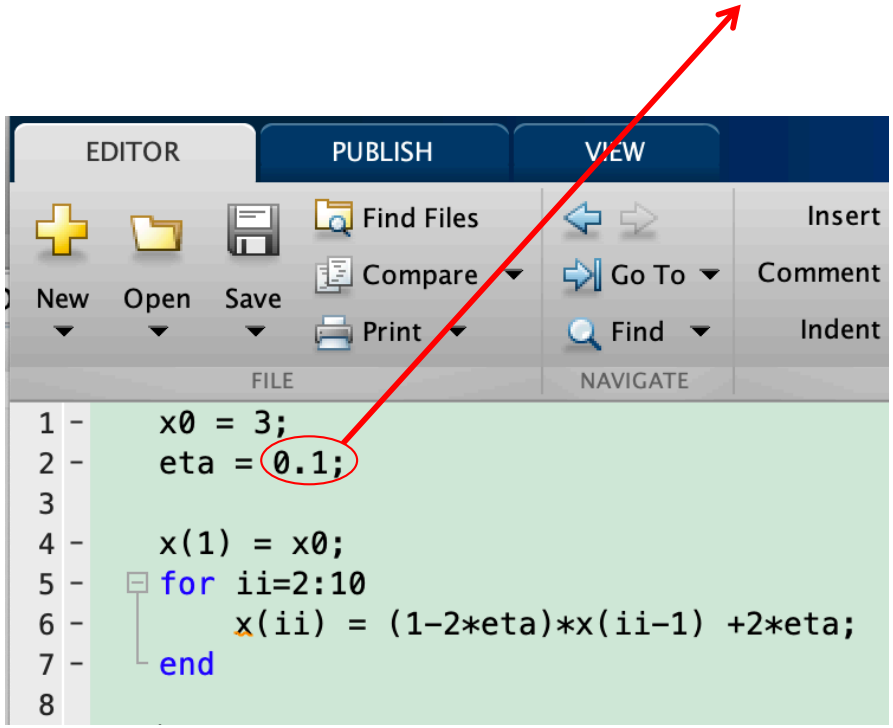


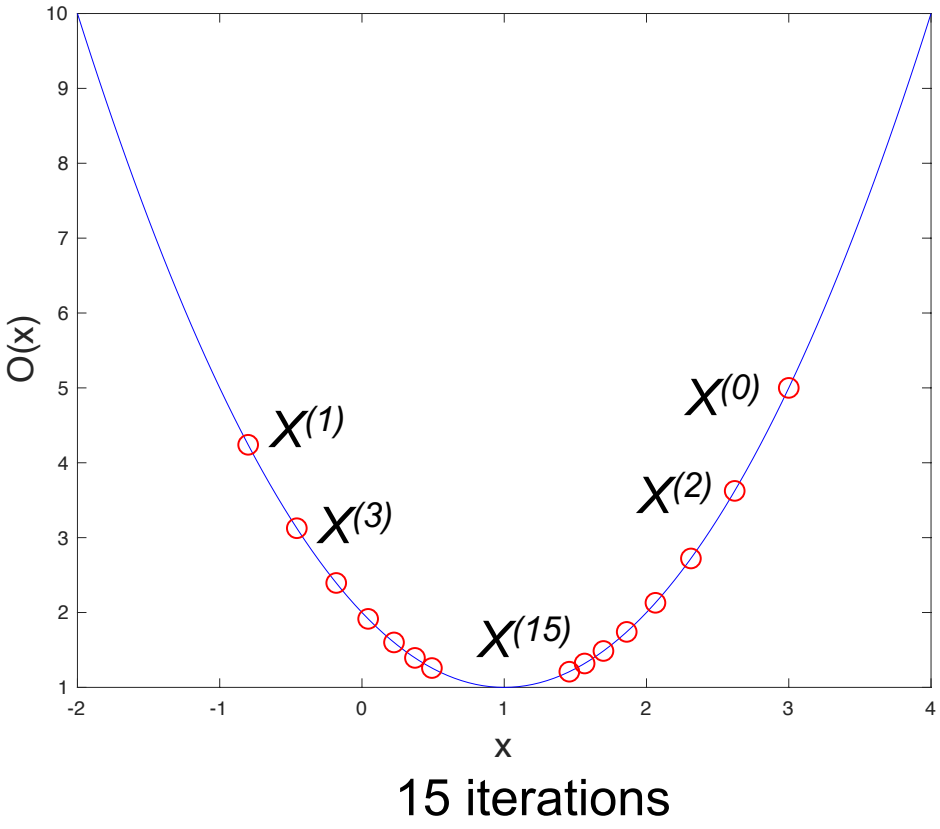
1 iteration

Try a simple example:

- Given a simple function ***O(x) = (x-1)² +2***, find its minimum by applying a gradient descent process.

What happens with $x^{(0)} = 3,\ \eta = 0.95$?





15 iterations

GD for Least Squares

- Gradient descent training for a least squares model:

$$\min_{\mathbf{w}} O(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2$$

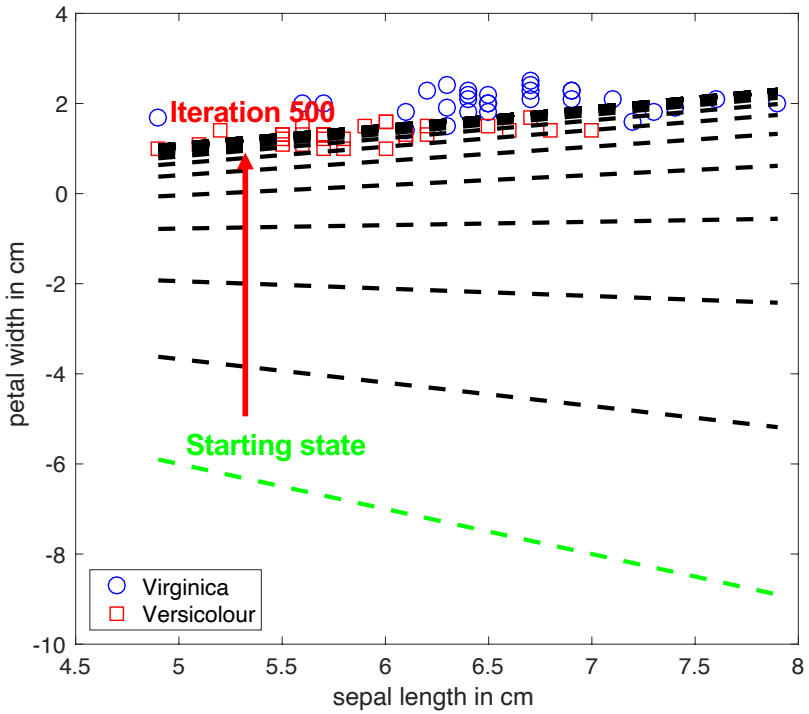
  - Gradient: $\nabla O(\mathbf{w}) = \tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{w} - \tilde{\mathbf{X}}^T\mathbf{Y}$

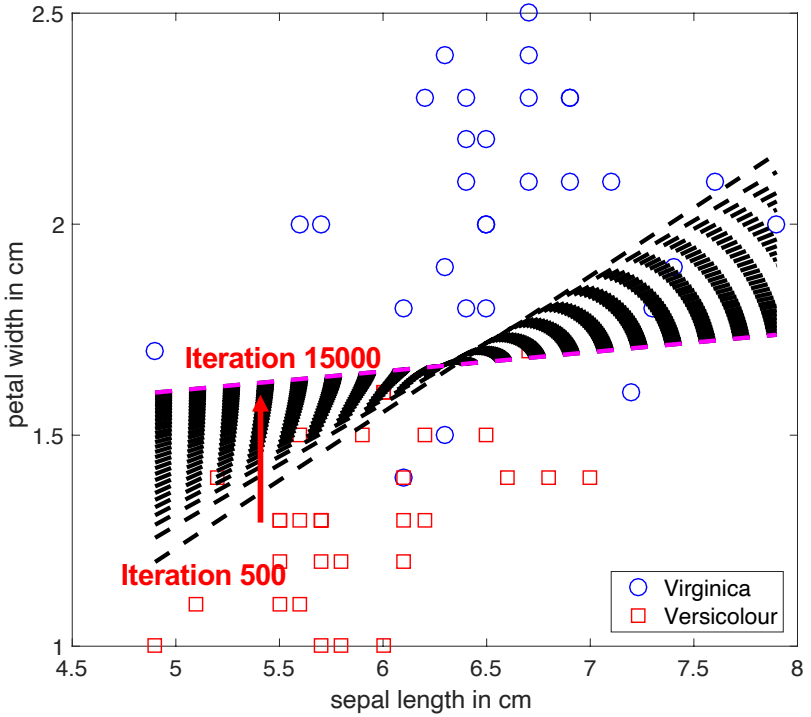
  - GD update: $\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta\left(\tilde{\mathbf{X}}^T\tilde{\mathbf{X}}\mathbf{w}^{(t)} - \tilde{\mathbf{X}}^T\mathbf{Y}\right)$

Example: Iris Classification

The University of Manchester

We observe the separation boundary change for the least squares linear model during gradient descent update.





Sequential (On-line) Training

- A learning process that involves the processing of the entire training set can be computationally costly for large datasets.
- Sequential (on-line) training updates model parameters after the presence of one training sample (or a small set of training samples).
- A sequential training process can be derived based on ***stochastic gradient descent***.

Stochastic Gradient Descent

The University of Manchester

Recall the error function computed using all the training samples: $O(\mathbf{w}) = \sum_{i=1}^{N} O_i(\mathbf{w})$

- Gradient descent (GD) computes the gradient of the error function using all the training samples.

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla O(\mathbf{w}^{(t)}) = \mathbf{w}^{(t)} - \eta \sum_{i=1}^{N} \nabla O_i(\mathbf{w}^{(t)})$$

Update using all the *N* training samples.

- Stochastic gradient descent (SGD) estimates the gradient of the error function using one training sample.

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla O_i(\mathbf{w}^{(t)})$$

Update using only one training sample.

- Mini-batch gradient descent (MBGD) estimates the gradient of the error function using a small set of training samples.

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \sum_{n \in I} \nabla O_i(\mathbf{w}^{(t)})$$

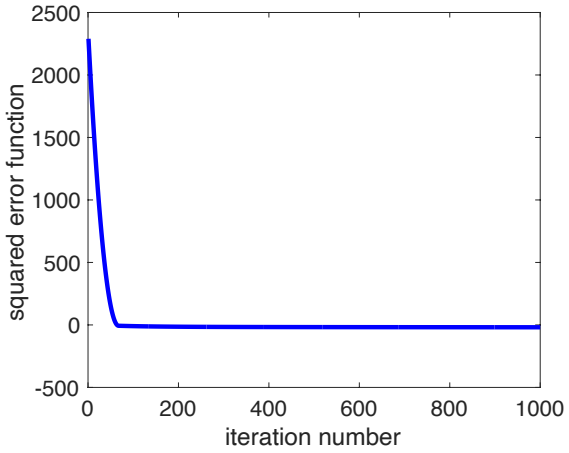
I denotes a small set of training samples.

Update using a subset of training samples.

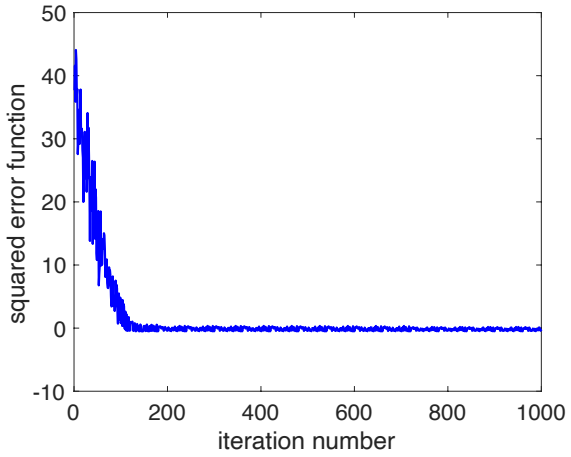
Example: GD and SGD Comparison, Iris

The University of Manchester

gradient descent (GD)



Stochastic gradient descent (SGD)

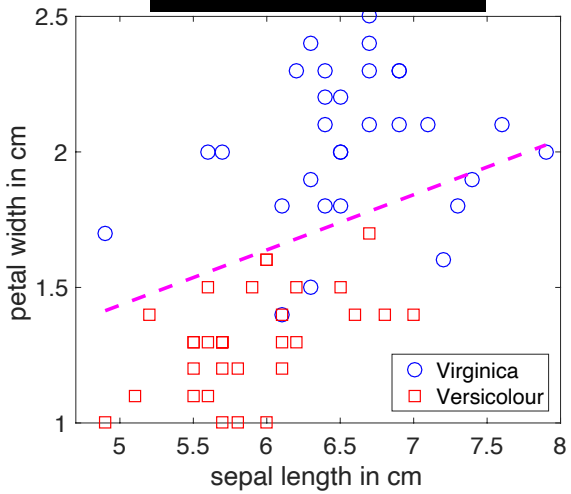


GD: update using true gradient computed from all the 60 training samples.

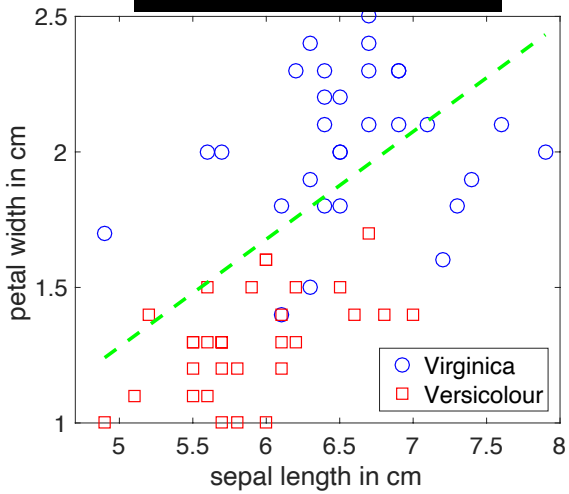
SGD: update using gradient estimated from one training sample.

Training samples and linear separation boundary.

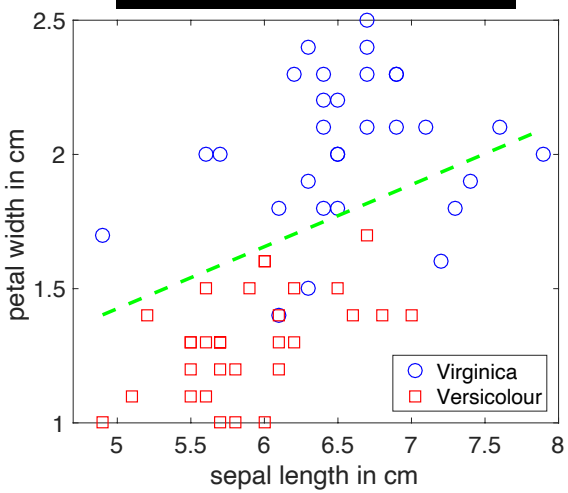
GD: 1000th iteration



SGD: 1000th iteration



SGD: 15000th iteration



Insurance Prediction Example

MANCHESTER 1824
The University of Manchester

Six drivers (A-F) have insured with a company, having similar auto insurance policies. The table lists their driving experience and monthly auto insurance premiums (MAIP). Predict the MAIP for two new drivers (G, H) according to their driving experiences.

Drivers	Driving Experience (in years)	MAIP (in dollars)
Driver A	5	64
Driver B	6	56
Driver C	12	50
Driver D	9	71
Driver E	15	44
Driver F	16	60
Driver G	2	?
Driver H	25	?

Linear Prediction

The University of Manchester

- Used model function for predicting MAIP given driver's experience:

$$\hat{y} = w_1 x + w_0$$

- Used loss function: sum-of-squares error of 6 drivers.

$$O(w_1, w_0) = \frac{1}{2}\Big[(5w_1 + w_0 - 64)^2 + (6w_1 + w_0 - 56)^2 + (12w_1 + w_0 - 50)^2 + (9w_1 + w_0 - 71)^2 + (15w_1 + w_0 - 44)^2 + (16w_1 + w_0 - 60)^2\Big]$$

Drivers	x	y
Driver A	5	64
Driver B	6	56
Driver C	12	50
Driver D	9	71
Driver E	15	44
Driver F	16	60

- Training: $\min O(w_1, w_0)$

Optimality Condition

The University of Manchester

- Compute the optimal w_1 and w_0:

$$\text{Optimality condition}: \frac{\partial O(w_1, w_0)}{\partial w_0} = 0, \quad \frac{\partial O(w_1, w_0)}{\partial w_1} = 0$$

$$\begin{aligned} & w_1 \sum_{i=1}^{6} x_i + 6w_0 - \sum_{i=1}^{6} y_i = 0; \\ & w_1 \sum_{i=1}^{6} x_i^2 + w_0 \sum_{i=1}^{6} x_i - \sum_{i=1}^{6} y_i x_i = 0; \end{aligned} \Rightarrow \begin{aligned} & 63w_1 + 6w_0 - 345 = 0 \\ & 767w_1 + 63w_0 - 3515 = 0 \end{aligned} \Rightarrow \begin{aligned} & w_1 = -1.02, \\ & w_0 = 68.2 \end{aligned}$$

Normal Equations

The University of Manchester

- Moore-Penrose inverse (or called pseudoinverse) :

$$\mathbf{w} = \tilde{\mathbf{X}}^{\dagger}\mathbf{y}$$

$$\tilde{\mathbf{X}} = \begin{bmatrix} 1 & 5 \\ 1 & 6 \\ 1 & 12 \\ 1 & 9 \\ 1 & 15 \\ 1 & 16 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} 64 \\ 56 \\ 50 \\ 71 \\ 44 \\ 60 \end{bmatrix}$$

$$\tilde{\mathbf{X}}^{\dagger} = \begin{bmatrix} 0.714 & 0.614 & 0.017 & 0.316 & -0.281 & -0.381 \\ -0.052 & -0.043 & 0.014 & -0.014 & 0.043 & 0.052 \end{bmatrix}$$

$$\mathbf{w} = [68.199, -1.020]^T$$

Drivers	x	y
Driver A	5	64
Driver B	6	56
Driver C	12	50
Driver D	9	71
Driver E	15	44
Driver F	16	60

Gradient Descent

$$w_i^{(t+1)} = w_i^{(t)} - \eta\left(\frac{\partial O(\mathbf{w}^{(t)})}{\partial w_i}\right)$$

- Used model function for predicting MAIP given driver's experience:

$$\frac{\partial O(w_1, w_0)}{\partial w_0} = \frac{1}{2}\sum_{i=1}^{6}\frac{\partial O_i(w_1, w_0)}{\partial w_0} = \sum_{i=1}^{6}(w_1 x_i + w_0 - y_i) = 63w_1 + 6w_0 - 345$$

$$\frac{\partial O(w_1, w_0)}{\partial w_1} = \frac{1}{2}\sum_{i=1}^{6}\frac{\partial O_i(w_1, w_0)}{\partial w_1} = \sum_{i=1}^{6}(w_1 x_i + w_0 - y_i)x_i = 767w_1 + 63w_0 - 3515$$

- We start from a random guess $w_0^0 = 2,\ \ w_1^0 = 4$, use learning rate $\eta = 0.001$, can get the following update in the 1st iteration.

$$w_0^1 = 2 - 0.001 \times (63 \times 4 + 6 \times 2 - 345) = 2.081$$

Stochastic Gradient Descent

- Update using driver A

$$O_A(w_0, w_1) = \frac{1}{2}(w_0 + 5w_1 - 64)^2$$

- Gradient estimated using driver A

$$\begin{bmatrix} \frac{\partial O_A}{\partial w_0} \\ \frac{\partial O_A}{\partial w_1} \end{bmatrix} = \begin{bmatrix} w_0 + 5w_1 - 64 \\ 5(w_0 + 5w_1 - 64) \end{bmatrix}$$

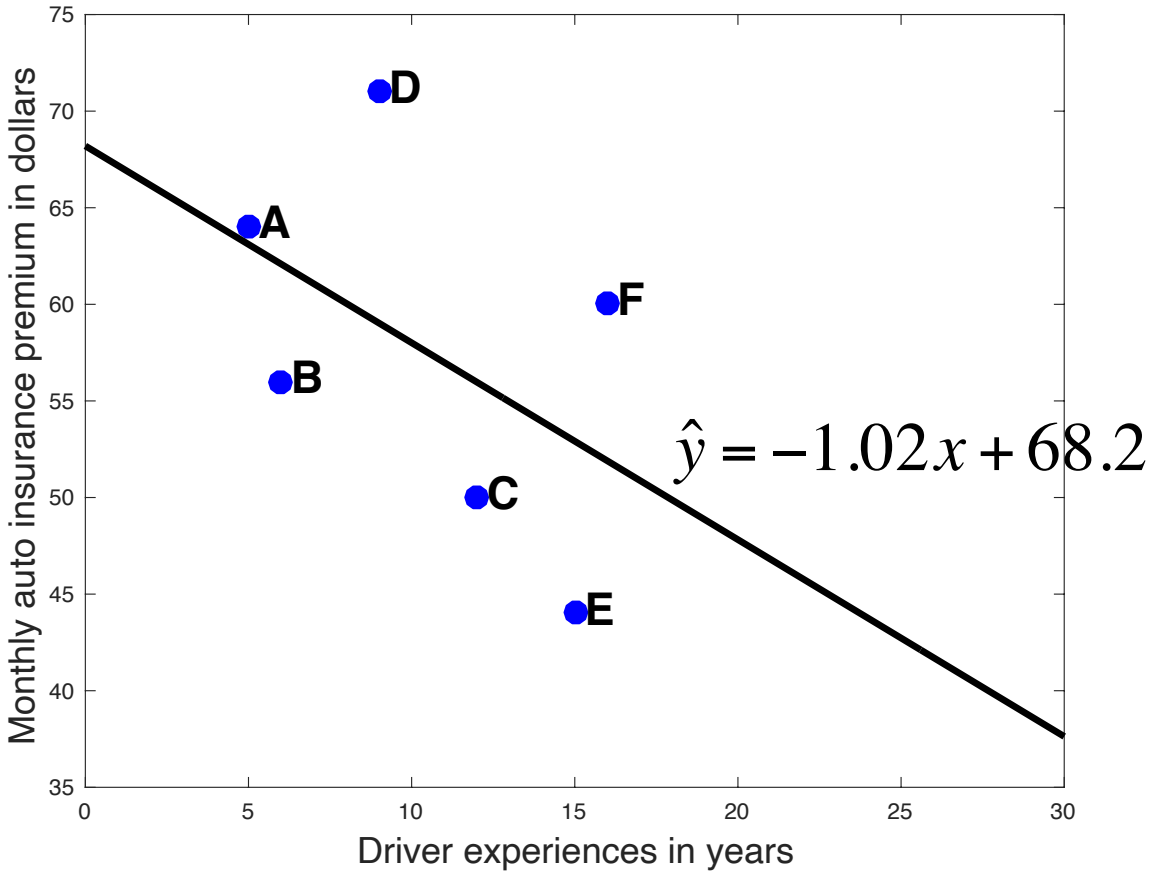
Drivers	x	y
Driver A	5	64
Driver B	6	56
Driver C	12	50
Driver D	9	71
Driver E	15	44
Driver F	16	60

- Update using driver A:

$$\begin{bmatrix} w_0^{(t+1)} \\ w_1^{(t+1)} \end{bmatrix} = \begin{bmatrix} w_0^{(t)} - \eta(w_0^{(t)} + 5w_1^{(t)} - 64) \\ w_1^{(t)} - 5\eta(w_0^{(t)} + 5w_1^{(t)} - 64) \end{bmatrix}$$

Insurance Prediction Example: Training

- The linear model trained using six training samples A-F:



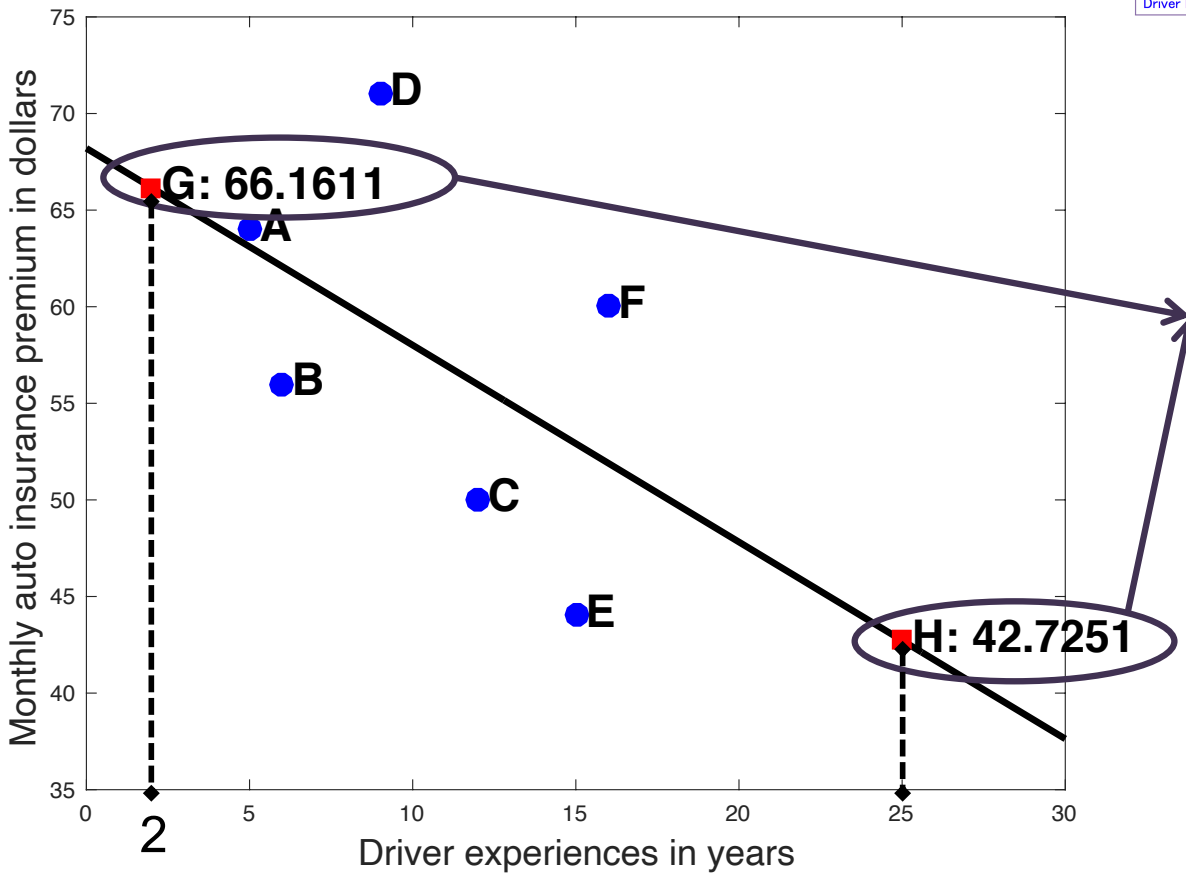
Drivers	x	y
Driver A	5	64
Driver B	6	56
Driver C	12	50
Driver D	9	71
Driver E	15	44
Driver F	16	60

Insurance Prediction Example: Testing

The University of Manchester

Trained linear model: $\hat{y} = -1.02x + 68.2$

Drivers	Driving Experience (in years)	MAIP (in dollars)
Driver A	5	64
Driver B	6	56
Driver C	12	50
Driver D	9	71
Driver E	15	44
Driver F	16	60
Driver G	2	?
Driver H	25	?



Predicting the MAIP for query drivers G and H:

$\hat{y}_G = -1.02 \times 2 + 68.2 \approx 66$

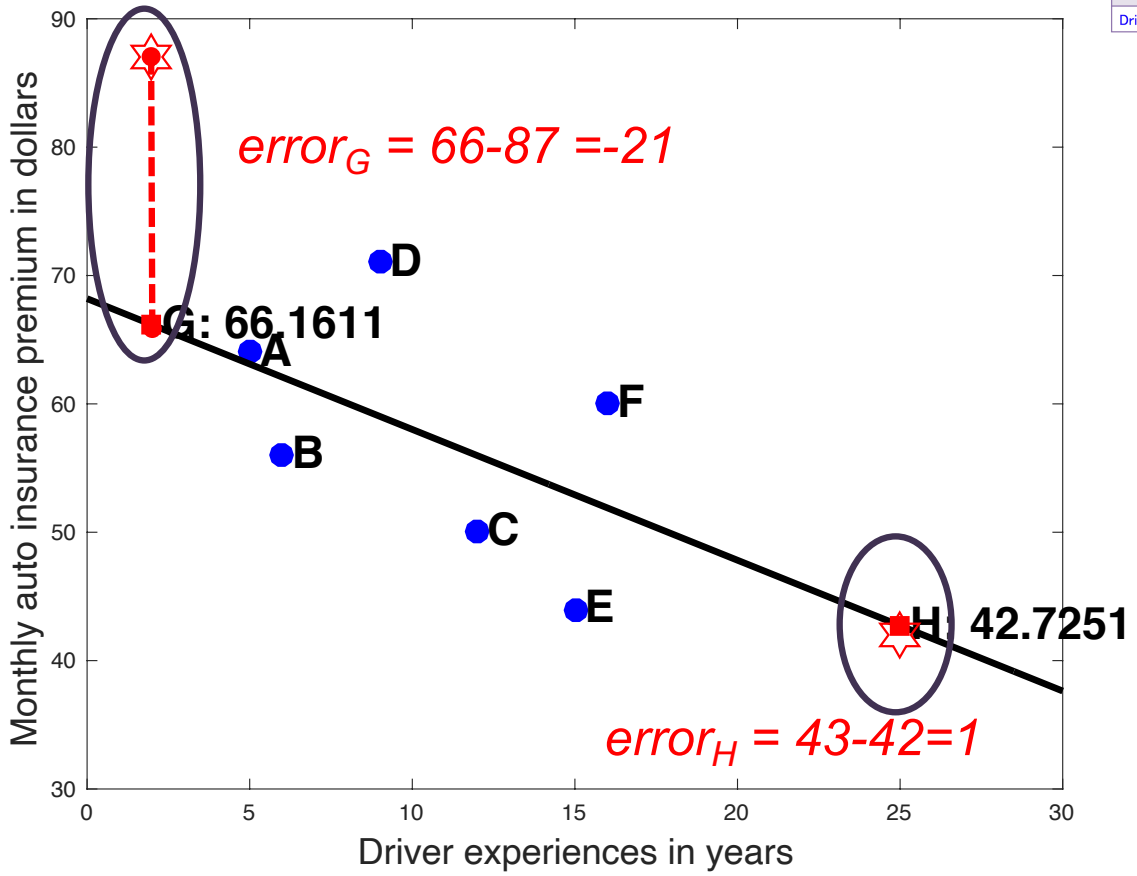
$\hat{y}_H = -1.02 \times 25 + 68.2 \approx 43$

Insurance Prediction Example: Testing

The University of Manchester

Trained linear model: $\hat{y} = -1.02x + 68.2$

Drivers	Driving Experience (in years)	MAIP (in dollars)
Driver A	5	64
Driver B	6	56
Driver C	12	50
Driver D	9	71
Driver E	15	44
Driver F	16	60
Driver G	2	?
Driver H	25	?



Compare the predicted MAIP with the real MAIP for drivers G and H.
real G: $87
real H: $42
predicted G: $66
predicted H: $43

Chapter 6 Summary: A, B and C

MANCHESTER 1824
The University of Manchester

- Typical approaches for minimising a loss function:
  - Zero gradient
  - Gradient descent
  - Stochastic gradient descent
  - Mini-batch gradient descent

- Examples and case studies:
  - (Regularized) linear least squares