**AI Challenger 场景分类**

**摘 要**

卷积神经网络是一种深度前馈神经网络，是近年来迅速发展并在计算机视觉、语音识别、自然语言处理等领域有着广泛运用的一种高效识别方法，已经成为众多科学领域的研究热点之一。本文运用卷积神经网络进行场景分类，通过卷积层对原始图像数据进行特征提取；通过池化层保持不变性并起到降维的作用，防止模型过拟合，增强模型的泛化能力。首先介绍卷积神经网络的发展历程和一些经典的模型，其次介绍数据来源与预处理的过程，然后完成模型的代码实现，并通过批量随机梯度下降进行训练，最后检验模型的分类效果并提出可能的改进方法。

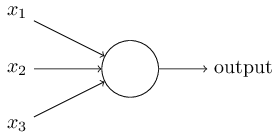
**Abstract**

Convolution neural network is a kind of deep feedforward neural network, which is an efficient identification method which has been widely used in recent years such as computer vision, speech recognition and natural language processing. It has become a hot research topic in many fields of science one. In this paper, the convolution neural network is used to classify the scene, and the original image data is extracted by convolution layer. The pooling layer remains invariant and reduces the dimensionality, preventing the model from over fitting and enhancing the generalization ability of the model. Firstly, the development history of convolution neural network and some classic models are introduced. Secondly, the data source and preprocessing process are introduced. Then the code of the model is implemented and trained by batch random gradient descent. Finally, test the model on validation dataset and give possible improvement methods.

**1 深度神经网络**

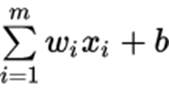
**1.1 DNN**

1957年，美国康奈尔航空实验室的Frank Rosenblatt发明了感知机算法。它的模型很简单，如下图：

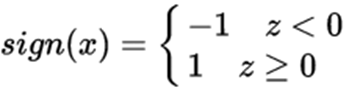


**图1-Perceptron**

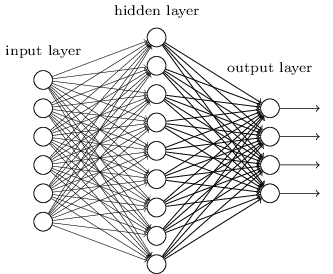
通过对输入做线性变换：



得到一个中间输出，然后对中间输出用一个激活函数进行变换：



得到最终输出。这个模型的缺陷在于它只能进行简单的二分类，无法满足工业界要求。因此引出了神经网络。神经网络的模型如下图：



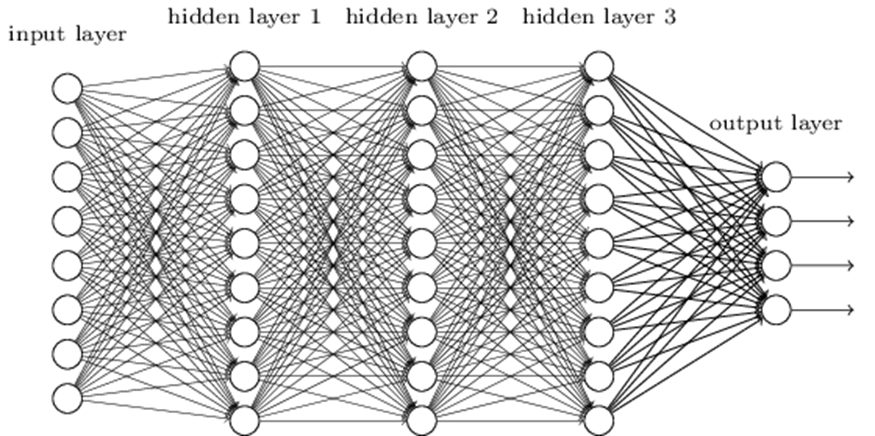
**图2-nn**

神经网络是对感知机的扩展，主要有三个特征：一是加入了隐藏层，增强了模型的表达能力；二是输出神经元可以有多个，即可以进行多类分类；三是改进了激活函数，由sign函数转变成sigmoid函数：



当然还包括后来出现的tanx, softmax,和ReLU等。

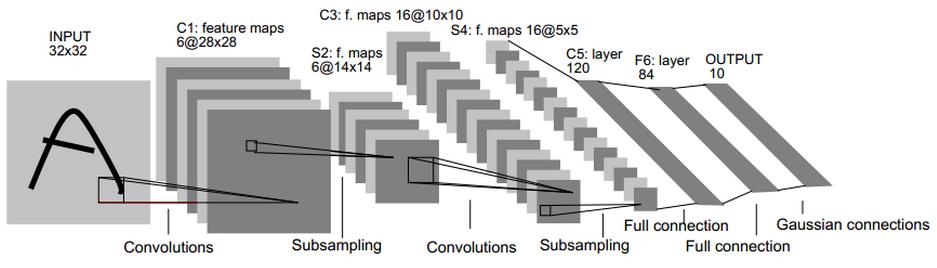
DNN即有很多隐藏层的神经网络，如下图：



**图3-dnn**

**1.2 CNN**

CNN即卷积神经网络，一个典型的卷积神经网络如下图:

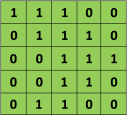


**图4-cnn**

图中是一个手写体图像识别的CNN模型。可以看出最左边的字母A的图像就是我们的输入层，计算机理解为输入若干个矩阵，这点和DNN基本相同；接着是卷积层（Convolution Layer），然后是池化层，卷积层+池化层的组合可以在CNN中出现多次；后面是几个全连接层，即普通的DNN结构。

卷积层的思想来源于哈佛的Hubel和Wiesel所做的猫的视觉实验，他们发现大脑视觉区神经是按列组织的，大脑处理图像是从简单的边缘结构开始的，卷积的过程实际上是一个探测事物轮廓和边缘结构的过程。其具体过程如下：

假设输入数据是个5\*5的矩阵，如下图：



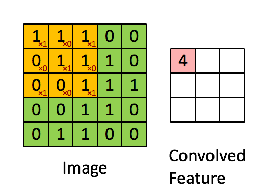
**图5-cnn\_input**

卷积核是个3\*3的矩阵，如下图：



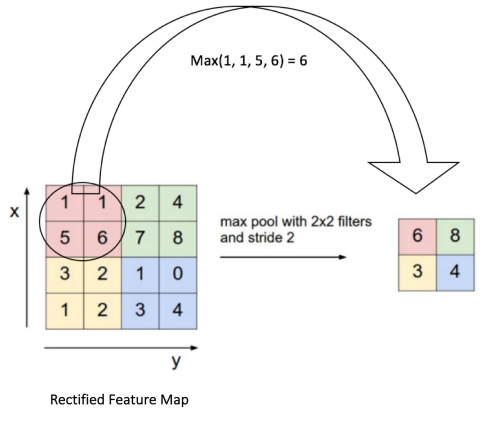
**图6-cnn\_kernel**

卷积步长为1，那么卷积的过程如下图：



**图7-convolution**

池化层是对输入张量的各个子矩阵进行压缩。以2\*2的池化核为例，步长为2，采用最大值池化，池化过程如下图：

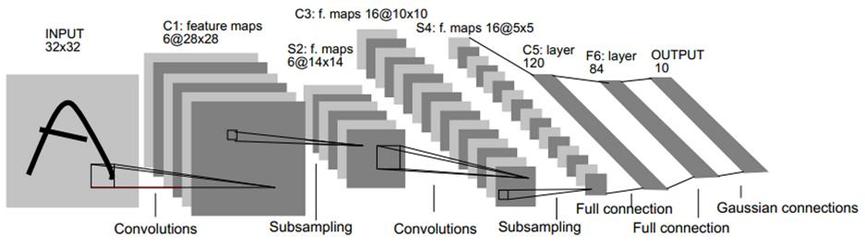


**图8-pool**

池化层的作用一是保持不变性，包括平移、旋转和尺度变化，二是特征提取，在保留主要特征的同时起到降维的作用，减小了过拟合，增强模型的泛化能力。

**1.3 LeNet**

LeNet-5是Yann LeCun在1998年设计的用于手写数字识别的卷积神经网络，是早期卷积神经网络中最有代表性的实验系统之一。结构如下图：



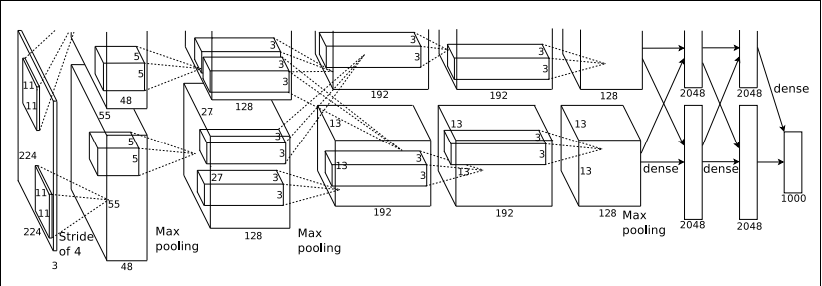
**图9-cnn**

可以看到，输入层为3232的像素矩阵，第一个卷积层卷积核为55，步长为1，第一个池化层为22，步长为2的最大值池化层，第二个卷积层卷积核为55，步长为1，第二个池化层为2\*2，步长为2的最大值池化层，然后是3个全连接层，输出层有10个神经元，对应从0到9的是个数字。

LeNet在手写体识别上的分类效果相当不错，并被应用于邮票系统手写体识别当中。

**1.4 AlexNet**

AlexNet是2012年ImageNet竞赛场景分类的冠军模型，它大幅度降低了分类错误率，Top5错误率达到了15.3%，后来的许多CNN模型也是在其基础上发展而来。其结构如下图：



**图10-alex\_net**

可以看到，输入层为224\*224\*3的矩阵（实际应该是227\*227\*3）,作者采用了双gpu并行运算，加速模型训练，因此整个网络结构分为上下两层。如下图：



**图11-alexnet\_structure**

整个模型的参数总量超过6000万个，相对于LeNet来说复杂了许多，这也得益于计算机计算能力的大幅度提高。

**2 数据**

**2.1 数据来源**

LeNet和AlexNet数据全部来源于AI Challenger。

训练集数据：ai\_challenger\_scene\_train\_20170904

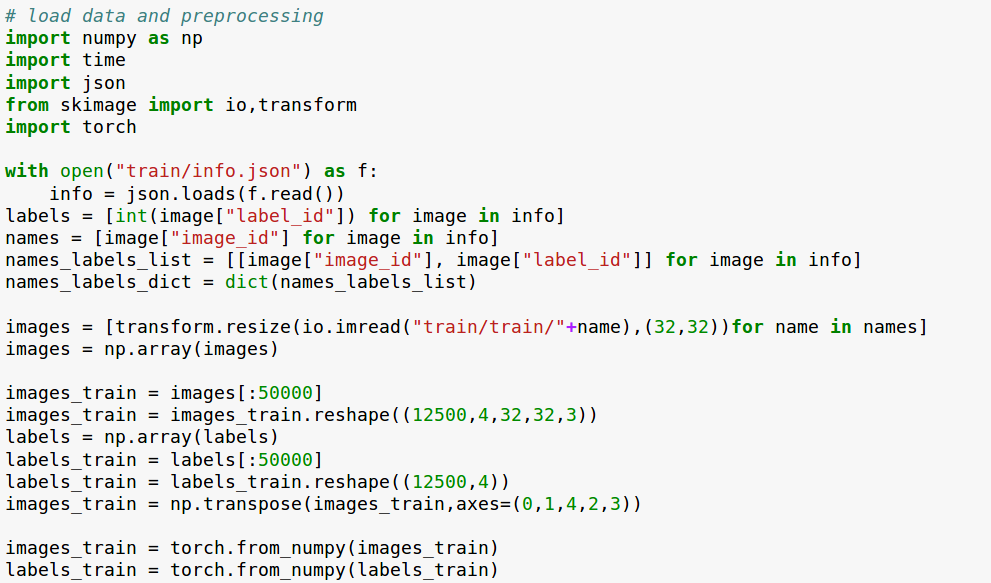
验证集数据：ai\_challenger\_scene\_validation\_20170908

测试集数据：ai\_challenger\_scene\_test\_a\_20170922

原始数据均为RGB三通道的图像，分辨率不一致。

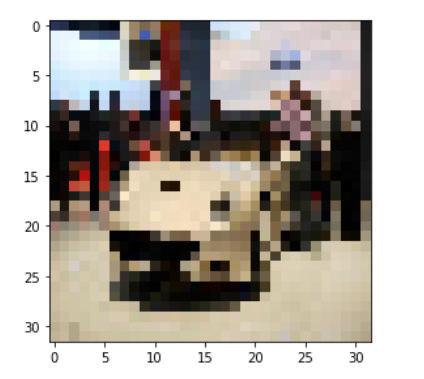
**2.2 数据预处理**

数据预处理主要采用python图像处理的相关包，具体代码如下：



**图12-preprocessing**

先将图像读取成numpy数组，并将尺寸转换为3\*32\*32，转换后的图片（车展）如下：



**图13-image**

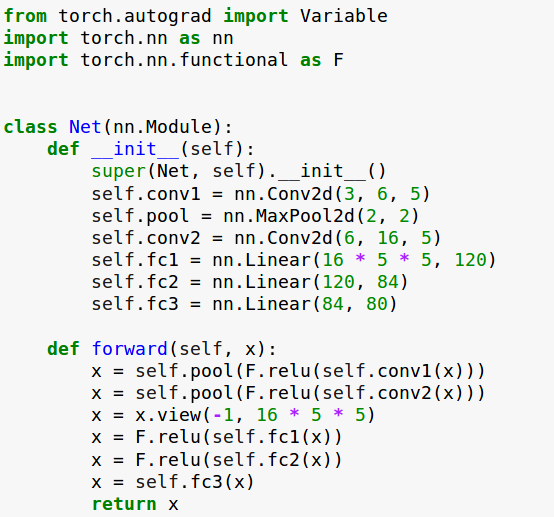
再对数组进行维度变换，每张图片的维度变为32\*32\*3，并将图片分为4张一组，作为批量梯度下降的batch size，最后将图片转化为tensor（张量）。

**3 建立模型**

考虑到tensorflow的复杂程度和学习成本，本文采用pytorch完成CNN的代码实现。

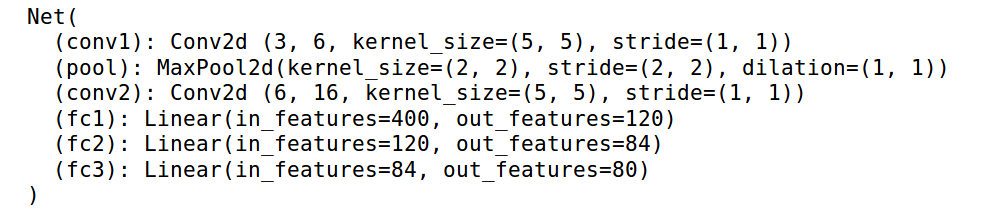
**3.1 LeNet**

LeNet的代码如下：



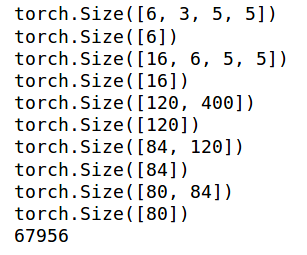
**图14-lenet\_code**

结构如下：



**图15-lenet\_structure**

参数结构如下：

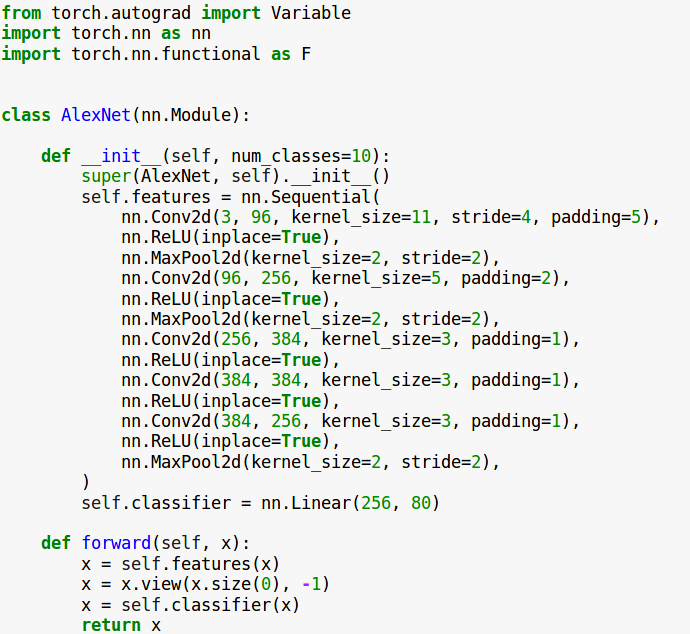


**图16-lenet\_parameters**

**3.2 AlexNet**

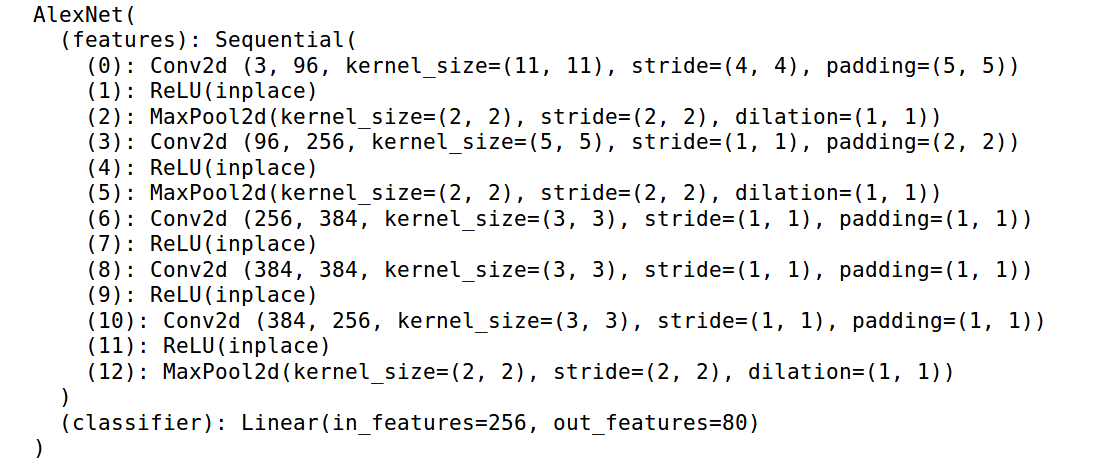
本文建立的模型与原始的AlexNet有所不同，一是输入数据，原始的AlexNet输入数据为3\*227\*227的矩阵，而本文由于内存和数据读取时间等原因将图像的尺寸定为3\*32\*32；二是由于原始的AlexNet的绝大多数参数都来自后面3个全连接层，带来巨大的运算量，对计算资源要求很高，训练时间过长，因此本文将3个全连接层改为1个；三是类别数，原始的AlexNet针对的是ImageNet竞赛，类别数为1000，而AI Challenger的类别数只有80；四是去掉了原始AlexNet中的dropout层。

AlexNet的代码如下：



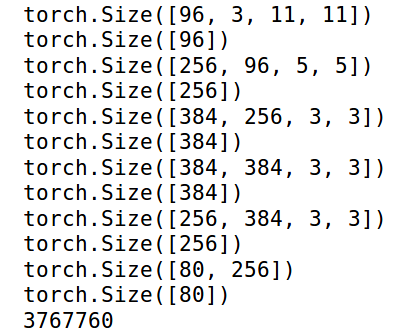
**图17-alexnet\_code**

结构如下：



**图18-alexnet\_structure**

参数结构如下：



**图19-alex\_net\_parameters**

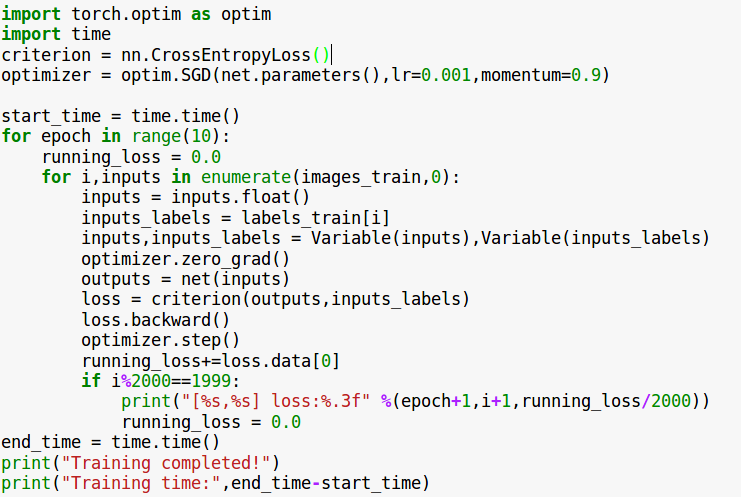
可以看到，相对于原始AlexNet的6000多万参数，本文所建立的模型只有370多万参数。

**4 训练模型**

本文采用的损失函数为Cross Entropy Loss（交叉熵损失函数），最优化方法为batch size为4的批量梯度下降。

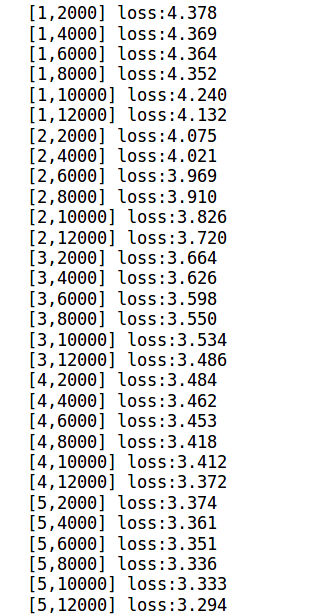
**4.1 LeNet**

代码如下：

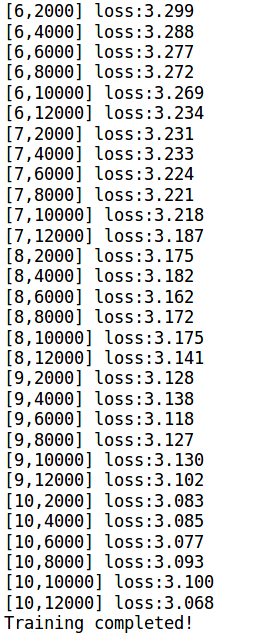


**图20-lenet\_training\_code**

在训练集上的交叉熵损失变化如下：



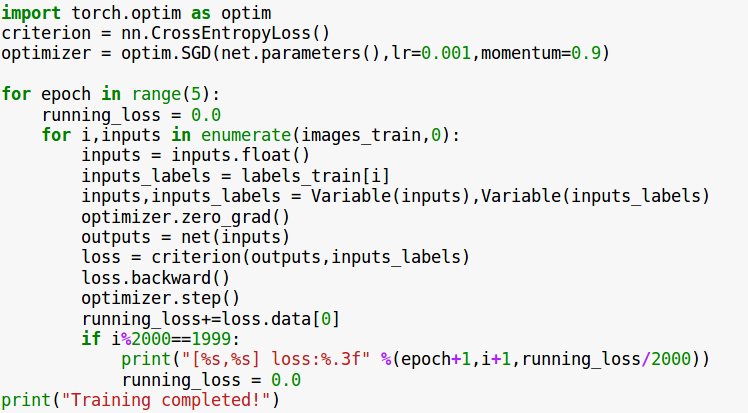
**图21-lenet\_training\_loss1**



**图22-lenet\_training\_loss2**

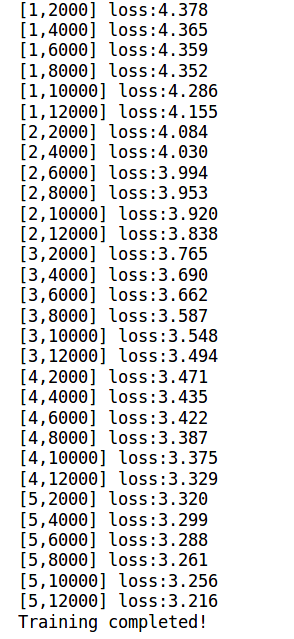
**4.2 AlexNet**

代码如下：



**图23-alexnet\_training\_code**

在训练集上的交叉熵损失函数如下：

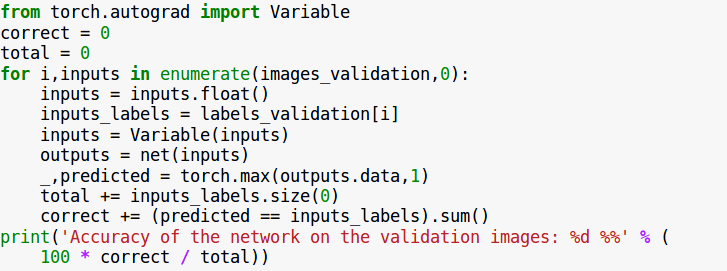


**图24-alexnet\_training\_loss**

**5 模型分类效果**

**5.1 LeNet**

代码如下：



**图25-lenet\_validation\_code**

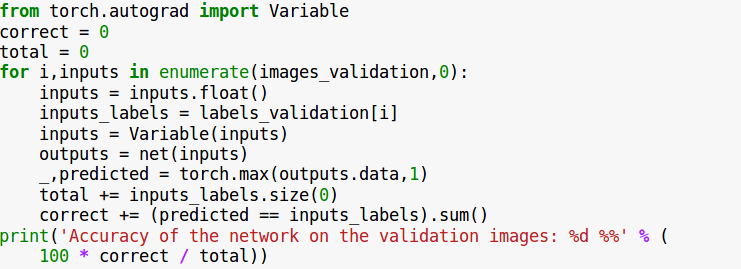
验证集上准确率如下：



**图26-lenet\_accuracy\_on\_validation**

**5.2 AlexNet**

代码如下：



**图27-alexnet\_validation\_code**

验证集上准确率如下：



**图28-alexnet\_accuracy\_on\_validation**

**6 总结**

从分类的效果来看，无论是LeNet还是AlexNet，效果都差强人意，远远达不到工业应用的标准。LeNet由于本身的网络深度不够，分类效果一般在意料之中，但是AlexNet在ImageNet上对1000类图片的Top1错误率超过50%，远远好于本文所建模型。本文作者认为原因首先是图像大小，本文的图像大小为32\*32，比起227\*227丢失了许多图像信息；其次是网络结构相对简单，去掉了dropout层和将3个全连接层变为1个全连接层，在简化模型的同时也减弱了模型的表达能力；最后是数据量，本文所选用的训练集只有50000张图片，通过增加数据量可以在一定程度上提高分类准确率。