


1

2

3

4

5



10个类别

28*28= 784

全连接层

x1
x2
x3
.
.
.
.
x_784

输出1 120

输出2 30

784*10

784

输出10 50

10

样本 --> 1 one_hot 10个类别

公式:

$$H_y(y) = - \sum_i y'_i \log(y_i)$$

注: y'_i 为真实结果, y_i 为softmax后结果

一个样本就有一个交叉熵损失

softmax

| | |
|------|----|
| 0.0 | [0 |
| 0.02 | 1 |
| 0.23 | 0 |
| . | 0 |
| . | 0 |
| . | 0 |
| . | 0 |
| . | 0 |
| . | 0 |
| 0.41 | 0] |

幻灯片布局

标题与副标题

更改母版

外观

☐ 标题

☐ 正文

☐ 幻灯片编号

背景

编辑母版幻灯片

图片分类 逻辑回归: sigmoid \rightarrow 某一个类别的概率 二分类

神经网络: 多分类

某一个样本 \rightarrow 得出属于全部类别的每一个概率

softmax

有多少类别, 输出就是多少个

猫

狗

人

羊

算法

策略

优化

线性回归

均方误差

梯度下降

逻辑回归

对数似然损失

梯度下降

二分类

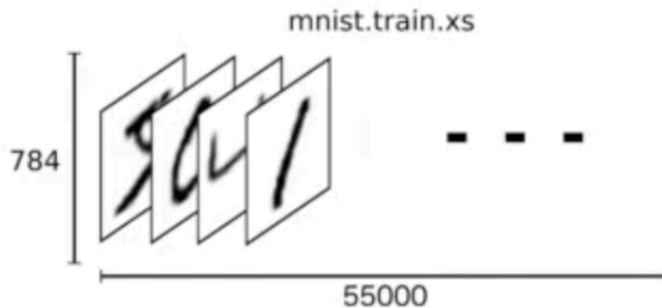
神经网络

交叉熵损失

文本



我们可以知道图片是黑白图片, 每一张图片包含28像素X28像素。我们把这个数组展开成一个向量, 长度是 $28 \times 28 = 784$ 。因此, 在MNIST训练数据集中, `mnist.train.images` 是一个形状为 `[60000, 784]` 的张量。



MNIST中的每个图像都具有相应的标签, 0到9之间的数字表示图像中绘制的数字。用的是one-hot编码

```
mn[0, 0, 0, 1, 0, 0, 0, 0, 0, 0] mnist.train.labels [55000, 10]
```

