第3次作业\_神经网络

电 25 吴晨聪 2022010311

# （1） 读懂代码，目前的代码输入特征是什么维度，有哪些？输出标签是什么维度，有哪些？

按文件夾分类:

1. ***Data\_Solver:***

data\_loader: 将数据按比例分割为训练和测试集，对数据进行预处理步骤，使得数据便于 LSTM 模型输入。

Quantile\_loss: 自定义的量化损失函数，用于训练 LSTM 模型时的误差计算，可以帮助模型学会预测特定量化水平下的目标值，为负载预测任务提供准确性和不确定性分析。

reshape\_data: 将原始数据转换成包含日期、时间和其他特征的适合 LSTM 模型使用的特征矩阵，以便用于 LSTM 负载预测模型，增强模型对时间依赖特征的捕捉能力。

to\_supervised: 将时间序列数据转换成 LSTM 模型的监督学习形式，即输入-输出对。这种转换将输入的时间序列数据分段，使模型可以学习在给定的输入时段后预测未来的目标值。

1. ***Model:***

实现了一个基于 LSTM 的时间序列预测模型，包含

模型结构定义: \_\_init\_\_(self, input\_size, hidden\_size, number\_layer, output\_size, output\_layer)

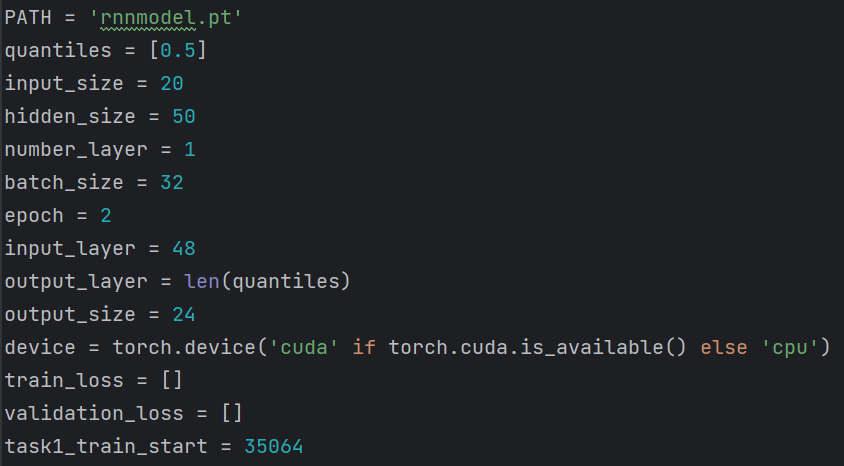
前向传播方法: forward(self, x)

预测方法: off\_predict(self, x)

训练过程: train(self, device, train\_data, validation, loss\_function, optimizer, batch\_size, epoch, train\_loss

1. ***pre\_train:***

整个 LSTM 负载预测项目的主程序，用于加载配置、准备数据、训练模型并可视化结果，它集成了之前定义Data\_Solver和Model活页夹中的所有函数和模型以及其他数据处理函数。



超参数设置: 包括隐藏层大小、批次大小、学习轮数（epoch）、输入/输出层的维度等。



模型初始化与训练: 调用 model.train 方法执行模型训练，每个 epoch 计算训练和验证损失，并记录在 train\_loss 和 validation\_loss 列表中。

***输入特征***

特征维度: 在 DataLoader 类中，将原始数据转换为 reshapeddata，并定义了输入特征的维度为 20（input\_size = 20）。

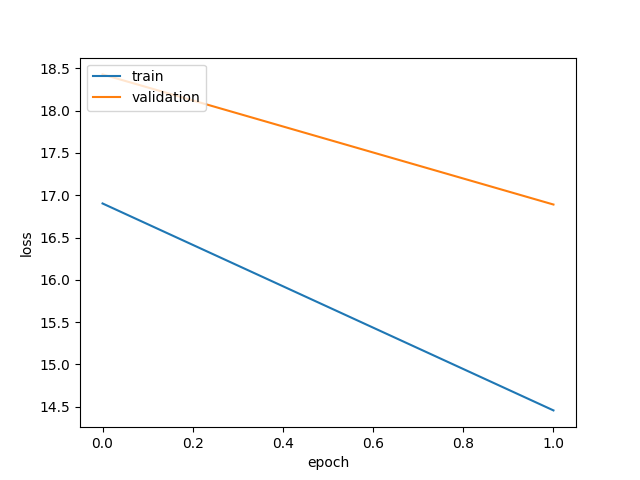
特征内容：负载值(1个)、星期(7个)、月份(12个)。

***输出标签***

标签维度: 标签的维度是 (batch\_size, 24)，其中 24 是预测的未来时间步数 (n\_out=24)，每个时间步只预测一个值（即未来负载值）。

标签内容：未来 24 个时间步的负载值

Python运行结果:

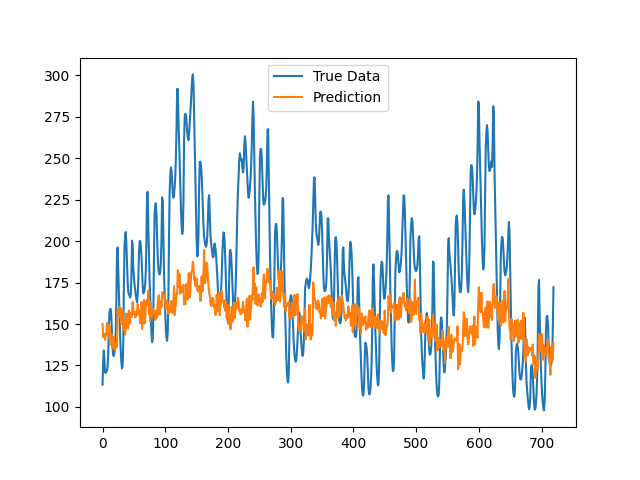
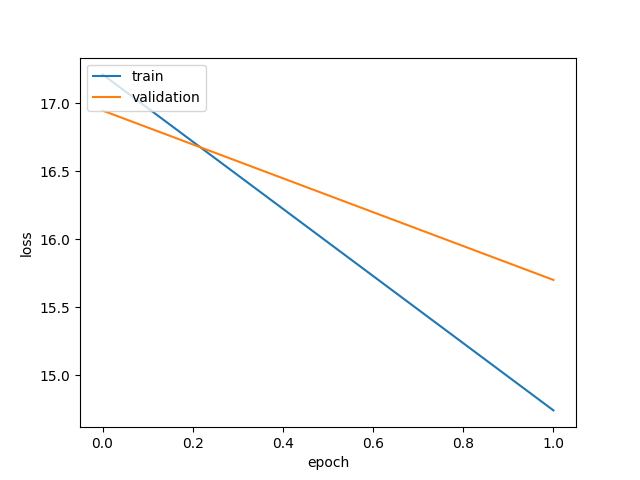
一張含有 螢幕擷取畫面, 文字, 繪圖, 圖表 的圖片

自動產生的描述

# （2） 正常的机器学习数据集应该有训练集、验证集、测试集三部分组成，目前代码仅划分了训练集和验证集，请大家读懂代码，自己增加测试集上预测精度评估的功能。

在 DataLoader 类的初始化中对数据进行三分割：训练集、验证集和测试集，其后对测试集数据进行缩放和分段，与训练集和验证集保持一致，最后在训练和验证完成后，加载训练好的模型，在测试集上生成预测，并评估测试集的精度。

Python运行结果:

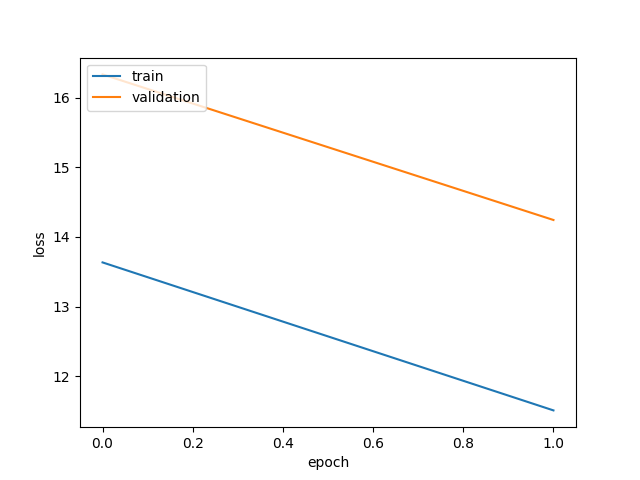
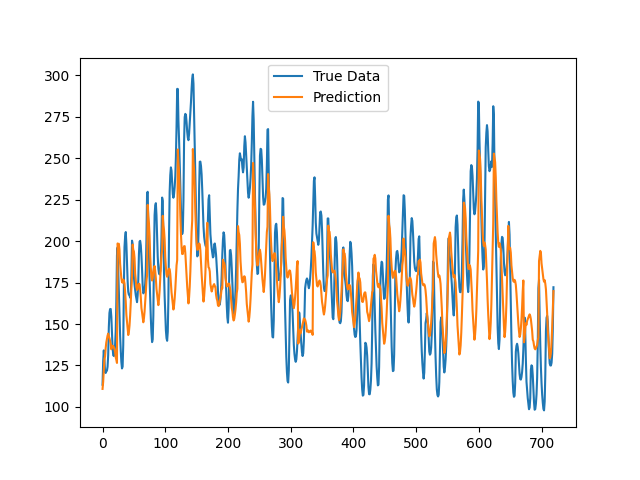


上图验证损失显着高于训练损失，说明模型在加入预测精度评估功能后可能存在过拟合，同时预测的精度也没有明显的提高。

**目前的模型预测精度可能并不令人满意，请大家对代码进行修改，提高模型精度，注意过拟合和欠拟合现象。**

将训练的优化器optimizer 选取为optim.Adam，学习率lr 设置为0.01，同时令模型的训练次数为2、每次训练样本数为16（这里如果令训练次数再增加，训练样本数再减少则可能会发生过拟合的现象，而且所消耗的时间也会大大增加），可以得到更好的训练结果。

Python运行结果:

此时的训练误差、验证误差相较开始已经下降了很多，预测的值也能基本跟上实际值的变化。