

深度学习课程实验1

SA2011050

吕瑞

2022/10/28

实验要求

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

$$y = \sin x + e^{-x}, x \in [0, 4\pi)$$

实验步骤

数据生成

prepare_darta.py

```
1 def sinx_ex(start, end, num_samples):
2     """生成  $y = \sin x + e^{-x}$  的数据集"""
3     x = torch.empty(num_samples, dtype=torch.float32).uniform_(start, end)
4     y = torch.sin(x) + torch.exp(-x)
5     ...
6
7 sinx_ex(0.0, 4*math.pi, 100000)
```

按照 6:2:2 的比例划分训练集、验证集、测试集

模型搭建

model.py

使用 pytorch 网络框架，继承 `torch.nn.Module` 类。激活函数使用 `torch.nn.Tanh()`

模型训练

train.py

损失函数: `nn.MSELoss()`

优化器: `optim.Adam(model.parameters(), lr=1r)`

```
1
2 def train(dataloader, model, loss_fn, optimizer, device):
3     # size = len(dataloader.dataset)
4     model.train()
5     for batch, (x,y) in enumerate(dataloader):
```

```

6         x, y = x.to(device), y.to(device)
7
8         # Compute prediction error
9         pred = model(x)
10        loss = loss_fn(pred, y)
11
12        # Backpropagation
13        optimizer.zero_grad()
14        loss.backward()
15        optimizer.step()
16

```

调参分析

train.py

```

1  # Set parameters
2  batch_size=512
3  width_list=[8,16,32]
4  depth_list=[1, 2]
5  lr_list = [0.8, 0.1, 1e-3]
6  epochs = 100

```

learning rate = 0.8: 所有的实验结果都在较大的loss处震荡, 如 width=8,depth=1 的情况下:

```

1  Epoch 100
2  -----
3  Avg valid loss 0.735035, Accuracy: 6.3%

```

learning rate = 0.1

depth\width	8	16	32
1	0.000195(best)	0.018375	0.258770
2	0.000442	0.000693	0.105980

learning rate = 1e-3

depth\width	8	16	32
1	0.138826	0.339530	0.286818
2	0.310492	0.079219	0.062479

最优模型测试结果:

```

1  data size = 10000 train valid test 6000 2000 2000 saved in my_data
2  test loss: 0.647263, test accuracy: 10.4%

```

从实验结果可以看出:

- 网络越宽，深度越深，网络越复杂，可学习的参数越多，模型收敛需要学习数据的次数会更多
- 学习率较大时，模型的学习结果会无法收敛，表现为 `loss` 值在某个数值处震荡。学习率较小时，模型需要训练更多的步数以达到收敛。
- `batch_size` 表示每次更新梯度需要用到的数据样本数量，它影响模型学习数据的速度，显然，`batch_size` 越大，模型学完一遍数据集所用的时间就越少。

参考资料

[Welcome to PyTorch Tutorials — PyTorch Tutorials 1.12.1+cu102 documentation](#)