

Machine learning in NP search

[2026.7.8@河南师范大学](#)

机器学习

- 一些概念的介绍
- 与新物理的关系
- 无法涵盖很多非常有用的知识
- **智能体实机操作, MLAnalysis skill**

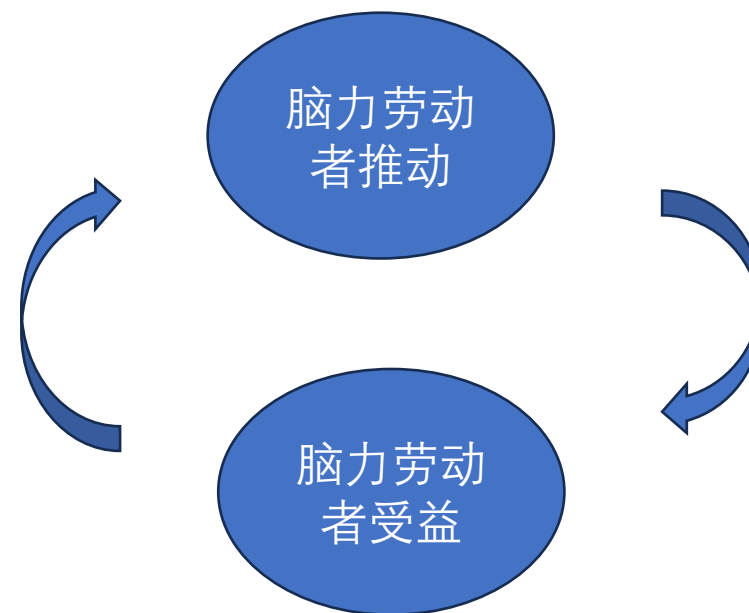
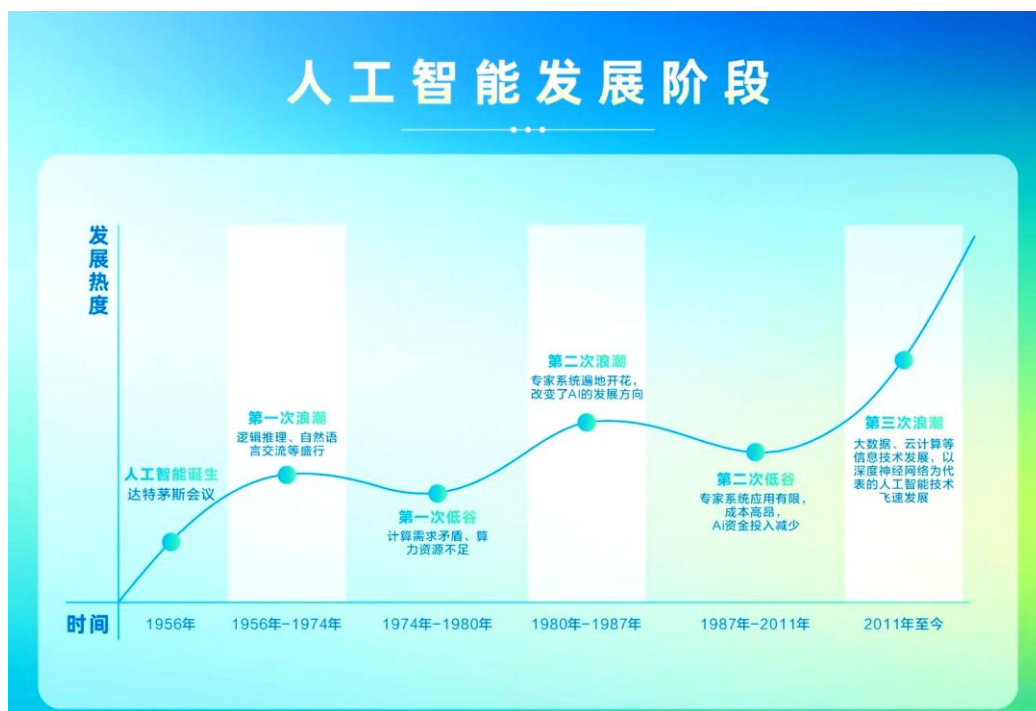
A tool-making animal



?

why人工智能?

- 人工智能必然是未来趋势
- 这个世界上一群聪明人想做成的一件事。



人工智能

- 不严谨的说， 数学=人工智能

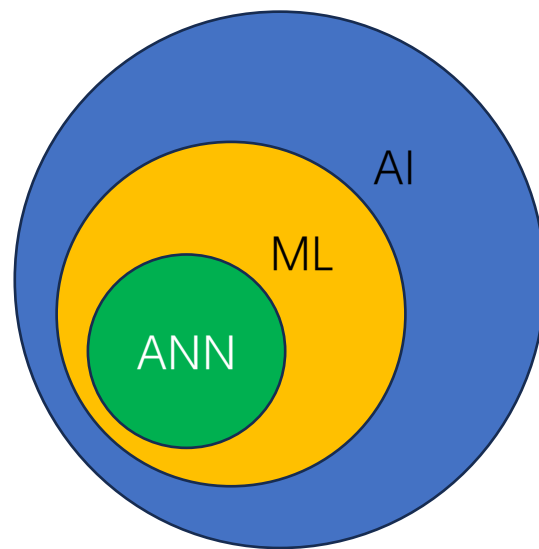
The ultimate goal of mathematics is to eliminate any need for intelligent thought.

—Alfred N. Whitehead

- Alpha go
- ChatGPT, deepseek
- Stable diffusion
- ...

机器学习

- 人工智能算法, (例如A*)
- 机器学习算法, (例如kmeans)
- 人工神经网络
- 从数据中得到规律



什么是学习?

- 计算一下 $5 \times 22 = ?$

九九乘法口诀表

1×1=1									
1×2=2	2×2=4								
1×3=3	2×3=6	3×3=9							
1×4=4	2×4=8	3×4=12	4×4=16						
1×5=5	2×5=10	3×5=15	4×5=20	5×5=25					
1×6=6	2×6=12	3×6=18	4×6=24	5×6=30	6×6=36				
1×7=7	2×7=14	3×7=21	4×7=28	5×7=35	6×7=42	7×7=49			
1×8=8	2×8=16	3×8=24	4×8=32	5×8=40	6×8=48	7×8=56	8×8=64		
1×9=9	2×9=18	3×9=27	4×9=36	5×9=45	6×9=54	7×9=63	8×9=72	9×9=81	

什么是学习？

- 先看一个小例子：
- 两列火车（A和B）相距 **200公里**，以 **50公里/小时** 的速度相向而行。一只飞虫从火车A出发，以 **75公里/小时** 的速度飞向火车B；到达火车B后立即折返飞向火车A，如此往复，直到两列火车相遇。
问：飞虫总共飞行了多少距离？

什么是学习？

- 故事的一个版本是这样的：
- 记者：你太厉害了，这么短的时间你就抓住了这个本质。
- 冯诺依曼：不，我使用了积分。

什么是学习？

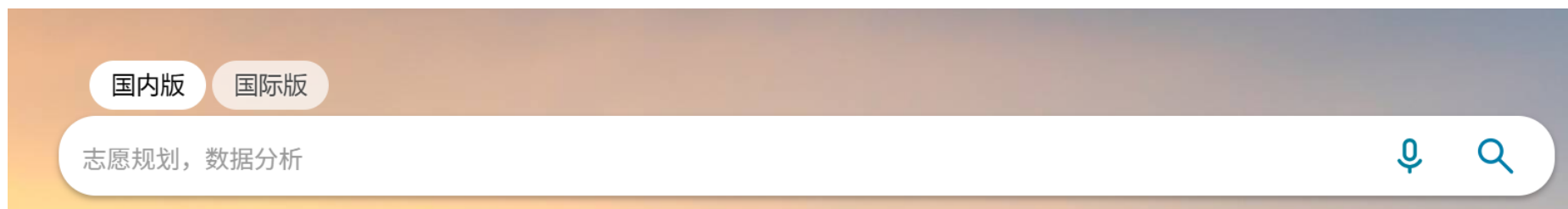
- 描述一下你看到的这个物品



穿越古代必备技能之一：制盐

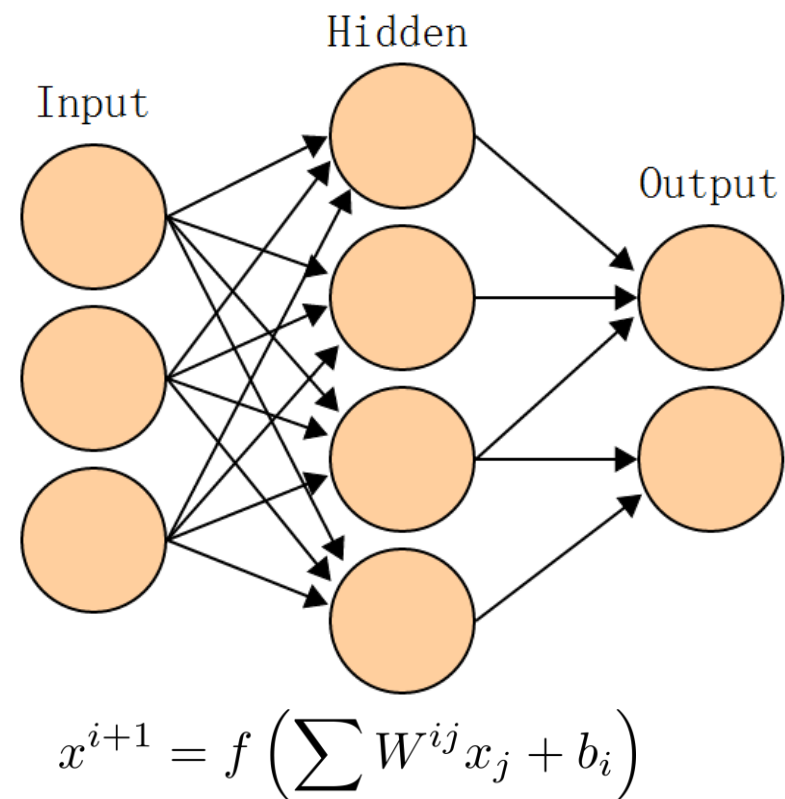
什么是学习？

- 我们处理数据的方式，取决于我们记住了什么（知识，规律…）
- 换言之：学习 = 以某种方式存储信息，以便于用来解决问题
- 从这个角度来说，机器学习有点像数据库



什么是机器学习？

- 人类大脑是怎么存储信息的？（人工神经网络就是个仿生学）



什么是机器学习?

- 从另一个角度看待机器学习: 函数 (输入->输出)



$$f(\text{猫}) = \text{“猫”}$$

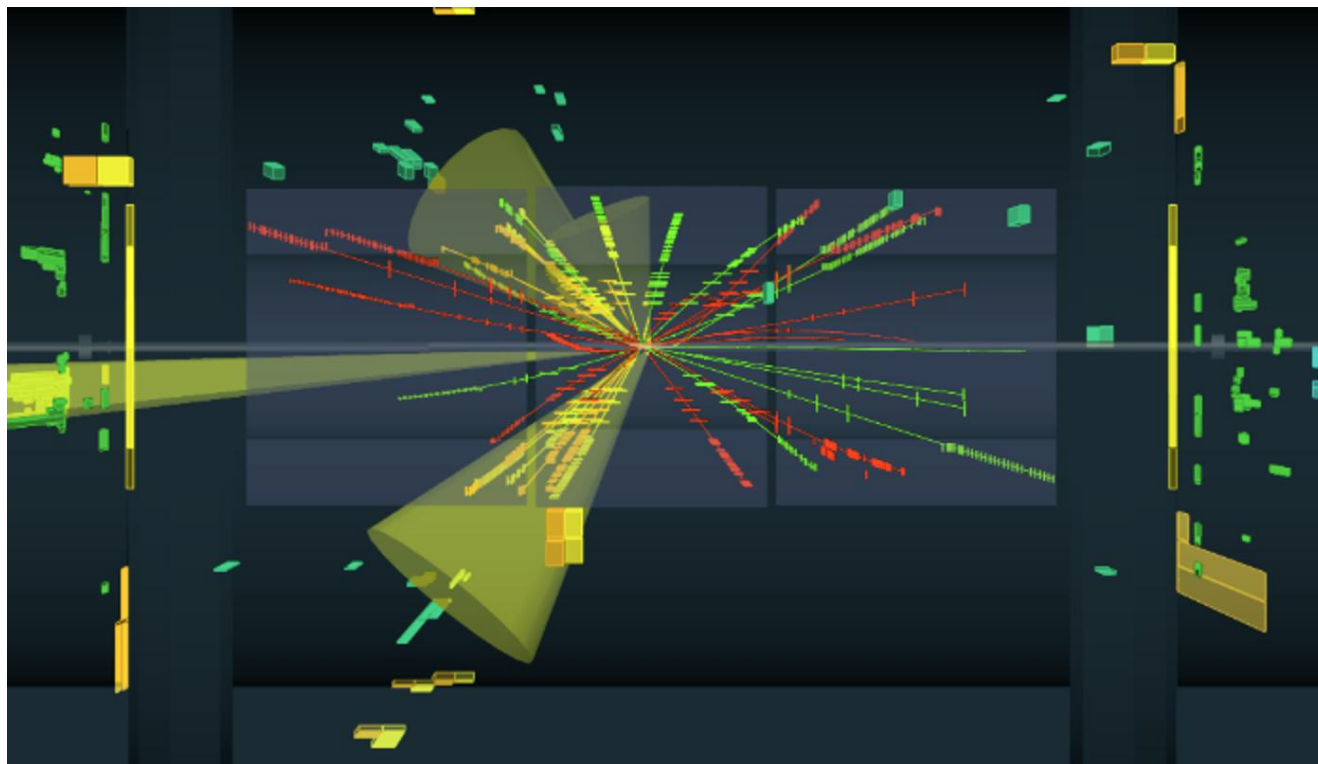
什么是机器学习?



- $f(\dots)$ = 踩油门

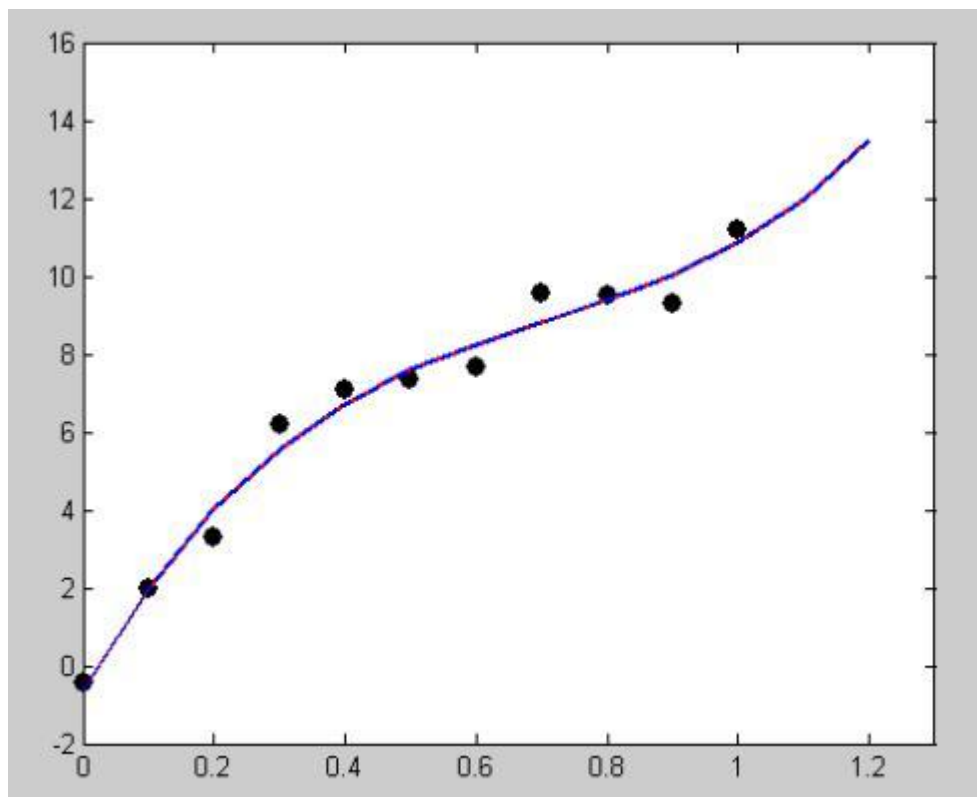
什么是机器学习？

- $f(\dots)$ =新物理信号 （假设我们能肉眼识别）



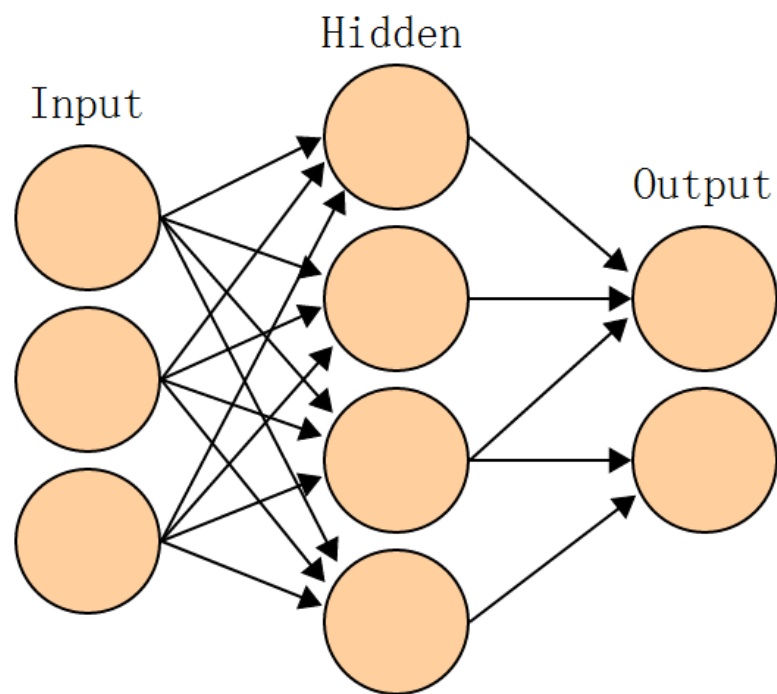
什么是机器学习？

- 学习（训练）：对函数的拟合



什么是机器学习？

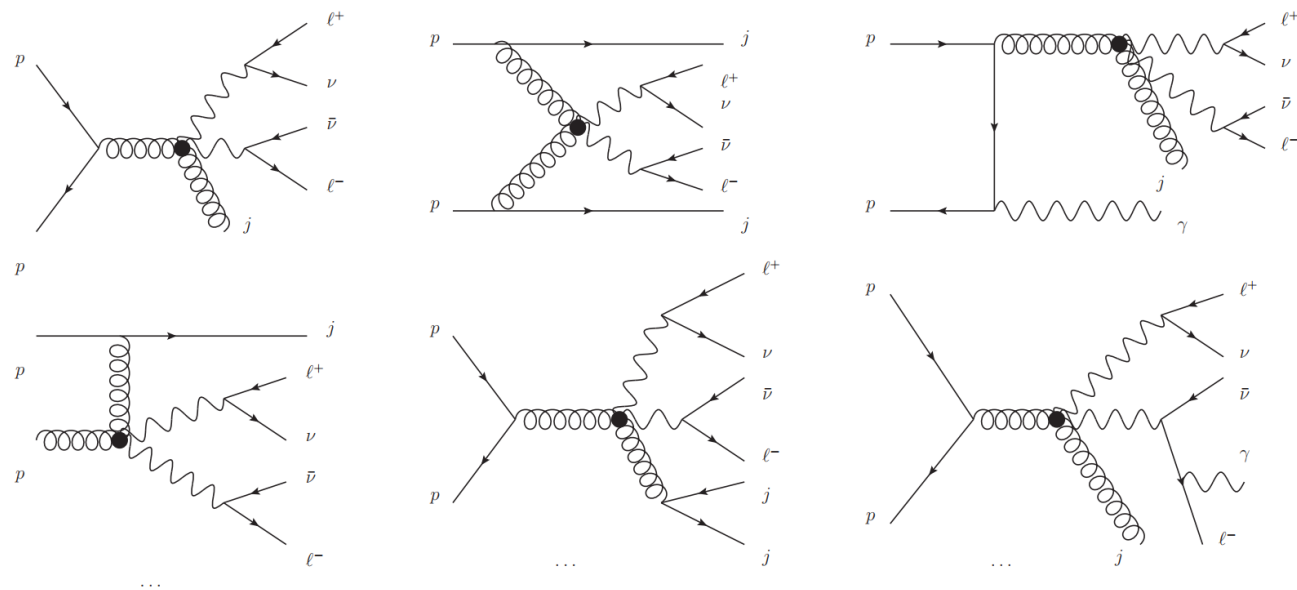
- 神经网络也是一个函数



$$x^{i+1} = f \left(\sum W^{ij} x_j + b_i \right)$$

为什么要在新物理研究中用机器学习?

- 典型的问题：搜索新物理信号，我们一般是怎么做的？

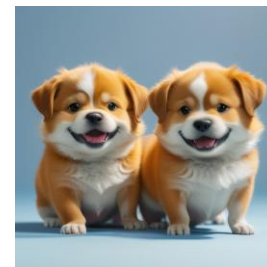


为什么要在新物理研究中用机器学习？

- 找不到新物理：增加亮度，研究更稀有的过程
- 稀有过程：末态粒子多，费曼图数量多，分析起来复杂
- 末态粒子多，可观测量也多，数据量较大

为什么要在新物理研究中用机器学习？ 在使用之前，先意识到边界

- 机器学习的使用本身是简单的（这正是发明它的意义所在）
- 真正困难的是， 找到一个非常适用于使用机器学习的场景
- 机器学习不是万能的（1万神经元 $\approx$ 果蝇幼虫）
- 不要为了使用机器学习而使用机器学习
- 不要止于“有效”（抹除实验）



机器学习 – 数据准备

- 很多机器学习算法处理的数据是一个“表格”
- 特征空间： feature space, （例如， 一个碰撞事例的可观测量）

$pp \rightarrow \gamma \nu \bar{\nu}$	E_γ	p_γ^x	p_γ^y	p_γ^z	E_T^x	E_T^y
	feature1	feature2	feature3	feature4	feature5	feature6
Event 1						
Event 2						
Event 3						
Event 4						
Event 5						

最好的情况是完全不用思考的情况。

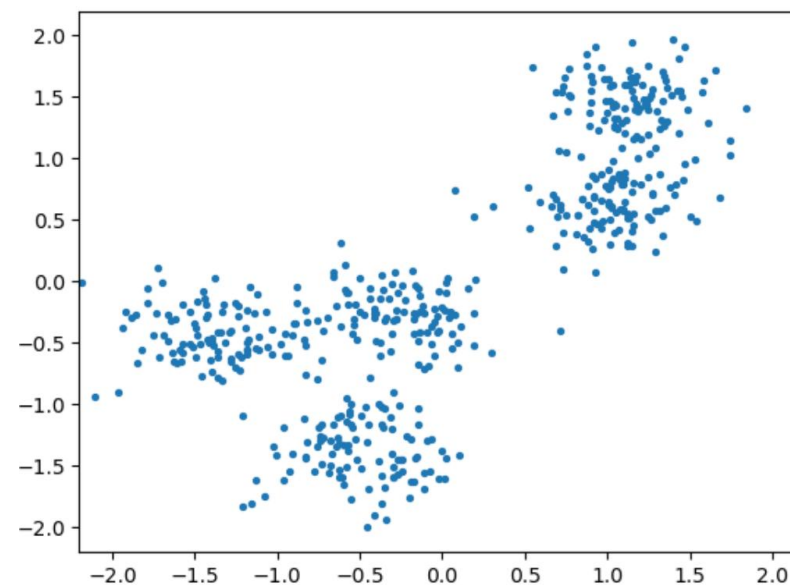
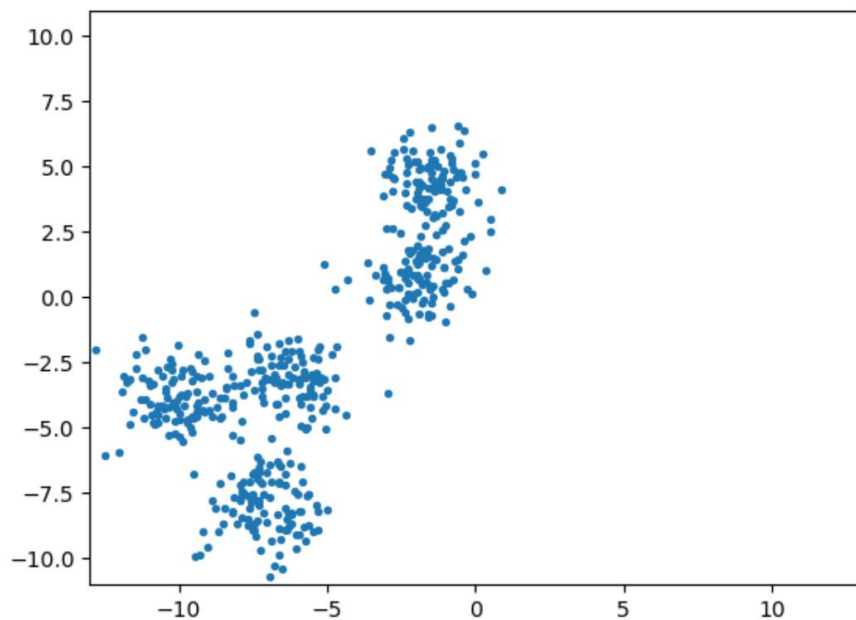
机器学习 – 数据准备

- 缺数据怎么办?
- 数据清理
- 补0
- 变成其它的样子
- Madgraph5 -> Machine Learning: MLAnalysis
- <https://gitee.com/NBAlexis/CutExperiment>
- 不用去学，后面会介绍智能体

机器学习 – 数据准备

- Z-SCORE Standardization
- 去中心化, 椭圆 -> 圆

$$x_i^j = \frac{v_i^j - \bar{v}^j}{\epsilon^j}$$



机器学习-一些简单的算法

- Kmeans
- PCA
- KNN
- Isolation forest
- SVM
- 朴素贝叶斯
- 决策树
- LOF
- ...



机器学习-一些简单的算法

- Kmeans
- PCA
- KNN
- Isolation forest
- SVM
- 朴素贝叶斯
- 决策树
- LOF
- ...

TMVA

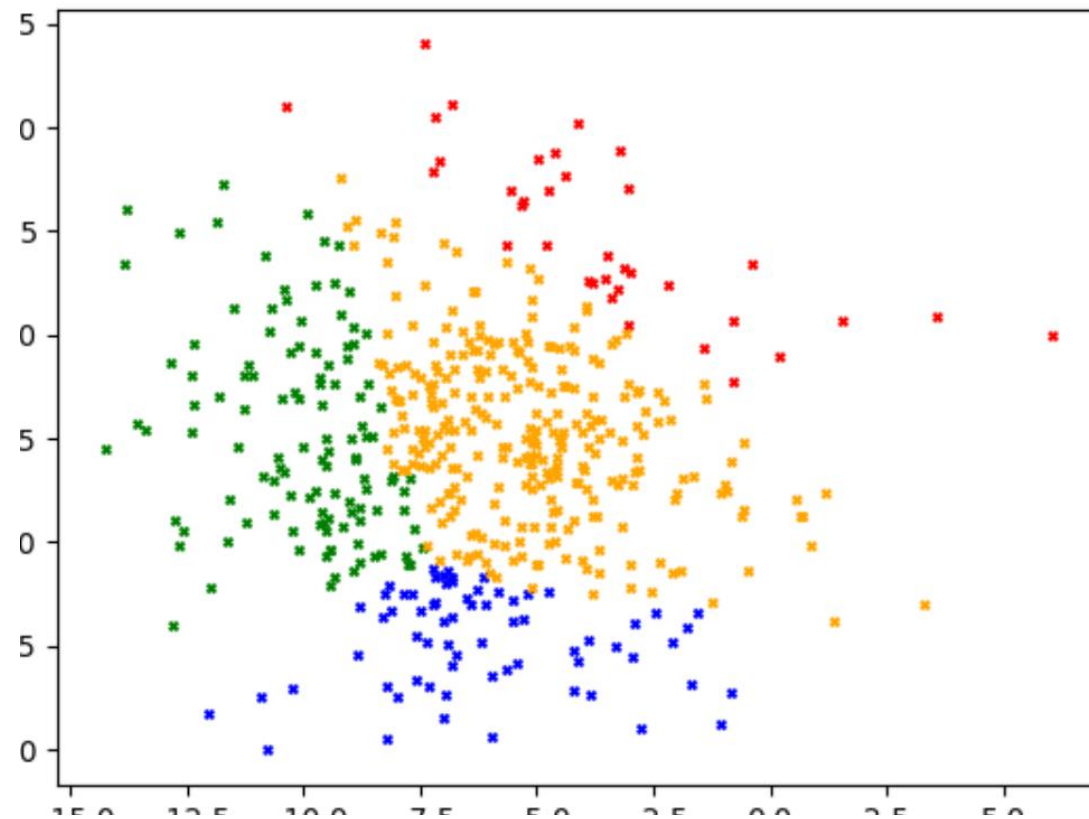
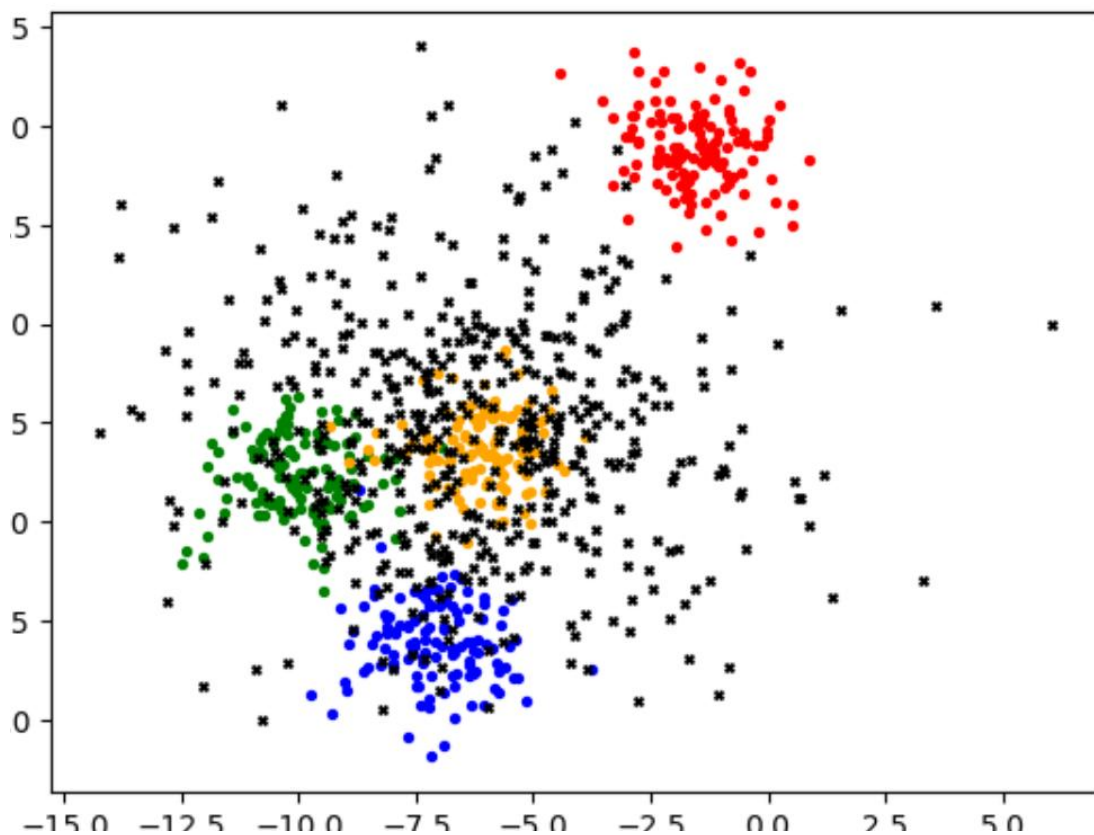
<https://root.cern/manual/tmva/>

The Toolkit for Multivariate Data Analysis with ROOT (TMVA) is a ROOT-integrated project providing a machine learning environment for the processing and evaluation of multivariate classification, both binary and multi class, and regression techniques targeting applications in high-energy physics. The package includes:

- Neural networks
 - Deep networks
 - Multilayer perceptron
- Boosted/Bagged decision trees
- Function discriminant analysis (FDA)
- Linear discriminant analysis (H-Matrix and Fisher discriminants)
- Multidimensional probability density estimation (PDE - range-search approach)
- Multidimensional k-nearest neighbour classifier
- Predictive learning via rule ensembles (RuleFit)
- Projective likelihood estimation (PDE approach)
- Rectangular cut optimisation
- Support Vector Machine (SVM)

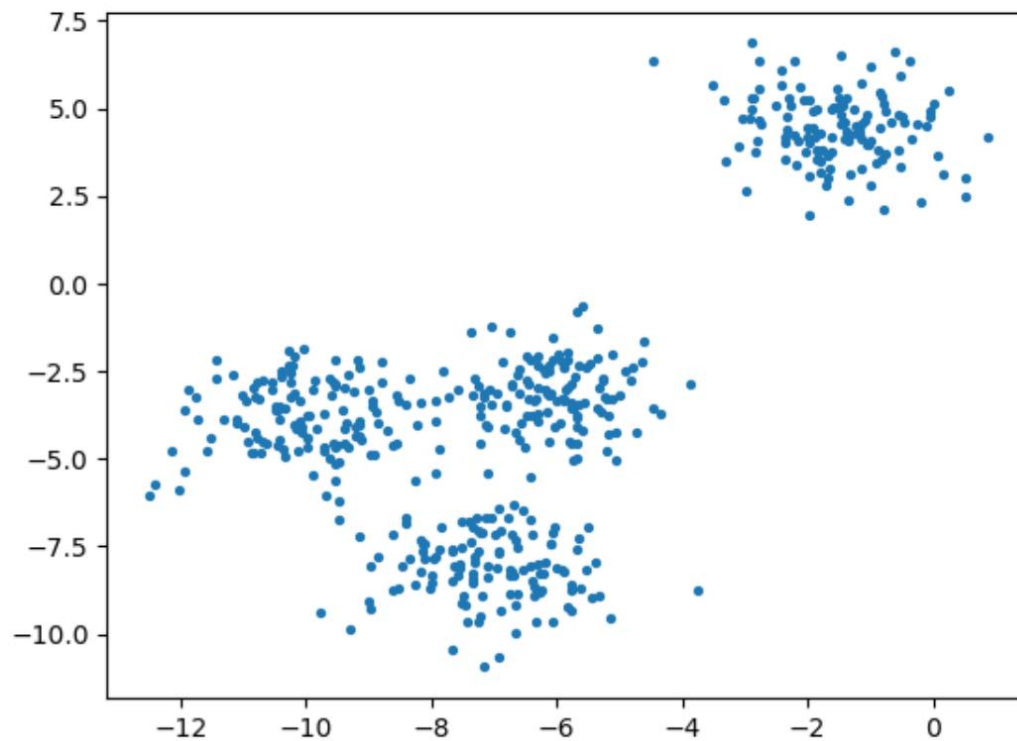
K-nearest neighbor KNN

- 分类问题（监督学习分类问题）



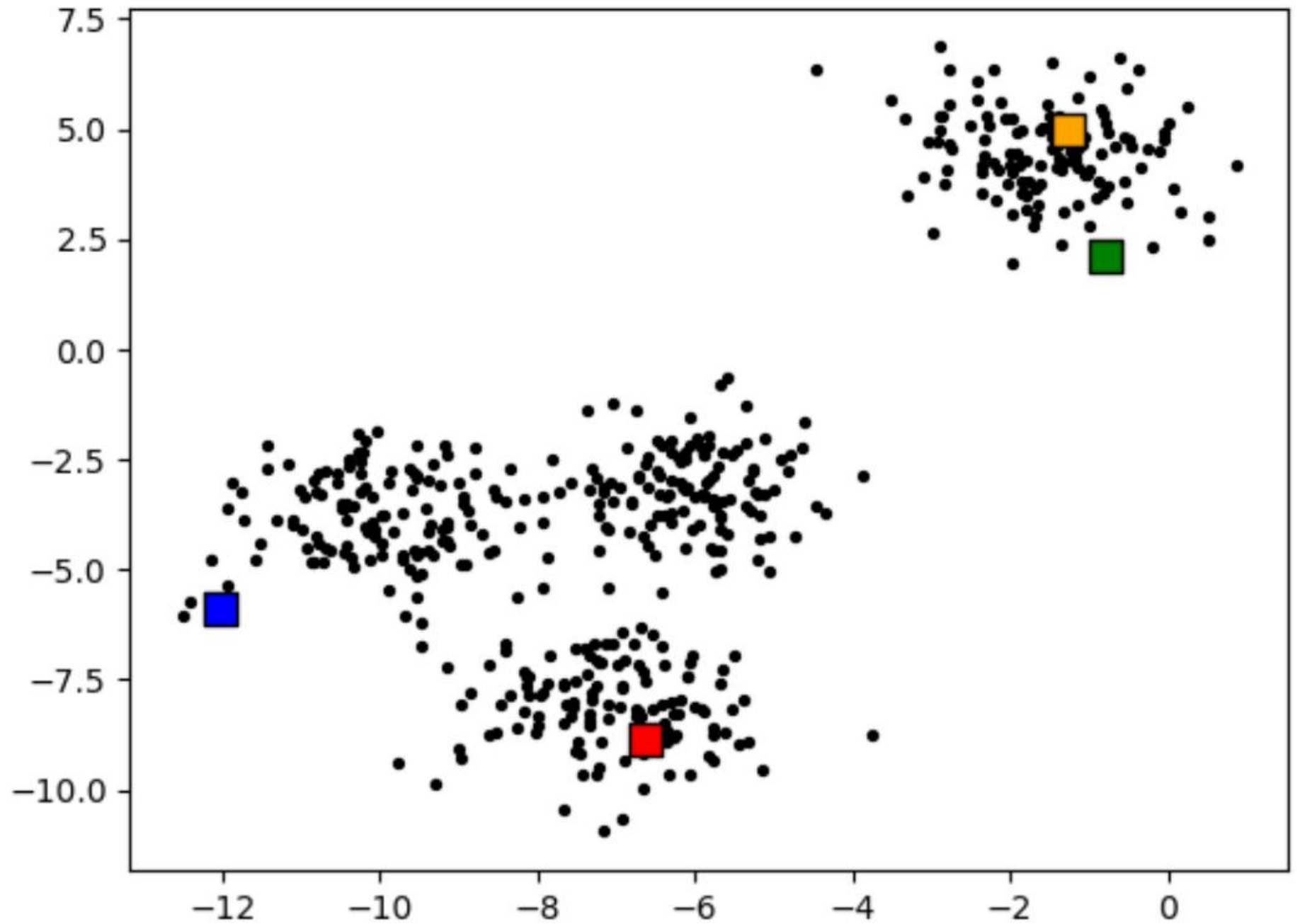
Kmeans

- 聚类问题（非监督分类）



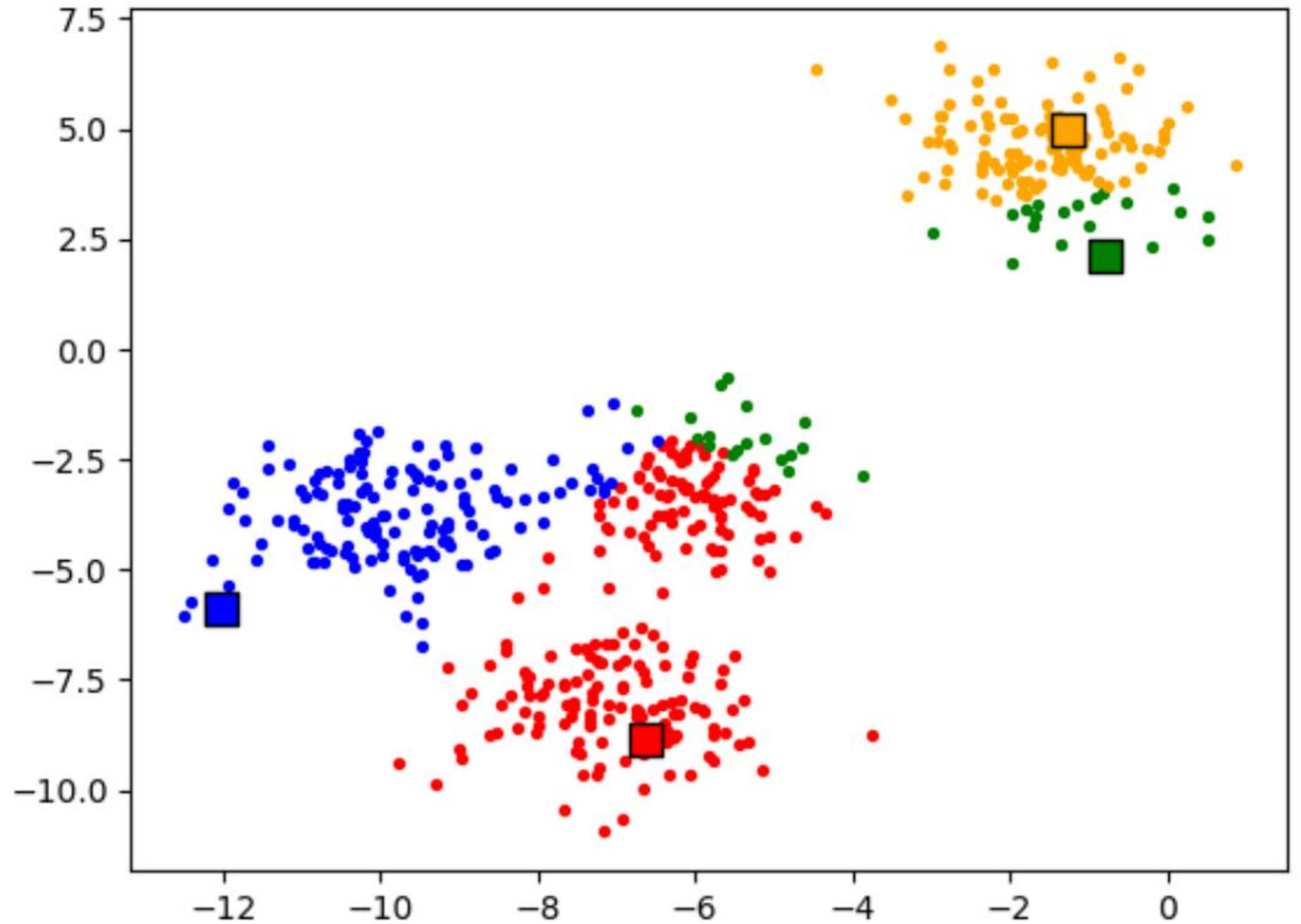
Kmeans

- 1:



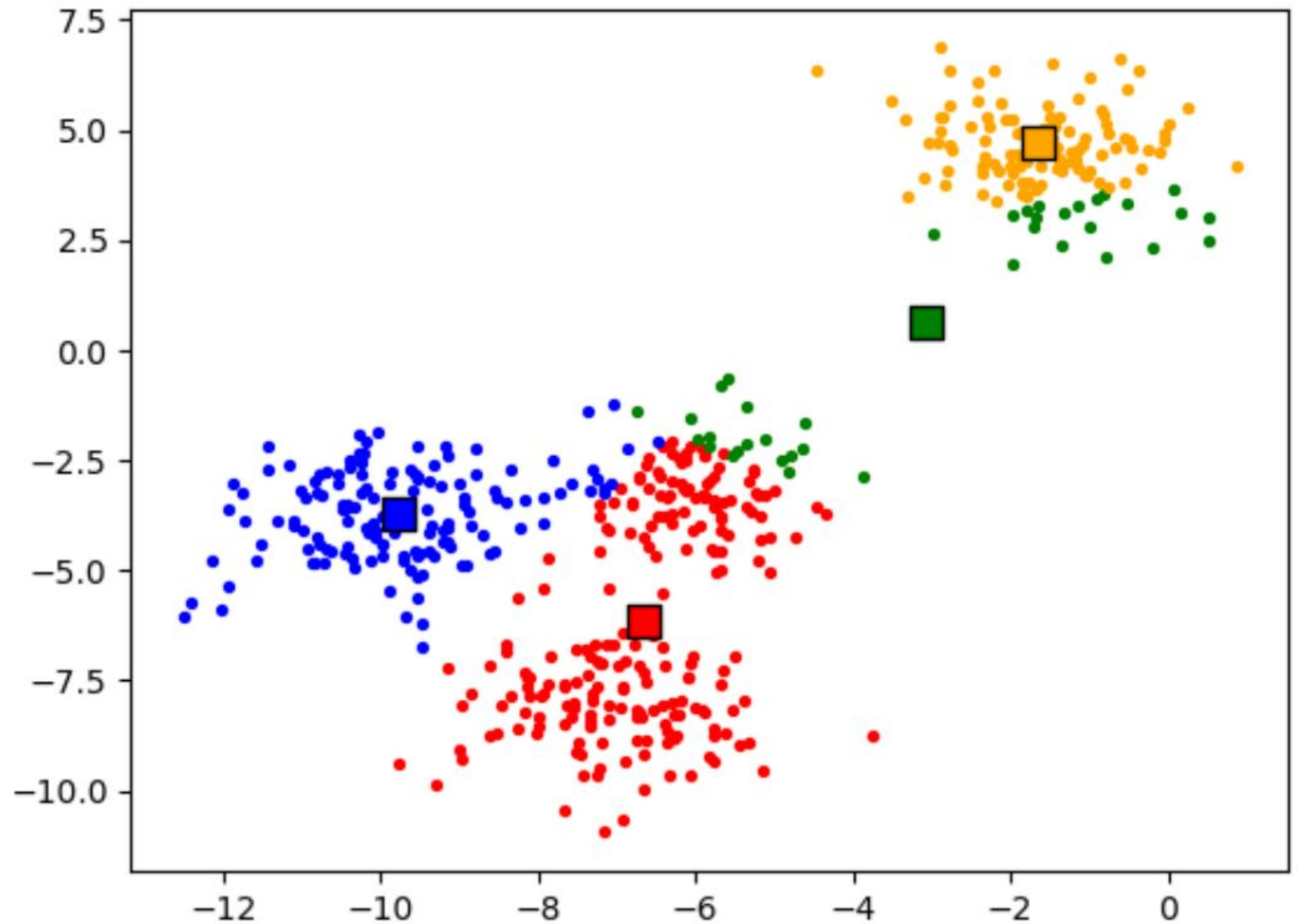
Kmeans

- 2:



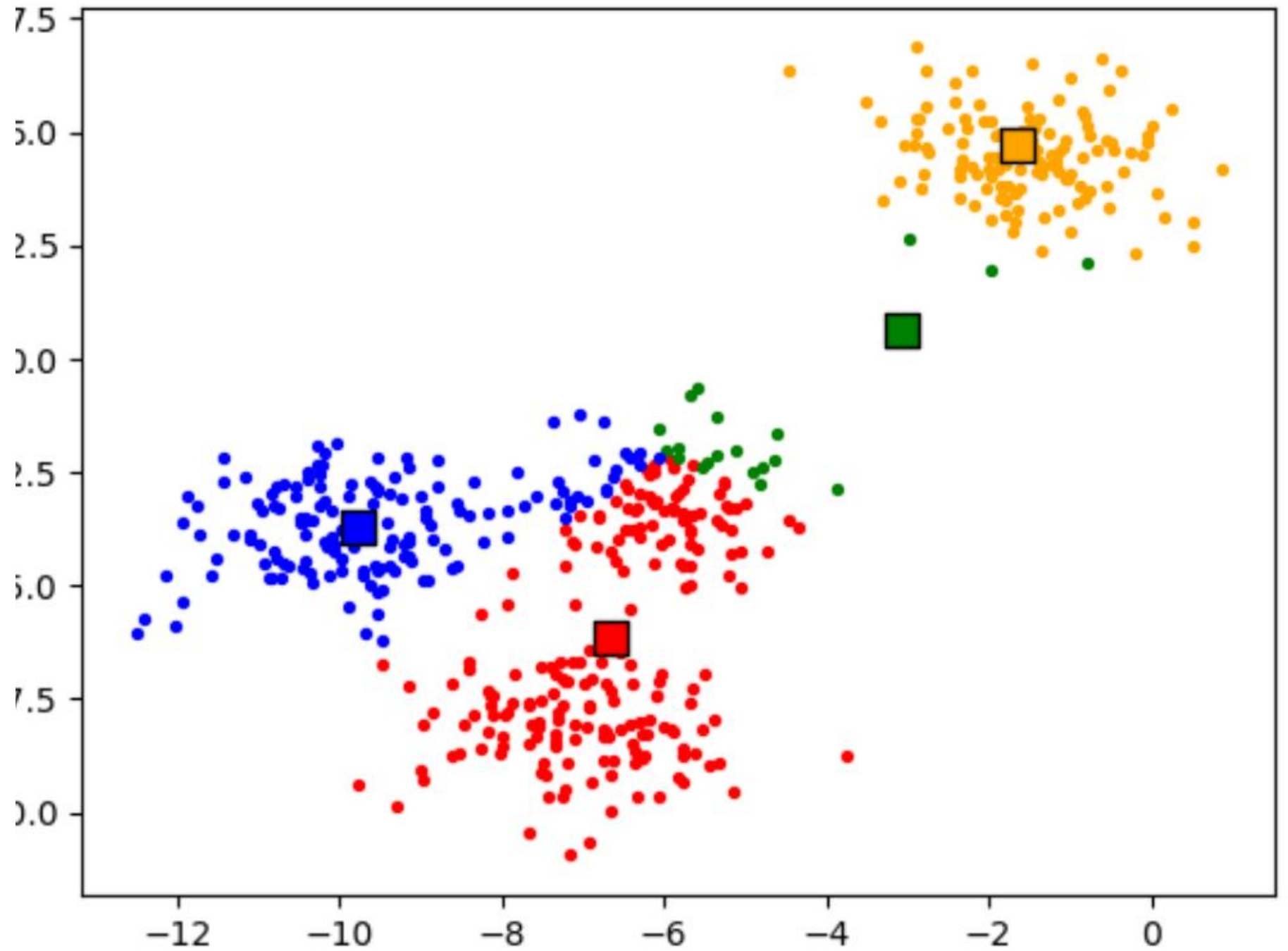
Kmeans

- 3:



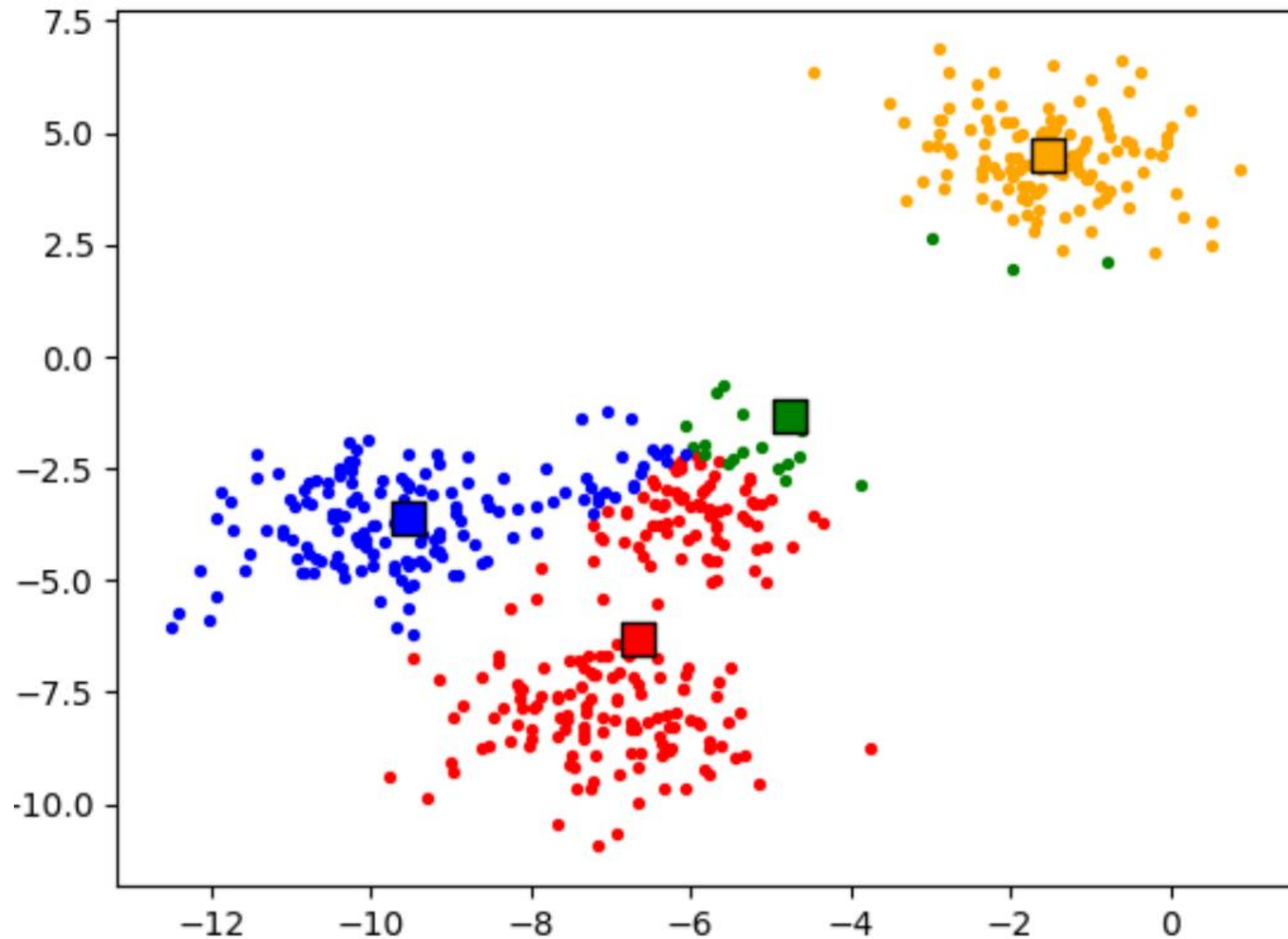
Kmeans

- 4:



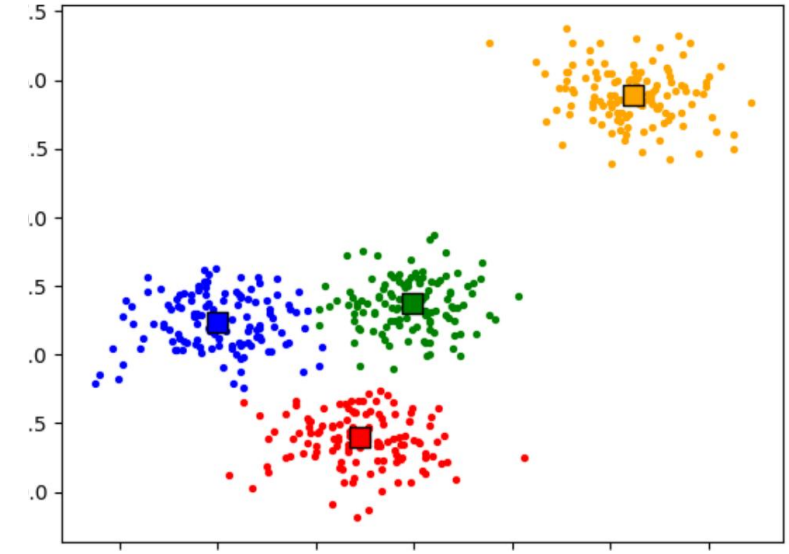
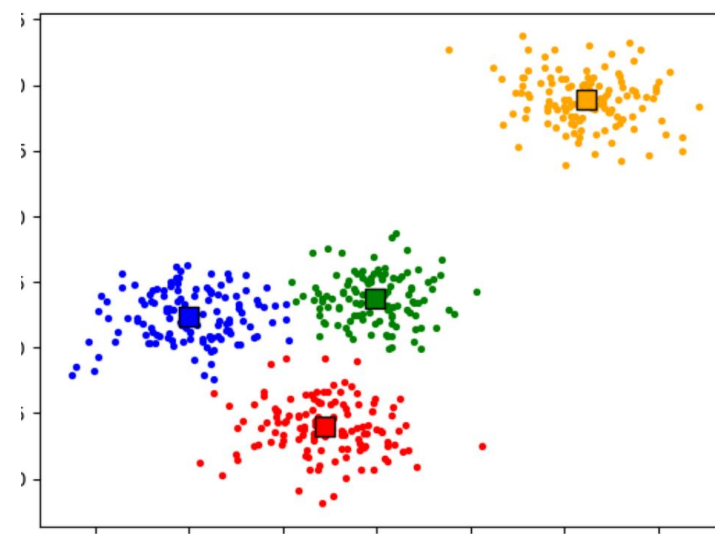
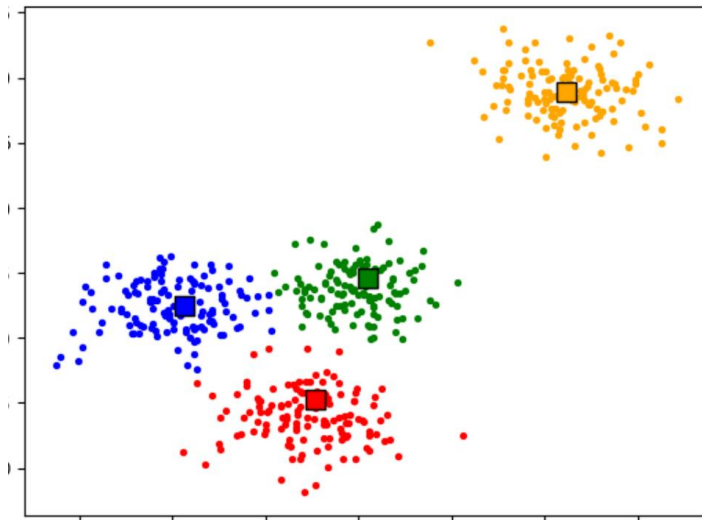
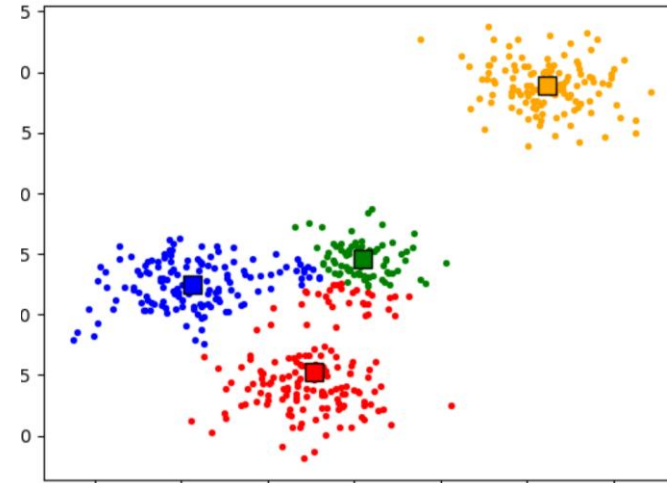
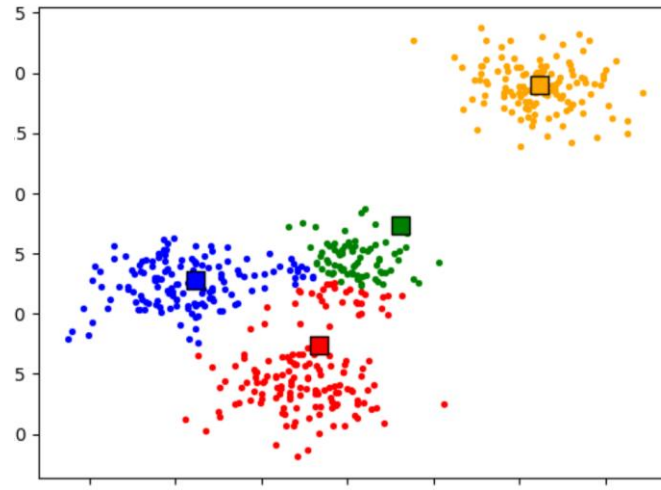
Kmeans

- 5:



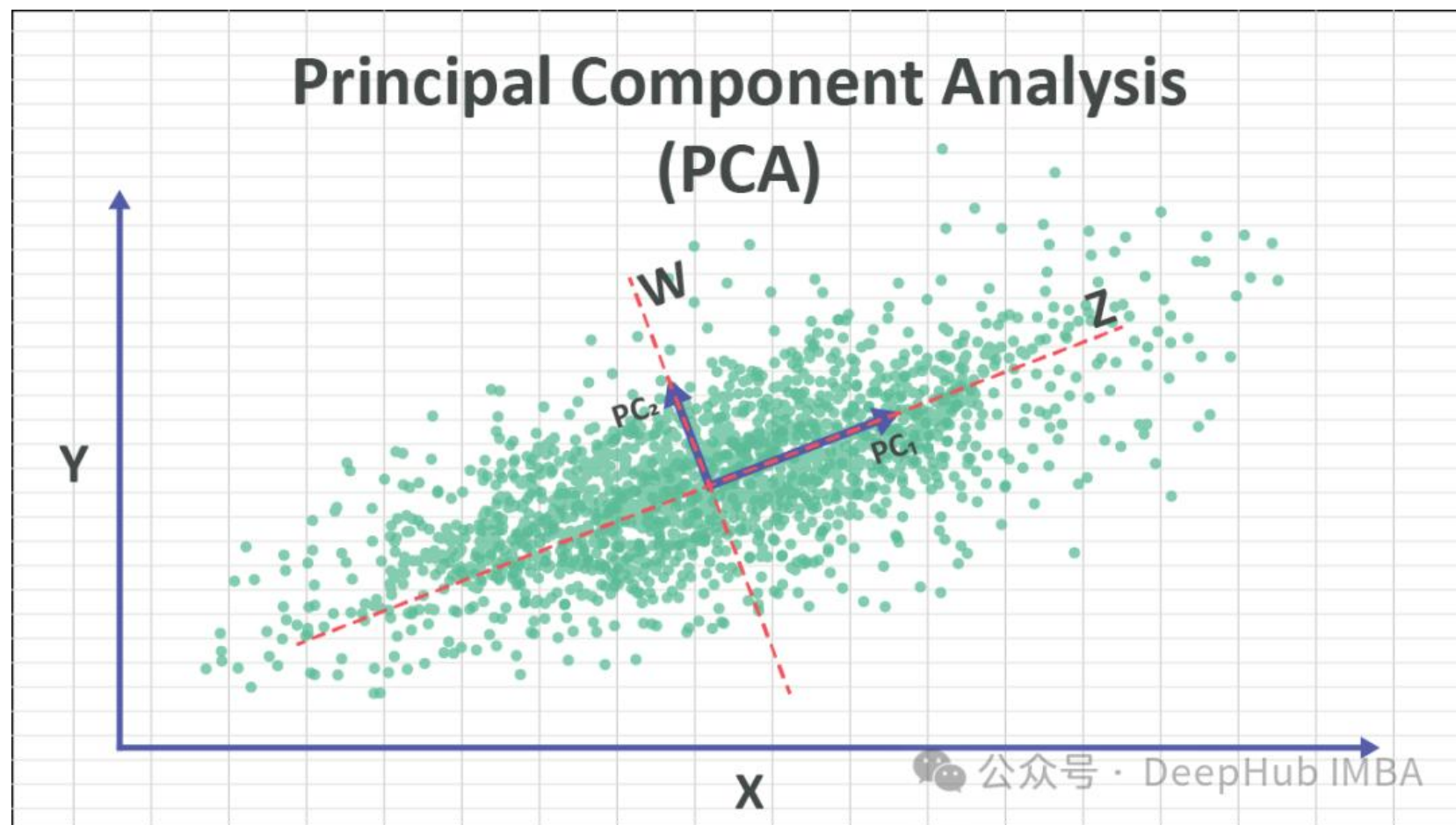
Kmeans

- n:



Principle component analysis(PCA)

- 主成分分析:



PCA

$$\text{Cov}(X, Y) = E[(X - E[X])(Y - E[Y])]$$

对于一个 d 维随机向量 $\mathbf{X} = (X_1, X_2, \dots, X_d)^\top$, 其协方差矩阵 Σ 是一个 $d \times d$ 的矩阵, 其中每个元素 Σ_{ij} 是随机变量 X_i 和 X_j 之间的协方差:

$$\Sigma = \begin{pmatrix} \text{Cov}(X_1, X_1) & \text{Cov}(X_1, X_2) & \cdots & \text{Cov}(X_1, X_d) \\ \text{Cov}(X_2, X_1) & \text{Cov}(X_2, X_2) & \cdots & \text{Cov}(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(X_d, X_1) & \text{Cov}(X_d, X_2) & \cdots & \text{Cov}(X_d, X_d) \end{pmatrix}$$

- (1) z-score
- (2) 协方差矩阵

$$X = (\vec{x}'^1, \vec{x}'^2, \dots, \vec{x}'^N)$$

$$\Sigma = XX^T$$

- (3) 计算M的本征矢量 (按本征值排序)

PCA

$$pp \rightarrow \gamma \nu \bar{\nu}$$

$$E_\gamma$$

$$p_\gamma^x$$

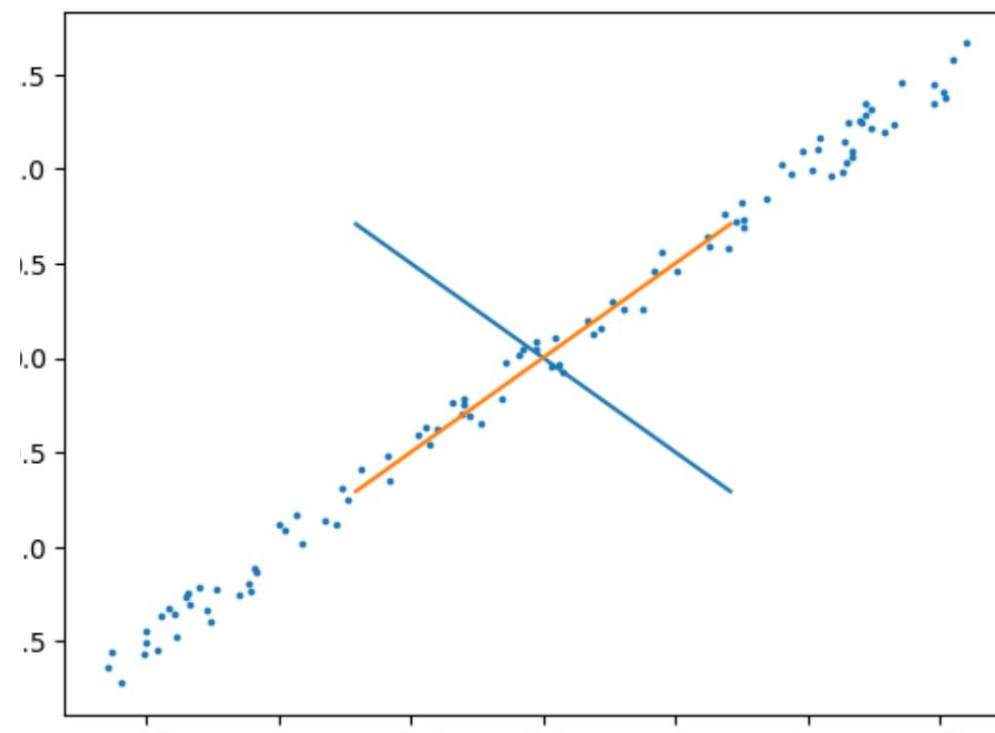
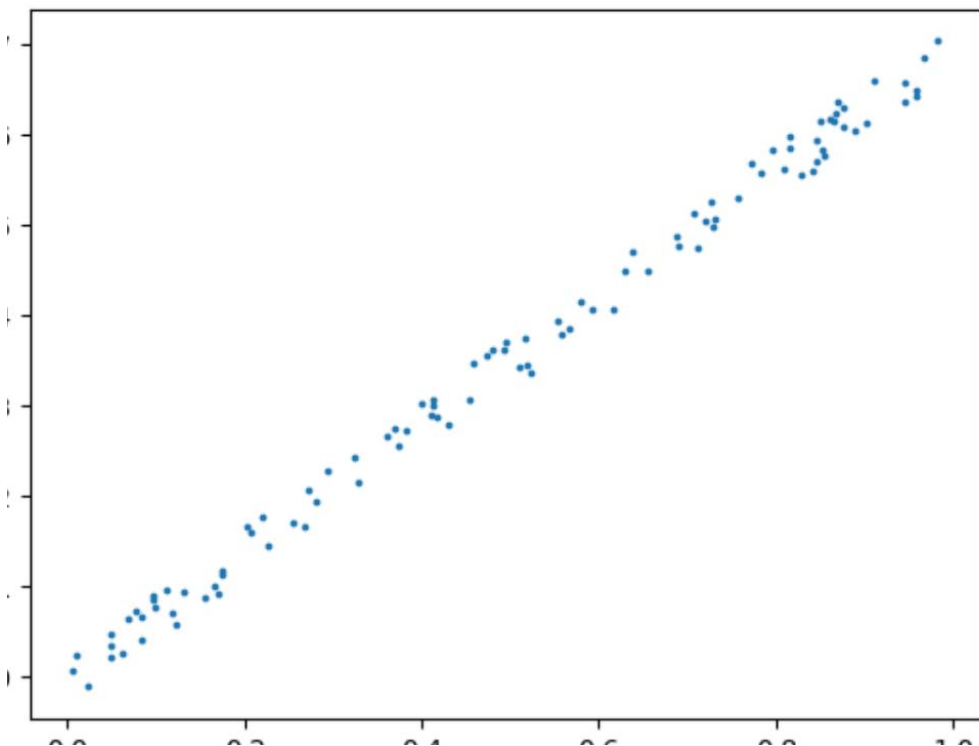
$$p_\gamma^y$$

$$p_\gamma^z$$

$$\cancel{E}_T^x$$

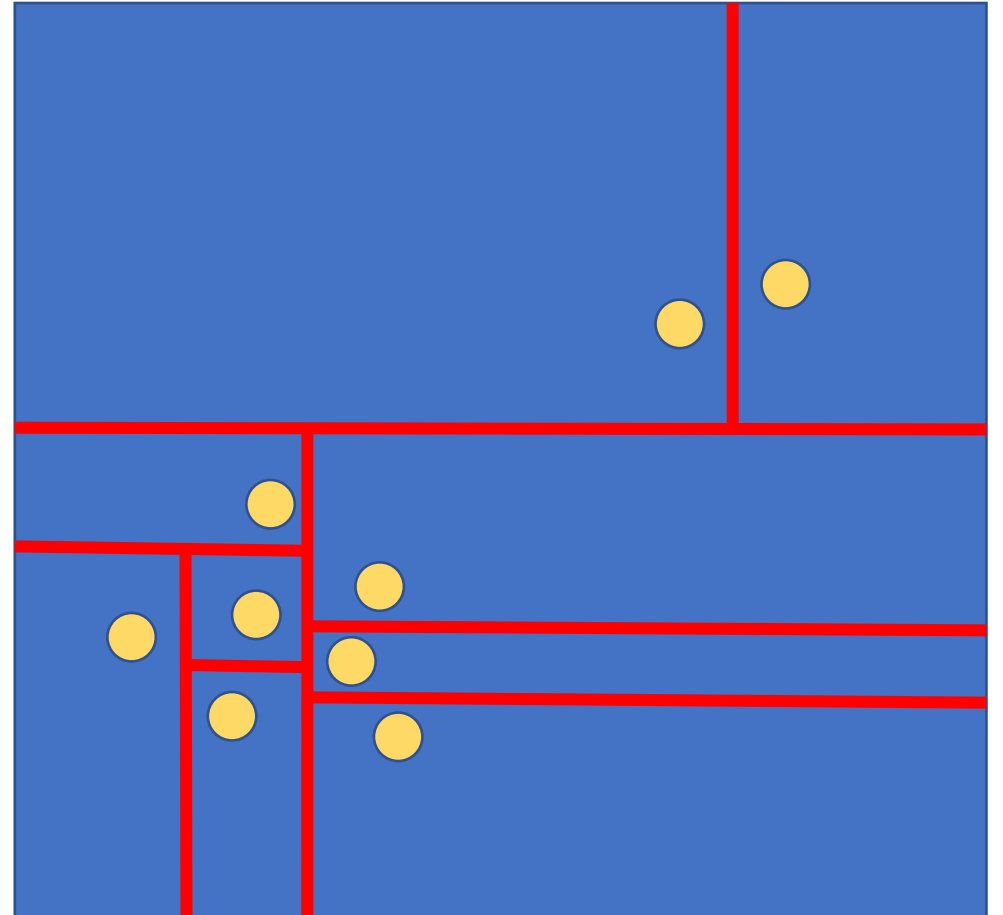
$$\cancel{E}_T^y$$

- 降维, 寻找异常...

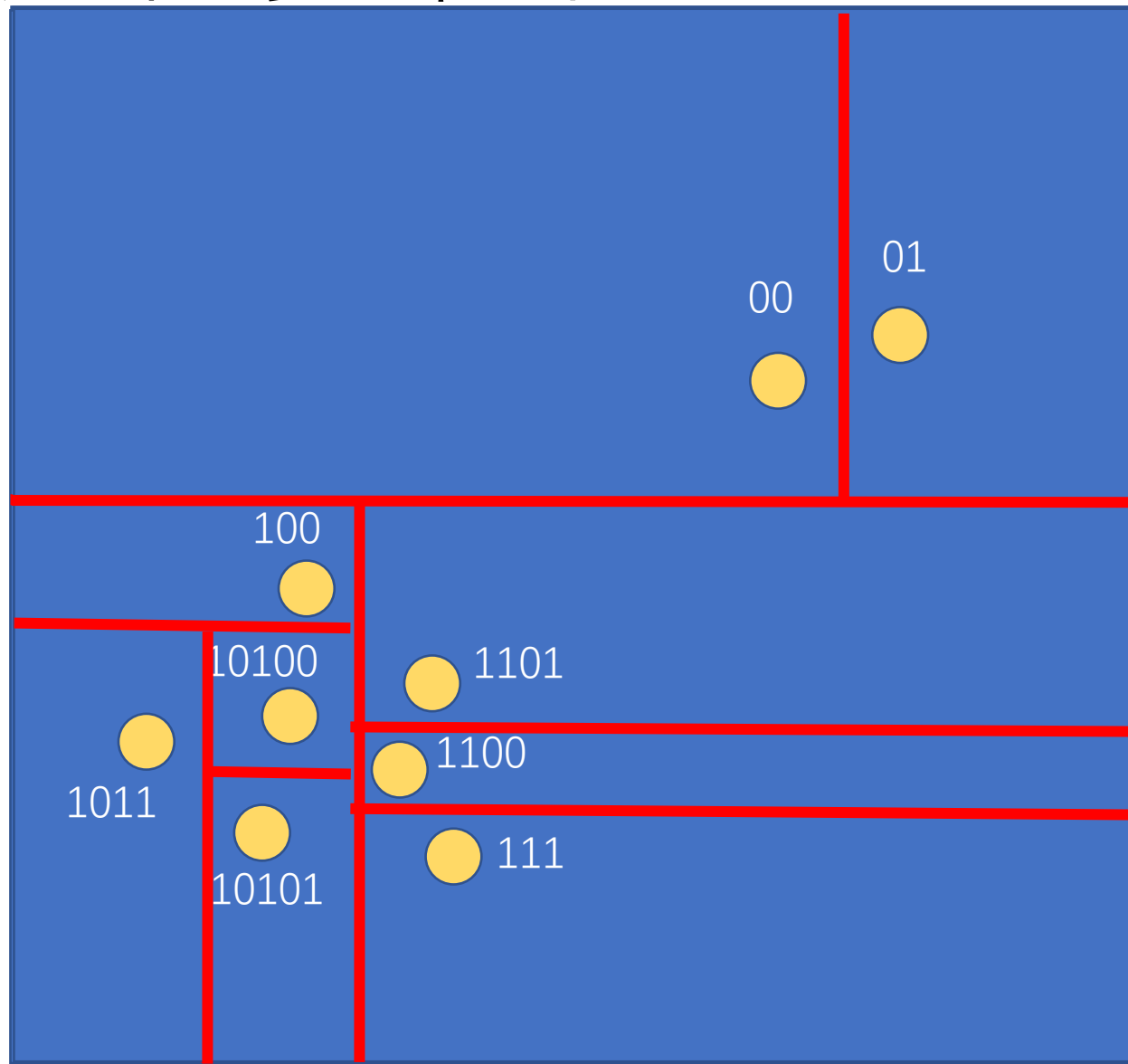


Isolation tree

- 随机选择一个还未分割的“叶子”
- 随机选择一个分割的维度
- 分割

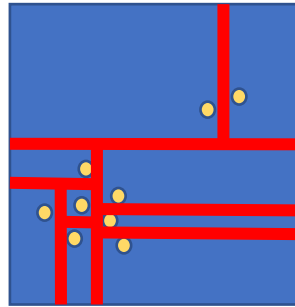


异常点，是最好分割的那些

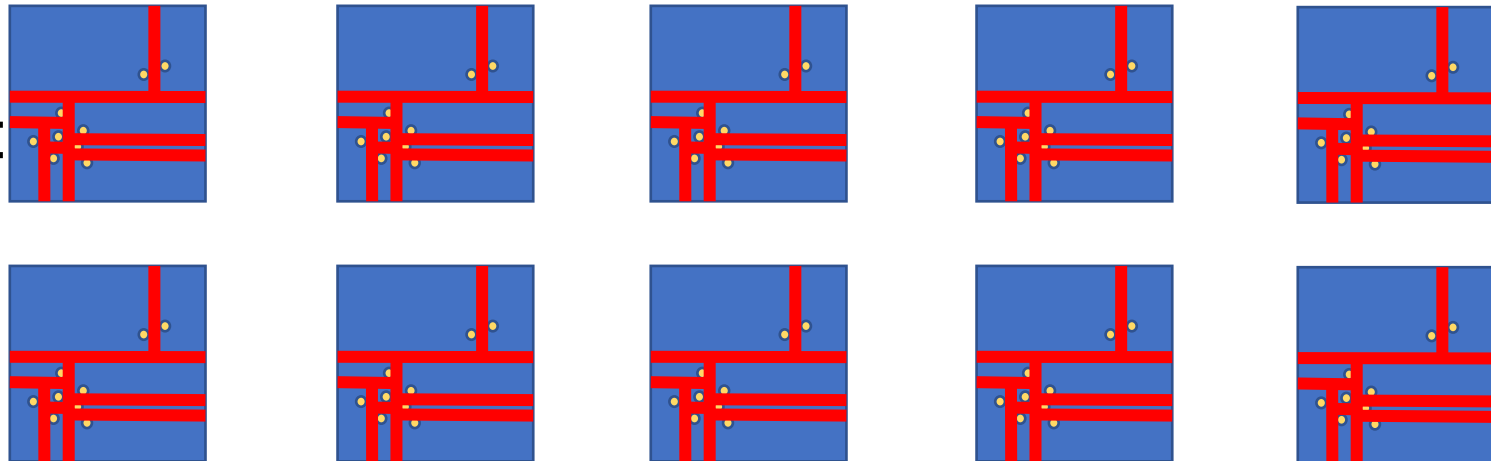


Isolation tree $\rightarrow$ isolation forest

- Isolation tree:



- Isolation forest:



Isolation tree

- 归一化：平均深度依赖于数据量，因此需要归一化

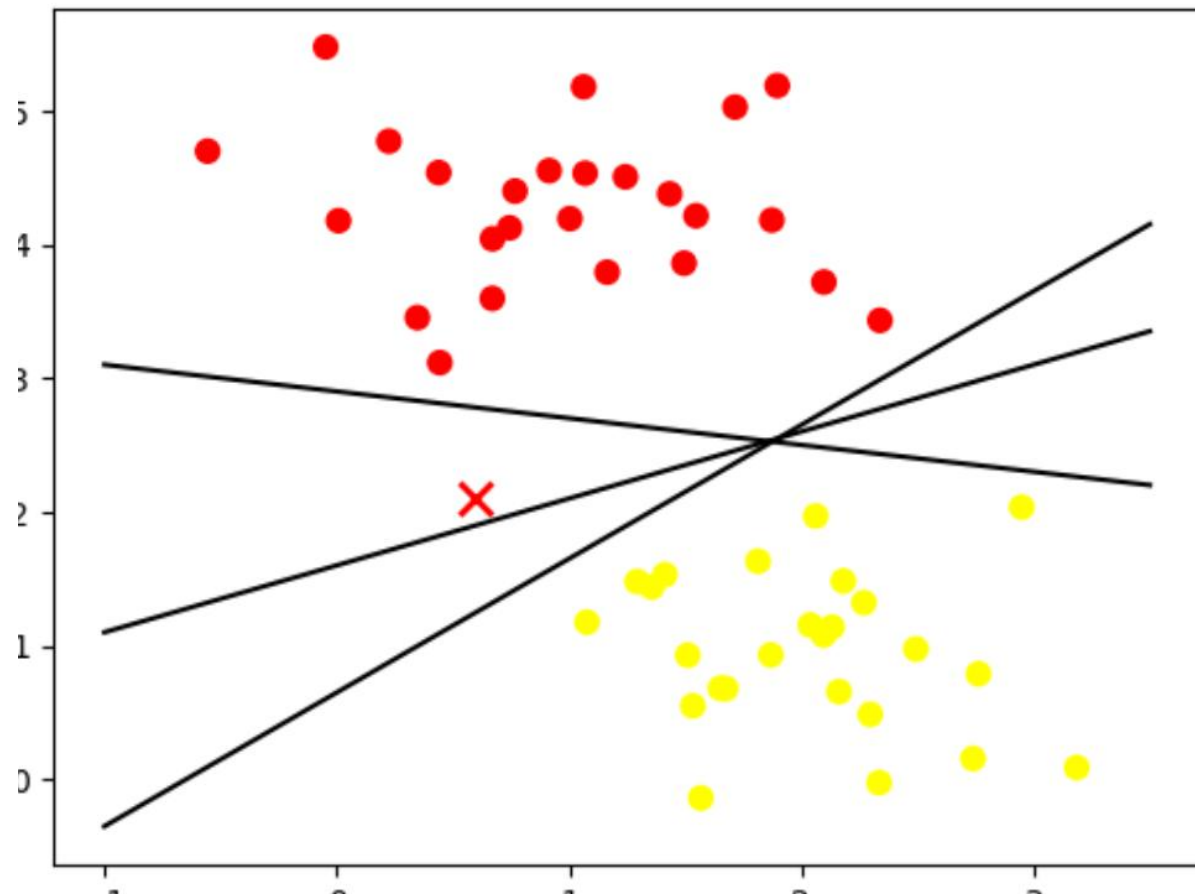
$$a = 2^{-\frac{\bar{L}}{c(N)}}$$

$$c(N) = 2H(N-1) - 2\frac{N-1}{N}$$

$$H(x) = \sum_{n=1}^x \frac{1}{n}$$

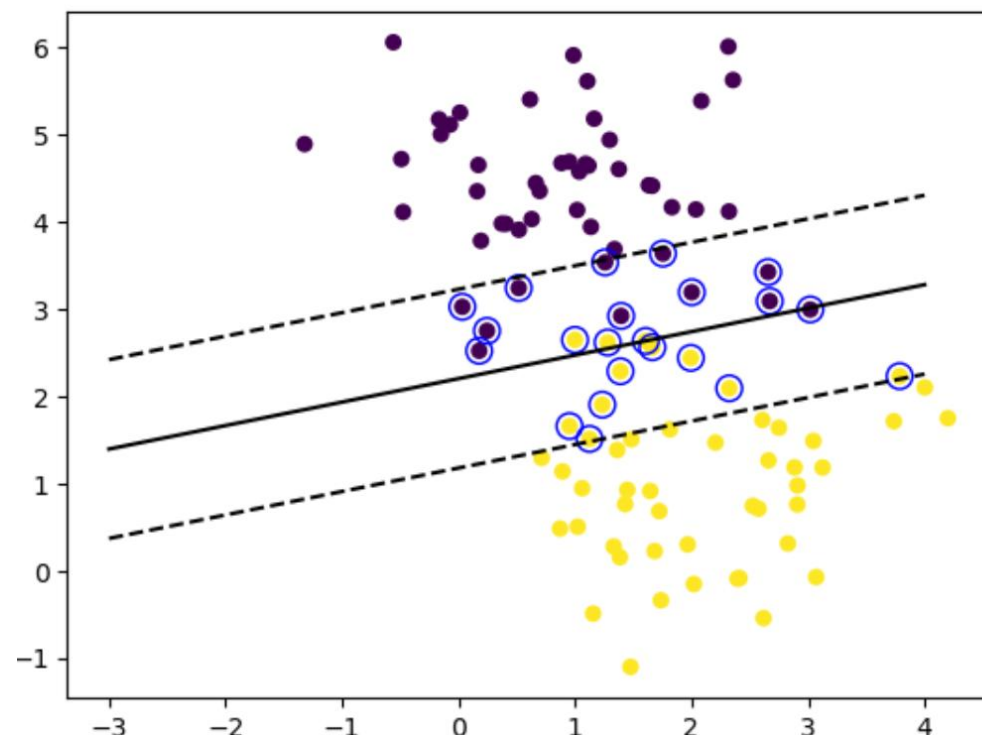
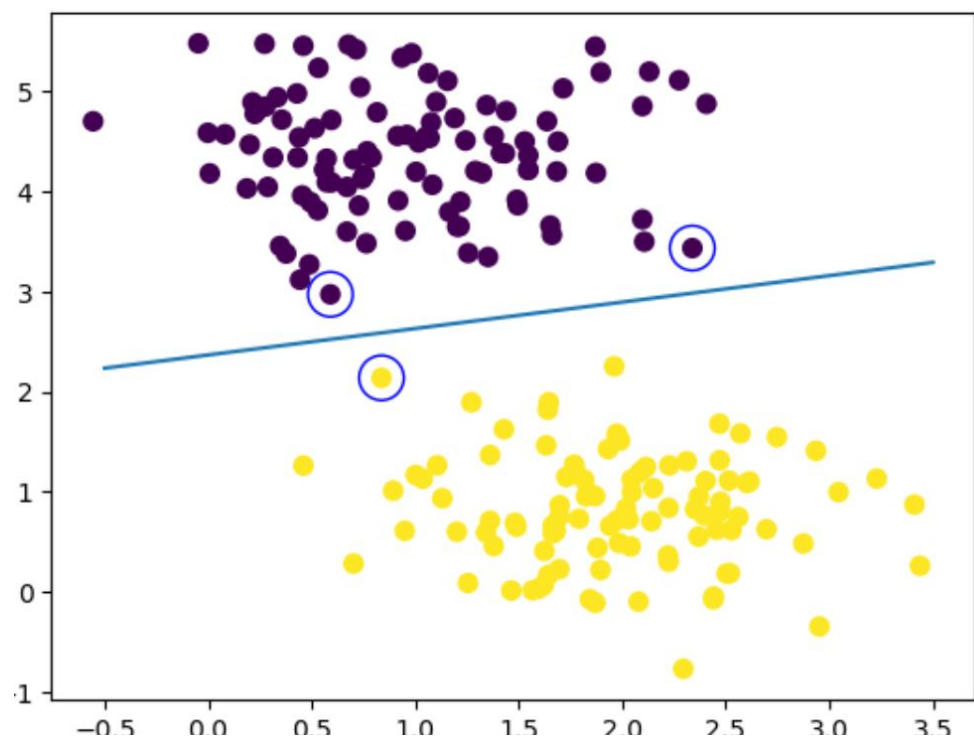
Support vector machine

- 最“好”的分割



SVM

- 硬分割, 软分割



一些问题:

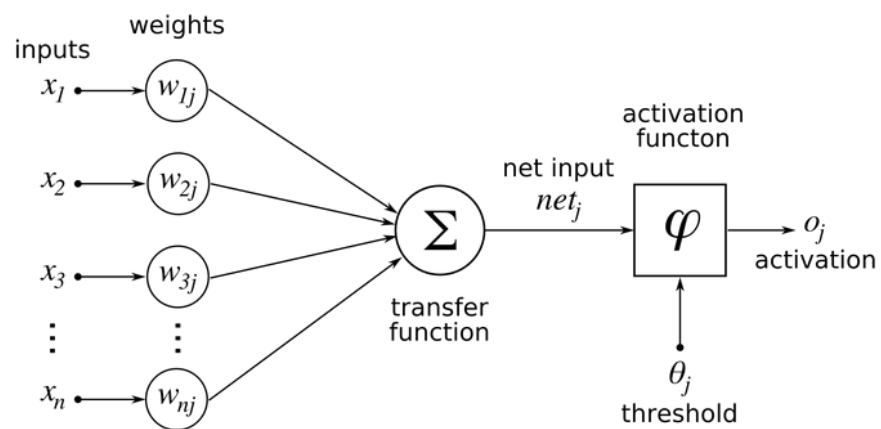
- 在搜索新物理中的应用
- 干涉项?
- 异常检测 vs 分类
- Kernel (距离定义)

使用AI学习AI

- Deepseek, ChatGPT, ...
- (实机演示环节的示例代码在deepseek的帮助下完成)

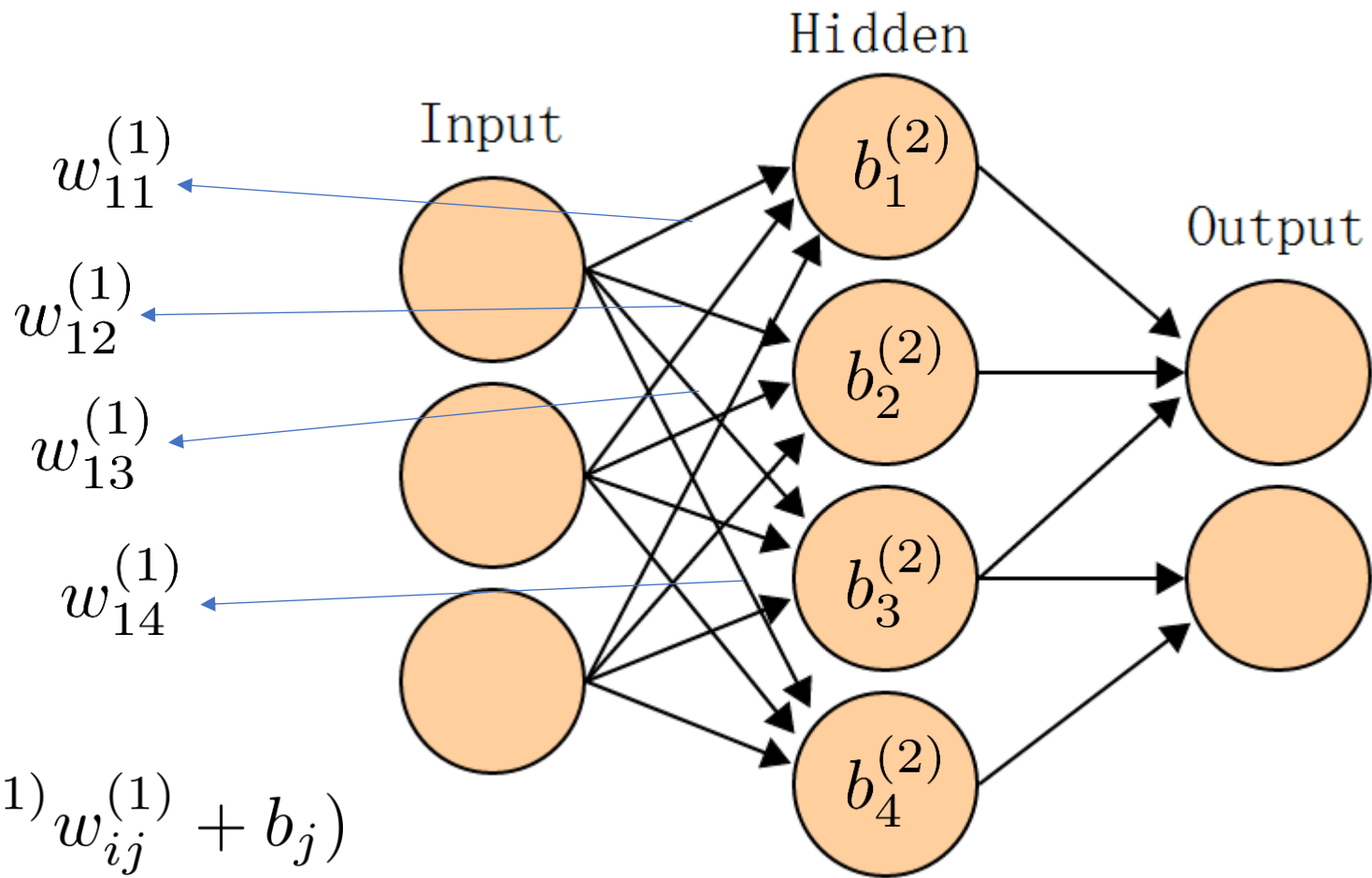
机器学习-神经网络

- 我们能不能数值模拟一个大脑?



神经网络

- 偏移矢量
- 权重矩阵
- 激活函数



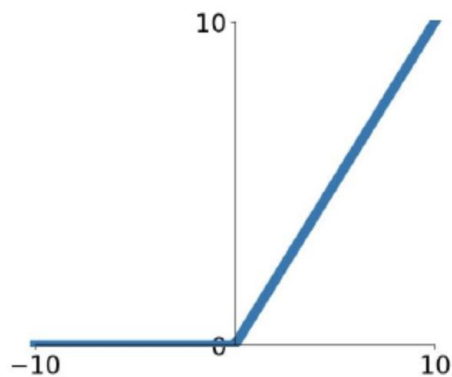
$$x_j^{(2)} = f(\sum x_i^{(1)} w_{ij}^{(1)} + b_j)$$

$$\vec{x}^{(2)} = f(W\vec{x}^{(1)} + \vec{b}^{(2)})$$

$$f_{relu}(x) = \begin{cases} x & x > 0 \\ 0 & x < 0 \end{cases}$$

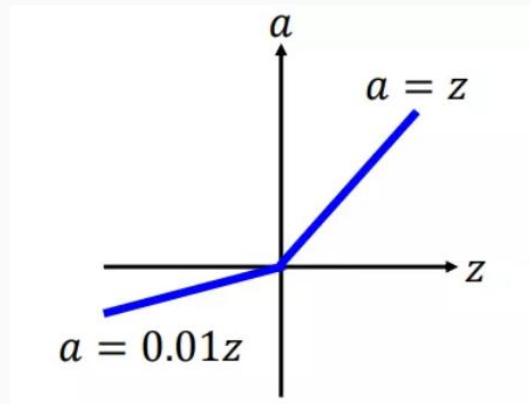
神经网络

- 激活函数，非线性

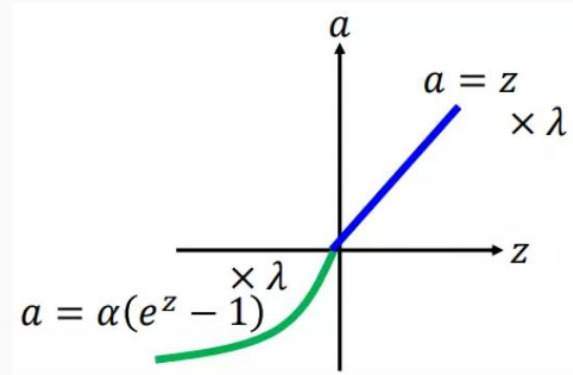


ReLU

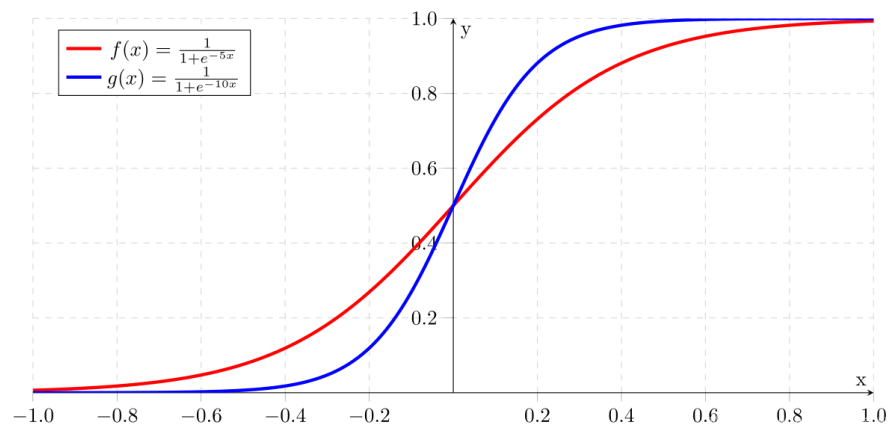
(Rectified Linear Unit)



<https://blog.csdn.net>

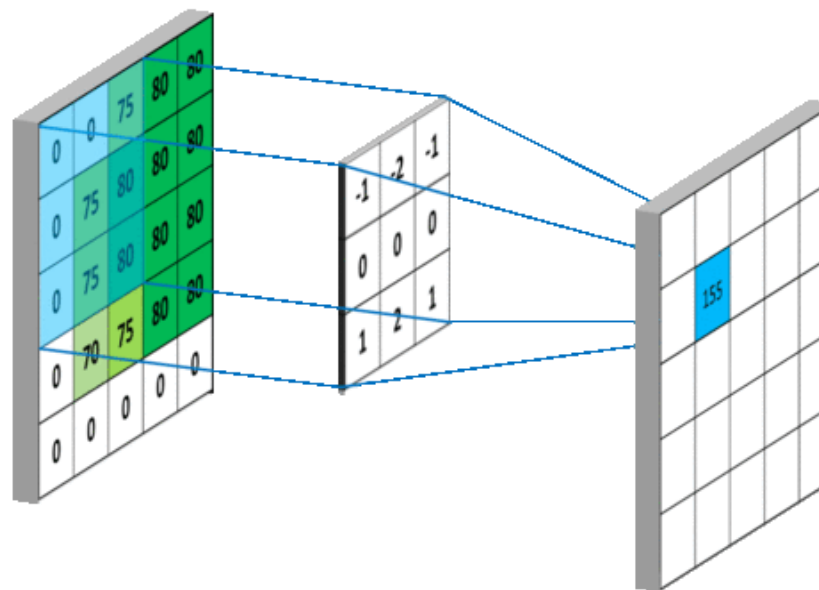
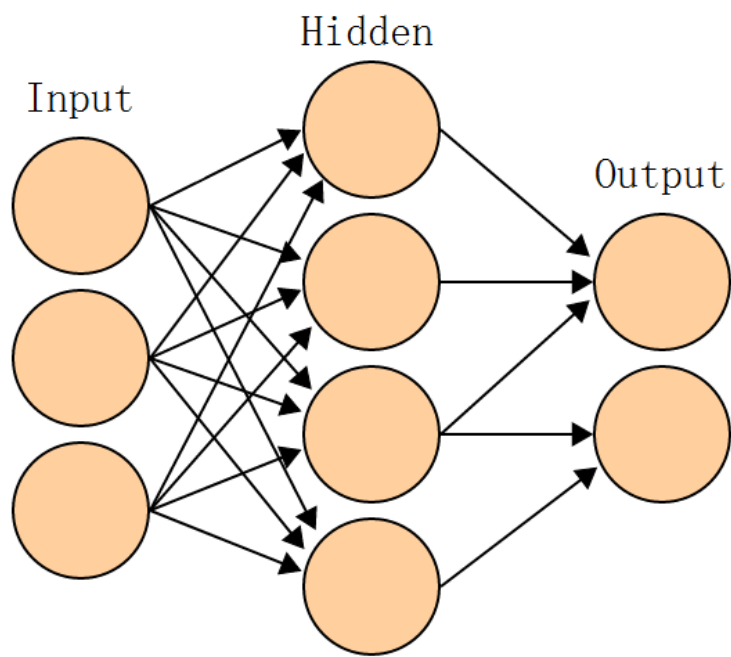


https://blog.csdn.net/weixin_37947156



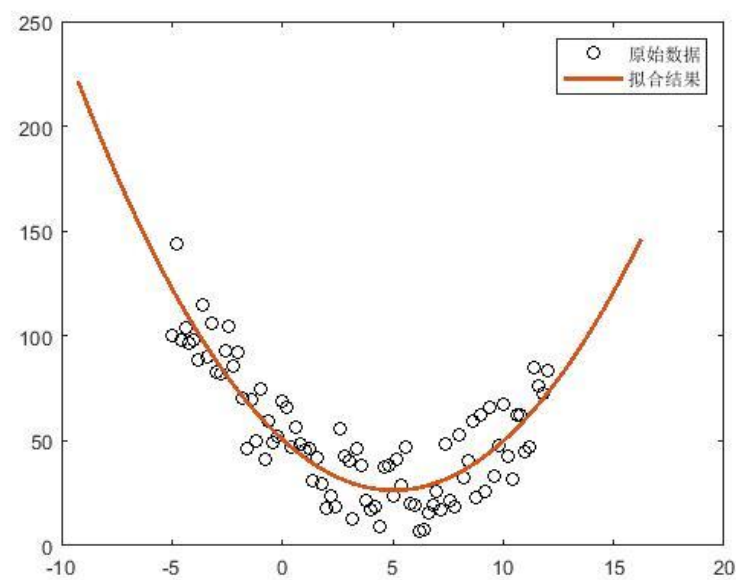
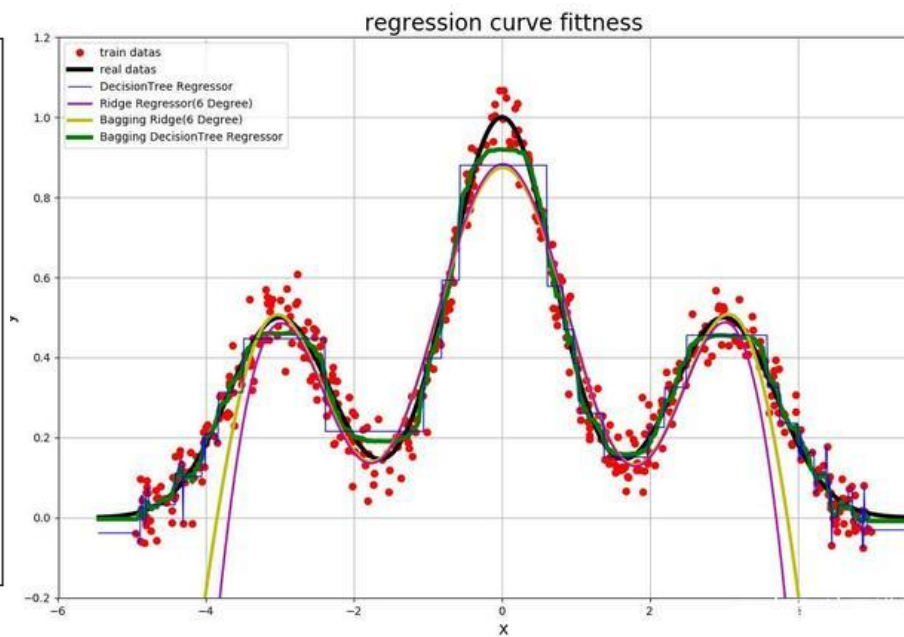
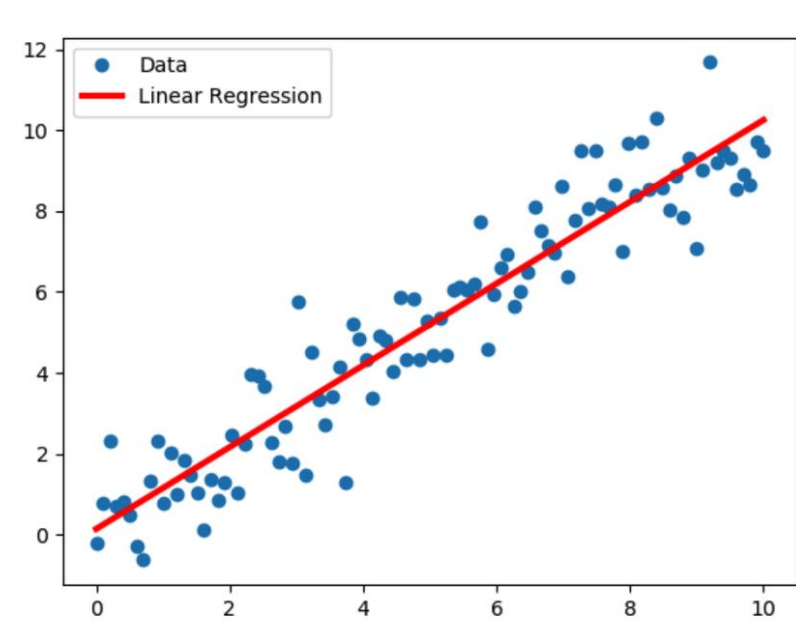
神经网络

- 连接层（全连接，CNN，dropout）



神经网络

- 训练=拟合

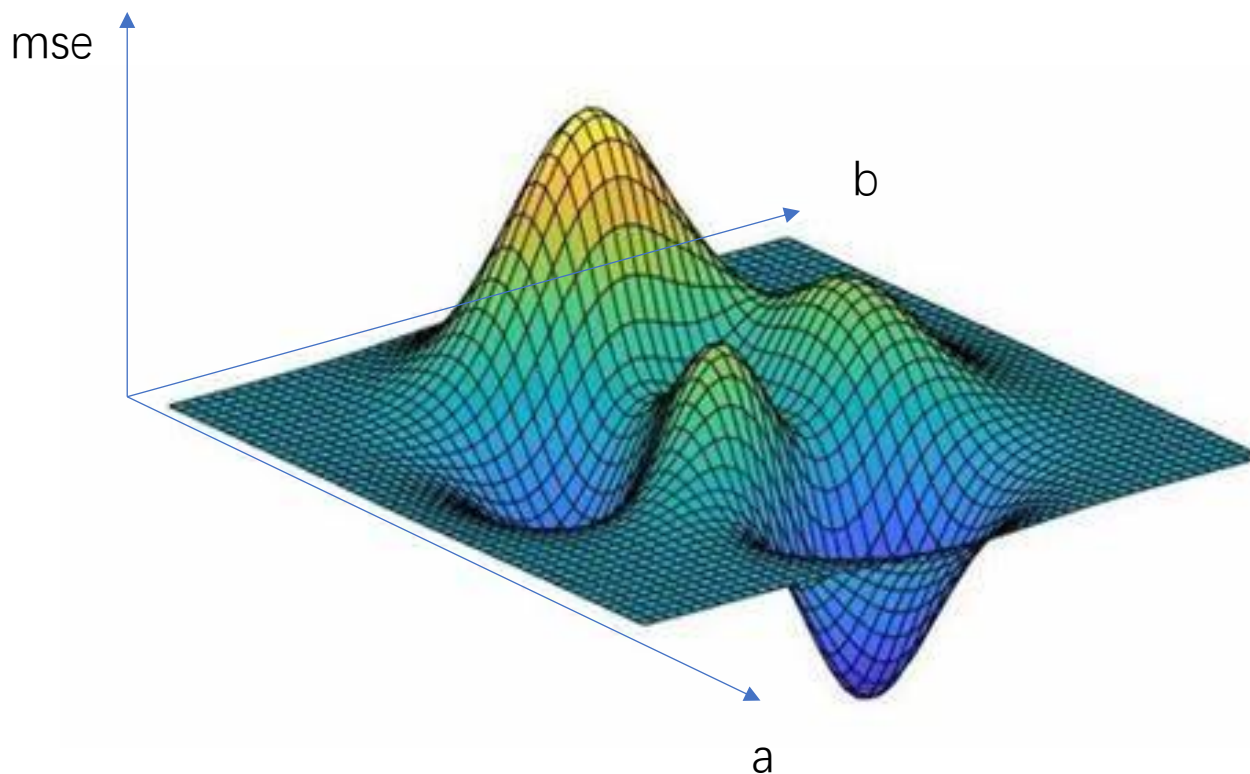


神经网络

样本

$$f_{a,b}(x) = ax + b$$

- 拟合=寻找n维曲面的最低点 $(x_i, y_i) \Rightarrow mse = \frac{1}{n} \sum_i (y_i - f_{a,b}(x_i))^2$



神经网络

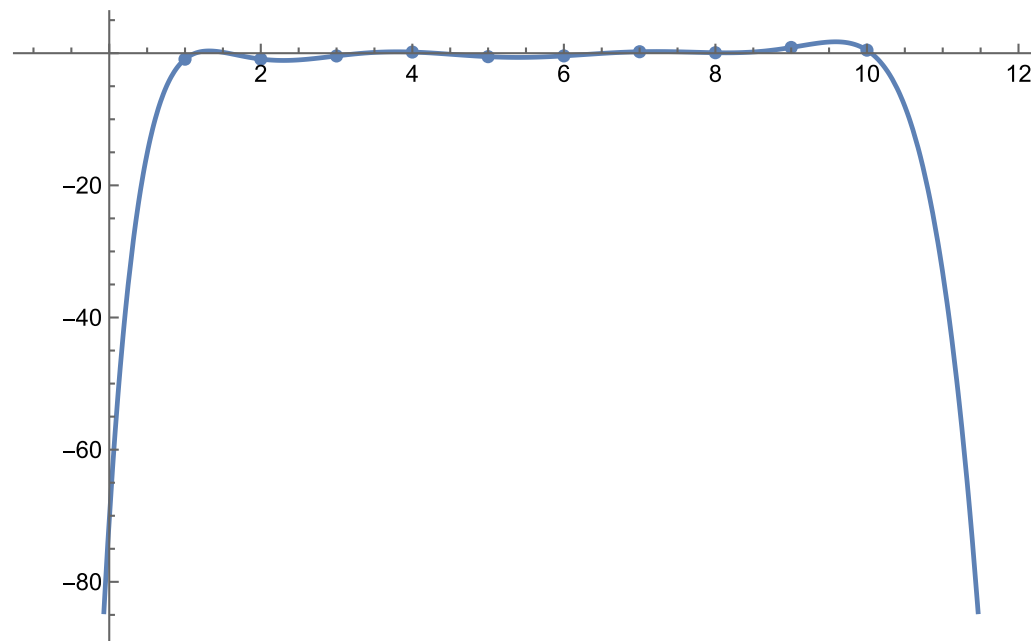
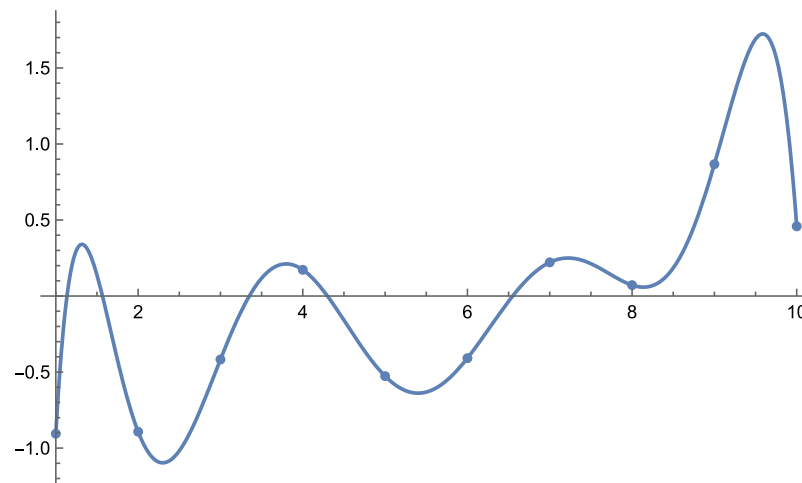
- Loss function: mse, mae, 交叉熵, ...
- Optimizer: 最速下降, ..., adam
- (无须担心, 全部都写好了, 直接拿来用)

神经网络

- 过拟合

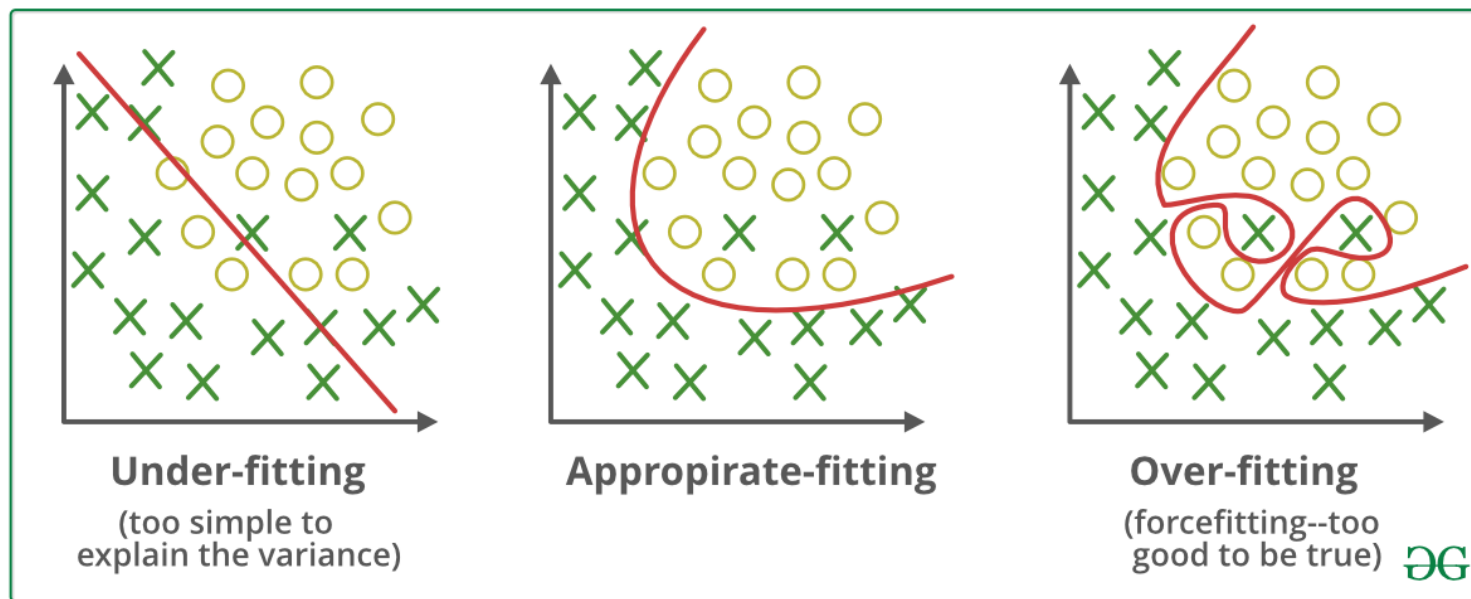
$$(x_i, y_i)$$

$$f_{a_i}(x) = \dots$$



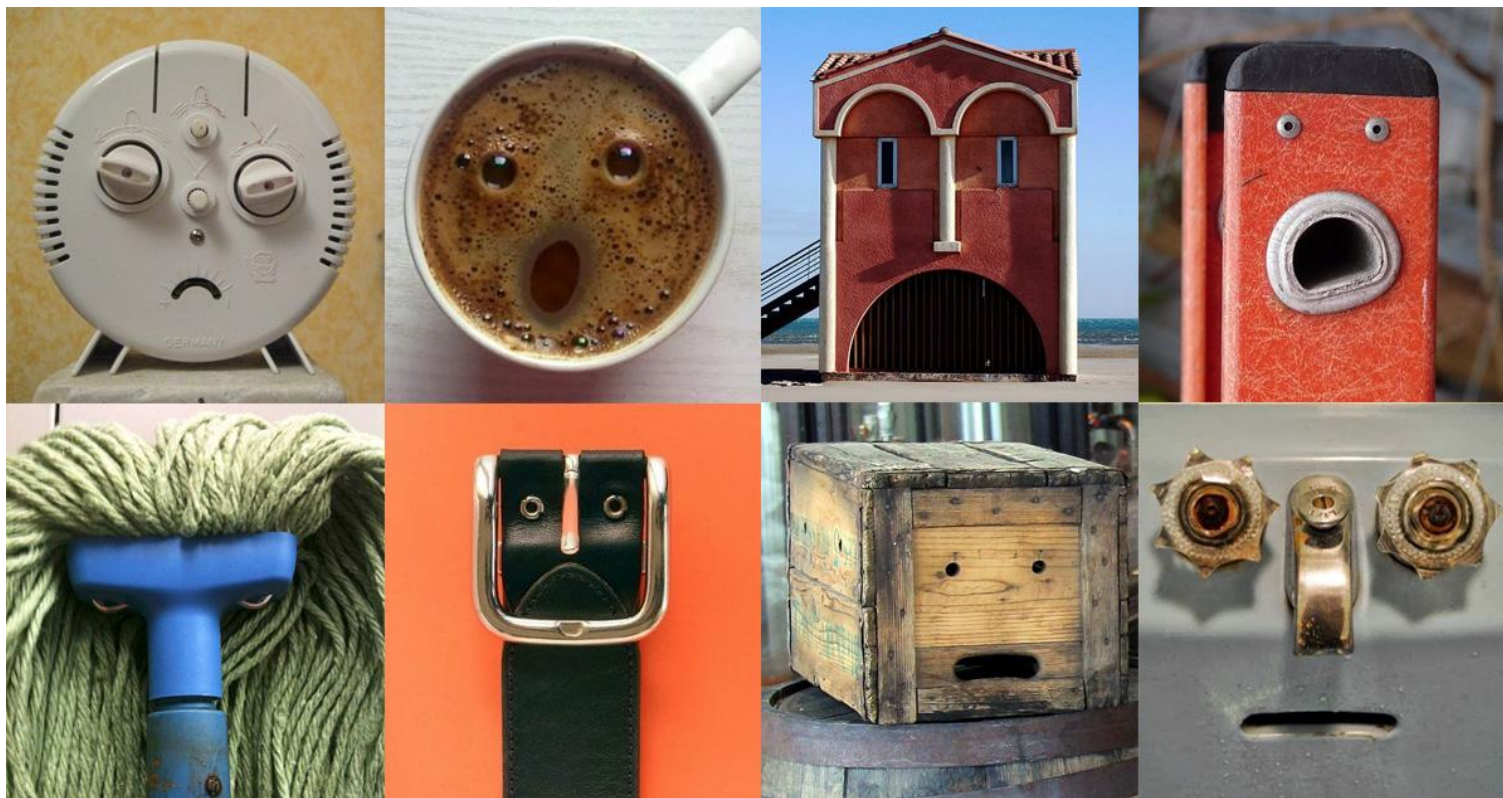
神经网络

- 过拟合-欠拟合



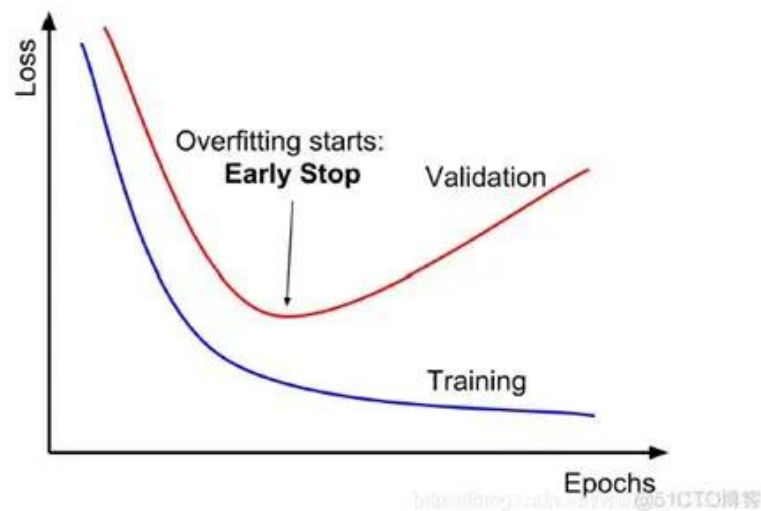
神经网络

- 空想性错视 (Pareidolia)



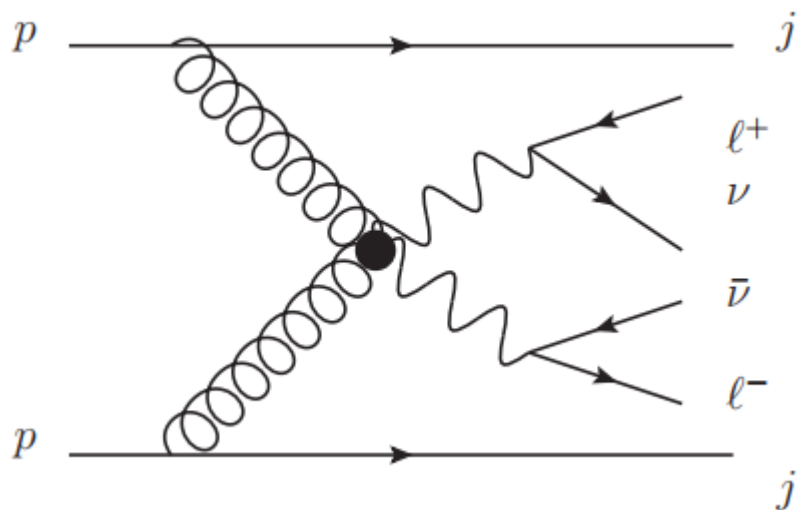
神经网络

- Early stopping
- 数据分组:
- Training
- Validation
- Testing



神经网络

- 最简单的任务：拟合（回归），分类

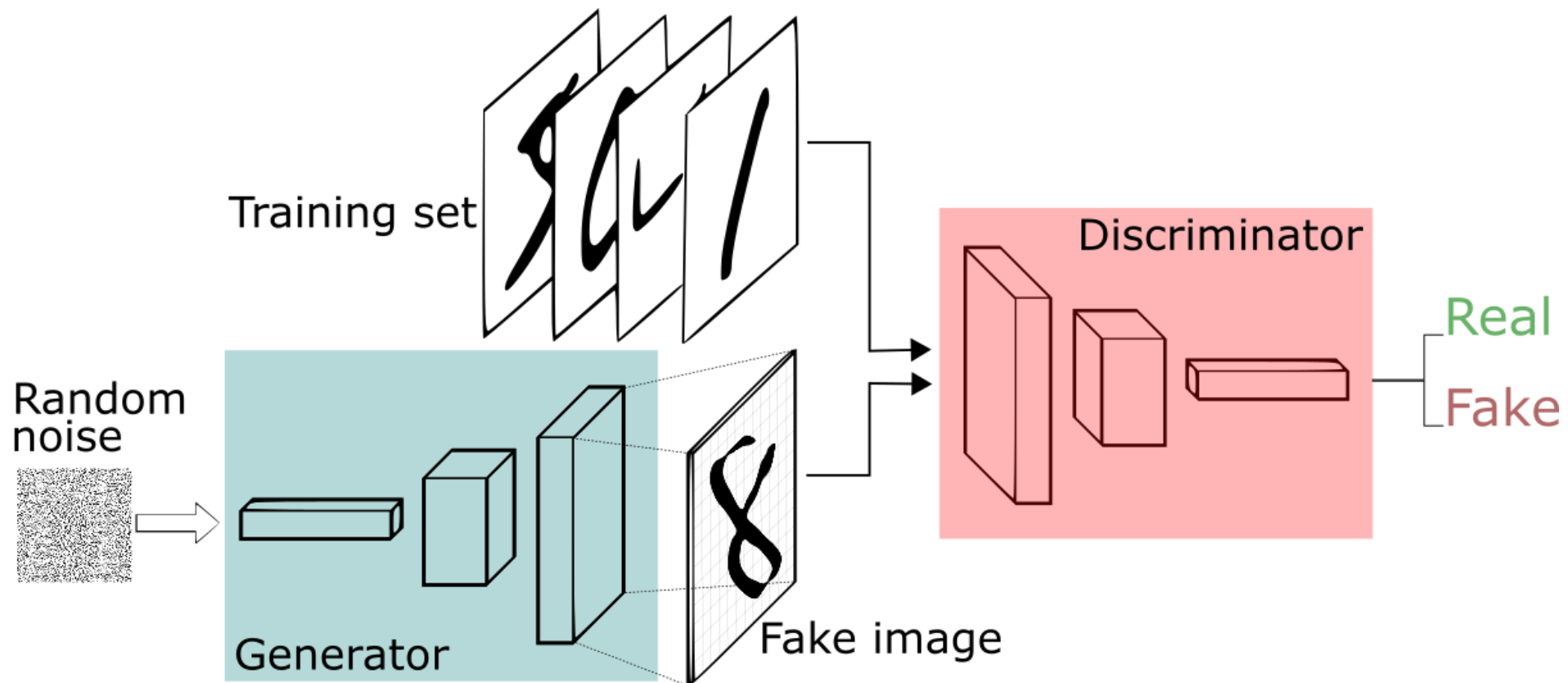


子过程质心系能量

运动学分析

由ANN得到解析公式？

其它网络：Generative Adversarial Network (GAN)



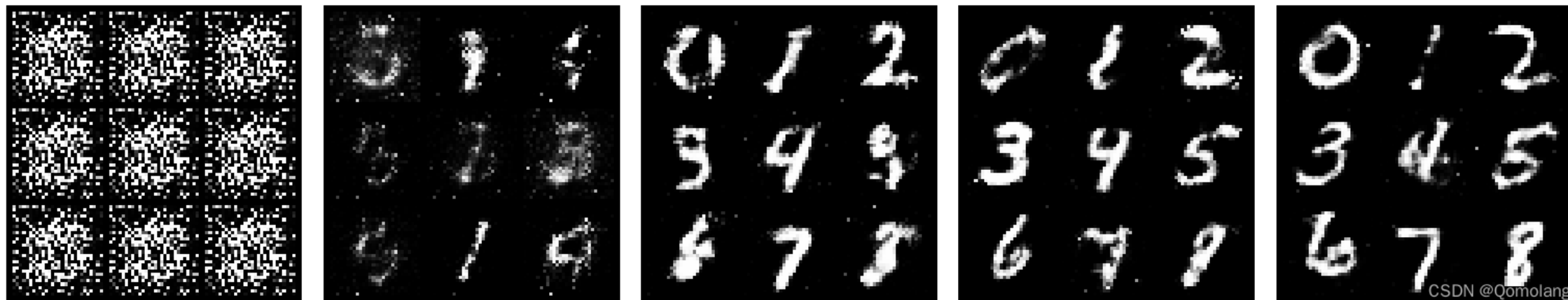
- GAN被用于的场景:
- 将照片变成漫画
- 换头照片, 换头视频



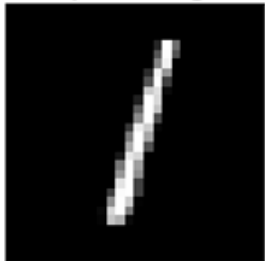
GAN手写数字生成

- 图片 -> distribution

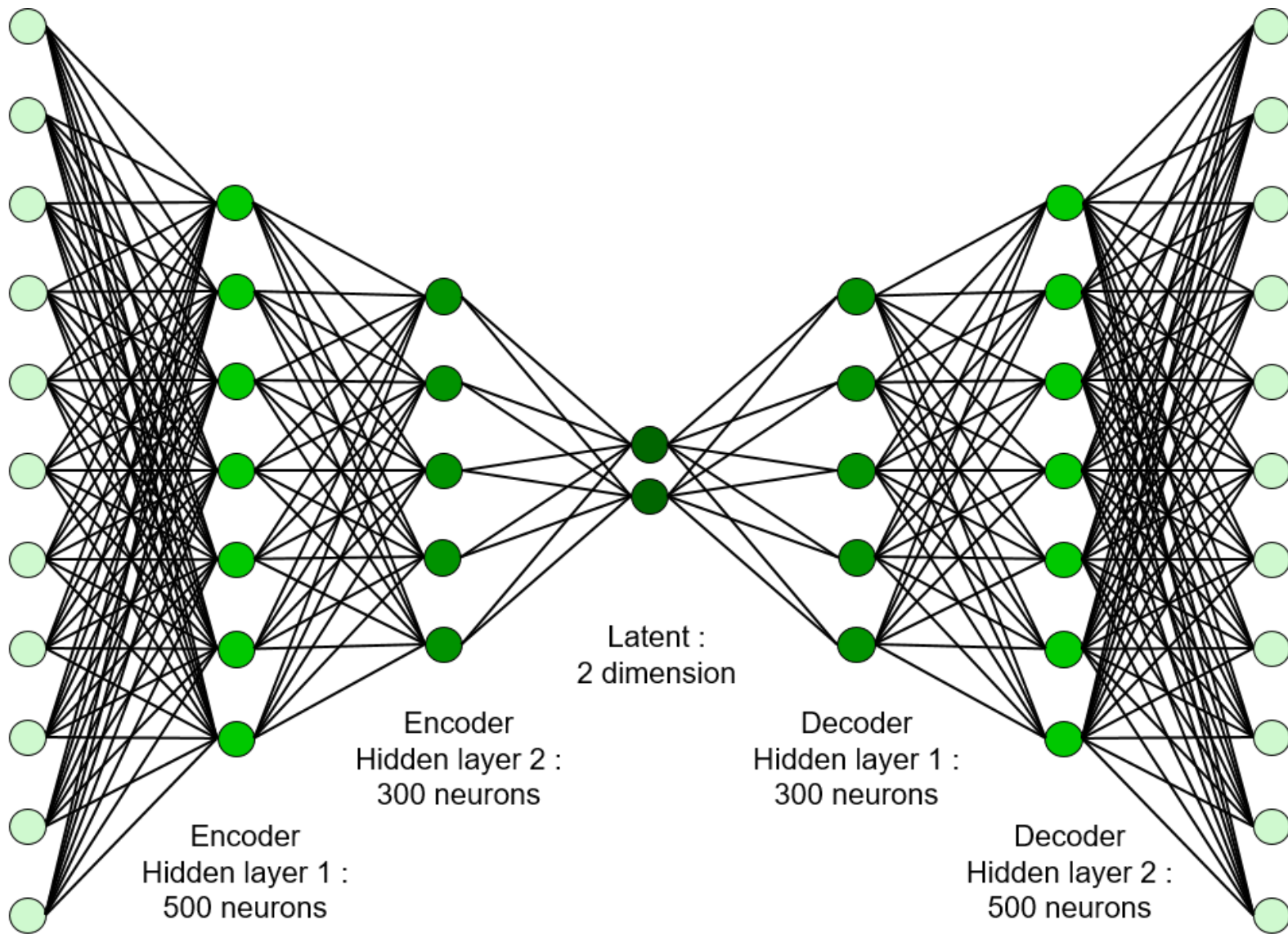
一层



Input Image



28 28



Input layer :
784 neurons

Reconstruct layer :
784 neurons

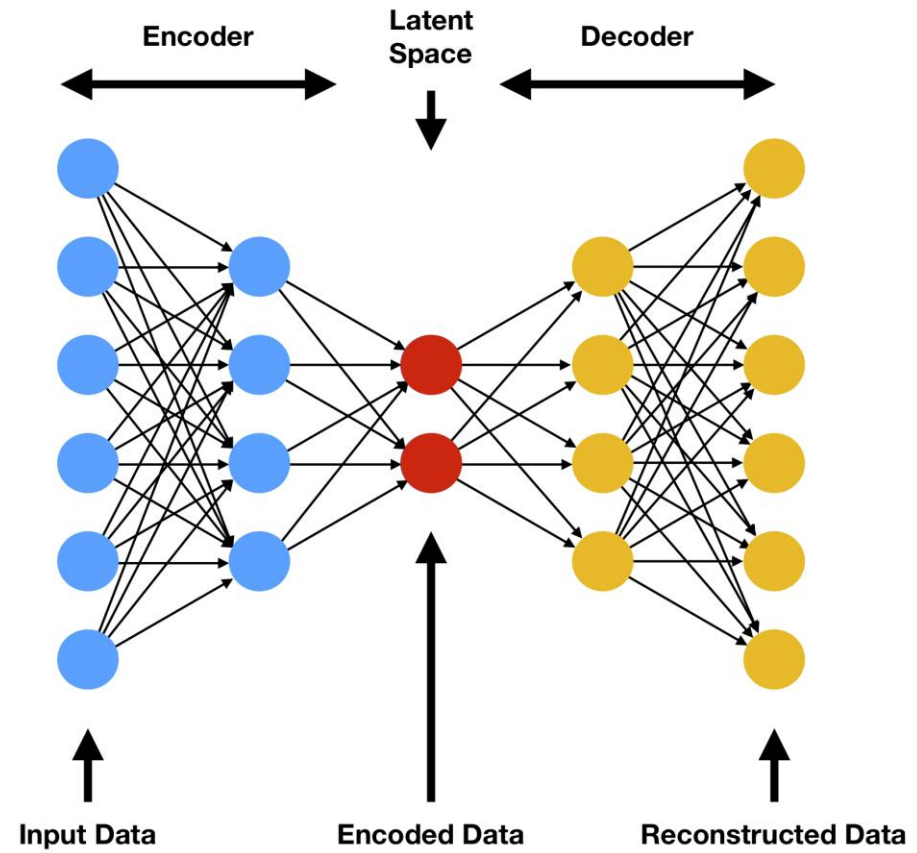


Reconstructed Image

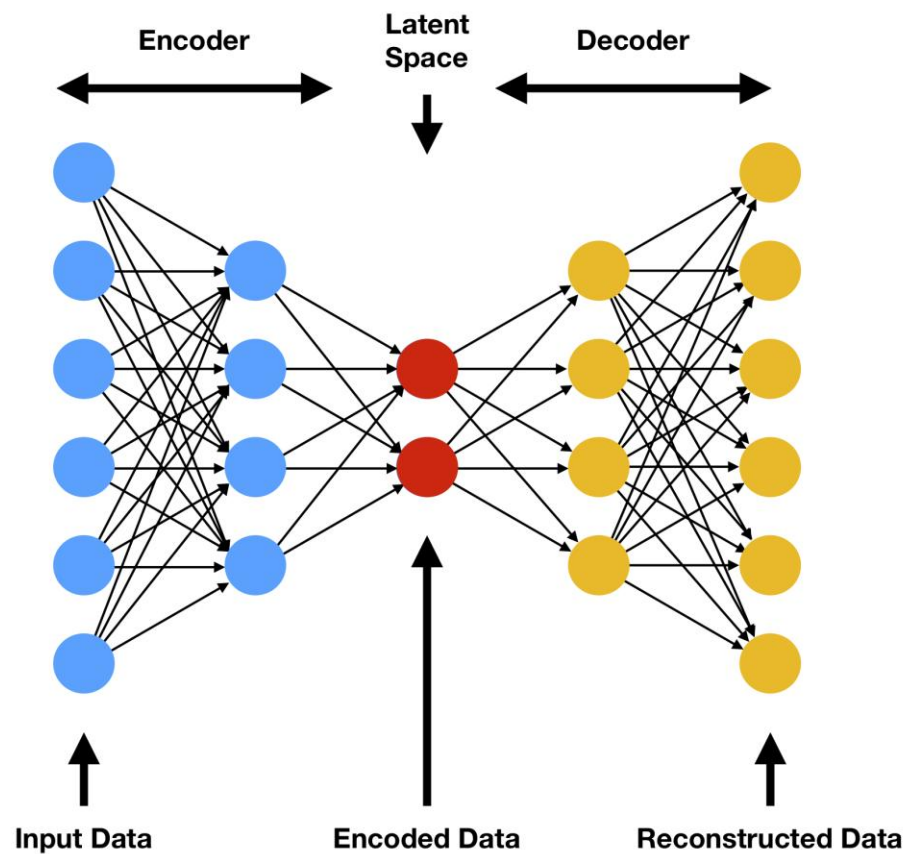


28 28

- Dimension reduction



- 什么叫latent space?
- 创造力
- 事例生成?
- 异常检测?



未涉及到但推荐了解的一些内容:

- 图神经网络
- Transformer
- U-Net, diffusion
- KAN
- ...

GPT, LLM

- Generative Pre-trained Transformer
- Large Language Model
- 就是把神经网络的规模变得极大（几百上千层），并且只喂给它海量的文本去训练。训练完成后，它变成了一个超级强大的“文字接龙高手”——你给它上文，它可以根据海量学来的规律，计算出概率最高的下文。

推理能力

- 语言模型本身自带逻辑连贯性
- 神经网络=仿生学
- 一个小游戏：我脑海里有一样东西，通过是/否问题，猜出来它是什么。

推理能力

- 一个小游戏：我脑海里有一样东西，通过是/否问题，猜出来它是什么。
- 一个小游戏：代码有个报错，通过调研代码，修改测试，猜出来正确的写法是什么。

智能体Agent

- 一套“感知-决策-执行-反思”的自主循环机制，让它能主动调用外部工具、读取环境反馈并据此修正下一步动作，从“被动回答”升级为“主动完成任务”。

Agent

- 自主规划：拆解步骤（context局限）
- 收集信息：LLM无状态（context压缩）
- 检验效果：工具调用

Agent

- Claude code
- Codex
- OpenCode
- Qorderwork
- 以太
- Kimi-code
- Zcode
- Mimo-code
- Openclaw
- Hermes
- Trae
- Cursor
- ...
- 调用API需要花钱



安装举例:

- Node.js
- cc switch
- Claude code

nvm-windows

- <https://github.com/coreybutler/nvm-windows/releases>

Contributors

 dependabot

▼ Assets 7

 nvm-noinstall.zip	6.01 MB	Jan 1, 2025
 nvm-noinstall.zip.checksum.txt	34 Bytes	Jan 1, 2025
 nvm-setup.exe	5.35 MB	Jan 1, 2025
 nvm-setup.zip	4.84 MB	Jan 1, 2025
 nvm-setup.zip.checksum.txt	34 Bytes	Jan 1, 2025
 Source code (zip)		Jan 1, 2025
 Source code (tar.gz)		Jan 1, 2025

 929  108  160  286  189  118 1k people reacted

- 安装完成会提示安装node.js的命令 (ubutun直接apt-get install)

cc - switch

- 配置API

▼ Assets 21

 CC-Switch-v3.16.5-Linux-arm64.ApplImage	sha256:e154c98ce4813a135...		84.8 MB	last week
 CC-Switch-v3.16.5-Linux-arm64.ApplImage.sig	sha256:8c56ec433563c091a...		424 Bytes	last week
 CC-Switch-v3.16.5-Linux-arm64.deb	sha256:e19bae7d649d90b7e...		11.4 MB	last week
 CC-Switch-v3.16.5-Linux-arm64.rpm	sha256:96dd5f4ad7ce9b0ea...		11.4 MB	last week
 CC-Switch-v3.16.5-Linux-x86_64.ApplImage	sha256:0de40fd51f5df67da...		87 MB	last week
 CC-Switch-v3.16.5-Linux-x86_64.ApplImage.sig	sha256:898138f23a4ac76fc...		424 Bytes	last week
 CC-Switch-v3.16.5-Linux-x86_64.deb	sha256:ba1c935d56f3d1bf4...		11.9 MB	last week
 CC-Switch-v3.16.5-Linux-x86_64.rpm	sha256:ee8c8a992cfa7ee62...		11.9 MB	last week
 CC-Switch-v3.16.5-macOS.dmg	sha256:1fc8b187bc7d1089e...		24.6 MB	last week
 CC-Switch-v3.16.5-macOS.tar.gz	sha256:109153d436592fb46...		25.2 MB	last week
 Source code (zip)				last week
 Source code (tar.gz)				last week
Show all 21 assets				

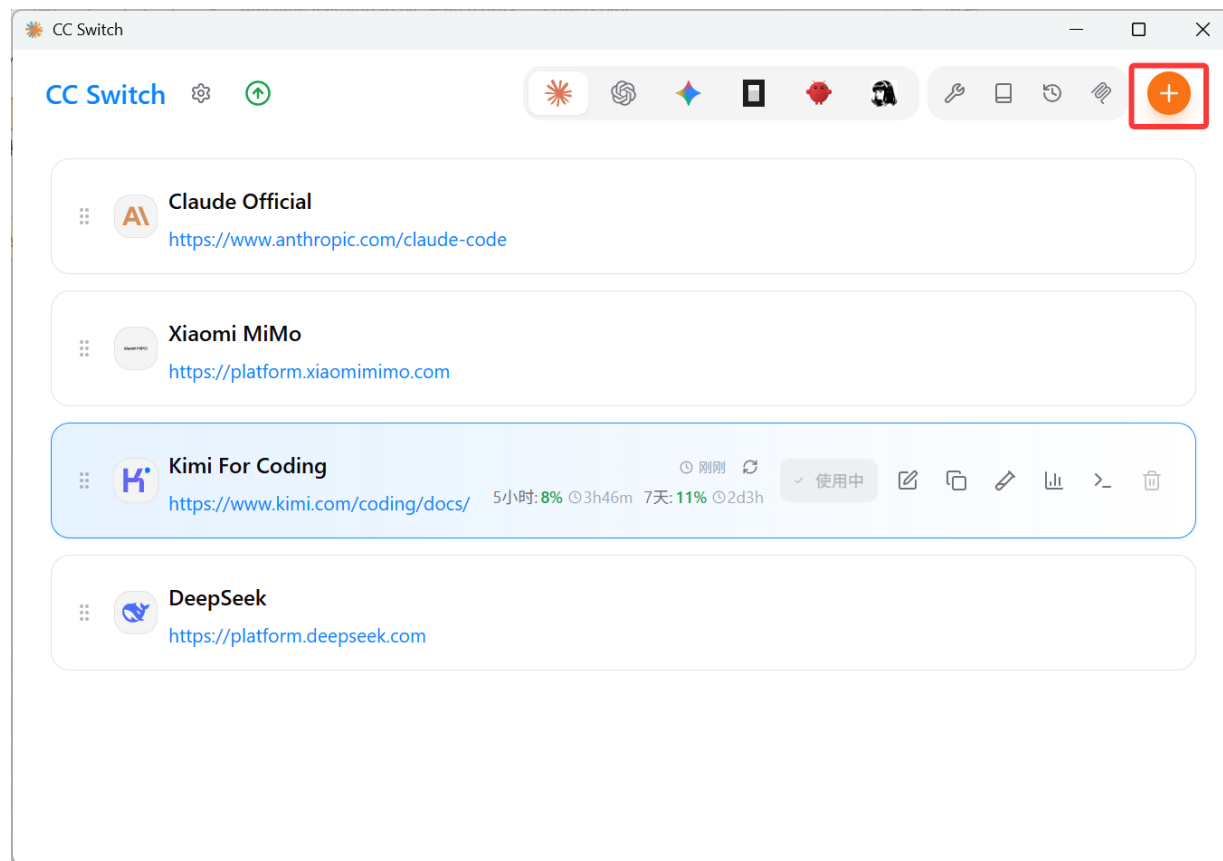
 141  18  17  8  11  7 160 people reacted

安装claude code

- `npm install @anthropic-ai/claude-code`

Anthropic接口

- cc – switch



- "hasCompletedOnboarding": true

我例子使用的

- Kimi code

K2.7 Code 正式版现已更新 [立刻体验](#)

Kimi Code

Kimi 会员计划中面向代码开发的权益，轻松兼容各类开发流程，
更快更稳响应请求，高效完成编程任务

安装 Kimi Code  `irm https://code.kimi.com/kimi-code/install.ps1` | ... 

一次订阅，全端畅写

Terminal

IDE

实机操作

- 1 - 领略一下Agent的威力
- 2 - 使用Agent配合MLAnalysis
- 3 - 安装和使用MLAnalysis skill

实机操作1： 泡泡

- 考虑带有三次项的 Higgs (ϕ^4) 理论，忽略规范场和费米子，固定四次项系数为 1。在有限温度下进行格点蒙特卡罗模拟。选取二次项和三次项系数使得经典势能出现两个非简并极小，较高极小为亚稳态（假真空），较低为真真空。从假真空均匀构型初始化，观察热涨落导致的一阶相变泡泡成核与演化。
 - (1) 展示典型泡泡构型图。
 - (2) 调整二次、三次项系数，观察泡泡形态随参数的变化。
 - (3) 构思一套用来描述泡泡的指标。
 - (4) 写一份 manuscript 草稿（英文, revtex），包含拉氏量公式、带泡泡的构型图、不同系数下的构型对比图，以及你选定的定量指标随系数的变化规律。
- 使用本地 venv 创建环境，本地 git 管理代码，本地 pdflatex 撰写。

实机操作2：截面计算

- 我电脑里有Mathematica, FeynCalc, FeynArts和nTGC的FeynArts模型文件。计算带nTGC贡献的, $e^+ e^- \rightarrow Z Z Z$ 的截面公式, 将SM, NP和干涉项分开写, 将结果写进一个manuscript (英文, revtex), 带费曼图的。注意: 末态3个粒子的相空间积分非常耗时, 不用计算, 只需要给出微分散射截面 (也就是矩阵元平方) 即可。相空间积分可采用数值积分 (CEPC能量, 单位fb)。提示: 先确定有几个图, 每个图的amplitude分开写 (例如A1,A2,...), 且确定它们的归属 (SM还是NP), 注意是Particle level的。然后分别计算每个图的平方, 和一个一个的干涉项A1A2, A1A3, ..., 然后分类 (纯SM, 纯NP, SM和NP的干涉) 加到一起, 必须给出求和后的显式解析结果。最后使用数值积分进行相空间积分, 给出截面的数值结果。

实机操作3: conway's life game

- 实现周期边界条件的Conway生命游戏，考虑这么一件事情：如果把稳定的Conway生命游戏的pattern看作时间周期格子configuration，那么可以类比定义spectrum。使用6*6格子，随机初态进行模拟，对比普通的Torus周期边界条件，K2 (Kelien bottle), P2 (projective plane)，其谱学是否有差异？使用本地的venv创建环境，本地的git管理代码，本地的pdflatex撰写论文（中文, revtex）。在论文中需要画图介绍三种边界条件，以及展示结果。注意：不要把谱学理解为对configuration的傅里叶变换，我这里专指周期变化pattern的频率（1/周期）的分布。
- 寻找横跨边界的例子（例如滑翔机），用gif动画展示不同边界条件下的区别。

实机操作4: kmeans

- 使用本地的venv管理环境，自己写一个kmeans代码（不要用已有的例如scikit-learn），展示kmeans的每一步，并制作成slide（本地有pdflatex）。

实机操作5: SVM NP

- 在文件夹里有sm-mssm.lhe和np-mssm.lhe两个数据文件，分别代表标准模型和新物理的事例，将 <https://gitee.com/NBAlexis/CutExperiment.git>克隆到本文件夹里使用，用它来把sm-mssm.lhe和np-mssm.lhe转成csv文件：你自己查看末态粒子是什么，施加对粒子数的cut后，将末态粒子按种类和能量排序，一个事例对应末态粒子四动量的拼接的矢量。然后，使用SVM算法训练超平面，以每个事例到超平面的距离（带正负的）作为score，绘制一个SM/NP的histogram图（PDF格式），使用本地PDF软件直接打开图。

实机操作6: ANN NP

- 在文件夹里有SM-5000.lhco和btb2-5000-NP.lhco两个数据文件，分别代表标准模型和新物理的事例，将<https://gitee.com/NBAlexis/CutExperiment.git>克隆到本文件夹里使用，用它来把lhco的数据转成csv文件：你自己查看末态粒子是什么，施加对粒子数的cut后，将末态粒子按种类和能量排序，一个事例对应末态粒子四动量的拼接的矢量。然后，使用简单的全连接ANN（自己的venv里安装pytorch）训练神经网络，给每个事例打分，绘制一个SM/NP的分数histogram图（PDF格式），使用本地PDF软件直接打开图。然后，btb2-5000-0.lhco到btb2-5000-10.lhco是同时有SM和NP（包括干涉项）的事例，用你训练的神经网络给它们打分，并用分数施加cut，压低背景，凸显信号。btb2-5000-0.lhco到btb2-5000-10.lhco是不同NP系数下的结果，以系数为横坐标，以截面为纵坐标，计算cut后的截面，并画图。原始截面和系数，在btb2-5000-n-banner.txt里。
- 假定亮度为 10 ab^{-1} ，那么signal significant = 2,3,5的预期系数约束分别是多少？

实机操作7: MLAnalysis skill

- **Skill**, 就是把人类的知识和经验, 用自然语言写成一份 AI 可以直接执行的工作手册。
- <https://gitee.com/NBAlexis/automated-smeft-ml-pheno.git>
- Kimi-code:
- .kimi-code/skills
- Claude code:
- .claude/skills
- 其它的: 可以直接询问。例如max-code(minimax), 自然语言询问安装skill的位置。

- 谢谢