



深度学习第一次实验作业报告

课程/2023 深度学习 学号/SA23229086 学生/郎文翀
大数据学院计算机技术班
2023 年 10 月 21 日星期六

1 实验要求

使用 pytorch 或者 tensorflow 手写一个前馈神经网络，用于近似函数：

$$y = \sin(x), x \in [0, 2\pi)$$

并研究网络深度、学习率、网络宽度、激活函数对模型性能的影响。

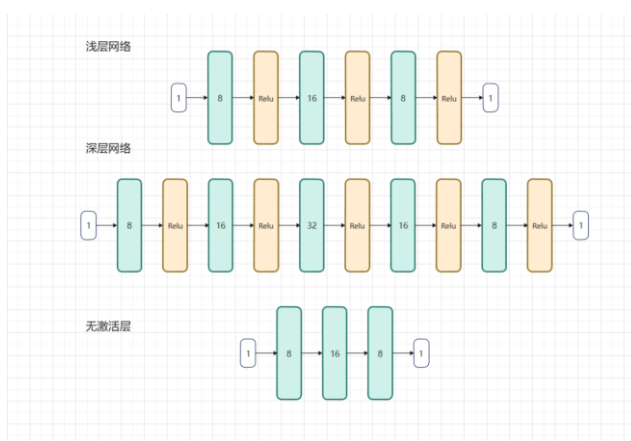
2 实验设计

2.1 设计需求

本次实验选择 Pytorch 实现，并使用 GPU 版本进行训练，最后采用 TensorBoard 完成绘图与实验结果分析，具体程序设计需求如下：

- 数据采样充分：数据集制作完备方便训练与测试。
- 模型参数设置：可以手动设置网络模型宽度与深度。
- 分批次训练：每 Epoch 数据分为 64 一组进行训练。
- 学习率调节：学习率在训练过程中参考 Epoch 进行调节。
- 对比实验：对比不同网络模型深度/宽度/学习率/激活函数对实验结果的影响。

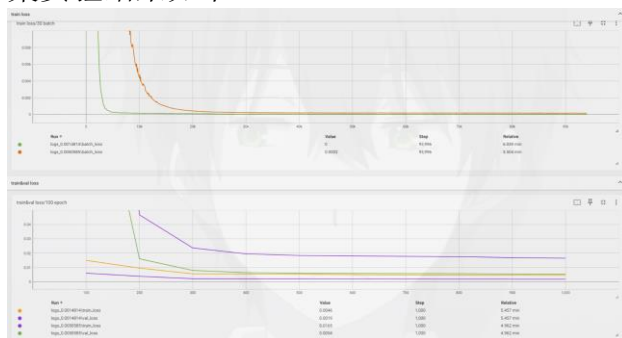
2.2 网络模型图



3 实验分析

3.1 网络深度/宽度对比

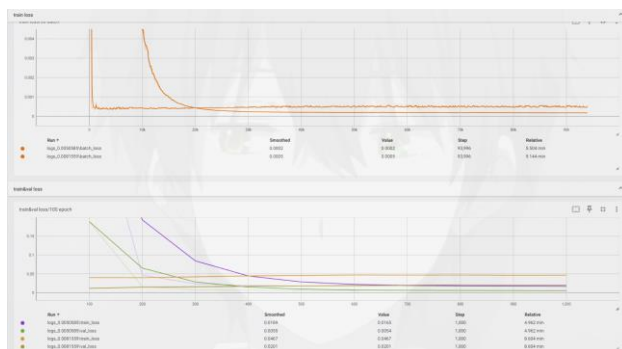
浅层网络与深层网络使用超参数 (Epoch=1000, batch_size=64) 一致，仅在网络模型设计中存在差异，30 step 后的均 batch loss 与 100 epoch 后的训练集与验证集实验结果如下：



经对比发现在 400 epoch/30k step 以后两个网络模型均已收敛，但是深层网络模型在最终测试集的 MSE 损失 (0.0014184) 要略小于浅层网络模型收敛后在测试集上的 MSE 损失 (0.0050585) 说明使用更复杂的网络模型包含更多的参数可以更好的拟合函数。

3.2 学习率调节对比

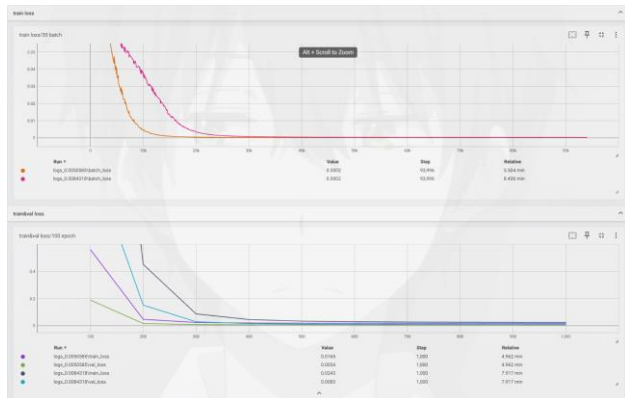
在学习率对比实验中首先分别选用初始值为 $1e-4$ 和 0.01 ，训练过程不改变学习率的实验结果：



发现初始值为 0.01 学习率的网络模型在训练末阶段 MSE 损失不降反升，这是因为学

习率设置过大导致模型无法收敛，因此学习率初始值要缩小，调整至 $1e-4$ 后得到正确的损失曲线。

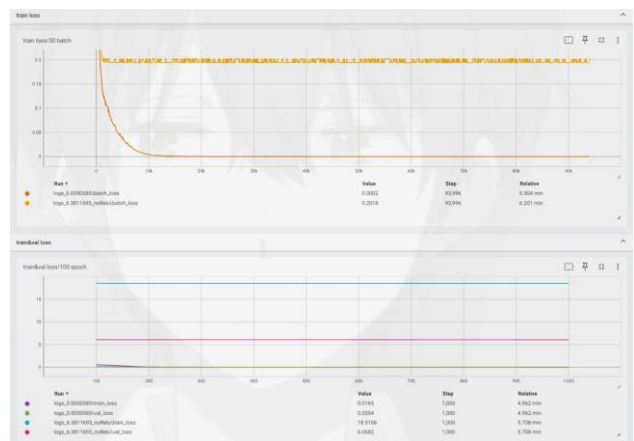
同时还对比了加入学习率优化器对实验结果的影响，实验结果如下：



对比发现加入学习率优化器后 MSE 损失更大，推测可能是因为学习率后期变小后造成模型学习过拟合训练集从而导致在测试集

上反而表现不佳虽然其在验证集上损失更小。

3.3 激活层对比



对比发现加入激活层 ReLU 前后的 MSE 损失可以明显观察到加入激活层后网络模型才能收敛有效拟合原函数。

4. Requirements.txt

```
abs1-py==2.0.0
autopep8==2.0.4
cachetools==5.3.1
certifi==2023.7.22
charset-normalizer==3.3.0
colorama==0.4.6
contourpy==1.1.1
cycler==0.12.1
fonttools==4.43.1
google-auth==2.23.3
google-auth-oauthlib==1.1.0
grpcio==1.59.0
idna==3.4
importlib-metadata==6.8.0
importlib-resources==6.1.0
kiwisolver==1.4.5
Markdown==3.5
MarkupSafe==2.1.3
matplotlib==3.8.0
netron==7.2.5
numpy==1.26.1
oauthlib==3.2.2
packaging==23.2
Pillow==10.1.0
protobuf==4.23.4
pyasn1==0.5.0
```

```
pyasn1-modules==0.3.0
pycodestyle==2.11.1
pyparsing==3.1.1
python-dateutil==2.8.2
requests==2.31.0
requests-oauthlib==1.3.1
rsa==4.9
six==1.16.0
tensorboard==2.15.0
tensorboard-data-server==0.7.1
tomli==2.0.1
tqdm==4.66.1
typing_extensions==4.8.0
urllib3==2.0.7
Werkzeug==3.0.0
zipp==3.17.0
```