Dian AI培训

贝贝组 黄涛

2018.07.23

Contents

How to learn machine learning?

Courses:

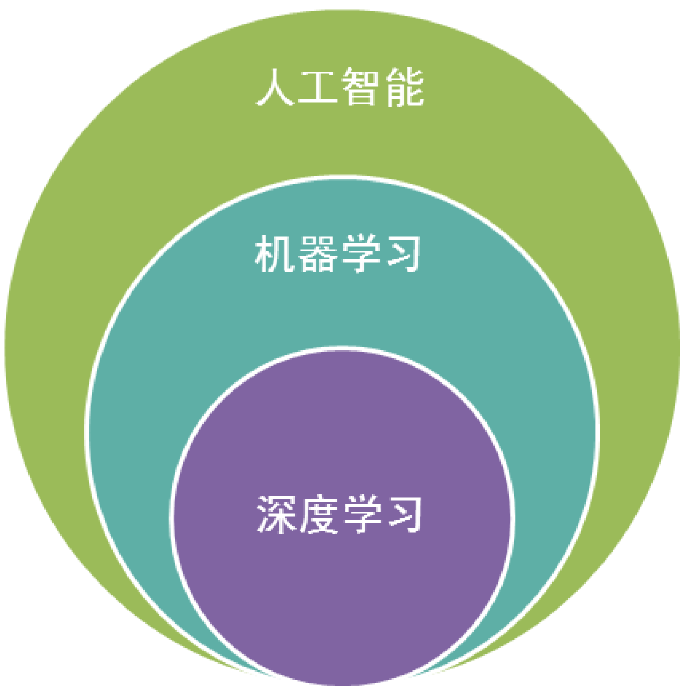
- Machine Learning - Adrew Ng
- CS231n - Convolutional Neural Networks for Visual Recognition
- CS224n - Natural Language Processing with Deep Learning
  - CS224n课程笔记-Hunto

Books:

- 机器学习-周志华
- Deep Learning - Ian Goodfellow and Yoshua Bengio and Aaron Courville
  - en: http://www.deeplearningbook.org/
  - cn: https://github.com/exacity/deeplearningbook-chinese

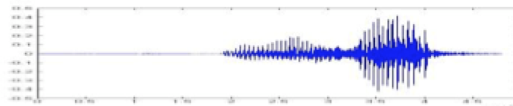
Relationship between AI,ML,DL

- AI: Artificial Intelligence
- ML: Machine Learning
- DL: Deep Learning

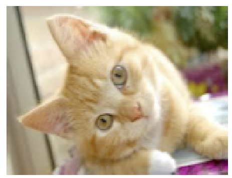


Machine Learning ≈ Looking for a Function

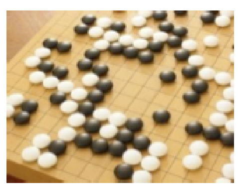
- Speech Recognition

$f($  $)=$ "How are you"

- Image Recognition

$f($  $)=$ "Cat"

- Playing Go

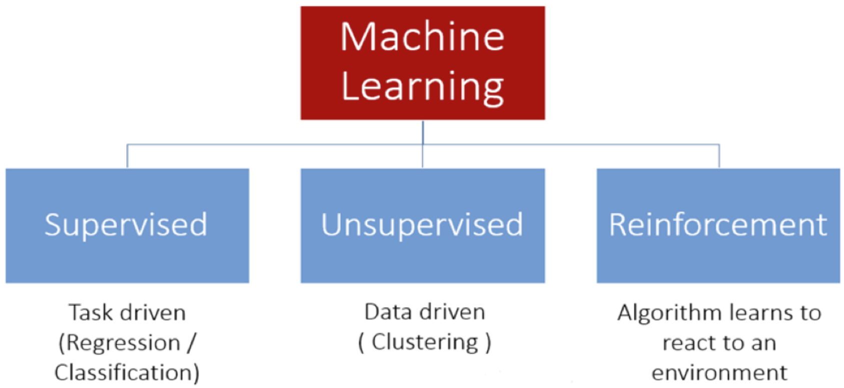
$f($  $)=$ "5-5" (next move)

- Dialogue System

$f($ "Hi" $)=$ "Hello"

(what the user said) (system response)

Types of Machine Learning



- Supervised Model(监督学习模型) : 任务驱动
  训练数据带有标签
    - Regression
    - Classification
- Unsupervised Model(无监督学习模型)： 数据驱动
  训练数据无标签
    - Clustering
- Reinforcement(强化学习)： 学习适应环境

Framework

A set of function: $f_1, f_2, \ldots, f_n$

example

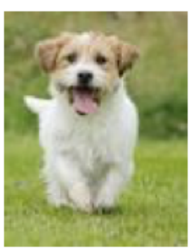
Image Recognition

$f($  $) =$ "cat"

Functions:

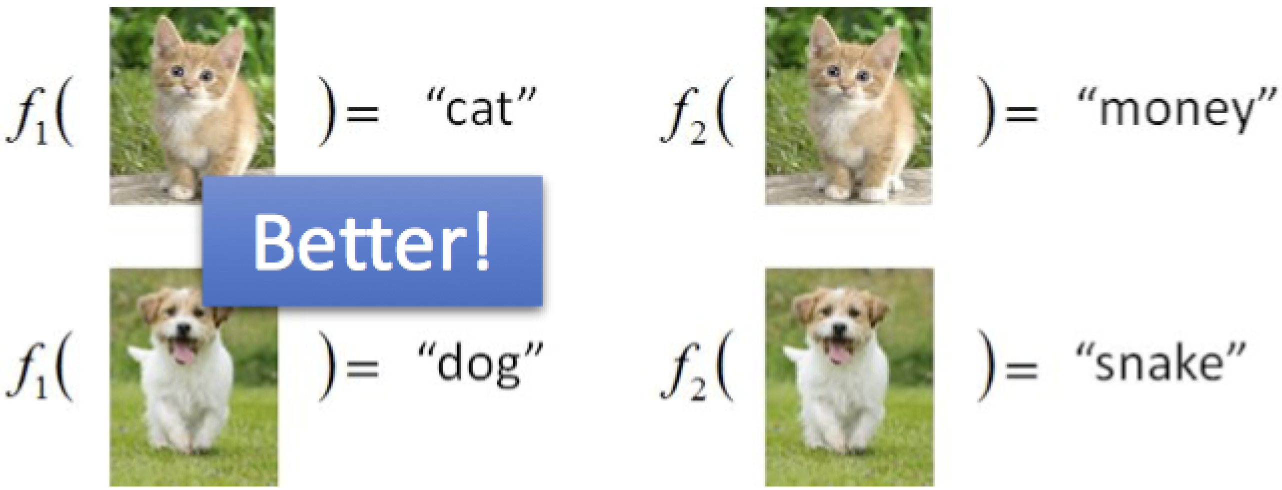
$f_1($  $) =$ "cat"

$f_2($  $) =$ "money"

$f_1($  $) =$ "dog"

$f_2($  $) =$ "snake"

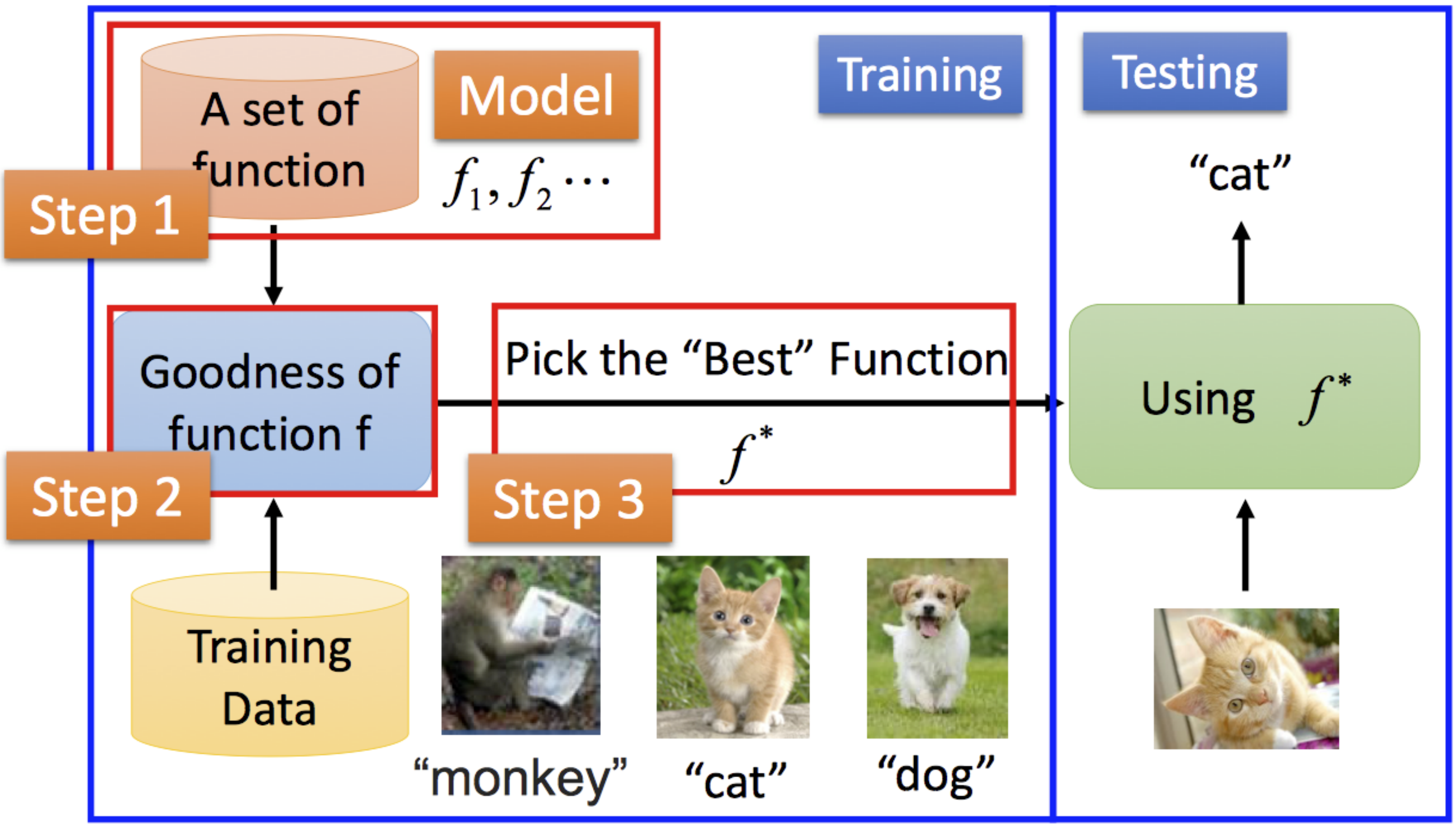
Framework



Supervised Learning

- We have **training data**.
- We have **a set of function**.
- $f(input) = f_{out},\ \ cmp(real\ data, f_{out}) => f_{score}$
- Pick the "best" function.

Supervised learning



How to train a supervised model?

1. Find an appropriate cost function for your task
2. Minimize the cost function

Example

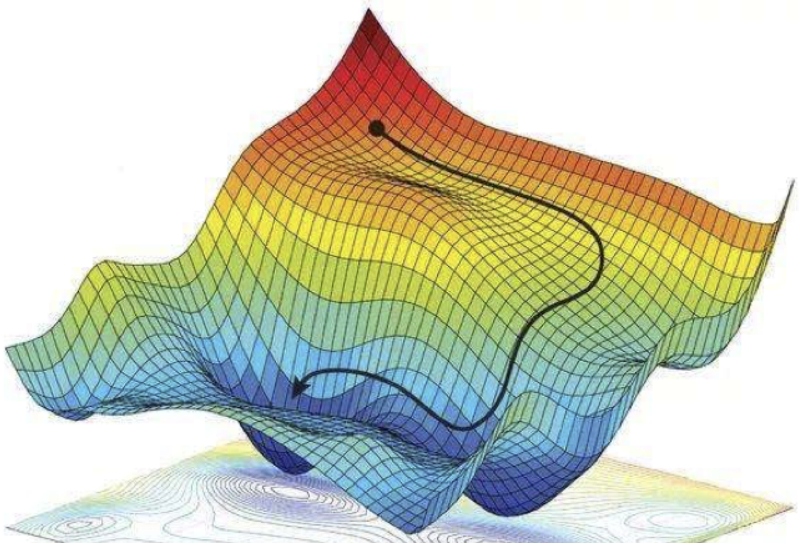
- Square loss for linear regression

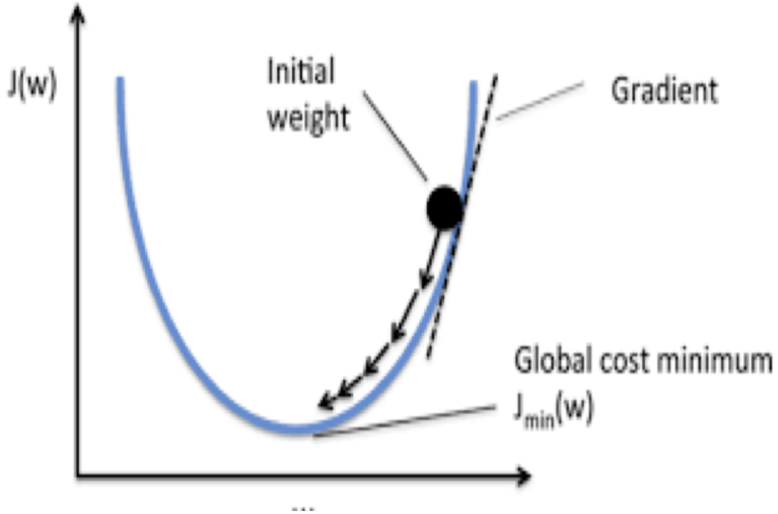
$$C = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

Minimize cost function

- Gradient descent

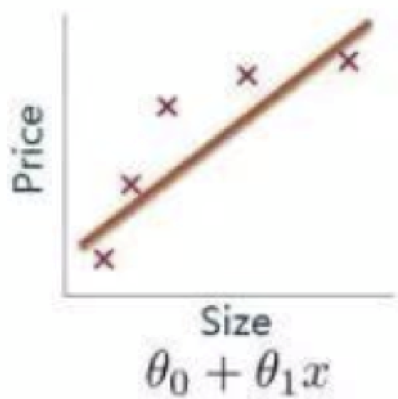
$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

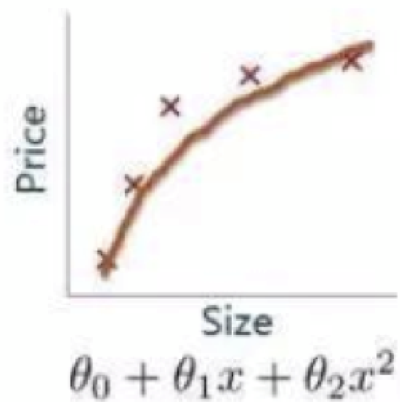


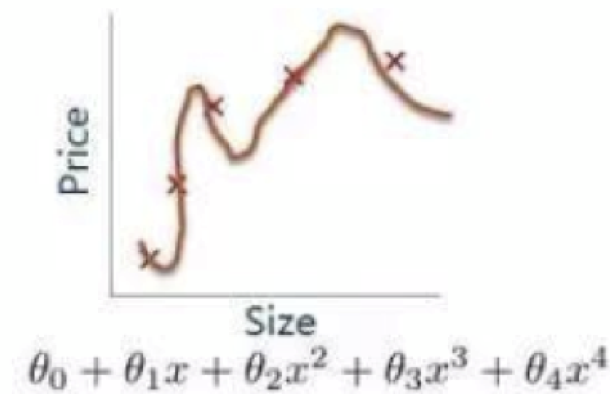


How to evaluation?

3 situations







Underfitting

Well-fitting

Overfitting

How to evaluation?

1. Split train and test dataset
   train: usually $\frac{2}{3}$~$\frac{4}{5}$ of data
2. Train model on train dataset, and using test dataset to evaluation the model.
3. Evaluation methods:
   - Accuracy（准确率）
   - Recall（召回率）
   - F-score（加权调和平均）: $F = \frac{(a^2+1)P \cdot R}{a^2(P+R)}$
     $a = 1 => F_1 = \frac{2 \cdot P \cdot R}{P+R}$
   - AUC
   - ...

ML Pipeline

- Data collecting
- Feature engineering
- Model building
- Application
- Feedback and Optimization

Feature Engineering

- Feature is the limitation of model.

  70% time for feature engineering, **30%** time for modeling

- Manual features
  - Statistical features
  - Time series
  - Prior knowledge

Example: stock risk prediction

Model

Classification Model

- Logistic regression
- SVM (Supported Vector Machine)
- Decision Tree

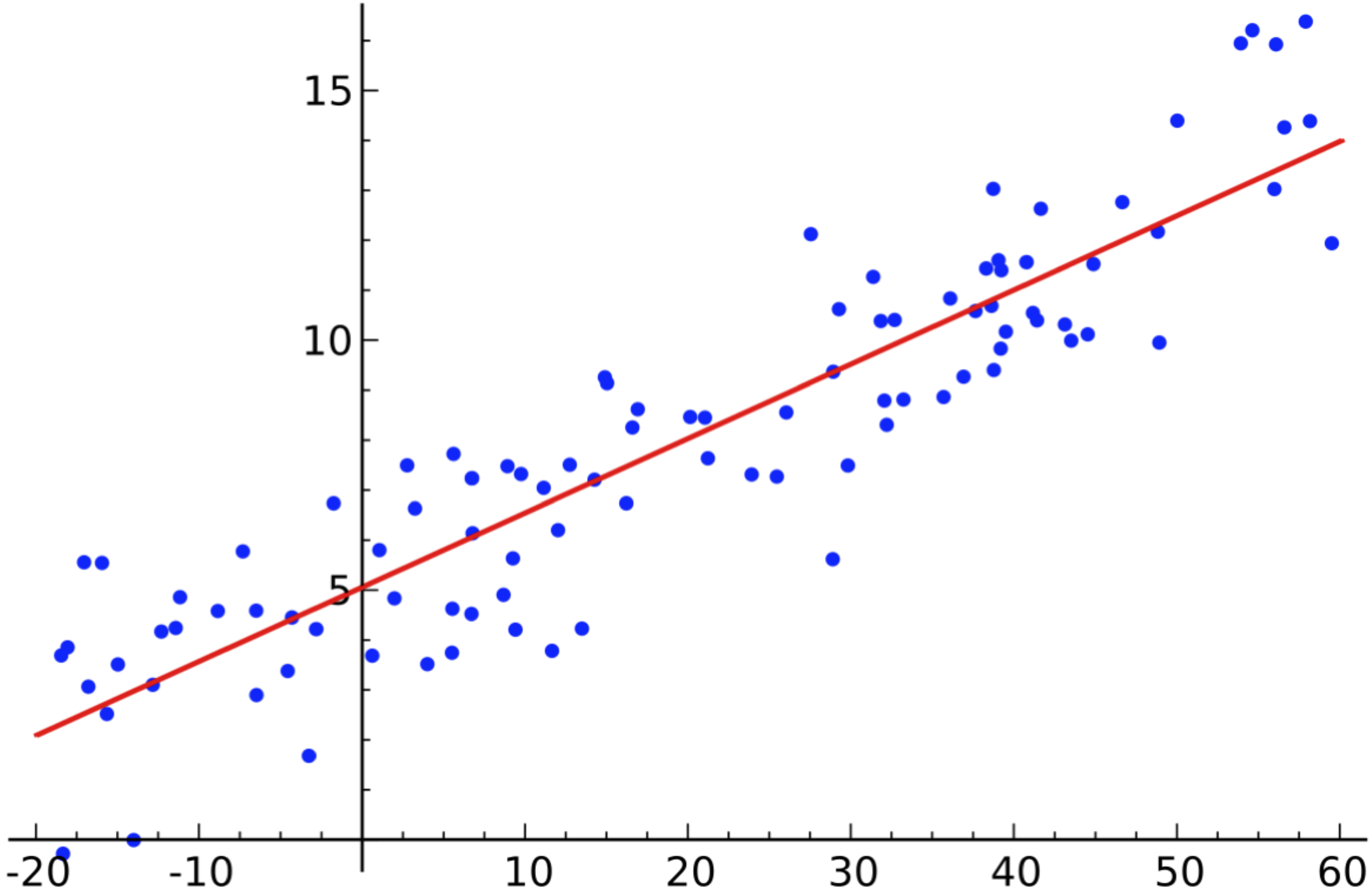
Regression Model

- Linear regression
- Non-linear regression

Clustering Model

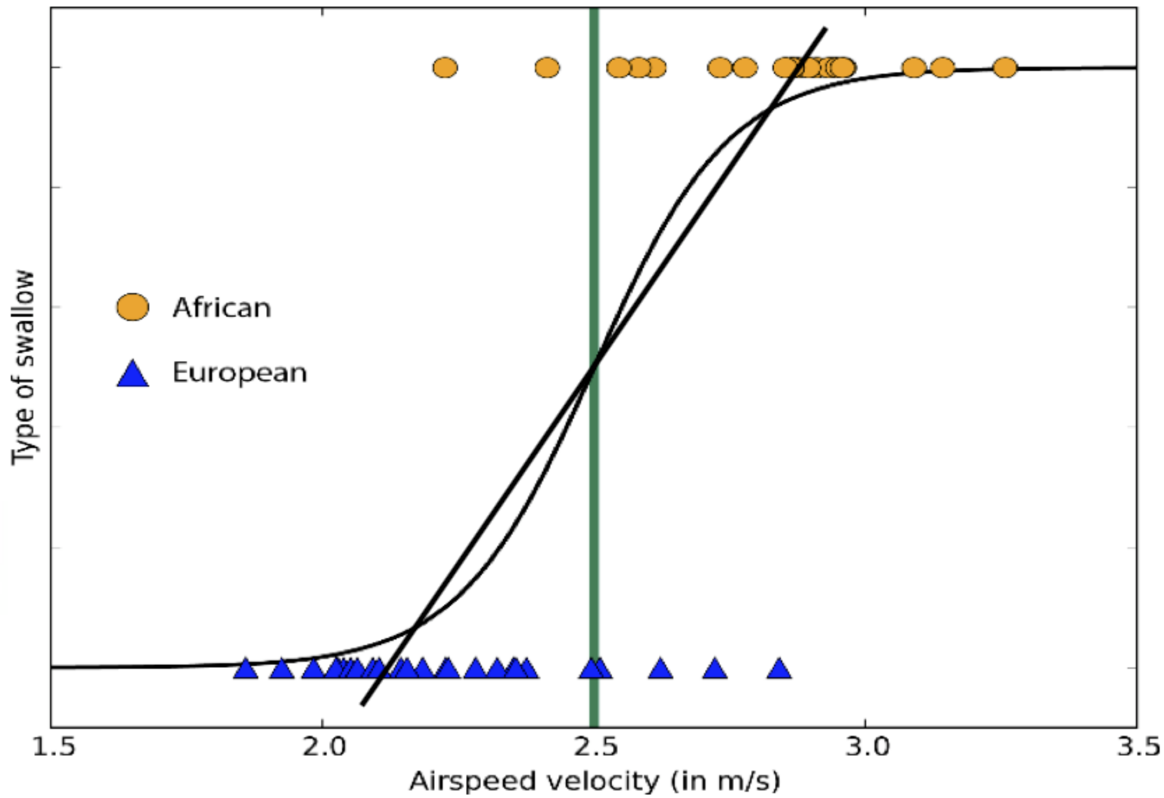
- K-means

Linear regression （线性回归）



$$g(x) = w_0 + w_1 x_1 + \ldots + w_n x_n$$

Logistic Regression （逻辑回归）

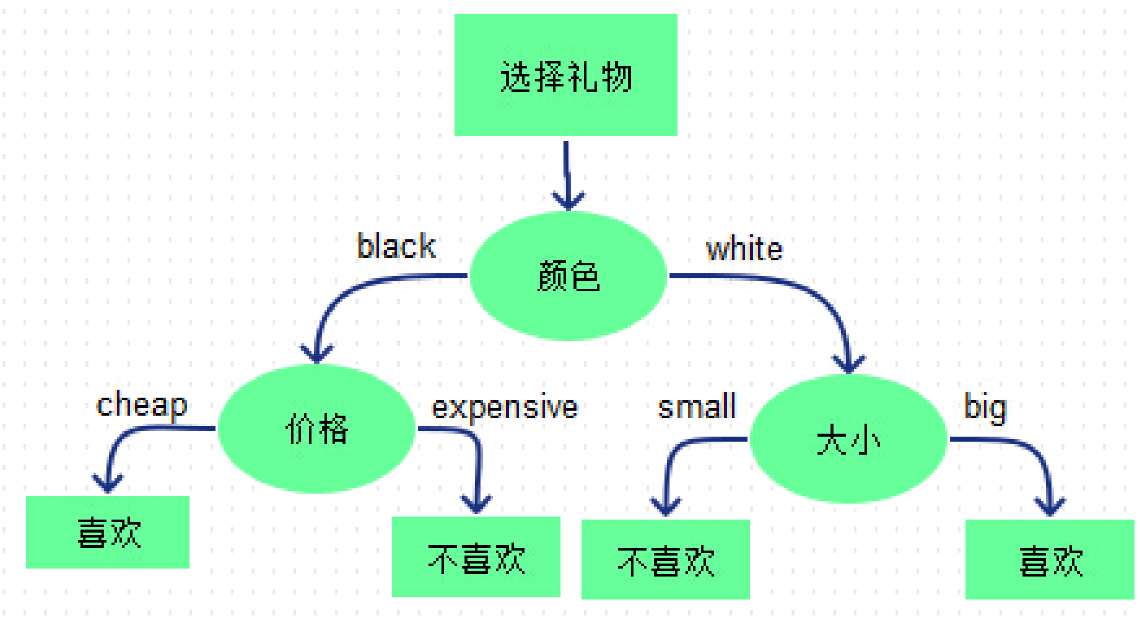


$$g(x) = w_0 + w_1 x_1 + ... + w_n x_n$$
$$sigmoid(x) = \frac{1}{1+e^{-x}}$$

$$P(y=1|x) = sigmoid(g(x)) = \frac{1}{1+e^{-g(x)}}$$

Decision Tree （决策树）

决策树是运用于分类的一种树结构，其中的每个内部节点代表对某一属性的一次测试，每条边代表一个测试结果，叶节点代表某个类或类的分布。
决策树的决策过程需要从决策树的根节点开始，待测数据与决策树中的特征节点进行比较，并按照比较结果选择选择下一比较分支，直到叶子节点作为最终的决策结果。



Deep Learning

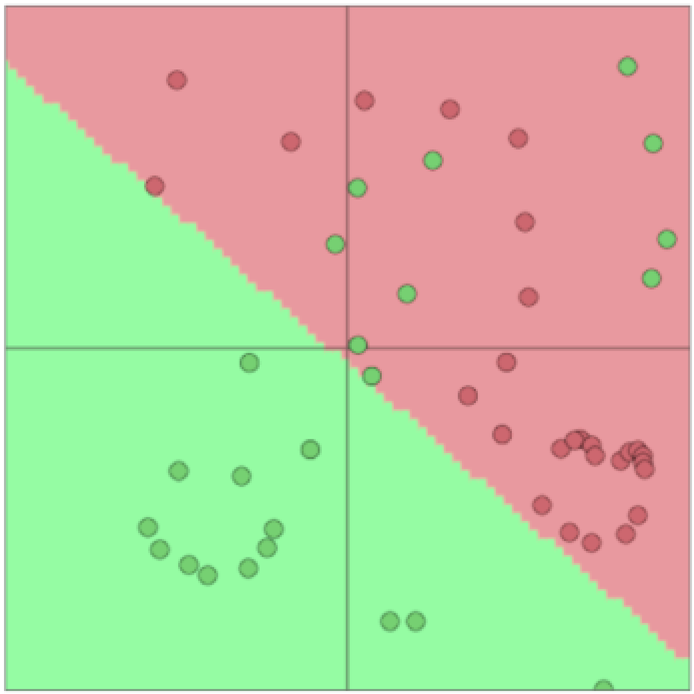
What's deep learning

Deep learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artifical neural networks.

Why deep learning is so hot?

- Coming up with features is difficult, time-consuming, requires expert knowledge. "Applied machine learning" is basically feature engineering. (Andrew Ng)
- But deep learning can automatically extract features!

传统机器学习的局限



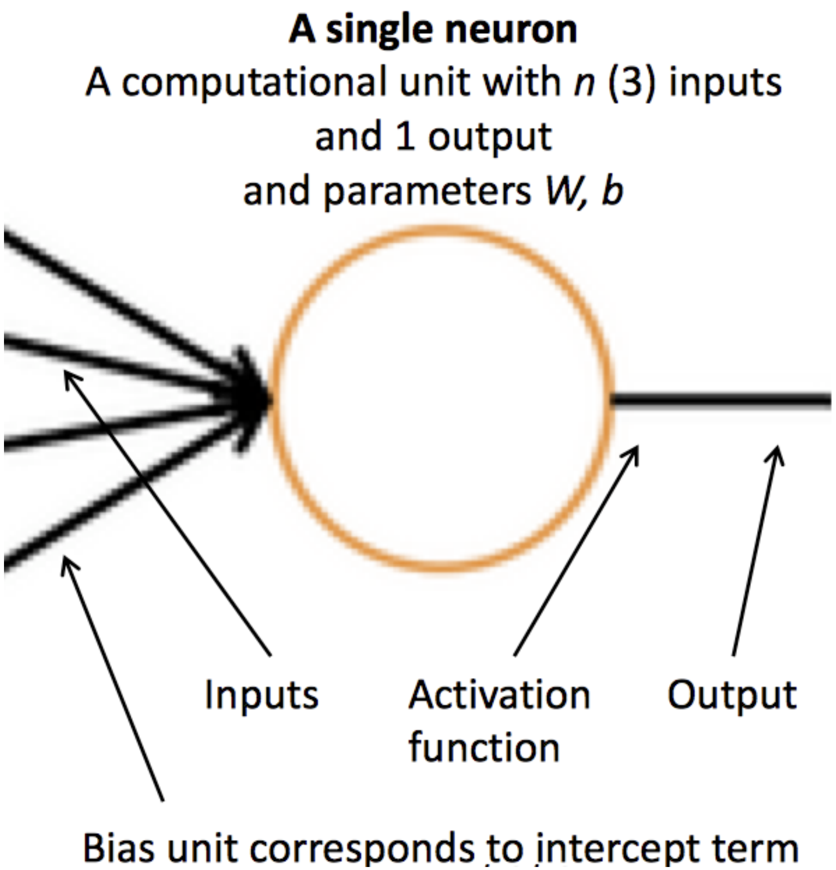
Softmax(logistic regression)只有线性的决策边界，分类效果有限，当问题变得复杂时，效果不好， 但神经网络可以学到复杂得多的特性和非线性决策边界。

Deep leariningi applications

- Computer Vison:
  - Image Recognition
- Speech Recognition
- Natural Language Processing
  - Machine translation
  - Sentiment analysis
  - Text classification

From logistic regression to neural networks

神经网络的每一个神经元都是一个二分类逻辑回归单元。

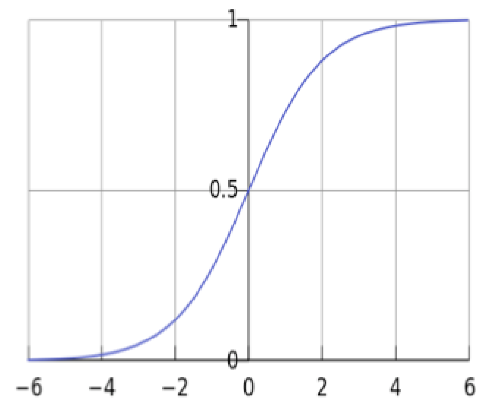


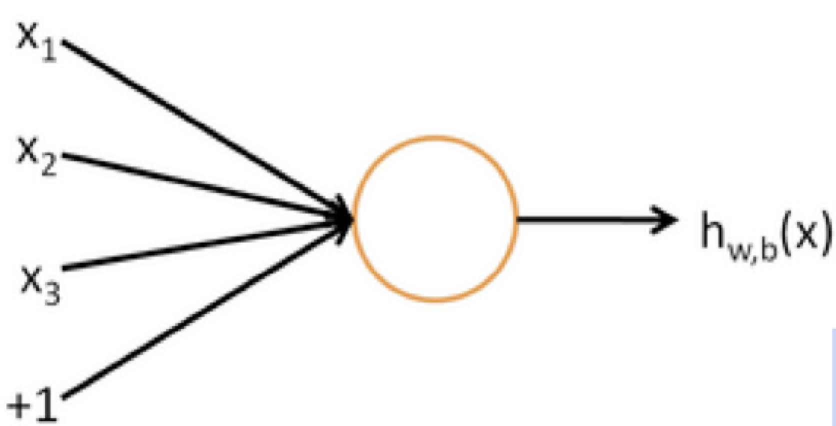
f = nonlinear activation fct. (e.g. sigmoid), w = weights, b = bias, h = hidden, x = inputs

$$h_{w,b}(x) = f(w^{\mathsf{T}} x + b)$$

b: We can have an “always on” feature, which gives a class prior, or separate it out, as a bias term

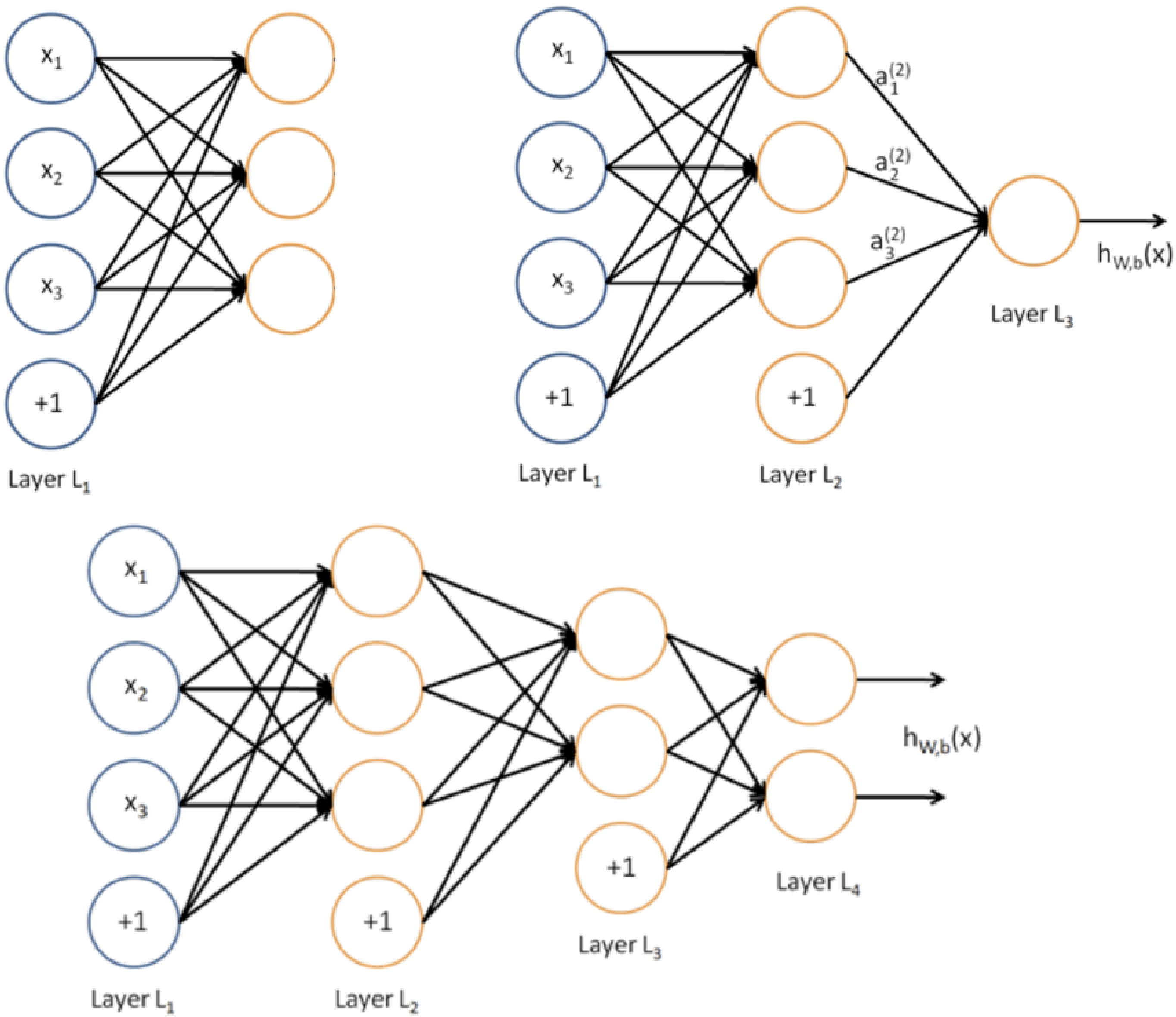
$$f(z) = \frac{1}{1+e^{-z}}$$



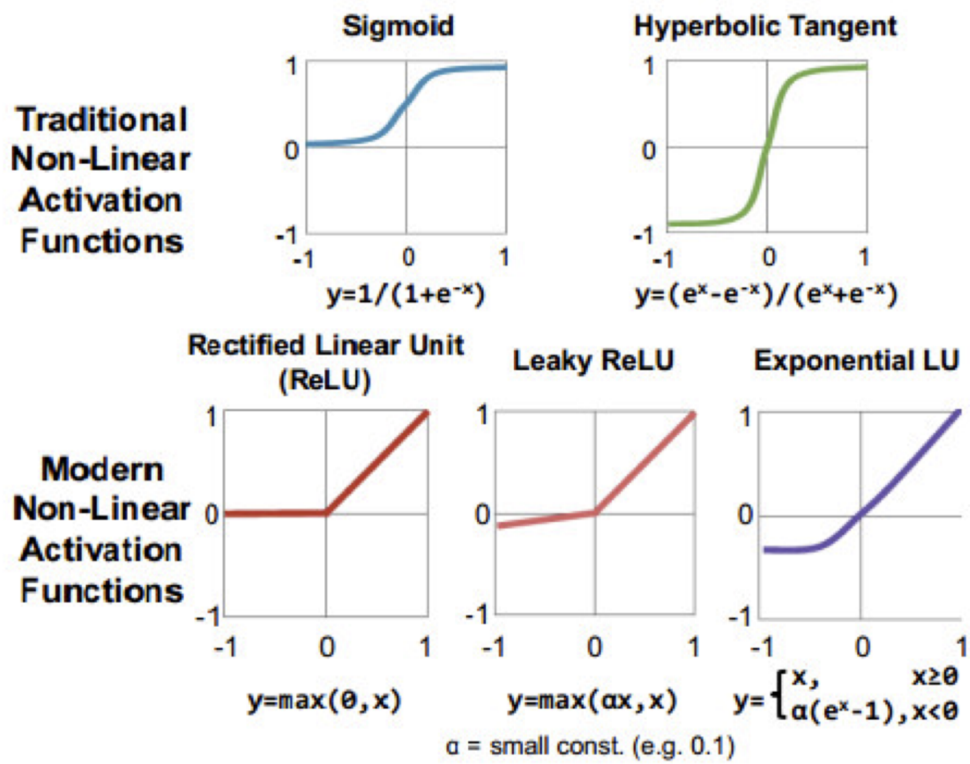


w, *b* are the parameters of this neuron i.e., this logistic regression model

Multi-layer Neural Networks



Activation function

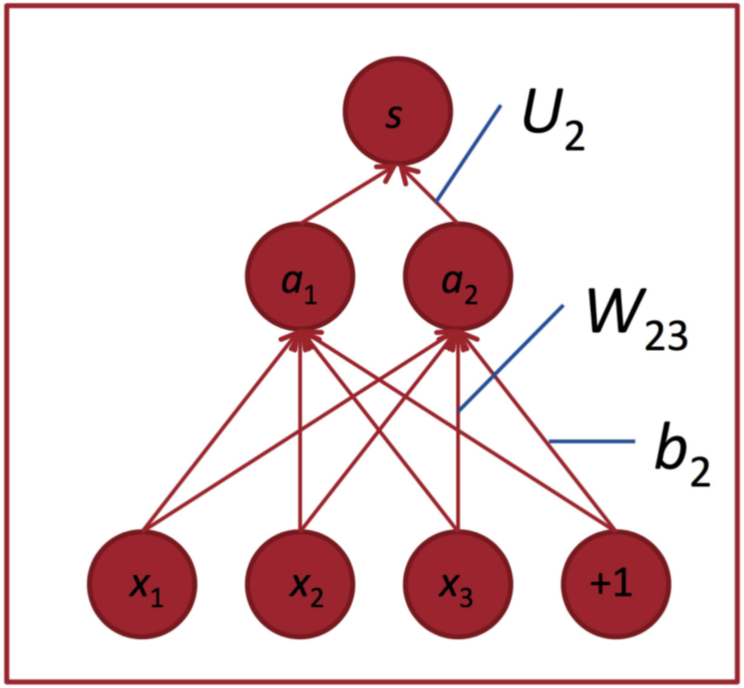


Why activation function must be `non-linear`?

多层线性系统叠加仍然是线性系统。

$$x \cdot W_1 \cdot W_2 \cdot \ldots \cdot W_n = x \cdot W$$

Feed-forward Computation



$$z = Wx + b\text{w}$$
$$a = f(z)$$
$$s = U^T a$$

Feed-forward computation

Example

1. We have $x = [0, 1, 2, 3]$
2. $z = Wx + b$, suppose $W_1 = 2, b_1 = 1, W_2 = 1, b = -1$
   $z_1 = [1, 3, 5, 7], z_2 = [-1, 0, 1, 2]$
3. Activation function, exp. : Softmax $f(x) = \frac{1}{1+e^{-x}}$
   $a_1 = [0.731, 0.952, 0.993, 0.999]$
   $a_2 = [0.268, 0.5, 0.731, 0.880]$
4. $s = U^T a$, suppose $U = [0.4, 0.6]$
   $s = [0.4, 0.6]^T \cdot [a_1, a_2] = [0.453, 0.680, 0.836, 0.928]$

Loss Function

softmax & cross-entropy loss(交叉熵损失)

- softmax

$$softmax(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j)}$$

- cross-entropy loss

$$CE(y, \hat{y}) = -\sum_{j=1}^{|V|} y_j log \hat{y}_j$$

- softmax cross-entropy loss

$$J(\theta) = -\sum_{j=1}^{|V|} y_j log(softmax(s_j))$$

Loss Function

Example

1. We have $x = [0, 1, 2, 3]$
2. $z = Wx + b$
3. Activation function, exp. : Softmax $f(x) = \frac{1}{1+e^{-x}}$
4. $s = U^T a$
   $s = [0.453, 0.680, 0.836, 0.928]$
5. $\hat{y} = softmax(s)$
   $\hat{y}_1 = [0.323, 0.257, 0.220, 0.201]$
6. Loss $J = -\sum_{j=1}^{|V|} y_j log(softmax(s_j))$, suppose
   $y = [1, 0, 0, 0]$
   $e = 1.130$

Backpropagation（反向传播）

Multi-layer neural networks

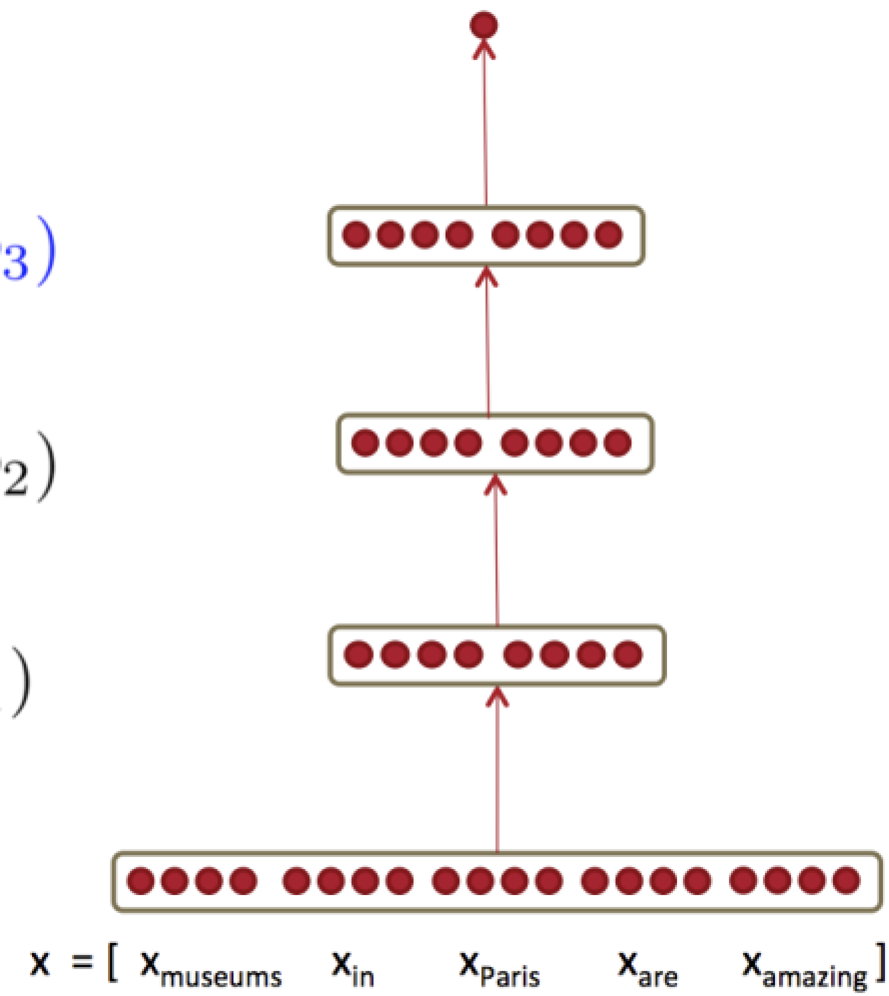
$$s = \boldsymbol{u}^T \boldsymbol{h}_3$$

$$\boldsymbol{h}_3 = f(\boldsymbol{W}_3 \boldsymbol{h}_2 + \boldsymbol{b}_3)$$

$$\boldsymbol{h}_2 = f(\boldsymbol{W}_2 \boldsymbol{h}_1 + \boldsymbol{b}_2)$$

$$\boldsymbol{h}_1 = f(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1)$$

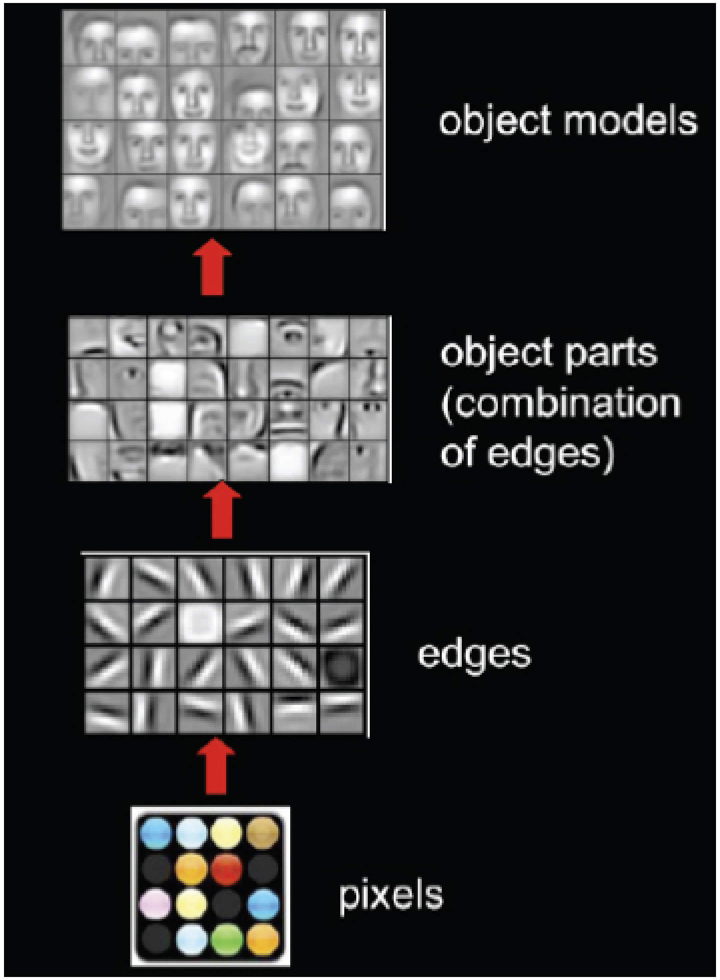
$$\boldsymbol{x} \quad \text{(input)}$$



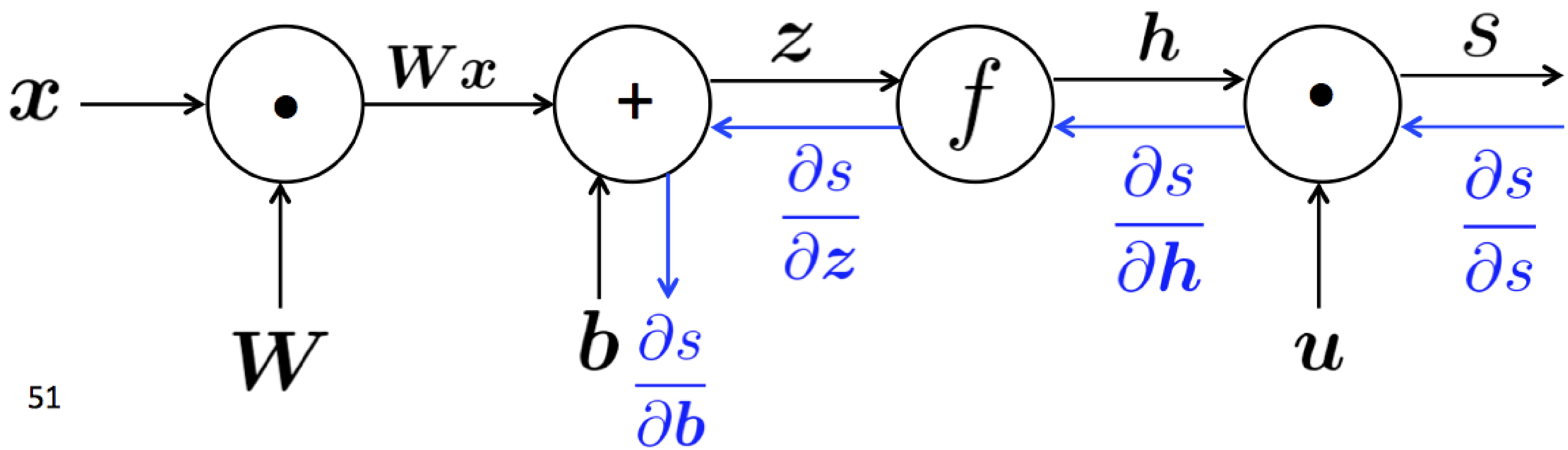
Backpropagation

Why we need multi-layer nets?

层数越多，可以表达的问题越复杂



Feed-forward & Backpropagation

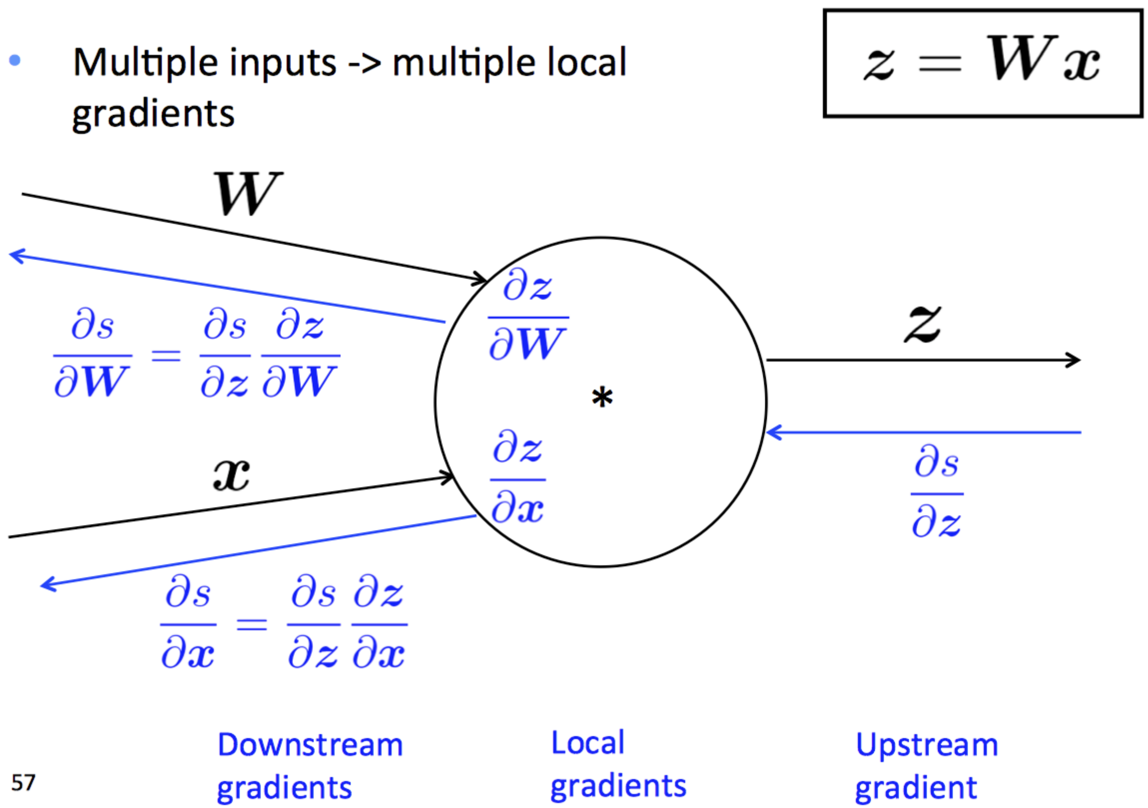


Why?

- Chain Rule（链式法则）

$$(g \circ f)'(x) = [g(f(x))]' = g'(f(x))f'(x) = \frac{du}{dx} \cdot \frac{dy}{dx}$$

Backpropagation



Chain Rule

$$u = g(y_1, y_2, \ldots, y_n), y = f(x_1, x_2, \ldots, x_n)$$

$$\frac{\partial u}{\partial x_i} = \sum_{j=1}^{n} \frac{\partial g}{\partial y_j} \cdot \frac{\partial y_j}{\partial x_i}$$

Optimizer

- SGD (Stochastic Gradient Descent / Mini-batch Gradient Descent)
    - Gradient Descent in a mini batch(a part of trainset)
- Momentum 90
- Adam(Adaptive Moment Estimation)

Review

Steps to train a neural network:

- Process train data: data to vector, split train/dev data.
- Initialize network weights.
- Feed-forward computation: input x -> output score.
- Cost computation: score vs. real y -> error.
- Backpropagation: propagate loss back.
- Optimizer: update weights.

Deep Reinforcement Learning

Closest to Artificial General Intelligence(AGI)

- It's the learning paradigm of biological.
- RL provides the resource of data.
- DL automatically extract features.
- When there's a general method for machine to build models, AGI will come.

Homework

- 深入了解本次培训的内容，如神经网络、激活函数等
- 熟悉深度学习框架Pytorch
  - Pytorch Tutorials
  - Pytorch 中文教程
- 使用Pytorch实现一个全连接神经网络，对MINIST数据集进行分类，有余力的同学可以尝试在cifar-10上的效果。