

深度学习

监督学习和非监督学习

监督学习

利用训练数据集学习一个模型，再用模型对测试样本集进行预测。

有无预期输出是**监督学习 (supervised learning)** 与**非监督学习 (unsupervised learning)** 的区别。

分类问题 (离散) 与**回归问题** (连续) 等都是监督学习。

非监督学习

为直接对数据进行建模。没有给定事先标记过的训练范例，所用的数据没有属性或标签这一概念。事先不知道输入数据对应的输出结果是什么。

如 **聚类算法**

二元分类

例：判定图像是否为猫

例如将矩阵中的像素值展开为一个向量 x 作为算法的输入

$$x = \begin{bmatrix} 255 \\ 231 \\ \vdots \\ 194 \\ 255 \end{bmatrix}$$

n_x 表示特征向量 x 的维度，有时简化为 n

Notation

单个样本由一对 (x, y) 表示

其中， x 是一个 n_x 维的特征向量， y 是一个标签 $y \in \{1, 0\}$

训练集包含 m 个训练样本

$$M_{test} = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$$

$$X = [x^{(1)} \ x^{(2)} \ \dots \ x^{(m)}]$$

X 是一个 $n_x * m$ 的矩阵

$$Y = [y^{(1)} \ y^{(2)} \ \dots \ y^{(m)}]$$

逻辑回归

给定的输入特征向量 x 和一幅图片对应我们希望识别这是否是一张猫的图片

因此我们想要一种算法能输出一个预测值 $\hat{y} = P(y = 1|x)$

约定逻辑回归的参数是 w, b

$$x \in R^{n_x}, w \in R^{n_x}, b \in R$$

Output:

$$\hat{y} = \sigma(w^T x + b)$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

总结一下就是

$$\hat{y}^{(i)} = \sigma(w^T x + b), \text{ where } \sigma(z) = \frac{1}{1 + e^{-z}}$$

Given $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$, want $\hat{y}^{(i)} \approx y^{(i)}$

Loss(error) function

平方误差

$$L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$$

损失函数

$$L(\hat{y}, y) = -(y \log \hat{y} + (1 - y) \log(1 - \hat{y}))$$

Cost Fuction

$$J(w, b) = \frac{1}{m} \sum_{i=1}^m L(\hat{y}^{(i)}, y^{(i)})$$

梯度下降法

Repeat

$$w := w - \alpha \frac{dJ(w)}{dw}$$

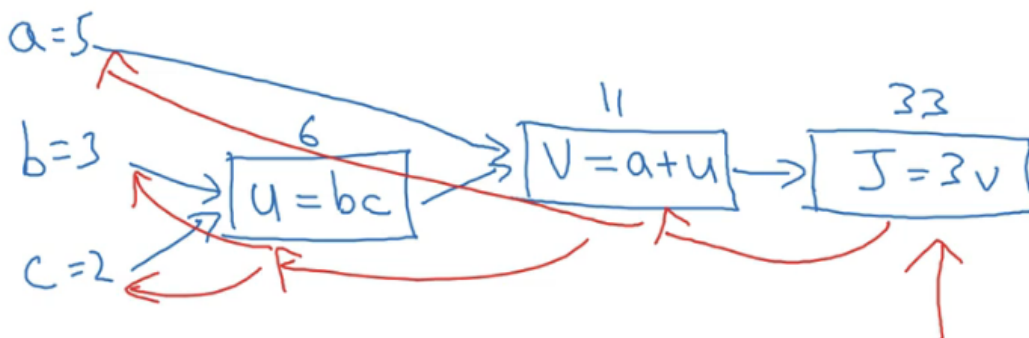
计算图

$$J(a, b, c) = 3(a + bc)$$

$$u = bc$$

$$v = a + u$$

$$J = 3v$$



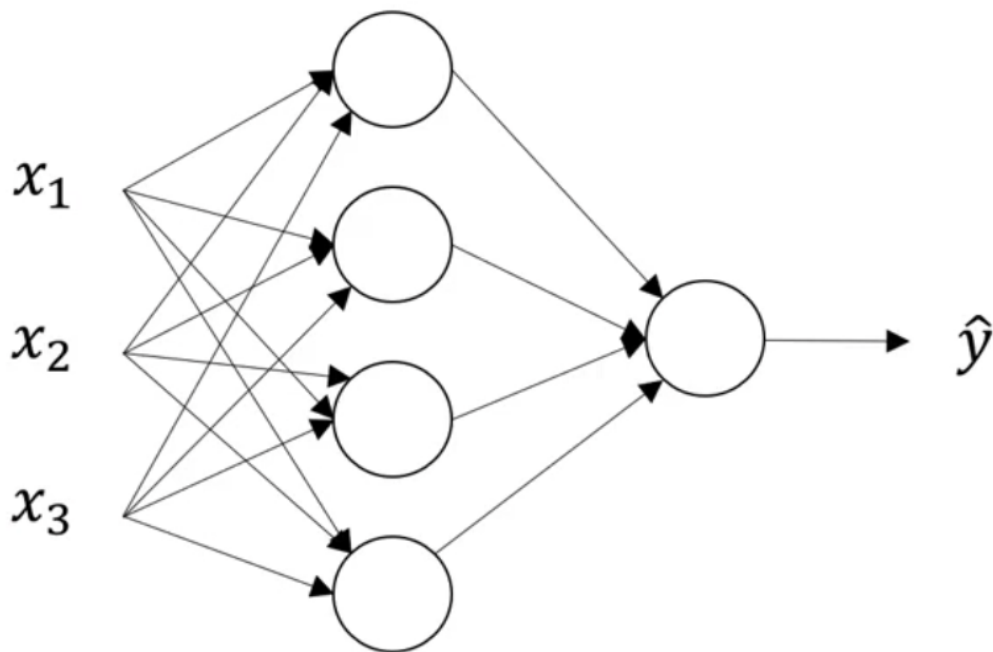
向量化

避免显式 for 循环

通过矩阵运算加速

$$\begin{aligned} J &= 0, \quad dw_1 = 0, \quad dw_2 = 0, \quad db = 0 \\ \text{for } i &= 1 \text{ to } m : \\ z^{(i)} &= w^T x^{(i)} + b \\ a^{(i)} &= \sigma(z^{(i)}) \\ J &+ = - \left[y^{(i)} \log a^{(i)} + (1 - y^{(i)}) \log (1 - a^{(i)}) \right] \\ dz^{(i)} &= a^{(i)} - y^{(i)} \\ \begin{bmatrix} dw_1 + = x_1^{(i)} dz^{(i)} \\ dw_2 + = x_2^{(i)} dz^{(i)} \end{bmatrix} & \quad dwt = x^{(i)} * dz^{(i)} \\ db &+ = dz^{(i)} \\ J &= J/m, \quad dw_1 = dw_1/m, \quad dw_2 = dw_2/m \\ db &= db/m \end{aligned}$$

神经网络



Input Layer

Hidden Layer

Output Layer

激活函数

在神经网络正向传播中

$$a = \sigma(z) = \frac{1}{1 + e^{-z}}$$
$$a = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

ReLU:

$$a = \max(0, z)$$

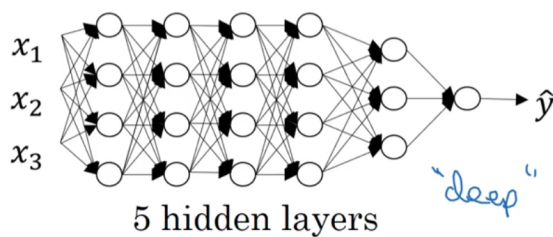
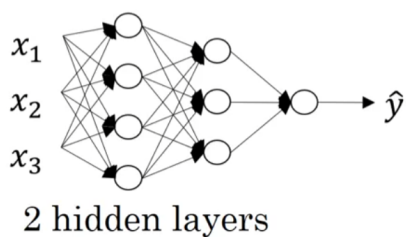
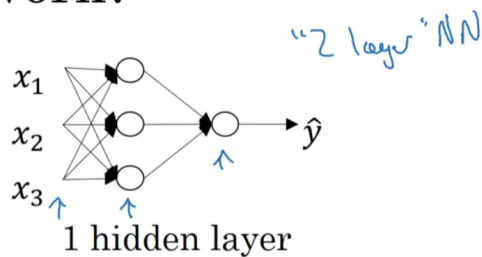
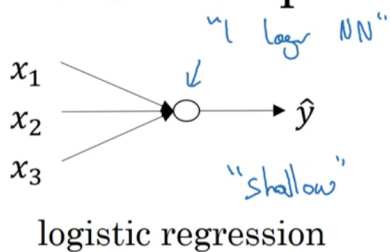
激活函数的导数

$$\sigma'(z) = \sigma(z)(1 - \sigma(z))$$

$$\tanh'(z) = 1 - \tanh^2(z)$$

深度神经网络

What is a deep neural network?



Andrew Ng

有些函数只能通过非常深层的神经网络能够学习

例如人脸识别

第一层 边框探测

第二层 面部器官探测

以此类推，每层都将上层作为输出层