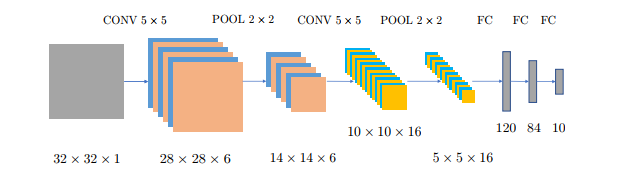
**深度学习第四次作业——CNN基本原理与实现**

1. **基本原理**

卷积神经网络(CNN)是一种特殊的FNN，其专门为图像数据所设计用于解决图像分类问题，目前以广泛应用于模式识别、图像特征提取与压缩等多个领域。

一个较为完整的CNN由卷积层、池化层和全连接层构成。卷积层利用多个适合图像阶数的滤波器扫描图片，进行特征提取，得到多个特征图（feature map）作为输出，而池化层将接受到的特征图尽可能保留原始特征的情况下进行压缩，将结果丢给下一个卷积层，然后重复此过程。重复多个卷积-池化过程后，将结果丢入一个全连接神经网络进一步进行特征提取并完成分类任务。

一个简单的CNN架构如下图：



可以看出，该网络共经历了两次卷积过程和两次池化过程，末端所连接的FC能够使整个网络具有解决多分类任务的能力。

1. **实验部分**
2. 实验目标：运用CNN完成基于Fashion数据集的多分类任务
3. 实验过程与结果
4. 导入相关Python工具库与数据预处理

导入本次实验所用到的算法库：



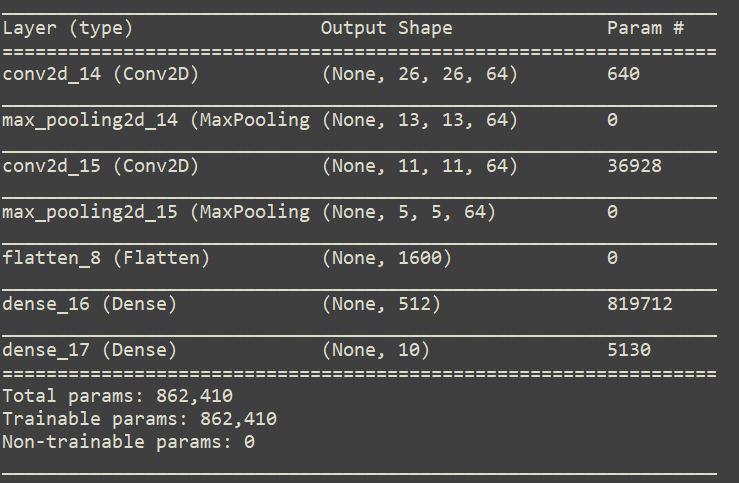
读取Fashion数据集，将数据集分为训练集（55000）、验证集（5000）和测试集（10000），利用matplotlib可视化该数据集中部分图片：



由于该数据集中数据阶数为2阶，为了适合之后的CNN架构，将图片数据升阶为3阶，并进行归一化处理。

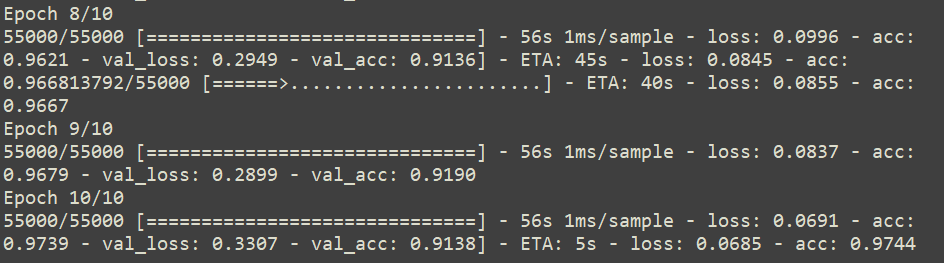
2）构建CNN

利用Tensorflow2.0中的keras库构建一个CNN框架，卷积层中激活函数设置为”ReLu”，输出层的激活函数设置为”softmax”,该模型由两个卷积层、两个池化层和全连接FNN所构成，共有862410个参数需要估计:

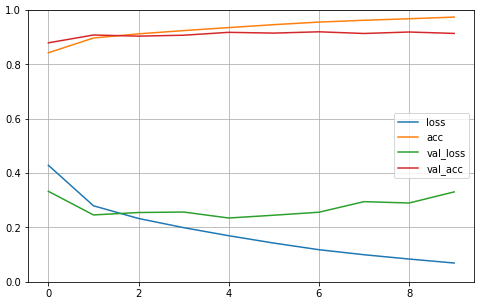


3）训练网络参数

将loss 设置为"sparse\_categorical\_crossentropy"损失，算法设置为adam算法，adam的训练周期epoch设置为10，batch-size设置为默认值，在训练集上进行训练。下图展示了最后3个epoch模型的training loss、training acc以及在验证集上的loss和acc。

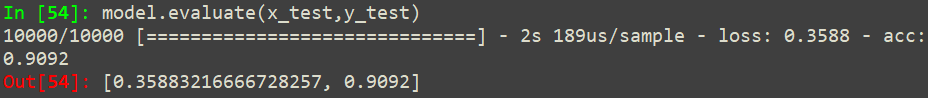


训练完毕后模型在training data上的精度高达97.44%且在验证集上的精度也高达91.38%，进一步画出学习曲线：



4）模型评价

评估以习得的参数在测试集上的表现：



精度为90.92%，说明训练出的模型可较好解决此多分类问题。

**三、Python代码**

