
高级量化交易技术

闫涛
科技有限公司
北京 2021.05.08
{yt7589}@qq.com

第零篇深度学习

第 3 章模型训练

Abstract

在本章中我们将详细讲解用于金融交易的 Transformer 网络的模型训练和预测过程。

1 模型训练与预测概述概述

1.1 训练过程

下面我们来看模型的训练过程，训练入口程序如下所示：

```

1  def train(self):
2      cmd_args = self.parse_args()
3      stock_symbol = 'sh600260'
4      batch_size = cmd_args.batch_size
5      NUM_CLS = 3
6      cmd_args.embedding_size = 5
7      seq_length = 11
8      cmd_args.num_heads = 4
9      cmd_args.depth = 6 # 原始值为2
10     train_iter, test_iter = self.load_stock_dataset(
stock_symbol, batch_size)
11     cmd_args.num_heads = 8
12     model = FmtsTransformer(emb=cmd_args.embedding_size, heads
=cmd_args.num_heads, depth=cmd_args.depth, \
13                             seq_length=seq_length, num_tokens=cmd_args.
vocab_size, num_classes=NUM_CLS, \
14                             max_pool=cmd_args.max_pool)
15     model.to(self.device)
16     opt = torch.optim.Adam(lr=cmd_args.lr, params=model.
parameters())
17     sch = torch.optim.lr_scheduler.LambdaLR(opt, lambda i: min
(i / (cmd_args.lr_warmup / cmd_args.batch_size), 1.0))
18     if cmd_args.continue_train:
19         e, model_dict, optimizer_dict = self.load_ckpt(self.
ckpt_file)
20         model.load_state_dict(model_dict)
21         opt.load_state_dict(optimizer_dict)
22     # training loop
23     cmd_args.num_epochs = 3
24     seen = 0
25     # early stopping 参数
26     best_acc = -1
27     acc_up = 0.0
28     min_acc_up = 0.000001 # 识别为精度提高的最小阈值
29     non_acc_up_epochs = 0 # 目前多少个epoch精度未提高

```



```
68         break
69         print(f'-- {"test" if cmd_args.final else "validation"} accuracy {acc:.3}')

```

Listing 1: 模型训练入口

代码解读如下所示：

- 第 5 行：Transformer 网络输出三种市场状态：上涨、下跌、震荡；
- 第 6 行：我们将每个时刻等价为一个单词，每个时刻可以用开盘、最高、最低、收盘、交易量来表示，因此相当于每个单词是 5 维向量；
- 第 7 行：我们通常考虑当前时刻，同时向前看 10 个时刻，因此每个样本包括 11 个时刻，相当于每个句子有 11 个单词，所以序列长度为 11；
- 第 8 行：共有 6 层结构；
- 第 10 行：共有 8 个自注意力；
- 第 12~15 行：初始化模型，其中 vocab_size=50000 为缺省值，max_pool 代表使用最大池化，并将其放入 GPU 中；
- 第 16 行：使用 adam 优化算法；
- 第 17 行：使用带有 warmup 的学习调整计划算法：...；
- 第 18~21 行：在之前训练的基础上继续训练；
- 第 22 行：训练集训练遍数；
- 第 23 行：共训练了多少个样本；
- 第 26 行：当前所取得的最佳精度；
- 第 27 行：当前精度提高的数值；
- 第 28 行：视为精度有提高的最小阈值，用于避免扰动情况；
- 第 29 行：目前已经有多少个 epoch 精度没有提高，如果达到最大允许的精度没有提高的 epoch 数，则停止训练过程；
- 第 30 行：允许连续多少个 epoch 精度没有显著提高；
- 第 31 行：训练指定个 epoch；
- 第 33 行：将模型置为训练状态；
- 第 34 行：读出训练集的每个迷你批次，循环进行处理；
- 第 35 行：所有参数的梯度设置为 0；
- 第 36 行：获取样本集和标签集，具体实现细节将在后面详细讲解；
- 第 37 行：调用 FmtsTransformer 模型计算网络输出结果；
- 第 38 行：使用负对数似然函数为代价函数，计算代价函数的值；
- 第 39 行：将代价函数值反向传播，求出梯度值；
- 第 42、43 行：当梯度所组成的向量长度大于 1 时，调整梯度值，使长度变为 1；
- 第 44 行：调用优化器对参数进行优化；
- 第 45 行：调用学习率变化方法，按需调整学习率；
- 第 46 行：统计当前学习了多少个样本；

- 第行::
- 第行::
- 第行::
- 第行::
- 第行::
- 第行::
- 第行::
- 第行::

1.2 预测过程

2 总结

