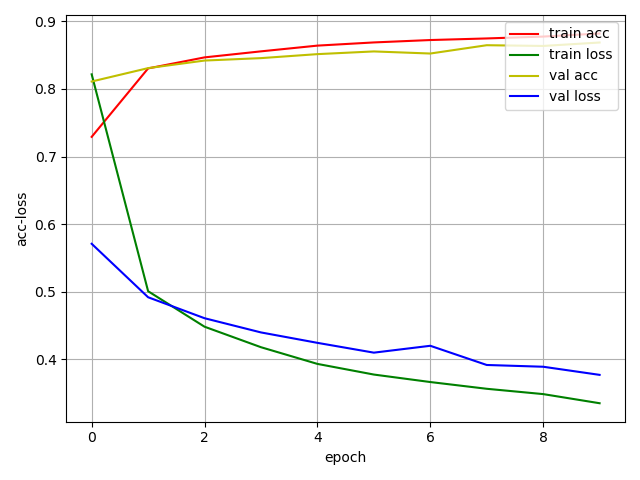
数字系统设计实验一

1. **TensorFlow实验**
2. (1) 该教程用的是DNN（全连接神经网络）结构，包括1个输入层和2个全连接层；损失函数是sparse categorical cross entropy（交叉熵损失）函数；优化方法是Adam。

(2) 常用的损失函数还有：均方误差损失、SVM合页损失、smooth L1损失，等等；

(3) 常用的优化方法还有：梯度下降法（GD，其中常用的是随机梯度下降法SGD）、共轭梯度法（CG）、牛顿法和拟牛顿法、Momentum、AdaGrad、Adadelta、RMSprop、Adam和Adamax、Nadam，等等。

1. 通过回调函数让程序自动绘制loss曲线，结果如下：



其中acc代表accuracy；train代表训练集，val代表测试集。

1. 卷积神经网络，对照组的结构为：

第1层，卷积层，输入28×28×1，卷积核大小5×5，通道数32；

第2层，池化层，池化大小2×2；

第3层，卷积层，卷积核大小5×5，通道数64；

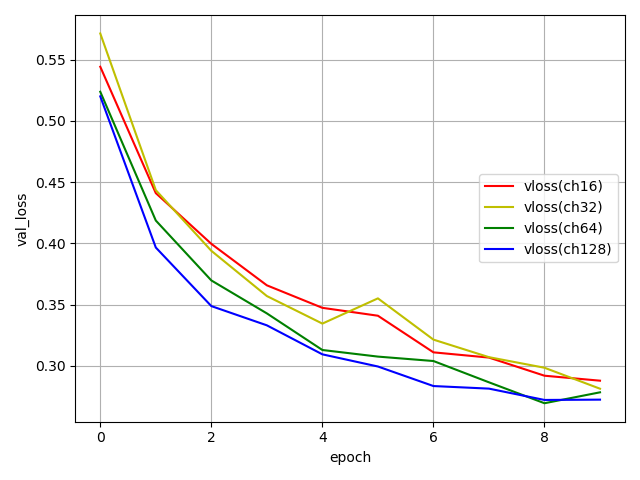
第4层，池化层，池化大小2×2；

第5层，全连接层，输出通道数为1000；

第6层，全连接层，输出通道数为10。

1. 增减卷积层通道数，实践中发现识别准确率很大程度上取决于第5层全连接层的参数数目，故而设计增减卷积层通道数的网络结构时应尽量保持全连接层的参数数目一致。

因此保持第3层通道数不变，将第1层的通道数分别改成16, 64, 128，与对照组的loss曲线作比较，结果如下：



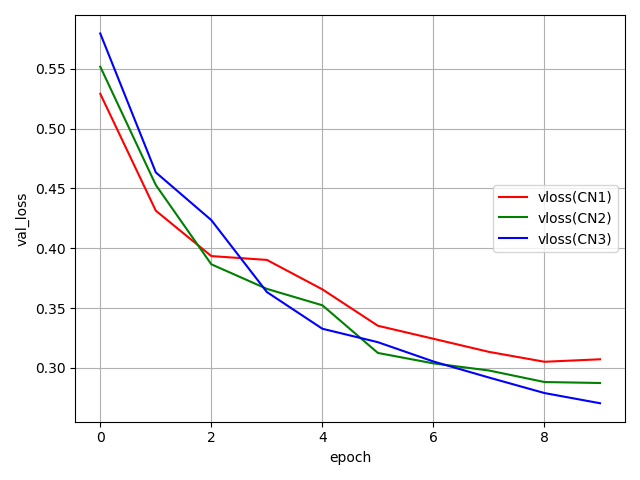
可见增加卷积通道数，loss曲线下降更多，但程度不大；当通道数较多时会出现loss随训练迭代次数反而增大的现象，推测可能与过拟合有关。

分析参数数目、运算量大小和测试集max准确率：

|  |  |  |  |
| --- | --- | --- | --- |
| 卷积层通道数 | 参数数目 | 运算量大小 | 测试集max准确率 |
| 16 | 1 061 090 | 30s | 89.66% |
| **32** | **1 087 106** | **44s** | **90.12%** |
| 64 | 1 139 138 | 73s | 90.17% |
| 128 | 1 243 202 | 155s | 90.46% |

可见随着卷积层通道数的增加，测试集上的最高准确率增加缓慢，运算量却几乎指数上升。

1. 增减卷积层数，确保第5层的参数数目一致；分别将第3、4层去掉，将第1层改成64通道，第二层改成6×6池化，得到一层卷积层的网络；将第4层池化层改成64通道、5×5卷积核的卷积层，得到三层卷积层的网络。与对照组的loss曲线作比较，结果如下：



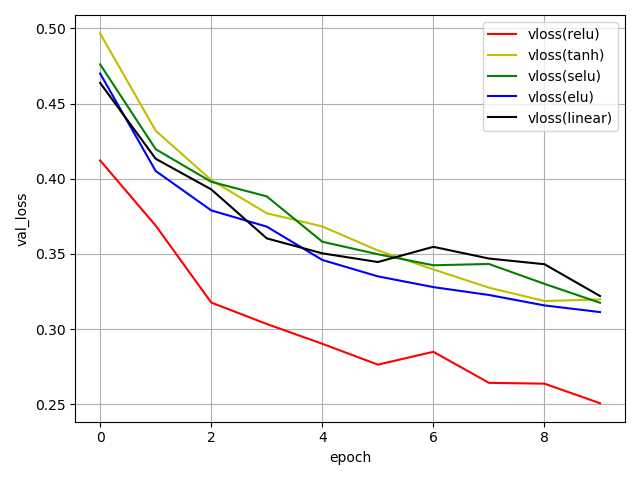
可见增加卷积层数，loss曲线下降更多。

分析参数数目、运算量大小和测试集max准确率：

|  |  |  |  |
| --- | --- | --- | --- |
| 卷积层数 | 参数数目 | 运算量大小 | 测试集max准确率 |
| 1 (645, p6) | 1 036 674 | 30s | 89.00% |
| **2 (325, p2, 645, p2)** | **1 087 106** | **42s** | **89.71%** |
| 3 (325, p2, 645, 645) | 1 189 570 | 50s | 90.60% |

可见随着卷积层数的增加，测试集上的最高准确率的增加较缓慢，运算量的增加也不太过分。

1. 采用不同的激活函数，这里采用简化的卷积神经网络结构，第1层是16通道、3×3卷积核的卷积层，第2层是2×2的池化层，第3层是1000输出通道的全连接层，第4层是10输出通道的全连接层。激活函数分别选取relu, tanh, selu, elu和linear，loss曲线如下：



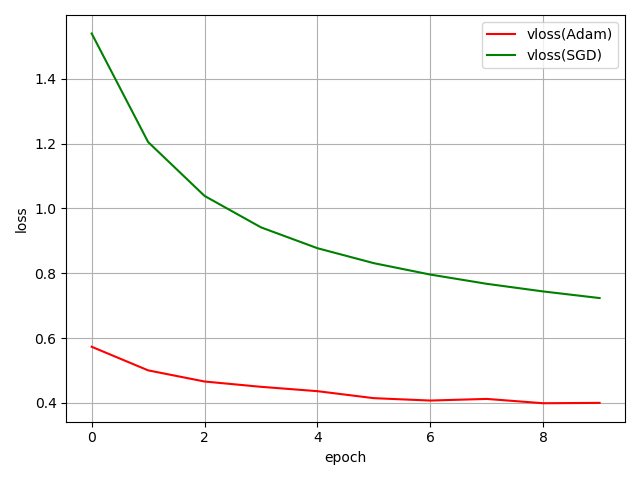
可见relu函数优势明显，其次稍有优势的是elu函数。

分析参数数目、运算量大小和测试集max准确率：

|  |  |  |  |
| --- | --- | --- | --- |
| 激活函数 | 参数数目 | 运算量大小 | 测试集max准确率 |
| relu | 2 715 170 | 23s | 91.00% |
| tanh | 2 715 170 | 23s | 88.68% |
| selu | 2 715 170 | 33s | 88.69% |
| elu | 2 715 170 | 24s | 88.81% |
| linear | 2 715 170 | 21s | 88.53% |

可见准确率上relu优势明显，其他几个函数相差不大；运算量上selu最大，linear最小，但小得不多。

1. 采用教程中的网络结构，分别用Adam和SGD优化方法：



可见Adam的收敛速度比SGD快许多。

1. **Jetson Nano实验**
2. 搭建TensorFlow环境：这部分主要由我的搭档朱结奥同学完成，我只是借用了一下他的成果。主要困难是选择版本正确的TensorFlow和TensorFlow-GPU，以及配置正确的运行环境。
3. 选取任务一第3项中采用relu激活函数的模型，具体结构是：

第1层，卷积层，输入28×28×1，卷积核大小3×3，通道数16；

第2层，池化层，池化大小2×2；

第3层，全连接层，输出通道数为1000；

第4层，全连接层，输出通道数为10。

这个模型的运算量和准确率都比较理想，并且是卷积神经网络，便于测试硬件性能。

在Jetson Nano上使用和不使用GPU，模型的运行速度和准确率分别是：

|  |  |  |
| --- | --- | --- |
| 运行平台 | 运行时间 | 测试集准确率 |
| Nano CPU | 8.98s | 89.80% |
| Nano GPU | 11.11s | 89.80% |

GPU上的运行速度稍慢于CPU上的运行速度，可能是因为Nano上的CPU和GPU不是一个档次的。因为测试模型和测试集都完全相同，所以准确率也相等。

1. 手动设置每次model输入的batch size，同样比较在CPU上和在GPU上的运行速度：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| batch大小 | 2 | 5 | 10 | 32 | 64 | 准确率 |
| Nano CPU | 74.7s | 34.4s | 15.7s | 10.4s | 9.5s | 90.15% |
| Nano GPU | 85.4s | 56.5s | 24.4s | 7.6s | 6.2s | 90.15% |

还可以作出曲线图：

可见当batch较小时，CPU性能好于GPU，batch较大时反之。由于测试模型是CNN结构，说明GPU适合处理大批量、带卷积的神经网络。