高级量化交易技术

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

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

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：

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

```
30          max_no_acc_up_epochs = 50 # 如果精度在这些epoch后还没提高
    则终止训练过程
31          for epoch in range(cmd_args.num_epochs):
32              print(f'\n epoch {epoch}')
33              model.train(True)
34              for batch in tqdm.tqdm(train_iter):
35                  opt.zero_grad()
36                  X, y = self.get_stock_batch_sample(batch,
    batch_size, cmd_args.embedding_size)
37                  y_hat = model(X)
38                  loss = F.nll_loss(y_hat, y)
39                  loss.backward()
40                  # clip gradients
41                  # - If the total gradient vector has a length > 1,
     we clip it back down to 1.
42                  if cmd_args.gradient_clipping > 0.0:
43                      nn.utils.clip_grad_norm_(model.parameters(),
    cmd_args.gradient_clipping)
44                  opt.step()
45                  sch.step()
46                  seen += X.size(0)
47              with torch.no_grad():
48                  model.train(False)
49                  tot, cor= 0.0, 0.0
50                  for batch in tqdm.tqdm(test_iter):
51                      X, y = self.get_stock_batch_sample(batch,
    batch_size, cmd_args.embedding_size)
52                      y_hat = model(X).argmax(dim=1)
53                      tot += float(X.size(0))
54                      cor += float((y == y_hat).sum().item())
55                  acc = cor / tot
56                  # 获取当前最佳测试集精度, 并保存对应的模型
57                  if best_acc < acc:
58                      acc_up = acc - best_acc
59                      if acc_up > min_acc_up:
60                          best_acc = acc
61                          non_acc_up_epochs = 0
62                          print('保存模型参数')
63                          self.save_ckpt(self.ckpt_file, epoch,
    model, opt)
64                  else:
65                      non_acc_up_epochs += 1
66                      if non_acc_up_epochs > max_no_acc_up_epochs:
67                          print('模型已经处于饱合状态, 停止训练过程'
    )
```

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

Listing 1: 模型训练入口

1.2 预测过程

2 总结

3 附录 X