

上海电力大学

神经网络与机器学习



题 目：_____短时交通流量预测_____

学 院：_____数理学院_____

专业年级：_____信息与计算科学专业 2022 级_____

学生姓名：_____刘伟涛_____学号：_____20222423_____

2024 年 12 月

一、案例背景

1. 小波理论

小波分析是针对傅里叶变换的不足发展而来的。傅里叶变换是信号处理领域中应用最广泛的一种分析手段，然而它有一个严重不足，就是变换时抛弃了时间信息，通过变换结果无法判断某个信号发生的时间，即傅里叶变换在时域中没有分辨能力、小波是一种长度有限、平均值为 0 的波形，它的特点包括：

①时域都具有紧支集或近似紧支集

②直流分量为 0

小波函数是由一个母小波函数经过平移与尺寸伸缩得到，小波分析即把信号分解成一系列小波函数的叠加。

小波变换是指把某一基本小波函数(t) 平移：后，再在不同尺度 a 下与待分析的信号 x(t) 做内积。

$$f_x(a, \tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t) \psi\left(\frac{t-\tau}{a}\right) dt \quad a > 0$$

等效的时域表达式为

$$f_x(a, \tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(\omega) \psi(a\omega) e^{i\omega\tau} d\omega \quad a > 0$$

式中，r 和 a 是里面的参数，r 相当于使镜头相对于目标平行移动，相当于使镜头相目标推进或远离。

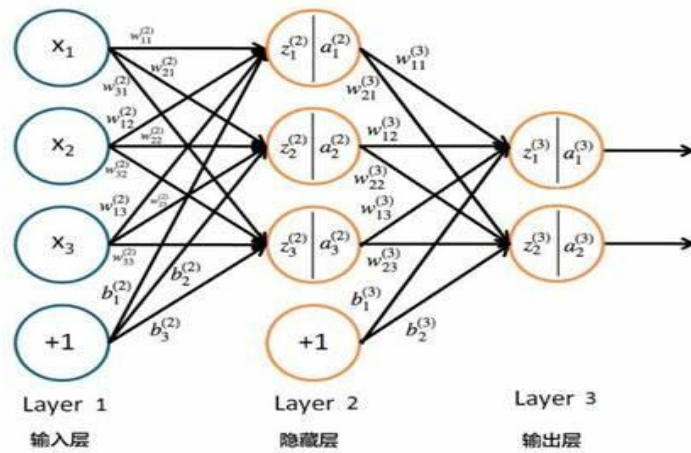
小波分析能够通过小波基函数的变换分析信号的局部特征，并且在二维情况先具有信号方向选择性能能力，因此，该方法作为一种数学理论和分析方法，引发了广泛关注。

2. 小波神经网络

小波神经网络是一种以 BP 神经网络拓扑结构为基础，把小波基函数作为隐含层节点的传递函数，信号前向传播的同时误差反向传播的神经网络。小波神经网络的拓扑结构如图所示。

其中， X_1, X_2, \dots 是小波神经网络的输入参数， Y_1, Y_2, \dots 是小波神经网络的预测输出，‘ ’ 为小波神经网络权值。

在输入信号序列为 $x_i(i=1,2,3,4\dots k)$ 时，隐含层输出公式为

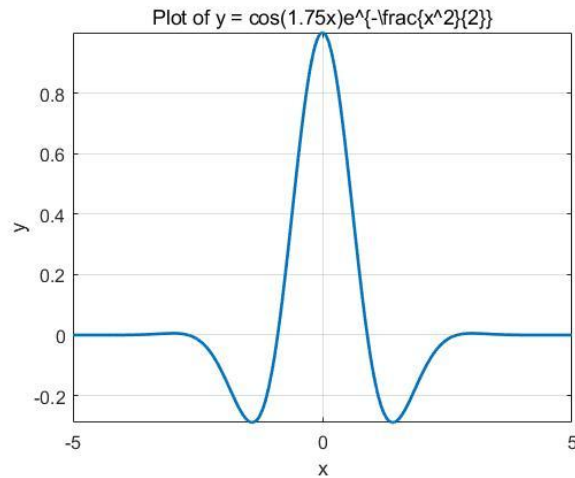


$$h(j) = h_j \left[\frac{\sum_{i=1}^k \omega_i x_i - b_j}{a_j} \right] \quad j = 1, 2, \dots, l$$

式中， $h(j)$ 为隐含层第 j 个节点输出值； w_{ij} 为输入层和隐含层的连接权值； b_j 为小波基函数 h_j 的平移因子； a_j 为小波函数 h_j 的伸缩因子； h_j 为小波基函数。本案例采用的小波基函数为 Morlet 母小波基函数，数学公式为

$$y = \cos(1.75x)e^{-x^2/2}$$

函数图形为



小波神经网络输出层计算公式为

$$y(k) = \sum_{i=1}^l \omega_k h(i) \quad k = 1, 2, \dots, m$$

式中， ω_k 为隐含层到输出层权值； $h(i)$ 为第 i 个隐含层节点的输出； l 为隐含层节点数； m 为输出层节点数。

小波神经网络权值参数修正算法类似于 BP 神经网络权值修正算法，采用梯度修正法修正网络的权值和小波基函数参数，从而使小波神经网络预测输出不断逼近期望输出。小波神经网络修正过程如下。

(1) 计算网络预测误差

$$e = \sum_{k=1}^n y_n(k) - y(k)$$

式中， $y_n(k)$ 为期望输出； $y(k)$ 为小波神经网络预测输出。

(2) 根据预测误差 e 修正小波神经网络权值和小波基函数系数

$$\omega_{n,k}^{(i+1)} = \omega_{n,k}^i + \Delta\omega_{n,k}^{(i+1)}$$

$$a_k^{(i+1)} = a_k^i + \Delta a_k^{(i+1)}$$

$$b_k^{(i+1)} = b_k^i + \Delta b_k^{(i+1)}$$

式中，是根据网络预测误差计算得到：

$$\Delta\omega_{n,k}^{(i+1)} = -\eta \frac{\partial e}{\partial \omega_{n,k}^{(i)}}$$

$$\Delta a_k^{(i+1)} = -\eta \frac{\partial e}{\partial a_k^{(i)}}$$

$$\Delta b_k^{(i+1)} = -\eta \frac{\partial e}{\partial b_k^{(i)}}$$

式中， η 为学习速率。

小波神经网络算法训练步骤如下。

步骤 1：网络初始化。随机初始化小波函数伸缩因子 a_k 、平移因子 b_k 以及网络连接权重 $\omega_{ij}\omega_{jk}$ ，设置网络学习速率 η 。

步骤 2：样本分类。把样本分为训练样本和测试样本，训练样本用于训练网络，测试样本用于测试网络预测精度。

步骤 3：预测输出。把训练样本输入网络，计算网络预测输出并计算网络输出和期望输出的误差 e 。

步骤 4：权值修正。根据误差 e 修正网络权值和小波函数参数，使网络预测值逼近期望值。

步骤 5：判断算法是否结束，若没有结束，返回步骤 3。

3. 交通流量预测

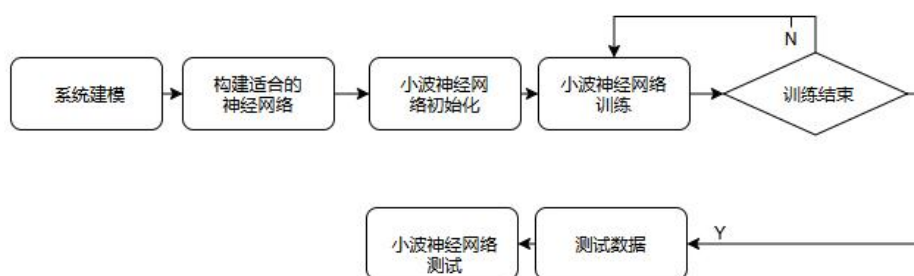
随着交通基础设施建设和智能运输系统的发展，交通规划和交通诱导已成为交通领域研究的热点。对于交通规划和交通诱导来说，准确的交通流量预测是其实现的前提和关键。交通流量预测根据时间跨度可以分为长期交通流量预测和短期交通流量预测：长期交通流量预测以小时、天、月甚至年为时间单位，是宏观意义上的预测；短时交通流量预测一般的时间跨度不超过 15 分钟，是微观意义上的预测。短时交通流量预测是智能运输系统的核心内容，智能运输系统中多个

子系统的功能实现都以其为基础。短时交通流量预测具有高度非线性和不确定性等特点，并且同时间相关性较强，可以看成是时间序列预测问题，比较常用的方法包括多元线性回归预测、AR 模型预测、ARMA 模型预测、指数平滑预测等。

二、模型建立

研究表明，城市交通路网中交通路段上某时刻的交通流量与本路段前几个时段的交通流量有关，并且交通流量具有 24 小时内准周期的特性。根据交通流量的特性设计小波神经网络，该网络分为输入层、隐含层和输出层三层。其中，输入层输入为当前时间点的前 n 个时间点的交通流量；隐含层节点由小波函数构成；输出层输出当前时间点的预测交通流量。

首先采集 4 天的交通流量数据，每隔 15 分钟记录一次该段时间内的交通流量，一共记录 384 个时间点的数据，用 3 天共 288 个交通流量的数据训练小波神经网络，最后用训练好的小波神经网络预测第 4 天的交通流量。基于小波神经网络的短时交通流量预测算法流程如图所示。



小波神经网络的构建确定小波神经网络结构。本案例采用的小波神经网络结构为 4—6—1：输入层有 4 个节点，表示预测时间节点前 4 个时间点的交通流量；隐含层有 6 个节点；输出层有 1 个节点，为网络预测的交通流量。网络权值和小波基函数在参数初始化时随机得到。

小波神经网络训练：用训练数据训练小波神经网络，网络反复训练 100 次

小波神经网络测试：用训练好的小波神经网络预测短时交通流量，并对预测结果进行分析。

三、编程实现

根据小波神经网络原理在 MATLAB 环境中编程实现基于小波神经网络的短时交通流量预测。

1. 小波神经网络初始化

从数据库中下载训练数据和预测数据，初始化小波神经网络结构、权值和小波函数参数，并对训练数据进行归一化处理。其中，input、output 分别为训练输入和输出数据，input_test、output_test 分别为预测输入和输出数据。

```
% 从.mat 文件中加载数据（假设 traffic_flux.mat 包含了 input, output,  
input_test, output_test 变量）  
load('traffic_flux.mat', 'input', 'output', 'input_test',  
'output_test');
```

```
% 网络结构初始化
```

```

H = 4; % 隐藏层神经元数量
N = 1; % 输出层神经元数量
n = 6; % 输入层神经元数量（这里假设输入数据的特征数为 6，需要根据实际情况调整）
M = H; % 假设输入层到隐藏层的权重矩阵维度与隐藏层神经元数量相匹配（这里可以是一个不同的值，但通常隐藏层神经元数量与权重矩阵的第二维相匹配）

% 权值和参数学习率
lr1 = 0.01; % 权重学习率
lr2 = 0.001; % 参数（a, b）学习率
maxgen = 100; % 网络迭代学习次数

% 网络权值初始化
Wjk = randn(n, M); % 输入层到隐藏层的权重
Wij = randn(N, H); % 隐藏层到输出层的权重（注意这里的 H 应该是隐藏层神经元数量）
a = randn(1, H); % 隐藏层神经元的激活函数参数（例如，在 Morlet 小波中可能是尺度参数）
b = randn(1, H); % 隐藏层神经元的激活函数参数（例如，可能是偏移参数）

% 权值学习增量初始化
d_Wjk = zeros(n, M);
d_Wij = zeros(N, H);
da = zeros(1, H);
db = zeros(1, H);

% 训练数据归一化
[inputn, inputps] = mapminmax(input'); % 注意：' 表示转置，但通常 mapminmax 的输入应该是列向量或矩阵的列
[outputn, outputps] = mapminmax(output'); % 同上
inputn = inputn'; % 转置回原始形状
outputn = outputn'; % 同上

```

2. 小波神经网络训练

用训练数据训练小波神经网络，使小波神经网络具有短时交通流量预测能力。

网络训练

% 假设 maxgen, input, output, n, M, N, lr1, lr2, a, b, Wij, Wjk 已经定义并初始化

```

for i = 1:maxgen
    error(i) = 0; % 记录每次误差

    % 网络训练
    for kk = 1:size(input, 1)
        % 提取输入输出数据

```

```

x = input(kk, :);
yqw = output(kk, :);

% 初始化网络预测输出
y = zeros(1, N);

% 网络预测输出
for j = 1:n
    net(j) = 0;
    for k = 1:M
        net(j) = net(j) + Wjk(j, k) * x(k);
    end
    net_ab(j) = (net(j) - b(j)) / a(j);
    temp = nynorlet(net_ab(j));
    for k = 1:N
        y(k) = y(k) + Wij(k, j) * temp;
    end
end

% 误差累积
error(i) = error(i) + sum((yqw - y).^2);
end
error(i) = sqrt(error(i) / size(input, 1)); % 计算均方误差

% 权值修正
for j = 1:n
    % 计算 d_Wij (Wij 修正值)
    temp = nynorlet(net_ab(j));
    for k = 1:N
        dWij(k, j) = -(yqw(k) - y(k)) * temp;
    end

    % 计算 d_Wjk (Wjk 修正值)
    temp = dmymorlet(net_ab(j));
    for k = 1:M
        d_wjk_sum = 0;
        for l = 1:N
            d_wjk_sum = d_wjk_sum + (yqw(l) - y(l)) * Wij(l, j);
        end
        d_Wjk(j, k) = -d_wjk_sum * temp * x(k) / a(j);
    end

    % 计算 db (b 修正值)
    d_b(j) = 0;
end

```

```

        for k = 1:N
            d_b(j) = d_b(j) + (yqw(k) - y(k)) * Wij(k, j);
        end
        db(j) = d_b(j) * temp / a(j);

        % 计算 d_a (a 修正值)
        d_a(j) = 0;
        for k = 1:N
            d_a(j) = d_a(j) + (yqw(k) - y(k)) * Wij(k, j);
        end
        da(j) = d_a(j) * temp * ((net(j) - b(j)) / b(j)) / a(j)^2;
    end

    % 权值参数更新
    Wij = Wij - lr1 * dWij;
    Wjk = Wjk - lr1 * d_Wjk;
    a = a - lr2 * da;
    b = b - lr2 * db;

    % 重置临时变量
    d_Wjk = zeros(n, M);
    dWij = zeros(N, n);
    da = zeros(1, n);
    db = zeros(1, n);
    y = zeros(1, N);
    net = zeros(1, n);
    net_ab = zeros(1, n);
end

% Morlet 小波函数
function y = nynorlet(t)
    y = exp(-(t.^2) / 2) .* cos(1.75 * t);
end

% Morlet 小波函数的偏导数
function y = dmymorlet(t)
    y = (-t .* exp(-(t.^2) / 2) .* cos(1.75 * t) - 1.75 * exp(-(t.^2) / 2) .* sin(1.75 * t));
end

```

3. 小波神经网络预测

用训练好的小波神经网络预测短时交通的形式表示小波神经网络预测结果。

% 假设 inputps 和 outputps 是之前通过 mapminmax 对训练数据归一化时得到的参数


```

% 预测输入归一化
x_test_norm = mapminmax('apply', input_test', inputps); % 注意：' 表示转置，使得每一列是一个样本
x_test_norm = x_test_norm'; % 转置回原始形状，使得每一行是一个样本

% 初始化变量
num_samples = size(x_test_norm, 1); % 样本数量
num_inputs = size(x_test_norm, 2); % 输入特征数量（应该与 n 相等）
yuce = zeros(num_samples, 1); % 用于存储预测结果的向量

% 网络权重
% Wjk: 输入层到隐藏层的权重矩阵
% Wij: 隐藏层到输出层的权重矩阵
% a, b: 隐藏层神经元的激活函数参数（

% 网络预测
for i = 1:num_samples
    x_test = x_test_norm(i, :); % 取出当前样本的输入特征
    net = zeros(1, H); % 隐含节点初始化（H 是隐藏层神经元数量）

    % 计算隐藏层输出
    for j = 1:H
        for k = 1:num_inputs
            net(j) = net(j) + Wjk(k, j) * x_test(k); % 计算输入层到隐藏层的加权和
        end
        net_ab(j) = (net(j) - b(j)) / a(j); % 应用激活函数前的变换
        temp = nynorlet(net_ab(j)); % 应用激活函数得到隐藏层输出
    end

    y = zeros(1, N); % 输出节点初始化（N 是输出层神经元数量）

    % 计算输出层输出
    for k = 1:N
        for j = 1:H
            y(k) = y(k) + Wij(k, j) * temp(j); % 计算隐藏层到输出层的加权和
        end
    end

    yuce(i) = y; % 存储预测结果
end

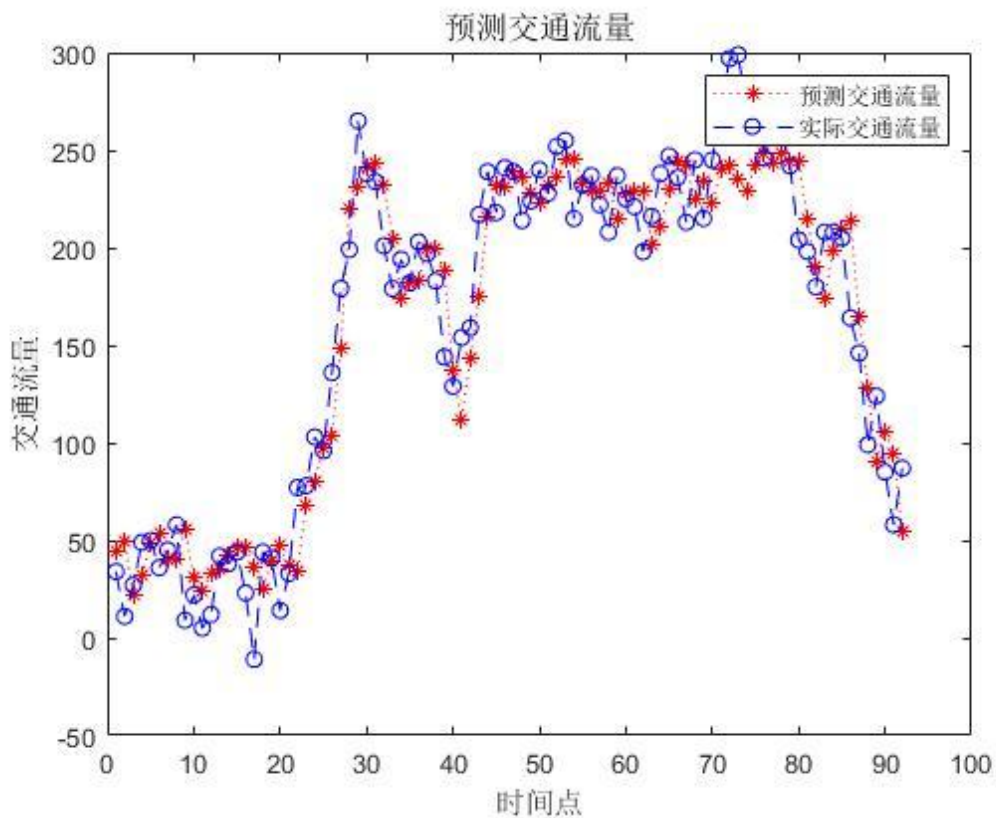
% 预测结果反归一化
yuce_denorm = mapminmax('reverse', yuce, outputps);

% 绘图

```

```
figure(1);
plot(yuce_denorm, 'r-'); % 绘制预测结果
hold on;
plot(output_test, 'b-'); % 绘制实际结果 title('预测交通流量', 'fontsize',
12);
legend('预测交通流量', '实际交通流量');
xlabel('时间点');
ylabel('交通流量');
```

4. 结果分析



上图是小波神经网络预测与实际情况的比较，由于 www.matlabsky.com 已经寄了，数据无从获得，上图数据是 CSDN 博主“配电网和 matlab”提供的的数据结果。

二、评语

成绩：_____

教师（签字/日期）：_____