

# 笔记

---

## 任务重点概念与基本思想

---

神经网络的终极目的是实现类人工智能的机器学习

计算机有快速执行基本命令的能力，人类有智能。

迭代：持续细化误差值，持续地，一点一点地改进答案（神经网络学习中的核心过程）。

算法：用一系列的计算机指令来达到某个目标。

布尔逻辑函数：就是or一真即真，and一假即假,xor（异或）相同为假不同为真。

pytorch:一个以Python优先的深度学习框架，不仅能够实现强大的GPU加速，同时还支持动态神经网络。

---

**监督学习：**同时带有输入x和输出标签y，关键在你给予了正确的学习算法示例

回归算法：回归方程，通过拟合直线或曲线来预测值

线性回归，很基础，不再赘述。

多元线性回归，这是task1所用到的算法。

分类算法：例如将客户群体细分，分到不同的需求方向

**无监督学习：**仅含有输入x，但无输出标签y，未给出正确示例

聚类算法：如对文章进行自动分类

异常检测算法：检测异常的算法，比如网络质量异常、用户访问行为异常、服务器异常、交换机异常和系统异常等

降维算法：对一个大的数据集尽可能损失小地对其进行压缩。

---

**预测器：**机器接受一个输入，并作出应有的预测，输出结果，被称为预测器。

**分类器：**基于已知的不同种类内容，划定一个界限，用来给未知种类的内容分类（假定种类是有限且已知的）。

多个分类器一起工作是神经网络的核心思想。

---

成本函数（代价函数）：残差平方和除以二倍个数。越小表示拟合程度越高

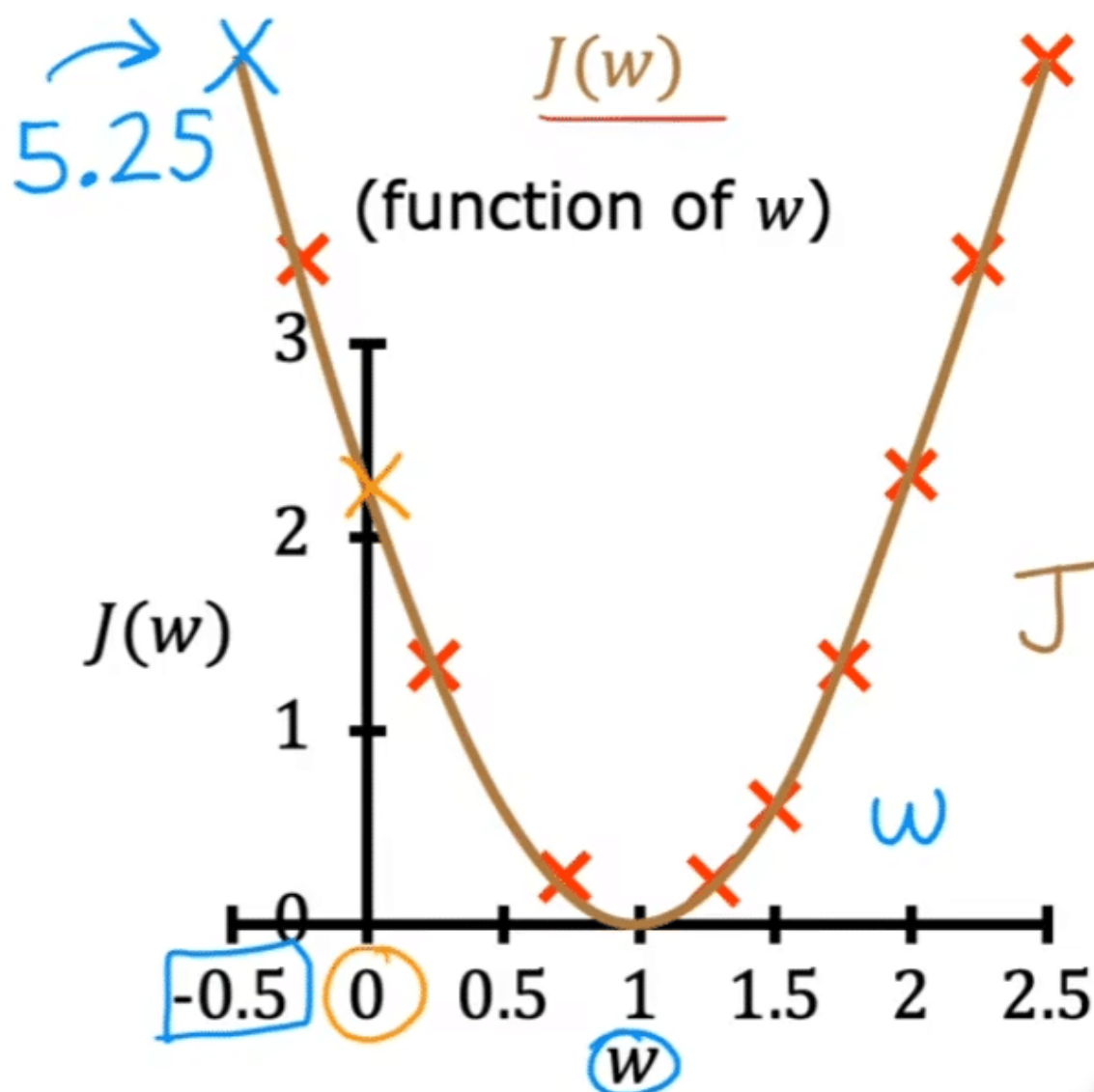
Model  $f_{w,b}(x) = wx + b$

Parameters  $w, b$

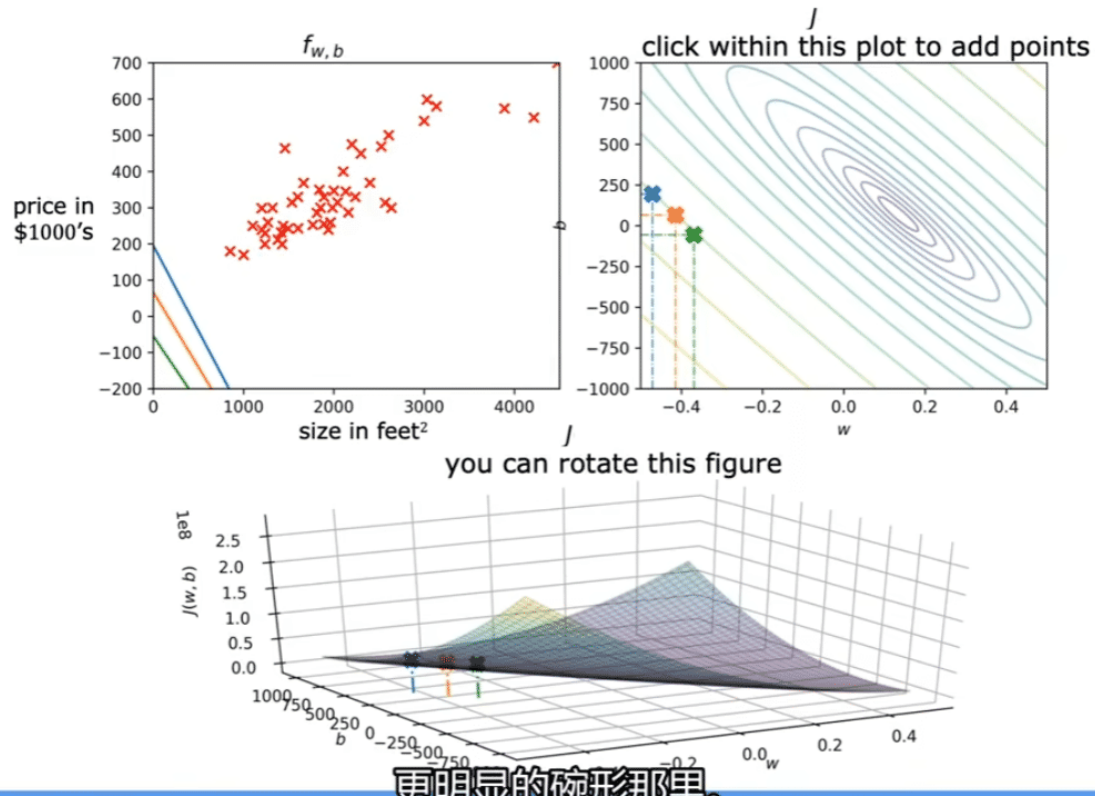
Cost Function  $J(w, b) = \frac{1}{2m} \sum_{i=1}^m (f_{w,b}(x^{(i)}) - y^{(i)})^2$

Objective  $\underset{w,b}{\text{minimize}} J(w, b)$

二维代价函数图像:



三维代价函数图像：



透过图像找最低点，也就是拟合最好的一组变量。

**独热码：**有多少个状态就有多少比特，而且只有一个比特为1，其他全为0的一种码制

**向量化（矢量化）：**

优点：代码更短，速度更快，因为矢量化后每一组数据在平行处理

## Vectorization

$$f_{\vec{w}, b}(\vec{x}) = \vec{w} \cdot \vec{x} + b$$

```
f = np.dot(w, x) + b
```

向前传播：信号是前向传播的，而误差是反向传播的

反向传播：通过反向传播把误差传播到每一层，然后整权重w

---

**神经元：**神经网络的基本单元，神经元的输出需要使用激活函数。

**输入层：**为数据特征输入层，输入数据特征个数就对应着网络的神经元数。

**隐藏层：**即网络的中间层，隐藏层层数可以为0或者很多层，其作用接受前一层网络输出作为当前的输入值，并计算输出当前结果到下一层。隐藏层是神经网络性能的关键，通常由含激活函数的神经元组成，以进一步加工出高层次抽象的特征，以增强网络的非线性表达。隐藏网络层数直接影响模型的拟合效果。

**输出层：**最终结果输出的网络层。输出层的神经元个数代表了分类标签的个数（注：在做二分类时，如果输出层的激活函数采用sigmoid，输出层的神经元个数为1个；如果采用softmax分类器，输出层神经元个数为2个）

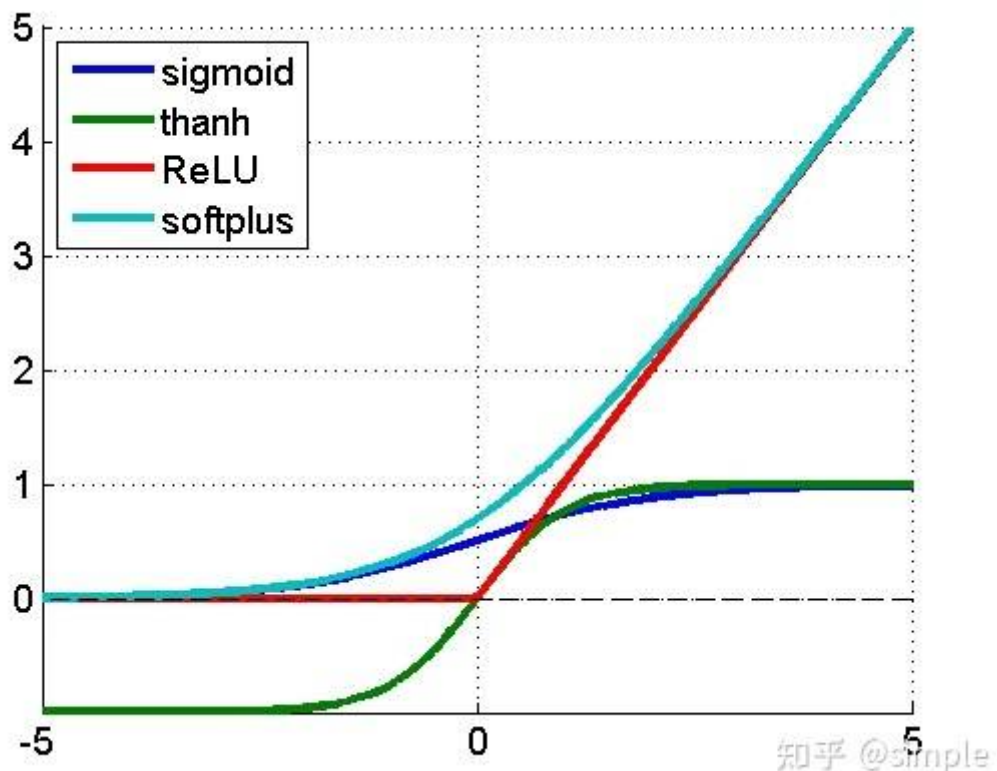
**卷积：**物理意义可以是系统某一时刻的输出是由多个输入共同作用（叠加）的结果。

---

**激活函数：**并不是去激活什么，而是指如何把“激活的神经元的特征”通过函数把特征保留并映射出来，即负责将神经元的输入映射到输出端。

种类：sigmoid函数，tanh函数，ReLU函数

ReLU函数表达式： $f(x) = \max(0, x)$



**权重：**指某一因素或指标相对于某一事物的重要程度，在神经网络中作用于输入->隐藏

,隐藏->隐藏 过程中。

**权重更新**：通过算法自动改变权重，达到更好的拟合效果。

---

**梯度**：在一个点上找到极小的附近的最低的一个点，移动到那个点称为梯度下降。并在新的点上重复以上动作，便找到局部最低点,称为批量梯度下降。

特性：不同的起点可能会导向不同的局部最低点

梯度下降算法：

$$tmp\_w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$
$$tmp\_b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

$w = tmp\_w$

$b = tmp\_b$

直到"收敛"，指w,b的值变化较小

"d/dw"在这里指偏导数，"a"指学习率

**学习率**：大小介于1和0，具体的数值可以决定梯度下降是降低多少。

过小因步骤多而速度慢，过大甚至会导致梯度下降不起作用。

**损失函数**：度量模型的预测值f(x)与真实值Y的差异程度的运算函数，损失函数越小，模型的鲁棒性（指控制系统在一定参数摄动下，维持其它某些性能的特性）就越好。

**过拟合**：过拟合就是训练样本得到的输出和期望输出基本一致，但是测试样本的输出和测试样本的期望输出相差却很大。当某个模型过度的学习训练数据中的细节和噪音，以至于模型在新的数据上表现很差，我们称过拟合发生了。模型泛化性能的变差。

---

**训练集**：形如此形,用于训练神经网络

## Terminology

Training Data used to train the model set:

size in feet <sup>2</sup>	price in \$1000's
2104	400
1416	232
1534	315
852	178
...	...
3210	870

Notation:

$x$  = "input" variable  
feature

$y$  = "output" variable  
"target" variable

$m$  = number of training examples

$(x, y)$  = single training example

$(x^{(i)}, y^{(i)})$

$(x^{(i)}, y^{(i)})$  =  $i^{\text{th}}$  training example

index

(1<sup>st</sup>, 2<sup>nd</sup>, 3<sup>rd</sup> ...)



**测试集：**用于检测神经网络的准确与否

**神经网络性能评价指标：**常见的包括误差、准确率(accuracy)、R2 score

---