深度学习基础理解

# 引言

如果要学习深度学习神经网络的相关知识，首先需要具备**高等数学**、**概率论与数理统计**和**线性代数**的数学基本知识。同时需要掌握**Notebook、Python或者其它主流编程语言**。本总结主要涉及的部分为从零基础开始到卷积神经网络之前的框架构建，因此出于通俗易懂的角度考虑，有些点理解得较为肤浅，请阅读者在往后学习时如果出现认识矛盾的地方注意甄别。

# 第一章 基本概念

## 1.1 输入和输出

通常我们在进行计算机编程任务时，在界面输入数值，最后返回想要的东西，譬如，输入数字1~10，然后通过程序实现筛选输出所有的偶数。在神经网络出现后进行了一些重命名，所以需要知道：

由于我们在利用机器学习网络是用来进行例如股价预测或者图像分类等任务，输入的数据自然是所要进行预测的股价历史数据或者需要分类的图像，这些数据都有相应的特征，而不是数学题中的数据是没有任何意义的，因此**特征可以理解为我们通常所说的输入**。

同时特征还有**另一层含义**，卷积神经网络的主要作用就是提取特征，这时候特征就不再是代表输入数据，而是代表**能够展现数据特点的那部分数据**，和前面的共同点在于，机器学习中的特征本质上就是数据。

对于输出，我们有很多说法了，机器学习的作用大体分为两种，一种是**回归任务**，输出的是连续数值，比如房价预测、股价预测、预测气温或者设备老化；另一种是**分类任务**，分类到一个或者多个类别，比如分一张图片哪一部分是人、哪一部分是狗。所以当在回归预测任务中，输出就叫做预测，而到了分类任务输出叫做标签。

# 第二章 机器学习的本质

## 2.1 内容引入

机器学习的整个过程可以简单理解为：先给你一些现有的输入和输出数据，然后找出规律，当一个新的输入数据进入后，可以通过学习到的规律给出输出数据。下面就给出一个任务来展现机器学习的过程。

现在有如表1所示的数据集，我们给计算机的任务是预测当特征（输入）x=4时，给一个预测值（输出），现在除已知数据集之外，还提供该数据集复合的模型为：

##### (2.1)

实际上很快可以说出但是仔细想想这个过程，我们为什么认为，主要还是用数据集中的值除以的值，然后为什么决定是一个可信度高的结果？是在假设的基础上，代入模型计算我的预测值和实际值之间相差为0，从而得到了的结论。那么机器学习也是如此，它关键解决两个问题，问题1：**机器学习如何得到最优的（权重）；**问题2：**机器学习如何判断不同权重之间的好坏。**

|  |  |  |  |
| --- | --- | --- | --- |
| x的取值 | 1 | 2 | 3 |
| y的取值 | 2 | 4 | 6 |

表2.1 引入案例数据集

### 2.2 最简线性模型学习过程

首先我们来解决**第二个问题**：机器学习如何判断不同权重之间的好坏。这里就要引入一个新的概念——**损失函数**。简单来说，损失函数就是定量反映了我的预测值和真实值之间的差异大小，如果一个枪手老是脱靶，那我们都知道他射得不准，同样，如果机器确定的模型损失函数太大，就证明它的效果不好。实际上，我们之前在得出的时候就在脑海下意识算了一个损失函数：

##### (2.2)

其中**表示预测值，表示真实值**。那么我们用到了很多各种各样的损失函数，通常不会用2.2所示的损失函数，有一个很重要的原因是它存在**不可导点**，至于原因稍后揭晓。本案例用到的是MSE损失函数，具体计算式为：

##### （2.3）

对于复杂的机器学习任务，可能一次预测N值，所以代表第i个值的预测值，代表第i个值的真实值，所以式2.3计算的是误差平方的均值（Mean Squared Error均方误差）。这时就可以写一个代码计算当权重从0到4之间变化时，MSE损失函数的大小变化（图2.1）但是机器学习是不会直接比较大小的，因为这种方法或许在这有用但是其它案例就不管用了。图2.1代码：[Basic-Understanding-of-deep-learning/chapter one/引入案例loss-w图像生成(穷举法) at main · tetanus-sy/Basic-Understanding-of-deep-learning](https://github.com/tetanus-sy/Basic-Understanding-of-deep-learning/blob/main/chapter%20one/%E5%BC%95%E5%85%A5%E6%A1%88%E4%BE%8Bloss-w%E5%9B%BE%E5%83%8F%E7%94%9F%E6%88%90(%E7%A9%B7%E4%B8%BE%E6%B3%95))。

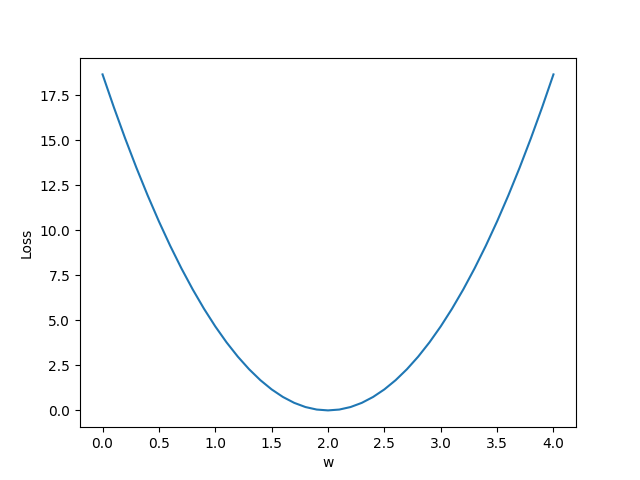


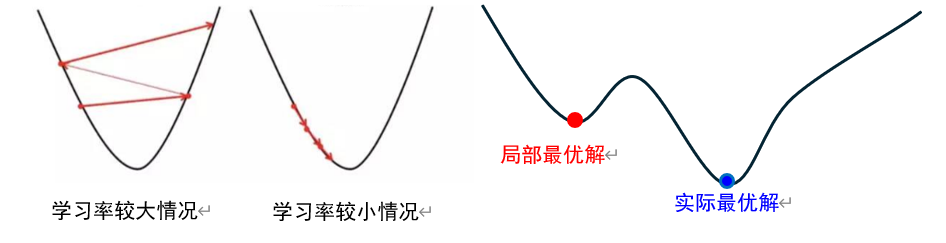
图2.1 引入案例损失函数和权重之间的变化关系图

接下来就可以讨论**第一个问题**，机器学习如何得到最优的（权重）？要解决这个问题，又要引出另一个核心：梯度下降算法，其**数学计算式**如下：

##### （2.4）

其中第t次迭代的权重；为学习率（step size），始终大于0，控制每次更新的步长；表示损失函数在处的梯度，表示参数在当前点的最陡下降方向。其中学习率就是一个超参数，在机器学习中，**超参数**可以理解为这个数的值是人为规定的，不是计算机训练出来的。超参数的设置对训练的效果有较大印象，以学习率为例，如果设置较大，它每一次都移动很远距离，那么很有可能损失函数算出来不是逐渐减小的，甚至越来越大**不收敛**（图2.2）。

结合图2.1来分析，计算机在最开始时会随机生成一个假设的权重，这时会出现两种情况：如果，这是损失函数值对权重的偏导数，则由式2.4可知，下一次的权重会增大向右移动，同理，当，则下一次权重会向左移动，在这个过程中我从假设权重到计算损失函数最终得到算一次**训练周期**（epoch）。经过若干个训练周期之后，最终会得到的结果是训练出的权重会越来越靠近函数的极小值。这里数学敏感的人会发现，梯度下降算法有一个问题，就是由于学习率步长不会跨度很大，那么如果函数有好几个极小值，很有可能最终训练出来的并不是最小值，也就是陷入**局部最优解**（图2.2），因此需要用到随机梯度下降算法，主要原理就是加上一个扰动使其可以跳出局部最优解区域，当然还会有其它问题如鞍点，这里就不多做叙述。



b

a

**图2.2 a.不同学习率训练轨迹；b.局部最优解和实际最优解**

总结以上内容，可以得出机器学习大体上就是通过假设权重，然后用梯度下降算法不断更新权重最后在一定训练周期内得到最优的权重，因此机器学习最核心的任务就是学习模型权重。

### 2.3 专有名词解析

我们来回顾一下每次训练的流程图(图2.3)，为了评价权重的效果所以是从特征输入开始一直到损失函数结束，但是要利用梯度下降算法更新下一次计算就需要求，这时计算导数通过链式法则是从右到左，这两个过程分别就是**正向传播（Forward）**和**反向传播(Backward)**。

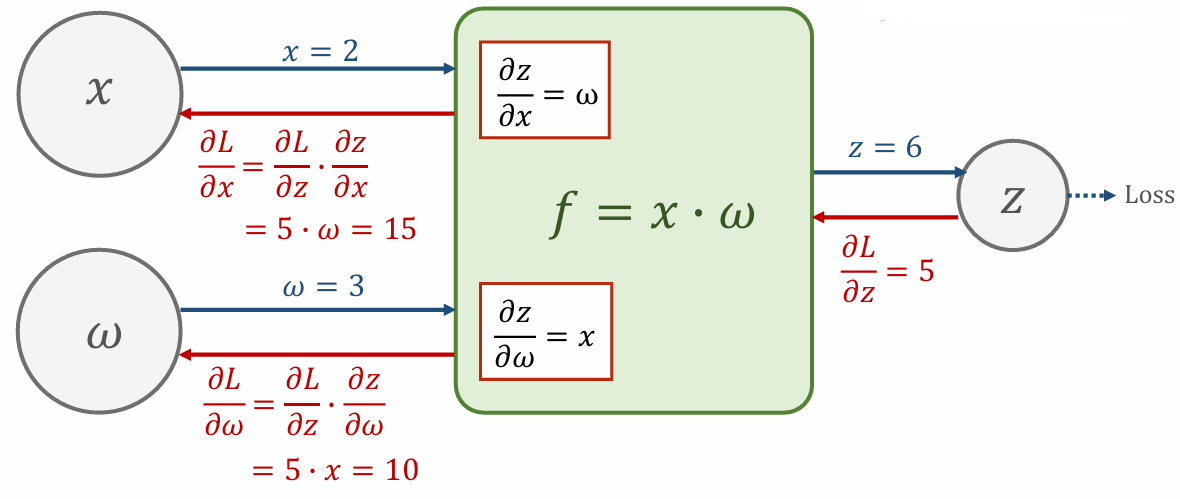


图2.3 案例训练流程图

这里还需要探讨一个问题，以图2.4结构所示的网络架构为例，在这个过程中如果是单纯的线性计算，过程如下：

##### （2.5）

##### （2.6）

##### （2.7）

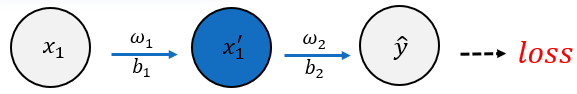


图2.4 单输入线性网络架构图

在这个过程中，可以发现，不论是多少层的线性运算网络，最终都可以用式2.7来表示，也就是输入和输出之间只存在一层线性关系，那么中间层的权重都不再具有意义，且这明显是不符合实际情况的，因此需要在每一层线性运算之后，对于输出的中间变量要进行一次非线性运算，从而层与层之间不再有线性关系，能够增加模型复杂性以满足处理复杂任务的要求。又一个重要元素被引入——激活函数（Nonlinear Function），从它的英文名字可以更直接看出它的作用。最早的非线性函数是sigmoid函数，表达式为：

##### （2.8）

现在对于大的神经网络通常用ReLu函数（图2.5）：

##### ) （2.9）

|  |  |
| --- | --- |
| Sigmoid 函数 - RAIS | Blog |  |
| sigmoid函数 | ReLu函数 |

图2.5 常见激活函数图像

对于更加复杂的线性模型（图2.6），在这个模型中每个层与层之间的每个参数都有权重交流，我们把这种层叫做**全连接层**。

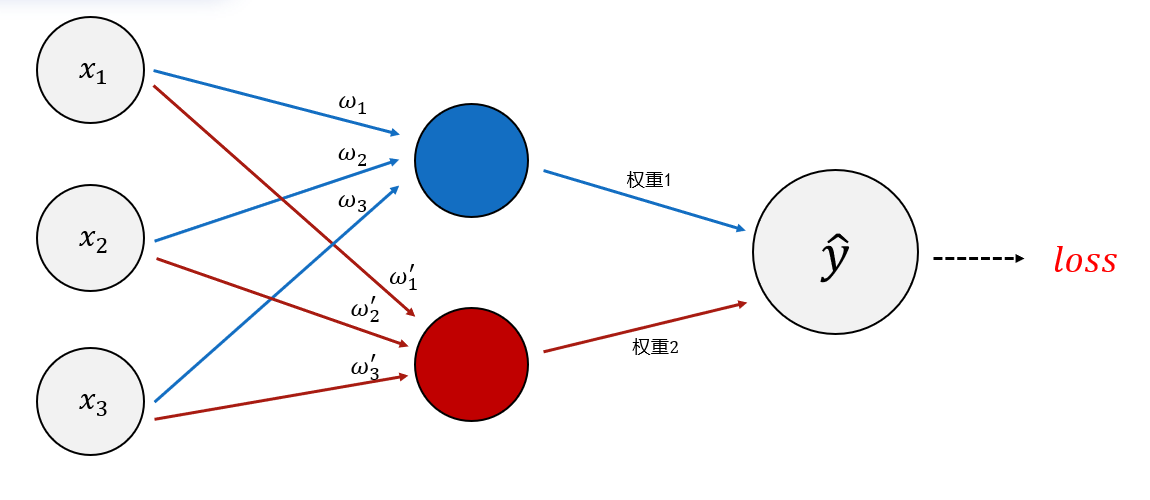


图2.6 全连接层示意图

### 2.4 线性模型训练结果

经过周期为100的训练，最后得到的权重为：；损失函数值为：，从图2.7可以看到，一共训练了100个周期，最后损失值逐渐逼近于0。

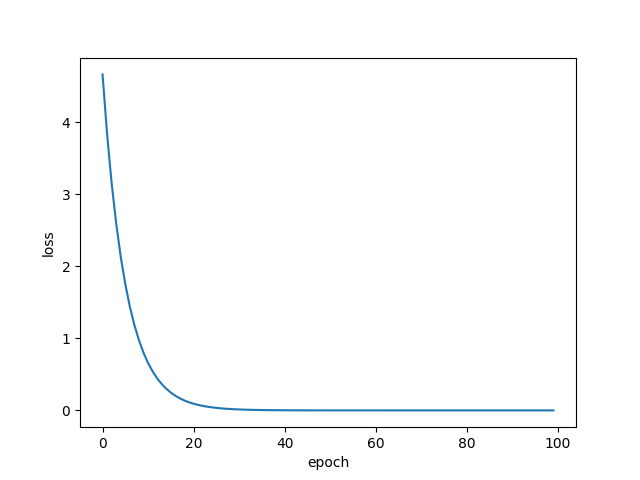


图2.7 损失函数随周期增长的变化图

线性模型图2.7代码：[Basic-Understanding-of-deep-learning/chapter one/引入案例loss-epoch变化图生成 at main · tetanus-sy/Basic-Understanding-of-deep-learning](https://github.com/tetanus-sy/Basic-Understanding-of-deep-learning/blob/main/chapter%20one/%E5%BC%95%E5%85%A5%E6%A1%88%E4%BE%8Bloss-epoch%E5%8F%98%E5%8C%96%E5%9B%BE%E7%94%9F%E6%88%90)

线性模型全过程代码：[Basic-Understanding-of-deep-learning/chapter one/引入案例训练完整流程 at main · tetanus-sy/Basic-Understanding-of-deep-learning](https://github.com/tetanus-sy/Basic-Understanding-of-deep-learning/blob/main/chapter%20one/%E5%BC%95%E5%85%A5%E6%A1%88%E4%BE%8B%E8%AE%AD%E7%BB%83%E5%AE%8C%E6%95%B4%E6%B5%81%E7%A8%8B)

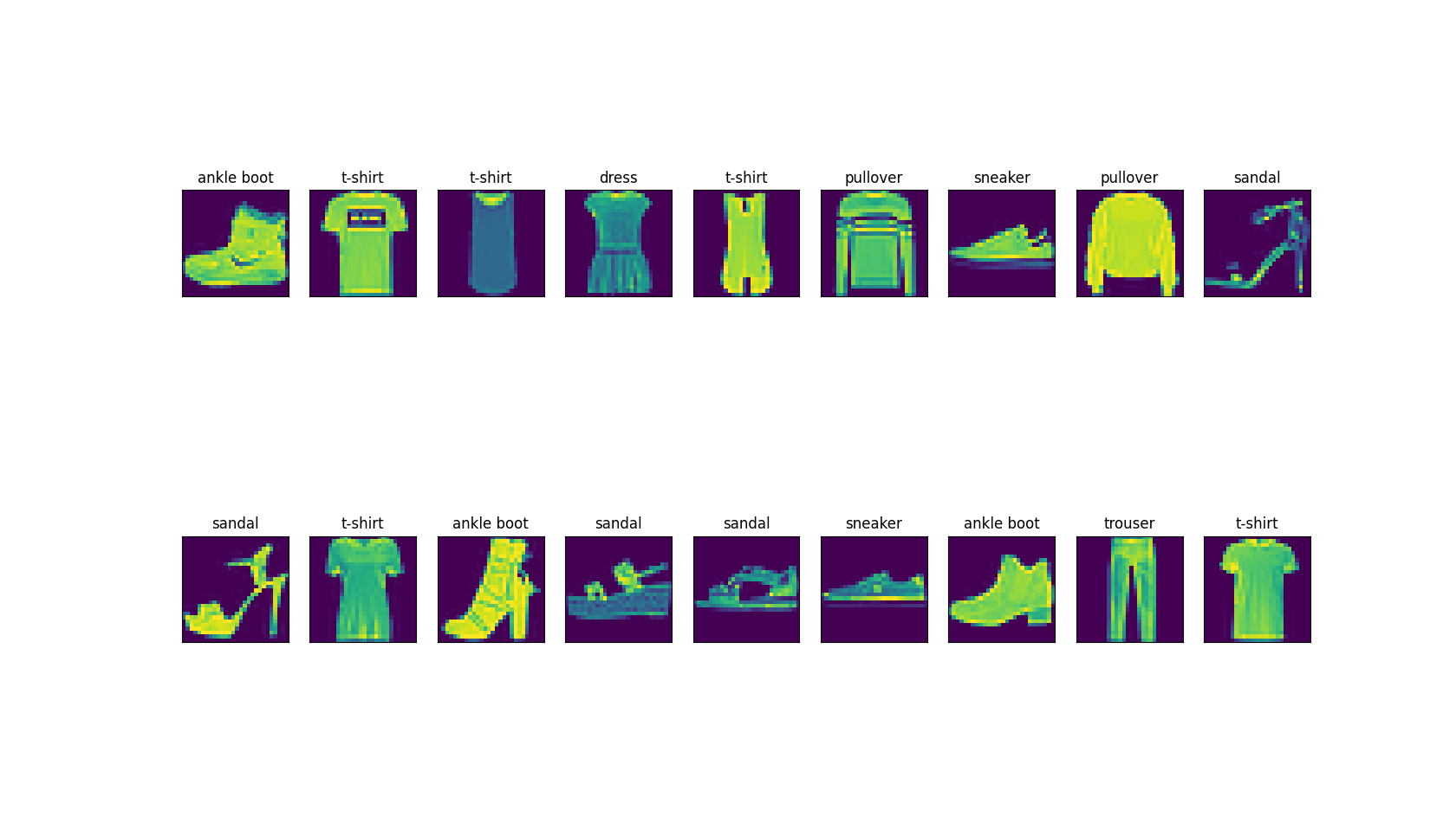
# 第三章 softmax回归

## 3.1 数据集介绍（输入）

对于一个机器学习模型，要了解它首先要知道它的特征（输入），而一个模型的输入是一个很庞大集合，这就是数据集。例如土木要通过深度学习来预测裂缝的开展，那么它要有大量的数据来支撑学习过程，一根梁上的制作就耗费不少，通常数据集需要上千上万个，所以数据是一个很昂贵的产品。

对于softmax回归来说，它的数据集用到的是fashion\_mnist，这个数据集中是一共有十类图像（图2.7），具体对应的标签如下：T-shirt/top(T恤/上衣)、Trouser (裤子)、Pullover (套头衫)、Dress (连衣裙)、Coat (外套)、Sandal （凉鞋）、Shirt (衬衫)、Sneaker (运动鞋) Bag(包)、Ankle boot (短靴)，每一张图像都是28×28像素的黑白照片。

图像集生成代码：[Basic-Understanding-of-deep-learning/Chapter three/图像分类数据集.py at main · tetanus-sy/Basic-Understanding-of-deep-learning](https://github.com/tetanus-sy/Basic-Understanding-of-deep-learning/blob/main/Chapter%20three/%E5%9B%BE%E5%83%8F%E5%88%86%E7%B1%BB%E6%95%B0%E6%8D%AE%E9%9B%86.py)



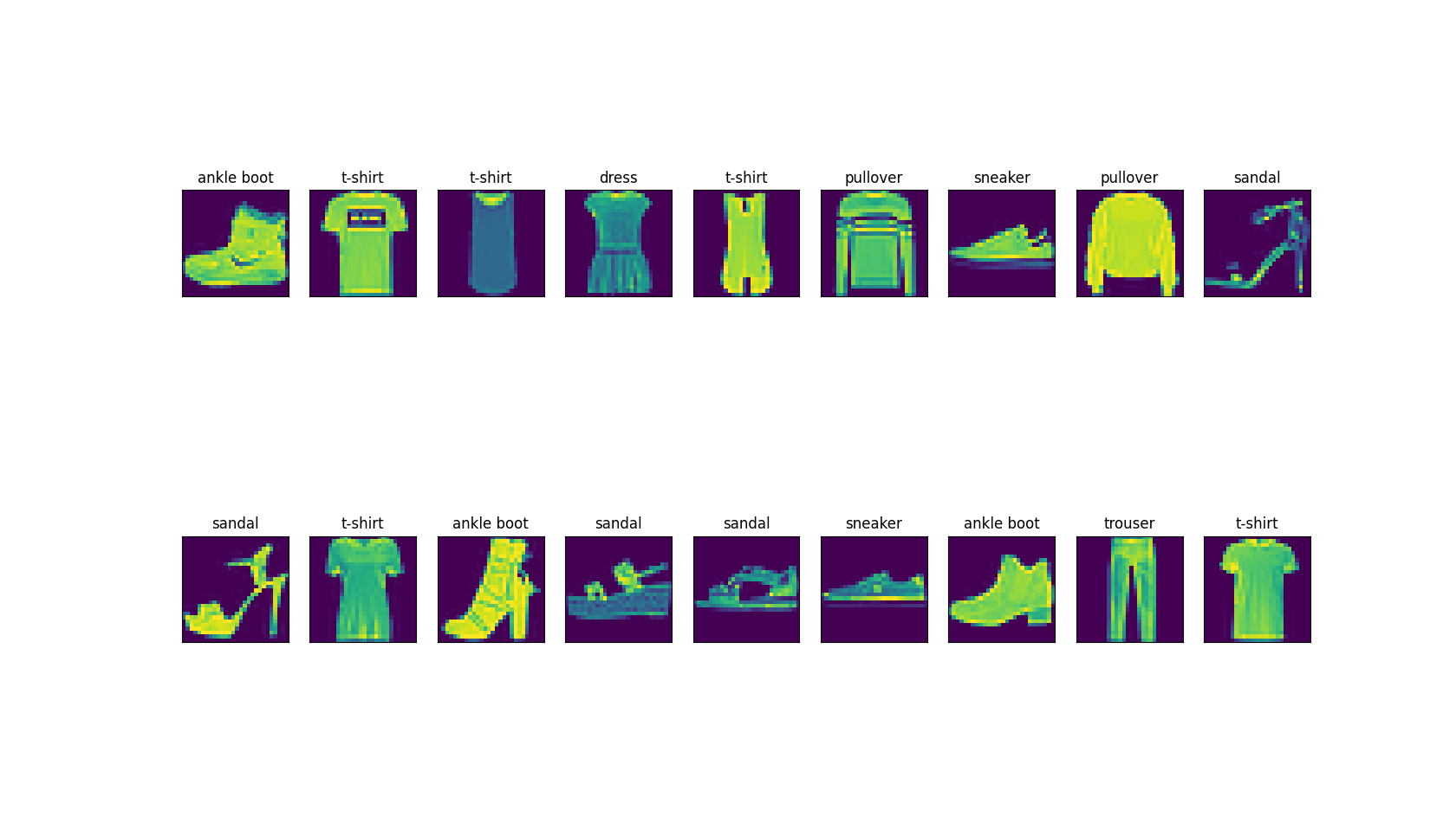


图3.1 Fashion\_Mnist数据库图片

接下来的解释对于整个机器学习理解来说至关重要，**一张图片在计算机中是如何呈现的**，对于黑白照片来说，通常是将0定义为黑色，然后从1到255分别代表不同灰度颜色。一个张量（式3.1）所形成的图像如图（3.2）：

##### [[0, 255],

##### [255, 0]] （3.1）

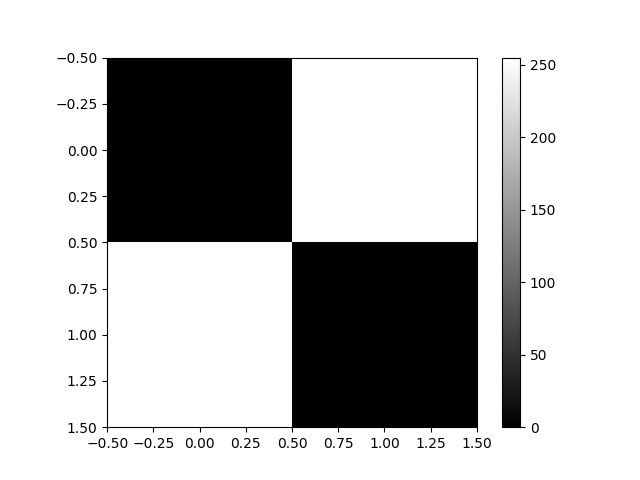


图3.2 张量[[0, 255],生成的黑白图片

[255, 0]]

对于全连接层的输入来说，需要将张量展平降至1维，变成[0, 255, 255, 0]的形式，对于卷积神经网络等，不需要做这个展平操作。因此，softmax回归的输入变量一共有28×28共786个特征输入。

### 3.2 分类任务（输出）

整个神经网络实际上比较简单，只包含一个输入层、一个输出层，没有中间隐藏层。softmax回归的输出量为10，分别对应10个类，如果输出值越大代表图片是该类的概率越大。但是对于线性函数来说，是会输出负数值的，从数学的角度上来说存在两个不合理性，**一是三个原始输出值相加不等于1，一是概率的取值范围是0~1**，softmax函数解决了这两个问题，表达式为：

##### (3.2)

其中为第i个原始输出值，图3.3给出一个三分类例子的softmax处理过程，在这个案例中最后0.38数值最大，认为类别1概率最大。

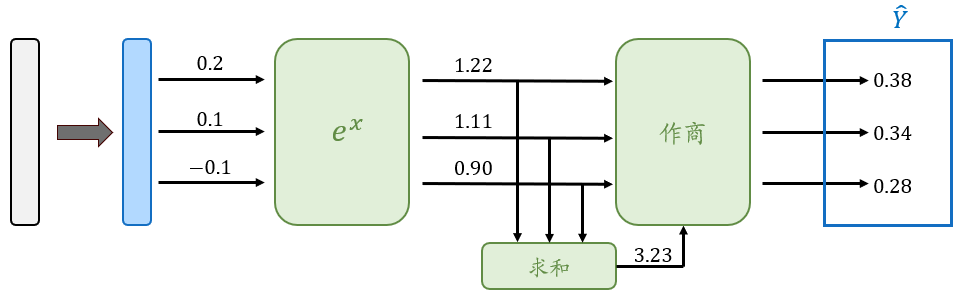


图3.3 三分类softmax处理

### 3.3 损失函数计算

损失函数用到的是**交叉熵函数**（Cross-Entropy）:

##### (3.3)

这里就是前面通过softmax处理后的输出量，真实值则是通过**独热编码**方式进行确定的，假设其中一张图片属于第一类，则真实值是（1，0，0），同理如果属于第三类，则真实值是（0，0，1）。按照图3.3中给的具体数据，在这个训练周期的损失函数计算值为。

### 3.4 训练结果

训练最终结果图3.4：

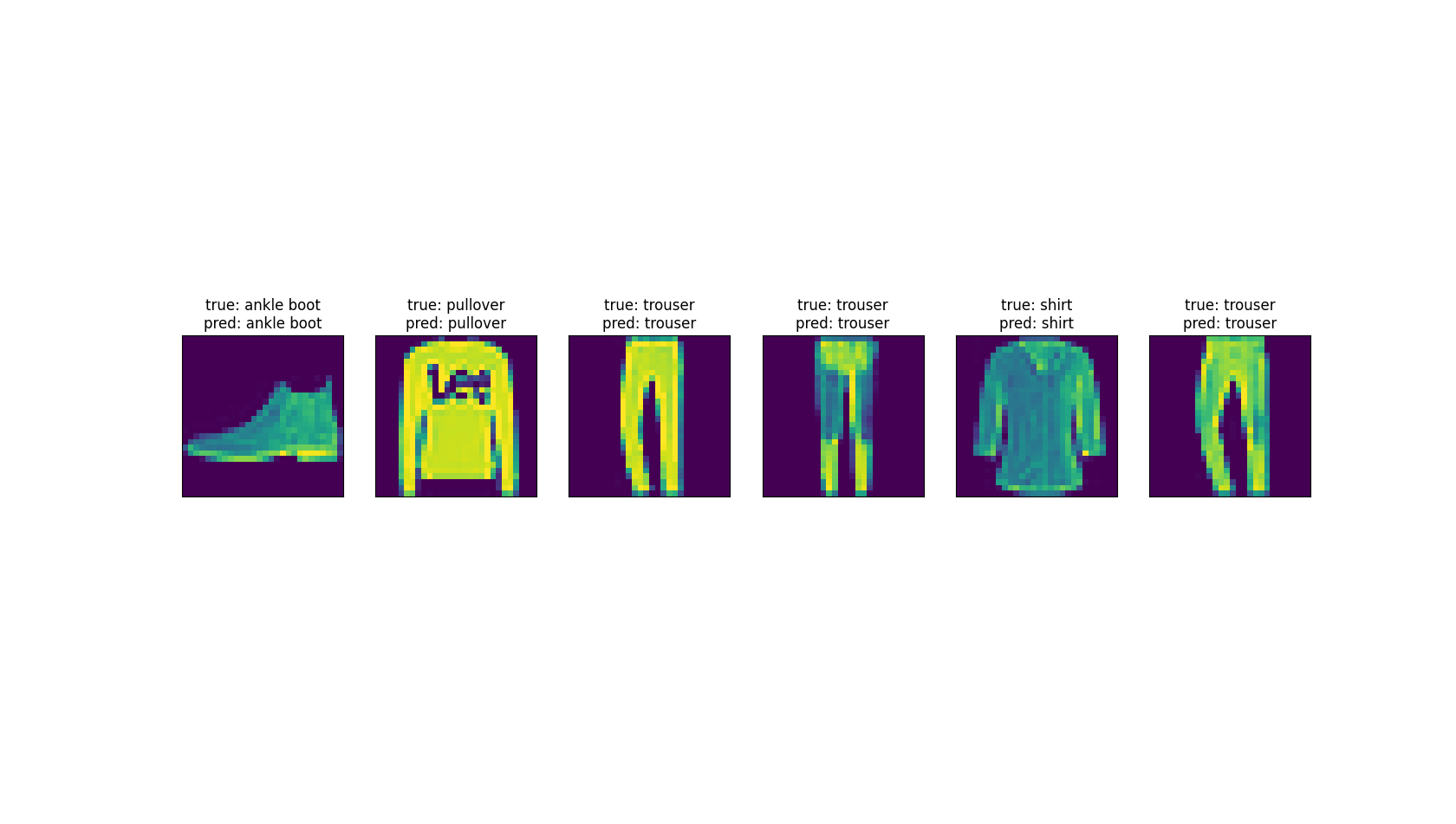
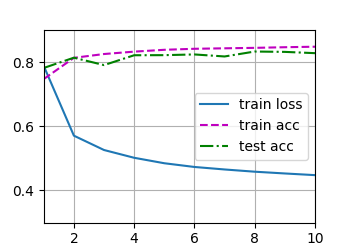


图3.4 训练结果

softmax回归代码实现：[Basic-Understanding-of-deep-learning/Chapter three/softmax回归的从零开始实现.py at main · tetanus-sy/Basic-Understanding-of-deep-learning](https://github.com/tetanus-sy/Basic-Understanding-of-deep-learning/blob/main/Chapter%20three/softmax%E5%9B%9E%E5%BD%92%E7%9A%84%E4%BB%8E%E9%9B%B6%E5%BC%80%E5%A7%8B%E5%AE%9E%E7%8E%B0.py)