

# **《智能信息网络》课程**

## **案例分析与实践报告**

**(第三次)**

**题目：网络流量智能预测实践报告**

**场景：网络智能优化**

**姓名：徐川峰**

**班级：2019211112**

**学号：2021523016**

**日期：2022.06.20**

# 1、研究背景与问题描述

## 1.1 背景需求

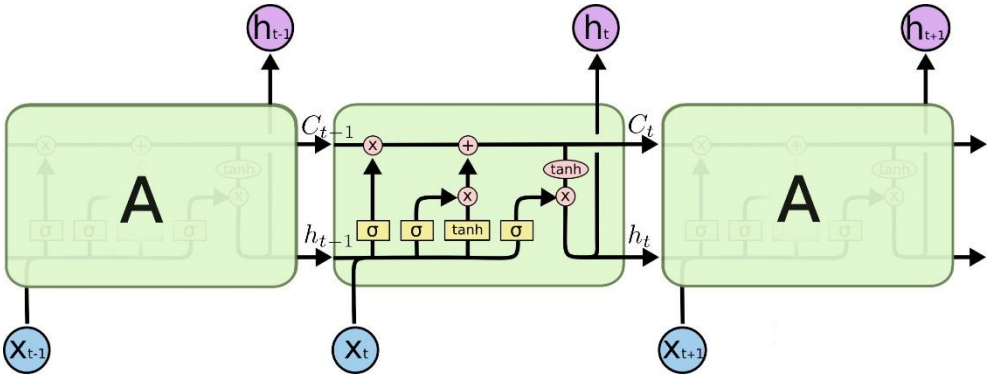
随着网络通信技术的快速发展，新的网络业务与网络应用不断增加,这对网络的服务质量，流量控制和网络管理均提出了很高的要求。而网络带宽资源有限，对网络流量进行预测可以有效提高带宽资源利用率。

## 1.2 问题描述

现有 2102 条已知的用户历史流量数据，需要用前 1908 条的数据来预测后 194 条的流量数据。

## 1.3 LSTM

LSTM 的全称是 Long Short Term Memory，顾名思义，它具有记忆长短期信息的能力的神经网络。LSTM 首先在 1997 年由 Hochreiter & Schmidhuber 提出，由于深度学习在 2012 年的兴起，LSTM 又经过了若干代大牛(Felix Gers, Fred Cummins, Santiago Fernandez, Justin Bayer, Daan Wierstra, Julian Togelius, Faustino Gomez, Matteo Gagliolo, and Alex Graves)的发展，由此便形成了比较系统且完整的 LSTM 框架，并且在很多领域得到了广泛的应用。

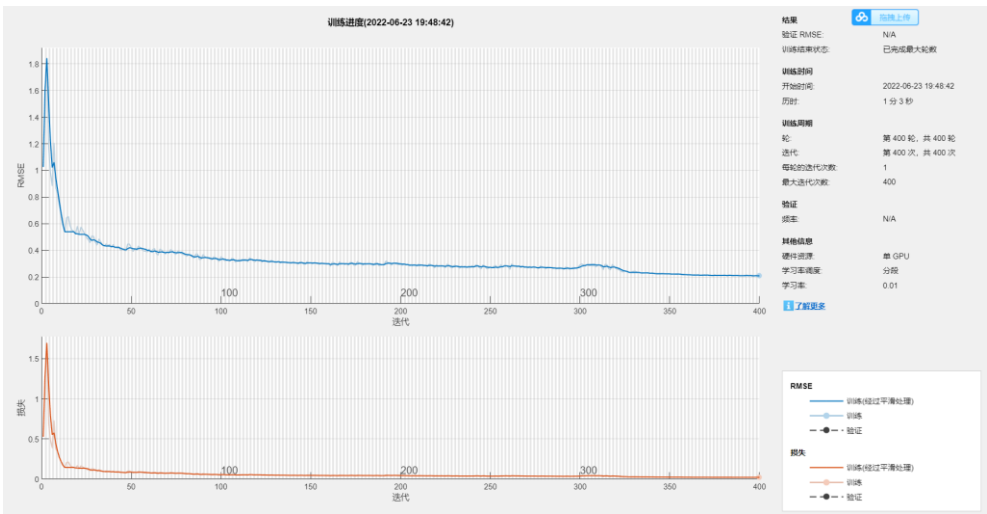


# 2、总体思路与方案论证

## 2.1 评估方法

- ①采用 LSTM 预测方法，利用前 1908 条的流量数据完成长期记忆模型的训练；
- ②用训练完成的预测模型对接下来的 194 条的用户数据进行预测；
- ③将预测得到的数据和已知的 194 条的用户数据进行对比，评估流量预测模型。

## 2.2 MATLAB 训练过程



### 3、关键技术研究是实现

#### 3.1 数据输入

```
f = xlsread('历史流量数据下载.xlsx','SheetJS','B1:B1908');
figure(1)
plot(f)
dataTrain=f(1:end,:);
```

#### 3.2 标准化数据

为了获得较好的拟合并防止训练发散，将训练数据标准化为具有零均值和单位方差。在预测时，必须使用与训练数据相同的参数来标准化测试数据。

```
mu = mean(dataTrain);
sig = std(dataTrain);
dataTrainStandardized = (dataTrain - mu) / sig;
```

#### 3.3 准备预测变量和响应

要预测序列在将来时间步的值，请将响应指定为将值移位了一个时间步的训练序列。也就是说，在输入序列的每个时间步，LSTM 网络都学习预测下一个时间步的值。预测变量是没有最终时间步的训练序列。

```
X = dataTrainStandardized(1:end-1);
Y = dataTrainStandardized(2:end);
XTrain = X';
YTrain = Y';
```

#### 3.4 定义 LSTM 网络架构

创建 LSTM 回归网络。指定 LSTM 层有 10 个隐含单元。

```
numFeatures = 1;
numResponses = 1;
numHiddenUnits = 20;%经过反复调参后 发现设置值为 20 效果较好! (*^▽^*)

layers = layerGraph();
TempLayers = [
    sequenceInputLayer(numFeatures)
    lstmLayer(numHiddenUnits, "Name", "lstm") % LSTM 层
    reluLayer("Name", "relu") % 学习函数
    fullyConnectedLayer(50, "Name", "fc1")
    dropoutLayer(0.2)
    fullyConnectedLayer(numResponses, "Name", "fc") % 全连接层(可以理解为单层 BP，作用为整合结果)
    regressionLayer("Name", "regressionoutput")];
layers = addLayers(layers, TempLayers);
```

指定训练选项。要防止梯度爆炸，将梯度阈值设置为 1。指定初始学习率 0.01，在 25 轮训练后通过乘以因子 0.05 来降低学习率。

```
options = trainingOptions('adam', ...  
    'MaxEpochs',400, ...  
    'GradientThreshold',1, ...  
    'InitialLearnRate',0.1, ...  
    'LearnRateSchedule','piecewise', ...  
    'LearnRateDropPeriod',320, ...  
    'LearnRateDropFactor',0.1, ...  
    'Verbose',0, ...  
    'Plots','training-progress');%迭代图
```

### 3.5 训练 LSTM 网络

使用 `trainNetwork` 以指定的训练选项训练 LSTM 网络。

```
net = trainNetwork(XTrain,YTrain,layers,options);
```

### 3.6 预测将来时间步

要预测将来多个时间步的值，使用 `predictAndUpdateState` 函数一次预测一个时间步，并在每次预测时更新网络状态。对于每次预测，使用前一次预测作为函数的输入。使用与训练数据相同的参数来标准化测试数据。

初始化网络状态，先对训练数据 `XTrain` 进行预测。接下来，使用训练响应的最后一个时间步 `YTrain(end)` 进行第一次预测。循环其余预测并将前一次预测输入到 `predictAndUpdateState`。

对于大型数据集、长序列或大型网络，在 GPU 上进行预测计算通常比在 CPU 上快，其他情况下，在 CPU 上进行预测计算通常更快。

```
net = predictAndUpdateState(net,XTrain);  
[net,YPred] = predictAndUpdateState(net,YTrain(end));  
  
for i = 2:194  
    [net,YPred(:,i)] = predictAndUpdateState(net,YPred(:,i-1),'ExecutionEnvironment','cpu');  
end  
  
%使用先前计算的参数对预测去标准化。  
YPred_sum = sig*YPred + mu;  
  
%使用预测值绘制训练时序。  
figure  
plot(dataTrain(1:end))  
hold on  
idx = 1909:(1909+194);  
plot(idx,[f(1908) YPred_sum],'-')
```

```

hold off
xlabel("days")
ylabel("mm")
xlim([0 1909+195])
title("Forecast")
legend(["Observed" "Forecast"])

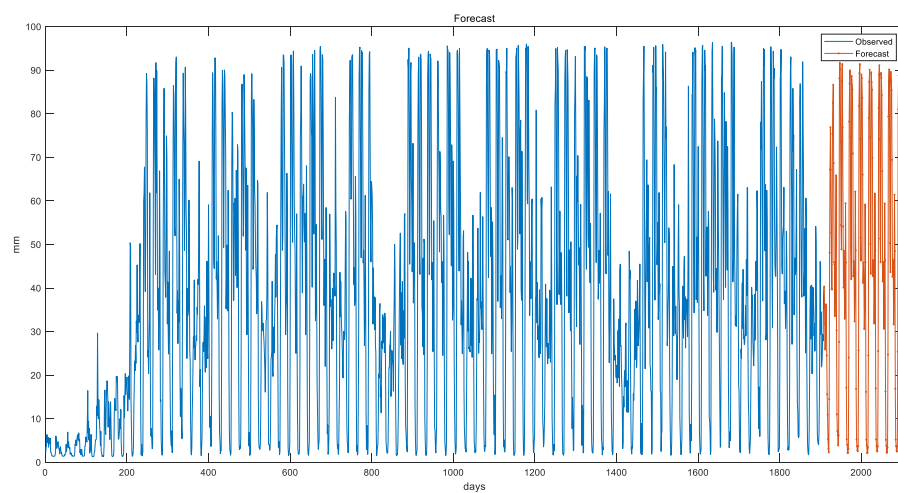
%将预测值与测试数据进行比较。

figure
plot(YPred_sum,'-')
hold off
legend(["Forecast"])
ylabel("data")
title("Forecast")

toc

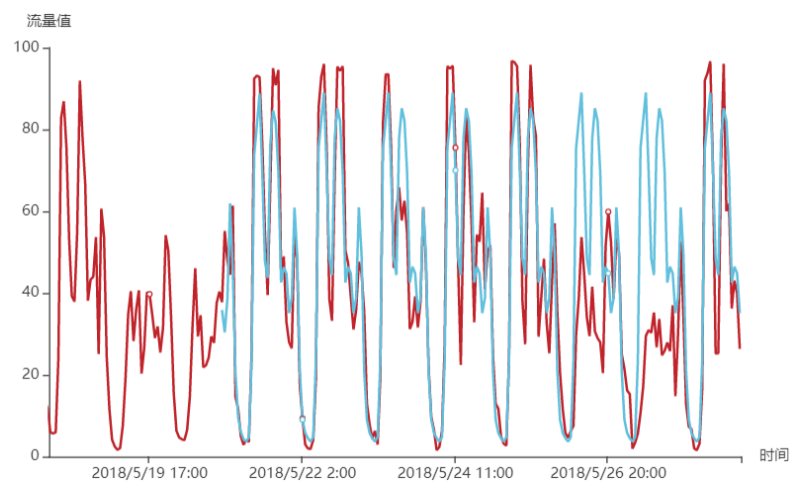
```

### 3.7 预测的数据结果



## 4、结果分析与结论

### 4.1 预测分析图



## 4.2 评估结果

将得到的预测结果进行上传后，根据最大偏差、平均绝对误差、均方误差、均方根误差以及平均绝对百分误差对结果进行评估。



## 5、课程学习收获与体会建议

在本次作业中，我首先观察了历史流量数据：可以发现，这 1908 个数据本身有一定的周期性，能够以时间为轴进行特性划分，使其具备一定的时间序列特性。所以根据时序特性对数据进行挖掘之后，就可以预测它未来未知的走势。

我采用了 LSTM 预测方法，通过多次反复调参 (numHiddenUnits=20, dropoutLayer 设为 (0.2)、MaxEpochs 迭代 400 次)，利用前 1908 条的流量数据完成长期记忆模型的训练，用训练完成的预测模型对接下来的 194 条的用户数据进行预测。

LSTM 可以在波动的时间序列中很好地工作，主要原因是它们具有快速适应趋势中急剧变化的固有能力。但同时我也发现，使用 LSTM 预测时间序列也有一定的局限性。从实验结果可以看出，我输出的是一个很平整的周期数据，并不存在像历史流量数据中会有某一段出现“突降”的情况，这说明，它的精确度会稍低，因为该模型在很大程度上取决于逐步的预测。

此外，通过查询资料我也认识到，这种弊端也可能与数据大小有关：LSTM 是一个神经网络，就像任何神经网络一样，需要大量数据才能对其进行正确训练。

因为自己对于 Python 语言的熟悉度不是很高，所以在本次作业中，我采用了 MATLAB 语言进行模型的设计，那么如何优化模型，也是我课后需要进一步探索的方向。我通过查询了 MATLAB 官网的帮助文档，发现当我们可以访问预测之间的时间步的实际值，就能够尝试使用观测值更新网络状态，重置网络状态可防止先前的预测影响对新数据的预测。当使用观测值而不是预测值更新网络状态时，预测更准确。

另外，也可以通过“删掉干扰数据”或者“把“数据切片”的方法，（我们需要明白这 2000 多组数据本身的周期性，它每过一段时间，流量波动都会减弱），这些有一定的困难和挑战性，课后我会查阅更多的资料进行拓展学习与进一步的思考。

## 附：参考文献

MathWorks Help Center. Time Series Forecasting Using Deep Learning

<https://ww2.mathworks.cn/help/deeplearning/ug/time-series-forecasting-using-deep-learning.html?jsessionid=00863774df4c48a0f74eefb0c4ca>