Lecture 2

Machine Learning – I

Linear Regression

Outline

- Brief review
- Artificial Intelligence Tribes
- Some concepts of Machine Learning
- Univariate linear regression
- Basics of Python

A brief history of AI

1943
artificial neuron

1950
Turing test

1956
Birth of AI

1959
General problem solver

1962
checkers-playing program

1964
ELIZA

1969
perceptrons

1973
Lighthill report

1980
XCON

1982
Hopfield Network

1986
Backpropagation

1989
Handwritten Digital Recognition

1997
Deep Blue

2004
HTM

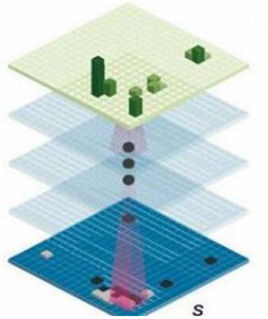
2006
Deep Learning

2012
CNN won ImageNet

2016
AlphaGo

Birth | Golden Years | First Winter | Revival | Second Winter | Rise | Boom

Keys triggered the revolution of AI



Models: Deep Neural Networks

AI

AlphaGo

Big Data

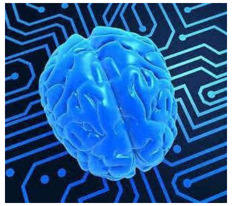
Computation Resource

What is artificial intelligence (AI)?

- Artificial intelligence refers to the simulation of human intelligence in machines.
- The intelligence refers to the ability of learning, memory, and prediction.
- It could be a robot, computer, program…

Learning ability

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning



Artificial Intelligence

Memory ability

- Knowledge
- Abstracting

Prediction ability

- Reasoning
- Understanding
- Judgment

What is artificial intelligence (AI)?

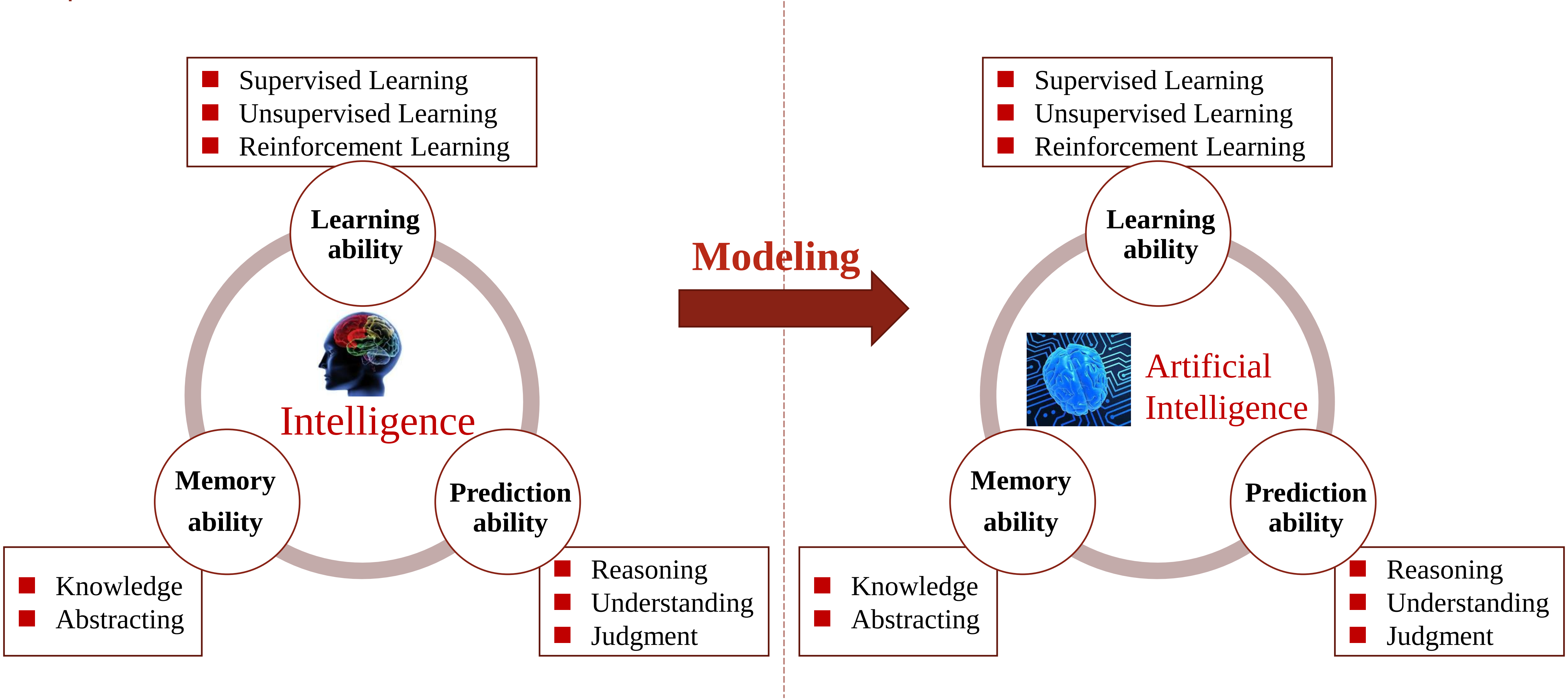
- **Weak AI or Narrow AI:**
  - •Weak AI is all around us and is the most successful realization of artificial intelligence to date.
  - •They are focused on a single task and achieve comparable performance to human beings.
  - •Generally, the weak AI system that plays go cannot play poker or chess.
  - •Examples: AlphaGO, “expert” medical diagnosis systems and stock-trading systems.

- **Strong AI or Artificial General Intelligence (AGI):**
  - •It is supposed to solve any problem rather than some specific task.
  - •The ultimate ambition of strong AI is to produce a machine whose overall intellectual ability is indistinguishable from that of a human being.
  - •Examples: the robots from *Westworld* or Data from *Star Trek: The Next Generation*.
  - •Is Strong AI Possible? & shall we pursue strong AI ?

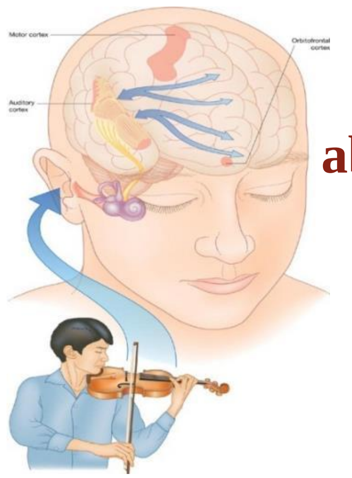
Outline

- Brief review
- Artificial Intelligence Tribes
- Some concepts of Machine Learning
- Univariate linear regression
- Basics of Python

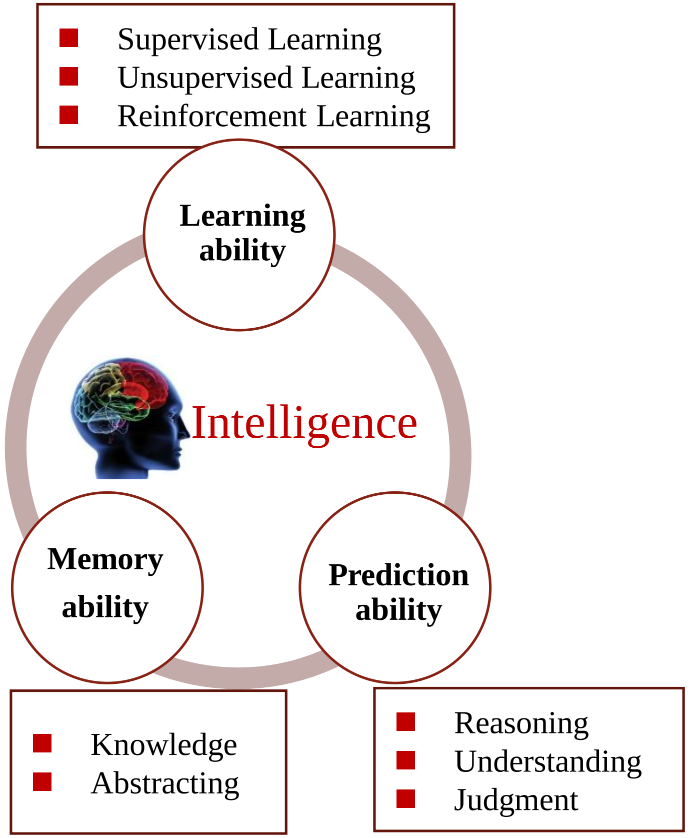
artificial intelligence tribes



artificial intelligence tribes



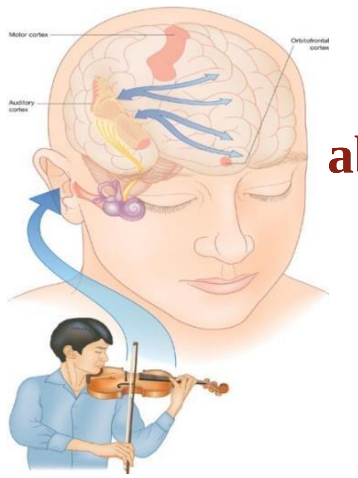
ability

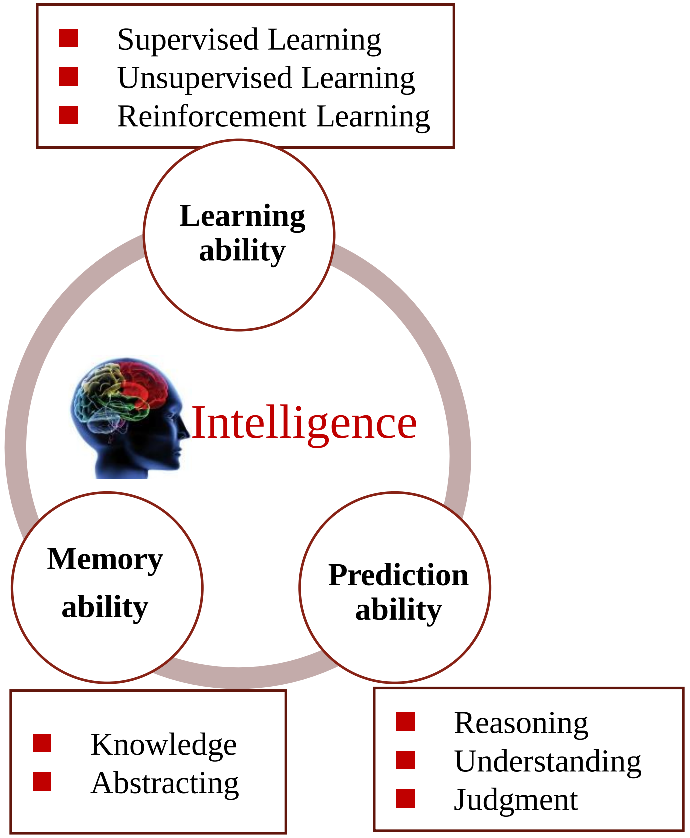


artificial intelligence tribes

ability

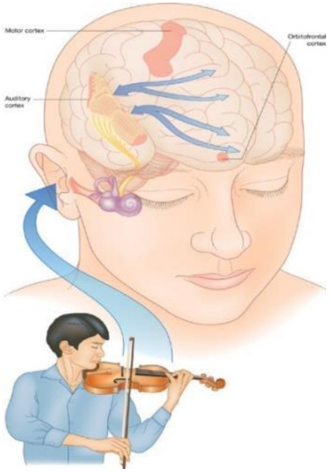
behavior

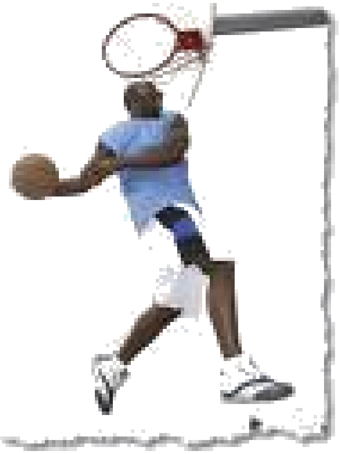


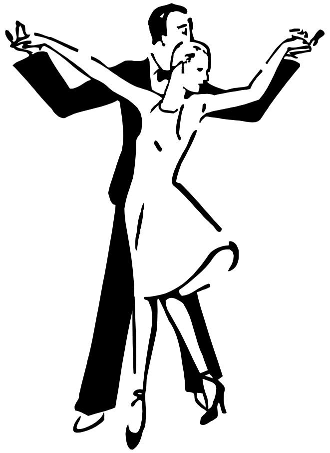


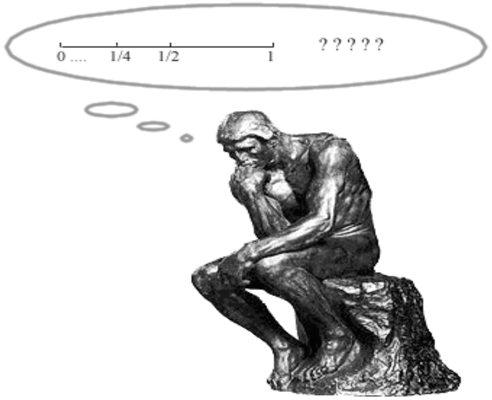
intelligence behavior

Behavior could be controlled by the brain.



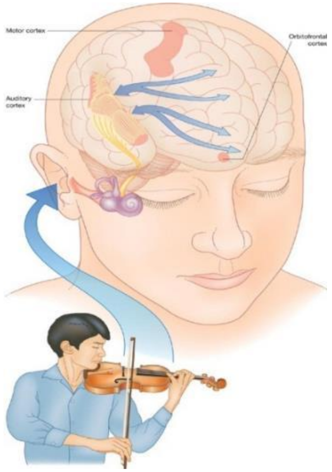




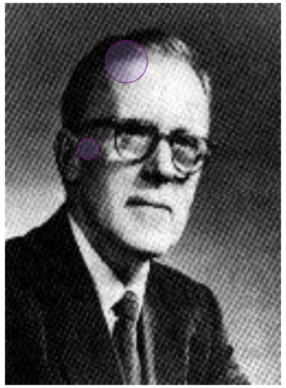


intelligence behavior

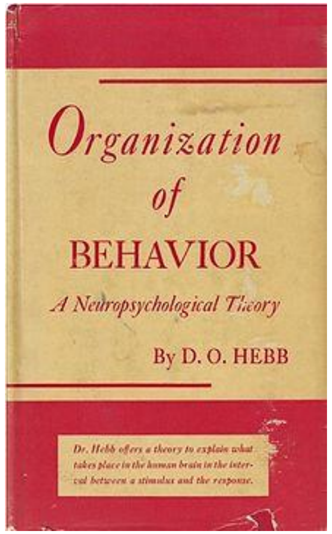
Behavior could be controlled by the brain.



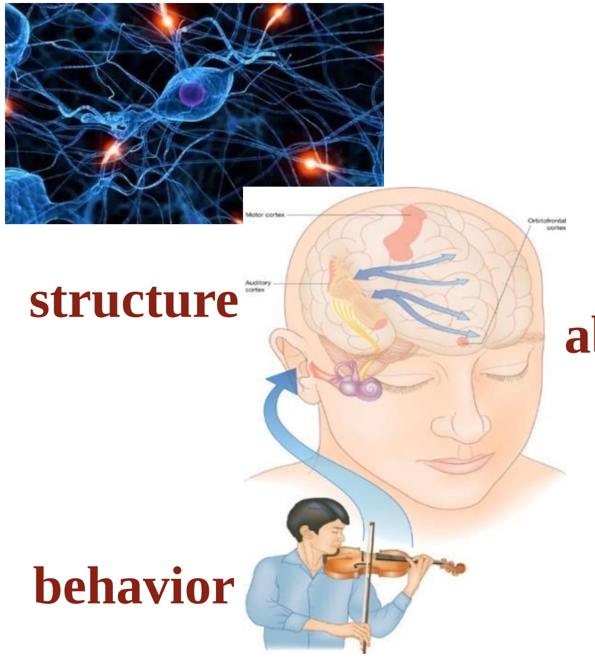
Behavior could be explained by the action of neurons.
-- D. O. Hebb, 1949



D. O. Hebb



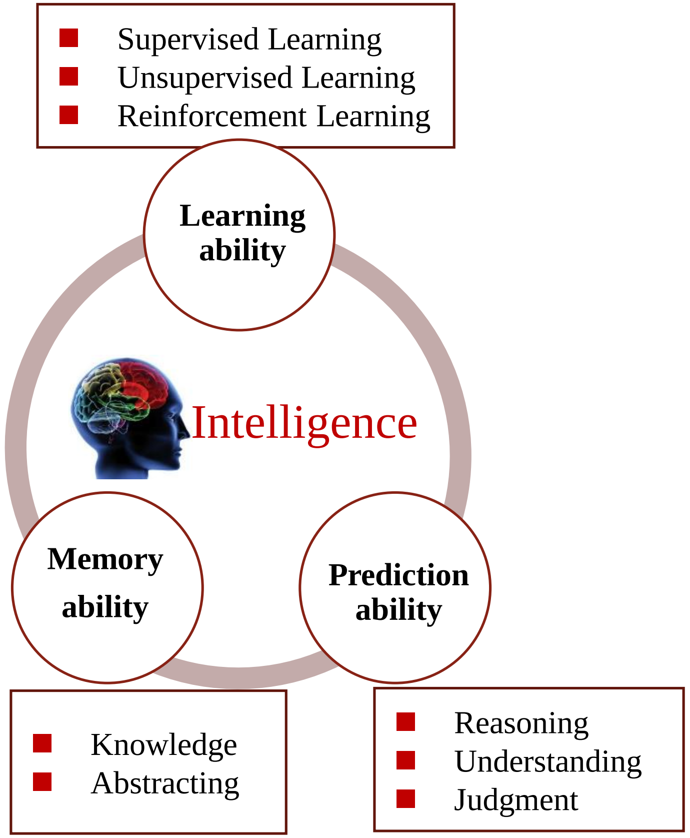
artificial intelligence tribes



structure

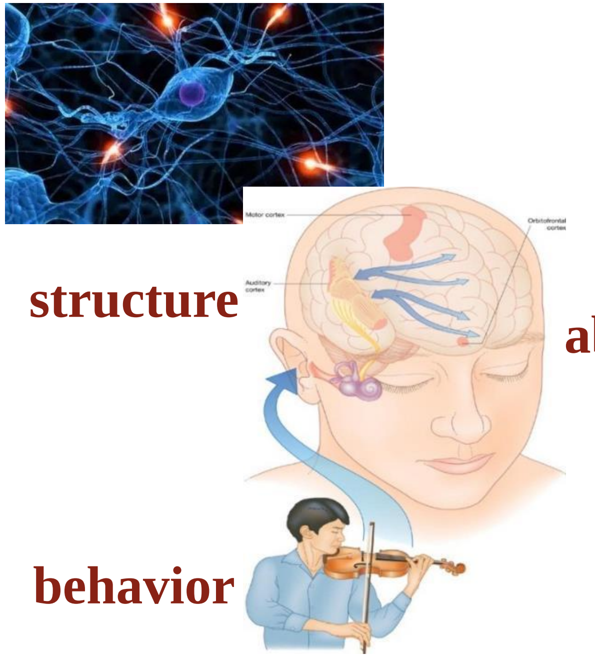
ability

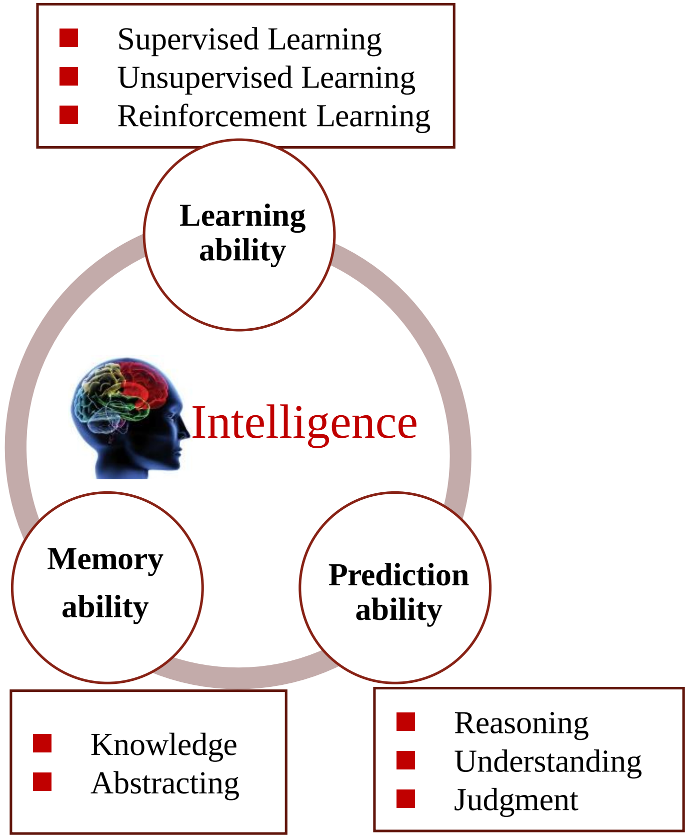
behavior



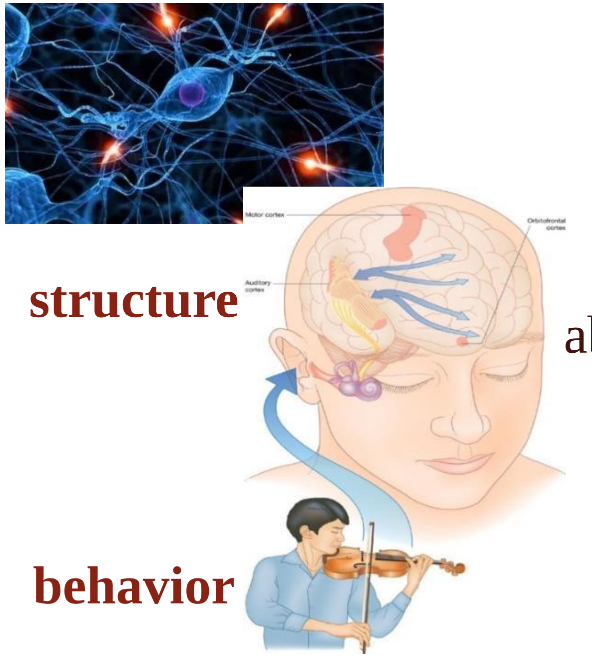
Neural network

artificial intelligence tribes





artificial intelligence tribes



structure

behavior

ability

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Learning ability

Intelligence

Memory ability

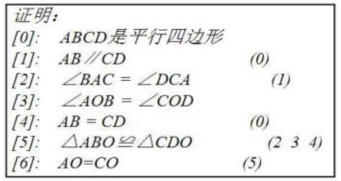
Prediction ability

- Knowledge
- Abstracting

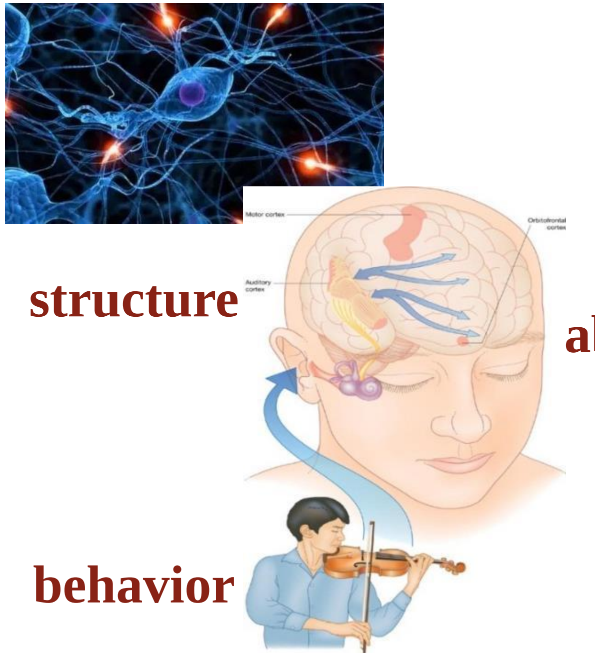
- Reasoning
- Understanding
- Judgment

Symbolicism

using symbolic rule-based systems to make inferences



artificial intelligence tribes



structure

ability

behavior

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Learning ability

Intelligence

Memory ability

Prediction ability

- Knowledge
- Abstracting

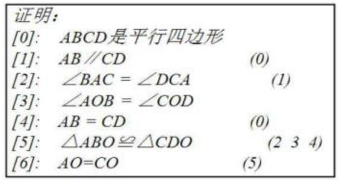
- Reasoning
- Understanding
- Judgment

Machine Learning

gives computers the ability to learn without being explicitly programmed

Symbolicism

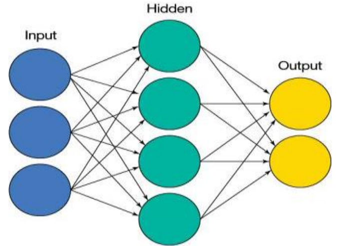
using symbolic rule-based systems to make inferences

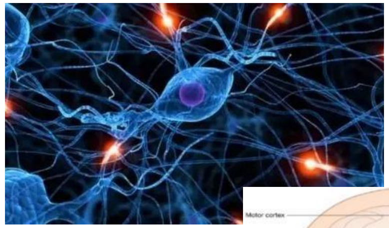


证明:
[0]: ABCD是平行四边形
[1]: AB∥CD (0)
[2]: ∠BAC = ∠DCA (1)
[3]: ∠AOB = ∠COD
[4]: AB = CD (0)
[5]: △ABO≌△CDO (2 3 4)
[6]: AO=CO (5)

artificial intelligence tribes

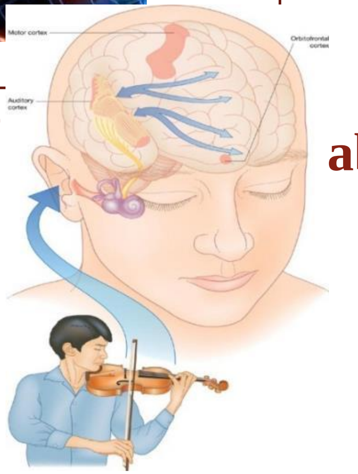
Connectionism





Deep Neural Networks

structure



ability

behavior

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Machine Learning

gives computers the ability to learn without being explicitly programmed

Learning ability

Intelligence

Memory ability

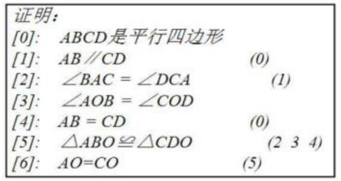
Prediction ability

- Knowledge
- Abstracting

- Reasoning
- Understanding
- Judgment

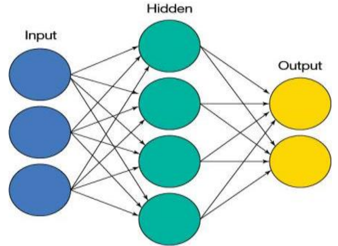
Symbolicism

using symbolic rule-based systems to make inferences

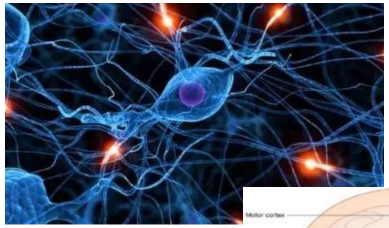

证明:
[0]: ABCD是平行四边形
[1]: AB∥CD (0)
[2]: ∠BAC = ∠DCA (1)
[3]: ∠AOB = ∠COD
[4]: AB = CD (0)
[5]: △ABO≌△CDO (2 3 4)
[6]: AO=CO (5)

artificial intelligence tribes

Connectionism

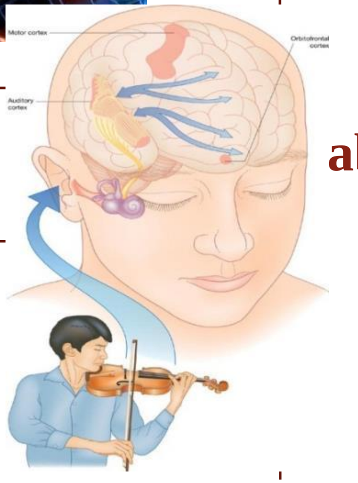


Deep Neural Networks



structure

behavior



ability

Machine Learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

gives computers the ability to learn without being explicitly programmed

Learning ability

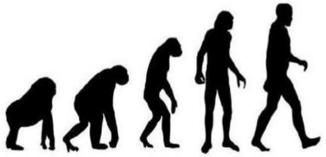
Intelligence

Memory ability

Prediction ability

Actionism

Cybernetics



Evolutionary computation

Symbolicism

using symbolic rule-based systems to make inferences

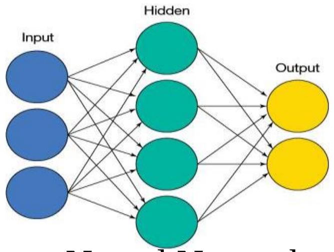
- Knowledge
- Abstracting

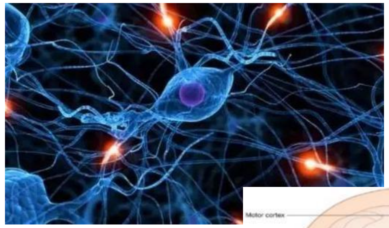
- Reasoning
- Understanding
- Judgment

证明:
[0]: ABCD是平行四边形
[1]: AB // CD (0)
[2]: ∠BAC = ∠DCA (1)
[3]: ∠AOB = ∠COD
[4]: AB = CD (0)
[5]: △ABO≌△CDO (2 3 4)
[6]: AO=CO (5)

artificial intelligence tribes

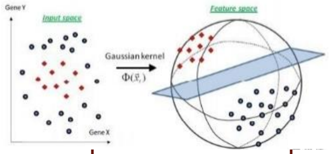
Connectionism





Deep Neural Networks

SVM



- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

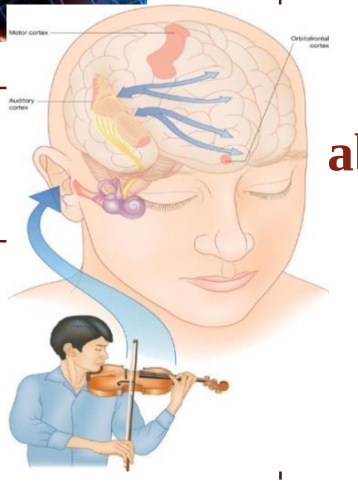
Machine Learning

gives computers the ability to learn without being explicitly programmed

Learning ability

structure

behavior



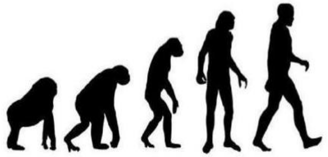
ability

Intelligence

Memory ability

Prediction ability

Actionism



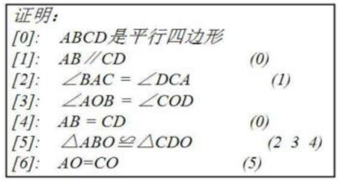
Evolutionary computation   Cybernetics

Symbolicism

using symbolic rule-based systems to make inferences

- Knowledge
- Abstracting

- Reasoning
- Understanding
- Judgment



证明:

[0]: ABCD是平行四边形

[1]: $AB \parallel CD$ (0)

[2]: $\angle BAC = \angle DCA$ (1)

[3]: $\angle AOB = \angle COD$

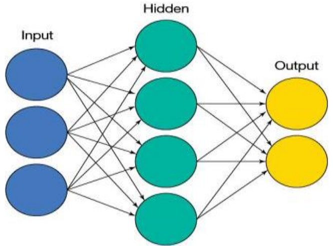
[4]: $AB = CD$ (0)

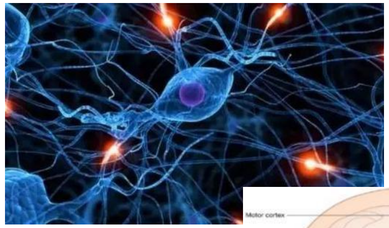
[5]: $\triangle ABO \cong \triangle CDO$ (2 3 4)

[6]: $AO=CO$ (5)

artificial intelligence tribes

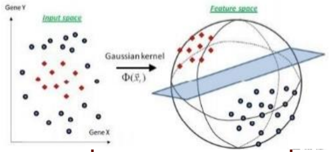
Connectionism





Deep Neural Networks

SVM



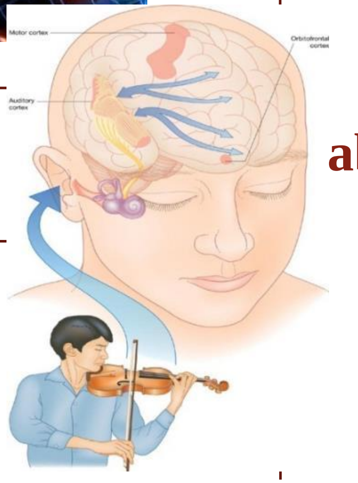
- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Machine Learning

gives computers the ability to learn without being explicitly programmed

structure

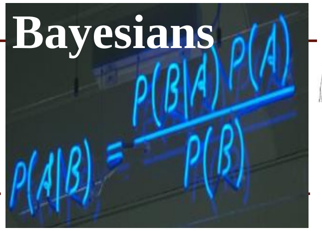
behavior

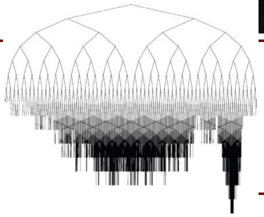


ability

Learning ability

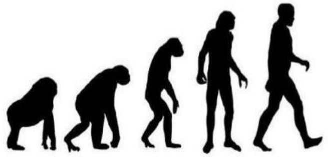
Intelligence





Trees

Actionism



Evolutionary computation Cybernetics

Memory ability

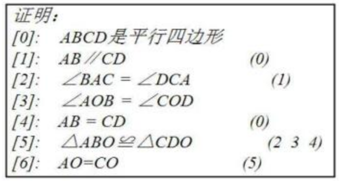
Prediction ability

- Knowledge
- Abstracting

- Reasoning
- Understanding
- Judgment

Symbolicism

using symbolic rule-based systems to make inferences



证明:

[0]:	ABCD是平行四边形	
[1]:	AB∥CD	(0)
[2]:	∠BAC = ∠DCA	(1)
[3]:	∠AOB = ∠COD	
[4]:	AB = CD	(0)
[5]:	△ABO≌△CDO	(2 3 4)
[6]:	AO=CO	(5)

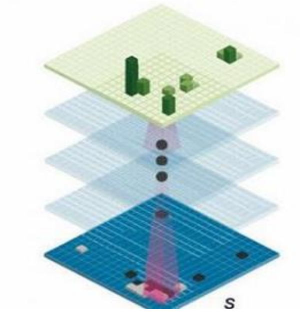
artificial intelligence tribes

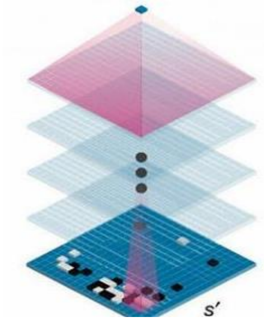
A combination of different approaches seems to be a better choice

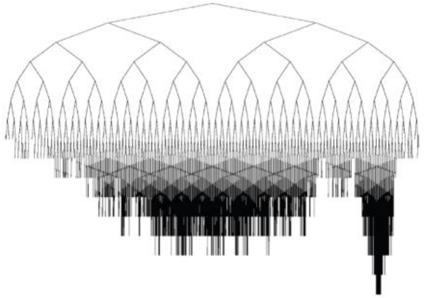
AlphaGo → **Policy Neural Network** → **Value Neural Network** → **MCTS** → **Move decision**

$F_1(s) = a$

$F_2(s') = z$

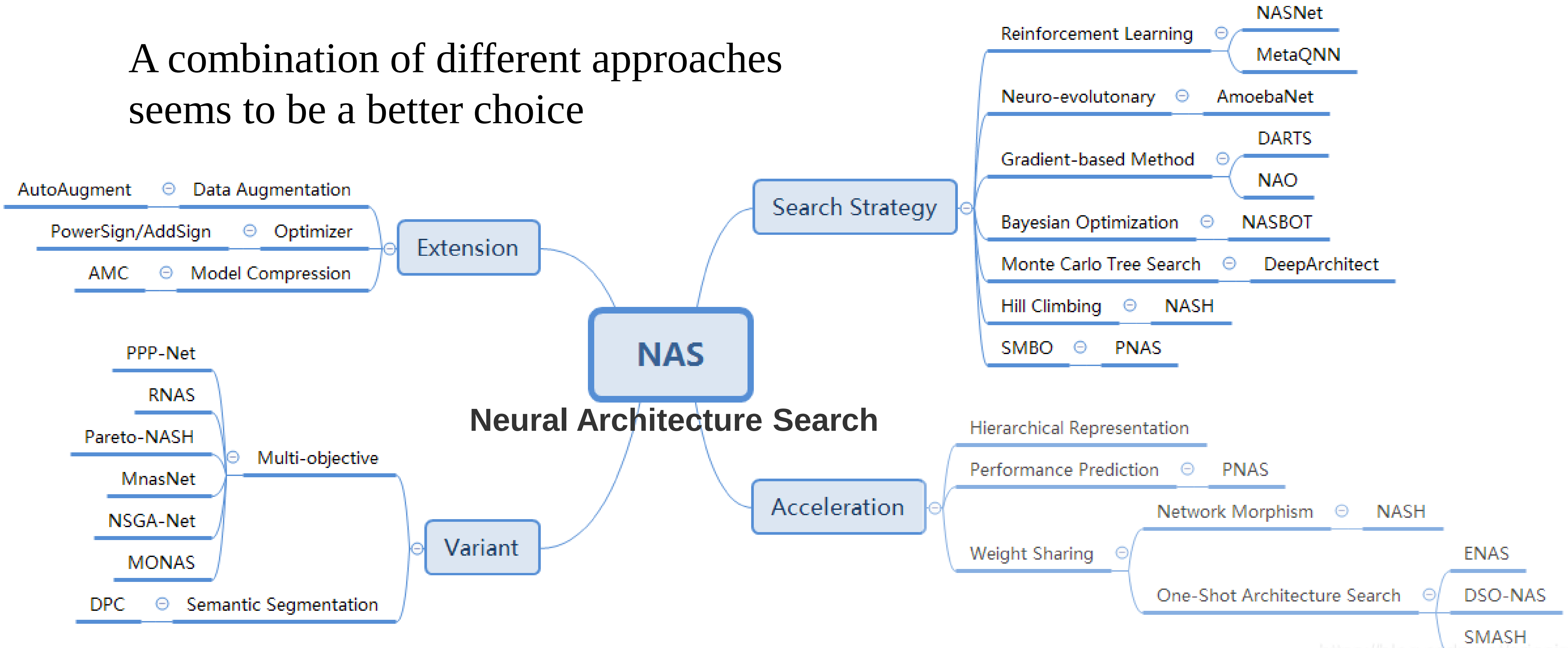






artificial intelligence tribes

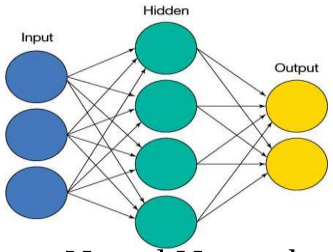
A combination of different approaches seems to be a better choice

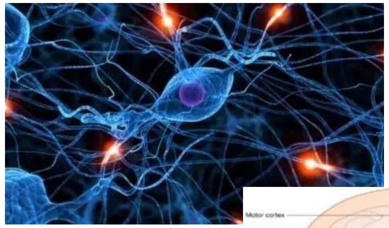


From https://blog.csdn.net/jinzhuojun/article/details/84698471

artificial intelligence tribes

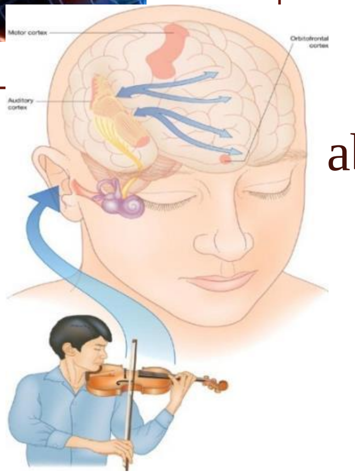
Connectionism





Deep Neural Networks

structure



ability

Machine Learning

gives computers the ability to learn without being explicitly programmed

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Learning ability

Intelligence

Memory ability

- Knowledge
- Abstracting

Prediction ability

- Reasoning
- Understanding
- Judgment

Outline

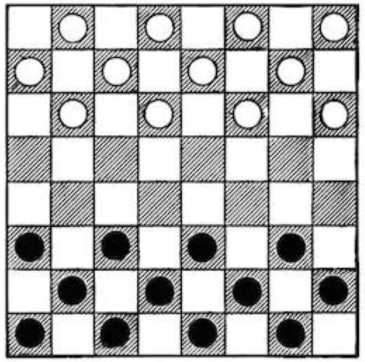
- Brief review
- Artificial Intelligence Tribes
- Some concepts of Machine Learning
- Univariate linear regression
- Basics of Python

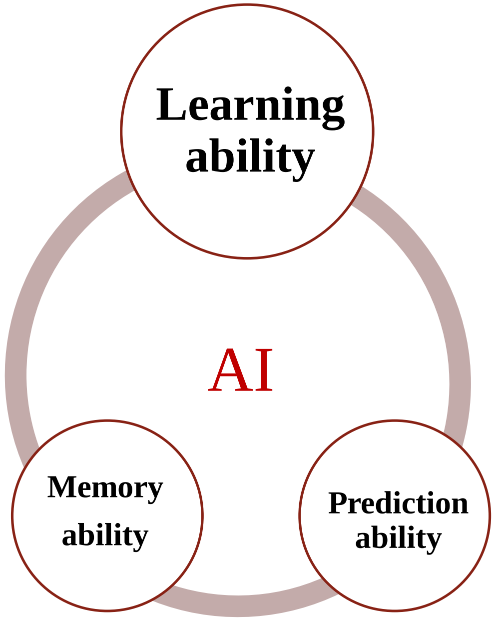
What is machine learning

Arthur Samuel (1959). Machine learning : Field of study that gives computers the ability to learn without being explicitly programmed.

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

I am not a very good checkers player.



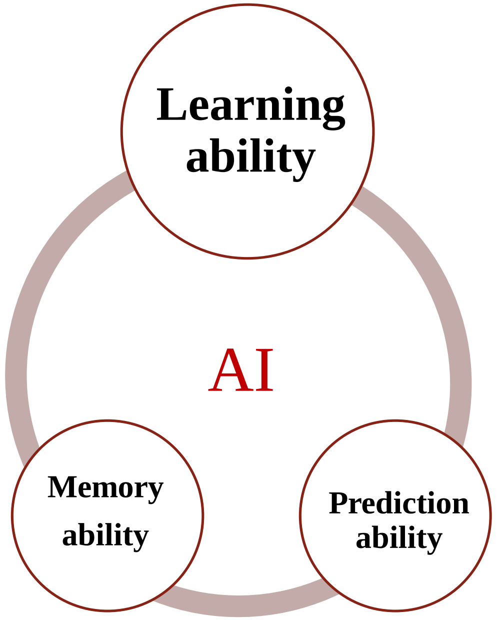


What is machine learning

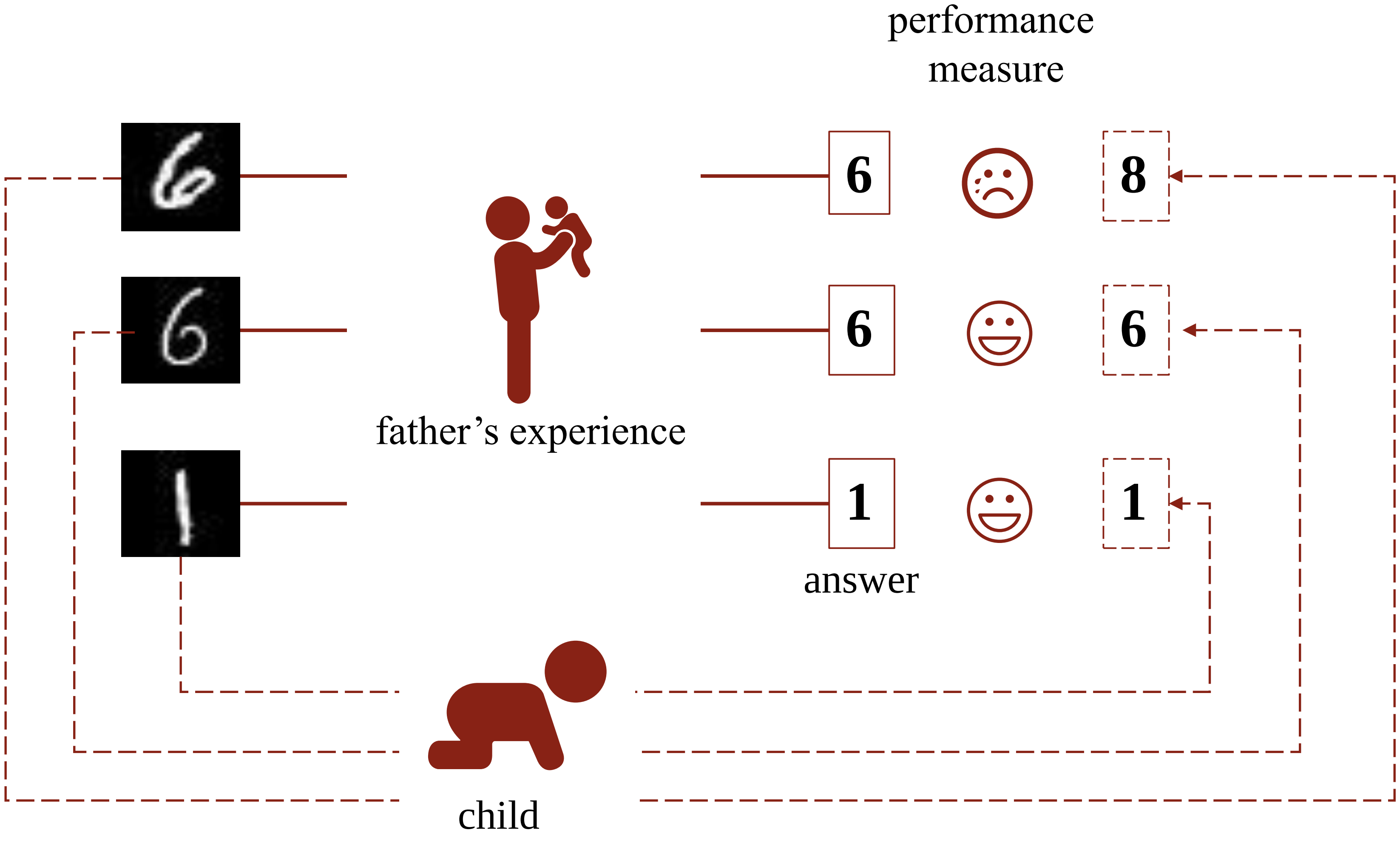
A widely quoted and formal definition is "*A computer program (agent) is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E*"

---Tom Mitchell(1998)

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning



What is learning

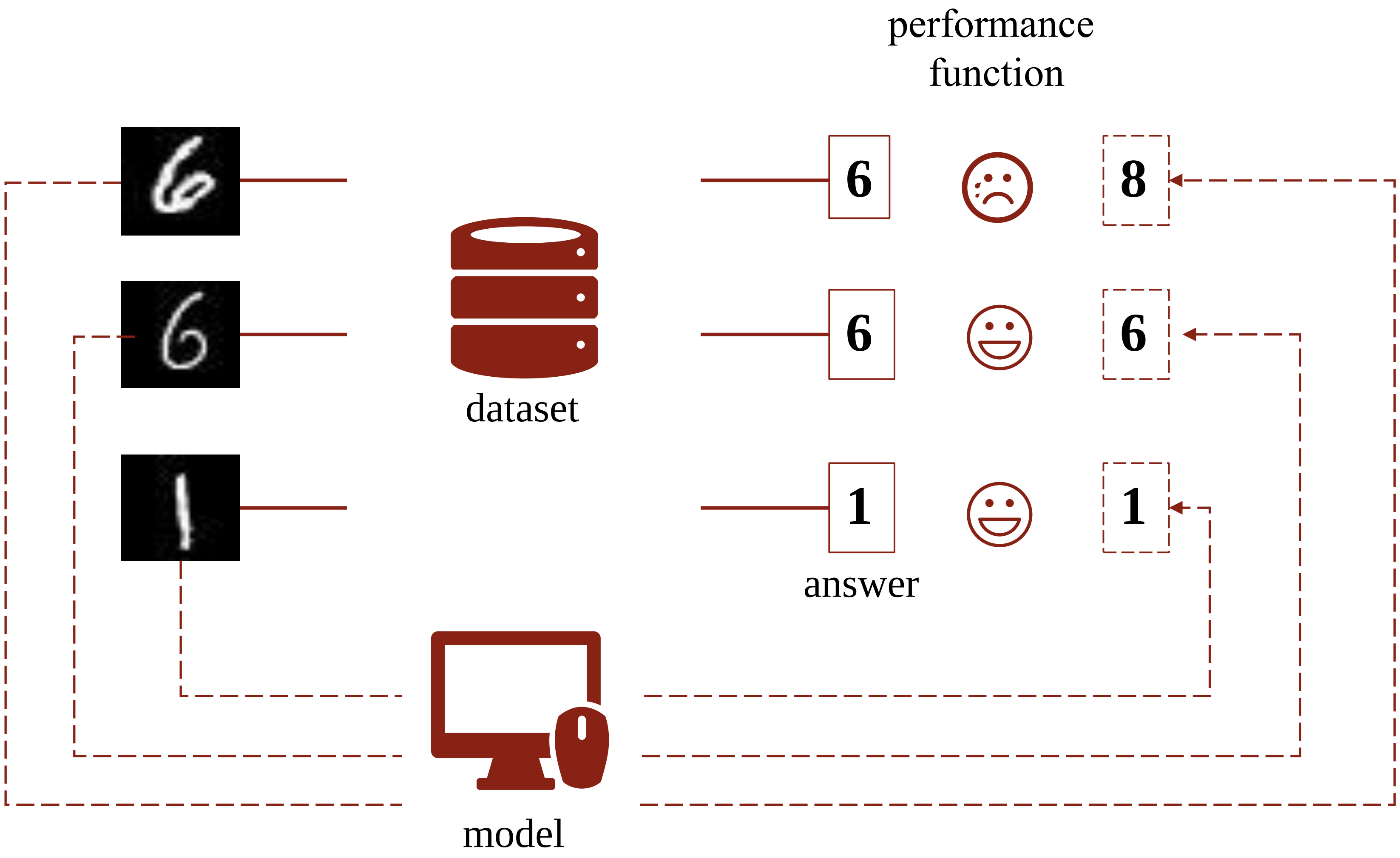


Learning

- **Task**: handwritten digit recognition
- **Agent**
- **Experience**: handwritten digits and the corresponding answers by father
- **Performance Measure**

The performance on Task is improved

What is machine learning

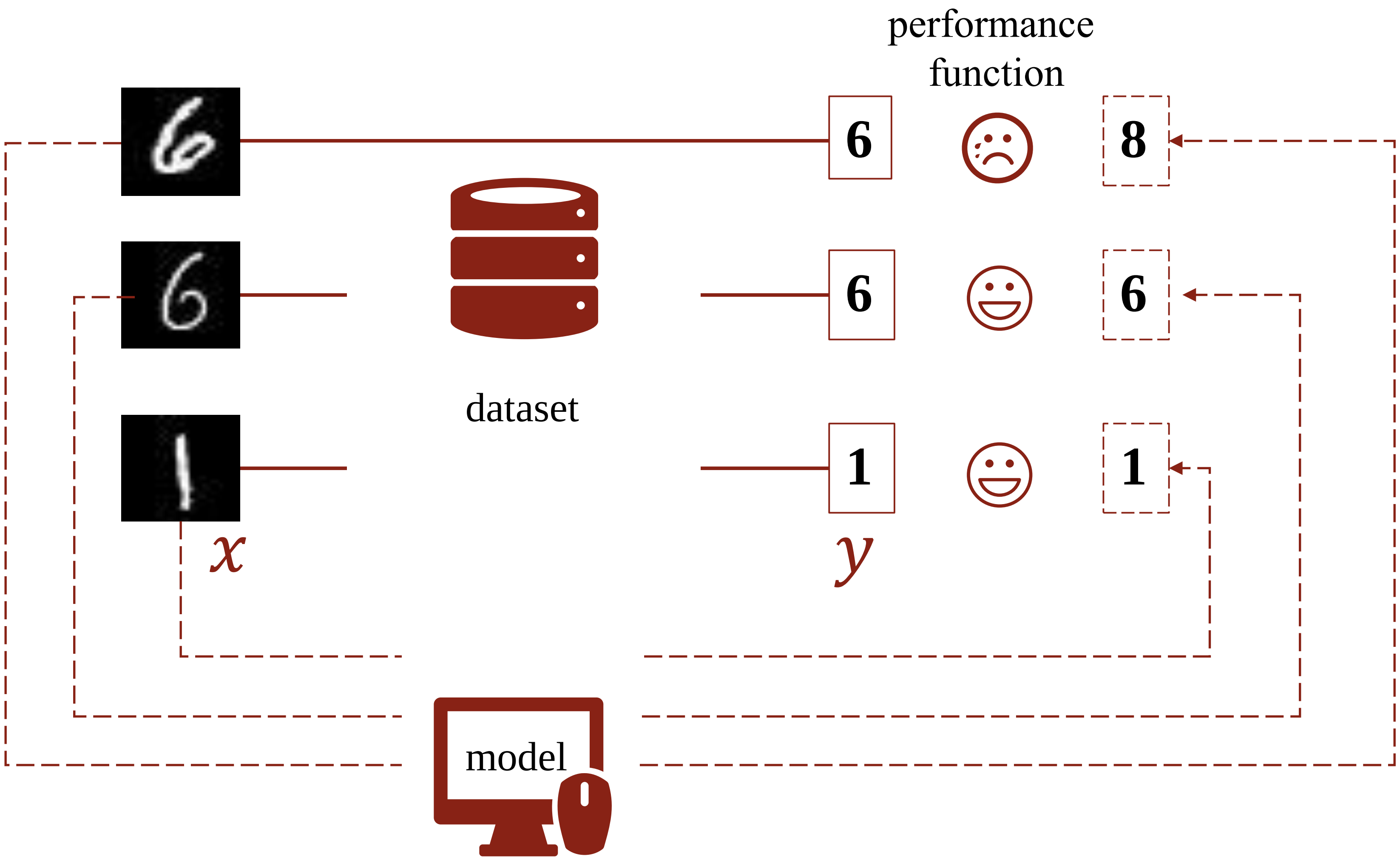


Machine Learning

- Task: handwritten digit recognition
- Model
- Dataset: handwritten digits and the corresponding answers
- Performance function

The performance on Task is improved

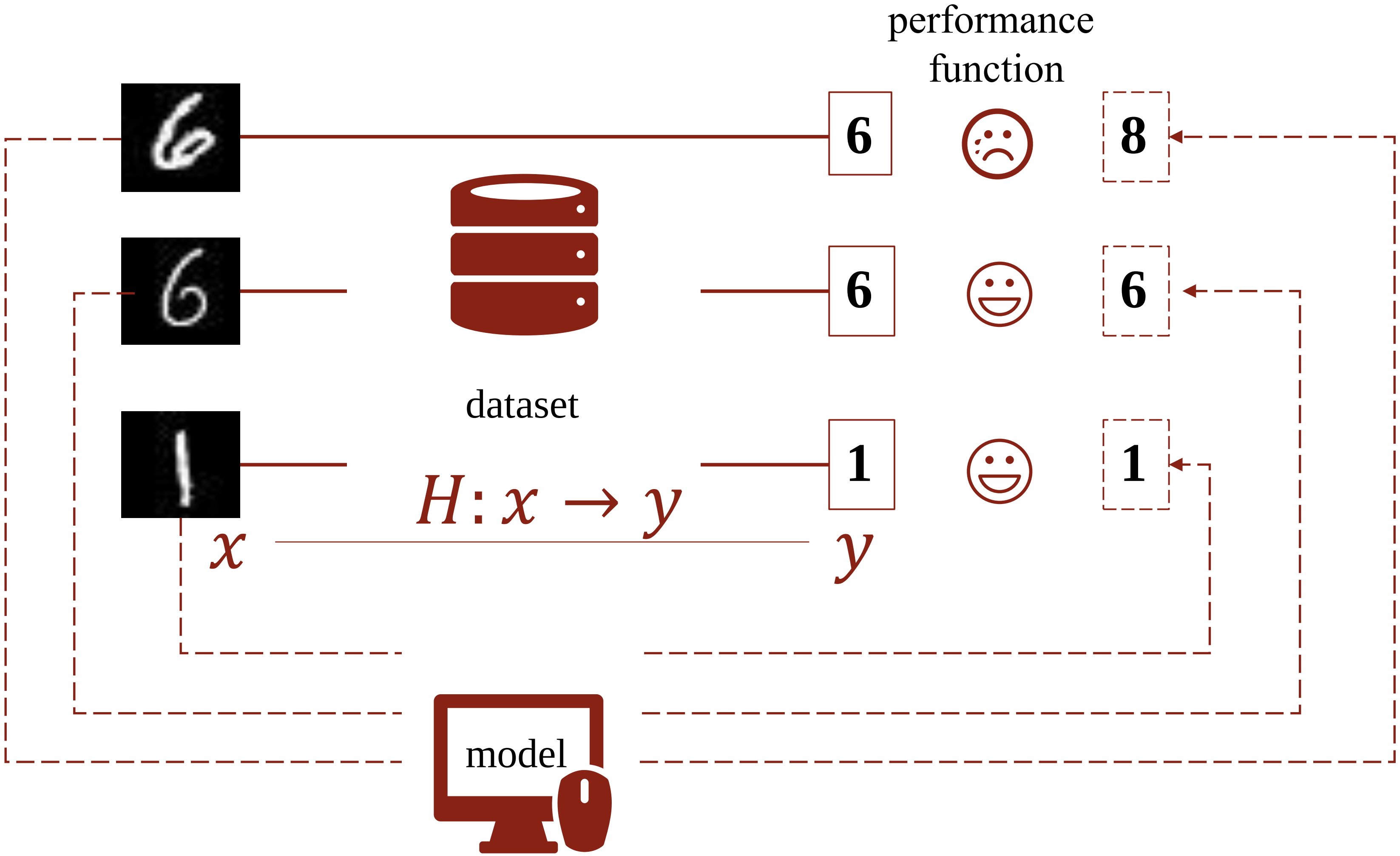
What is machine learning



Machine Learning

- Task: handwritten digit recognition
- Model
- Dataset: $D = \{(x, y)\}$
- Performance function: measure the learning

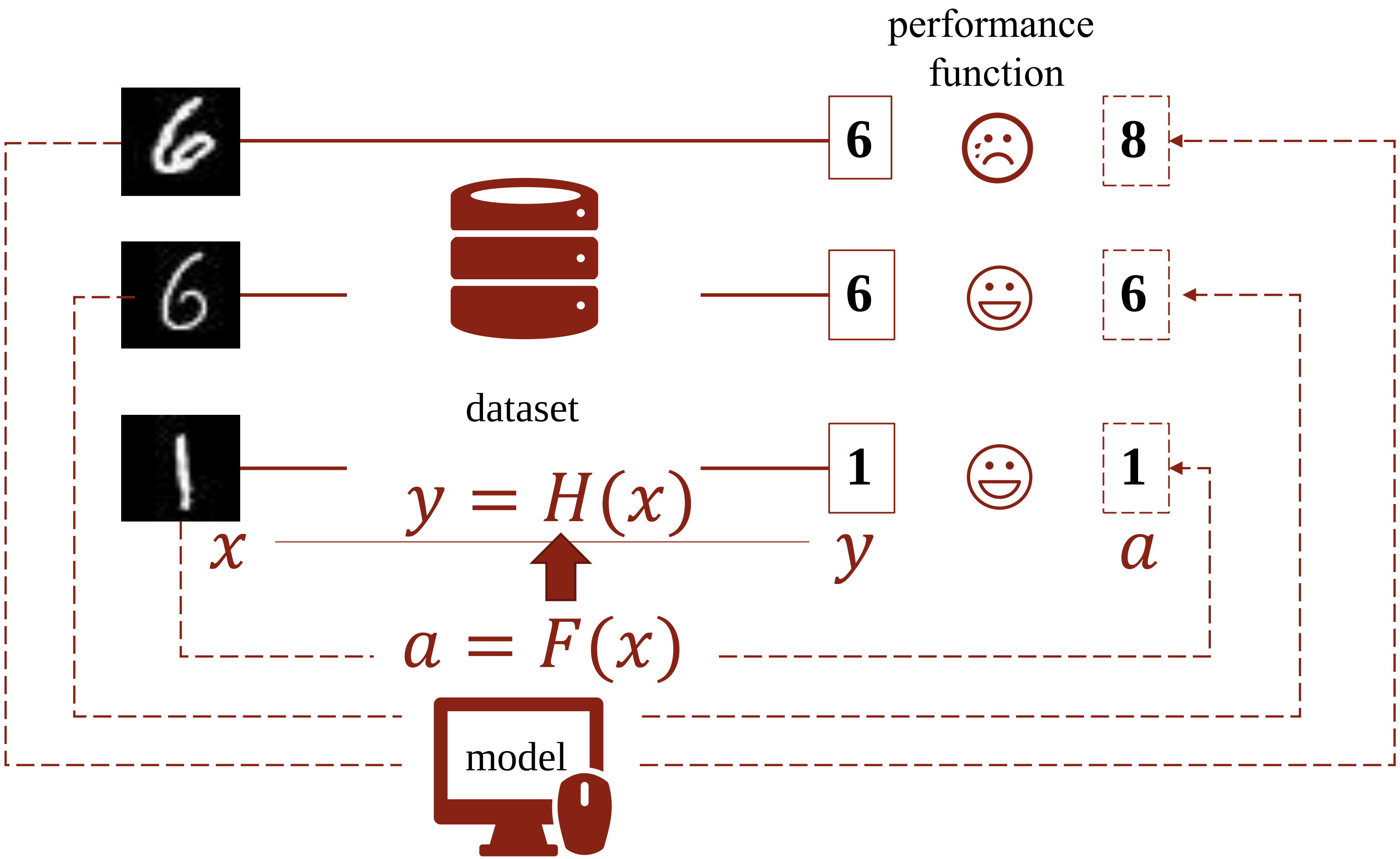
What is machine learning



Machine Learning

- Task: handwritten digit recognition
- Model
- Dataset: $D = \{(x, y)\}$
- Performance function: measure the learning

What is machine learning

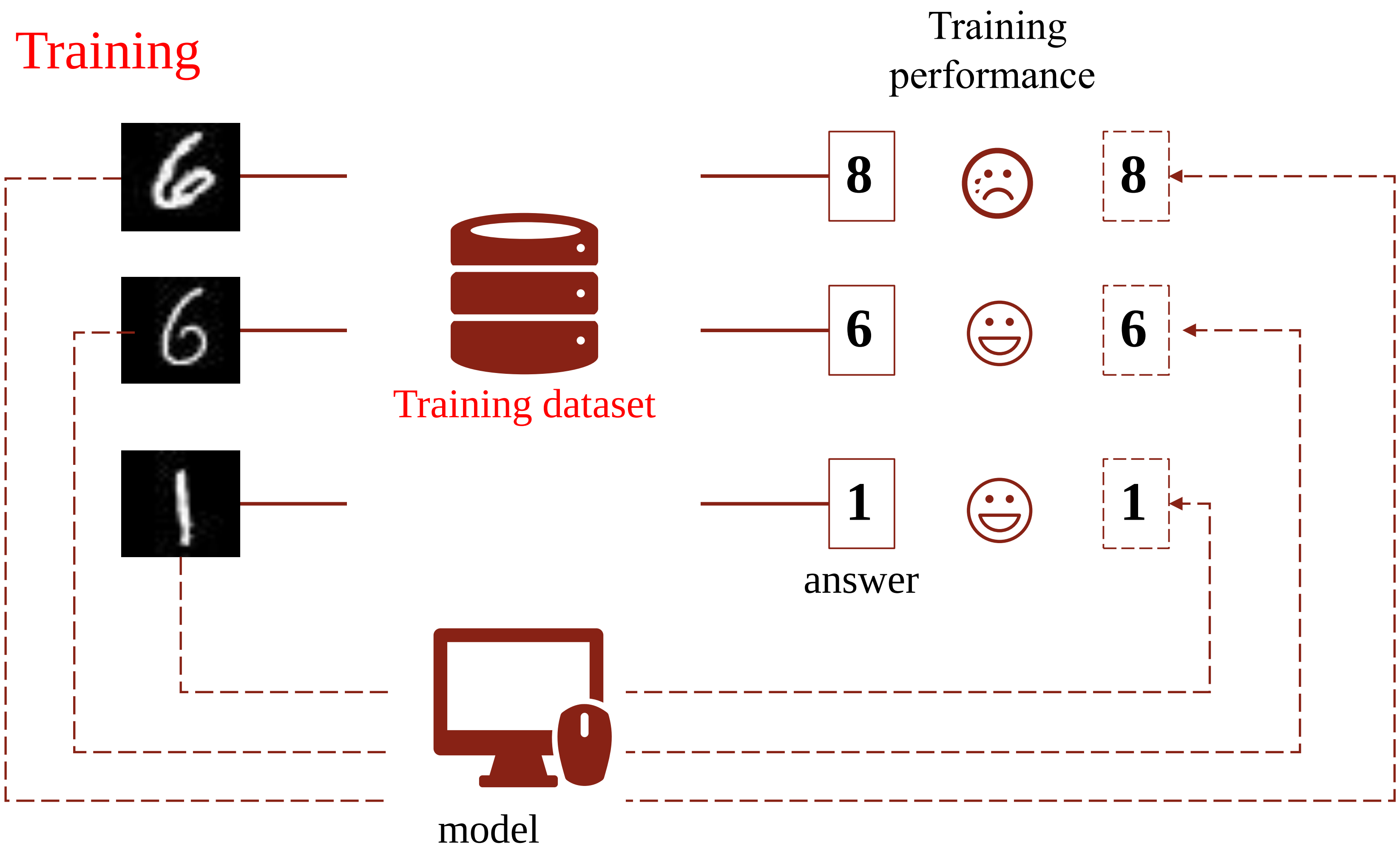


Machine Learning

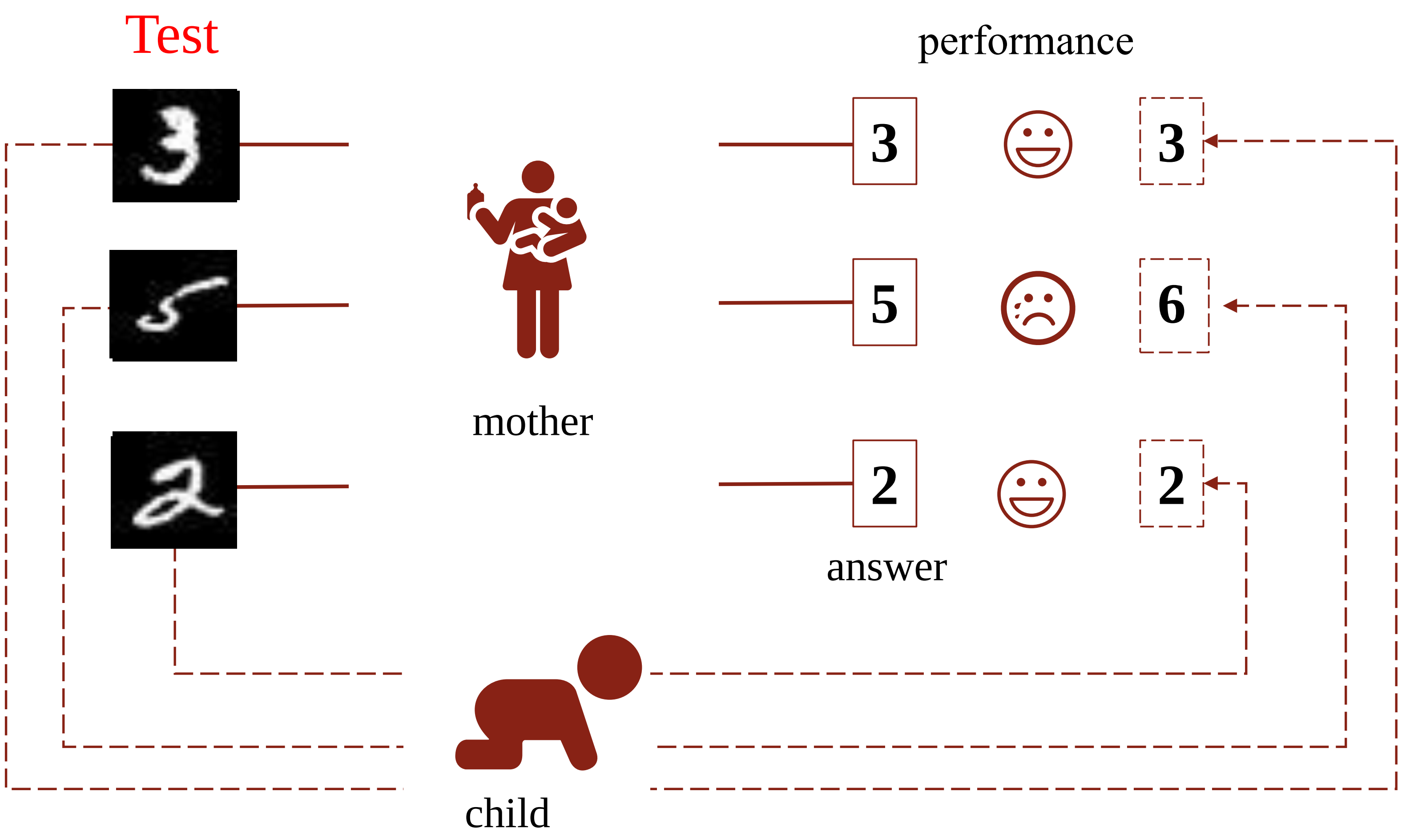
- Task: handwritten digit recognition
- Model: $a = F(x)$
- Dataset: D={ (x,y)}
- Performance function: $J(y, a)$

Given a task T and dataset D, machine learning is to approximate the unknown function $y = H(x)$ by updating the model $a = F(x)$ to improving the performance function $J(y, a)$ on task T

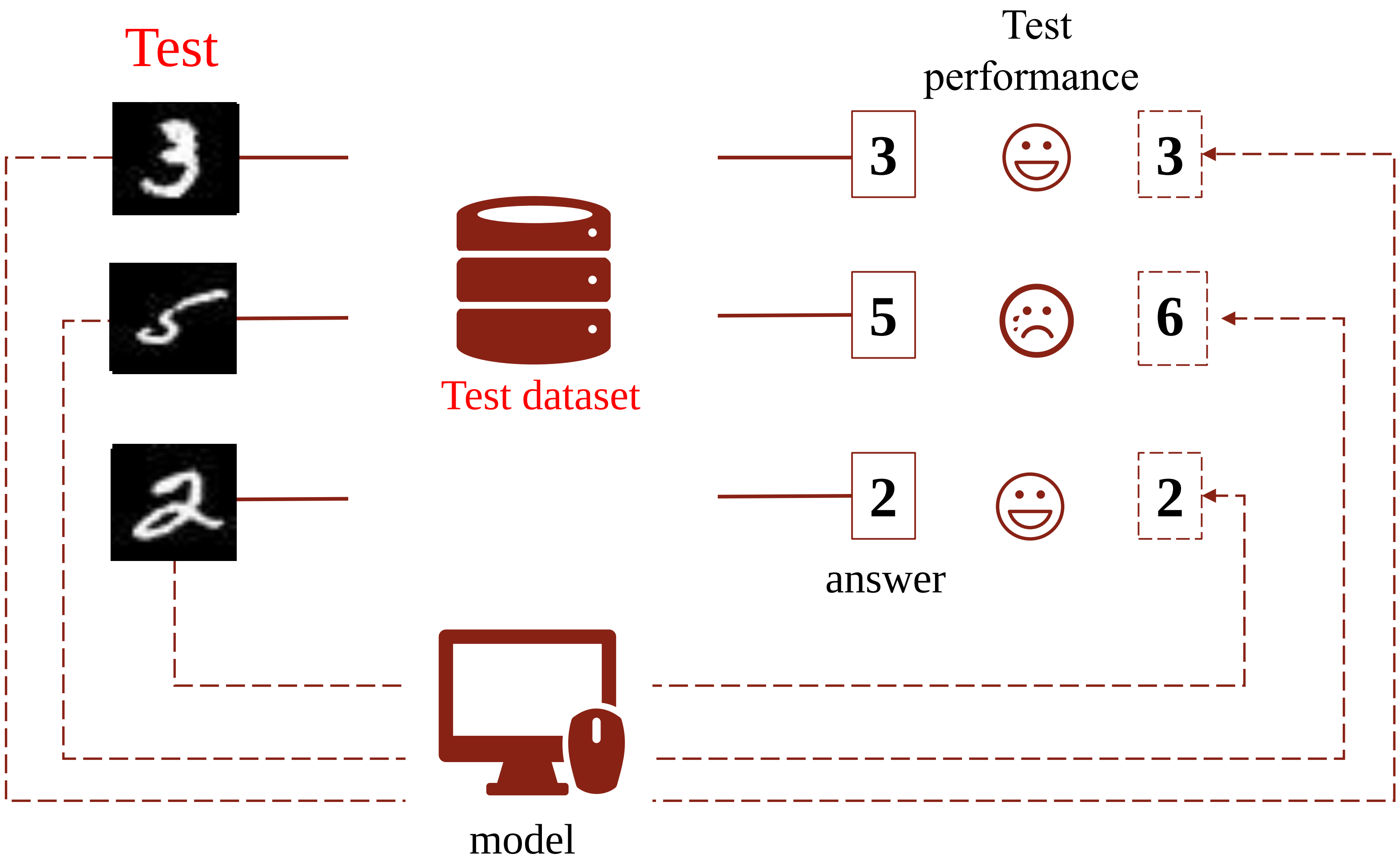
The workflow of machine learning



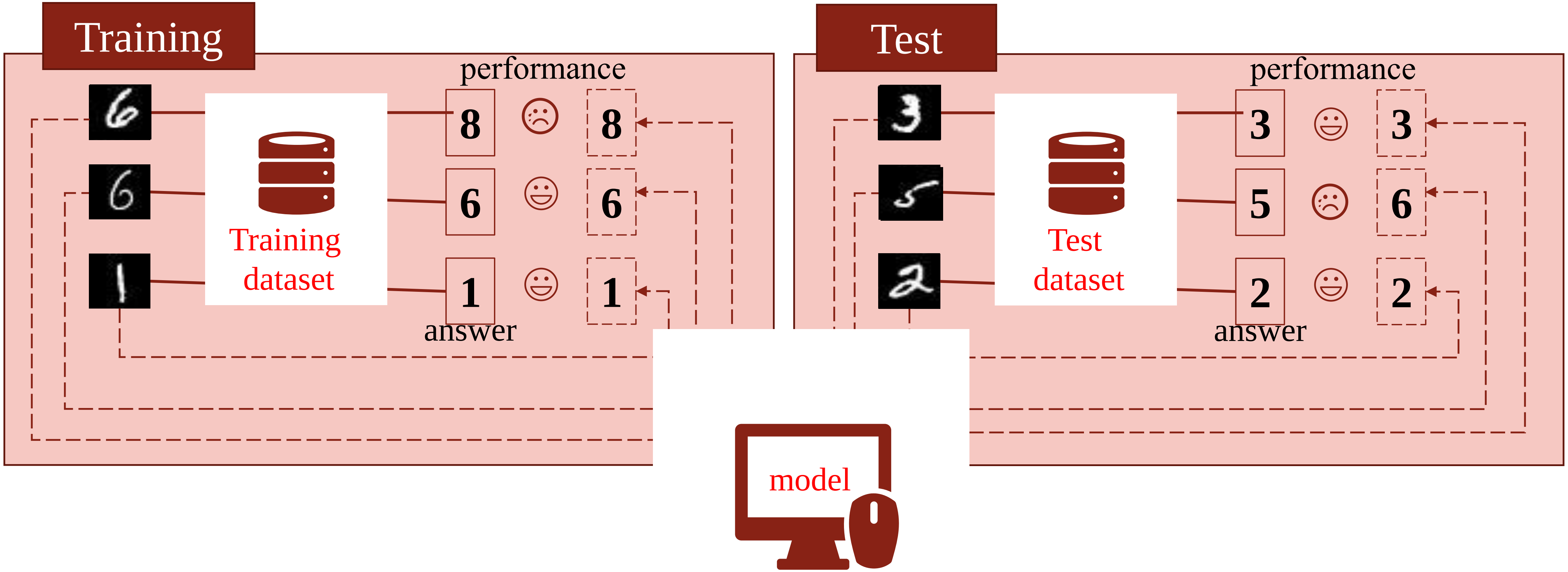
The workflow of machine learning



The workflow of machine learning

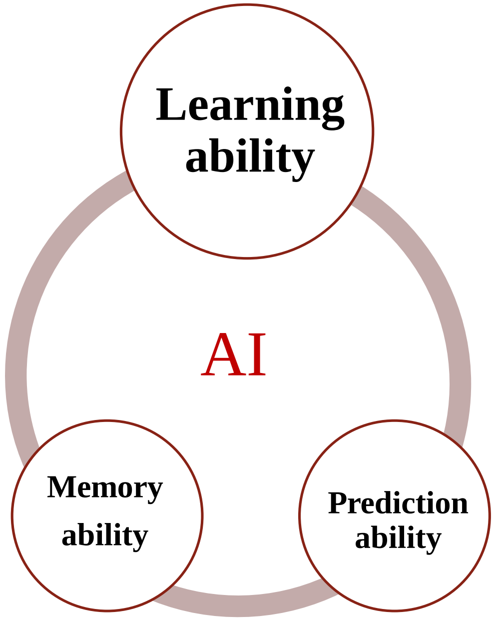


The workflow of machine learning



machine learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning



Supervised Learning

- **The dataset contains the true answer(label):**

  $D=\{ (x,y)\}$

- **Classification**
- **Regression**

Unsupervised Learning

Reinforcement Learning

Outline

- Brief review
- Some concepts of Machine Learning
- Univariate linear regression
- Basics of Python

Outline

- Brief review
- Some concepts of Machine Learning
- Univariate linear regression
  - An Example
  - Least Squares Approximations
  - Steepest Gradient Descend Method
- Basics of Python

Univariate linear regression

Univariate linear regression

Regression

Regression

Given a dataset

$$D = \{(x \in R^n, y \in R)\},$$

regression is to learn a function

$$a = F(x),$$

where the output of the model a is close to the real label y.

- Dataset: $D = \{(x \in R^n, y \in R)\}$
- Model: $a = F(x)$

Machine Learning

- Task T
- Model: $a = F(x)$
- Dataset: D={ (x,y)}
- Performance function:

$$J(y, a)$$

Given a task T and dataset D, machine learning is to approximate the unknown function $y = H(x)$ by updating the model $a = F(x)$ to improving the performance function $J(y, a)$ on task T

Regression

Regression

Given a dataset

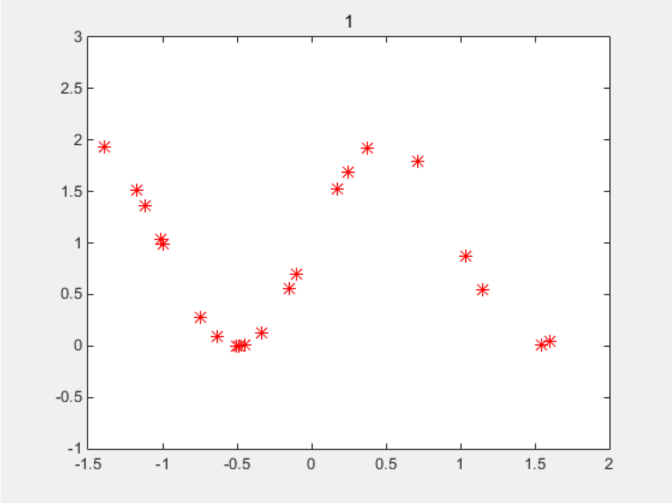
$$D = \{(x \in R^n, y \in R)\},$$

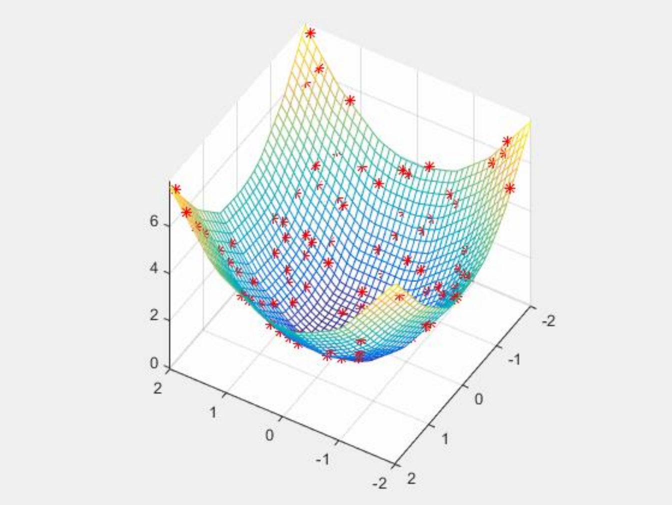
regression is to learn a function

$$a = F(x),$$

where the output of the model a is close to the real label y.

- Dataset: $D = \{(x \in R^n, y \in R)\}$
- Model: $a = F(x)$





Regression

Regression

Given a dataset

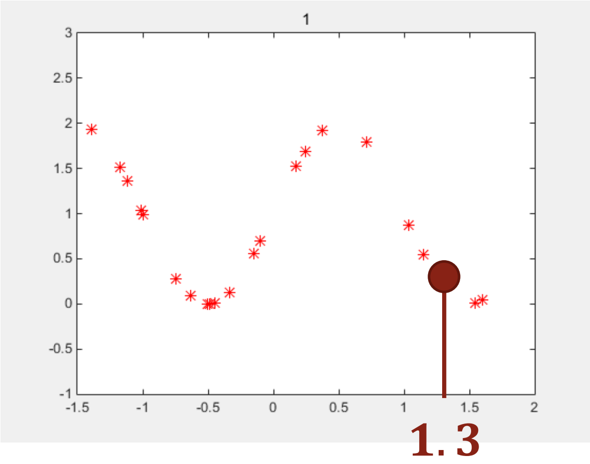
$$D = \{(x \in R^n, y \in R)\},$$

regression is to learn a function

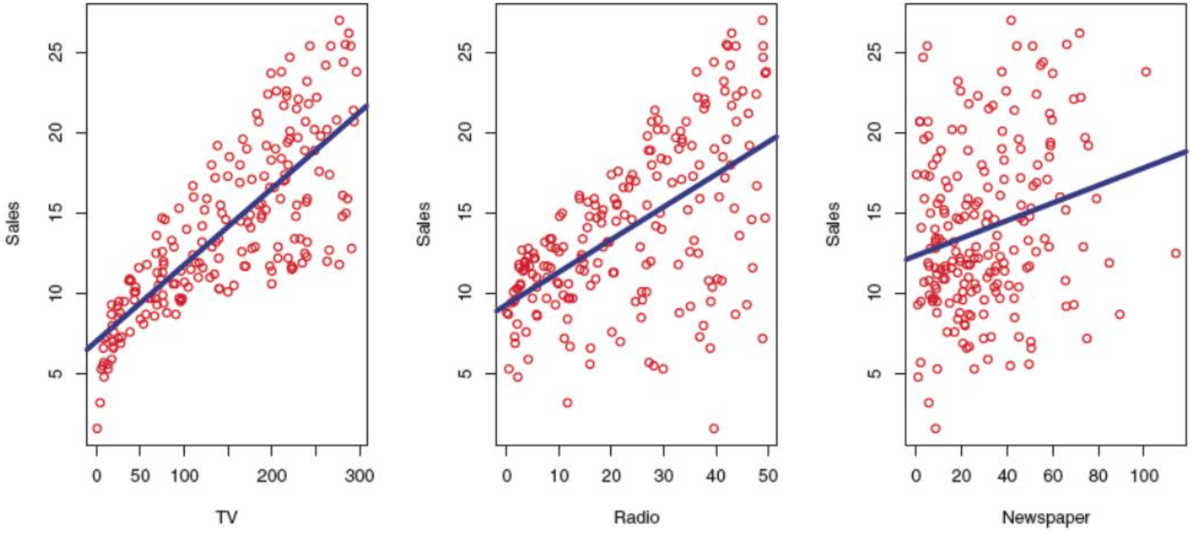
$$a = F(x),$$

where the output of the model a is close to the real label y.

- Dataset: $D = \{(x \in R^n, y \in R)\}$
- Model: $a = F(x)$



1. Predict real value for new point
2. interpret the relation of variables



Linear Regression

Regression

Given a dataset

$$D = \{(x \in R^n, y \in R)\},$$

regression is to learn a function

$$a = F(x),$$

where the output of the model a is close to the real label y.

- Dataset: $D = \{(x \in R^n, y \in R)\}$
- Model: $a = F(x)$

Linear Regression

Given a dataset

$$D = \{(x \in R^n, y \in R)\},$$

linear regression is to learn a linear function

$$\boldsymbol{a = wx + b}$$

where the output of the model a is close to the real label y.

- Dataset: $D = \{(x \in R^n, y \in R)\}$
- Linear Model: $a = \boldsymbol{wx + b}$

Linear Regression

Linear Regression

Given a dataset

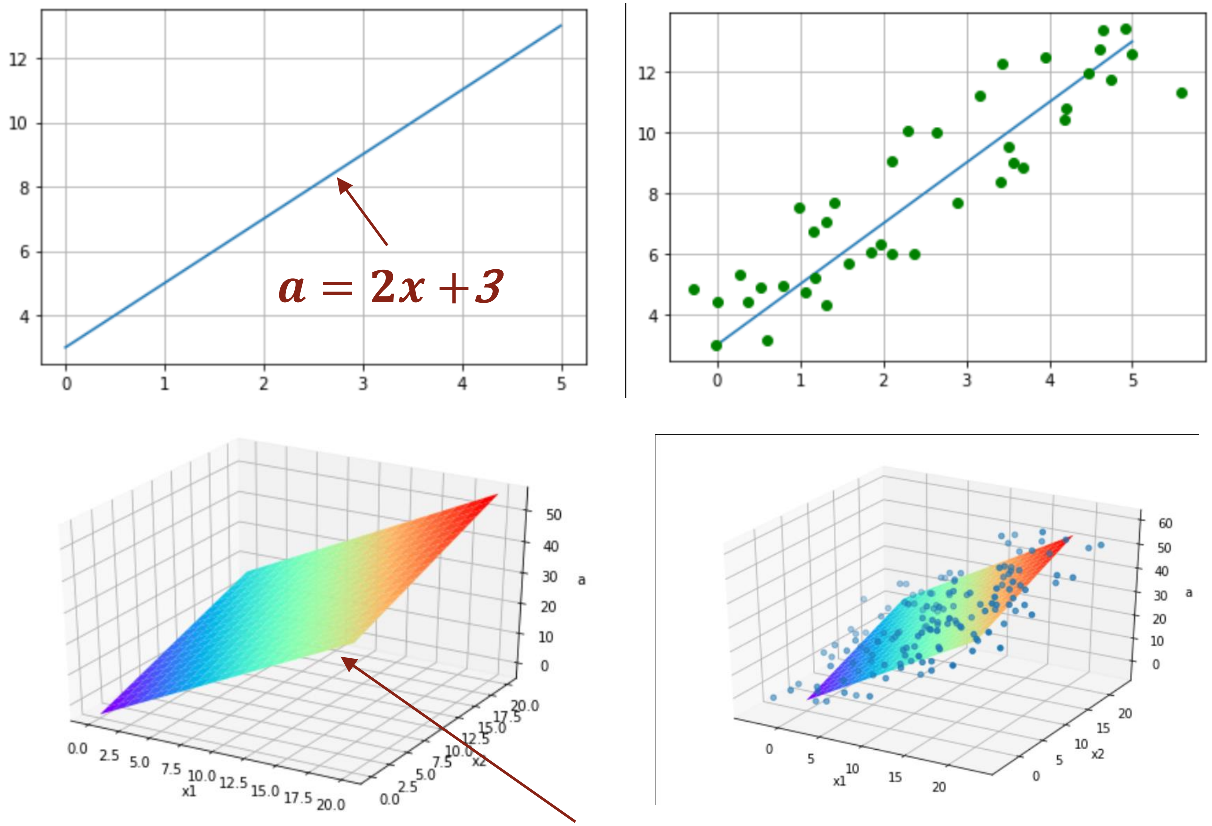
$$D = \{(x \in R^n, y \in R)\},$$

linear regression is to learn a linear function

$$\boldsymbol{a = wx + b}$$

where the output of the model a is close to the real label y.

- Dataset: $D = \{(x \in R^n, y \in R)\}$
- Linear Model: $a = \boldsymbol{wx + b}$



$$\boldsymbol{a = 2x_1 + x_2 - 4}$$

$$\boldsymbol{a = [2, 1] \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 4}$$

Univariate Linear Regression

Univariate Linear Regression

Given a dataset

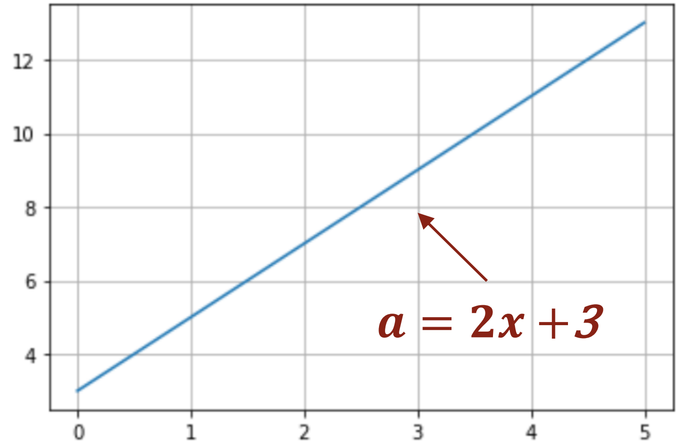
$$D = \{(x \in R, y \in R)\},$$

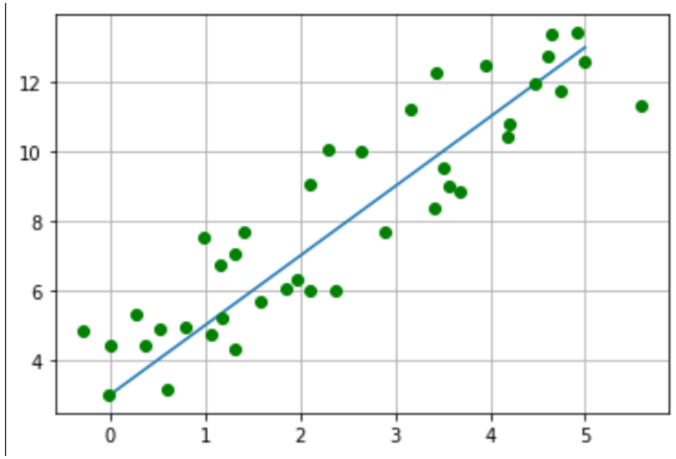
univariate linear regression is to learn a linear function

$$\boldsymbol{a} = \boldsymbol{w}\boldsymbol{x} + \boldsymbol{b}$$

where the output of the model a is close to the real label y.

- Dataset: $D = \{(x \in R, y \in R)\}$
- Linear Model: $a = \boldsymbol{w}\boldsymbol{x} + \boldsymbol{b}$





Univariate Linear Regression

Univariate Linear Regression

Given a dataset

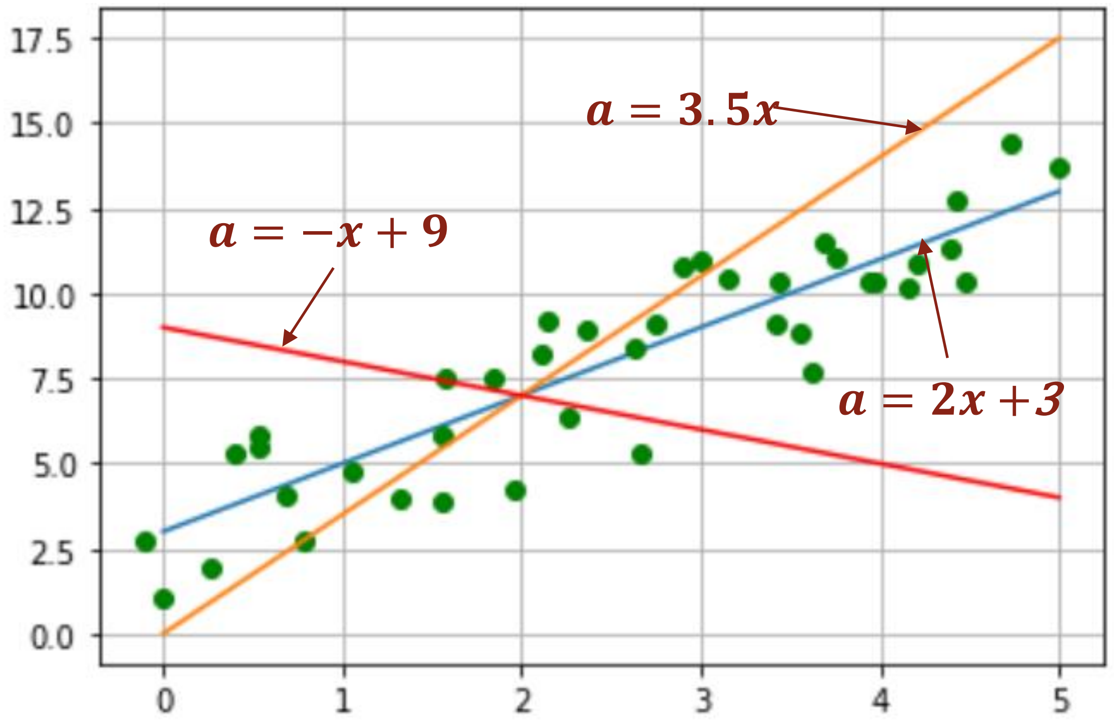
$$D = \{(x \in R, y \in R)\},$$

univariate linear regression is to learn a linear function

$$\boldsymbol{a = wx + b}$$

where the output of the model a is close to the real label y.

- Dataset: $D = \{(x \in R, y \in R)\}$
- Linear Model: $a = F(x)$



How to find the appropriate w and b ?

An Example

Please predict the price for a house with 5 square!!!

square	price
1.156762	3.949326
2.624116	1.746431
2.943006	9.902035
2.499967	5.326710
3.530516	10.569117
4.045524	12.493749
5.607250	14.531507
5.784322	15.758228
7.016050	12.235891
8.304229	12.536069
7.351775	19.349313
8.799763	18.347272
9.346700	18.812099
10.232547	19.750414
11.872116	24.672962

An Example

Univariate Linear Regression

- Data:
  - Sample: $(x \in R, y \in R)$, where x is square and y is the price.
  - Dataset: $D = \{(x^i \in R, y^i \in R) | i \in [1, m]\}$, where m=10 is the number of the samples.
- Linear Model:

$$a = wx + b$$

- Object: For each x^i, let a^i is close to y^i

?

square	price
1.156762	3.949326
2.624116	1.746431
2.943006	9.902035
2.499967	5.326710
3.530516	10.569117
4.045524	12.493749
5.607250	14.531507
5.784322	15.758228
7.016050	12.235891
8.304229	12.536069
7.351775	19.349313
8.799763	18.347272
9.346700	18.812099
10.232547	19.750414
11.872116	24.672962

D

$(x \in R, y \in R)$

An Example

Univariate Linear Regression

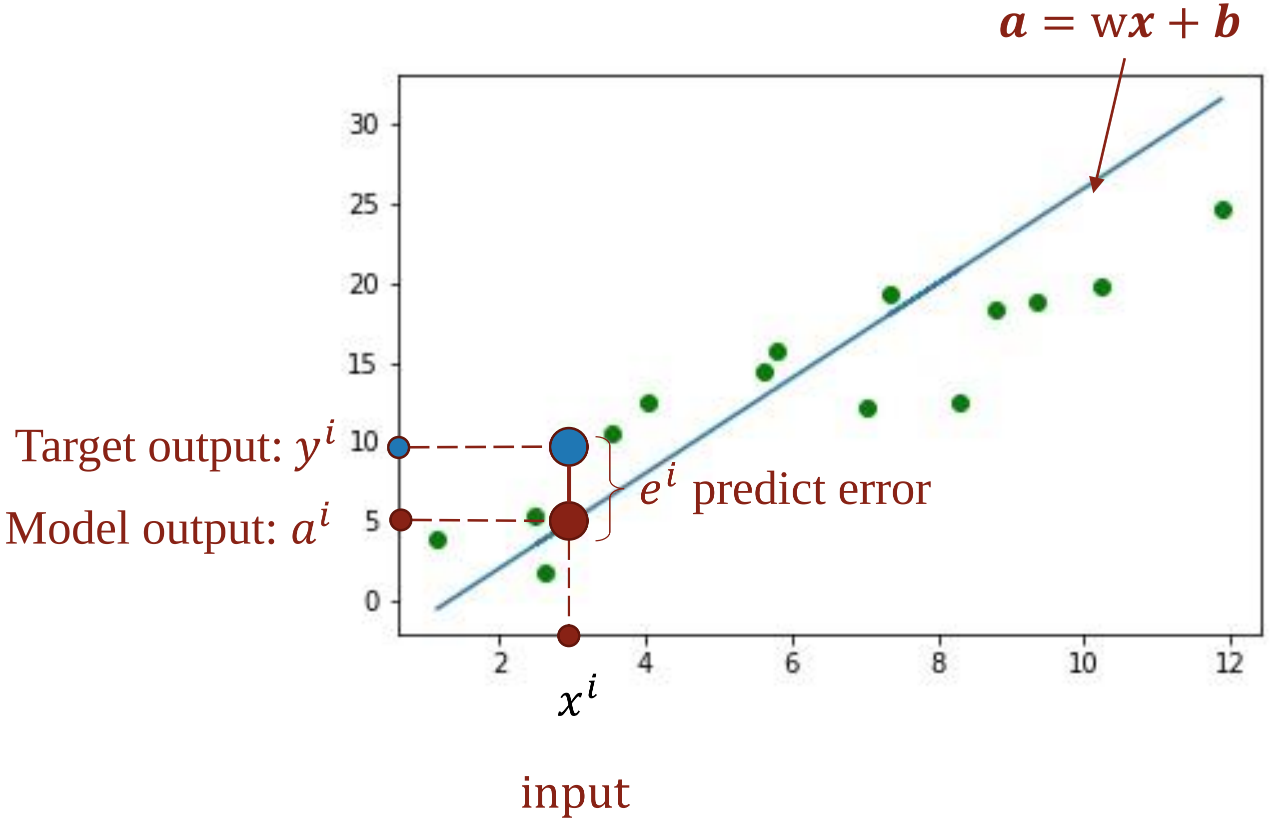
- Data:
  - Sample: $(x \in R, y \in R)$, where x is square and y is the price.
  - Dataset: $D = \{(x^i \in R, y^i \in R) | i \in [1, m]\}$, where m=10 is the number of the samples.
- Linear Model:

$$a = wx + b$$

- Object: For each x^i, let a is close to y^i

For each (x^i, y^i), let e^i as small as possible

?



An Example

Univariate Linear Regression

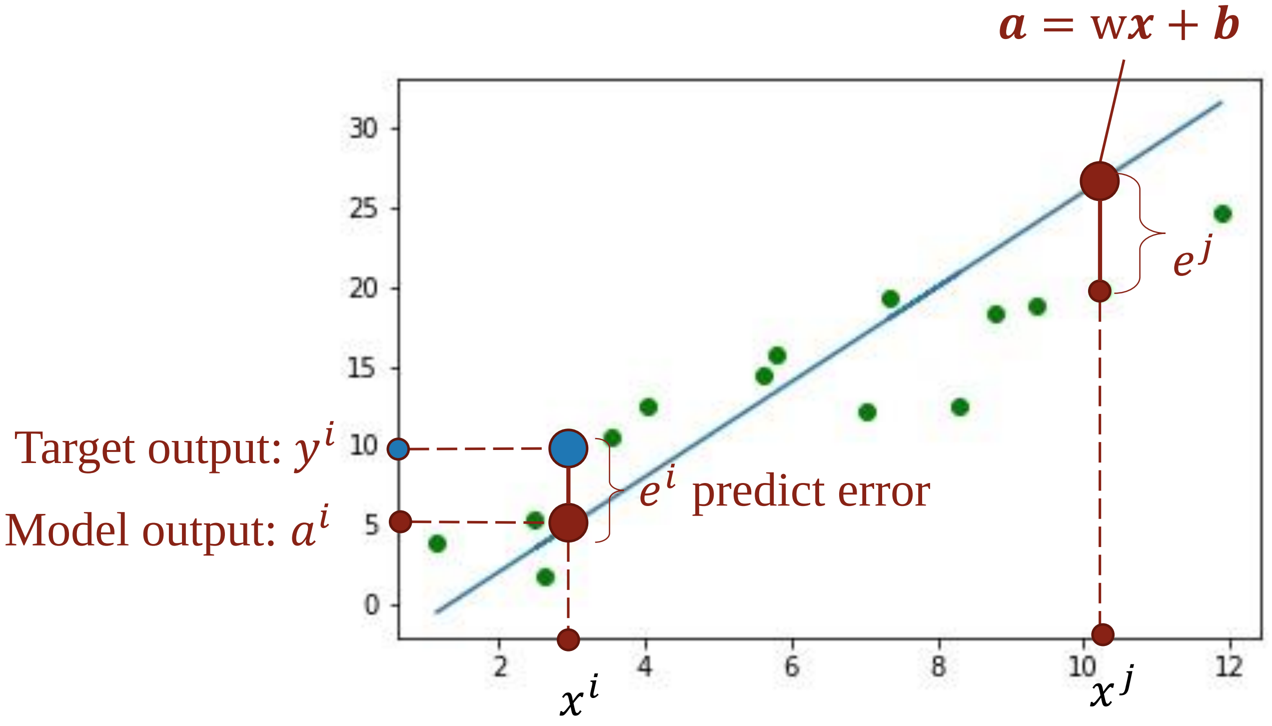
- Data:
  - Sample: $(x \in R, y \in R)$, where x is square and y is the price.
  - Dataset: $D = \{(x^i \in R, y^i \in R) | i \in [1, m]\}$, where m=10 is the number of the samples.
- Linear Model:

$$a = wx + b$$

- Object: For each x^i, let a is close to y^i

For each (x^i, y^i), let e^i as small as possible

$$e^i = \frac{1}{2}(a^i - y^i)^2$$



input

$$e^i = (a^i - y^i) \qquad e^i = |a^i - y^i| \qquad e^i = \sqrt{(a^i - y^i)^2}$$

Cost function

Univariate Linear Regression

- Data:
  - Sample: $(x \in R, y \in R)$, where x is square and y is the price.
  - Dataset: $D = \{(x^i \in R, y^i \in R) | i \in [1, m]\}$, where m=10 is the number of the samples.
- Linear Model:

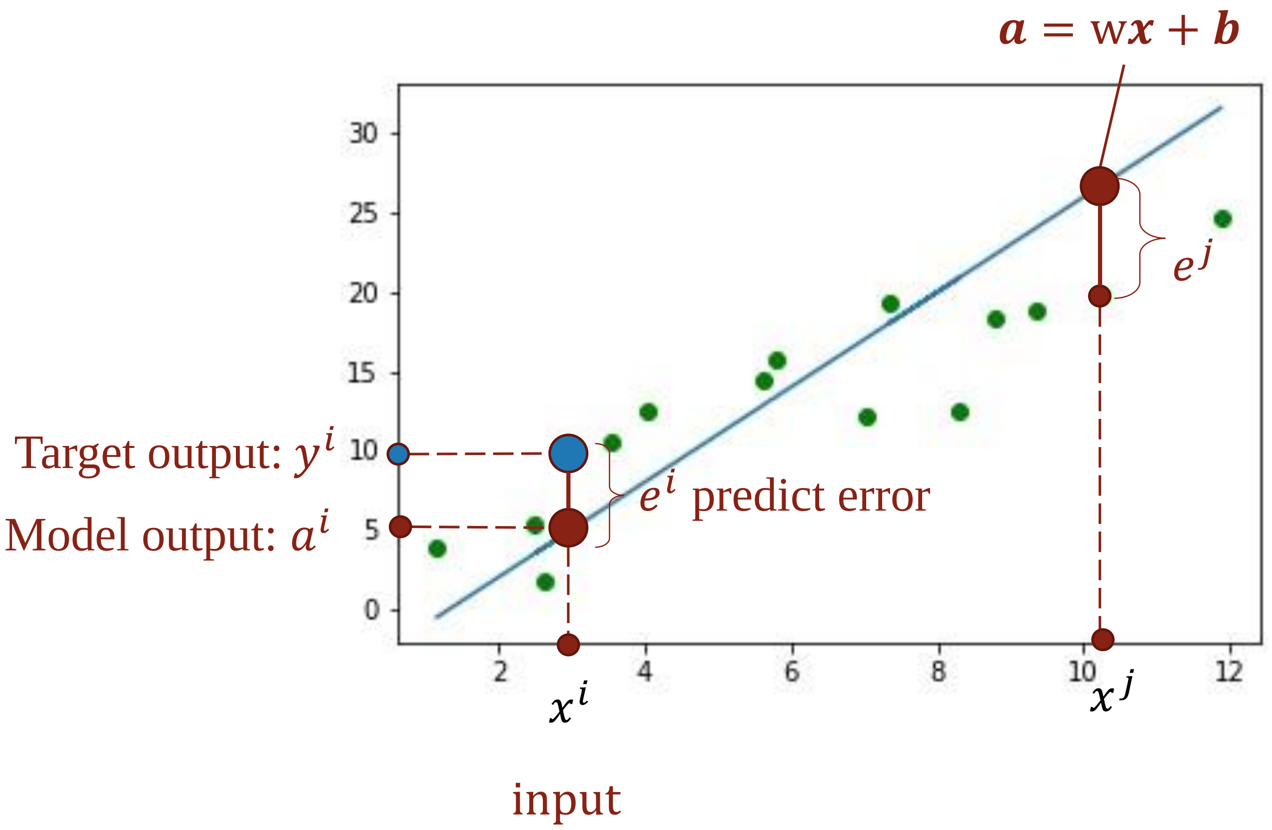
$$a = wx + b$$

- Object: For each x^i, let a is close to y^i

For each (x^i, y^i), let e^i as small as possible

$$e^i = \frac{1}{2}(a^i - y^i)^2$$

Given D, let cost function $J = \frac{1}{2m}\sum_{i=1}^{m}(a^i - y^i)^2$ as small as possible



Cost function

Univariate Linear Regression

- Data:
  - Sample: $(x \in R, y \in R)$, where x is square and y is the price.
  - Dataset: $D = \{(x^i \in R, y^i \in R) | i \in [1, m]\}$, where m=10 is the number of the samples.
- Linear Model:

$$a = wx + b$$

- Object: For each x^i, let a is close to y^i

For each (x^i, y^i), let e^i as small as possible

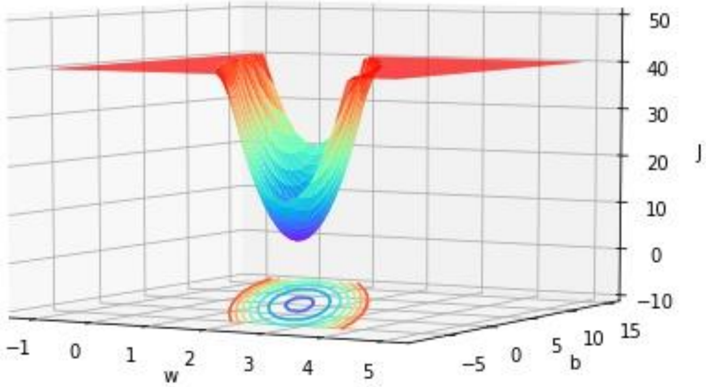
$$e^i = \frac{1}{2}\left(a^i - y^i\right)^2$$

Given D, let cost function $J = \frac{1}{2m}\sum_{i=1}^{m}(a^i - y^i)^2$ as small as possible

cost function: $$J = \frac{1}{2m}\sum_{i=1}^{m}(a^i - y^i)^2 \quad \boxed{(x, \quad y) \in D}$$

$$= \frac{1}{2m}\sum_{i=1}^{m}(w x^i + b - y^i)^2$$

variables



$$\underset{w,b}{\operatorname{argmin}} \frac{1}{2m}\sum_{i=1}^{m}(a^i - y^i)^2 = \underset{w,b}{\operatorname{argmin}} J(w, b)$$

Cost function

Univariate Linear Regression

- Data:
  - Sample: $(x \in R, y \in R)$, where x is square and y is the price.
  - Dataset: $D = \{(x^i \in R, y^i \in R) | i \in [1, m]\}$, where m=10 is the number of the samples.
- Linear Model:

$$a = wx + b$$

- Object:

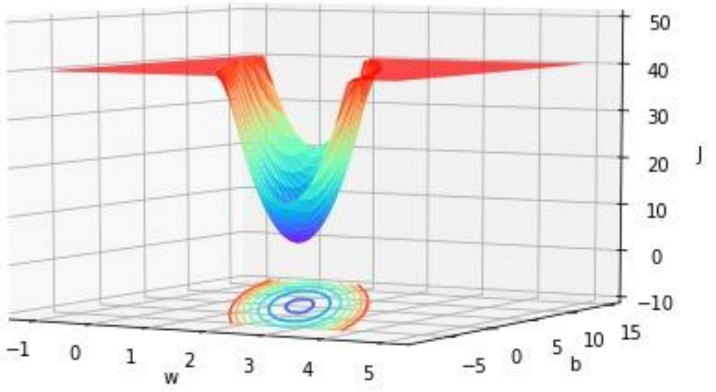
$$\underset{w,b}{\operatorname{argmin}} J(w, b) = \underset{w,b}{\operatorname{argmin}} \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$$

?

cost function: $J = \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$ $\boxed{(x, \quad y) \in D}$

$$= \frac{1}{2m} \sum_{i=1}^{m} (w x^i + b - y^i)^2$$

variables



How to find minimum point *(w,b)* of cost function *J*

minimum point

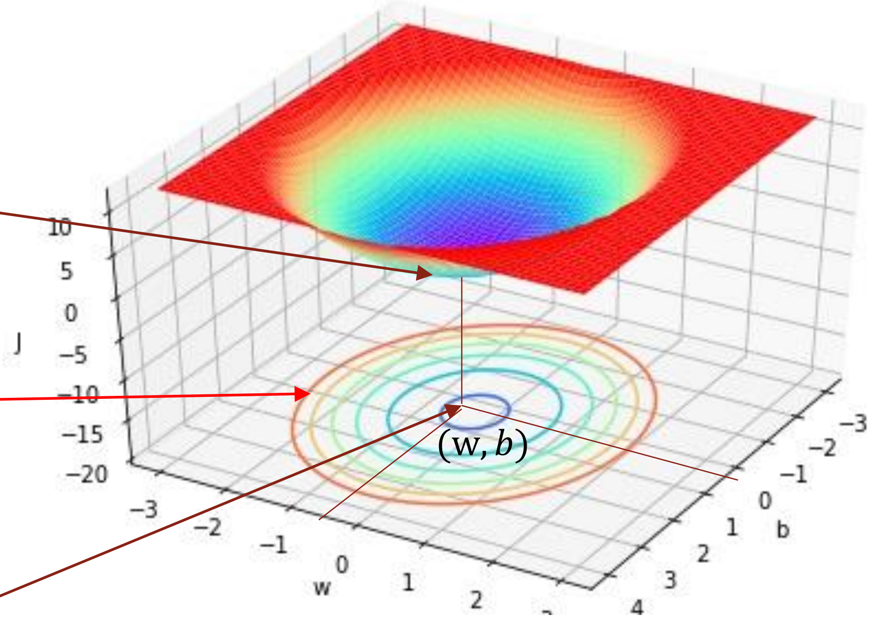
General Nonlinear function $J(w,b)$, (w^*,b^*) is a minimum point if

$$J(w^*,b^*) \le J(w,b)$$

for any (w,b) that very close to(w^*,b^*).

Minima

Contour



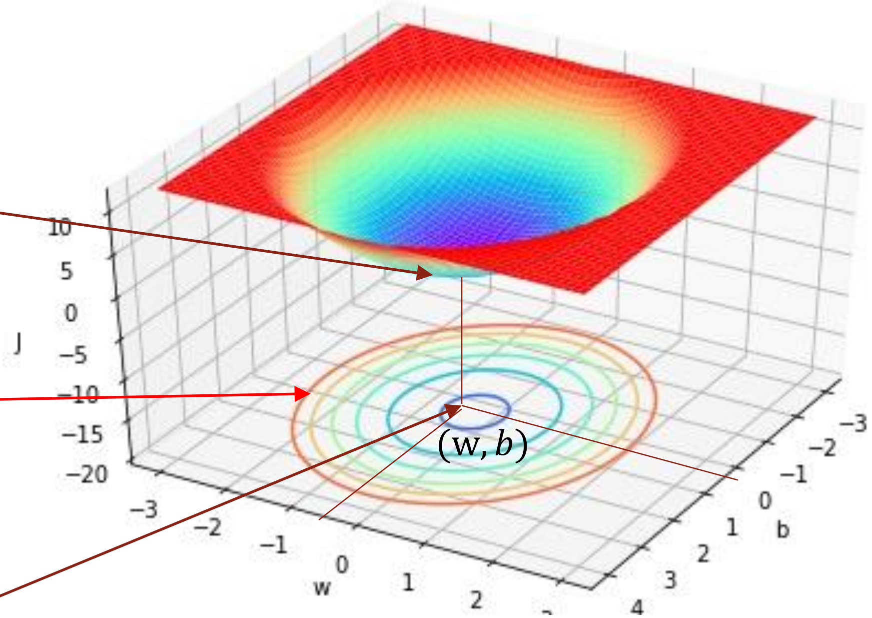
Minimum point

minimum point

General Nonlinear function $J(w,b)$, (w^*,b^*) is a minimum point if
$$J(w^*,b^*) \le J(w,b)$$
for any (w,b) that very close to (w^*,b^*).

Minima

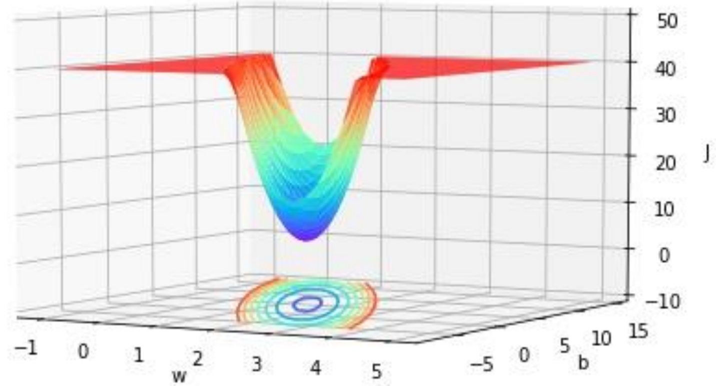
Contour



Minimum point

cost function: $J = \dfrac{1}{2m}\sum_{i=1}^{m}(a^i - y^i)^2$

$$= \frac{1}{2m}\sum_{i=1}^{m}(w\,x^i + b - y^i)^2$$



How to find minimum point *(w,b)* of cost function *J*

minimum point

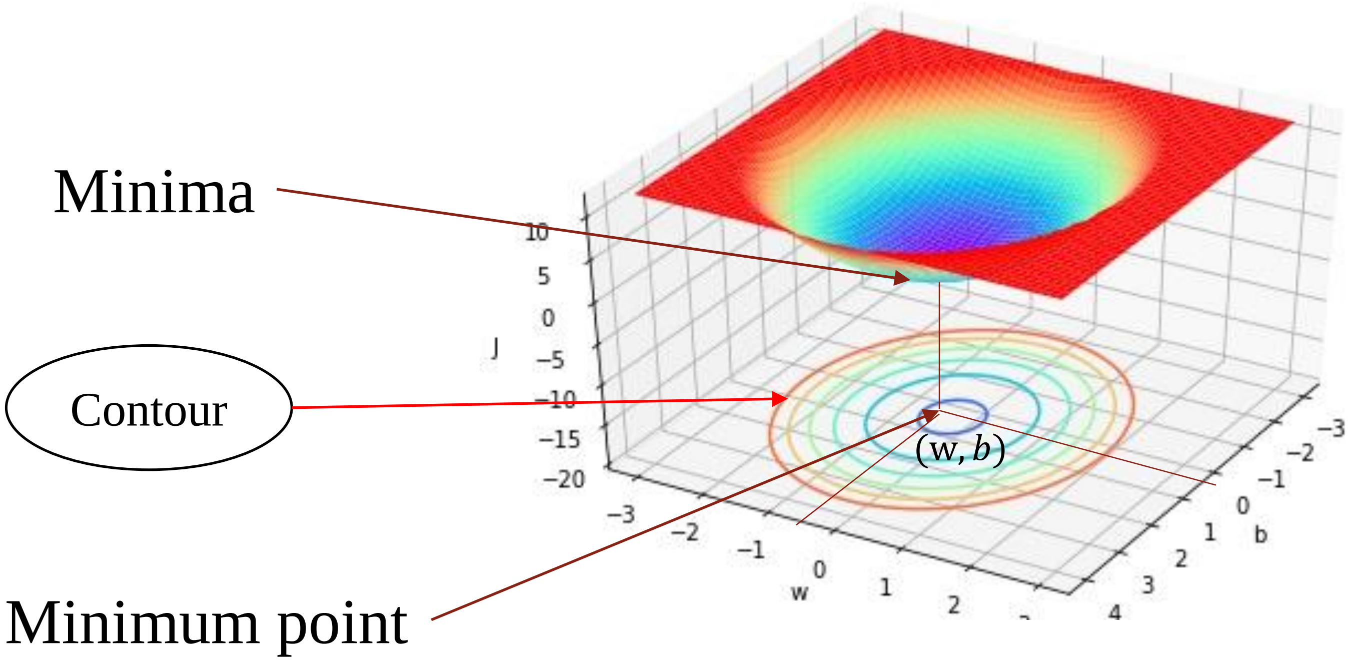
General Nonlinear function $J(w, b)$, (w^*, b^*) is a minimum point if

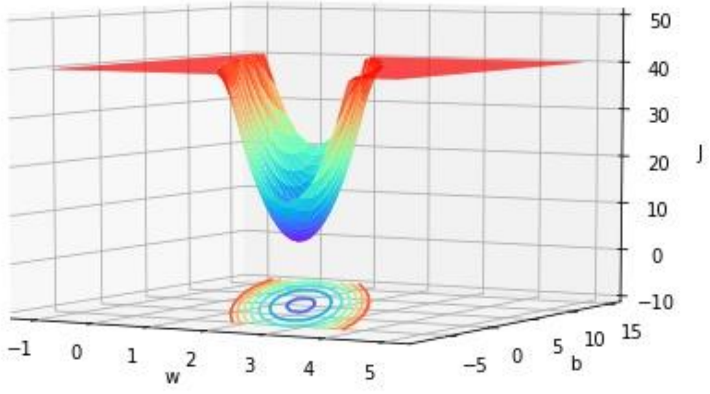
$$J(w^*, b^*) \le J(w, b)$$

for any (w, b) that very close to (w^*, b^*).

For a differentiable function $J\ (w, b)$, if (w^*, b^*) is a minimum point of J, then the following equation holds:

$$\begin{cases} \left.\dfrac{\partial J}{\partial w}\right|_{w=w^*} = 0 \\ \left.\dfrac{\partial J}{\partial b}\right|_{b=b^*} = 0 \end{cases}$$





Least Squares Approximations

$$\frac{\partial J}{\partial w} = \frac{1}{2m}\sum_{i=1}^{m}\frac{\partial(w\,x^i+b-y^i)^2}{\partial(w\,x^i+b-y^i)}\cdot\frac{\partial(w\,x^i+b-y^i)}{\partial w}$$

$$\frac{\partial J}{\partial w} = \frac{1}{m}\sum_{i=1}^{m}(wx^i+b-y^i)\cdot x^i \qquad \mathbf{=0}$$

$$\frac{\partial J}{\partial b} = \frac{1}{2m}\sum_{i=1}^{m}\frac{\partial(w\,x^i+b-y^i)^2}{\partial(w\,x^i+b-y^i)}\cdot\frac{\partial(w\,x^i+b-y^i)}{\partial b}$$

$$\frac{\partial J}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}(wx^i+b-y^i) \qquad \mathbf{=0}$$

For a differentiable function $J\,(w, b)$, if (w^*, b^*) is a minimum point of J, then the following equation holds:

$$\begin{cases}\left.\dfrac{\partial J}{\partial w}\right|_{w=w^*} = 0\\[2ex] \left.\dfrac{\partial J}{\partial b}\right|_{b=b^*} = 0\end{cases}$$

cost function: $$J = \frac{1}{2m}\sum_{i=1}^{m}(a^i-y^i)^2$$

$$= \frac{1}{2m}\sum_{i=1}^{m}(w\,x^i+b-y^i)^2$$

Least Squares Approximations

$$\frac{1}{m}\sum_{i=1}^{m}(wx^i + b - y^i) \qquad = 0$$

$$\sum_{i=1}^{m}\left(wx^i - y^i\right) + \underbrace{\sum_{i=1}^{m} b}_{mb} \qquad = 0$$

$$b = \frac{1}{m}\sum_{i=1}^{m} y^i - w\frac{1}{m}\sum_{i=1}^{m} x^i$$

$$\frac{\partial J}{\partial w} = \frac{1}{m}\sum_{i=1}^{m}(wx^i + b - y^i)\cdot x^i$$

$$\frac{\partial J}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}(wx^i + b - y^i)$$

Least Squares Approximations

$$\frac{1}{m}\sum_{i=1}^{m}\left(wx^i + b - y^i\right)\cdot x^i = 0$$

$$b = \frac{1}{m}\sum_{i=1}^{m} y^i - w\frac{1}{m}\sum_{i=1}^{m} x^i$$

$$\sum_{i=1}^{m}\left[wx^i + \frac{1}{m}\sum_{j=1}^{m} y^j - w\frac{1}{m}\sum_{j=1}^{m} x^j - y^i\right]\cdot x^i = 0$$

$$\sum_{i=1}^{m}\left[w\left(x^i - \frac{1}{m}\sum_{j=1}^{m} x^j\right) + \frac{1}{m}\sum_{j=1}^{m} y^j - y^i\right]\cdot x^i = 0$$

$$w\left(\sum_{i=1}^{m} x^i\, x^i - \underline{\sum_{i=1}^{m}\frac{1}{m}\sum_{j=1}^{m} x^j\, x^i}\right) = \sum_{i=1}^{m} y^i\cdot x^i - \underline{\sum_{i=1}^{m}\frac{1}{m}\sum_{j=1}^{m} y^j\, x^i}$$

$$\frac{1}{m}\sum_{i=1}^{m}\left[x^i\sum_{j=1}^{m} x^j\right] = \frac{1}{m}\left(\sum_{i=1}^{m} x^i\right)^2 \qquad \frac{1}{m}\sum_{i=1}^{m}\left[x^i\sum_{j=1}^{m} y^j\right]$$

$$\boxed{\begin{aligned}
b &= \frac{1}{m}\sum_{i=1}^{m} y^i - w\frac{1}{m}\sum_{i=1}^{m} x^i\\
w &= \frac{\sum_{i=1}^{m} y^i\cdot x^i - \frac{1}{m}\sum_{i=1}^{m}\left[x^i\sum_{j=1}^{m} y^j\right]}{\sum_{i=1}^{m} x^i\, x^i - \frac{1}{m}\left(\sum_{i=1}^{m} x^i\right)^2}\\
&\Downarrow \begin{cases}\bar{y} = \frac{1}{m}\sum_{i=1}^{m} y^i\\ \bar{x} = \frac{1}{m}\sum_{i=1}^{m} x^i\end{cases}\\
b &= \bar{y} - w\bar{x}\\
w &= \frac{\sum_{i=1}^{m} y^i\cdot x^i - m\bar{y}\bar{x}}{\sum_{i=1}^{m} x^i\, x^i - m\bar{x}^2}
\end{aligned}}$$

Least Squares Approximations

Univariate Linear Regression

- Data:
  - Sample: $(x \in R, y \in R)$, where x is square and y is the price.
  - Dataset: $D = \{(x^i \in R, y^i \in R) | i \in [1, m]\}$, where m=10 is the number of the samples.
- Linear Model:

$$a = wx + b$$

- Object:

$$\underset{w,b}{\operatorname{argmin}} J(w, b) = \underset{w,b}{\operatorname{argmin}} \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$$

$$b = \frac{1}{m} \sum_{i=1}^{m} y^i - w \frac{1}{m} \sum_{i=1}^{m} x^i$$

$$w = \frac{\sum_{i=1}^{m} y^i \cdot x^i - \frac{1}{m} \sum_{i=1}^{m} \left[x^i \sum_{j=1}^{m} y^j\right]}{\sum_{i=1}^{m} x^i x^i - \frac{1}{m} \left(\sum_{i=1}^{m} x^i\right)^2}$$

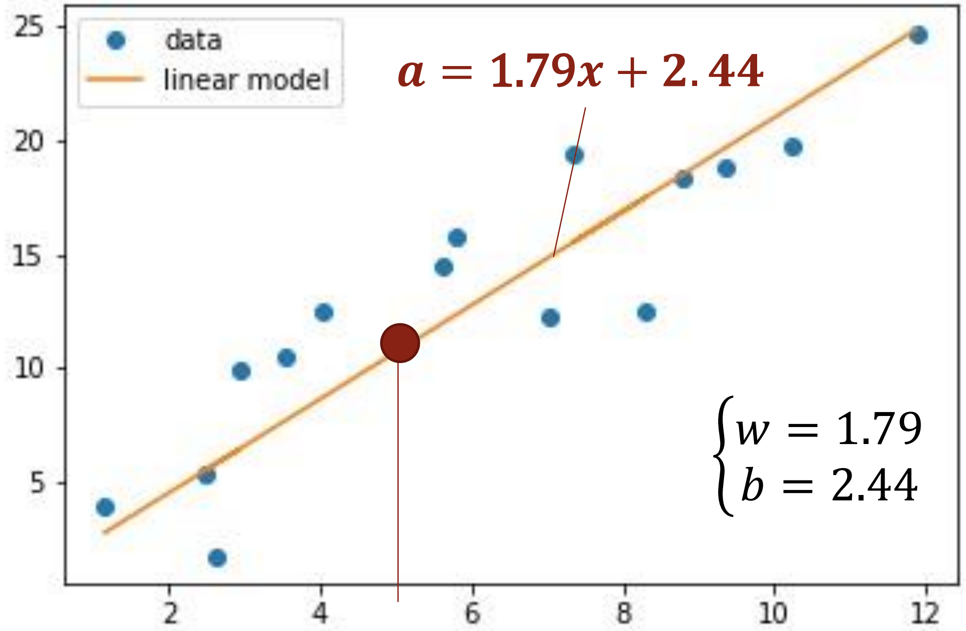
$$\Bigg\downarrow \begin{cases} \bar{y} = \frac{1}{m} \sum_{i=1}^{m} y^i \\ \bar{x} = \frac{1}{m} \sum_{i=1}^{m} y^i \end{cases}$$

$$b = \bar{y} - w\bar{x}$$

$$w = \frac{\sum_{i=1}^{m} y^i \cdot x^i - m\bar{y}\bar{x}}{\sum_{i=1}^{m} x^i x^i - m\bar{x}^2}$$

Least Squares Approximations

square	price
1.156762	3.949326
2.624116	1.746431
2.943006	9.902035
2.499967	5.326710
3.530516	10.569117
4.045524	12.493749
5.607250	14.531507
5.784322	15.758228
7.016050	12.235891
8.304229	12.536069
7.351775	19.349313
8.799763	18.347272
9.346700	18.812099
10.232547	19.750414
11.872116	24.672962



predict the price for a house with 5 square feet

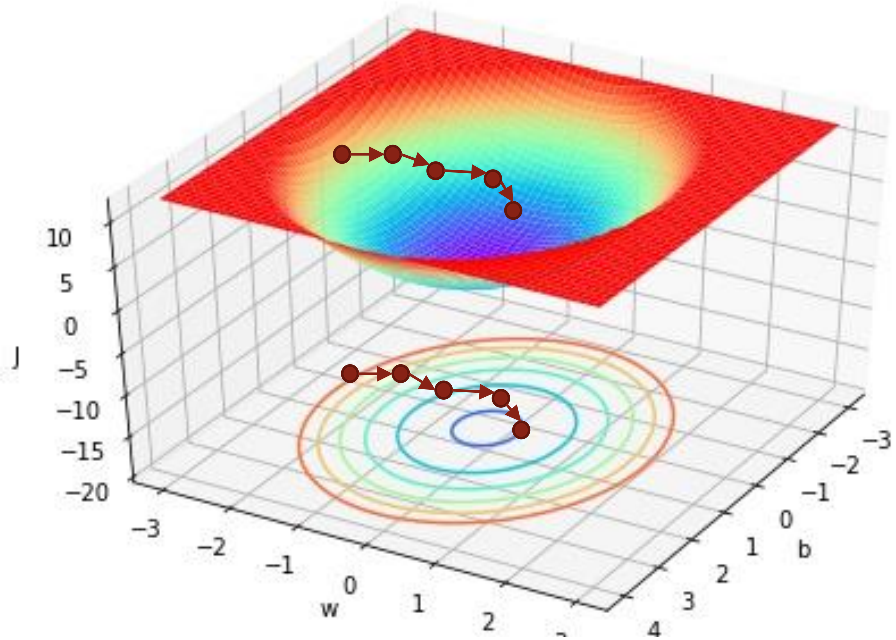
$$1.79 * 5 + 2.44 = 10.71$$

$$\begin{cases} w = \dfrac{\sum_{i=1}^{m} y^i \cdot x^i - m\bar{y}\bar{x}}{\sum_{i=1}^{m} x^i\, x^i - m\bar{x}^2} \\ b = \bar{y} - w\bar{x} \end{cases}$$

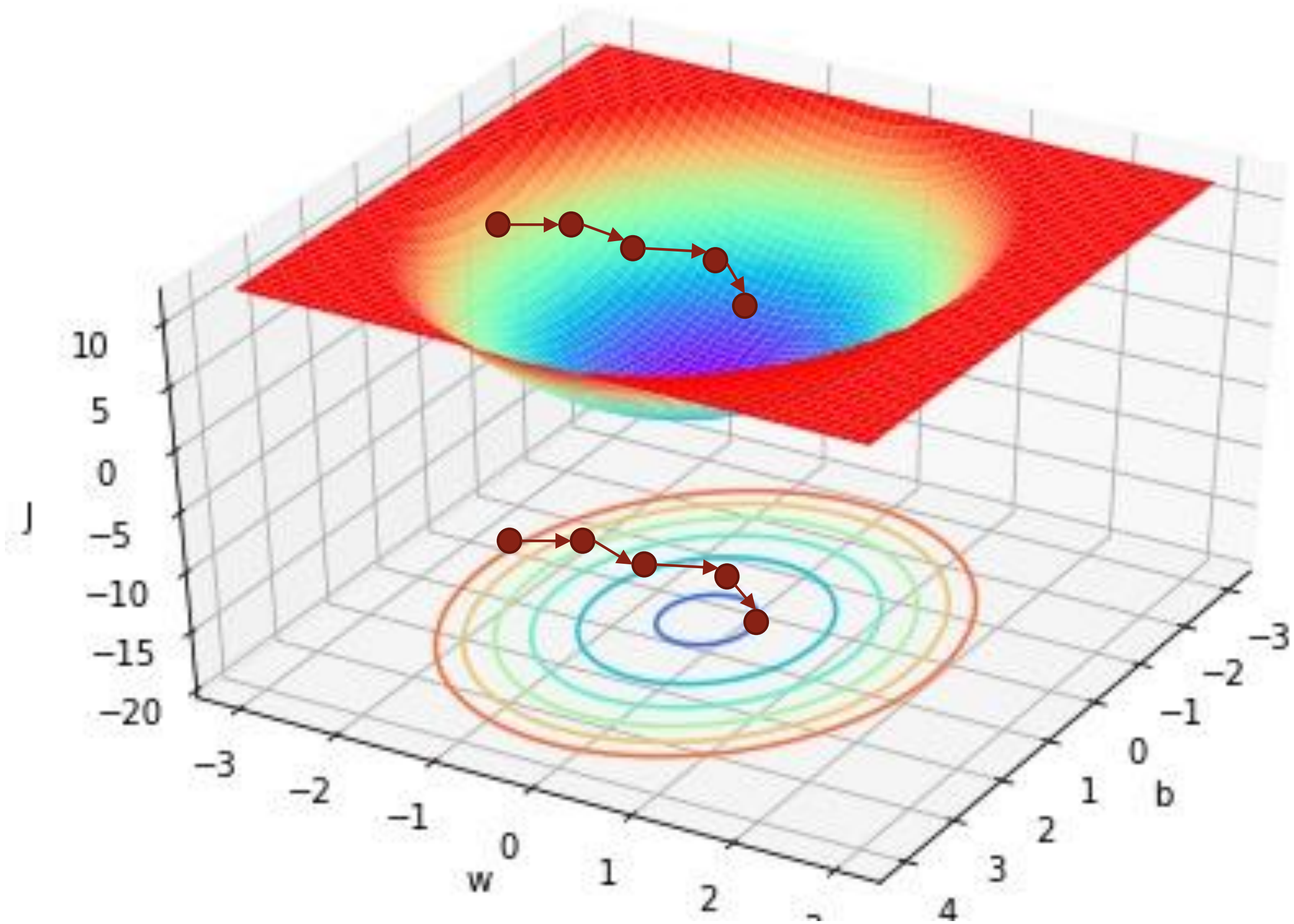
$$\begin{cases} \bar{y} = \dfrac{1}{m}\sum_{i=1}^{m} y^i \\ \bar{x} = \dfrac{1}{m}\sum_{i=1}^{m} y^i \end{cases}$$

Iteration method

Iteration method

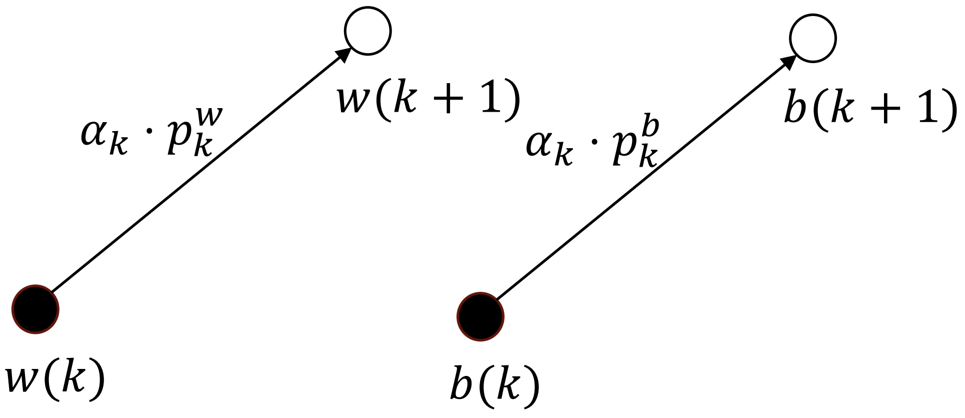


Iteration method



Finding a minimum point step by step

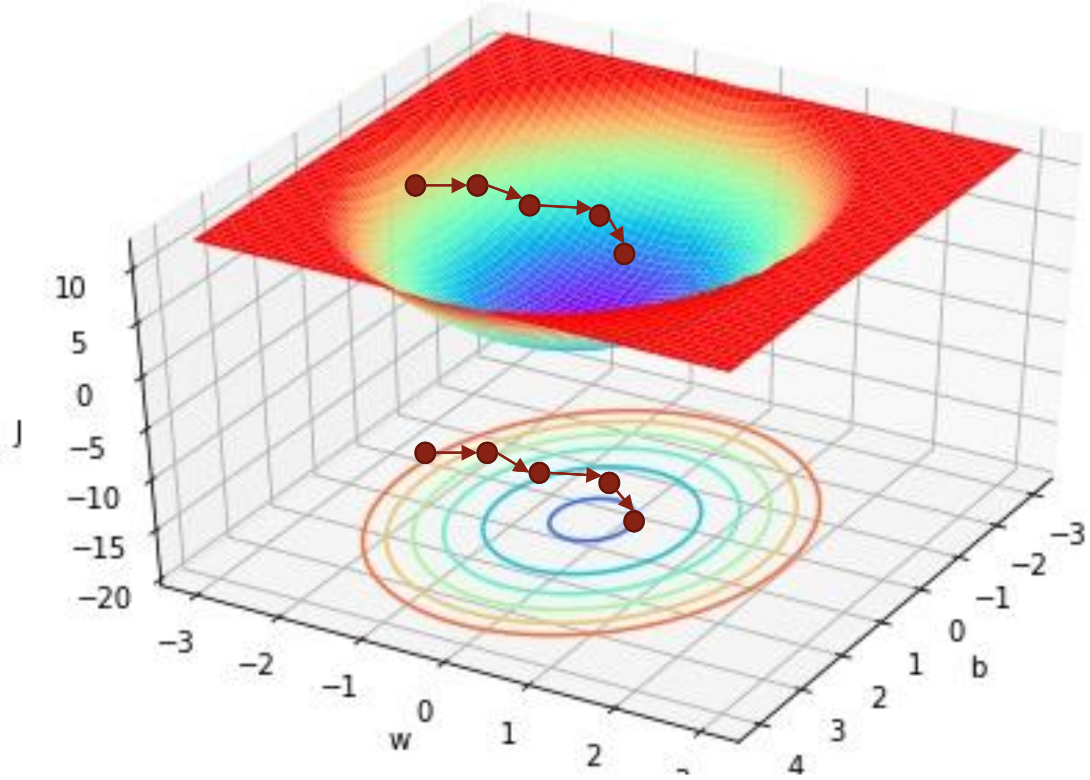
$$w(k+1) = w(k) \ + \alpha_k \cdot p_k^w$$
$$b(k+1) = b(k) \ + \alpha_k \cdot p_k^b$$

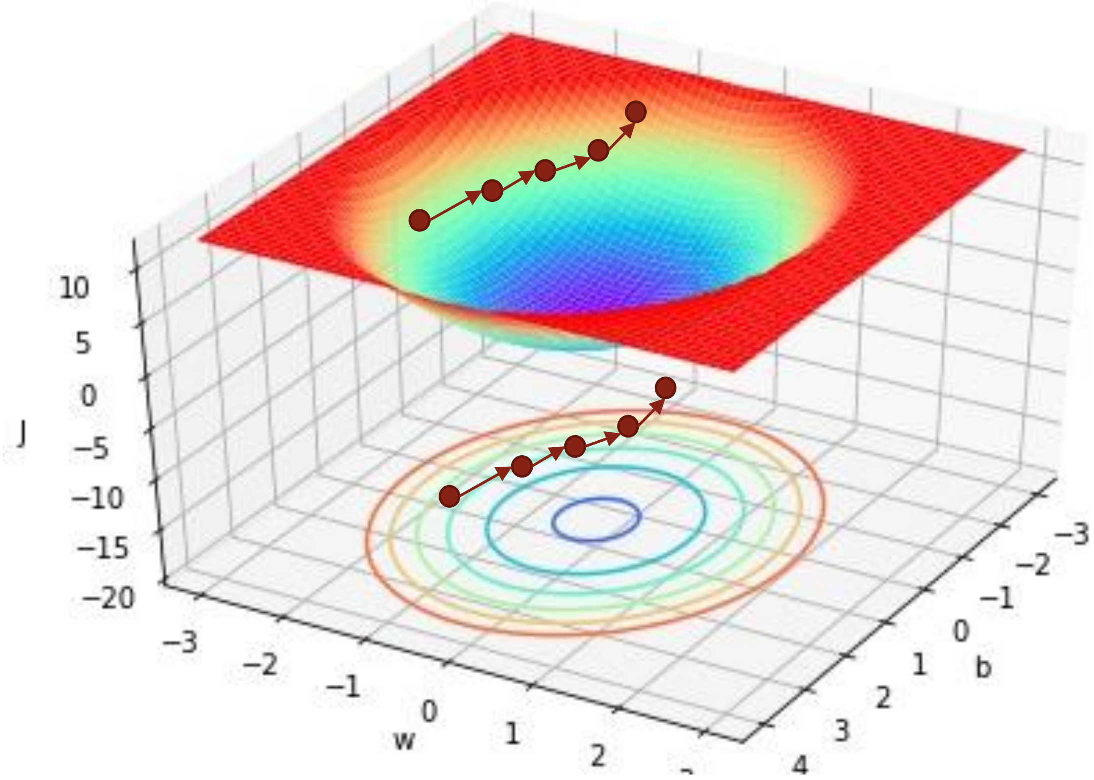


p_k^w , p_k^b are called searching direction

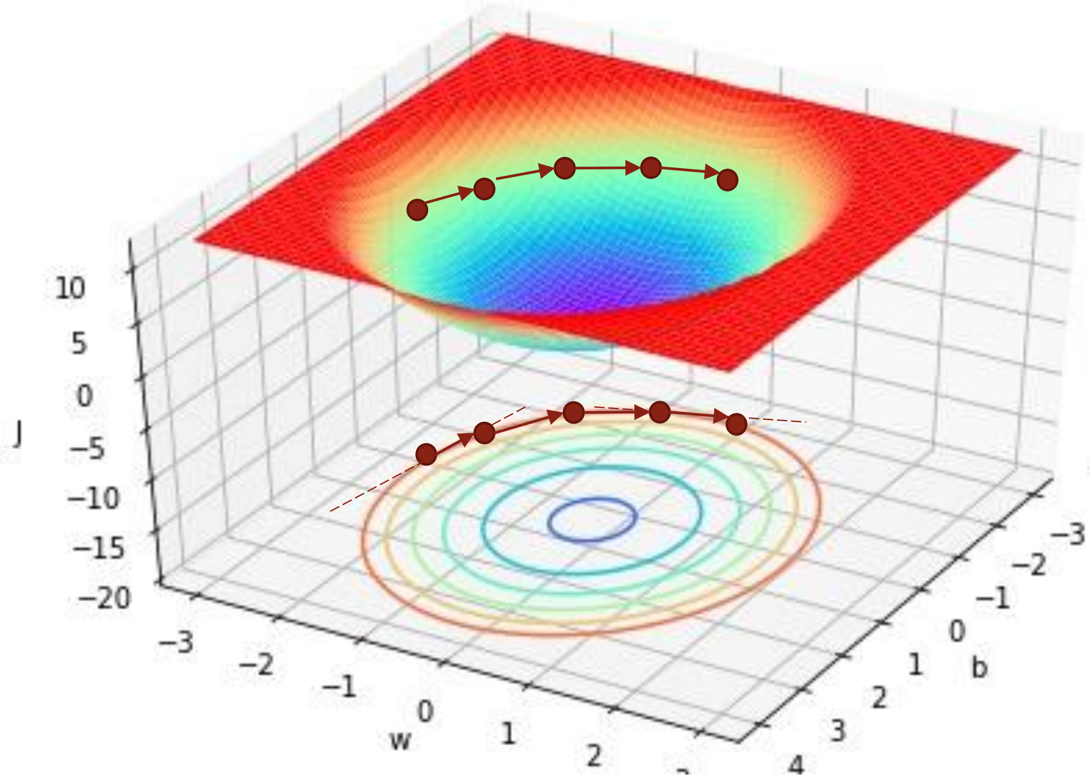
α_k is learning rate at step k.

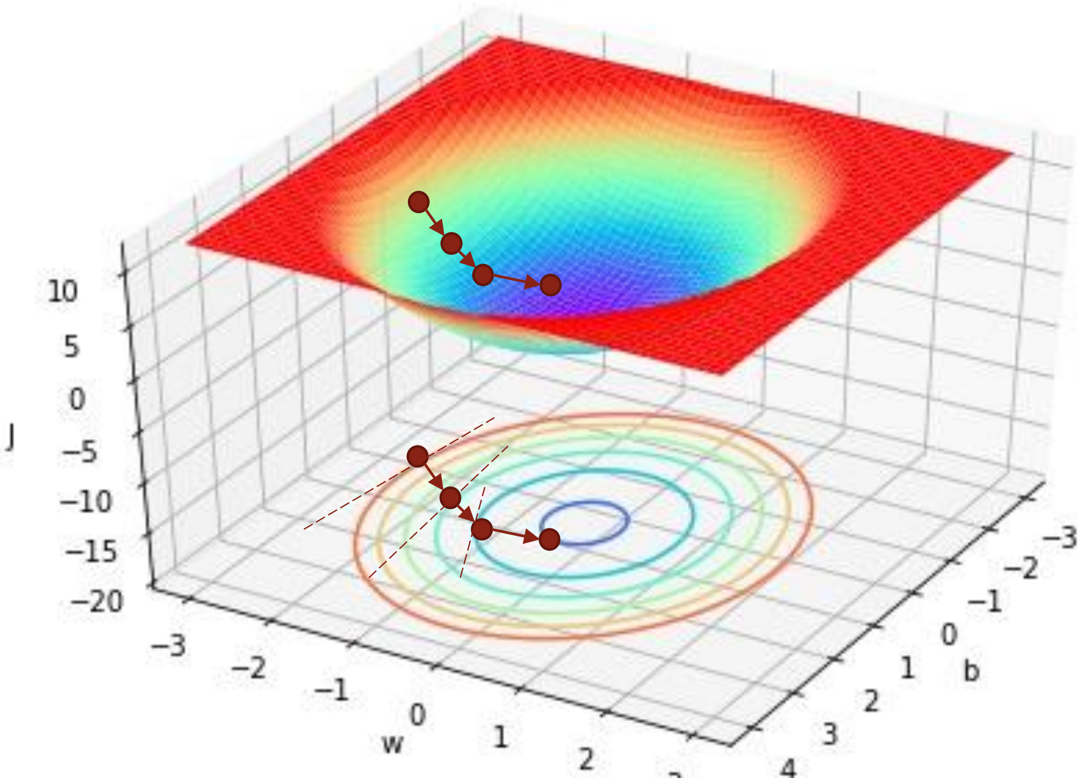
Iteration method





Iteration method



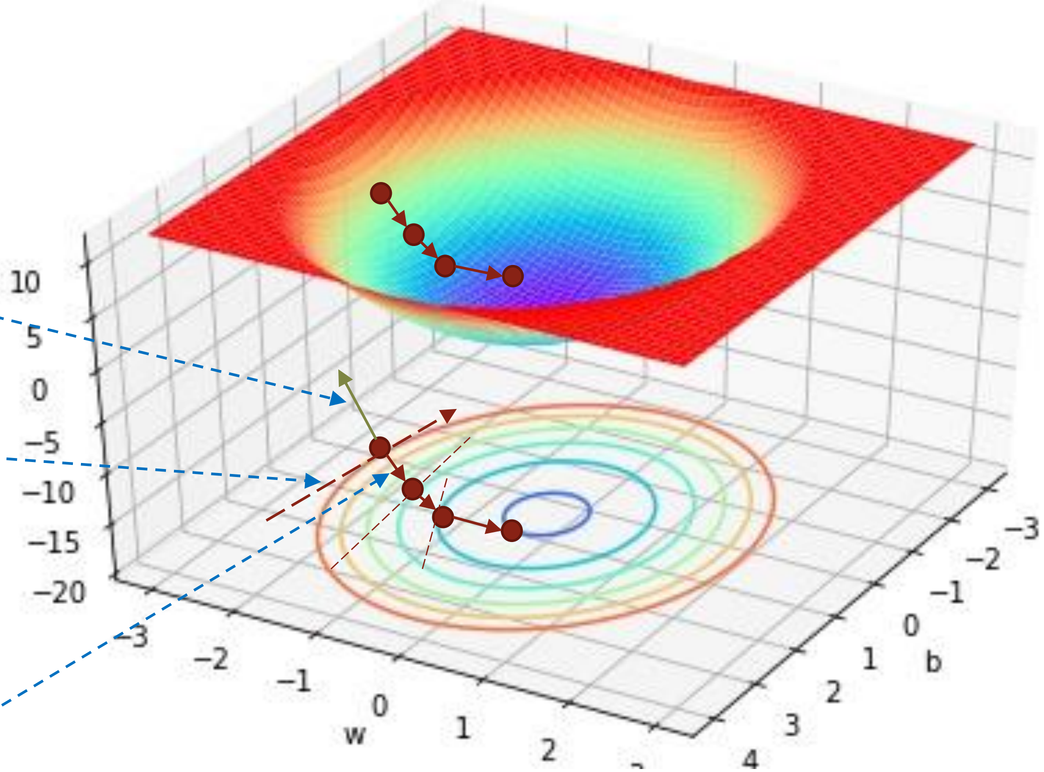


Steepest Gradient Descend Method

Fastest increasing direction

Slowest changing direction

Steepest descent direction



Gradient: $\begin{cases} \left.\dfrac{\partial J}{\partial w}\right|_{w=w(k)} \\ \left.\dfrac{\partial J}{\partial b}\right|_{b=b(k)} \end{cases}$

Steepest Descent Algorithm:

$$\begin{cases} p_k^w = -\left.\dfrac{\partial J}{\partial w}\right|_{w=w(k)} \\ p_k^b = -\left.\dfrac{\partial J}{\partial b}\right|_{b=b(k)} \end{cases}$$

$$\begin{cases} w(k+1) = w(k) - \alpha_k \cdot \left.\dfrac{\partial J}{\partial w}\right|_{w(k)} \\ b(k+1) = b(k) - \alpha_k \cdot \left.\dfrac{\partial J}{\partial b}\right|_{b(k)} \end{cases}$$

Steepest Gradient Descend Method

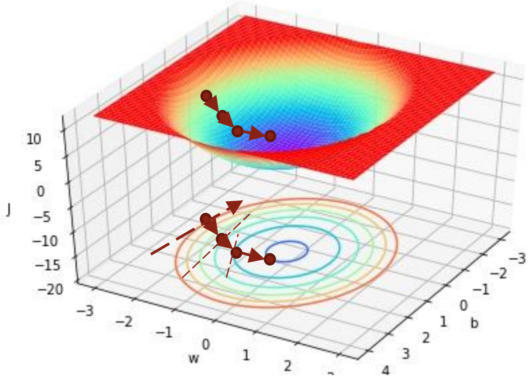
Univariate Linear Regression

- Data:
  - Sample: $(x \in R, y \in R)$, where x is square and y is the price.
  - Dataset: $D = \{(x^i \in R, y^i \in R) | i \in [1, m]\}$, where m=10 is the number of the samples.
- Linear Model:

$$a = wx + b$$

- Object:

$$\underset{w,b}{\operatorname{argmin}} J(w,b) = \underset{w,b}{\operatorname{argmin}} \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$$



Steepest Descent Algorithm:

$$\begin{cases} w(k+1) = w(k) - \alpha_k \cdot \left.\dfrac{\partial J}{\partial w}\right|_{w(k)} \\ b(k+1) = b(k) - \alpha_k \cdot \left.\dfrac{\partial J}{\partial b}\right|_{b(k)} \end{cases}$$

$$\frac{\partial J}{\partial w} = \frac{1}{m} \sum_{i=1}^{m} (wx^i + b - y^i) \cdot x^i$$

$$\frac{\partial J}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} (wx^i + b - y^i)$$

Steepest Gradient Descend Method

Univariate Linear Regression

- Data:
  - Sample: $(x \in R, y \in R)$, where x is square and y is the price.
  - Dataset: $D = \{(x^i \in R, y^i \in R) | i \in [1, m]\}$, where m=10 is the number of the samples.
- Linear Model:

$$a = wx + b$$

- Object:

$$\underset{w,b}{\operatorname{argmin}} J(w,b) = \underset{w,b}{\operatorname{argmin}} \frac{1}{2m} \sum_{i=1}^{m} (a^i - y^i)^2$$

Steepest Descent Algorithm

Input: D, w, b, α

for k in $1,2,\dots,K$:

{

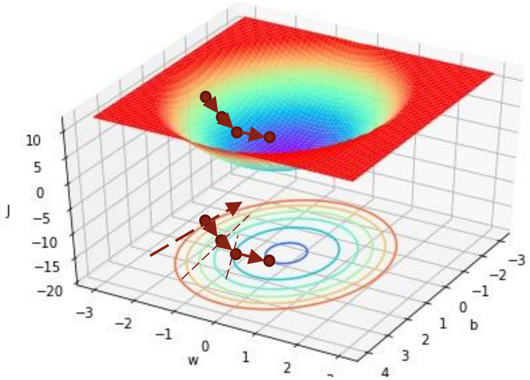
$$\frac{\partial J}{\partial w} \leftarrow \frac{1}{m} \sum_{i=1}^{m} (wx^i + b - y^i) \cdot x^i$$

$$\frac{\partial J}{\partial b} \leftarrow \frac{1}{m} \sum_{i=1}^{m} (wx^i + b - y^i)$$

$$w \leftarrow w - \alpha \frac{\partial J}{\partial w}$$

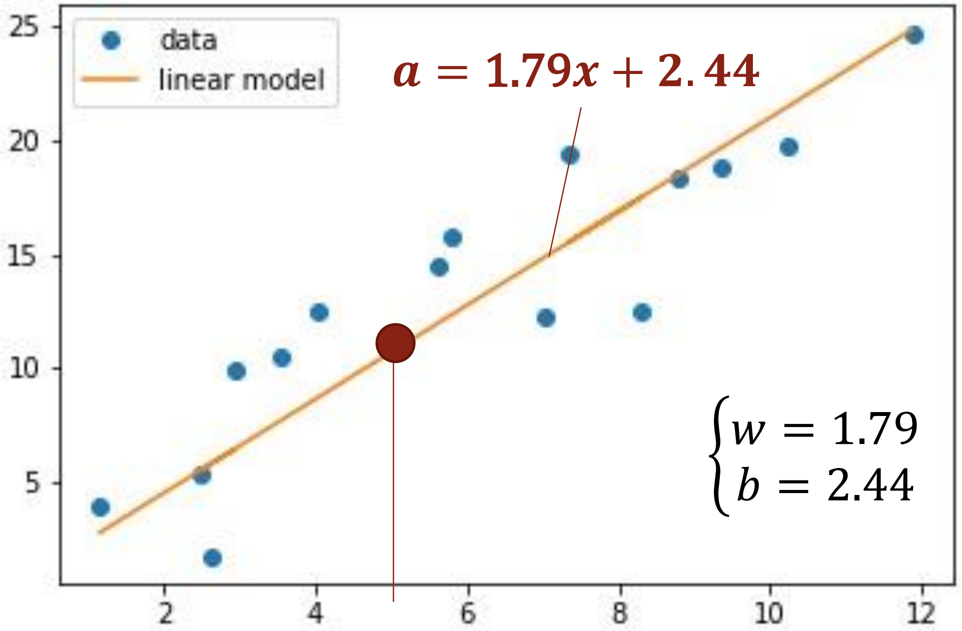
$$b \leftarrow b - \alpha \frac{\partial J}{\partial b}$$

}



Steepest Gradient Descend Method

square	price
1.156762	3.949326
2.624116	1.746431
2.943006	9.902035
2.499967	5.326710
3.530516	10.569117
4.045524	12.493749
5.607250	14.531507
5.784322	15.758228
7.016050	12.235891
8.304229	12.536069
7.351775	19.349313
8.799763	18.347272
9.346700	18.812099
10.232547	19.750414
11.872116	24.672962



predict the price for a house with 5 square feet

$$1.79 * 5 + 2.44 = 10.71$$

Steepest Descent Algorithm

Input: D, w, b, α

for k in $1,2,\dots,K$:

{

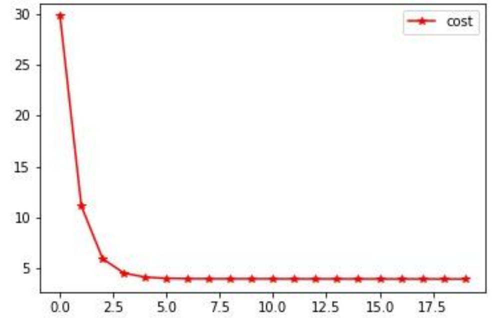
$$\frac{\partial J}{\partial w} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(wx^i + b - y^i)\cdot x^i$$

$$\frac{\partial J}{\partial b} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(wx^i + b - y^i)$$

$$w \leftarrow w - \alpha\frac{\partial J}{\partial w}$$

$$b \leftarrow b - \alpha\frac{\partial J}{\partial b}$$

}



Thanks!!!