

基于Python的杭嘉湖平原洪水位预报研究

|  |  |
| --- | --- |
| **姓 名:** | **刘越** |
| **届 别:** | **2023届** |
| **指导教师:** | **李云** |
| **撰写时间:** | **2021年1月** |

# 摘要

本课题以杭嘉湖区为例，采用人工智能的方法，重点研究人类活动剧烈的平原河网地区洪水位预报。通过收集整理近10年杭嘉湖地区日降雨量、太湖日水位、杭嘉湖地区代表站嘉兴站水位及杭嘉湖南排工程排水量等基础数据，基于Python语言，选择BP人工神经网络。构建杭嘉湖地区洪水位预报人工智能模型，选定11场典型降雨场次作为训练样本，用以训练。

首先通过网络初始化，为网络中权重系数赋随机值；其次，输入参数归一化；其三，通过正向传播对网络中各权重系数赋值，其中各神经元阈值函数采用通用sigmoid函数进行处理；第四，通过对比输出值与实测水位结果误差，反向对网络权重系数进行调整，并循环迭代，直至实测与模拟输出值误差低于误差限mse\_final后，基于BP人工神经网络的杭嘉湖洪水位预测模型构建成功。

基于选定的2010年~2019年中11场已发生的典型降雨过程，预报洪水位峰值数据并分析模拟精度误差。采用2020年验证时段6月21日~7月19日场次降雨过程模拟，BP人工神经网络模型计算效率及预报精度都较好。

**关键词：**Python；杭嘉湖平原；洪水位预报；BP人工神经网络模型；神经元节点

# 目 录

[摘要 I](#_Toc16962)

[目 录 II](#_Toc23329)

[正 文 1](#_Toc8957)

[1.前言 1](#_Toc20981)

[1.1研究背景及意义 1](#_Toc14232)

[1.2国内外研究现状 1](#_Toc23674)

[1.3研究思路 1](#_Toc9253)

[2 数据准备与模型选择 3](#_Toc17183)

[2.1 人工神经网络模型选择 3](#_Toc31087)

[2.2 BP人工神经网络洪水位预测 3](#_Toc32054)

[2.3数据准备 4](#_Toc7275)

[3 研究方法 5](#_Toc8993)

[3.1研究区域概况 5](#_Toc30249)

[3.1.1自然地理 5](#_Toc19204)

[3.1.2区域水系 5](#_Toc6222)

[3.1.3水文气象 6](#_Toc27598)

[3.2降雨场次筛选 6](#_Toc8801)

[3.3基于BP神经网络的杭嘉湖洪水位预报模型构建 7](#_Toc480)

[3.3.1网络拓扑结构确定 7](#_Toc22124)

[3.3.2基于Python的人工神经网络编程 8](#_Toc18844)

[4 预测成果对比分析 10](#_Toc32708)

[4.1对于11场次洪水位预测成果分析 10](#_Toc18326)

[4.2 2020年汛期嘉兴洪水位预报分析 11](#_Toc9461)

[5 结论与展望 12](#_Toc21322)

[5.1小结 12](#_Toc1570)

[5.2展望 12](#_Toc16513)

[6 收获与体会 13](#_Toc13150)

[7致谢 13](#_Toc25798)

[参考文献 14](#_Toc8900)

[附录 15](#_Toc24985)

# 正 文

# 1.前言

## 1.1研究背景及意义

长三角平原是我国人口大区之一，产业密集，经济发展较好。也是洪涝灾害严重的地区之一。杭嘉湖平原地处长江三角洲南侧，经济发展水平高，但地面地势低、平，再加上受杭州湾和黄浦江吴淞口潮水感潮影响，排水困难，容易发生洪涝灾害，影响经济发展安全。因此，加强杭嘉湖地区洪水预报，提前做好防洪准备，具有重要意义。

在传统的杭嘉湖地区洪水预报中，主要依靠传统的水文水动力数学模型进行模拟。但传统的水文水动力数学模型计算条件复杂、计算速度相对较慢。

随着人工智能等新兴技术兴起，为杭嘉湖东部平原地区洪水预报研究提供了新的途径。本次研究主要探索人工智能新兴技术在水利行业的应用，为快速响应防汛工作提供支撑。

## 1.2国内外研究现状

上个世纪五十年代，学者麦Carthy在DartmouthSociety上正式提出了AI的概念。但是由于当时的技术制约，人工智能没有大发展。直到近十年来，人工智能技术快速发展。人工智能的学习途径，不需要搞清楚事物发生的原理，而是用巨量的数据和算法来“训练”计算机，研究事物发生的规律。从当前AI技术的发展前景来看，深度学习在人工智能发展过程中扮演着重要角色[1]。

近年来，国内学者通过将人工智能算法应用与洪水预报研究中，并取得了一定成果。有学者研发了对遥感影像检测识别的人工智能算法，构建人工智能和大数据驱动的新一代水文模型，对河南省2016~2018年山洪进行了预报模拟[2]。有学者采用遗传算法(GA)，构建了人工智能模型，湖北一水库14次洪峰洪水总量模拟[3]。有学者采用人工智能BP算法，对太湖湖泊洪水进行了预报模拟[4]。

## 1.3研究思路

本课题以杭嘉湖区为例，采用人工智能的方法，重点研究人类活动剧烈的平原河网地区洪水位预报。通过收集整理近10年杭嘉湖地区日降雨量、太湖日水位、杭嘉湖地区代表站嘉兴站水位及杭嘉湖南排工程排水量等基础数据，基于Python语言，选用合适的人工智能算法，构建杭嘉湖地区洪水位预报人工智能模型，选定典型降雨场次作为训练样本，开展网络模型参数训练与测试。根据未来预报的降雨，运用编写的智能算法预报杭嘉湖地区洪水位，并与实际数据比较，验证本课题研究成果的合理性，为杭嘉湖地区洪水位快速预报提供技术参考。本研究技术路线如图1所示。



图1 课题研究技术路线图

**内容1：选取典型年并完成相关数据收集及初步分析**

收集杭嘉湖区近10年降雨、水位及区域引排资料，初步分析区域降雨、南排水量与杭嘉湖代表站嘉兴站水位相关关系，为人工智能模型提供数据基础。

**内容2：构建杭嘉湖洪水快速预报人工智能模型**

确定神经网络拓扑结构，并依据内容1研究成果确定神经网络输入层及输出层，并根据输入输出层元素确定隐藏层网络框架。基于人工神经网络算法，构建洪水快速预报人工智能模型（如图2所示）。

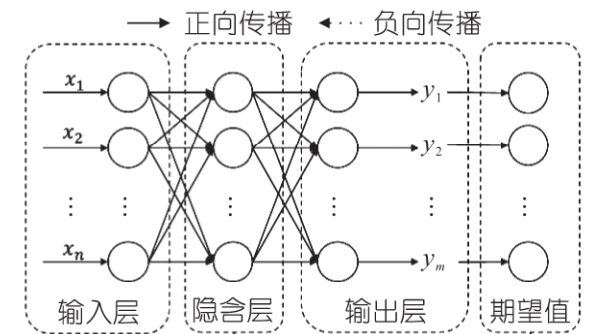


图2 人工神经网络架构示意图

**内容3：预测成果对比分析**

采用2020年汛期典型时段降雨及嘉兴站水位实测数据，采用人工智能模型对嘉兴站水位进行预测预报，并从计算效率及模拟精度方面进行分析，验证本课题研究成果的合理性。

# 2 数据准备与模型选择

## 2.1 人工神经网络模型选择

人工神经网络是计算机机器学习领域的算法之一，会复杂的逻辑运算和非线性映射。人工神经网络的优点在于只需提供一定数量的具有代表性的数据，不需了解这些数据的物理演化信息，也不必担心这些数据存在噪声或错误，仍可建立输入与输出关系。人工神经网络特有的非线性适应性信息处理能力，克服了非结构化等信息处理中传统人工智能方法所存在的不足。得到了广泛应用。

1943年，沃伦·麦卡洛克和沃尔特·皮茨建立了神经网络计算模型。1957年，弗兰克·罗森布拉特发明了加减法构建了两层学习网络。1974年，保罗·韦伯斯采用了误差反向传导的方法，使人工神经网络获得了新的训练方法。1985年，Rumelhart和McClelland构建了反向传播BP网络误差学习算法。2006年，Geoffrey Hinton将人工神经网络改造成了深度学习。

结合本课题训练样本实际，本次采用应用较为广泛且网络结构稳定的BP人工神经网络模型开展地区洪水位预报研究。

## 2.2 BP人工神经网络洪水位预测

BP(Back-Propagation)网络主要采用误差逆向传播算法训练，利用误差对网络中的权重系数和阈值进行不断调整。使得最终计算结果与实际值误差很小，来实现精确模拟。

人工神经网络最重要的组成部分是人工神经元，每一个神经网络都由多组互连的人工神经元构成。

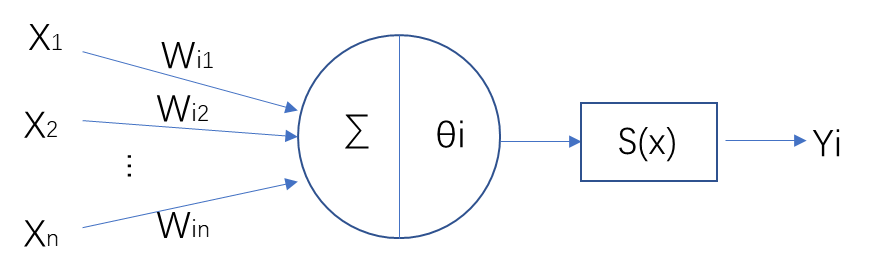
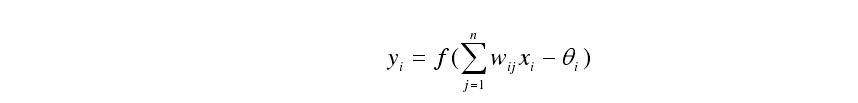


图3 神经元模型示意图

如图3，神经元的输入数据为x1，x2…xn，各神经元权重系数为wi1，wi2…win，神经元的阈值为θi，神经元的激励函数为S（x），神经元的输出数据为yi。神经元的数学表达式为：



BP 神经网络模型有一个输入层，一个隐含层和一个输出层。如图4所示。输入层是x1…xj…xM。隐含层包括权重系数和阈值分析计算，wij是指输入层第i个节点到第j个点的权重系数，（x）是指隐含层中的激励函数，wki是指输出层第k个节点到第i个节点的权重系数，ak是指输出层第k个节点的阈值，（x）是指输出层的激励函数。ok是指输出层第k个节点的输出。

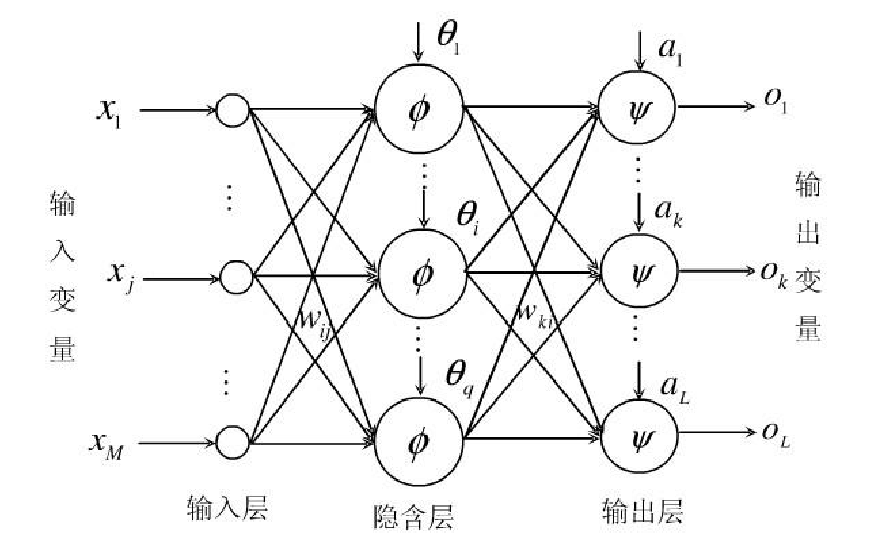


图4 神经网络结构示意图

## 2.3数据准备

通过收集整理杭嘉湖区域降雨、水位及外排水量等数据，获得了该地区2010年~2020年逐日面降雨及嘉兴雨量站点降雨数据共4018条记录，获取杭嘉湖区域水位代表站嘉兴站逐日早8:00报汛水位数据及杭嘉湖南排工程盐官上河闸、盐官枢纽、长山闸、南台头闸、独山闸等工程逐日排水水量数据。

# 3 研究方法

## 3.1研究区域概况

### 3.1.1自然地理

杭嘉湖区位于浙江省北部、江苏省南部、上海市西部地区，是一个省际边界地区，区域总面积7480 km2，其中浙江省所辖面积最多，占80%。杭嘉湖区是平原地区，地形呈北低南高，北部地区地面高程约3.2m，南部地区约6-7m。

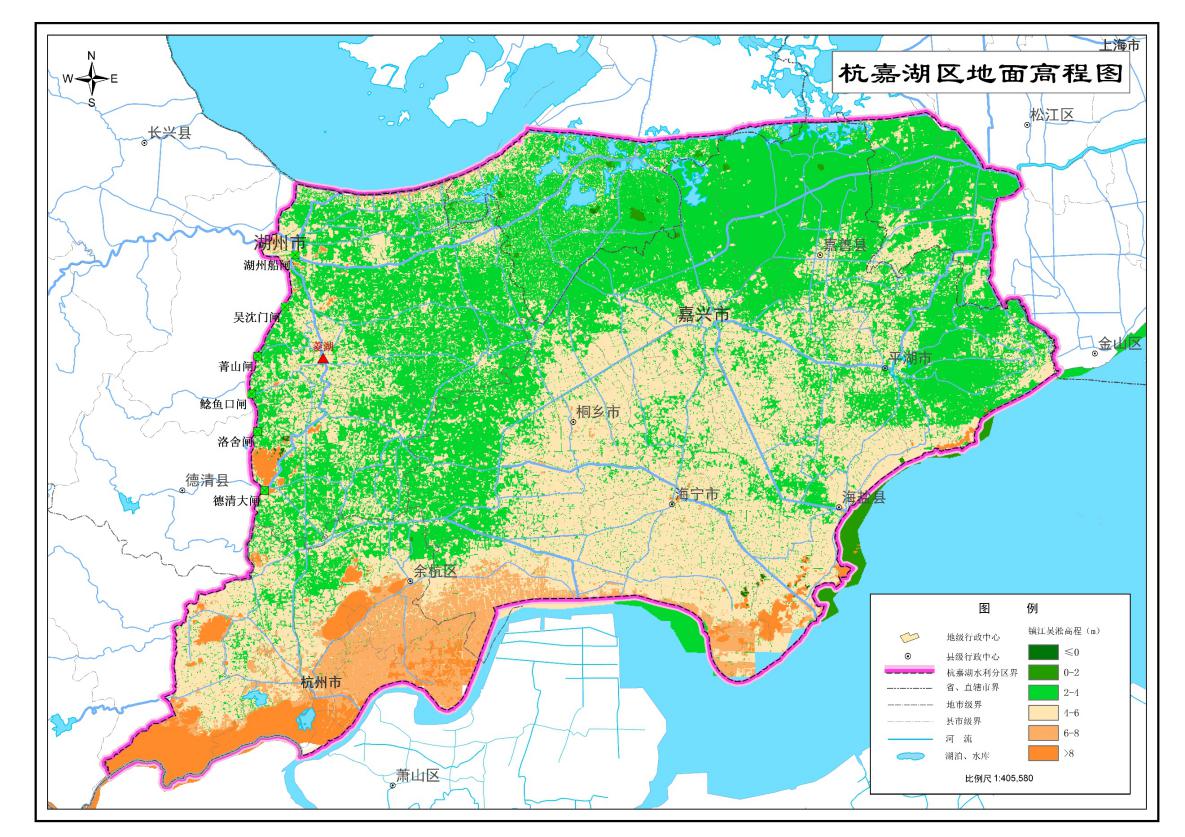


图5 杭嘉湖区地面高程图

### 3.1.2区域水系

杭嘉湖区西部是苕溪水系，对本区域产生很大威胁。北部是太湖和太浦河，东部是黄浦江水系。杭嘉湖区骨干河湖主要包括京杭运河、红旗塘、大泖港、长山河、盐官下河、海盐塘和上塘河等。



图6杭嘉湖地区水系图

杭嘉湖区洪水以排入黄浦江为主。主要有三个排水路径：一是承纳杭嘉湖区西部洪水，由頔塘、双林塘等河道汇入太浦河，再排入黄浦江；二是西北洪水经红旗塘等河道汇入园泄泾，再排入黄浦江；三是西部洪水经杭平申线、广陈塘等汇入大泖港，再排入黄浦江。

杭嘉湖区洪水也可往南排入杭州湾，主要由南台头、长山河、盐官下河等，南排杭州湾的骨干河道外排，对降低杭嘉湖区洪水作用明显。

上塘河水系，与运河水系、南排水系相对独立，基本不产生水量交换。

### 3.1.3水文气象

位于亚热带季风气候区的杭嘉湖区雨量充沛，多年平均降水量保持在1200m以上，温度适宜，多年平均气温保持在16℃左右。杭嘉湖区降雨量年内分配不均匀，梅汛期（4~7月）占年雨量的45%左右，台汛期（8~10月），一般发生短历时大暴雨。杭嘉湖区降雨量在不同年份间存在明显变化，丰水年（如1954年）最大年降水总量达到1905mm，枯水年份（如1978年）最小年降水总量全年为850mm。

## 3.2降雨场次筛选

杭嘉湖区域主要参考的代表水位站点是嘉兴站，当洪水位超过警戒水位3.3m，定义为超警戒水位洪水，简称超警洪水。本次筛选2010年~2019年杭嘉湖区嘉兴站水位超过警戒水位3.3m的降雨场次共11场降雨作为训练样本，进行人工神经网络参数训练。

## 3.3基于BP神经网络的杭嘉湖洪水位预报模型构建

### 3.3.1网络拓扑结构确定

考虑到降雨是影响水位的最关键的自然因素、南排工程排水量是影响水位的最关键的人为因素，BP神经网络输入层共4个输入要素，包括时间、杭嘉湖面降雨、嘉兴雨量站雨量及南排水量，输出层为1个输出要素，即嘉兴水位。其中隐含层个数根据需通过试算确定。网络拓扑结构如下图所示：

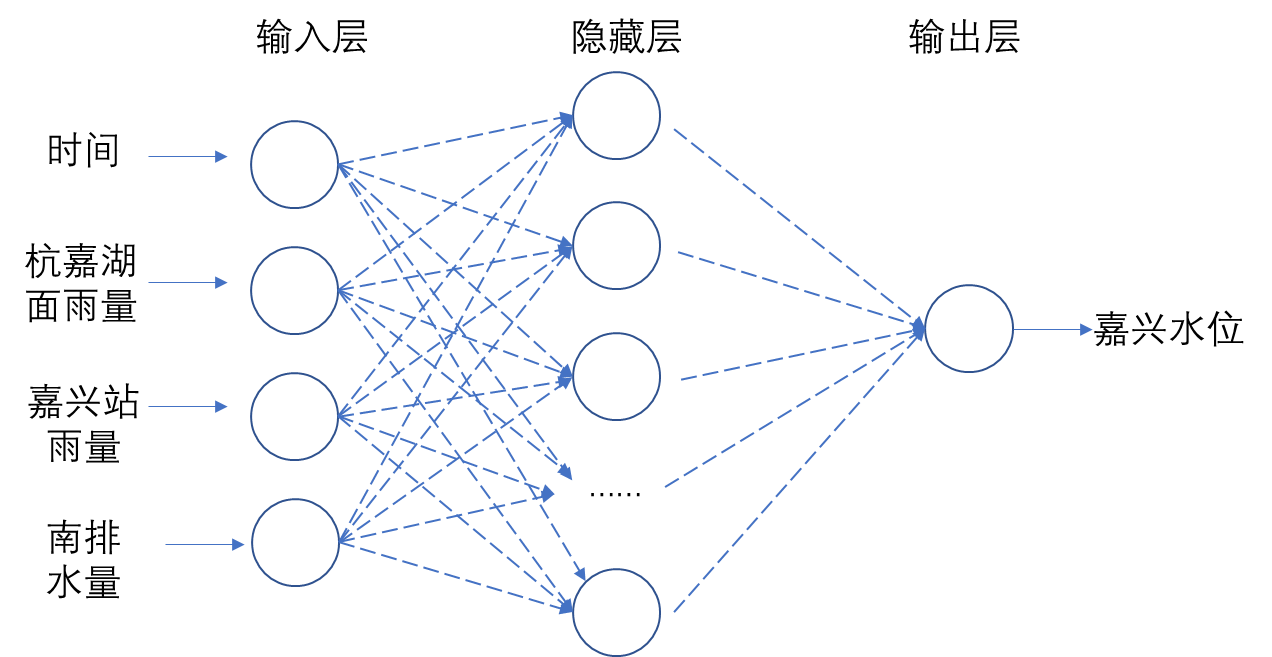


图7 杭嘉湖洪水位预报人工神经网络模型网络结构

对模型收敛速度、计算成果精确程度影响较大的是隐藏层中的神经元节点数。假如隐藏层中的神经元节点数太少，那么易导致网络结构太简单，难以真实审客观地反映机理演化过程。假如隐藏层中的神经元节点数太多，可产生模型训练效果较好的结果。可是，也会导致模型过拟合的情况，对未来预测效果产生影响。因此，也要避免出现过拟合的情况。

为确定神经网络隐含层合理节点数量，随机选取2016.6.9~2016.7.8场次降雨，将该时段的日期、杭嘉湖区面降雨、嘉兴站雨量、南排水量等4项作为输入要素进行输入，在最大迭代数max\_epochs=6000次、学习率learn\_rate=0.025、误差限mse\_final=6.5e-4，相同参数条件下，设定不同隐含层神经元节点数进行分析，模拟成果见图8。

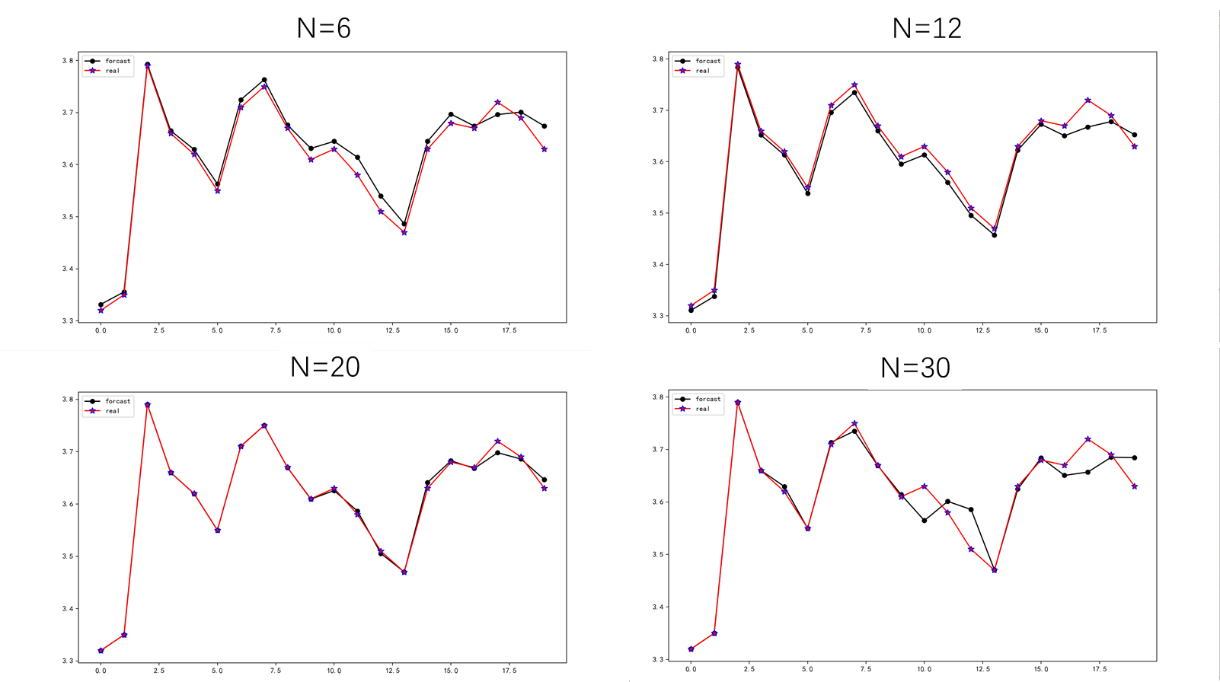


图8不同隐含层节点数模拟成果

分别采用不同隐含层节点数量构建人工神经网络拓扑关系，对2016年6-7月降雨场次嘉兴水位模拟成果显示，隐含层节点越多，水位模拟效果越好，当N=20时，水位过程与实测拟合精度最高，当隐含层节点数量增加为N=30时，出现模型过拟合情况，在部分时间，比如，极值水位模拟值和实测值的重合度更好一些。但从全局最优角度考虑，过拟合后过程中水位出现误差较大的现象难以消除，水位全过程模拟精度有所降低。

通过不断试算比选，最终确定本次研究BP人工神经网络的隐含层由20个神经元节点构成。

### 3.3.2基于Python的人工神经网络编程

Python语言具有较好的解释性、编译性、互动性和可读性。这项研究是基于Python语言环境进行的。利用pycharm搭建平台，通过引入numpy、pandas、matplotlib和sklearn等外部扩展程序库实现数据计算和处理。

编程界面如图9所示。Python具体程序见附录。



图9 Python编程界面示意图

在确定网络拓扑结构后，BP人工神经网络采用训练样本对网络结构中的权重系数进行训练，如图10所示。首先通过网络初始化，为网络中权重系数赋随机值；其次，输入参数归一化；其三，通过正向传播对网络中各权重系数赋值，其中各神经元阈值函数采用通用sigmoid函数进行处理；第四，通过对比输出值与实测水位结果误差，反向对网络权重系数进行调整，并循环迭代，直至实测与模拟输出值误差低于误差限mse\_final后，基于BP人工神经网络的杭嘉湖洪水位预测模型构建成功。

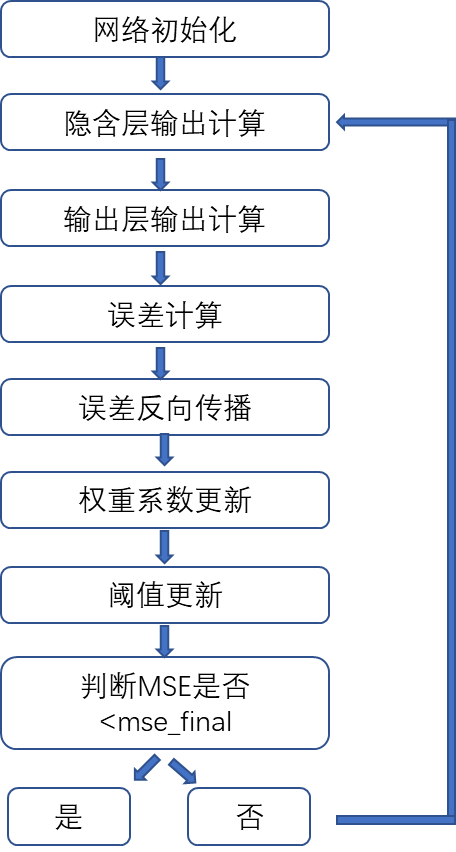


图10 BP人工神经网络算法流程图

# 4 预测成果对比分析

## 4.1对于11场次洪水位预测成果分析

基于上述确定的2010年~2019年中11场典型降雨过程，嘉兴水位达到峰值的前1d作为预报依据时间，预报依据时间前的降雨过程、嘉兴水位、南排水量均采用实测数据，并对预报依据时间后未来1d的水位峰值进行预测，并通过预测峰值和实际洪峰水位值进行比较，分析本次BP神经网络洪水位预报模型的准确性。

表1列出了11场次降雨预测洪峰与实际洪峰数据及模拟精度误差，其中，11场次降雨中，2016.5.17~2016.6.4模型预测未来1d水位涨至3.34m，实际水位则达到3.65m，预报较实际偏低31cm，此外10场次降雨预测洪水位峰值与实测峰值误差均在5cm以内，满足目前地区水位预报精度±10cm要求。BP人工神经网络洪水位预报准确率达到91%。

表1 2010年~2019年场次降雨洪水位未来1d峰值预测分析表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **场次** | **Log10（MSE\_history）** | **实际峰值**  **(m)** | **预测峰值**  **(m)** | **预测-实际**  **(m)** |
| 1 | 2011.6.10~2011.6.21 | -3.187 | 3.89 | 3.87 | -0.02 |
| 2 | 2013.10.3~2013.10.9 | -3.189 | 4.29 | 4.32 | 0.03 |
| 3 | 2014.7.21~2014.8.21 | -1.957 | 3.69 | 3.69 | 0 |
| 4 | 2015.6.12~2015.7.14 | -2.11 | 3.9 | 3.91 | 0.01 |
| 5 | 2015.11.10~2015.11.27 | -3.187 | 3.36 | 3.31 | -0.05 |
| 6 | 2016.5.17~2016.6.4 | -2.7 | 3.65 | 3.34 | ***-0.31*** |
| 7 | 2016.6.17~2016.7.8 | -3.18 | 3.72 | 3.73 | 0.01 |
| 8 | 2018.6.18~2018.7.9 | -3.182 | 3.43 | 3.47 | 0.04 |
| 9 | 2019.2.3~2019.2.24 | -3.18 | 3.45 | 3.44 | -0.01 |
| 10 | 2019.6.15~2019.6.24 | -3.1 | 3.52 | 3.52 | 0 |
| 11 | 2019.6.27~2019.7.4 | -3.187 | 3.47 | 3.47 | 0 |

## 4.2 2020年汛期嘉兴洪水位预报分析

为进一步分析本次开发的人工智能模型在杭嘉湖地区洪水位预报方面的计算精度及模拟速度，对2020年汛期6.21~7.19场次降雨的区域洪水位进行模拟计算，验证BP神经网络成果的合理性。

从模型计算效率上来看，若采用BP人工神经网络计算，累计耗时2.5s。

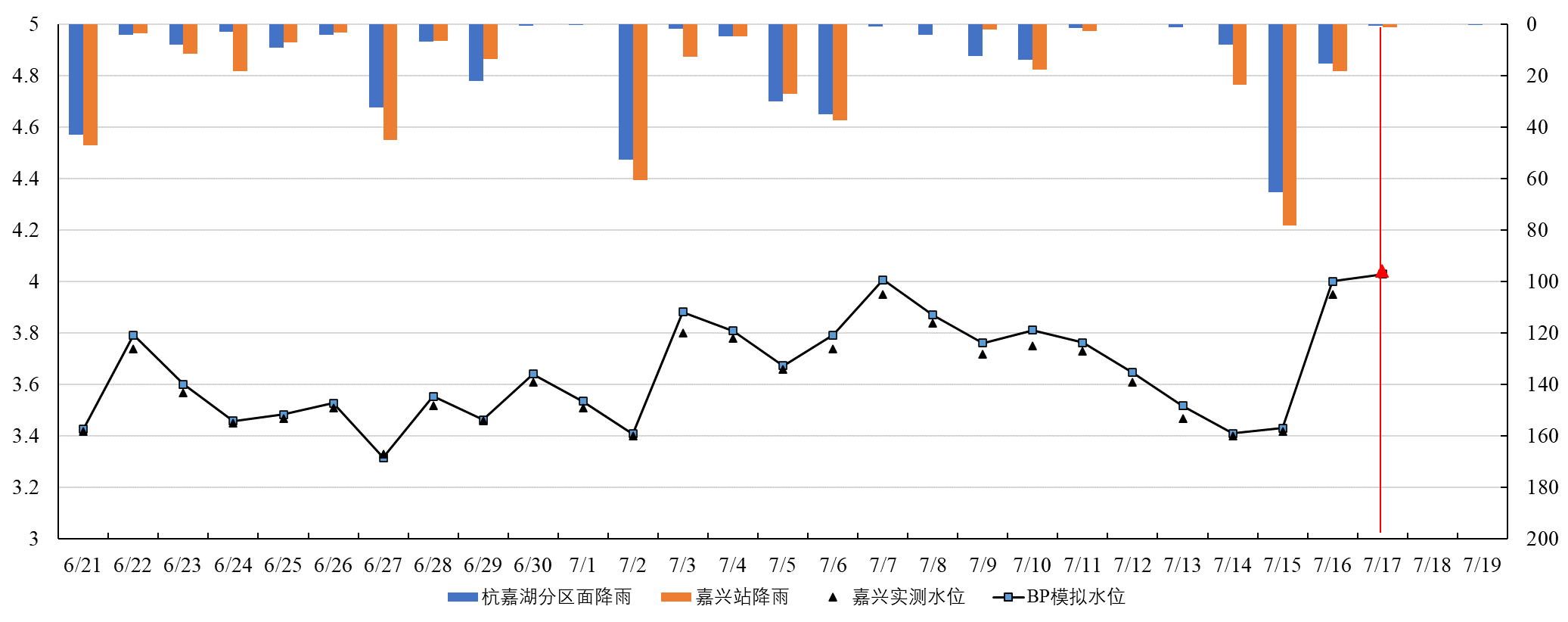


图11 2020验证时段人工智能模型与2020年实测水位对比图

从模型模拟精度上来看，6.21~7.19场次降雨过程如图4.2-2所示，7月15日区域面平均雨量和嘉兴站雨量分别为65.1mm和78mm，均达到场次日降雨最大值，嘉兴站实测水位在7月17日达到峰值4.04m。从图中可知，BP神经网络7月17日洪水位计算峰值为4.03m，较实测仅偏低1cm。可见，基于BP人工神经网络的杭嘉湖区洪水位预测模型成果合理、计算速度快。

# 5 结论与展望

## 5.1小结

本研究采用人工智能方法，运用Python语言构建基于BP人工神经网络的杭嘉湖洪水位快速预报模型，并对模型训练及预报成果进行分析。本研究主要得出以下结论：

（1）分别采用不同隐含层节点数量构建人工神经网络拓扑关系，对典型场次降雨嘉兴水位模拟成果分析可知，隐含层节点越多，水位模拟效果越好，当N=20时，水位过程与实测拟合精度最高，当隐含层节点数量增加为N=30时，出现模型过拟合情况，通过比选，最终确定本次研究所采用的BP人工神经网络拓扑结构中隐含层由20个神经元节点组成。

（2）基于选定的2010年~2019年中11场已发生的典型降雨过程，预报洪水位峰值数据并分析模拟精度误差，其中10场次降雨预测洪水位峰值与实测峰值误差均在5cm以内，满足目前地区水位预报精度±10cm要求，BP人工神经网络洪水位预报准确率达到91%。

（3）采用2020年验证时段6月21日~7月19日场次降雨过程模拟，BP人工神经网络模型计算效率及预报精度都较好。

## 5.2展望

本研究采用人工智能技术，尝试水文预报新路径，提升域防洪快速决策支持能力。采用了BP人工神经网络算法构建了杭嘉湖平原洪水位预报模型，对未来一天洪水位峰值进行了预报研究，研究成果可知，采用该方法可以较为快速的完成未来一天的代表站点洪水位预测预报，模拟精度尚可。洪水应对还需要三天和七天甚至时间更长的预见能力，中长期的预报具有重要意义。但是，中长期预报受影响因素较多，包括气候、人类活动等因素，且有些难以监测，没有监测数据。因此，需要进一步完善人工智能算法网络架构，提升人工智能的防洪决策支撑能力。

# 6 收获与体会

这次是我首次进行课题研究。这一让人印象深刻的活动，我在此学习到了洪水预报的艰辛、Python语言的编写、BP人工神经网络、论文报告的编排。但这里面还有很多值得我去继续学习下去、钻研下去的东西。我希望能有更多的机会来研究自己的小课题，同时锻炼自己的能力。

# 7致谢

感谢太湖流域管理局水利发展研究中心提供的宝贵资料。

感谢辅导老师李老师和各个帮助过我的同学们！

# 参考文献

[1] 祁乐,谢邦昌.准AI时代:人工智能的应用与挑战[J].2018，第9期，23-25.

[2] 基于人工智能和大数据驱动的新一代水文模型及其在洪水预报预警中的应用[J].2019.5,第5期第29卷,11,22.

[3] 人工智能在洪水预报中的应用[J].2005.9,第31卷第9期,12-15.

[4] 太湖洪水预报的前向人工神经网络全局最优逼近方法[J].2001.7,第29卷第4期,84-90.

# 附录

**Python程序代码：**

**import** numpy **as** np *#引入外部扩展程序库NumPy并简写为“np”***import** pandas **as** pd *#引入外部扩展程序库pandas并简写为“pd”***import** matplotlib **as** mpl *#引入外部扩展程序库matplotlib并简写为“mpl”，该库属于画图的指令库***import** matplotlib.pyplot **as** plt *#引入外部扩展程序库matplotlib中的pyplot命令，并简写为“plt”，该库属于画图的指令库***from** sklearn.preprocessing **import** MinMaxScaler *#从外部扩展程序库sklearn中的preprocessing命令中引入MinMaxScaler指令，用于归一化  
  
## 设置字符集，防止中文乱码*mpl.rcParams[**'font.sans-serif'**] = [**u'simHei'**] *##用来正常显示中文标签*mpl.rcParams[**'axes.unicode\_minus'**] = **False** *##用来正常显示正负号*df = pd.read\_csv(**'JXwater.csv'**, encoding=**'utf-8'**) *#从文件JXwater.csv中引入数据，数据格式为utf-8  
  
# 神经网络模型相关参数*max\_epochs = 60000 *#迭代次数*learn\_rate = 0.025 *#学习率*mse\_final = 6.5e-4 *#误差限阈值*input\_number = 4 *#模型设置4个输入*out\_number = 1 *#模型设置1个输出*hidden\_unit\_number =20 *#隐藏层个数设置为20  
  
# 神经网络结构矩阵  
# 20\*4 矩阵，输入层到隐藏层的系数矩阵*w1 = 0.5 \* np.random.rand(hidden\_unit\_number, input\_number) - 0.1 *#返回hidden\*input维度随机数组  
# 20\*1 矩阵，隐藏层到输出层的系数矩阵*b1 = 0.5 \* np.random.rand(hidden\_unit\_number, 1) - 0.1 *#返回hidden\*output维度随机数组  
  
# 1\*20 矩阵*w2 = 0.5 \* np.random.rand(out\_number, hidden\_unit\_number) - 0.1  
*# 1\*1 矩阵*b2 = 0.5 \* np.random.rand(out\_number, 1) - 0.1  
*# 读取特征属性X  
# X为输入层，输入要素为日期，杭嘉湖分区平均降雨，嘉兴站降雨降雨和南排水量*x = df[[**'日期'**,**'杭嘉湖分区平均'**, **'嘉兴站降雨'**,**'南排水量'**]]  
*# Y为输出层，输处为嘉兴站水位*y = df[[**'嘉兴水位'**]]  
time=df[[**'日期'**]] *#变量time来代表日期  
# print(x) #输出X  
# print(y) #输出Y  
# print(time) #输出日期*sample\_number = x.shape[0] *#sample\_number为样本数量*print(**"sample\_number= "**,sample\_number) *#输出样本数量  
# 因为x和y的数据取值范围太大了，所以做一个归一化操作（使用区间缩放法）*x\_scaler = MinMaxScaler(feature\_range=(-1, 1))  
y\_scaler = MinMaxScaler(feature\_range=(-1, 1))  
x = x\_scaler.fit\_transform(x)  
y = y\_scaler.fit\_transform(y)  
  
*# 为了后面和w进行矩阵的乘法操作，对相应矩阵进行转置*sample\_in = x.T  
sample\_out = y.T  
*#输出结果用于检查*print(sample\_in)  
print(sample\_out)  
*#定义sigmoid阈值函数***def** sigmoid(z):  
 **return** 1.0 / (1.0 + np.exp(-z))  
  
mse\_history = [] *#新建空数组，后面用于存放真实值与实际值误差，并通过迭代来逐步缩小误差，直到满足<mse\_final迭代停止  
# BP神经网络的计算主程序***for** i **in** range(max\_epochs): *# max\_epochs迭代次数  
 # FP过程，误差前向传播过程  
 # 隐藏层的输出* hidden\_out = sigmoid(np.dot(w1, sample\_in).transpose() + b1.transpose()).transpose()  
  
 *# 输出层的输出（为了简化我们的写法，输出层不进行sigmoid激活）* network\_out = (np.dot(w2, hidden\_out).transpose() + b2.transpose()).transpose()  
  
 *# 误差值* err = sample\_out - network\_out *#误差值err=实际输出值-网络训练输出值* mse = np.average(np.square(err)) *#求均方误差* mse\_history.append(mse) *#误差累积值存放在mse\_history中* min\_mse=min(mse\_history[i-1],mse\_history[i]) *#min\_mse为累积误差的最小值  
 #如果mse小于误差限阈值则迭代终止，跳出循环* **if** mse < mse\_final:  
 **break** *# BP 过程，误差反向传播过程* delta2 = -err *#输出层到隐藏层误差* delta1 = np.dot(w2.transpose(), delta2) \* hidden\_out \* (1 - hidden\_out) *#隐藏层到输入层误差* dw2 = np.dot(delta2, hidden\_out.transpose())  
 db2 = np.dot(delta2, np.ones((sample\_number, 1)))  
 dw1 = np.dot(delta1, sample\_in.transpose())  
 db1 = np.dot(delta1, np.ones((sample\_number, 1)))  
 w2 -= learn\_rate \* dw2 *#误差调整后的w2* b2 -= learn\_rate \* db2  
 w1 -= learn\_rate \* dw1 *#误差调整后的w1* b1 -= learn\_rate \* db1  
*#=====================成果图输出（包括1个误差曲线图、1个BP网络计算成果和实际输出结果的水位过程对比图，  
#以及1个未来1d BP网络预测成果和实际水位过程对比图====================================  
#fig.1#误差曲线图*mse\_history10 = np.log10(mse\_history) *#取误差对数值，为了看到误差收敛的变化情况*min\_mse = min(mse\_history10)  
print(**"min\_mse"**,min\_mse) *#输出min\_mse*plt.plot(mse\_history10) *#画误差变化过程图*print(**"min\_mse="**,min\_mse)  
plt.plot([0, len(mse\_history10)], [min\_mse, min\_mse])  
ax = plt.gca()  
ax.set\_yticks([-2, -1, 0, 1, 2, min\_mse])  
ax.set\_xlabel(**'iteration'**)  
ax.set\_ylabel(**'MSE'**)  
ax.set\_title(**'Log10 MSE History'**)  
plt.show()  
  
df1 = pd.read\_csv(**'JXwater\_test.csv'**, encoding=**'utf-8'**) *# 读取预测集  
#预测集，利用前面训练好的BP神经网络，将预测数据所谓输入给到BP，获得输出的水位预测结果*x1 = df1[[**'日期'**,**'杭嘉湖分区平均'**, **'嘉兴站降雨'**,**'南排水量'**]] *#预测集输入*y1 = df1[[**'嘉兴水位'**]] *#真实值*x1\_scaler = MinMaxScaler(feature\_range=(-1, 1))  
y1\_scaler = MinMaxScaler(feature\_range=(-1, 1))  
x1 = x1\_scaler.fit\_transform(x1)  
y1 = y1\_scaler.fit\_transform(y1)  
test\_in=x1.T  
*## 仿真输出和实际输出对比图*hidden\_out = sigmoid((np.dot(w1, sample\_in).transpose() + b1.transpose())).transpose()  
network\_out = (np.dot(w2, hidden\_out).transpose() + b2.transpose()).transpose()  
sample\_out = y\_scaler.inverse\_transform(y)  
*# # 隐藏层输出，w1,w2已经训练好，可以直接调用  
# # hidden\_out = sigmoid((np.dot(w1, test\_in).transpose() + b1.transpose())).transpose()  
# # 输出层输出  
# # network\_out = (np.dot(w2, hidden\_out).transpose() + b2.transpose()).transpose()  
# 反转获取实际值*network\_out = y\_scaler.inverse\_transform(network\_out.T)  
*# print('network\_out',network\_out[:,0])  
  
#fig.2#嘉兴水位BP模拟与实测过程对比图*fig, axes = plt.subplots(nrows=1, ncols=1, figsize=(12, 10)) *#设置图片的位置及大小（figsize）*line1, = axes.plot(network\_out[:, 0], **'k'**, marker=**'o'**) *#图中第一条线，训练值，用黑色圆点表示*line2, = axes.plot(sample\_out[:, 0], **'r'**, markeredgecolor=**'b'**, marker=**'\*'**, markersize=9) *#图中第二条线，实测值，用五角星表示*axes.legend((line1, line2), (**'forcast'**, **'real'**), loc=**'upper left'**)  
axes.set\_title(**'嘉兴水位模拟与实测过程'**)  
  
*#fig.3# BP预测输出和实际输出对比图*test\_hidden\_out = sigmoid((np.dot(w1, test\_in).transpose() + b1.transpose())).transpose()  
test\_network\_out = (np.dot(w2, test\_hidden\_out).transpose() + b2.transpose()).transpose()  
test\_out = y1\_scaler.inverse\_transform(y1)  
*# 反转获取实际值*test\_network\_out = y1\_scaler.inverse\_transform(test\_network\_out.T)  
test\_out = y1\_scaler.inverse\_transform(y1)  
print(**"test\_network\_out"**,test\_network\_out)  
*#调用作图指令↓*fig, axes = plt.subplots(nrows=1, ncols=1, figsize=(12, 10))*#设置图片的位置及大小（figsize）*line1, = axes.plot(test\_network\_out[:, 0], **'k'**, marker=**'o'**)*#图中第一条线，预测值，用黑色圆点表示*line2, = axes.plot(test\_out[:, 0], **'r'**, markeredgecolor=**'b'**, marker=**'\*'**, markersize=9)*#图中第二条线，实测值，用五角星表示*axes.legend((line1, line2), (**'forcast'**, **'real'**), loc=**'upper left'**)  
axes.set\_title(**'嘉兴水位预测集与实测'**)  
  
plt.show()*#在计算机界面显示上述3幅图*