**人工神经网络HW2 Cifar-10 Classification with MLP and CNN**

冯卓尔

计86 2017011998

fengzhuoer-thu@outlook.com

**摘要**：本文主要记录了Cifar-10 Classification with MLP and CNN作业中的一些实验，测试了MLP与CNN激活函数在本任务中的优劣，BN层与Drop-out层在相应网络中的性能，以及测试了各项参数对任务的准确率与loss中的影响。

**关键字**：多层感知机（MLP） 卷积神经网络（CNN） 数字图像识别 神经网络

**一、网络基本参数介绍**

**1.1 MLP全连接网络参数**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 层 | 1 | 2 | 3 | 4 | 5 |
| 类型 | Linear | BN | ReLU | Dropout | Linear |
| 输入节点数 | 784 | 256 | 256 | 256 | 256 |
| 输出节点数 | 256 | 256 | 256 | 256 | 10 |

Learning rate = 0.001

Decay=0.9995

**1.2 CNN网络参数**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 编号 | 类型 | 输入 | 输出 | kernel（卷积、池化） | Dropout |
| 1 | Conv2d | Self.x\_(32\*32\*3) | Conv1 | 40\*8 |  |
| 2 | bn | Conv1 | Bn1 |  |  |
| 3 | Relu | Bn1 | Relu1 |  |  |
| 4 | Dropout | Relu1 | Dp1 |  | 0.5 |
| 5 | Maxpool2d | Dp1 | Pool1 | 2\*2 |  |
| 6 | Conv2d | Pool1 | Conv2 | 40\*8 |  |
| 7 | Bn | Conv2 | Bn2 |  |  |
| 8 | Relu | Bn2 | Relu2 |  |  |
| 9 | Dropout | Relu2 | Dp2 |  | 0.5 |
| 10 | Maxpool2d | Dp2 | Pool2 | 2\*2 |  |
| 11 | linear | Pool2.reshape(-1, 160) | 10 |  |  |

Learning rate = 0.001

Decay=0.9995

**1.3 运行环境**

Google Cloud n1-standard-4 Instance (4 vCPU, 15 GB RAM) with 1 NVIDIA Tesla P100 GPU

**二、实验结果与讨论**

**2.1 如何填写网络参数设置**

对于训练过程，in\_train=True，resue = False；对于validation过程，由于需要使用到与训练过程的同一组参数，因此不需要进行再次训练改变参数，in\_train=False，resue = True。

在BN层与drop out层中，封装的一些函数在is\_train与否的状态下表现不同，因而每次进行forward时需要知道当前网络的is\_train状态，即每次传参都需要有is\_train，is\_train来控制BN层与drop out层的行为。

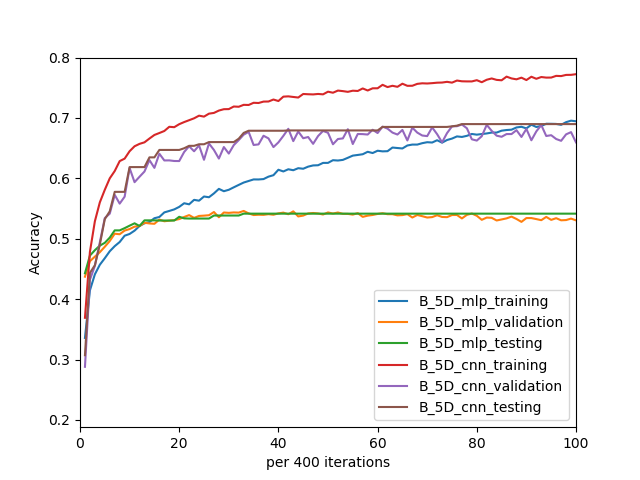
在网络的\_\_init\_\_函数中，forward会被调用两次。如果不使用variable\_scope功能，那么通过self.x\_会建立两张绝缘的相同网络，它们的参数不共享（但是事实上两次作用的参数应当共享）。Reuse=true时第二次调用forward时model就不需要重新建立一个新的网络，而是使用一个名为’model’的已建立网络中的参数进行第二次调用。如果reuse=false，同时建立两个重名为‘model’的网络会引发报错。

之所以train与test需要分开执行，是因为在train过程中，BN层与Drop-out是为了降低网络对训练集的依赖而做的防止过拟合的措施，而测试时并不需要，因而可以通过is\_train来控制网络训练的逻辑，使训练的时候BN与drop-out层发挥应有的作用，而测试的时候不发挥。

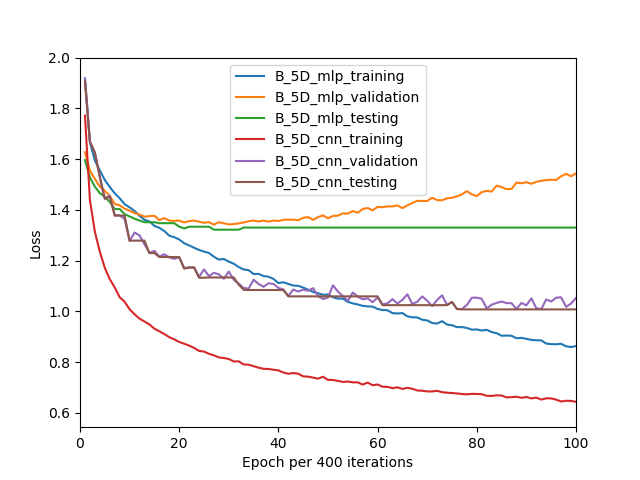
**2.2 training loss/acc与validation loss /acc的绘制**

从图中我们可以观察到，两个模型的曲线出现以下特征.

随着迭代的增加，training acc不断增加，loss不断减少；但是validation和testing acc到一定时候会停止增长甚至会减少，loss到一定时候会停止增长甚至会增加（B\_5D\_mlp\_validation loss）。出现这种现象的原因应该是过拟合。在这个任务中，由于mlp的感知能力有限，在感知范围内的训练完成后会过度，对于未感知到的特征会忽略，因而mlp模型过拟合现象更加明显。使过拟合现象更加明显的做法可以将dropout-rate调整至接近0的小数，反之增大dropout rate（<1）能够缓解过拟合现象。



图一 MLP/CNN网络training, validation, testing 准确率



图二 MLP/CNN网络training, validation, testing loss

**2.3 MLP与CNN模型在处理cifa-10问题上的区别**

在2.2中的图线中可以看出，CNN的testing acc显著比MLP的高，CNN的testing loss比MLP的低。

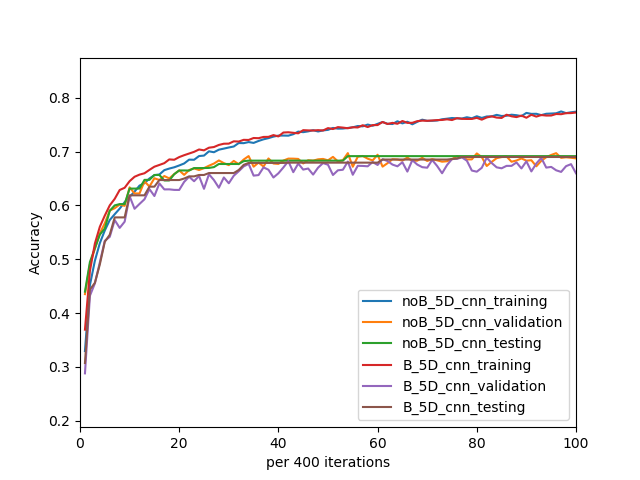
另外，MLP的validation loss较training loss显著要高，在testing loss停止减小后，MLP网络的training loss仍然在减小，但是validation loss反而变大了（40个epoch之后），过拟合现象比较严重；而同时期的CNN网络中并未发现类似的过拟合现象。

|  |  |  |
| --- | --- | --- |
| 网络 | MLP | CNN |
| Training\_acc | 0.6946 | **0.7725** |
| Training\_loss | 0.8632 | **0.6434** |
| Validation\_acc | 0.5305 | **0.6596** |
| Validation\_loss | 1.5434 | **1.0526** |
| Testing\_acc | 0.5416 | **0.6899** |
| Testing\_loss | 1.3302 | **1.0077** |

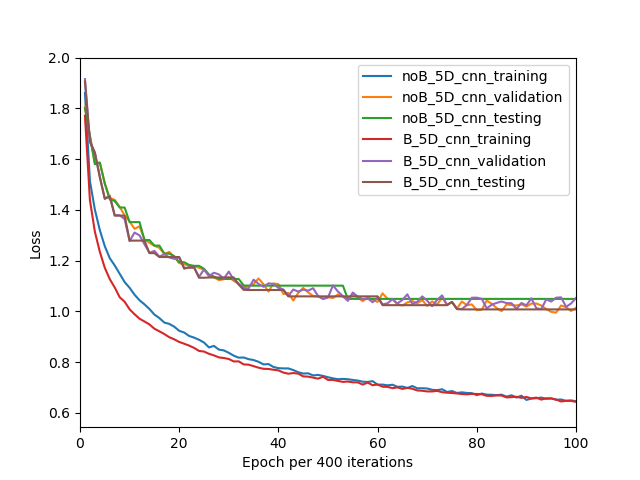
**2.4 BN层对MLP与CNN模型在处理Cifar-10问题上的性能影响**

为了研究这个问题，我删去了CNN中的两个BN层，MLP中的唯一BN层，并且将drop rate调节至CNN网络性能最优的0.5，进行了实验。将其结果与带有BN层的结果一起绘制于下图中。

**CNN**：可以看出，从收敛后的结果上看，BN层似乎不能明显提升CNN网络的testing acc，但是在loss曲线中，BN层的存在下CNN网络的收敛速度提升了一点点。在得到近乎相同的结果前提下，收敛速度的提升对于一个网络的性能也有很大的提升。



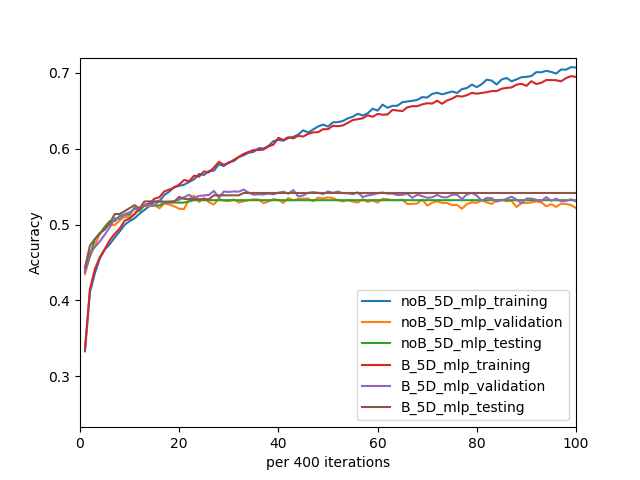
图三 CNN网络有无BN层的training, validation, testing acc对比



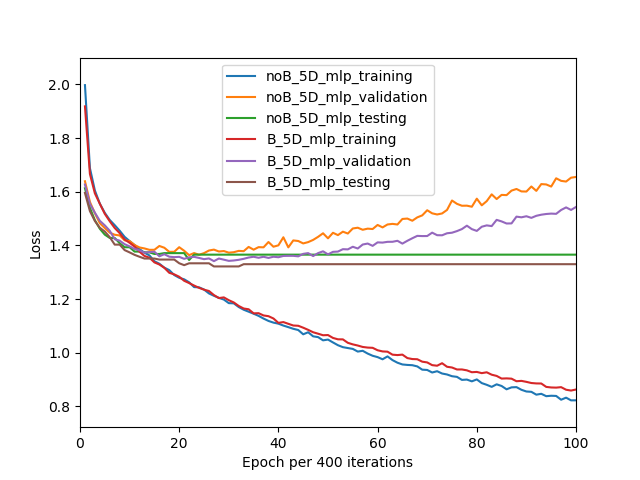
图四 CNN网络有无BN层的training, validation, testing loss对比

|  |  |  |
| --- | --- | --- |
| 网络 | CNN+BN | CNN |
| Training\_acc | **0.7725** | 0.7704 |
| Training\_loss | **0.6434** | 0.6459 |
| Validation\_acc | 0.6596 | **0.6873** |
| Validation\_loss | 1.0526 | **1.0140** |
| Testing\_acc | 0.6899 | **0.6915** |
| Testing\_loss | **1.0077** | 1.0490 |

**MLP**：与CNN类似，BN层没有显著对模型的acc、loss结果有所影响，但是BN层加快了网络的收敛速度。与CNN不同的是，由于MLP模型的过拟合现象比较严重，BN层的存在还一定程度上减缓了过拟合趋势（两者的validation loss曲线）。



图五 MLP网络有无BN层的training, validation, testing acc对比



图六 MLP网络有无BN层的training, validation, testing loss对比

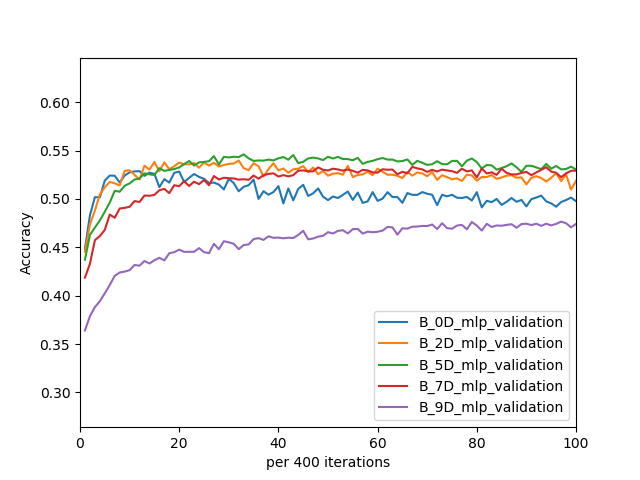
|  |  |  |
| --- | --- | --- |
| 网络 | MLP+BN | MLP |
| Training\_acc | 0.6946 | **0.7069** |
| Training\_loss | 0.8632 | **0.8229** |
| Validation\_acc | **0.5305** | 0.5216 |
| Validation\_loss | **1.5434** | 1.6552 |
| Testing\_acc | **0.5416** | 0.5321 |
| Testing\_loss | **1.3302** | 1.3657 |

但是10月22日lecture中博士学长讲述，BN层目前在CNN中的作用还是一个未知数，它的理论预测作用似乎与实际作用并不相符。具体BN在网络中的作用除了上述可能，其余有待商榷。

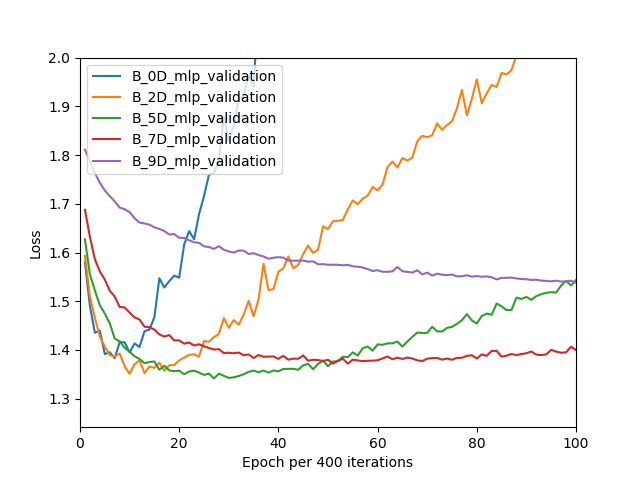
**2.5 droping-rate层对MLP与CNN模型在处理cifar-10问题上的性能影响**

为了研究这个问题，我设置了五组drop-rate值，分别是{0, 0.2, 0.5, 0.7, 0.9}，分别记为0D，2D，5D，7D，9D。其中，0D是不加drop-out层的结果。对于MLP模型与CNN模型（均含有BN），我分别进行了实验，获得了以下的loss与acc曲线。

**MLP**：从图中可以看出，当drop-rate大致在0.5左右的时候，MLP网络的validation acc能够达到比较高的程度，同时它的loss也是同类中相比最低的。如果drop-rate过大，MLP的感知能力会有明显受阻，acc显著减低，loss显著变大；如果不添加drop-rate，过拟合会限制模型的性能提升。其原因或许是由于我们建立的单隐藏层MLP的感知能力有限，比较容易出现过拟合问题，而drop-out层能够通过减低MLP的感知“强度”，即削弱mlp对于线性全连接特征的感知，来避免过拟合，从而提高测试的acc与降低loss。



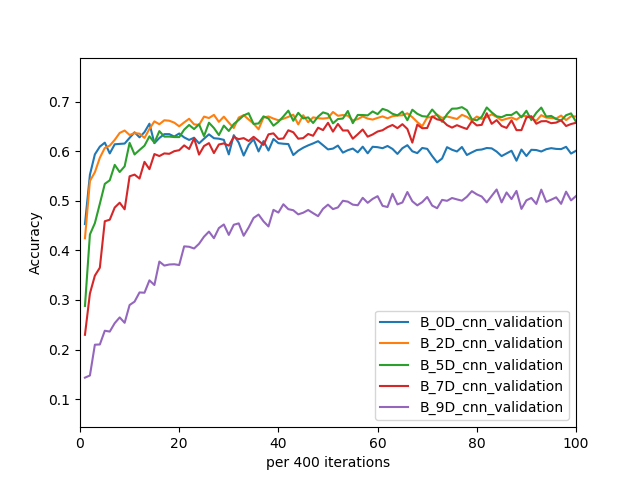
图七 MLP网络不同drop rate条件下validation acc对比



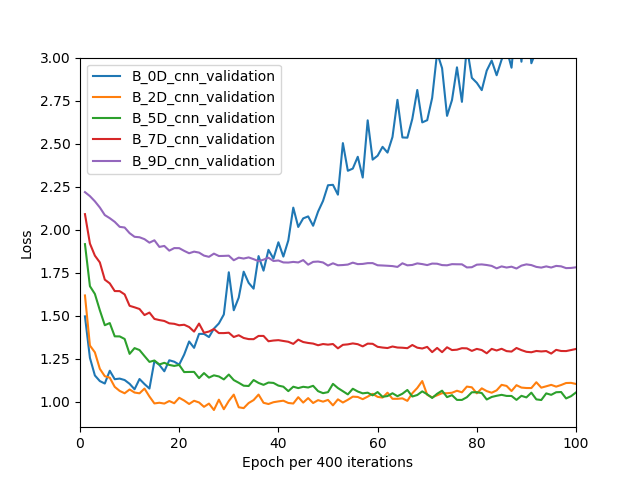
图八 MLP网络不同drop rate条件下validation loss对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MLP网络drop rate | 0 | 0.2 | 0.5 | 0.7 | 0.9 |
| Training\_acc | **0.9577** | 0.8630 | 0.6946 | 0.5604 | 0.3944 |
| Training\_loss | **0.1264** | 0.4004 | 0.8632 | 1.2275 | 1.6849 |
| Validation\_acc | 0.4977 | 0.5187 | **0.5305** | 0.5293 | 0.4741 |
| Validation\_loss | 3.2886 | 2.0836 | 1.5434 | **1.3999** | 1.5381 |
| Testing\_acc | 0.5282 | 0.5402 | **0.5416** | 0.5333 | 0.4787 |
| Testing\_loss | 1.3852 | 1.4320 | **1.3302** | 1.3580 | 1.5198 |

**CNN**：对于CNN模型，其感知能力远强于MLP的感知能力，本身CNN出现过拟合的线性在较低的drop rate下就能够被消除，倘若drop-rate如果过大（如0.9），CNN丢失了许多感知能力，降低了其本身的性能，表现CNN的validation acc降低，loss升高。与MLP类似，在drop-rate约等于0.5左右时，CNN的网络性能达到最优。



图九 CNN网络不同drop rate条件下validation acc对比



图十 CNN网络不同drop rate条件下validation loss对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| CNN网络drop rate | 0 | 0.2 | 0.5 | 0.7 | 0.9 |
| Training\_acc | **0.9821** | 0.8768 | 0.7725 | 0.6964 | 0.5428 |
| Training\_loss | **0.0535** | 0.3396 | 0.6434 | 0.8656 | 1.2836 |
| Validation\_acc | 0.6011 | **0.6704** | 0.6596 | 0.6576 | 0.5097 |
| Validation\_loss | 3.2527 | 1.1021 | **1.0526** | 1.3055 | 1.7805 |
| Testing\_acc | 0.6437 | 0.6791 | **0.6899** | 0.6652 | 0.5241 |
| Testing\_loss | 1.1039 | **0.9695** | 1.0077 | 1.2848 | 1.7724 |

**2.6 training loss与validation loss的区别原因**

现象是training loss远低于validation loss。原因是过拟合现象的存在，如2.2中讨论的，无论是感知能力特别有限的MLP还是CNN，在cifar-10任务中其能够感知到的sample的特征是有限的，在训练中，训练得到的结果是对于其感知到的特征是最优的，但是对于未感知到的特征，拟合得越好感知性越差。即便是用同样的sample去测试，模型的training loss也会低于validation loss。另外，2.1中也讨论了为何training与test需要分别进行，分别进行时因为BN与drop-out的性能不同，得出来的loss自然不同：在training时，数据集被进行了归一化与drop，在testing时并无如此操作，得出来的loss是不一样的。

调大drop-out值（0.5左右为宜）能够帮助缓解过拟合现象。

**2.7 CNN卷积层、池化层对性能的影响**

此部分实验在GPU上运行，但是由于没有及时下载数据，没法获得图像与结果，只能够定性阐述。

在我的实验中，卷积层的stride（x, y）的第二个参数y有2，4，8等选择，以2、4为最佳；第一个参数x有10，20，40，80等选择，实验结果中以40为最佳。卷积层反映的是CNN的感知能力，一个维度尽可能大（如40）能够获取到二维的信息最多。

池化层有（1，1）（2，2）（4，4）等多种选择，在实验中以（1，1）为最佳，此时事实上没有池化的模糊操作。为了体现池化的意义，我选择了（2，2）将4个像素取平均到一个值，加快了运算效率。