

doi:10.3969/j.issn.1001-893x.2017.01.001

引用格式:朱江,宋永辉,刘亚利.基于改进型深度学习的流量预测[J].电讯技术,2017,57(1):1-8. [ZHU Jiang, SONG Yonghui, LIU Yali. Traffic prediction based on modified deep learning[J]. Telecommunication Engineering, 2017, 57(1): 1-8.]

## 基于改进型深度学习的流量预测\*

朱江,宋永辉\*\*,刘亚利

(重庆邮电大学 移动通信技术重庆市重点实验室,重庆 400065)

**摘要:**为了解决无线网络中流量的预测精度不高的问题,提出了一种自适应分组的栈式自编码(AG-SAEs)深度学习预测方法。在数据的预处理过程中,首先使用最大最小方式对数据进行归一化处理,并提出一种新型的自适应分组方法,把归一化后的链路数据进行关联性分组;然后,基于深度学习方法建立了一个多输入多输出的预测模型,并将分组后的数据输入到预测模型中,对该模型进行训练来建立输入和输出流量之间的映射关系;最后,为了进一步提高预测精度,在模型的训练过程中,使用改进型的牛顿法来进行权值参数更新。仿真实验以及和其他算法对比的结果证实了所提方案具有更小的预测相对误差。

**关键词:**认知网络;流量预测;深度学习;自适应分组

**中图分类号:**TN929.5   **文献标志码:**A   **文章编号:**1001-893X(2017)01-0001-08

## Traffic Prediction Based on Modified Deep Learning

ZHU Jiang, SONG Yonghui, LIU Yali

(Chongqing Key Laboratory of Mobile Communications Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

**Abstract:** To solve the problem of low traffic prediction accuracy in wireless networks, a method based on the adaptive grouping stacked auto-encoders (AG-SAEs) deep learning is proposed. In data preprocessing, the maximum and minimum method is used to normalize the data, and a novel adaptive grouping method is adopted to divide the normalized data into different groups adaptively. Then, a multi-input multi-output prediction model based on the deep learning model is established. All the groups are input to the stacked auto-encoder model to train the model and map the relationship between input and output traffic. Finally, in order to further improve the prediction accuracy, the modified Newton method is used to update the weight parameters in the model training section. The simulation experiment and comparison with other methods show that the proposed method processes a smaller prediction relative error.

**Key words:** cognitive network; traffic prediction; deep learning; adaptive grouping

### 1 引言

认知网络是将认知功能应用到网络的各个层中,以提升网络的整体性能。其中,流量问题是研究

的一个方面。无线网络流量预测的问题,是通过收集各个链路的历史和当前的数据信息,来对每一个链路未来的流量进行预测。对网络流量的研究和分

\* 收稿日期:2016-05-23;修回日期:2016-10-09    Received date:2016-05-23;Revised date:2016-10-09

基金项目:国家自然科学基金资助项目(61271260);重庆市科委自然科学基金资助项目(cstc2015jcyjA40050)

\*\* 通信作者:songyh1110@163.com    Corresponding author:songyh1110@163.com

析,一方面可以帮助“数据”选择更好的链路,进行更快地收发信息;另一方面可以提高整个系统的数据传送效率,防止链路或节点的数据拥堵,提高用户服务质量等。设计一个具有高精度的网络流量的预测模型,可以提高认知网络的认知功能<sup>[1-2]</sup>。

针对流量预测,目前国内外的研究中使用较多的预测方法是自回归滑动平均模型<sup>[3-4]</sup>。然而,该方法在参数和表达式确定上的困难,使得回归分析在某些情况下的应用受到限制,造成预测的准确度不是很高,并且该方法往往是对单个对象点的预测。此外,基于深度学习(Deep Learning, DL)<sup>[5-9]</sup>的模型相继被提出。文献[1]在对多信道流量的预测研究中使用了一个多层感知的前馈神经网络模型,由多个编码器叠加而成,该模型属于前馈深度网络(Feed-Forward Deep Network, FFDN)<sup>[10]</sup>。文献[11]使用深度信任网络(Deep Belief Network, DBN)对交通流量进行预测的研究中,提出使用分组的思想,而该文献是在预测模块对权值进行分组。也有众多学者倾向于浅层学习预测模型<sup>[12-15]</sup>,如文献[12]采用隐马尔科夫和传统神经网络模型对单个无线节点的流量进行对比预测,最终结果显示,神经网络模型具有更优的预测性能。然而,目前的研究也存在一定的局限性:一是预测对象较为单一,没有进行多维度问题的考虑;二是浅层预测模型简单但预测精度相对较低;三是深层学习的模型较为复杂。

为了进一步提高预测精度简化模型结构,本文提出了一种基于栈式自编码的深度学习模型,来构造一种多输入多输出的预测模型,解决了预测对象单一的问题。栈式自编码的模型结构独特的训练机制使得该模型的结构相对于 FFDN 和 DBN 模型简单容易实现,而且预测精度也高于浅层学习预测模型。本文采用输入分组的方法可有效地将具有相关性的预测对象进行分组;同时,采用更加有效的参数更新方法来加速模型的训练,也为预测精度的提高做了进一步的铺垫。

## 2 系统模型

本文的系统预测模型建立在栈式自编码深度学习模型的基础之上,如图 1 所示。

深度神经网络模型输入层到隐含层之间,隐含层  $i \sim i+1$  和隐含层到输出层之间都是通过非线性函数的线性组合来实现输入  $R^n$  到输出  $R^m$  的映射

关系。通过构建多输入、多输出和多隐含层模型,可以深度刻画多个链路之间的时间维度和频率维度之间的非线性关系。另外,在输入层通过对链路进行自适应分组,对数据进行分类学习和预测,来建立一个自适应分组的深度学习模型。

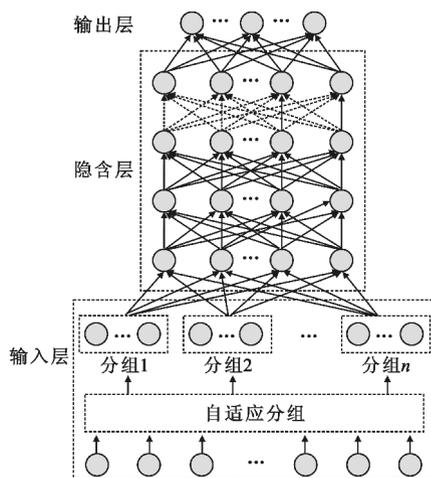


图 1 自适应分组深度学习模型

Fig. 1 Adaptive grouping deep learning model

## 3 基于 AG-SAEs 的流量预测方法

### 3.1 自适应分组算法

在无线网络中,“数据”往往选择状态较好的链路,因此我们根据链路上的流量值进行链路关联性分组,提出一种自适应分组(Adaptive Grouping, AG)算法将多个链路进行自适应分组。

AG 算法执行步骤如下:

**Step 1** 分别计算  $k$  条链路的流量均值,并设定初始质心个数为  $k$ ,初始质心

$$x' = \frac{1}{P} \sum_{i=1}^P x_i \quad (1)$$

为各条链路数值的均值,其中,  $P$  为链路上数据的个数。

**Step 2** 分别计算各个链路或分组的质心之间的距离:

$$O_{j,l} = \|x'_j - x'_l\|^2, j, l = 1, 2, \dots, k. \quad (2)$$

**Step 3** 如果  $O_{j,l} \leq 0$ , 那么将两个对象进行合并,得到一个分组  $G_o: \{j, l \in G_o\}$ , 并更新分组的质心  $x'_o$ ; 如果  $O_{j,l} > 0$ , 那么,各自分别组成一个分组  $G_j: \{j \in G_j\}$  和  $G_l: \{l \in G_l\}$ , 且各分组的质心为组内链路数据的质心  $x'_j$  和  $x'_l$ 。其中,  $0$  为预先设定的

一个阈值,其值越大分组个数越少,其值越小,分组个数越多。

**Step 4** 按照下面的公式更新质心:

$$\begin{cases} x'_{o'} = \frac{1}{2}(x'_{j'} + x'_{l'}), & x'_{o'} \in G_{o'}, O_{j,l} \leq 0 \\ x'_{o'} = x'_{j'}, & x'_{j'} \in G_j, O_{j,l} > 0 \\ x'_{o'} = x'_{l'}, & x'_{l'} \in G_l, O_{j,l} > 0 \end{cases} \quad (3)$$

**Step 5** 重复 Step 2 ~ Step 4,直到分组或质心不再发生变化,然后,跳转至 Step 6。

**Step 6** 将得到的分组输入到栈式自编码模型中。

### 3.2 栈式自编码模型

栈式自编码(Stacked Auto-encoders, SAEs)深度学习模型是一个多隐含层网络。模型训练过程中采用贪婪式的逐层训练机制<sup>[16]</sup>。对于一个训练样本数据  $\{x^1, x^2, x^3, \dots\}$ ,  $x^i \in \mathbb{R}^d$ , 把  $x^i$  作为输入数据输入到隐含层中,通过公式(4)得到编码的结果  $y(x^i)$ ,然后使用公式(5)得到解码后的输出  $z(x^i)$ 。

$$y(x^i) = f(W_1 x^i + b), \quad (4)$$

$$z(x^i) = g(W_2 y(x^i) + c). \quad (5)$$

式中:  $W_1$  是编码权值矩阵,  $b$  是偏置向量,  $W_2$  是解码权值矩阵,  $c$  是解码偏置向量。我们使用 sigmoid 函数作为编码和解码的传递函数:

$$f(x) = g(x) = \frac{1}{1 + \exp(-x)}. \quad (6)$$

通过最小化代价函数来获得模型的参数,即权值矩阵和偏置向量。代价函数表示为

$$L(X, Z) = \operatorname{argmin}_{x, w \in (0, 1)} \frac{1}{2} \sum_{i=1}^N \|x^i - z(x^i)\|^2 + \frac{\lambda}{2} J_{w^l}, \quad (7)$$

$$J_{w^l} = \sum_{i=1}^{n_l-1} \sum_{j=1}^{s_{l+1}} (W_{ji}^l)^2. \quad (8)$$

式中:  $N$  为输入的样本总数,  $\lambda$  为权值矩阵的权重参数,  $n_l$  表示权值矩阵的个数,  $W_{ji}^l$  表示第  $l$  个权值矩阵,  $s_l$  和  $s_{l+1}$  分别表示相邻的两个隐含层中的神经元的个数。代价函数的第一项表示单层自编码输出数据和输入数据之间的残差;第二项表示模型的权重衰减项,其作用是减小权重的幅度,防止过度拟合。这里引入稀疏性限制<sup>[17-18]</sup>后,对于神经元  $j$ , 它的平均活跃度表示为

$$\rho'_j = \frac{1}{N} \sum_{i=1}^N y_j(x^i). \quad (9)$$

式中:  $y_j(x^i)$  表示在输入为  $x^i$  时神经元  $j$  的活跃度,  $N$  为输入的样本总数。稀疏性惩罚因子形式如下:

$$KL(\rho \parallel \rho'_j) = \rho \ln \frac{\rho}{\rho'_j} + (1 - \rho) \ln \frac{1 - \rho}{1 - \rho'_j}. \quad (10)$$

在加入了稀疏性惩罚项后,代价函数的形式变为

$$H = L(X, Z) + \gamma KL(\rho \parallel \rho'_j). \quad (11)$$

式中:  $\gamma$  是稀疏项权重。那么,我们需要通过训练来最小化  $H$  以获得模型的参数  $W$  和  $b$ 。

在 SAEs 的最顶层设置一个逻辑回归预测器,本层不需考虑稀疏性的影响。

### 3.3 改进型 LBFGS 法权值更新规则

自编码网络的目标函数(即代价函数)表示为

$$H(X, Z) = \frac{1}{2} \sum_{i=1}^N \|y_i - y'_i\|^2 + \frac{\lambda}{2} J_{w^l} + \gamma KL(\rho \parallel \rho'_j). \quad (12)$$

式中:  $x_i$  表示目标数据的值,  $z(x^i)$  表示预测的数据值。在逐层训练的过程中,原始数据通过单层模型后得到一个输出,以目标函数  $H$  来表征系统预测性能。在最小化目标函数时,采用 LBFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) 法来进行权值矩阵和偏置向量的更新。权值更新规则表示为

$$W^m(k+1) = W^m(k) - A_k \frac{\partial H}{\partial W^m}, \quad (13)$$

$$b^m(k+1) = b^m(k) - A_k \frac{\partial H}{\partial b^m}. \quad (14)$$

式中:  $A_k$  为赫森矩阵的逆。损失函数  $H$  对权值参数的偏导数计算如下:

$$\begin{aligned} \frac{\partial H}{\partial W_{j,i}^m} &= \frac{\partial H}{\partial f_m} \frac{\partial f_m}{\partial W_{j,i}^m} = \\ &= -(y_{m,i} - y'_{m,i}) \cdot f'_m \cdot y'_{m-1,i} + \lambda W_{j,i}^m = \\ &= -\delta_j^m \cdot y'_{m-1,i} + \lambda W_{j,i}^m, \end{aligned} \quad (15)$$

$$\frac{\partial H}{\partial b_j^m} = \frac{\partial H}{\partial f_m} \frac{\partial f_m}{\partial b_j^m} = -(y_{m,i} - y'_{m,i}) \cdot f'_m = -\delta_j^m. \quad (16)$$

式中:  $m$  表示第  $m$  层。残差的反向传递表示为

$$\delta^m = \delta^{m+1} \cdot W^{m+1} \cdot f'_m. \quad (17)$$

### 3.4 LBFGS 算法收敛性分析

这里用  $h(x)$  表示目标函数(12),即

$$\min_{x \in \mathbb{R}^n} h(x) = \min_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \sum_{i=1}^N \|x_i - z(x^i)\|^2 + \frac{\lambda}{2} J_{w^l} + \gamma KL(\rho \parallel \rho'_j) \right\}. \quad (18)$$

式中:  $h: \mathbb{R}^n \rightarrow \mathbb{R}$  二次连续可微。在 BFGS 算法中赫森矩阵的逆表示为

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k^T s_k \quad (19)$$

式中:  $\rho_k = \frac{1}{y_k^T s_k}$ ,  $V_k = I - \rho_k y_k s_k^T$ 。若在第  $k$  步存储了  $m$

对最近的  $(s_i, y_i)$ ,  $i = k - m + 1, L, K$ , 所以 LBFGS 中的  $H_{k+1}$  为

$$H_{k+1} = (V_k^T \cdots V_{k-m+1}^T) H_{k+1}^{(0)} (V_{k-m+1}^T \cdots V_k^T) + \rho_{k-m+1} (V_{k-1}^T \cdots V_{k-m+2}^T) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2}^T \cdots V_{k-1}^T) + \cdots + \rho_k s_k s_k^T \quad (20)$$

步长因子  $\alpha_k$  由 Wolfe 准则确定, 在非单调策略下, 记  $R_k = \eta_k h_{I(k)} + (1 - \eta_k) h_k^{[19]}$ , 那么步长因子满足的条件为

$$\begin{cases} h(x_k + \alpha_k d_k) \leq R_k + \delta_1 \alpha_k g_k^T d_k \\ g(x_k + \alpha_k d_k)^T d_k \geq \max\{\delta_2, 1 - (\alpha_k \|d_k\|^p) g_k^T d_k\} \end{cases} \quad (21)$$

式中:  $p \in (-\infty, 1)$ ,  $\delta_1, \delta_2 \in (0, 1)$ 。

**定理 1** 如果假设 1 成立, 并且序列  $\{x_k\}$  为算法产生的序列, 则  $\{h(x_{I(k)})\}$  收敛。

证明: 假设 (1) 水平集  $\Omega = \{x | h(x) \leq h(x_0)\}$  有界; (2) 函数  $h(x)$  在  $\Omega$  上二次连续可微; (3)  $h(x)$  一致凸, 即在两个正数  $N_1$  和  $N_2$  使得对任何  $z \in \mathbb{R}^n$  和  $x \in \Omega$  有

$$N_1 \|z\|^2 \leq z^T G(x) z \leq N_2 \|z\|^2$$

由  $R_k$  和  $h(x_{I(k)})$  的定义及线搜索准则 (21) 得

$$\begin{aligned} R_k &= \eta_k h(x_{I(k)}) + (1 - \eta_k) h_k \leq \\ &\eta_k h(x_{I(k)}) + (1 - \eta_k) h(x_{I(k)}) = \\ &h(x_{I(k)}), \end{aligned}$$

$$\begin{aligned} h(x_{I(k+1)}) &= \max_{0 \leq j \leq n(k+1)} \{h_{k-j+1}\} \leq \\ &\max_{0 \leq j \leq n(k)+1} \{h_{k-j+1}\} = \\ &\max\{h(x_{I(k)}), h_{k+1}\} \leq \\ &\max\{h(x_{I(k)}), R_k\} \leq \\ &h(x_{I(k)}) \end{aligned}$$

故  $\{h_{I(k)}\}$  单调下降, 又

$$h_{k+1} \leq h_{I(k+1)} \leq h_{I(k)} \leq h_0,$$

即序列  $\{x_k\}$  含于  $\Omega$  中, 所以  $\{h(x_{I(k)})\}$  收敛。

### 3.5 算法步骤

在对系统模型进行训练时, 为了防止梯度扩散情况的出现, 采用贪婪式的逐层训练方法, 自底向上对整个网络进行训练。这种训练方式可概括为两个过程: 通过训练来初始化模型参数和模型参数微调。AG-SAEs 模型的逐层训练过程如图 2 所示。

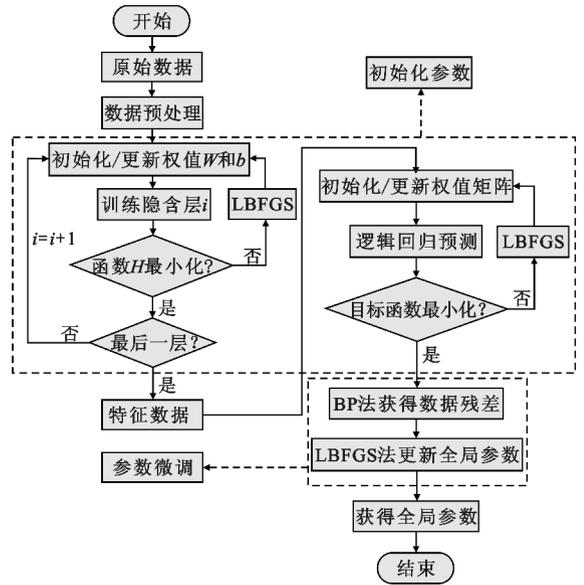


图 2 AG-SAEs 模型训练流程图

Fig. 2 Training flow chart of AG-SAEs model

### 3.6 算法复杂度分析

下面对算法的复杂度进行分析和对比。迭代的次数为  $T$ , 模型层数为  $K$ , 问题维度为  $D$ , 那么, 数据进行分组运算的计算量为  $O(D)$ , 每迭代一次, 误差的反向传播为  $O(D)$ , 参数更新为  $O(D)$ , 每一个隐含层编码和解码的和为  $O(2D)$ 。因此, 本文算法的综合计算复杂度为  $O(4T \times K \times D)$ 。算法复杂度比较如表 1 所示,  $M$  为 RBF 的中心个数。由表可知, 本文所提出的方法在复杂度方面同其他方法相比基本都处于同一个量级, 并没有成规模地增加。

表 1 不同算法复杂度比较

Tab. 1 Complexity comparison among different algorithms

算法	复杂度
AG-SAEs	$O(4T \times K \times D)$
BP	$O(3T \times D)$
RBF	$O(2T \times M \times D)$
LR	$O(2T \times D)$

## 4 仿真实验与分析

### 4.1 数据预处理

本次仿真实验使用某网络公司提供的 11 个链路的无线数据, 分为训练部分 (2 764 组) 和测试部

分(1 373 组)。在预处理过程中,按照公式(22)对数据进行归一化处理:

$$\bar{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (22)$$

在对流量进行预测时,通过使用前  $k$  个数据来预测后面一个,采用滑动窗口方式(窗口大小为  $k$ ,每次滑动 1 个单位)对训练数据进行重构来产生输入数据向量,如表 2 所示。

表 2 训练数据重构

Tab. 2 Training data reconstruction

输入数据向量	目标数据
$x_1, x_2, \dots, x_k$	$x_{k+1}$
$x_2, x_3, \dots, x_{k+1}$	$x_{k+2}$
$\vdots$	$\vdots$
$x_{N-k}, x_{N-k+1}, \dots, x_{N-1}$	$x_N$

### 4.2 预测性能指标

我们使用平均绝对误差 (Mean Absolute Error, MAE)、平均相对误差 (Mean Relative Error, MRE) 和均方根误差 (Root Mean Square Error, RMSE) 来评价模型的性能。3 个指标的表达式分别为

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - z(x^i)|, \quad (23)$$

$$MRE = \frac{1}{n} \sum_{i=1}^n \frac{|x_i - z(x^i)|}{z(x^i)}, \quad (24)$$

$$RMSE = \left( \frac{1}{n} \sum_{i=1}^n |x_i - z(x^i)|^2 \right)^{\frac{1}{2}}. \quad (25)$$

式中:  $x_i$  表示目标数据,  $z(x^i)$  表示预测的流量数据值。

### 4.3 算法收敛性

通过仿真实验,我们可以获得前面给出的目标函数在每一次迭代后的值。

图 3 显示的是在本文方法 (LBFGS)、共轭梯度法 (CGD) 和梯度下降法 (GD)<sup>[20]</sup> 这 3 种参数更新方法下的系统目标函数与迭代次数之间的关系图。从图中可以看出,目标函数开始处于一个较大的值,然而随着算法不断迭代和更新参数,最终结果快速收敛到一个较小的值,并且本文方法相比其他两种方法具有更快的收敛速度。

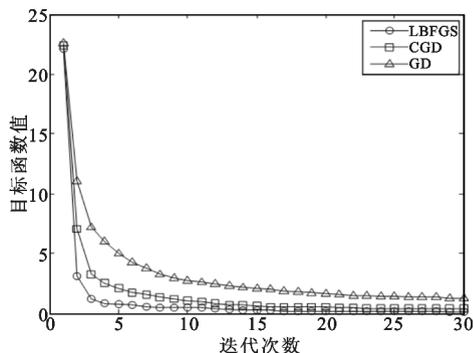


图 3 目标函数与迭代次数关系图

Fig. 3 Relationship between objective function and iteration number

### 4.4 不同的隐含层和单元个数性能分析

深度神经网络的模型结构对系统的性能影响主要是模型的层数和节点单元个数,我们使用这个模型分别进行 3、5 和 10 步的预测。隐含层的个数采用 1~7 层,隐含层节点的个数采用 60~300 之间的数,通过实验得到的 3 种情况下的最优模型结构,如表 3 所示。

表 3 模型结构

Tab. 3 Structure of the model

$k$	隐含层数	单元数
3	2	[70, 70]
5	2	[110, 110]
10	3	[220, 220, 220]

对于 3 步的预测,最优的结构是 2 个隐含层,且每一层的单元数是 [70, 70]。在 5 步预测中,最优模型结构是 2 个隐含层,每一层的单元数是 [110, 110]。在 10 步的预测中,模型结构为 3 个隐含层,每一层的单元数是 [220, 220, 220]。可以看出,隐含层的个数不能太多,当所讨论的问题规模不是很大时,隐含层个数一般在 4 层以内进行选择。

确定了模型的结构后,我们使用本文提出的模型在 3 种情况下进行性能测试。图 4 是在不同的步长预测下,不同的隐含层个数的相对误差曲线对比。从图中可以看出,就本文所研究的数据,合适的隐含层个数为 4 层以下。随着隐含层个数的逐渐增加,预测的相对误差也逐步增加。当隐含层的个数达到 7 层时, MRE 已经增加了将近 1 倍。因此,本文选择 2~3 个隐含层来建立模型。产生这种现象的原因是,单个隐含层的网络模型结构简单,不适合进行具有较多数据的训练和预测。隐含层个数多的时候,由于在微调部分中会有误差的传递,尽管具有较小

的训练误差,但是会产生较大的泛化误差。因此,当隐含层个数逐渐增加时,预测误差会先下降后上升。

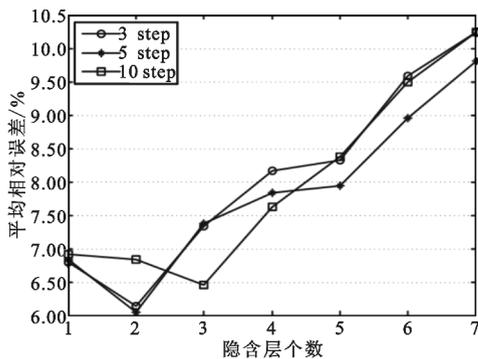


图4 隐含层个数性能对比

Fig. 4 Performance comparison of different hidden size

#### 4.5 AG-SAEs 和 SAEs 对比

我们使用已经确定的模型进行数据预测。首先,对比本文提出的 AG-SAEs 和 SAEs 在相同条件下的预测结果。

由表 4 可知,本文通过对研究对象进行关联性自适应分组,模型的整体性能优于原栈式自编码模型。从数据上看,本文所提出的模型在预测平均相对误差上提高了将近一个百分点。所以,在使用深度学习对数据进行学习和预测时,对原数据做自适应分类和分组有助于提升系统的预测准确度。

表 4 AG-SAEs 和 SAEs 性能对比

Tab. 4 Performance comparison between AG-SAEs and SAEs

k	MRE/%	
	AG-SAEs	SAEs
3	6.14	6.99
5	6.04	6.96
10	6.45	6.88

#### 4.6 优化方法对比

本文采用改进型的 LBFGS 法对模型的参数进行更新,相比梯度下降法(GD),其具有更快的收敛速度和更优的收敛方向。图 5 是本文所采用的方法和梯度下降法以及共轭梯度法(CGD)的预测相对

误差对比。

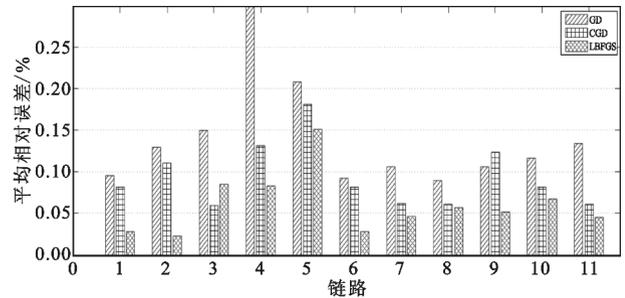


图 5 不同的参数更新方法的 MRE 对比

Fig. 5 MRE comparison among updating methods with different parameter

从图 5 可以看出,本文所采用的方法的相对误差明显小于梯度下降法和共轭梯度法。从数据上看,本文采用方法的平均相对误差为 6.04%,共轭梯度法的相对误差为 9.36%,梯度下降法的平均相对误差为 15.68%。梯度下降法是基于目标函数梯度的线性收敛,当问题规模较大时,收敛速度尤其慢,而本文方法确保了近似矩阵的正定性,总是朝着最优值收敛。因此,本文所采用的方法具有更快的收敛速度和较正确的收敛方向。

#### 4.7 算法性能对比

我们把本文模型分别和反向传播(Back Propagation, BP)神经网络、径向基函数(Radial Basis Function, RBF)神经网络和逻辑回归(Logistic Regression, LR)法进行对比。神经网络方法在流量预测上具有很好的性能,而回归分析也是一种较常使用的数值预测方法。BP 法使用 sigmoid 激活函数进行数值逼近,通过反向传播误差来调节参数。RBF 神经网络方法的基函数采用距离函数,使用高斯函数作为激活函数。BP 和 RBF 法的权值调节方法采用负梯度下降法,这种方法存在收敛速度慢和易出现局部极小值的问题<sup>[21-22]</sup>。LR 方法的损失函数模型中,由于参数权重问题,容易出现过拟合现象,导致泛化能力相对差一些。表 5 为在本文方法和另外 3 种方法的预测性能对比。

表 5 算法和步长的性能对比

Tab. 5 Performance comparison of algorithms and step sizes

算法	MAE			MRE/%			RMSE		
	k=3	k=5	k=10	k=3	k=5	k=10	k=3	k=5	k=10
AG-SAEs	2.34	2.29	2.56	6.14	6.04	6.45	3.11	3.05	3.28
BP	4.51	4.02	3.68	11.66	10.39	9.50	5.69	5.07	4.64
RBF	5.01	5.06	5.09	12.95	13.17	13.26	6.32	6.35	6.39
LR	5.36	5.23	4.99	14.01	13.72	12.91	6.65	6.51	6.28

从表 5 看,对相同的数据进行预测,在不同的预测步长下,栈式自编码模型具有更好的相对误差。3 种情况下的预测准确度都已经超过了 93%,并且具有更小的绝对误差和均方根误差。AG-SAEs 模型首先对输入数据进行分组,然后通过特征学习,将提取出来的特征数据送入到预测层进行预测,这种处理方式在相对误差上比反向传播法减少了 4%~5%,比径向基函数法和逻辑回归法减少了 6%~8%。比较这 4 种方法,可以看出 AG-SAEs 模型具有更好的鲁棒性。这 3 种情况下的预测相对误差都在 7% 以下,并且相差不大。其他 3 种方法都会有一些性能上的波动,但是波动性并不是很大,原因在于本文的 3 种预测只是在于步长的不同,即考虑的是预测点之前的  $k$  个数据,单个点的数据值并没有改变。因此,3 种情况下的预测结果相对较稳定。

图 6 显示了分别使用自适应分组的栈式自编码深度学习(DL)方法、反向传播(BP)神经网络方法、径向基函数(RBF)神经网络和逻辑回归(LR)方法下的预测准确度在 90% 以上的数据所占的比率。本文方法在这 3 种情况下的预测都有超过 80% 的数据的预测值准确度高于 90%。BP 法在第一种情况下只有 70% 的预测数据的准确度在 90% 以上,在后两种情况下也都达到了 80%。相比之下,RBF 法和 LR 法准确度较低,LR 法在这 3 种情况下均只有 70% 多的数据准确度超过了 90%,而 RBF 法只有一半的数据预测准确度达到了 90%。因此,可以证实本文方法在流量预测上具有较高的准确度,即较小的预测相对误差。

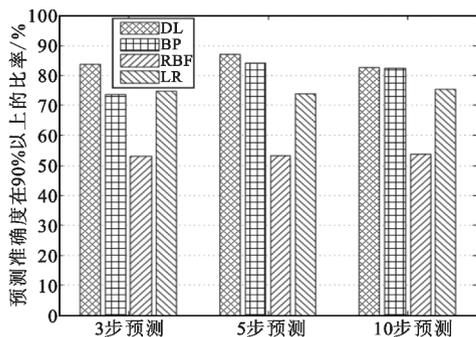


图 6 预测准确度在 90% 以上的数据比率  
Fig. 6 Ratio of the prediction accuracy above 90%

## 5 结束语

我们提出使用 AG-SAEs 深度学习的方法,通过对无线链路进行关联性分组,将得到的分组数据加

载到一个有多隐含层的栈式自编码网络来提取隐含在数据中的潜在非线性特征,以实现对未来流量的预测。这里不仅考虑了时间维度的相关性,同时也考虑了频率维度的相关性。首先通过逐层的方式进行初始化和训练参数,然后进行参数微调来优化系统模型,采用了更为有效的 LBFGS 法来更新参数。我们通过和反向传播神经网络法、径向基函数神经网络法和逻辑回归法进行对比,结果也证实了本方法在预测上具有更小的相对误差。虽然本文方法在计算复杂度上相对其他方法有所增加,但是在预测精度上却有明显的提高。

接下来的工作需要将本文方法进行其他网络环境下数据集的测试,来验证算法的可靠性。同时,对于深度学习,还需要考虑使用不同的预测器进行预测。

## 参考文献:

- [1] LIU Y, TAMMA B R, MANOJ B S, et al. Traffic prediction for cognitive networking in multi-channel wireless networks [C]//Proceedings of 2010 IEEE Conference on Computer Communications Workshops (INFOCOM). San Diego, CA: IEEE, 2010: 1-6.
- [2] 李红岩. 认知无线电系统中频谱可预测性的递归定量分析[J]. 电讯技术, 2015, 55(2): 124-128.  
LI Hongyan. Recurrence quantification analysis of spectrum predictability in cognitive radio systems [J]. Telecommunication Engineering, 2015, 55(2): 124-128. (in Chinese)
- [3] FENG H, SHU Y. Study on network traffic prediction techniques [C]//Proceedings of 2005 IEEE International Conference on Wireless Communications (WiCOM). Montreal, Canada: IEEE, 2005: 1041-1044.
- [4] DARJI M P, DABHI V K, PRAJAPATI H B. Rainfall forecasting using neural network: a survey [C]//Proceedings of 2015 International Conference on Advances in Computer Engineering and Applications (ICACEA). Ghaziabad, India: IEEE, 2015: 706-713.
- [5] HUA Y, GUO J, ZHAO H. Deep belief networks and deep learning [C]//Proceedings of 2014 International Conference on Intelligent Computing and Internet of Things (ICIT). Seville, Spain: IEEE, 2014: 1-4.
- [6] KUANG P, CAO W N, WU Q. Preview on structures and algorithms of deep learning [C] //Proceedings of 2014 11th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICC-WAMTIP). Chengdu: IEEE, 2014: 176-179.
- [7] HUANG W H, HONG H K, SONG G, et al. Deep process neural network for temporal deep learning [C] //Proceedings of 2014 IEEE International Joint Conference on Neural Networks (IJCNN). Beijing: IEEE, 2014: 465-472.

- [8] SPENCER M, EICKHOLT J, CHENG J. A deep learning network approach to ab initio protein secondary structure prediction[J]. IEEE/ACM Transactions on Computational Biology & Bioinformatics, 2015, 12(1): 103–112.
- [9] 周佳俊, 欧智坚. 深层神经网络预训练的改进初始化方法[J]. 电讯技术, 2013, 53(7): 895–898.  
ZHOU Jiajun, OU Zhijian. Improved initialization of pre-training in deep neural network[J]. Telecommunication Engineering, 2013, 53(7): 895–898. (in Chinese)
- [10] 尹宝才, 王文通, 王立春. 深度学习研究综述[J]. 北京工业大学学报, 2015, 41(1): 48–59.  
YIN Baocai, WANG Wentong, WANG Lichun. A survey on deep learning research[J]. Journal of Beijing University of Technology, 2015, 41(1): 48–59. (in Chinese)
- [11] HUANG W, SONG G, HONG H, et al. Deep architecture for traffic flow prediction: deep belief networks with multitask learning[J]. IEEE Transactions on Intelligent Transportation Systems, 2014, 15(5): 2191–2201.
- [12] SIVAKUMAR R, ASHOK KUMAR E, SIVARADJE G. Prediction of traffic load in wireless network using time series model[C]// Proceedings of 2011 International Conference on Process Automation, Control and Computing(PACC). Coimbatore, India: IEEE, 2011: 1–6.
- [13] CHAN K Y, DILLON T S, SINGH J, et al. Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and Levenberg-Marquardt algorithm[J]. IEEE Transactions on Intelligent Transportation System, 2012, 13(2): 644–654.
- [14] KUMAR K, PARIDA M, KATIYAR V K. Short term traffic flow prediction for a nonurban highway using artificial neural network[J]. Procedia – Social and Behavioral Sciences, 2013, 104: 755–764.
- [15] ZHENG W, LEE D H, ZHENG W, et al. Short-term freeway traffic flow prediction: Bayesian combined neural network approach[J]. Journal of Transportation Engineering, 2006, 132(2): 114–121.
- [16] BENGIO Y, LAMBLIN P, POPOVICI D, et al. Greedy layer-wise training of deep networks[C]// Proceedings of 2007 Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2007: 153–160.
- [17] PALM R B. Prediction as a candidate for learning deep hierarchical models of data[D]. Palm, Denmark: Technical University of Denmark, 2012.
- [18] IDYA M, PHIL M. Sparse coding: a deep learning using unlabeled data for high-level representation[C]// Proceedings of 2014 IEEE World Congress on Computing and Communication Technologies(WCCCT). Tiruchirappalli, India: IEEE, 2014: 124–127.
- [19] AHOOKHOSH M, AMINI K. An efficient nonmonotone trust-region method for unconstrained optimization[J]. Numerical Algorithms, 2012, 59(4): 523–540.
- [20] WU M, CHEN L. Image recognition based on deep learning[C]// Proceedings of 2015 Chinese Automation Congress(CAC). Wuhan: IEEE, 2015: 1–6.
- [21] WANG S, SONG T. The application on the forecast of steam turbine exhaust wetness fraction with GA BP neural network[C]// Proceedings of 2012 World Automation Congress(WAC). Puerto Vallarta, Mexico: IEEE, 2012: 1–5.
- [22] QU J, ZHOU J, CHEN N. Groundwater table prediction based on improved PSO algorithm and RBF neural network[C]// Proceedings of 2010 International Conference on Artificial Intelligence and Computational Intelligence. Sanya: IEEE, 2010: 228–232.

### 作者简介:



朱江(1977—),男,湖北人,2009年于电子科技大学获博士学位,现为副教授,主要研究方向为认知无线电、移动通信;

Email: zhujiang@cqupt.edu.cn

宋永辉(1991—),男,河北人,硕士研究生,主要研究方向为认知无线电、网络优化;

Email: songyh1110@163.com

刘亚利(1989—),男,河南人,硕士研究生,主要研究方向为认知无线电、网络优化。