The First Report

Wang Mingxi

Jilin University
College of Computer Science and Technology

(None)

Overview

Learning content
Hands on practice
Plan for the following week

Learning content

- Decision tree
- Ensemble Learning
- Neural Network

Hands on practice

- Generative Adversarial Network GAN

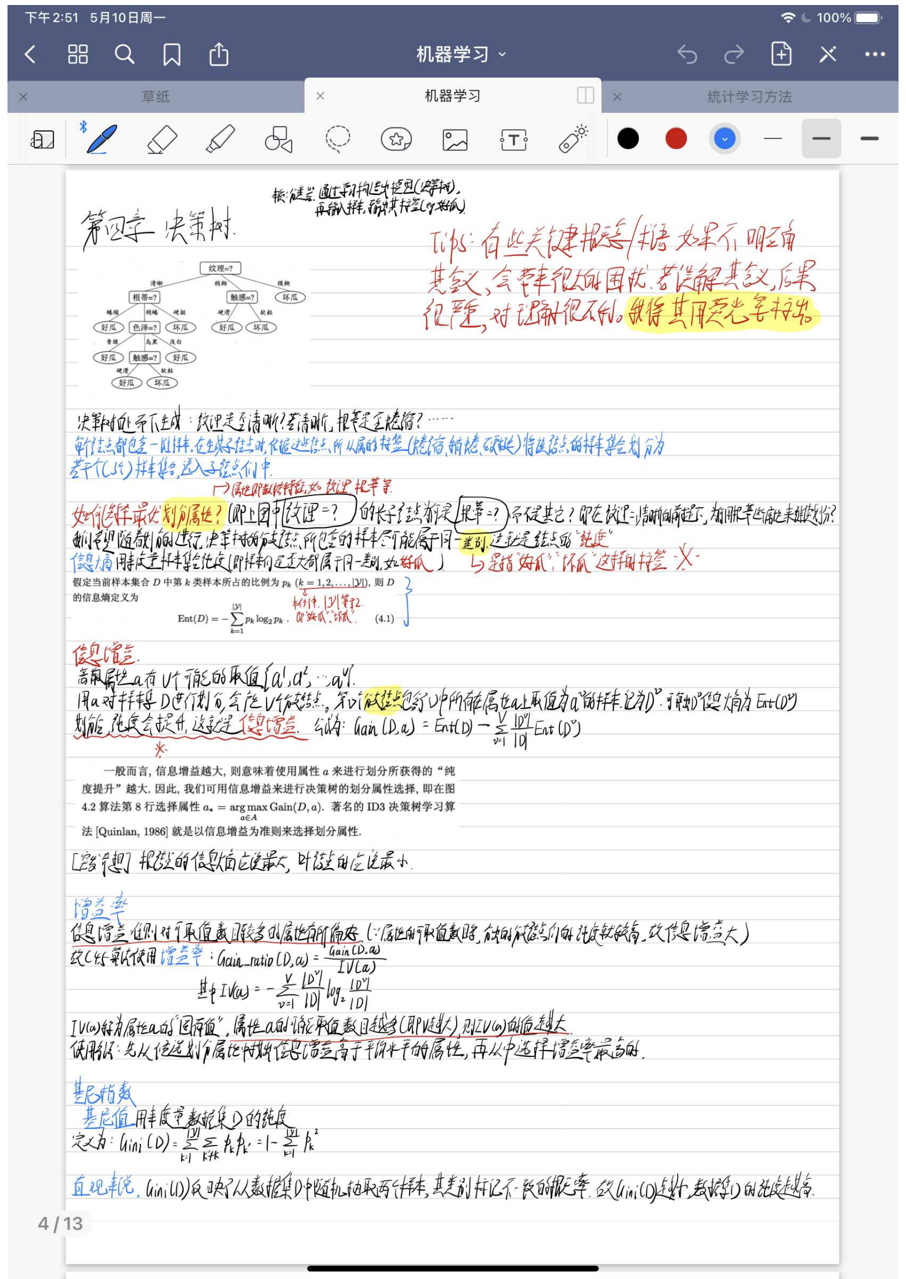
Plan for the following week

- Plan for the following week

Learning content

Decision tree

Decision tree:A decision tree is a prediction model used to predict the category of samples.In the structure of these trees, leaf nodes give categories and inner nodes represent attributes.



核心：就是通过学习构造出树（决策树），再输入样本，预测其标签（eg.好瓜）

第四章 决策树

Tips：有些关键概念/术语 如果不明白其含义，会带来很大的困扰。若误解其含义，后果很严重，对理解很不利。我将其用荧光笔标出。

决策树如何生成：纹理是否清晰？若清晰，根蒂是否蜷缩？……

每个结点都包含一组样本，在生成子结点时，依据这些结点所从属的标签（蜷缩，稍蜷，硬挺）将该结点的样本集合划分为若干个（3个）样本集合，送入子结点们中。

↱属性即数据特征，如 纹理，根蒂等

如何选择最优划分属性？（即上图中 纹理=? 的长子结点为何是 根蒂=? 而不是其它？即在纹理=清晰的前提下，为何用根蒂蜷缩来继续划分？

我们希望随着划分的进行，决策树的分支结点所包含的样本尽可能属于同一类别。这就是结点的"纯度"

信息熵 用来度量样本集合纯度（即样本们是否大都属于同一类别，如好瓜） ↳ 是指"好瓜""坏瓜"这样的标签 ×

假定当前样本集合 D 中第 k 类样本所占的比例为 p_k $(k = 1, 2, \ldots, |\mathcal{Y}|)$，则 D 的信息熵定义为

$$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k . \qquad (4.1)$$

本例中，|Y|等于2，即"好瓜""坏瓜"

信息增益

离散属性 a 有 V 个可能的取值 $\{a^1, a^2, \ldots, a^V\}$.

用 a 对样本集 D 进行划分，会产生 V 个分支结点。第 v 个分支结点包含了 D 中所有在属性 a 上取值为 a^v 的样本，记为 D^v，可算出 D^v 的信息熵为 $\mathrm{Ent}(D^v)$

划分后，纯度会提升，这就是信息增益。公式：$\mathrm{Gain}(D,a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}(D^v)$

一般而言，信息增益越大，则意味着使用属性 a 来进行划分所获得的"纯度提升"越大. 因此，我们可用信息增益来进行决策树的划分属性选择，即在图 4.2 算法第 8 行选择属性 $a_* = \underset{a \in A}{\arg\max}\, \mathrm{Gain}(D, a)$. 著名的 ID3 决策树学习算法 [Quinlan, 1986] 就是以信息增益为准则来选择划分属性.

[我的猜想] 根结点的信息熵应该最大，叶结点的应该最小.

增益率

信息增益准则对可取值数目较多的属性有所偏好（∵属性的可取值数目多，分出的分支结点们的纯度就较高，故信息增益大）

故C4.5算法使用增益率：$\mathrm{Gain\_ratio}(D,a) = \frac{\mathrm{Gain}(D,a)}{IV(a)}$

其中 $IV(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} \log_2 \frac{|D^v|}{|D|}$

$IV(a)$ 称为属性 a 的"固有值"，属性 a 的可能取值数目越多（即 V 越大），则 $IV(a)$ 的值通常越大.

使用方法：先从候选划分属性中找出信息增益高于平均水平的属性，再从中选择增益率最高的.

基尼指数

基尼值 用来度量数据集D的纯度

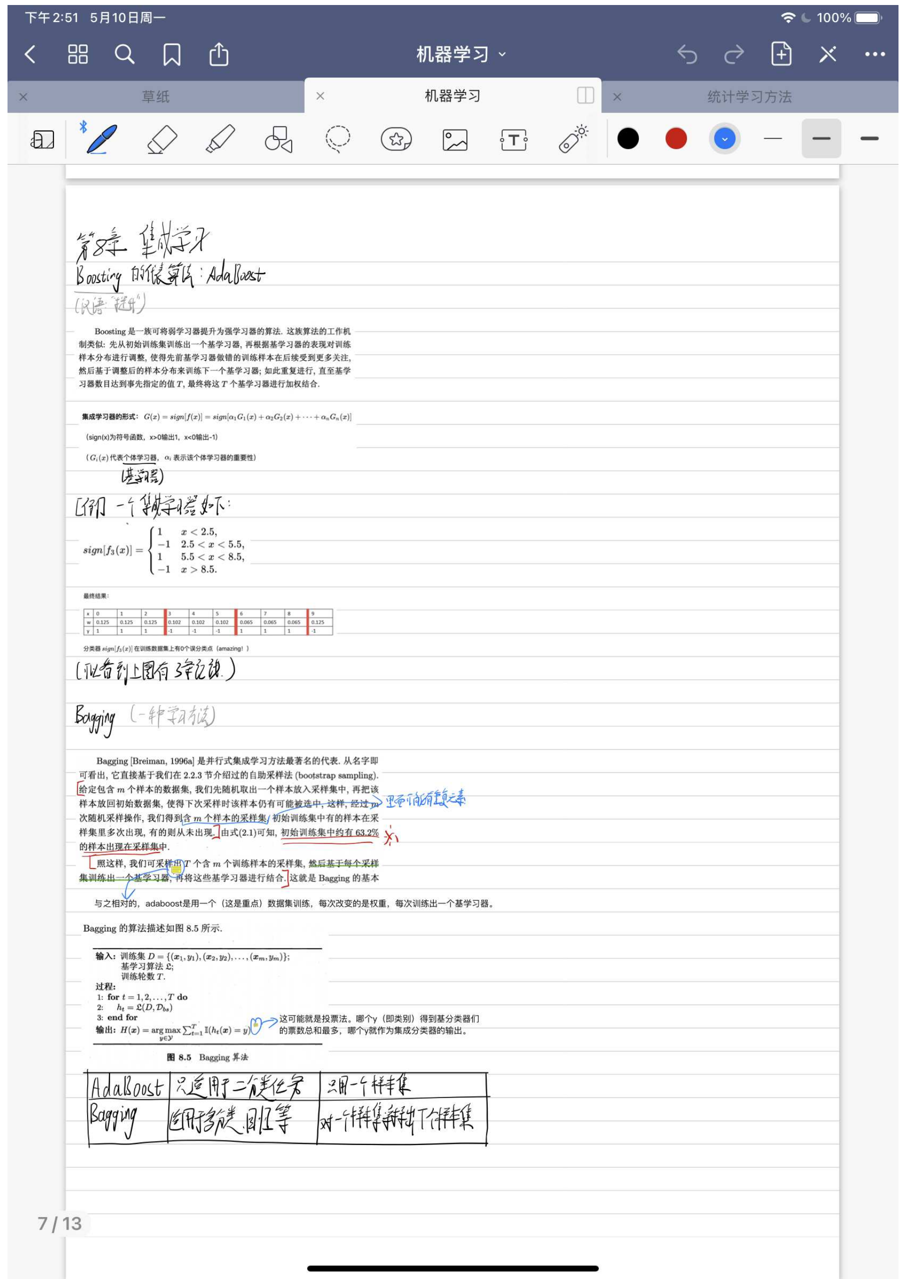
定义为：$\mathrm{Gini}(D) = \sum_{k=1}^{|\mathcal{Y}|} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|\mathcal{Y}|} p_k^2$

直观来说，$\mathrm{Gini}(D)$ 反映了从数据集D中随机抽取两个样本，其类别标记不一致的概率。故 $\mathrm{Gini}(D)$ 越小，数据集D的纯度越高.

4 / 13

Ensemble Learning

Ensemble Learning:A decision tree is a prediction model used to predict the category of samples.In the structure of these trees, leaf nodes give categories and inner nodes represent attributes.



第8章 集成学习

Boosting 的代表算法：AdaBoost

(汉语 "提升")

Boosting 是一族可将弱学习器提升为强学习器的算法. 这族算法的工作机制类似: 先从初始训练集训练出一个基学习器, 再根据基学习器的表现对训练样本分布进行调整, 使得先前基学习器做错的训练样本在后续受到更多关注, 然后基于调整后的样本分布来训练下一个基学习器; 如此重复进行, 直至基学习器数目达到事先指定的值 T, 最终将这 T 个基学习器进行加权结合.

集成学习器的形式： $G(x) = sign[f(x)] = sign[\alpha_1 G_1(x) + \alpha_2 G_2(x) + \cdots + \alpha_n G_n(x)]$

(sign(x)为符号函数，x>0输出1，x<0输出-1)

（$G_i(x)$ 代表个体学习器，α_i 表示该个体学习器的重要性）

(基学习器)

[例] 一个集成学习器如下:

$$sign[f_3(x)] = \begin{cases} 1 & x < 2.5, \\ -1 & 2.5 < x < 5.5, \\ 1 & 5.5 < x < 8.5, \\ -1 & x > 8.5. \end{cases}$$

最终结果：

x	0	1	2	3	4	5	6	7	8	9
w	0.125	0.125	0.125	0.102	0.102	0.102	0.065	0.065	0.065	0.125
y	1	1	1	-1	-1	-1	1	1	1	-1

分类器 $sign[f_3(x)]$ 在训练数据集上有0个误分类点（amazing！）

(对应到上图有3条红线.)

Bagging (一种学习方法)

Bagging [Breiman, 1996a] 是并行式集成学习方法最著名的代表. 从名字即可看出, 它直接基于我们在 2.2.3 节介绍过的自助采样法 (bootstrap sampling). 给定包含 m 个样本的数据集, 我们先随机取出一个样本放入采样集中, 再把该样本放回初始数据集, 使得下次采样时该样本仍有可能被选中, 这样, 经过 m 次随机采样操作, 我们得到含 m 个样本的采样集, 初始训练集中有的样本在采样集里多次出现, 有的则从未出现. 由式(2.1)可知, 初始训练集中约有 63.2% 的样本出现在采样集中.

里面可能有重复元素

照这样, 我们可采样出 T 个含 m 个训练样本的采样集, ~~然后基于每个采样集训练出一个基学习器~~, 再将这些基学习器进行结合. 这就是 Bagging 的基本

与之相对的，adaboost是用一个（这是重点）数据集训练，每次改变的是权重，每次训练出一个基学习器。

Bagging 的算法描述如图 8.5 所示.

```
输入: 训练集 $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$;
      基学习算法 $\mathfrak{L}$;
      训练轮数 $T$.
过程:
  for $t = 1, 2, \ldots, T$ do
      $h_t = \mathfrak{L}(D, \mathcal{D}_{bs})$
  end for
输出: $H(x) = \arg\max_{y \in \mathcal{Y}} \sum_{t=1}^{T} \mathbb{I}(h_t(x) = y)$
```

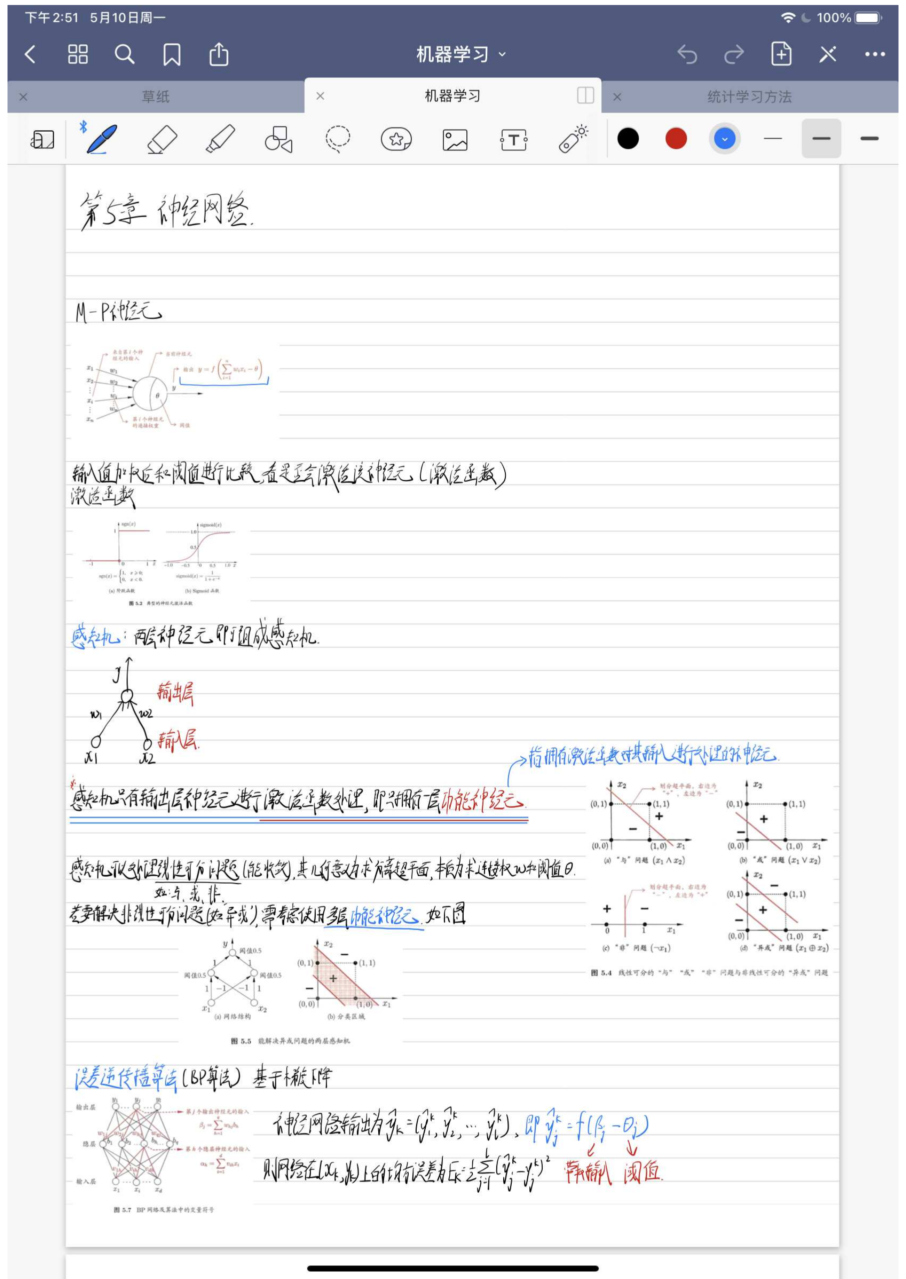
这可能就是投票法。哪个y（即类别）得到基分类器们的票数总和最多，哪个y就作为集成分类器的输出。

图 8.5 Bagging 算法

AdaBoost	只适用于二分类任务	只用一个样本集
Bagging	适用于多分类、回归等	对一个样本集采样出T个样本集

Neural Network

Neural Network:A mathematical or computational model that imitates the structure and function of a biological neural network and is used for estimating or approximating functions.


下午2:51 5月10日周一
100%
机器学习
草纸
机器学习
统计学习方法

第5章 神经网络

M-P神经元

来自第 i 个神经元的输入
当前神经元
输出 $y = f\left(\sum_{i=1}^{n} w_i x_i - \theta\right)$
第 i 个神经元的连接权重
阈值
x_1
x_2
x_i
x_n
w_1
w_2
w_i
w_n
θ
y

输入值加权和和阈值进行比较,看是否会激活该神经元(激活函数)
激活函数

sgn(x)
sigmoid(x)
$\mathrm{sgn}(x) = \begin{cases}1, & x \geqslant 0;\\ 0, & x < 0.\end{cases}$
$\mathrm{sigmoid}(x) = \frac{1}{1+e^{-x}}$
(a) 阶跃函数
(b) Sigmoid 函数

图 5.2 典型的神经元激活函数

感知机:两层神经元即可组成感知机.

y
输出层
w_1
w_2
输入层
x_1
x_2

感知机只有输出层神经元进行激活函数处理,即只拥有一层功能神经元.
→指拥有激活函数对其输入进行处理的神经元.

感知机可以处理线性可分问题(能收敛),其几何意义为求解超平面,本质为求连接权w和阈值θ.
如:与、或、非.
若要解决非线性可分问题(如异或),需考虑使用多层功能神经元. 如下图

划分超平面,右边为"+",左边为"−"
(a) "与"问题 $(x_1 \wedge x_2)$
(b) "或"问题 $(x_1 \vee x_2)$
划分超平面,右边为"−",左边为"+"
(c) "非"问题 $(\neg x_1)$
(d) "异或"问题 $(x_1 \oplus x_2)$

图 5.4 线性可分的"与""或""非"问题与非线性可分的"异或"问题

阈值0.5
阈值0.5
阈值0.5
(a) 网络结构
(b) 分类区域

图 5.5 能解决异或问题的两层感知机

误差逆传播算法(BP算法) 基于梯度下降

输出层
隐层
输入层
第 j 个输出神经元的输入 $\beta_j = \sum_{h=1}^{q} w_{hj} b_h$
第 h 个隐层神经元的输入 $\alpha_h = \sum_{i=1}^{d} v_{ih} x_i$

图 5.7 BP 网络及算法中的变量符号

神经网络输出为 $\hat{y}_k = (\hat{y}_1^k, \hat{y}_2^k, \dots, \hat{y}_l^k)$, 即 $\hat{y}_j^k = f(\beta_j - \theta_j)$

则网络在 (x_k, y_k) 上的均方误差为 $E_k = \frac{1}{2}\sum_{j=1}^{l}(\hat{y}_j^k - y_j^k)^2$ 带权输入 阈值.

Hands on practice

Generative Adversarial Network GAN

- Generating Adversarial networks (GAN) is a method of unsupervised learning in which two neural networks play each other against each other.

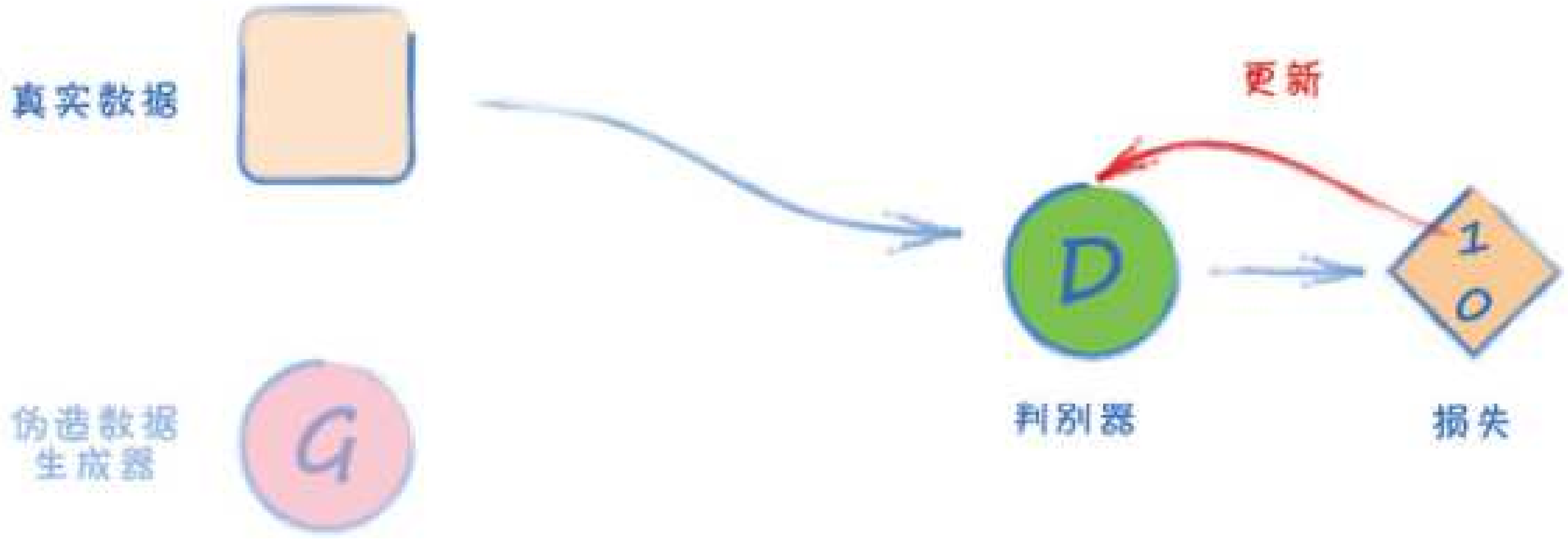


Figure 1

Generative Adversarial Network GAN

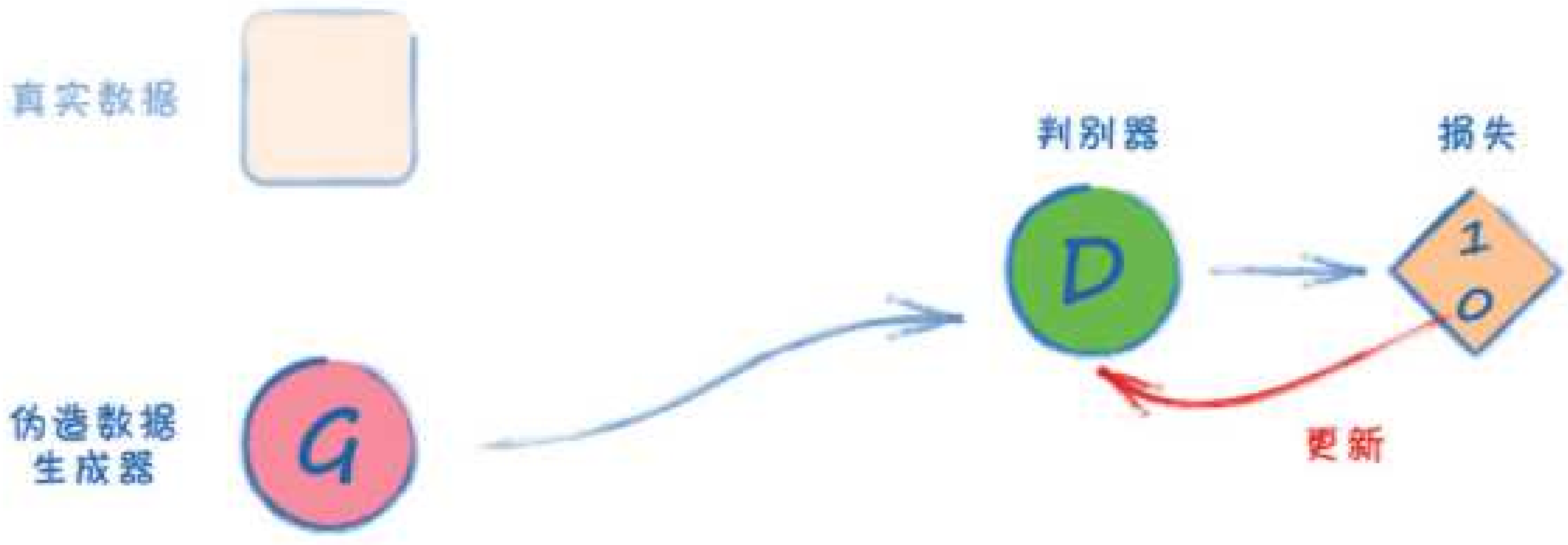


Figure 2: preview

Generative Adversarial Network GAN

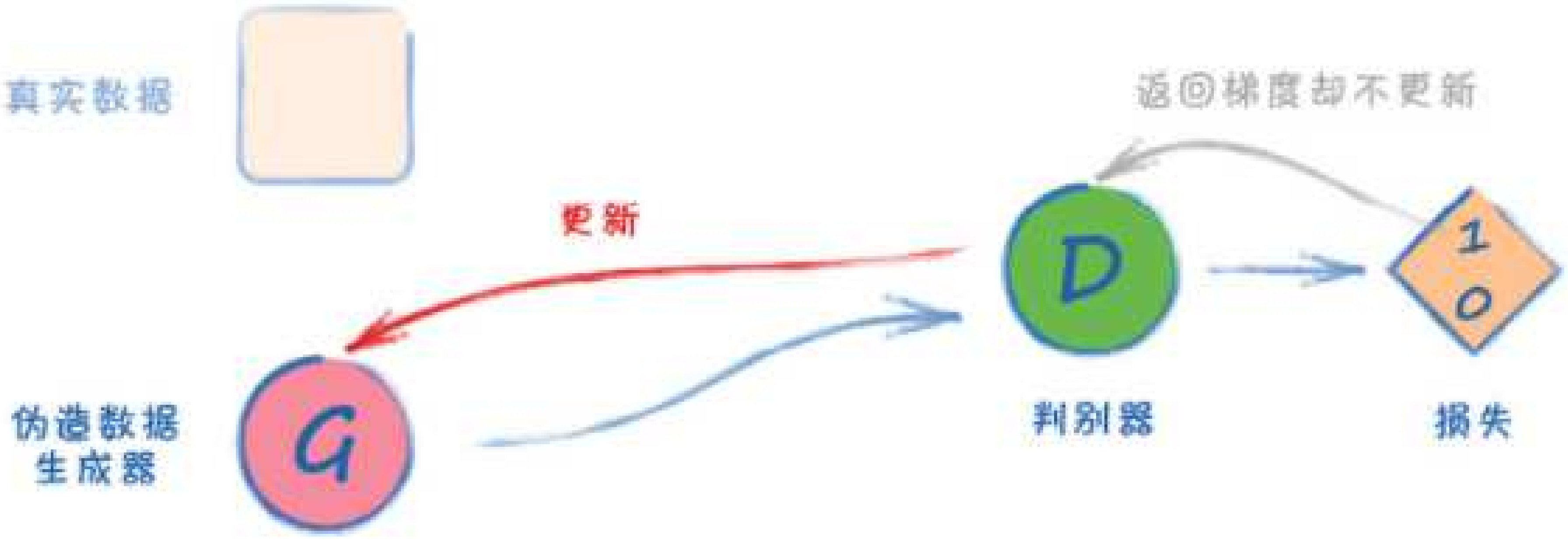


Figure 3: preview

Learning content

Hands on practice

Plan for the following week

Plan for the following week

Plan for the following week

Plan for the following week

Learning content
Hands on practice
Plan for the following week
Plan for the following week

- I’ll continue to learn the basics next week.
- I’m going to go ahead and program this stuff out.

Contact Information

Wang Mingxi
College of Computer Science and Technology
Jilin University, China

✉ mxwang@tulip.academy

🏠 Team for Universal Learning and Intelligent Processing