

## 1、前言

之前所学的神经网络所用的语言是 Octave，且代码结构比较底层，在掌握机器学习原理之后，考虑使用框架模式开发，着重关注数据选取，神经网络模型构建以及优化，提高编程效率。因此，今天，我将使用 Keras 进行开发。

## 2、Keras 简介

Keras 是一套 API，后端可以基于 Tensorflow 或者 Theano，python3.7 目前不支持 Tensorflow，因此使用 Theano。Keras 是为人类设计的而非机器，它遵循减少认知困难的宗旨，使编程人员着眼于机器学习的项目中，而不是代码上的问题。另外，由于它容易部署在其他平台上，因此有越多的人使用它。

## 3、Sequential 模式

Keras 有两套模式，一种是题目 Sequential 顺序式，还有一种是函数式 API 调用（可以定义复杂模型），首先使用 Sequential 顺序式编程，来看一个简单的例子吧。

```
from keras import Sequential
from keras.layers import Dense, Activation
import numpy as np

data = np.random.random((1000, 100))
labels = np.random.randint(2, size=(1000, 1))

model = Sequential(
    [
        Dense(32, input_dim=100),
        Activation('relu'),
        Dense(1),
        Activation('sigmoid')
    ]
)

model.compile(optimizer='rmsprop',
              loss='binary_crossentropy', metrics=['accuracy'])
model.fit(data, labels, epochs=10, batch_size=32)
```

这是一个二分类问题，首先导入 keras 包下的 Sequential 以及 layer 中的 Dense 和 Activation，再引入 numpy 用于创建训练集。

【第一步】构建神经网络层，包括每层单元数，激活方式 s

【第二步】创建优化器、损失函数和检验标准

【第三步】开始训练，训练集、迭代次数，每次训练的 batch。

就目前来看，使用 keras 框架使用神经网络的确简洁太多，并且也不致于让编程人员不知道在做什么。接下来，讨论一下这里边的细节问题，看看每个参数都有些什么作用。

```
model = Sequential(  
    [  
        Dense(32, input_dim=100),  
        Activation('relu'),  
        Dense(1),  
        Activation('sigmoid')  
    ]  
)
```

第一步，使用 Sequential 构建神经网络层结构，Dense 中第一个参数为目标个数，如隐藏层单元数和输出层单元数，第二参数为上一层特征数（只需设置第 2 层隐藏层的该参数就行，其余的模型会自己推导）；激励方式，常见的有 logistic 回归中 sigmoid 函数。

除了这种设置方式外，还可以使用 model.add() 方式设置。

```
model.compile(optimizer='rmsprop',  
loss='binary_crossentropy', metrics=['accuracy'])
```

第二步，设置优化器、损失函数（使目标函数最小的方程）以及检验标准。

Optimizer 可以设置为 SGD 优化器和 RMSprop 优化器等，关于优化器相关知识，后面会提到。

损失函数，类似于 Octave 中的代价函数，通过最小化这个函数，使得模型最优。

Metrics，评估标准，每次迭代后，都会计算一次这个值为多少，以及评判模型拟合的程度。

```
model.fit(data, labels, epochs=10, batch_size=32)
```

第三步，训练，前两个参数为训练集，batch\_size 为每次梯度更新的样本数，默认 32，epochs，为迭代次数。

后续步骤，还可以进行模型评估 evaluate 和模型预测 predict

以上便是神经网络简单的搭建模型，后端是基于 Theano，下面，将对 Theano 底层做初步了解。

## 4、Theano 与 Octave

Theano 作为 keras 后端的神经网络层底层实现，它的实现机理与 Octave 类似的，首先需要了解一个很重要的实现过程，即定义一个 Theano.function，给定输入与输出的映射关系，即可使用该 function 得到由输入获取的输出，因此，无论是对线性回归还是分类问题，甚至还会包含神经网络层中的隐藏层，都可以用该方法获取代价值和偏导数，再使用梯度下降算法更新该值。除此之外，针对过拟合，Theano 也有相应的处理方法。