

目录

Introduction to AI	2
一、 Things about AI	2
二、 Applications of AI	6
三、 Some other things	8
Decision making	9
一、 Logic and uncertainty	9
二、 Probability theory	10
三、 Random variables & Probability & Bayer's	12
ML	14
一、 Many kinds of learning	14
Supervised Learning	16
一、 Decision Trees	16
二、 Classification	20
RL	23
一、 Introduction	23
二、 MDP	25
三、 Value-based RL	26
四、 Policy-based RL	28
DL	29
一、 Introduction	29
二、 Elements of NN	31
三、 Activation Function	33
四、 Tasks in training NN	35
五、 Loss function	36
六、 Gradient descent (vanilla GD)	37
七、 GD optimization	39
八、 Overfitting	41
九、 NN architectures	43
Practical AI Applications	45
Computer Vision	48
一、 Basics	48
二、 Basic Operations & Concepts	51
三、 Image Classification	53
四、 Semantics Segmentation	54
五、 Object Recognition	55
Natural Language Processing	56
一、 Introduction	56
二、 DL for NLP	59
三、 NLP applications	62
Limitations & Future	63

Introduction to AI

一、Things about AI

1. 什么是 AI

the science and engineering of making intelligent machines
use computers to understand human intelligence
use computer science and robust datasets to solve problems

2. 什么是 weak AI

- (1) weak AI = Narrow AI = Artificial Narrow Intelligence (ANI)
- (2) focused to perform specific tasks
- (3) 例子: alexa、SIRI 都属于这个

3. 什么是 Strong AI

- (1) strong AI = AGI + ASI
- (2) 还不存在

4. 什么是 AGI

- (1) AGI = Artificial General Intelligence = general AI
- (2) have an intelligence equal to human
- (3) have self-aware consciousness

5. 什么是 ASI

- (1) ASI = Artificial Super Intelligence = superintelligence
- (2) surpass the intelligence and ability of human

6. AI 的四种程度

- (1) Reactive machines
 - no memory can't use past experiences to inform future ones
 - task-specific
 - 例子: IBM 的 chess program
- (2) Limited memory
 - have memory use past experience to inform future decisions
 - 例子: decision-making functions in self-driving cars
- (3) Theory of mind
 - have social intelligence to understand emotions
 - infer human intentions and predict behavior
- (4) Self-awareness
 - have a sense of self and consciousness

understand their current state
不存在

7. AI 的基础学科是什么?

philosophy	哲学
math	数学
economics	经济学
linguistics	语言学
neuroscience	神经学
psychology	心理学
control theory	控制理论

8. AI 是如何工作的?

learning
reasoning
self-correction
creativity

9. AI 的分支?

AI > ML > DL

10. AI 有什么优点

- | | |
|---|------------|
| (1) good at detail-oriented jobs | 细节控 |
| (2) deliver consistent results | 一致性 |
| (3) save labour and increase productivity | 节约劳动力提高生产力 |
| (4) AI-powered virtual agents are always available | 总在上班 |
| (5) reduced time for data-heavy tasks | 节省时间 |
| (6) improve customer satisfaction through personalization | 个人化 |

11. AI 有什么缺点

- | | |
|--|-----|
| (1) Expensive | 贵 |
| (2) requires deep technical expertise | 难 |
| (3) limited supply of qualified workers to build AI tools | 人才少 |
| (4) reflect the biases of its training data | 偏见 |
| (5) lack of ability to generalize from one task to another | 不通用 |
| (6) eliminates human jobs, increase unemployment rates | 抢工作 |

12. AI 在 ethics 方面会遇到的问题?

- | | |
|----------------------|------|
| (1) Misuse | 不当使用 |
| (2) Legal Concerns | 法律问题 |
| (3) Data Privacy | 数据隐私 |
| (4) Job Displacement | 取代工作 |
| (5) Training Bias | 偏见 |
| (6) Interpretability | 不理解 |

13. 详细解释

(1) Misuse

used for malicious purposes

例子: create deepfakes、 phishing attacks

(2) Legal Concerns

AI-generated libel and copyright infringement 诽谤和侵权

需要健全的法律来解决这个问题

(3) Data Privacy

在 banking、 healthcare、 law 领域的 AI 应用 rely on vast amounts of sensitive data

保证对数据隐私的保护并且遵循相关法规对保护个人隐私很重要

(4) Job Displacement

automation enabled by AI 能导致 job losses and significant disruptions in the workforce

帮助人们提高技能, 或者学习别的技能很重要

(5) Training Bias

can perpetuate biases present in the training data

导致 unfair or discriminatory 结果

例如: 会把 he 和 doctor 联系起来, 而不是使用性别中立的语言

(6) Interpretability

Deep Learning 和 GAN (generative adversarial network)的 AI 算法 can be difficult to interpret

这在具有法规遵从性要求的行业中提出了挑战, 在这些行业中, 可解释性是满足法律义务所必需的。

14. 如何解决 AI 的 ethical challenge

(1) responsible AI development

(2) robust regulations

(3) ongoing monitoring

(4) transparency

(5) stakeholder engagement

15. AI 为什么很重要

(1) Potential to Transform:

revolutionize various aspects

(2) Business Automation:

used in businesses to automate tasks

例子: customer service、 lead generation、 fraud detection、 quality control

(3) Superior Performance:

quickly analyse large volumes of documents

例子: repetitive and detail-oriented tasks

(4) Efficiency and Accuracy

complete tasks quickly with few errors

场景: analyse extensive data sets

(5) Generative AI Tools

innovative solutions and creative outputs

场景: education、marketing、product design

二、Applications of AI

1. AI 的应用有什么？

(1) Automation

automate repetitive, rules-based tasks, expanding task volume and types

(2) ML

act without explicit programming

(3) DL

automates predictive analytics

(4) CV

see and analyse visual information

(5) Self-Driving Cars

使用 CV, image recognition, DL 来 navigate + avoid obstacles

(6) Text, Image, Audio Generation

Generative AI

(7) Healthcare

improve diagnoses

mine patient data

(8) Business

enhance customer service

AI 有潜力 revolutionize product design and disrupt business models

(9) Education

automates grading

adapts to student needs

provide additional support

(10) Law

document classification

data description

outcome prediction

(11) Entertainment and Media

targeted advertising

content recommendation

script creation

automated journalism

movie production

2. RL 有什么应用

(1) Self-driving Cars

trajectory optimization

motion planning

controller optimization

learning policies for parking, lane changing, overtaking

AWS DeepRacer

- (2) Industry Automation 工业自动化
 - DeepMind cooling Google Data Center
- (3) NLP
 - process human language by computer programs
 - question answering
 - text summarization
 - machine translation
 - translation
 - sentiment analysis
 - speech recognition
- (4) Finance and Trading
 - automate trading
 - make financial decisions
 - IBM's RL-based platform for financial trades
- (5) Healthcare
 - provide treatment policies for patients
 - dynamic treatment regimes and medical diagnosis
- (6) News Recommendation 新闻推荐
 - track user preferences for personalized news recommendations
 - 考虑的因素: news features, reader behavior, context
- (7) Gaming
 - AlphaGo 通过 self-play 掌握了 the game of go
- (8) Bidding and Marketing 竞价与营销
 - enables real-time bidding to balance the trade-off between the competition and cooperation among advertisers
- (9) Robotics Manipulation 机器人操作
 - 让机器人抓取许多种没有在训练中见过的物体

三、Some other things

1. 科学技术的 grand challenges 是什么?

- (1) understand the brain
- (2) reasoning, cognition, creativity
- (3) creating intelligent machines

2. 什么是 Turing Test

- (1) rule: a judge talks with a human and a machine
- (2) goal: the judge tries to determine whether he is talking with a human or a machine

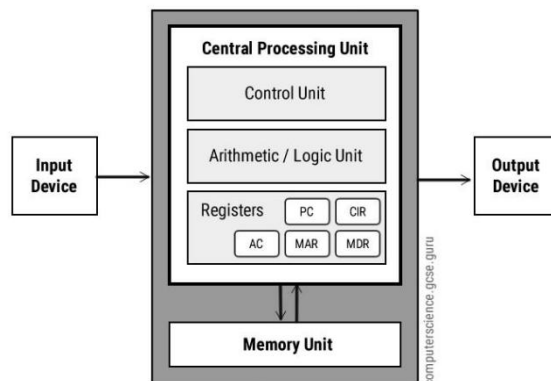
3. Turing Test 的优缺点?

- (1) 优点: serve as a method to assess the level of AI
- (2) 缺点: focus solely on external behavior
influenced by subjective judgements

4. 冯诺依曼结构 Von Neumann Architecture

input -- CPU -- output

|
memory unit



Decision making

一、Logic and uncertainty

1. 什么是 uncertainty

没有 explicitly considered in the agent's knowledge base 的 summary

2. uncertainty 必然出现的原因

an agent's incomplete or incorrect understanding of its environment

3. uncertainty 的类型

- | | |
|------------------------|--|
| (1) in prior knowledge | some causes of a disease are unknown |
| (2) in actions | actions are arbitrary |
| (3) in perception | sensors don't return exact or complete information |

4. 解决 uncertainty 的方法

- | | | |
|-----------------------------|-----|------------------------|
| (1) Default reasoning | 乐观的 | abnormalities are rare |
| (2) Worst-case reasoning | 消极的 | |
| (3) Probabilistic reasoning | 现实的 | |

5. 讲讲 Worst-case reasoning

做最坏打算，然后选择能在最坏的情况下还能得到最好结果的行动。

选择最大化效用函数的行动 maximize a utility function

这种策略在处理不确定性和风险时非常有用，可以帮助我们避免最坏的结果。

缺点：

not worth the effort

waste resources

restricted way of handling an emergency 会限制我们应对突发事件的方式，因为我们总是在准备最坏的情况，而不是灵活地应对实际发生的情况

6. 为什么 application fails?

- | | |
|---------------------------|--------------------------------------|
| (1) Laziness | too much work, too hard to use |
| (2) Theoretical ignorance | no complete theory (medical science) |
| (3) Practical ignorance | not all tests can be run |

二、Probability theory

1. 什么是 Decision theory

Decision theory = utility theory + probability theory

2. 什么是 utility theory

represent and infer preferences

3. conditional probability

$$P(A \cap B) = P(A|B) P(B)$$

题目 1

A university has three residences for students. The first has 500 single bedrooms, the second has 1000 and the third has 2000. When a student is allocated randomly to a residence, what is the probability that it is the second residence?

$$\Pr(\text{Second Residence}) = \frac{1000}{500 + 1000 + 2000} = \frac{1000}{3500} = 0.29$$

题目 2

What is the probability that I draw a two or a four when I pick a card at random from a pack of playing cards?

$$\Pr(2 \text{ or } 4) = \Pr(2) + \Pr(4) = \frac{4}{52} + \frac{4}{52} = 0.15$$

题目 3

An air steward is training with a well-known UK airline:

He knows that of the 200 trained stewards currently employed by the Company; 80 work on the London-Paris route;

50 on London-Amsterdam and the remainder on internal flights.

What is the probability that, after training, the new steward's work will take him out of the UK?

$$P(\text{Out of UK}) = P(\text{London - Paris}) + P(\text{London - Amsterdam}) = \frac{80}{200} + \frac{50}{200} = 0.65$$

题目 4

A company has an important order to manufacture next week. The order requires two components, A and B , to be delivered in advance by two separate companies. The chance of component A arriving this week is $5/6$. The chance of component B arriving this week is $1/2$. What is the chance that:

- A and B arrive in time?
- A or B arrive in time?

$$P(A \cap B) = P(A) \times P(B) = 5/6 \times 1/2 = \mathbf{5/12}$$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) = 5/6 + 1/2 - 5/12 = \mathbf{11/12}$$

题目 5

A production line has 3 machines in series. The probability each day that they breakdown is $1/10$, $1/25$ and $1/50$ respectively. What is the probability that **at least one** of the machines breaks down on a given day?

“At least one” so calculate probability that all working

	M1	M2	M3
P(breakdown)	1/10	1/25	1/50
P(works o.k.)	9/10	24/25	49/50

Assume independence:

$$P(\text{all work}) = P(\text{M1 works}) \times P(\text{M2 works}) \times P(\text{M3 works}) \\ = 9/10 \times 24/25 \times 49/50 = 10584/12500$$

$$P(\text{at least one breaks down}) \\ = 1 - P(\text{all work}) = 1 - 10584/12500 = 1916/12500 = \mathbf{0.15}$$

题目 6

	Toothache	¬Toothache
Cavity	0.04	0.06
¬Cavity	0.01	0.89

$$P(\text{Cavity} \cap \text{Toothache}) = P(\text{Cavity}|\text{Toothache}) \times P(\text{Toothache})$$

$$P(\text{Cavity}|\text{Toothache}) = P(\text{Cavity} \cap \text{Toothache}) / P(\text{Toothache}) \\ = 0.04/0.05 = \mathbf{0.8}$$

三、Random variables & Probability & Bayer's

1. 什么是 Random variables

Variable	Domain
<i>Age</i>	$\{1, 2, \dots, 120\}$
<i>Weather</i>	$\{\text{sunny, dry, cloudy, rain, snow}\}$
<i>Size</i>	$\{\text{small, medium, large}\}$
<i>Blonde</i>	$\{\text{true, false}\}$

2. 什么是 Probability distribution

Example:

$$\begin{aligned}
 P(\text{Weather} = \text{sunny}) &= 0.6 \\
 P(\text{Weather} = \text{rain}) &= 0.2 \\
 P(\text{Weather} = \text{cloudy}) &= 0.18 \\
 P(\text{Weather} = \text{snow}) &= 0.02
 \end{aligned}$$

Then $P(\text{Weather}) = \{0.6, 0.2, 0.18, 0.02\}$

$P(\text{Weather})$ is called a probability distribution for the random variable Weather

3. 什么是 Expected value

The expected value of U after doing A is

$$E[U] = \sum_{i=1, \dots, n} p_i U(x_i)$$

Function U of X

E.g., U is the utility of a state

❖ Random variable X with n values x_1, \dots, x_n and distribution (p_1, p_2, \dots, p_n)

E.g.: X is the state reached after doing an action A under uncertainty

4. 什么是 JPD

(1) Joint Probability Distribution

Example

$Sky : \{sunny, cloudy, rain, snow\}$

$Wind : \{true, false\}$

$P(Wind, Sky) =$

	sunny	cloudy	rain	snow
true	0.3	0.15	0.17	0.1
false	0.3	0.05	0.01	0.01

(2) 例题

	$S = sunny$	$S = cloudy$	$S = rain$	$S = snow$	$P(W)$
W	0.3	0.15	0.17	0.01	0.63
$\neg W$	0.3	0.05	0.01	0.01	0.37
$P(S)$	0.6	0.20	0.18	0.02	1.00

- $P(S = rain \cap W) = 0.17$
- $P(rain) = P(rain \cap W) + P(rain \cap \neg W) = 0.17 + 0.01 = 0.18$
- $P(W) = 0.30 + 0.15 + 0.17 + 0.01 = 0.63$
- $P(rain | W) = P(rain \cap W) / P(W) = 0.17 / 0.63 = 0.27$

最下面的那个容易算错

(3) 例题

	Toothache	\neg Toothache
Cavity	0.04	0.06
\neg Cavity	0.01	0.89

$$P(\text{Toothache}) = 0.04 + 0.01 = 0.05$$

$$P(\text{Toothache} \cup \text{Cavity}) = 0.04 + 0.01 + 0.06 = 0.11$$

5. 什么是 Bayes's rule

$$P(B | A) = \frac{P(A | B) P(B)}{P(A)}$$

6. 什么是 CPT

conditional probability table

7. 例题

A 是 independent of B

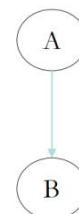
B 是 conditionally dependent on A

$$P(\neg A) = 1 - P(A)$$

$$P(\neg B | A) = 1 - P(B | A)$$

$$P(\neg B | \neg A) = 1 - P(B | \neg A)$$

其实就是把最前面的字母加否定，其他的字母不用动



ML

一、Many kinds of learning

1. 什么是学习

goal-directed process of a system that
improves the knowledge by exploring experience and prior knowledge

2. 什么是 ML

- (1) 创造算法 that can learn from data + make decisions based on patterns observed
- (2) need human intervention
- (3) rely on human-designed feature representations
(optimize weights to make a best final prediction 优化一下权重)

3. 什么是 DL

- (1) uses an artificial neural network to reach accurate conclusions
- (2) no human intervention,
- (3) use multiple layers to learn data representations
- (4) a subfield of ML

4. 什么是 Supervised Learning

- (1) learn with **labeled** data
- (2) observe **input-output pairs** and learns **a function** that maps from input to output
- (3) suitable for **predictive data labeling**
- (4) 例子: **classification** (email, image)
regression (predicting real-valued outputs)
object detection
semantic segmentation
image captioning

5. 什么是 Unsupervised Learning

- (1) discover patterns in **unlabeled** data
- (2) learn the hidden features, structures of data **without feedback**
- (3) suitable for **describing data**
- (4) 例子: **clustering**
dimensionality reduction

6. 什么是 RL

- (1) use **unlabeled** data
- (2) learn from the **environment** by interacting with it
- (3) perform actions based on **rewards** as **feedback**
- (4) maximize the **reward** by taking right **actions**.

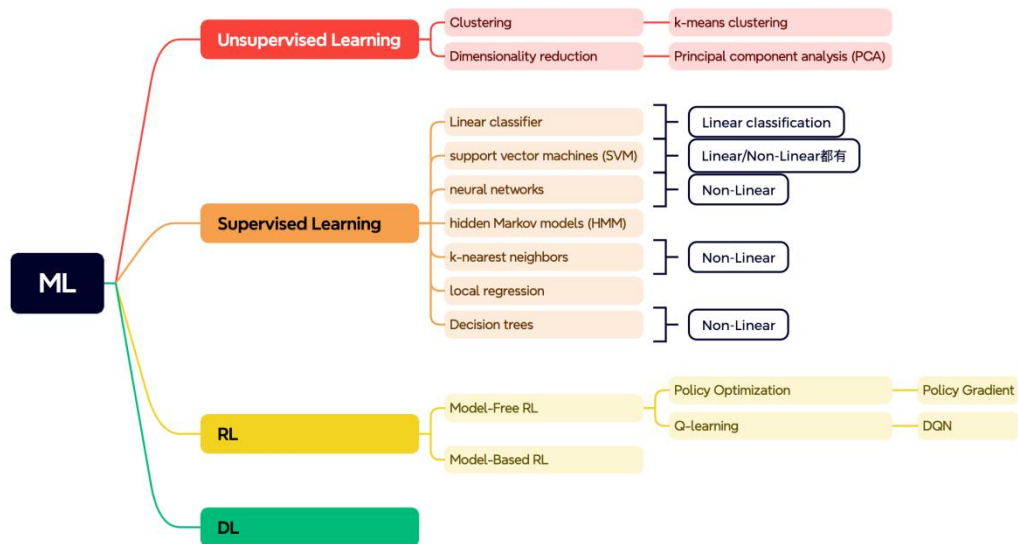
(4) 例子: learn to play Go 下围棋

7. 常见的 ML 有哪些?

unsupervised learning + supervised learning + reinforcement learning

8. ML 是如何工作的

- (1) **model** 模型
problem modelling 建模
define the assumption space
- (2) **strategy** 策略
determine objective function 选函数
- (3) **algorithm** 算法
solve for model parameters 调参



Presented with xmind

Supervised Learning

一、Decision Trees

1. 什么是 decision tree

- (1) a simple and efficient form of learning from examples
- (2) maps [object with attributes] to [discrete values] according to the values of the attributes
- (3) act as classifier
- (4) make predictions by recursively splitting on different attributes according to the tree structure

2. Decision tree 的局限性

- (1) noise
- (2) overfitting
- (3) missing data
- (4) multi-valued attributes (颜色可能是红黄蓝)
- (5) continuous-valued attributes (一个人的体重在一个区间内浮动)

3. 什么是 goal predicate

the predicate to be implemented by a decision tree

4. 什么是 training set

the set of examples used to build the tree

5. 什么是 positive example

satisfy the goal predicate

6. 什么是 information theory

help us to choose the best attribute

choose attribute that gives the highest gain

7. 什么是 entropy

a measure of disorder or uncertainty, basis of constructing a decision tree

low entropy --> value sampled are more predictable

Entropy

$$H(X) = - \sum_{i=1}^n P(X=i) \log_2 P(X=i)$$

Conditional Entropy

$$H(Y|X) = - \sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 p(y|x)$$

8. low entropy 是什么

distribution of variable has many peaks and valleys

histograms have many low and highs

value sampled are more predictable

9. high entropy 是什么

variable has uniform like distribution

flat histogram

value sampled are less predictable

10. 什么是 information gain

measures the reduction in entropy by splitting a dataset according to a given value of a random variable

$$I(X_n, Y) = H(Y) - H(Y|X_n)$$

n = number of splits

11. 最后选择 information gain 最大的 attribute 作为 decision tree 的 root

计算题 1

1) Entropy of fair coin toss

$$H(P(h), P(t)) = H\left(\frac{1}{2}, \frac{1}{2}\right) = -\frac{1}{2} \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2} = \frac{1}{2} + \frac{1}{2} = 1 \text{ bit}$$

2) Entropy of a loaded coin toss where $P(head) = 0.99$

$$\begin{aligned} H(P(h), P(t)) &= H\left(\frac{99}{100}, \frac{1}{100}\right) \\ &= -0.99 \log_2 0.99 - 0.01 \log_2 0.01 \approx 0.08 \text{ bits} \end{aligned}$$

3) Entropy of a loaded coin toss with heads on both side

$$H(P(h), P(t)) = H(1, 0) = -1 \log_2 1 - 0 \log_2 0 = 0 - 0 = 0 \text{ bits}$$

计算题 2

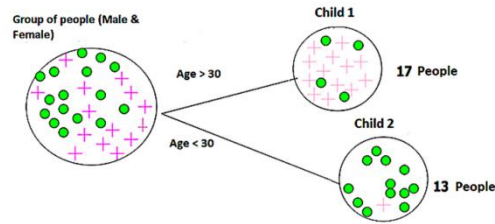
$$P(yes) = \frac{p}{p+n} \qquad P(no) = \frac{n}{p+n}$$

□ Entropy of a correct decision tree:

$$H\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

计算题 3

Example:



Find:

- 1) Entropy $H(\text{People})$;
- 2) Entropy $H(\text{Child}_1)$;
- 3) Entropy $H(\text{Child}_2)$;
- 4) Information Gain I for 1) - 3).

$$1) H(\text{People}) = -\left(\frac{14}{30} \log_2 \frac{14}{30}\right) - \left(\frac{16}{30} \log_2 \frac{16}{30}\right) = 0.996$$

$$2) H(\text{Child}_1) = -\left(\frac{13}{17} \log_2 \frac{13}{17}\right) - \left(\frac{4}{17} \log_2 \frac{4}{17}\right) = 0.787$$

$$3) H(\text{Child}_2) = -\left(\frac{1}{13} \log_2 \frac{1}{13}\right) - \left(\frac{12}{13} \log_2 \frac{12}{13}\right) = 0.391$$

$$4) \text{Weighted average entropy of children} = -\left(\frac{17}{30} \cdot 0.787\right) - \left(\frac{13}{30} \cdot 0.391\right) = 0.615$$

Information Gain $I = 0.996 - 0.615 = 0.38$ for the split

计算题 4

Question: Find Entropy of a Joint Distribution $H(X, Y)$

	Cloudy	Not cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

$$X = \{\text{Raining}, \text{Not Raining}\}, Y = \{\text{Cloudy}, \text{Not cloudy}\}$$

Entropy

$$H(X, Y) = -\sum_x \sum_y P(x, y) \log_2 P(x, y)$$

$$H(X, Y) = -\frac{24}{100} \log_2 \frac{24}{100} - \frac{1}{100} \log_2 \frac{1}{100} - \frac{25}{100} \log_2 \frac{25}{100} - \frac{50}{100} \log_2 \frac{50}{100} \approx 1.56 \text{ bits}$$

计算题 5

	Cloudy	Not cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

Find:

- 1) Entropy of Cloudiness Y .
- 2) Entropy of Cloudiness Y , given it is raining?
- 3) Entropy of Cloudiness Y , given whether or not it is raining?
- 4) How much information about cloudiness do we get by discovering whether it is raining?

答案:

1) Entropy of Cloudiness Y .

$$H(Y) = -\frac{24}{100} \log_2 \frac{24}{100} - \frac{25}{100} \log_2 \frac{25}{100} \approx 0.994 \approx 1 \text{ bit}$$

2) Entropy of Cloudiness Y , given it is raining?

$$\begin{aligned} H(Y|X = \text{raining}) &= - \sum_{y \in Y} p(y|\text{raining}) \log_2 p(y|\text{raining}) \\ &= -\frac{24}{25} \log_2 \frac{24}{25} - \frac{1}{25} \approx 0.24 \text{ bits} \end{aligned}$$

$$p(y|x) = \frac{p(x,y)}{p(x)} \text{ and } p(x) = \sum_y p(x,y) \text{ (sum in a row)}$$

二、Classification

1. classification 都有什么种类?

- (1) nearest neighbor classifier
- (2) k-nearest neighbors classifier
- (3) linear classifier
- (4) SVM
- (5) Binary Classification
- (6) Multi-class Classification

2. 什么是 Nearest Neighbor Classifier

- (1) assign the class label of the nearest training data point for each test data point
- (2) 采用一个 distance function 来找到最近的 neighbor
- (3) 每个 pixel 之间的距离都计算了

3. Nearest Neighbor Classifier 有什么缺点

- (1) must remember all training data and store it for future comparisons
- (2) require comparisons between all training images
- (3) expensive

4. 什么是 k-Nearest Neighbors Classifier

- (1) consider multiple neighboring data points to classify a test data point
- (2) the class of test example is obtained by voting according to the distance to the 3 closest points

5. 什么是 Linear

- (1) input data 能 linearly separable 的话就可以使用 linear classifier
- (2) 线性是指可以用直线（在二维情况下）、平面（在三维情况下）或超平面（在更高维情况下）表示的关系。在机器学习中，线性模型是指模型的输出是输入特征的线性组合。

6. 什么是 Non-Linear

- (1) result in non-linear decision boundaries
- (2) deal with non-linearly separable data
- (3) Features are obtained as non-linear functions of the inputs
- (4) 非线性是指不能用直线、平面或超平面表示的关系。在机器学习中，非线性模型可以捕获输入和输出之间更复杂的关系。

7. 什么是 Linear Classifier

- (1) find a linear function of the inputs that separates the classes
- (2) use pairs of inputs and labels to find weights matrix W and bias vector b
- (3) decision boundary 是线性的
2D 空间里是 straight line, 3D 空间里是 flat plane, 3D 及以上是 hyperplane
- (4) 是高级 classification 算法 (SVM, NNs) 的基石

(5) 包括 logistic regression, linear SVM, Perceptron

8. 找 Linear Classifier 中最好参数的方法?

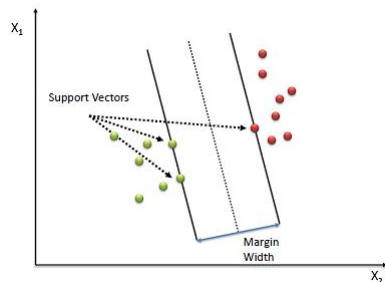
Perceptron

详细讲:

- (1) parameters are updated until a minimal error is reached
- (2) single layer
- (3) doesn't use backpropagation (深度学习那里详细讲了)

9. 什么是 support vectors

data points that define the maximum margin width

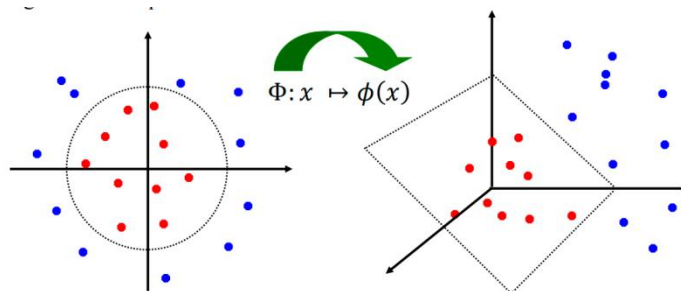


10. 什么是 SVM

- (1) Support Vector Machines
- (2) solve optimization problem
identify a decision boundary that correctly classifies the examples
increase the geometric margin between the boundary and all examples
- (3) suitable for Binary Classification
- (4) find a plane to separate different kinds of data

11. 什么是 Non-linear SVM

- (1) [original input space] is mapped to [a higher-dimensional feature space where the training set is linearly separable]
- (2) define a non-linear kernel function to calculate a non-linear decision boundary in the original feature space
- (3) 对于线性不可分的数据, SVM 使用核技巧将数据映射到高维空间使其变得线性可分, 然后在那里找到超平面。



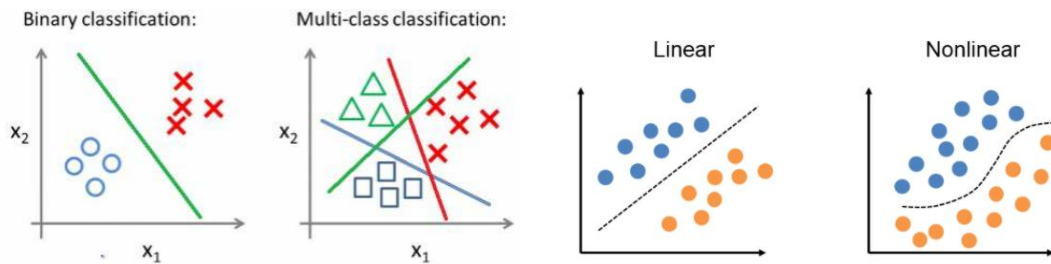
12. 什么是 Binary Classification

- (1) output labels: 0 or 1

- (2) 例子: benign or malignant tumor 良性 or 恶性肿瘤
spam or no-spam email 垃圾 or 有用邮件
- (3) 二分类是指分类任务只有两个类别

13. 什么是 Multi-class Classification

- (1) 分成 3 个或更多种
- (2) 例子: 手写数字识别, 0 到 9 有 10 个类别
- (3) Binary 和 multi-class 都可以 linearly 或 non-linearly separate



14. 什么是 No-Free-Lunch Theorem

No single classifier works the best for all possible problems

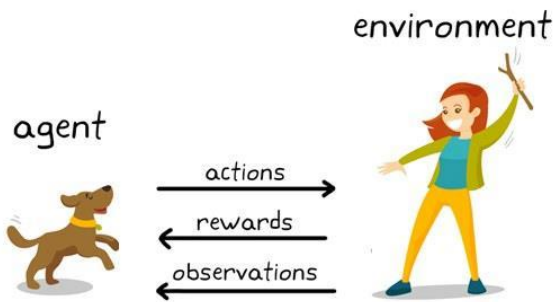
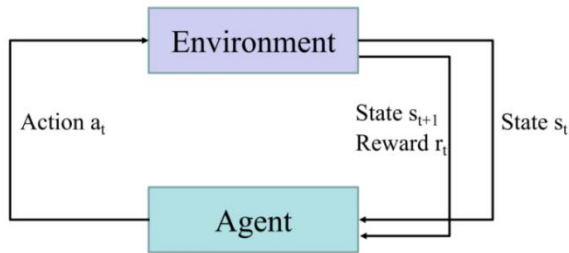
RL

一、Introduction

1. RL 的 key elements?

environment、agent、state、action、reward

2. RL 的 workflow?



3. RL 都有哪些分类?

model-based

model-free

policy-based

value-based

on/off-policy



4. 什么是 Model-free RL

- (1) learn strategies directly **without explicit model of the environment**
- (2) agent interacts with the real environment
- (3) rely on real environment feedback and reward
- (4) 例子: learn from doing and interacting with the real world

5. 什么是 Model-based RL

- (1) **model the environment** and plan future actions
- (2) agent constructs a simulated model
- (3) 收到的信息是 transition probability 和 reward
- (4) 例子: theorizing in our mind before taking an action

6. 什么是 Value-based RL

- (1) select the action that has the highest value function in a given state
- (2) 优点: find the optimal policy **efficiently**
 - have high sample efficiency
- 缺点: can't solve problem with continuous action space
 - sensitive to hyperparameters

7. 什么是 Policy-based RL

- (1) select actions directly by learning policy functions
- (2) 优点: deal with **continuous** action spaces
 - easier to converge in real environment
- 缺点: require more training data
 - often converge to a local optimum

8. 什么是 On-policy RL

- (1) learn from the policy that is currently followed during exploration
- (2) 例子: learn from our own experience

9. Off-policy RL

- (1) learn from a different policy instead of the currently followed one
- (2) learn from others to gain experience

二、MDP

1. 什么是 Markov Process / Markov Chain

- (1) a stochastic model describing a sequence of possible events
- (2) the probability of each event is only related to the state attained in the previous event
- (2) stationary assumption: state transition probabilities has nothing to do with time

2. 什么是 MDP

- (1) Markov Decision Process
- (2) 是 RL 的 mathematical description
- (3) 定义: $\langle S, A, R, P, \gamma \rangle$
 - S: states
 - A: actions
 - R: reward
 - P: transition probability
 - γ : discount factor in $[0, 1]$ 如果重视 long-term 回报, 那么就让 γ 大一点

3. MDP 的过程?

environment initializes a state at time step $t=0$
agent selects action a_t
environment returns reward r
environment gives next state s_{t+1}
agent receives reward r_t and next state s_{t+1}

4. Deterministic 和 Stochastic 的区别

- (1) Deterministic: 下一个 state 和相应 reward 只被现在 state 以及选择的 action 有关
- (2) Stochastic: 下一个 state 和相应 reward 被 a probability distribution 决定

三、Value-based RL

1. 什么是 state value function?

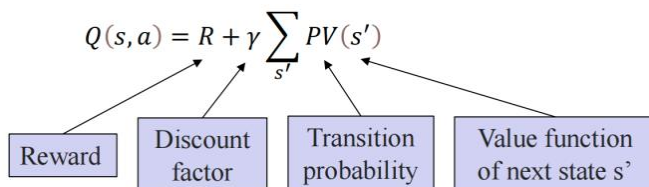
- (1) expected accumulative reward an agent can obtain in the current environment state
- (2) help the agent evaluate the goodness of states to make better decisions
- (3) 评估一个 state

$$V^{\pi}(s) = E\left(\sum \gamma^t r_{t+1} | s_t = s, \pi\right)$$

2. 什么是 Q-value function?

- (1) expected accumulative reward form taking action a at state s
- (2) 也叫 state-action value function
- (3) 评估一个 state-action pair
- (4) 给一个 state and action 作为 input, 然后就会 output Q-value

$$Q^{\pi}(s, a) = E\left(\sum \gamma^t r_{t+1} | s_t = s, a_t = a, \pi\right)$$



3. 什么是 Bellman equation

- (1) help agent to iterate on value, thus optimizing the policy
- (2) 现在的 reward 加上 expected reward from the future actions

$$\mathbb{E}\left[R_{t+1} + \gamma V(S_{t+1}) | S_t = s\right]$$

4. 什么是 Q-learning

- (1) use function approximator to estimate Q-value function
- (2) function approximator 是一个 Q-table
- (3) 步骤

initialize the Q-table

[choose an action

execute the action and obtain a reward

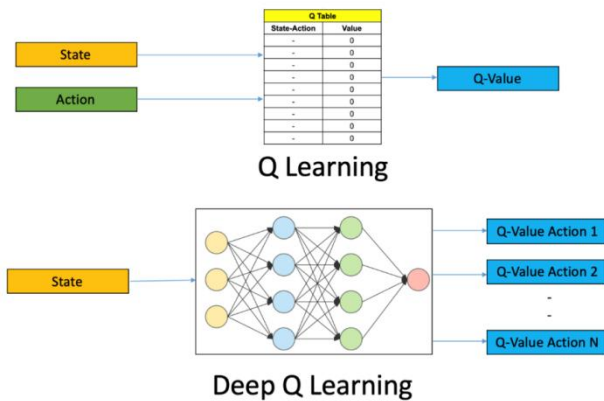
update the Q-value in Q-table]循环

5. 什么是 Deep Q-learning

- (1) 使用 DQN (Deep Q-network)
- (2) function approximator 是一个 DNN
- (3) 有太多 states 或 continuous state space 的时候, 我们不能列出所有的 state-action pair in the Q-table

所以使用 DNN (deep neural network) 来 make decisions in extremely large domains

- (4) represent value function by Q-network with weights ω
- (5) 用 SGD (stochastic gradient descent) 来 optimize loss function



6. 训练 DQN 有什么方法?

- (1) experience replay 经验回放
- (2) fixed Q-target 固定 Q 目标

7. 什么是 experience replay

- (1) 把每一个状态转变和决策存储在一个叫做经验回放内存的数据结构中。
在训练时不使用最新的经验，而是从这个内存中随机抽取一部分经验来进行学习。
好处：打破数据之间的关联性，提高学习的稳定性。
- (2) remove correlations of training data 的方法：
store dataset from prior experience
- (3) 步骤
 - a. sample an experience tuple from the dataset
 - b. compute the target value for the sampled state
 - c. use SGD to update the network weights

8. 什么是 Fixed Q-target

- (1) 为了增加稳定性，fix the target network weights

四、Policy-based RL

1. 什么是 policy

a function from S to A that specifies what action to take in each time

2. optimal policy 的目标是什么?

find an optimal policy that maximizes accumulative discount reward $\sum_{\geq 0} \gamma^t r_t$

3. 什么是 policy gradient

define a class of parametrized policies

define its value based on the discounted accumulative reward

do gradient ascent on policy parameters 唯一的梯度上升

4. Policy-based RL 比 Value-based RL 有什么优点

(1) Value-based 的 Q-function 可能会很复杂, 不可能算出每个 state-action pair 的 value

例如: a robot grasping an object 有 high-dimensional 的 state 和 action

(2) 可以直接学习一个 policy, 简单很多

5. 什么是 Actor-critic architecture

(1) combine value-based and policy-based

(2) actor: decide which action to take in a given state, learn the policy

critic: evaluate the value of the action at the state, estimates the value function

DL

一、Introduction

1. 深度学习的原理

apply a multi-layer process to learn rich hierarchical features

例子: data representations

2. 深度学习的过程

Input \rightarrow Feature extraction + Classification \rightarrow Output

3. 机器学习和深度学习的过程区别

ML 的 feature extraction 是人做的

4. 深度学习的优点/特点, 为什么深度学习有用

- (1) require large amounts of training data
- (2) learn in supervised and unsupervised manner
- (3) effective at learning patterns
- (4) represent an effective end-to-end learning system
- (5) provide a flexible, learnable framework to represent visual, text, linguistic

information

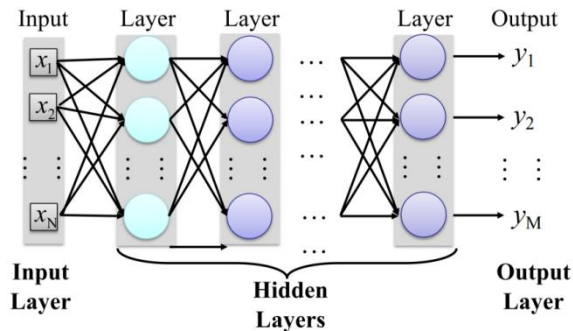
- (6) outperform other ML techniques

5. 什么是神经网络 (neural networks) (NN)

- (1) NN can approximate any arbitrary complex continuous function
- (2) use nonlinear mapping of the inputs x to the outputs $f(x)$ to compute complex decision boundaries

6. 什么是 Deep 神经网络

- (1) 有许多 hidden layers
- (2) Fully-connected layers (也即 Multi-Layer Perceptron or MLP)
- (3) Each neuron is connected to all neurons in the succeeding layer

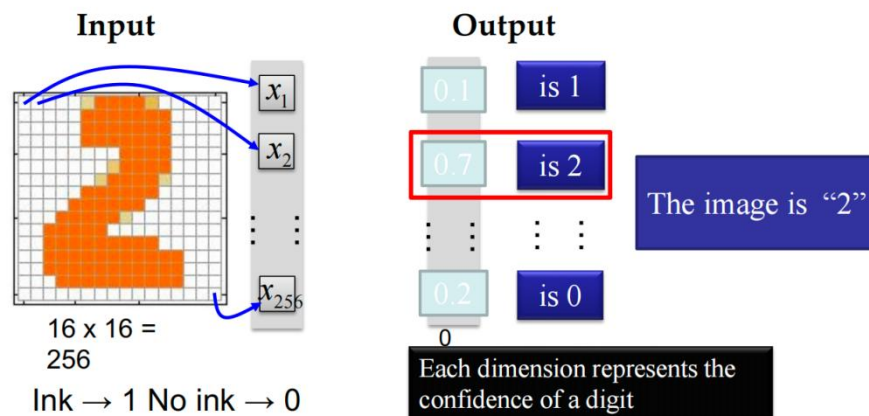


7. 为什么用 deep NN?

- (1) 实际上 deep NN 更好只是个 empirical observation
- (2) 数学上, deep 的和 one-layer 的 representational power 一样
- (3) deep 的表现更好, 但是在 a certain number of layers 之后, the performance plateaus

8. 什么是 MNIST

- (1) MNIST: Modified National Institute of Standards and Technology database
- (2) 是 NN 的一个例子
- (3) 是一个 handwritten digit recognition
- (4) input: the intensity of each pixel 每个像素的强度
- (5) output: the class of the digit 数字的类, 也就是哪个数字



每个维度表示数字置信度, 出来的是一个概率, 70%的可能是 2

二、Elements of NN

1. 神经网络的 elements 是什么？分别有什么作用？

- (1) neurons fundamental units of NN, map inputs into an output number
- (2) layers facilitate the [structured arrangement and processing] of data
- (3) weights parameters that transform input data
- (4) biases parameters that allow for greater flexibility
- (5) activation functions introduce non-linearities

2. 重要公式 $\sigma(Wx + b) = a$

 $\sigma(z)$: activation function

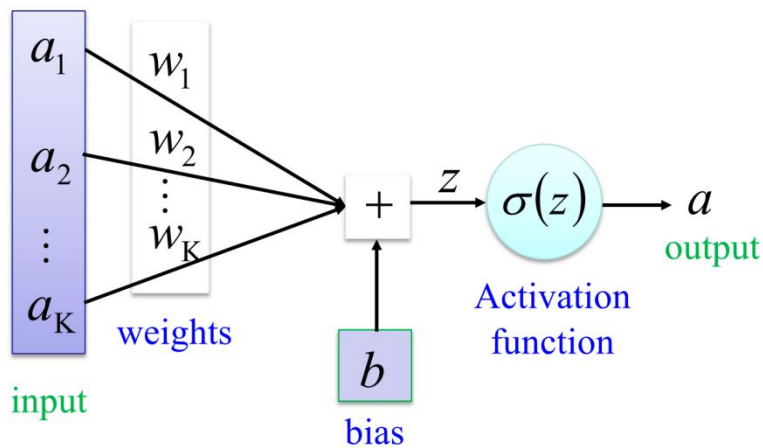
W: weights

x: input

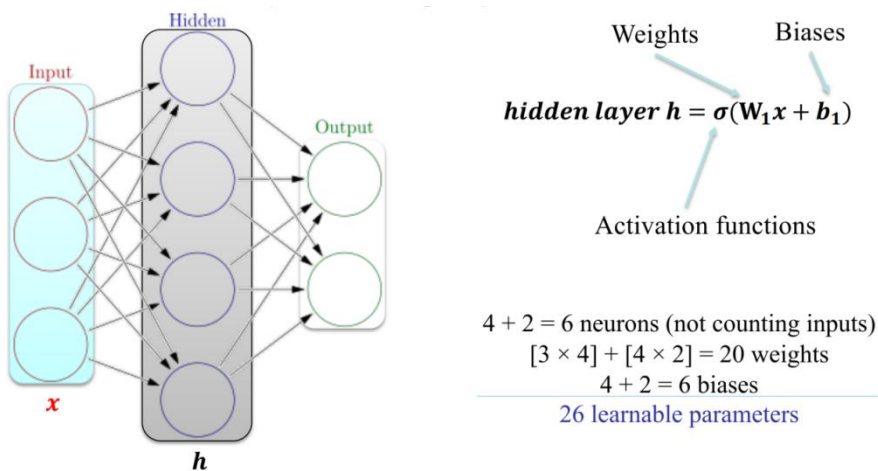
b: bias

a: output

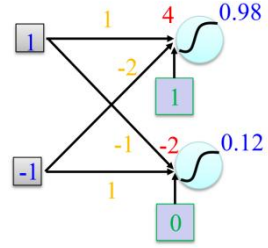
3. 需要会画的 diagram:



4. 算 neurons 例子:



$$\sigma(Wx + b) = a$$

$$\sigma\left(\begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}\right) = \begin{bmatrix} 0.98 \\ 0.12 \end{bmatrix}$$


The diagram illustrates a simple neural network with two input nodes (1, -1), two hidden nodes (1, 0), and two output nodes (0.98, 0.12). The weights for the connections are 1, -2, -1, and 1. The biases for the hidden nodes are 1 and 0. The output nodes are represented by circles with a sigmoid function symbol, and their values are 0.98 and 0.12.

5. 太多 neurons 的优缺点:

- (1) 优点: improve representation
- (2) 缺点: may overfit

三、Activation Function

1. 常用的激励函数有什么？

- (1) Linear function the output signal is proportional to the input signal
- (2) Sigmoid function [0,1]
- (3) Tanh function [-1,1]
- (4) ReLU threshold at zero

2. 什么是 Linear function

$$f(x) = cx$$

- (1) 用在 regression problems 里面
- (2) the output signal is proportional to the input signal
- (3) 如果 $c=1$ ，线性激励函数就叫 identity activation function 恒等激励函数

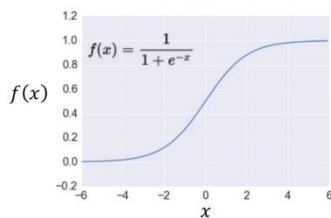
3. 什么是 Non-linear activations

- (1) 用在 non-linear data representations 里面
- (2) NN with large number of layers can approximate complex functions

4. 什么是 Sigmoid function (s 形的)

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

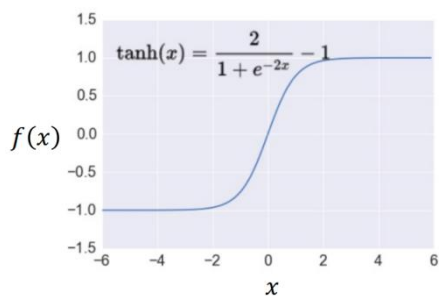
- (1) squash a number into the range between 0 and 1
- (2) 用的少 (less common in modern NNs)
- (3) When the neuron's activation is 0 or 1, sigmoid neurons saturate
Gradients at these regions are almost zero



5. 什么是 Tanh function

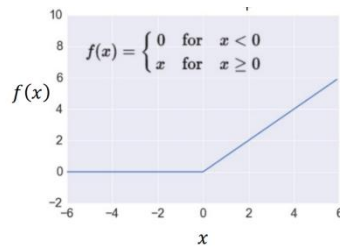
$$\tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

- (1) squash a number into the range between -1 and 1
- (2) the output is zero-centered (对称, 所以比 sigmoid 更好)
- (3) 和 sigmoid 一样会 saturate



6. 什么是 ReLU (Rectified Linear Unit) $f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$

- (1) take a real-valued number and threshold at zero (从 0 开始)
- (2) form: linear+non-saturating 上面没有平台, 所以不会 saturate



7. ReLU 的优缺点?

(1) 优点:

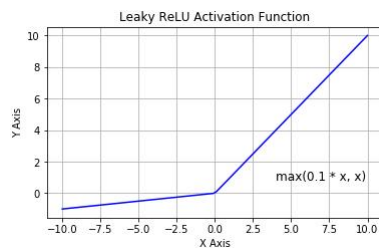
- fast to compute than others
- threshold a matrix at zero
- accelerate the convergence of gradient descent
- prevent the gradient vanishing problem (linear, non-saturating)

(2) 缺点:

- the gradients can become zero and the neuron will not activate again
- 例: learning rate 太大的时候

8. 什么是 Leaky ReLU

(1) $x < 0$ 时有 small slope(0.01 的斜率什么的)

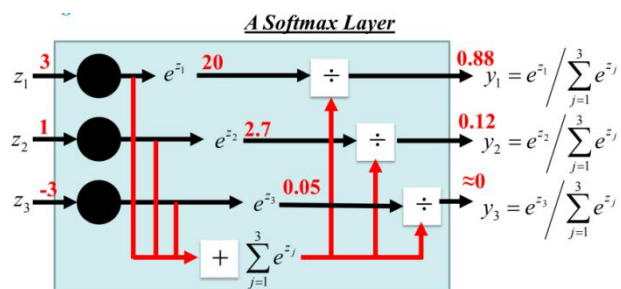


9. 什么是 Softmax

- (1) output a probability value between 0 and 1
- (2) Probability:

$$0 < y_i < 1$$

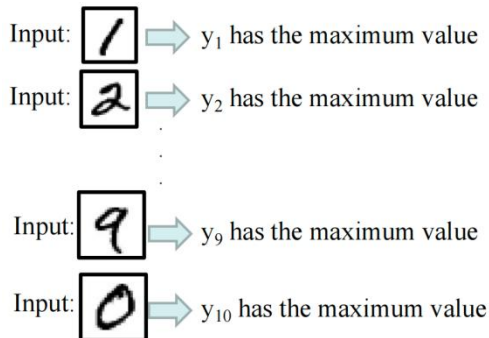
$$\sum_i y_i = 1$$



四、Tasks in training NN

1. 为什么要训练 NN

To train a NN, set the parameters θ such that for a training subset of images, the corresponding elements in the predicted output have maximum values



2. 机器学习的难点: Train a model to learn a set of parameters θ that are optimal

3. 参数 θ 都有什么

weight matrices and bias vectors from all layers

$$\theta = \{W^1, b^1, W^2, b^2, \dots, W^L, b^L\}$$

(公式: $\sigma(Wx + b) = a$)

4. 什么是 Data preprocessing

(1) help convergence during training

(2) Mean subtraction (水平垂直居中对齐)

结果: obtain zero-centered data

方式: subtract the mean for each individual data dimension (feature)

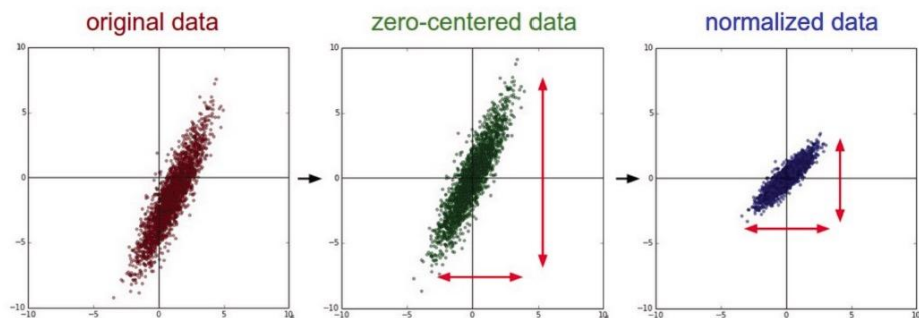
(3) Normalization (拉伸)

方式 1: Divide each feature by its standard deviation

方式 2: scale the data within the range [0,1] or [-1, 1]

结果: obtain standard deviation of 1 for each data dimension (feature)

例: image pixel intensities are divided by 255 to be scaled in the [0,1] range



五、Loss function

1. 什么是 loss function

- (1) calculate the difference between the model prediction and the true label
- (2) 可以是 mean-squared error, cross-entropy
- (3) 如果有 N 个图片的训练集, 计算 total loss over all images: $L(\theta) = \sum_{n=1}^N L_n(\theta)$

2. Classification Tasks 的 loss function 是什么

- (1) Cross-entropy

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K \left[y_k^{(i)} \log \hat{y}_k^{(i)} + (1 - y_k^{(i)}) \log (1 - \hat{y}_k^{(i)}) \right]$$

- (2) y_i 是 ground-truth class labels
 \hat{y}_i 是 model predicted class labels

3. Regression Tasks 的 loss function 是什么

- (1) mean squared error 或 mean absolute error

$$\text{Mean Squared Error} \quad \mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n (y^{(i)} - \hat{y}^{(i)})^2$$

$$\text{Mean Absolute Error} \quad \mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n |y^{(i)} - \hat{y}^{(i)}|$$

六、Gradient descent (vanilla GD)

1. 什么是梯度下降算法?

- (1) find the optimal parameter θ to minimize the loss
- (2) apply iterative refinement of the parameter θ 迭代细化
- (3) use the opposite direction of the gradient of the loss function to update θ

例: $\nabla L(\theta) = \frac{\partial L}{\partial \theta_i}$

(4) The gradient of the loss function $\nabla L(\theta)$ gives the direction of fastest increase of the loss function $L(\theta)$

2. GD 算法的步骤?

- (1) Randomly initialize the parameters, θ^0
- (2) Compute the gradient of the loss function using backpropagation at the initial parameters θ^0 : $\nabla L(\theta^0)$
- (3) Update the parameters: $\theta^{\text{new}} = \theta^0 - \alpha \nabla L(\theta^0)$ (α is the learning rate)
- (4) Go to step 2 and repeat

3. GD 的问题

- (1) the local minima problem
- (2) very slow at plateaus $\nabla L(\theta) \approx 0$
- (3) get stuck at saddle points $\nabla L(\theta) = 0$ 鞍点
- (4) Vanishing/Exploding Gradient Problem

4. 什么是 backpropagation

- (1) calculate the gradient of the loss function
- (2) traverse the network in reverse order (from the outputs y backward towards the inputs x 来计算 $\nabla L(\theta)$)
- (3) the chain rule is used to compute the partial derivatives of the loss function
- (4) each update for θ takes one forward and one backward pass
- (5) 题外话, 现在 scikit-learn 有 automatic calculation of the gradients, 不需要手算

5. backpropagation 有什么缺点

- (1) wasteful to compute the loss over the entire training dataset to perform a single parameter update for large datasets. (例: ImageNet 有 14M images)

6. back propagation 用在了哪里?

- (1) GD 算法用了
- (2) RNN 那里也用了
- (3) Perceptron 没用

7. 什么是 forward propagation

pass the input x through the hidden layers to obtain model outputs y

8. 什么是 Learning Rate

- (1) gradient: 告诉我们 direction in which the loss has the steepest rate of increase
- (2) learning rate: 告诉我们 how far along the opposite direction we should step

9. learning rate 快慢的影响?

- (1) 太快: the loss increases or plateaus too quickly
- (2) 太慢: the loss decreases too slowly + takes many epochs to reach a solution

10. 什么是 local minima problem

- (1) stop when a local minimum of the loss surface is reached
- (2) don't guarantee a global minimum
- (3) loss surface $L(\theta)$ can be complex
- (4) random initialization in NN will cause different initial parameters θ^0
each GD may result in different minima
NN may produce different outputs

11. 什么是 Vanishing/Exploding Gradient Problem

- (1) 梯度太小 Vanishing, 导致很小的 update of the parameters, 学的很慢
- (2) 梯度太大 Exploding, 导致很大的 update of the parameters

12. 怎么解决 Vanishing/Exploding Gradient Problem

- (1) change learning rate
- (2) ReLU activations (linear, non-saturating)
- (3) Regularization (也解决过拟合)
- (4) LSTM (有一个 memory cell, update at each step in the sequence)

13. 什么东西 mitigate 了 Vanishing/Exploding Gradient Problem

- (1) ResNet vanishing

14. 什么东西会出现 Vanishing/Exploding Gradient Problem

- (1) GD vanishing/exploding
- (2) RNN vanishing

七、GD optimization

1. 梯度下降法的 optimization methods 有哪些?

- (1) mini-batch GD
- (2) stochastic GD with momentum
- (3) Adam

2. Mini-batch GD 的步骤

- (1) compute the loss on a mini-batch of images
- (2) update the parameters θ
- (3) repeat until all images are used
- (4) at the next epoch, shuffle the training data and repeat the above process

3. Mini-batch GD 的优点

- (1) faster training

4. Mini-batch 能用的原理

- (1) the gradient from a mini-batch is the gradient from the entire training set

5. 什么是 Stochastic Gradient Descent

- (1) use mini-batches that consist of a single input example
例: one image mini-batch

6. SGD 的优缺点?

- (1) 优点: very fast
- (2) 缺点: may cause significant fluctuation in the loss function (所以用得少)

7. 什么是 Gradient Descent with Momentum

- (1) use the momentum of the gradient for parameter optimization
- (3) movement = negative of gradient + momentum

8. GD with Momentum 是怎么 update parameter 的

- (1) update the parameters θ in the direction of the weighted average of the past gradients

之前的 GD: $\theta^{\text{new}} = \theta^0 - \alpha \nabla L(\theta^0)$

也就是: $\theta^t = \theta^{t-1} - \alpha \nabla L(\theta^{t-1})$

现在的: $\theta^t = \theta^{t-1} - \alpha \nabla L(\theta^{t-1}) - \beta V^{t-1}$

- (2) V 的那项就是 momentum

β 是个系数, 通常是 0.9

9. momentum 有什么作用

- (1) accumulate the gradient from the past several steps

10. 什么是 Adam

(1) Adaptive Moment Estimation

(2) compute weighted average of past gradients (first moment of the gradient)

$$V^t = \beta_1 V^{t-1} + (1 - \beta_1) \nabla \mathcal{L}(\theta^{t-1})$$

compute weighted average of past squared gradients (second moment of the gradient)

$$U^t = \beta_2 U^{t-1} + (1 - \beta_2) (\nabla \mathcal{L}(\theta^{t-1}))^2$$

(3) parameter update:

$$\theta^t = \theta^{t-1} - \alpha \frac{\hat{V}^t}{\sqrt{\hat{U}^t} + \epsilon}$$

其中:

$$\hat{V}^t = \frac{V^t}{1 - \beta_1} \text{ and } \hat{U}^t = \frac{U^t}{1 - \beta_2}$$

(4) $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$

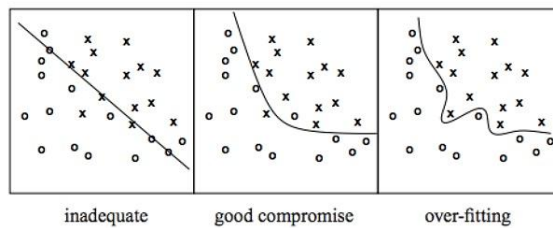
八、Overfitting

1. 什么是 Generalization

underfitting 或 overfitting 都会让通用性很差

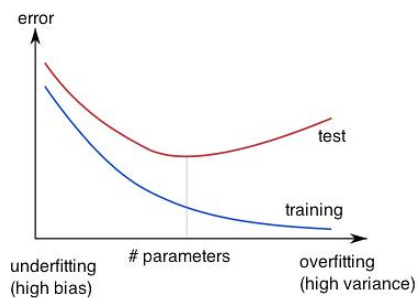
2. 什么是 Overfitting

- (1) a model fits the noise in the data instead of the underlying relationship
- (2) fit the training data very well
- (3) fail to generalize to validation data



3. underfitting 和 overfitting 有什么区别?

	underfitting	overfitting 过拟合
characteristics	too simple to represent all the relevant class characteristics	too complex and fit irrelevant characteristics (noise) in data
parameters	参数太少	参数太多
training set 的错误率	high	low
validation set 的错误率	high	high



4. 怎么解决过拟合?

- (1) Regularization
- (2) Dropout
- (3) Early Stopping
- (4) 题外话, CNN 里面的 pooling 也可以防止过拟合

5. 什么情况下会出现 overfit

- (1) decision tree
- (2) 太多 neurons

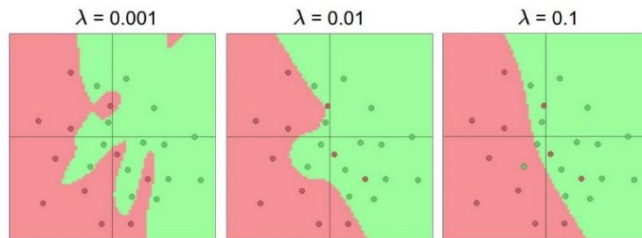
6. 什么是 Regularization

- (1) a regularization term that penalizes large weights is added to the loss function

7. 什么是 Weight Decay coefficient λ

- (1) determine how dominant the regularization is during the gradient computation

λ 大一点的效果好



8. 什么是 Dropout

- (1) randomly drop some units and their connections

- (2) dropout rate p

通常是 20%-50% 的概率 drop 一些 unit, 防止过于依赖一些特征

(3) 通常在每次迭代中, 对每个神经元都独立地进行是否丢弃的决定, 每次的训练过程中的网络结构都有所不同。

9. 什么是 Early Stopping

- (1) stop when validation accuracy has not improved after n epochs

- (2) 一边训练一边检测 validation set 的正确率

- (3) n 叫做 patience (没耐心之后就停止了)

- (4) 通常来说 $\frac{\text{validation}}{\text{train}} = 25\% - 75\%$

九、NN architectures

1. NN 的 architecture 都有哪些

- (1) Batch normalization layer
- (2) CNN
- (3) ResNet
- (4) RNN
- (5) LSTM

2. 什么是 Batch Normalization Layer

- (1) 和 data preprocessing 很像
- (2) calculate the mean μ and variance σ of input data

normalize the data to a zero mean and unit variance $\hat{x} = \frac{x-\mu}{\sigma}$

3. Batch Normalization 有什么优点

- (1) alleviate the problems of proper parameter initialization
- (2) faster convergence training
- (3) larger learning rate
- (4) reduce the internal covariate shift

4. 什么是 CNN (CV 提到)

- (1) Convolutional Neural Networks 卷积神经网络
- (2) 主要用于 image data
 - robust to spatial translations of objects in images
- (3) use a convolutional operator to extract data features

5. CNN 有什么优点

- (1) parameter sharing
- (2) efficient to train
- (3) less parameters than NNs with fully-connected layers

6. CNN 的过程

a convolutional filter will slide/convolve across the image

7. 什么是 pooling

卷积之后会有 max pooling 或者 average pooling 把一个区域取最大值/取平均

8. pooling 的优点是什么?

- (1) reduce the spatial size of feature map
- (2) reduce parameters
- (3) prevent overfitting

9. 这些层的位置

【convolutional layers】【max pooling layer】【fully-connected layers】【BatchNorm layers】
【activation layers (softmax) 】

10. CNN 一般是什么作用?

feature extraction

11. fully-connected layers 和 softmax 一般是什么作用?

classification

2. 深度学习的过程

Input → Feature extraction + Classification → Output

12. 什么是 ResNet (NLP 提到)

- (1) Residual CNN
- (2) introduce identity skip connections
- (3) inputs are propagated in each layer and then added to the output
- (4) mitigate vanishing gradient problem
- (5) 其他最先进 NN 的 base model
- (6) 可以训练非常深的神经网络 (1000 多层)

13. 什么是 RNN (NLP 提到)

- (1) Recurrent Neural Networks
- (2) trained by backpropagation-through-time 时间反向传播
- (3) model **sequential data** + data with varying length of inputs and outputs
例如: video、text、speech、DNA sequences、human skeletal data
- (4) introduce recurrent connections between neurons
- (5) process sequential data one at a time by selectively passing information across a sequence
- (6) capture correlations in sequential data
- (7) 前面的输入会被存在 model's internal state 并且影响 the model predictions
- (8) 比 CNN 更容易出现 vanishing gradient problem
(随着序列长度的增加, 梯度可能会变得非常小, 导致训练过程变得非常困难)

14. 什么是 LSTM (NLP 提到)

- (1) Long Short-Term Memory networks
- (2) a variant of RNN
- (3) can learn long-term correlations within data sequences
- (4) mitigates the vanishing/exploding gradient problem
(有一个 memory cell, update at each step in the sequence)

15. LSTM cell 的组成

- (1) Memory cell update at each step in the sequence
- (2) Input Gate protect the current step from irrelevant inputs
- (3) Output Gate prevent current step from irrelevant information to later steps
- (4) Forget Gate limit information passed from on cell to the next

Practical AI Applications

1. 什么是 scikit-learn

simple and efficient tools for predictive data analysis

accessible to everybody

reusable in various contexts

built on Numpy, SciPy, matplotlib

open source, commercially usable

2. scikit-learn 有哪些工具?

Classification

Regression

Clustering

Dimensionality reduction

Model selection

Pre-Processing

3. regression 的例子?

random forest

4. dimensionality reduction 的例子?

principal component analysis (PCA)

5. DL 有什么 framework?

Caffe | Caffe2 | torch | PyTorch | TensorFlow | Chainer | Keras

6. 什么是 Keras

high-level neural networks API

使用 TensorFlow as the compute backend

included in TensorFlow 2 as tf.keras

7. 什么是 tensor

a specialized data structure in DL

tensor = multidimensional array

比 matrix 的维度再高一阶

8. 什么是 TensorFlow

an open-source DL library

easy to learn and use

9. 什么是 PyTorch

open-source ML library

GPU-based tensor library, leverages the power of GPU
 automatic computation of gradients
 easy to test and develop new ideas
 concise, close to Python conventions
 implemented algorithms and components
 an efficient library for dynamic neural networks
 developed by Facebook

10. TensorFlow 和 PyTorch 对比

TensorFlow	PyTorch
C++写的, fast and efficient	Python 写的, 内核是 C++, more accessible
rich feature, used for training data	flexible, data size can be changed while training
strong documentation	popular at research level
the most used ML library	growing rapidly
many API available	many libraries available
support JavaScript, C++, Java, Python	support Python
can be performed on mobile devices	

11. TensorFlow 的 training flow?

data ingestion and transformation
 model building
 training
 saving

12. TensorFlow 的 programming model?

express numeric computation as a graph
 nodes: operations which have inputs and outputs
 edges: tensors

13. 什么是 graph

- (1) the computation process can be viewed as a graph

14. 什么是 static graph

- (1) define the entire computation graph before performing the computation
- (2) Data is calculated according to the defined calculation graph
- (3) no intermediate results
- (4) hard to debug
- (5) use specific syntax as control method
- (6) more optimization strategies --> better performance
- (7) low memory usage
- (8) direct deployment

15. 什么是 dynamic graph

- (1) generate computational graphs as they are computed
- (2) the complete graph is known when the computation is completed
- (3) get intermediate results
- (4) easy to debug
- (5) use front-end language syntax as control method
- (6) limited optimization --> poor performance
- (7) large memory usage
- (8) no direct deployment

16. 什么是 variable

stateful nodes which output their value

17. 什么是 placeholder

nodes whose value is fed in at execution time

18. operations 包括什么

MatMul

Add

ReLU

Computer Vision

一、Basics

1. CV 是什么?

analyse pictures and videos in order to achieve results similar to those as by humans

2. CV 的目标是什么

make computers understand images and video

3. CV 的大体过程是怎样的

Image acquisition

Image interpretation

4. 建立 CV 模型的 fundamental steps?

- | | |
|-----------------------------------|---|
| (1) data collection | capture an image |
| (2) data cleaning | noise reduction |
| | filter the data and remove unclear pictures |
| (3) data preparation | resize the pictures to common size |
| (4) build and train the model | coding, identify relevant characteristics,
choose important features |
| (5) classification or recognition | make sense of the visual information |

5. CV 的 techniques

classification

semantic segmentation

object detection

instance segmentation

6. CV 的基础操作有哪些?

Convolution

De-convolution

Dilated Convolution

Pool

Flatten

Normalization

Feature extraction

7. CV 经常用的 network?

AlexNet

VGGNET

ResNet

8. AlexNet 的创新之处

ReLU

Normalization

Dropout regulation

max pooling

9. VGGNET 创新之处

smaller filter 3*3

multiple filters within each layer

10. ResNet 创新之处

fewer filters --> reduce computational complexity

11. CV 有什么应用?

safety

health

security

content creation

AR VR

search engines

12. dimension 是什么

the number of pixels across the image's height and width 就是长乘宽

13. pixel value 是什么?

the intensity of the pixels

14. 图片的几种类型

pixel value 是 binary 的 01 串

pixel value 是 gray scale 的 一堆小数

pixel value 是 color 的 RGB 三张图片合起来

pixel value 是 multimodal 的

HDR images

Multispectral and Hyperspectral Images RGB 的三色 channel 加上几个 infrared channel

label images

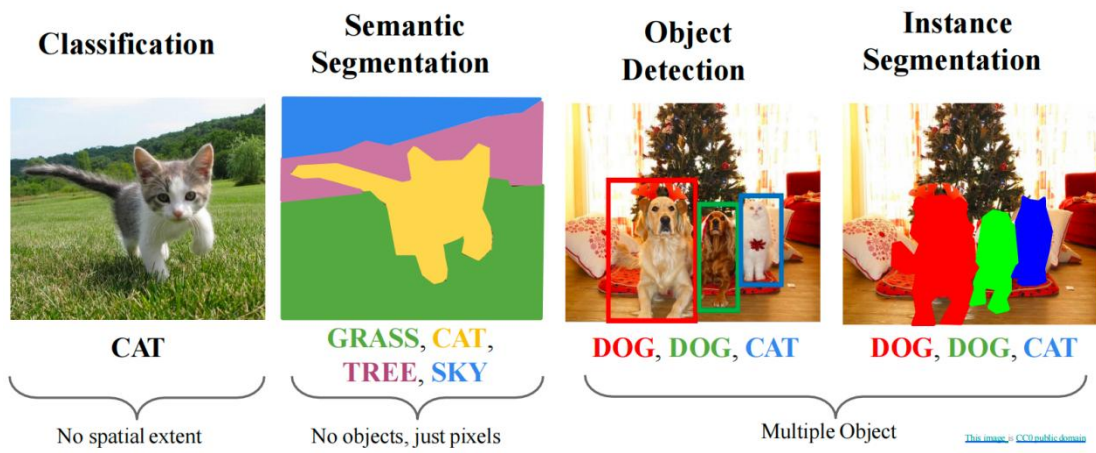
15. 彩色图片是怎么存储的?

RGB 三种颜色的图片, 电脑读这仨然后合起来

16. HDR 是什么?

high dynamic range

attempt to capture the whole tonal range of real-world scenes



二、Basic Operations & Concepts

1. 使用 fully connected network 会有什么问题

overfitting too many parameters in the weight matrix

2. CNN 的好处

local correlation 旋转拉伸也不怕

parameter sharing 都用的一个 kernel, 里面的数值都一样

reduce connectivity to local regions

3. Convolution 的基础性质

kernel 也叫 receptive field, 就是那个小框

stride pixels moved by the convolution kernel at each step

padding fill the boundary

channel 也叫做 layer, 一共卷积多少次

4. 什么是 padding

(1) 主要是做题的时候会看到, 分为两种, 一个是 valid padding, 一个是 same padding

(2) valid padding 就是 no padding, 不会添加小框框

(3) same padding 会添加小框框来使得卷积后的输出尺寸与输入尺寸相同

(4) 所以做题的时候看到 valid padding 不用管他

5. 什么是 deconvolution/ transposed convolution

a transposition computation after converting a convolution kernel to a sparse matrix

6. 什么是 dilated convolution

expand the receptive field

L is the expansion rate

insert L-1 spaces between elements inside the convolutional kernel to form inflated convolution

7. 什么是 flatten

transform 2D feature maps into a 1D vector

8. normalization 的好处

improve training stability 稳定

accelerate convergence 快速

achieve better generalization 通用

分类: layer, batch, instance, group

9. 什么是 feature

a scalar describing the property of the object

10. 什么是 good features

objects from the same class have similar feature values

objects from different class have different values

11. 什么是 feature extraction

extract features good for classification

12. 先 feature extraction 然后 classification

13. 什么是 classification

a math function or algorithm to assign a feature to a class

14. 什么是 class

a set of patterns that share common properties

15. 什么是 pattern

N-dimensional feature vector

三、Image Classification

1. 什么是 image classification?

use pixels to determine the category of image

2. 为什么需要很多 layers

a hierarchical architecture is more efficient because intermediate computations can be re-used

例子: DL architectures are efficient because they use distributed representations

3. 结构的几个特点

distributed representations

feature sharing

compositionality

4. Performance Metrics

True Positive 真阳性, 病人, 对了

False Positive 假阳性, 没病, 再测一次就好了

False Negative 假阴性, 病人, 这个问题很大, 会放跑很多病人, 漏诊很多人

True Negative 真阴性, 没病, 对了

relevant elements: 生病了的人, TP+FN

selected elements: 测出来的阳性: TP+FP

Precision: $\frac{TP}{TP+FP}$ 就是测出来的里有多少确实是病人, 反映精确度

Recall: $\frac{TP}{TP+FN}$ 就是有多少病人测了出来, 反映召回率

Accuracy: $\frac{TP+TN}{TP+FP+TN+FN}$ 测对了的一共有多少, 反映准确率

5. confusion matrix 是什么

check where the model is incorrect

reflects which classes are correlated (for multi-class classification)

6. thresholding 是什么

in a binary classification, choose the probability of belonging to a class

四、Semantics Segmentation

1. 什么是 image segmentation

do stereoscopic processing

detect motion

recognize objects

break an image into groups, based on similarities of the pixels

2.

4-connected region 是可以横着竖着把所有像素连起来

8-connected region 是需要靠斜着走才能都连起来

3. 什么是 paired training data

label each pixel in the image with a category label

4. 怎么 classify each pixel of a new image?

1 sliding window 缺点 for some patch, impossible to classify without context

2 convolution

缺点 reduce feature spatial sizes to go deeper, but segmentation requires the output size to be the same as input size

3 fully convolution

5. 讲讲 fully convolution

两种思路

1 use network with only convolutional layers make predictions for pixels all at once

2 use convolutional layers with downsampling and upsampling

第一种很贵

6. downsampling 的例子?

pooling

strided convolution

7. upsampling 的例子?

unpooling

strided transposed convolution

五、Object Recognition

1. 什么是 object recognition

find and identify objects in an image or video

2. 在 CV 中的应用?

autonomous navigation

recognize obstacles

augmented reality

overlay digital information on real-world objects

robotics

identify objects for manipulation

3. object recognition 包括什么

detection 检测到物体

description 描述一下

classification 分个类

identification 识别是什么

understanding 理解一下

4. single object 和 multiple objects 的区别

single: needs classification+localization

treat localization as a regression problem

multiple: each image needs a different number of outputs, 一个 object 一组坐标

5. localization 是什么

包含 x,y,w,h 四个空间信息

x,y: 方框左上角的坐标

h: 方框高度

w: 方框宽度

6. 什么是 selective search

find image regions that are likely to contain objects

fast to run, give 2000 region proposals in a few seconds

7. 什么是 R-CNN

Region-CNN

从 region proposal 中得到 regions of interest

变形, 改一下大小

forward each region through convolutional network

classify regions with SVMs

Natural Language Processing

一、Introduction

1. 什么是 NLP?

a field in AI and linguistics

a study of interaction between computers and human languages

empower computers to understand

interpret

generate human language

facilitate communication between human and computers

NLP= CS+AI+Computational Linguistics

2. NLP 为什么很重要?

automate tasks

understand data

multilingual communication

personalized content creation

3. NLP 有什么应用?

sentiment analysis

machine translation

speech recognition

chatbots

4. NLP 用什么模型?

Hidden Markov's Chain Model

5. NLP 的流程

Tokenization break text into sentences and words, lemmatize

Morphology part of speech tagging, stemming, NER

Syntax constituency/dependency parsing

Semantics coreference resolution, wordsense disambiguation

Discourse task-dependent

6. NLP 的 approach 有什么?

training a named entity recognition tagger (NER)

feature representation: bag of words

7. NLP 的挑战

often relies on language banks

text preprocessing

tokenization
word frequency
stemming&lemmatization
POS/NER tagging
Parsing

8. 为什么 NLP 难?

ambiguity
non-standard language
more complex languages

9. NL 的几个问题

ambiguity a single utterance can have multiple readings
anaphora use pronouns to refer to entities already introduced (代词)
indexicality utterance situation (时间地点)
vagueness
discourse structure
metonymy use a noun to stand for another (read Shakespeare)
metaphor non-literal usage of words and phrases (the process won't die)
noncompositionality (baby shoes, basketball shoes)

10. 什么是 ambiguity

(1) a single utterance can have multiple readings

(2) 分类

lexical	I saw her duck.	move head downwards to avoid being hit/ animal
syntactic	The chicken is ready to eat.	can be eaten/ will eat food
semantic		
referential		

11. HMM 是怎么工作的

the sentence is tagged as noun, verb and so on
calculate the probability of this sequence of tags
probability: transition probability+emission probability

12. 怎么计算 transition probability

<S>开始, <E>结束, 中间放上词性
数一下每个词性跟在每个词性后面的次数
然后算出 xx 词性后面是 xx 词性的概率

13. 怎么计算 emission probability

数一个词作为不同的词性的次数, 画一个 counting table
算每个词性中每个词出现的概率, 画一个 probability table

14. 什么是 communication

exchange of information brought by signs drawn from shared system

15. communication 的组成部分?

Intention

Generation

Synthesis

Perception

Analysis

Disambiguation

Incorporation

16. Communication go wrong 的原因

Insincerity 不真诚

Speech recognition errors 没听清

Ambiguous utterance 歧义

Different contexts 内容不同

17. 什么是 language

enable us to communicate

tied to thinking

18. 什么是 speech

communication act

19. utterances 包括什么

talking

writing

facial expression

gesture



20. image captioning 的过程?

detect words -- generate sentences -- re-rank sentences

二、DL for NLP

1. 用 DL 解决 NLP 的核心思想?

represent words as dense vectors (每个维度都有数字)

2. 用 DL 解决 NLP 的方法

Word embedding

N-Grams

RNN

LSTM

3. 什么是 text preprocessing

clean and transform raw text data into a format suitable for analysis

4. 什么是 tokenization

break down the text into the smallest unit

5. 什么是 word frequency

measures how many times that word appears in the entire corpus.

6. 什么是 stemming

find the root of words

7. 什么是 lemmatization

find the form of the related word

longer process to calculate than stemming

8. 什么是 POS/NER tagging

label the words according to their word types

9. 怎么训练一个 NER tagger

用 HMM

Input layer -- hidden layer -- NER tag/POS tag

10. 什么是 Parsing

examine whether a document makes sense by comparing its contents to the principles of formal grammar

process of finding a parse tree for a given input string

11. 什么是 parser?

report errors with the syntax

assist the generation of a parse tree

12. 自然语言是 context dependent 的

13. vector space 的好处?

enable the computation of similarity between words (cosine similarity)

14. 怎么构造 vector space?

用 word embedding

15. 什么是 word embedding?

turn words into numbers to use by many ML algorithms

16. embedding 的方法?

Word2Vec

17. Word2Vec 的两种训练算法?

CBoW (continuous bag of words) predict center word from context (完形填空)

Skip-gram predict context from center word (扩句)

18. CBoW 和 Skip-gram 分别的优势?

CBoW faster to train+大数据 work well

Skip-gram suitable for capturing **semantic relationships** between words+小数据 well

19. 什么是 learned vectors

semantically close words are near each other (dog 和 cat 挨着)

syntactic relationships are preserved with relative positioning (slow-slower 和 fast-faster 的连线方向差不多, 首都指向国家的方向差不多)

20. 什么是 word analog

例如: king-man+woman=queen

21. measuring similarity 的公式?

$$(\cos\theta) = (A \cdot B) / (\|A\| * \|B\|)$$

dot product 和 magnitude

22. 什么是 sliding window

W 是中心词两边的宽度, S 是每次挪多少

23. 用 sliding window 的好处?

learn features of each word on its own, given a text corpus

don't need heavy preprocessing

word vectors can be used as features for lots of supervised learning applications

24. 什么是 language modeling, 有什么应用?

assign high probabilities to well-formed sentences

应用: generation, speech recognition, machine translation

25. 什么是 N-Grams

N=1 的时候就是一个词地分析

N=2 就是分析两个相邻的词

N=3 就是三个相邻的一起分析

问题: N=5 就是极限了

26. 什么是 RNN

use past information without restricting the size of the context

缺点: can't recall information long time ago

27. 什么是 LSTM (long short term memory networks)

1 can mitigate **vanishing gradient problem**

2 can handle long term dependencies

3 have **gating mechanism** to regulate the flow of information

4 have a **memory cell** to store and retrieve information over long sequences

详细版:

1 can mitigate vanishing gradient problem

(因为 it incorporates specialized memory cells and gating mechanisms to learn and store information over long sequences)

2 can handle long term dependencies

(因为能够 maintain and update cell states over time)

3 have gating mechanism to regulate the flow of information

(allowing them to selectively update and use information from previous time steps)

(gates that control forgetting, adding, updating, outputting information)

4 have a memory cell to store and retrieve information over long sequences

三、NLP applications

1. ChatGPT 模型的原理?

compute the probability of the next token in sequence

2. ChatGPT 的核心技巧

unsupervised pre-training techniques

展开讲: help to achieve dialogue generation

automatically learn the laws and features of the language from a large amount of unlabeled data

Limitations & Future

1. 现在的 AI 技术有什么限制?

data dependency

interpretability and explainability

generalization

computation and resource requirements

energy consumption

robustness and security

2. 什么是 data dependency

AI relies on large amounts of high-quality data for training and learning.

Limited or biased data can lead to inaccurate or biased AI systems

3. 什么是 Interpretability

many AI algorithms are considered black boxes

it is hard to understand and interpret their decision-making process

4. 什么是 Explainability

explain model predictions or decision results.

5. 什么是 Domain Shift

(1) a phenomenon in ML and statistics

(2) the statistical properties of data will change if it transitions from one domain or distribution to another

(3) occur when a model that is trained on data from one source domain performs poorly when applied to a different target domain

6. 怎么解决 Domain Shift

(1) Zero-shot learning

(2) Knowledge transfer

(3) GAN

7. 什么是 Computation and resource requirements

AI algorithms require significant computing power and resources

8. AGI 是什么组成的

(1) $DL+IL+RL=AGI$

(IL 是模仿学习 imitation)

9. AGI 有什么

(1) AlphaGo

(2) ChatGPT

- (3) AlphaFold
- (4) ClimaX

10. AGI 会如何发展

- (1) from common to professional
- (2) more media will be involved
- (3) LLMs leverage tools to affect real world
- (4) automatic driving
- (5) Natural language becomes a new programming language
- (6) AI for science