

1.机器学习基础~~找到函数

笔记本： 【课】原理-李宏毅 deep learning

创建时间： 2023/3/18 11:09

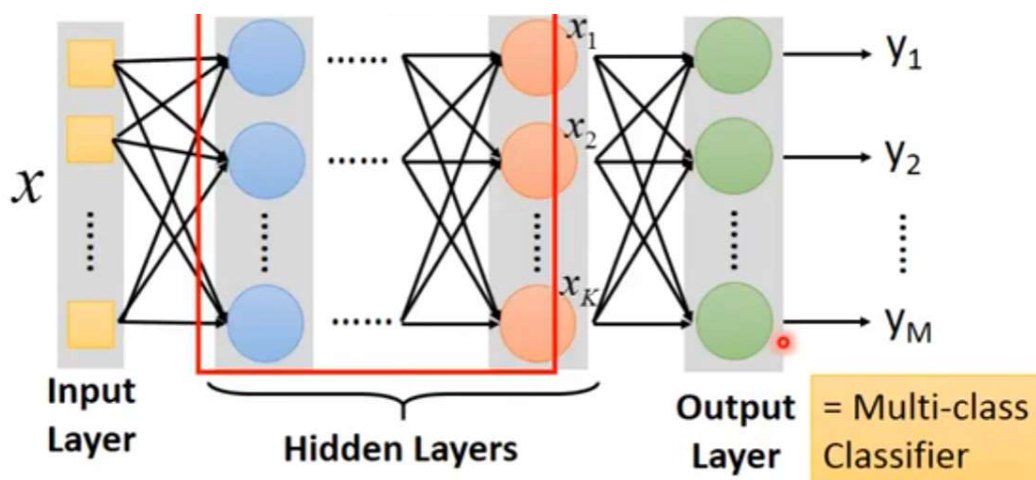
更新时间： 2023/4/21 16:38

作者： 1256876216@qq.com

URL: <https://fanyi.baidu.com/#zh/en/%E4%B8%80%E7%9B%B4%E5%9D%9A%E6...>

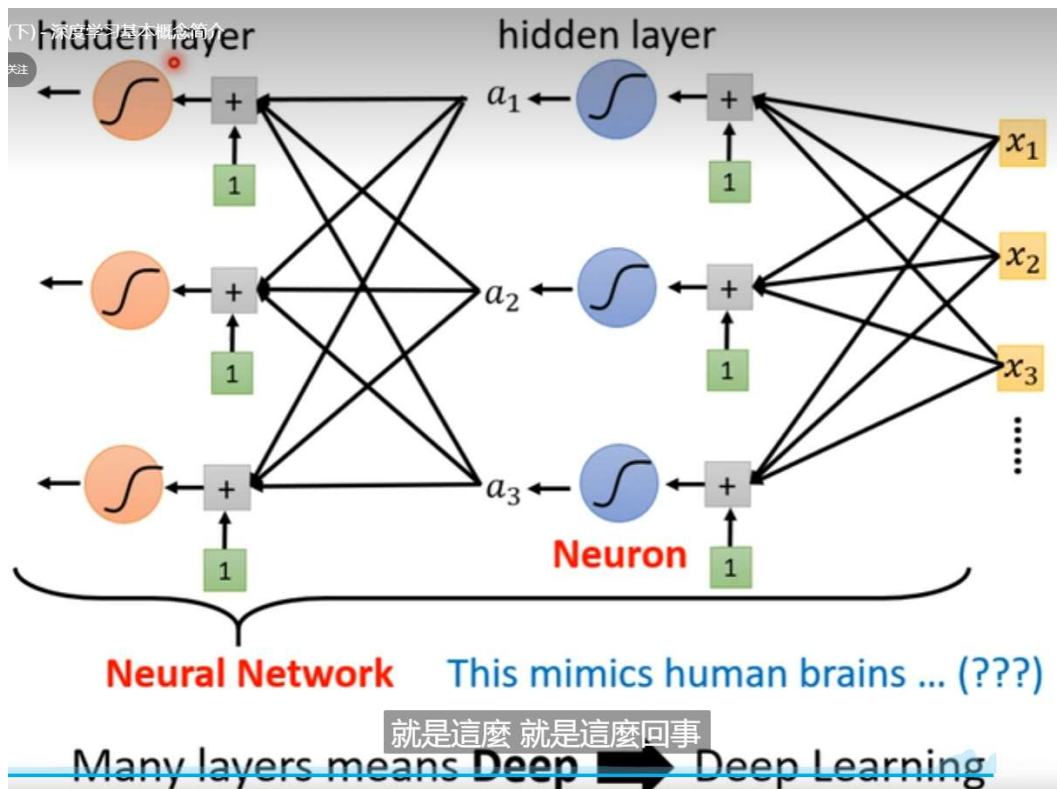
一、机器学习分类：

- 有监督 supervised leaning 自监督self-suoervised leaning
- 无监督： unsupervised leaning
- 强化学习： reinforcement leaning RL



二、预训练方法”

- pre-train 预训练：训练出一个基础模型foundation model____bert预训练模型 340兆参数 (nlp自然语言处理模型)
- fine-tune：模型调优 在pre-train的模型上加数据，不用从头开始训练
- downstream Task：下游任务



三、任务分类——by不同类别的处理函数

1. regression : 回归
 1. 输出是连续型数值
2. classification: 分类
 1. 输出离散型数值。预设的值 AlphaGo围棋 落子19*19格子
3. structured Learning: 生成有机物的物件 (例如: 影响、问句)
 1. 又叫生成式学习 generative Learning
 2. 两种策略: 各个击破、一次到位

参数和超参数:

- 参数: 机器训练过程中自己找出的函数参数
- 超参数: 人为预先设定的参数
 - 例如学习率

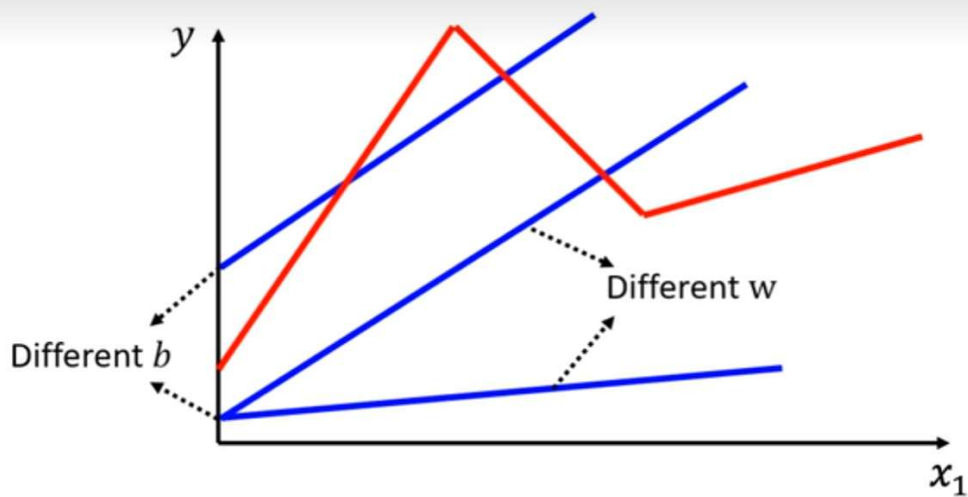
设定范围: 神经网络的结构model: 类神经网络结构 (CNN、RNN、Transformer等) 、决策树

设定标准: 监督学习、半监督学习、强化学习等

达成目标: 梯度下降、遗传算法等

linear Model线性模型

Linear models are too simple ... we need more ^{ambiguous 没一个拼对的} sophisticated modes.

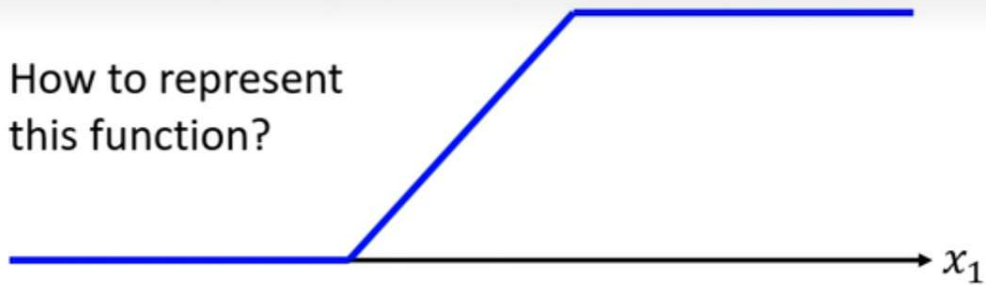


Linear models have severe limitation. **Model Bias**

red curve = constant + sum of a set of



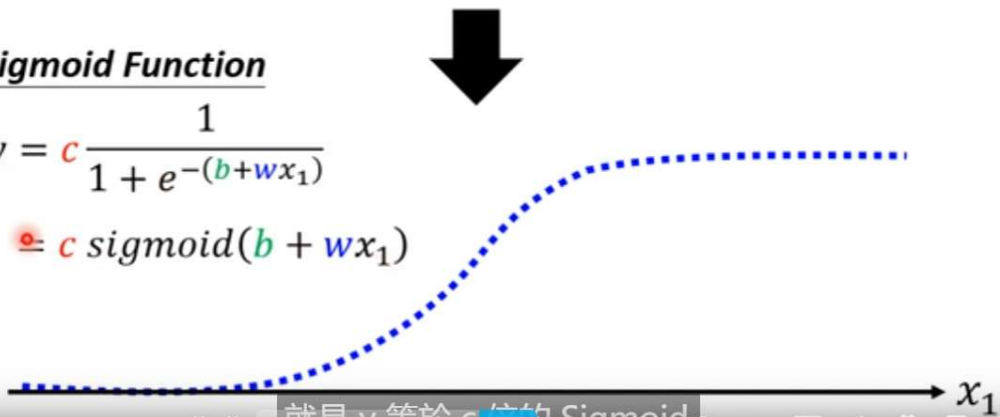
How to represent this function?



Sigmoid Function

$$y = c \frac{1}{1 + e^{-(b+wx_1)}}$$

$$\hat{=} c \text{ sigmoid}(b + wx_1)$$



sigmioid 拟合

激活函数 activation function :

Sigmoid → ReLU

$$y = b + \sum_i c_i \text{sigmoid} \left(b_i + \sum_j w_{ij} x_j \right)$$

Activation function

$$y = b + \sum_{2i} c_i \max \left(0, b_i + \sum_j w_{ij} x_j \right)$$

过拟合 overfitting: 训练集上表现号, 测试集上表现差

3 layer	4 layer
0.14k	0.10k
0.38k	0.44k

on unseen data



四、建模步骤:

1、定义函数的算式: Model (带有位置参数的函数): $y = b + wX1$

- w: weight 权重
- b: bias 偏差
- y: 标签 label (输出的结果)
- X1: feature 特征 (输入参数)

2、定义 LOSS (损失): 也是函数 $L(b, w)$

- 输出: 对b和w参数的函数, 来判别这组变量的好坏
- 每一项差值 $e = |y - y^{\wedge}|$ (其中一种) e: error y实际值 y^{\wedge} 预测值 (y hat): 分类会根据不同人物进行选择差值计算公式
 - $e = |y - y^{\wedge}|$: 平均绝对误差MAE (mean absolute error)
 - $e = (y - y^{\wedge})^2$: 平均平方误差 MSE (mean square error) 一般在异常值敏感时
- LOSS: $L = 1/N$ (e 求和) 训练数据真实标签与预测标签的差值的平均值: 越小代表模型效果越好

3、Optimization 最优化 找出能让 LOSS最小的参数 w 和 b, $w^*, b^* = \arg \min L$

- gradient Descent 梯度下降方法 (唯一)
- 控制单一变量, w-L对应曲线上, 随机选择某一点为初始值W0, 计算该点切线斜率是 $\partial L / \partial w$, 向 L小的方向跨出一步
- 步伐大小: 变量姨塔 * 斜率 $|w=w0$ (变量姨塔: 学习率) 学习率是超参数
- 局部最优local minima和全局最优global minima 通过上面步调可能出现局部最优不是全局最优的情况

$$\eta \frac{\partial L}{\partial w} |_{w=w^0}$$

Optimization of New Model

$$\theta^* = \arg \min_{\theta} L$$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \end{bmatrix}$$

➤ (Randomly) Pick initial values θ^0

$$\mathbf{g} = \begin{bmatrix} \frac{\partial L}{\partial \theta_1} |_{\theta=\theta^0} \\ \frac{\partial L}{\partial \theta_2} |_{\theta=\theta^0} \\ \vdots \end{bmatrix} \quad \begin{bmatrix} \theta_1^1 \\ \theta_2^1 \\ \vdots \end{bmatrix} \leftarrow \begin{bmatrix} \theta_1^0 \\ \theta_2^0 \\ \vdots \end{bmatrix} - \begin{bmatrix} \eta \frac{\partial L}{\partial \theta_1} |_{\theta=\theta^0} \\ \eta \frac{\partial L}{\partial \theta_2} |_{\theta=\theta^0} \\ \vdots \end{bmatrix}$$

$$\mathbf{g} = \nabla L(\theta^0) \quad \text{那假設你這邊參數有1000個} \quad - \eta \mathbf{g}$$

五、chatGPT:

g: Generative
p: Pre-trained
t: Transformer

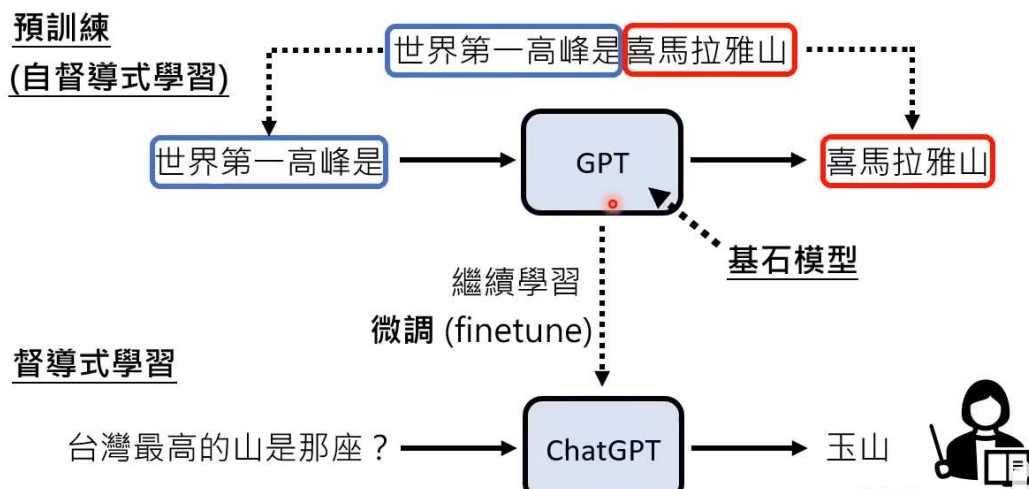
强化学习的文本学习模型 && 有监督的打分模型相互作用（交替训练），最开始需要人工标注

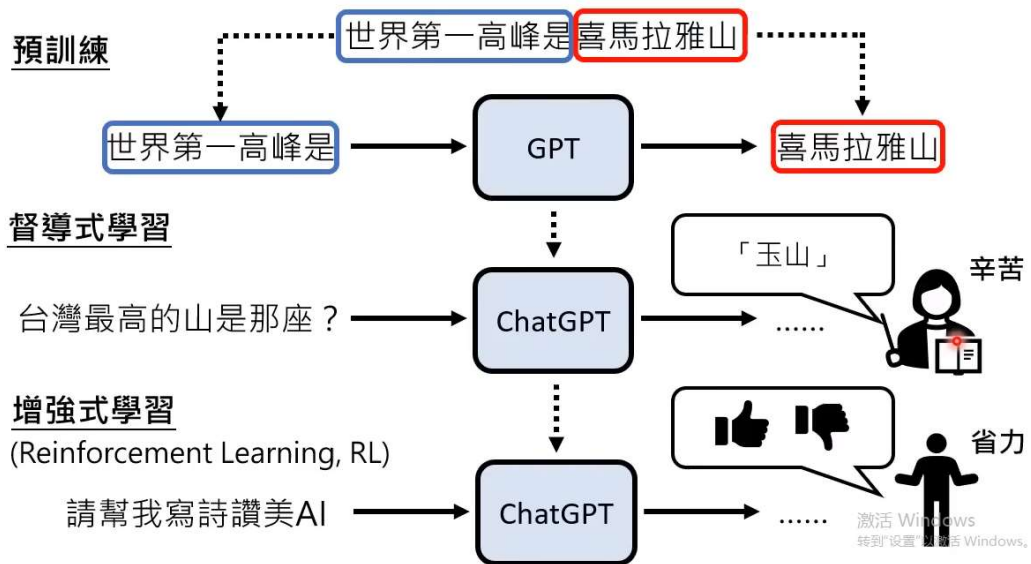
打分模型的得分目标：得分-差异（生成模型（强化学习）与 打分模型（有监督模型）的概率分布的比值的对数（差异）越小越好）+ 在其他下游任务的模型泛化表现

chatGpt分类:

对于用户而言是生成式的形式，实际为把生成式学习拆解成多个分类问题

参数
shuffle





Backpropagation

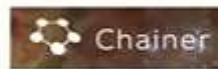
- Backpropagation: an efficient way to compute $\partial L / \partial w$ in neural network



theano

Caffe

Microsoft
CNTK



libdnn
台大周伯威
同學開發

Ref:

http://speech.ee.ntu.edu.tw/~tlkagk/courses/MLDS_2015_2/Lecture/DNN%20backprop.ecm.mp4/index.html

Created with EverCam
<http://www.camdemy.com>