



(12) 发明专利申请

(10) 申请公布号 CN 104900063 A

(43) 申请公布日 2015. 09. 09

(21) 申请号 201510345977. 1

(22) 申请日 2015. 06. 19

(71) 申请人 中国科学院自动化研究所
地址 100080 北京市海淀区中关村东路 95 号

(72) 发明人 熊刚 亢文文 朱凤华 王飞跃
吕宜生 董西松

(74) 专利代理机构 北京博维知识产权代理事务所 (特殊普通合伙) 11486
代理人 方振昌

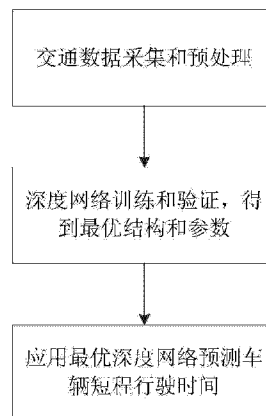
(51) Int. Cl.
G08G 1/01(2006. 01)
G06N 3/02(2006. 01)

权利要求书1页 说明书6页 附图2页

(54) 发明名称
一种短程行驶时间预测方法

(57) 摘要

本发明公开一种短程行驶时间预测方法,包括以下步骤:步骤 S1:利用交通数据采集装置进行数据采集并对采集的数据采用归一化方法进行预处理;步骤 S2:应用预处理好的数据,训练堆叠自编码器深度网络,得到最优网络结构和对应参数,生成最优堆叠自编码器深度网络;步骤 S3:调用训练好的最优堆叠自编码器深度网络预测车辆短程行驶时间。本发明能够深度挖掘输入变量之间的非线性关系和其它内在特征,预测精度高,鲁棒性好。



1. 一种短程行驶时间预测方法,其特征在於,包括以下步骤:

步骤 S1 :利用交通数据采集装置进行数据采集并对采集的数据采用归一化方法进行预处理;

步骤 S2 :应用预处理好的数据,训练堆叠自编码器深度网络,得到最优网络结构和对应参数,生成最优堆叠自编码器深度网络;

步骤 S3 :调用训练好的最优堆叠自编码器深度网络预测车辆短程行驶时间。

2. 如权利要求 1 所述的一种短程行驶时间预测方法,其特征在於,步骤 S1 所采集的数据包括车辆行驶速度、交通流量、交通流密度或车道平均车辆数、信号时间以及车辆在预测起止点之间的实际行驶时间。

3. 如权利要求 1 所述的一种短程行驶时间预测方法,其特征在於,步骤 S1 对采集的数据采用归一化方法进行预处理的具体方法为:选取待归一化变量的最小值 v_{\min} 和最大值 v_{\max} ,对 $[v_{\min}, v_{\max}]$ 之间的任意值 v ,其归一化后的值 v_n 为

$$v_n = \frac{v - v_{\min}}{v_{\max} - v_{\min}} \quad \circ$$

4. 如权利要求 1 所述的一种短程行驶时间预测方法,其特征在於,步骤 S2 生成最优堆叠自编码器深度网络的步骤包括:

步骤 S21 :将归一化处理后的数据按比例分为训练集和测试集;

步骤 S22 :设置堆叠自编码器深度网络的网络结构:包括组成堆叠自编码器的自编码器个数 L ,以及每一个自编码器隐藏层节点个数 n_l , $l = 1, 2, \dots, L$;

步骤 S23 :利用训练数据集,非监督地逐层训练每个自编码器的参数;

步骤 S24 :将步骤 S23 中训练好的自编码器堆叠起来,并在顶层加入预测器,再次利用训练数据,有监督地微调深度网络各层参数;

步骤 S25 :利用测试数据集,在训练好的深度网络上进行前向计算,得到测试集上的平均绝对预测误差;

步骤 S26 :改变自编码器个数 L 以及每一个自编码器隐藏层节点个数 n_l 的数值,重复步骤 S22 ~ S25 ;选取最小误差对应的网络结构和参数作为堆叠自编码器深度网络的最优网络结构和参数。

5. 如权利要求 1 所述的一种短程行驶时间预测方法,其特征在於,所述步骤 S3 进一步包括以下步骤:

步骤 S31 :当新的车辆到达时,采集需要的输入数据并以同样的参数做归一化处理;

步骤 S32 :将归一化后的输入数据带入训练好的深度网络进行前向计算,得到对应的归一化预测值;

步骤 S33 :将步骤 S32 得到的归一化预测值反归一化,得到车辆在预测起止点之间的预测行驶时间。

一种短程行驶时间预测方法

技术领域

[0001] 本发明属于智能交通系统领域,特别涉及一种短程行驶时间预测方法。

背景技术

[0002] 在公交信号优先 (Transit Signal Priority, TSP) 系统中,往往需要在路口前一定距离处检测公交车的到达,并预测公交车行至路口的时间,以便给予及时、准确的信号优先控制。许多研究者和工程师都认识到,精确的行驶时间预测是成功实施公交信号优先的关键。

[0003] 现有的车辆行驶时间预测模型主要有四类,分别是朴素模型、基于交通理论的模型、基于数据的模型和混合模型。朴素模型不需要任何训练和参数估计,只是简单地将最近发生的行驶时间作为下一预测值,或者对历史行驶时间进行简单的筛选和加权作为预测值,或者二者的结合;这种方法不能适应复杂多变的交通环境。基于交通理论的模型试图在每一个时间步里重建交通状态,然后从状态变量推测行驶时间,这类模型通常在各类交通仿真软件里面实现,例如 CORSIM、PARAMICS 和 DynaMIT 等,它们可以详尽地描述交通状态的细节,但是建立这类模型往往需要大量的专业知识。这类模型的预测精度取决于重建和实际的交通状态之间的相似性,但是由于现实世界的复杂性,这种相似性往往不能保证。得益于统计和机器学习方法的发展,涌现出许多基于数据的预测模型并得到广泛应用,这类模型的目的是找到一种恰当的数据解释函数 f 及其参数,以最小的误差将输入映射到输出,其中人工神经网络 (Artificial Neural Network, ANN) 模型具有强大的表述数据中非线性关系的能力,得到了广泛的研究,并衍化出多种变体。而混合模型一般将两种或多种不同类型的模型组合,集取各自的优势以进一步提高预测性能,例如将季节性自回归移动平均模型和自适应卡尔曼滤波结合构成多步预测器,利用贝叶斯规则解释多个 ANN 模型输出的混合 ANN 模型等。涉及混合模型的研究相对较少,此类模型的预测性能还有待进一步验证。

[0004] ANN 模型自从被提出以来就在预测领域得到重视和快速发展,因为建立这类模型不需要太多领域相关专业知识和预测性能一直在不断提高。传统的浅层 ANN 模型需要大量的训练数据以提高精度,然而数据量过大又容易陷入过拟合。增加网络层次可以解决这个问题,但是传统的训练方法面对深层网络收敛速度缓慢。最近几年深度学习网络研究的突破解决了这一矛盾,它被证明能够有效地解决预测和其它许多问题。

发明内容

[0005] 本发明针对短程行驶时间预测的实际应用需求,提出一种短程行驶时间预测方法,能够深度挖掘输入变量之间的非线性关系和其它内在特征,预测精度高,鲁棒性好。

[0006] 本发明提出的一种短程行驶时间预测方法,包括以下步骤:

[0007] 步骤 S1:利用交通数据采集装置进行数据采集并对采集的数据采用归一化方法进行预处理;

[0008] 步骤 S2 :应用预处理好的数据,训练堆叠自编码器深度网络,得到最优网络结构和对应参数,生成最优堆叠自编码器深度网络;

[0009] 步骤 S3 :调用训练好的最优堆叠自编码器深度网络预测车辆短程行驶时间。

[0010] 优选的,步骤 S1 所采集的数据包括车辆行驶速度、交通流量、交通流密度或车道平均车辆数、信号时间以及车辆在预测起止点之间的实际行驶时间。

[0011] 优选的,步骤 S1 对采集的数据采用归一化方法进行预处理的具体方法为:选取待归一化变量的最小值 v_{\min} 和最大值 v_{\max} ,对 $[v_{\min}, v_{\max}]$ 之间的任意值 v ,其归一化后的值 v_n 为

$$[0012] \quad v_n = \frac{v - v_{\min}}{v_{\max} - v_{\min}} \circ$$

[0013] 优选的,步骤 S2 生成最优堆叠自编码器深度网络的步骤包括:

[0014] 步骤 S21 :将归一化处理后的数据按比例分为训练集和测试集;

[0015] 步骤 S22 :设置堆叠自编码器深度网络的网络结构:包括组成堆叠自编码器的自编码器个数 L ,以及每一个自编码器隐藏层节点个数 n_1 ;

[0016] 步骤 S23 :利用训练数据集,非监督地逐层训练每个自编码器的参数;

[0017] 步骤 S24 :将步骤 S23 中训练好的自编码器堆叠起来,并在顶层加入预测器,再次利用训练数据,有监督地微调深度网络各层参数;

[0018] 步骤 S25 :利用测试数据集,在训练好的深度网络上进行前向计算,得到测试集上的平均绝对预测误差;

[0019] 步骤 S26 :改变自编码器个数 L 以及每一个自编码器隐藏层节点个数 n_1 的数值,重复步骤 S22 ~ S25 ;选取最小误差对应的网络结构和参数作为堆叠自编码器深度网络的最优网络结构和参数。

[0020] 优选的,所述步骤 S3 进一步包括以下步骤:

[0021] 步骤 S31 :当新的车辆到达时,采集需要的输入数据并以同样的参数做归一化处理;

[0022] 步骤 S32 :将归一化后的输入数据带入训练好的深度网络进行前向计算,得到对应的归一化预测值;

[0023] 步骤 S33 :将步骤 S32 得到的归一化预测值反归一化,得到车辆在预测起止点之间的预测行驶时间。

[0024] 本发明通过构造基于堆叠自编码器的深度网络预测模型,充分利用实际产生的大量交通数据训练模型,可以自动地深度挖掘输入变量之间的非线性关系和其它内在特征。本发明具有通用性,能够轻易部署在不同的预测地点,并且具有秒级的预测精度和较高的鲁棒性。

附图说明

[0025] 图 1 是本发明短程行驶时间预测方法框架示意图;

[0026] 图 2 是本发明生成最优堆叠自编码器深度网络的流程示意图;

[0027] 图 3 是单隐藏层自编码器结构示意图;

[0028] 图 4 是本发明用于预测的深度网络结构示意图。

具体实施方式

[0029] 下面将结合附图对本发明加以详细说明,应指出的是,所描述的实施例仅旨在便于对本发明的理解,而对其不起任何限定作用。

[0030] 如图 1 所示,本发明的一种短程行驶时间预测方法具体地包括以下步骤:

[0031] 步骤 S1:利用交通数据采集装置进行数据采集并对采集的数据采用归一化方法进行预处理。

[0032] 本方法训练阶段需要采集的数据包括车辆行驶速度、交通流量、交通流密度或车道平均车辆数、信号时间以及车辆在预测起止点之间的实际行驶时间。其中车辆行驶速度和交通流量可以通过环形线圈检测器直接得到。交通流密度或车道平均车辆数可以通过在预测起止点处分别设置环形线圈检测器,然后统计经过两个检测器的车辆数之差得到(车道数为已知常量)。信号时间可以通过连接路口信号控制器得到。预测起止点之间的实际行驶时间可以通过检测器时间戳信息得到。除了环形线圈检测器外,还可以采用更高级的检测器取得以上数据,例如视频检测器、GPS 装置和 AVL 装置等。

[0033] 接下来对采集的数据进行预处理,采用归一化方法。对于某个待归一化的变量,取其最小值 v_{\min} 和最大值 v_{\max} ,则对 $[v_{\min}, v_{\max}]$ 之间的任意值 v ,其归一化后的值 v_n 如公式 (1) 所示。

$$[0034] \quad v_n = \frac{v - v_{\min}}{v_{\max} - v_{\min}} \quad (1)$$

[0035] 对所有变量(包括输出变量)做以上归一化处理得到归一化后的数据集。

[0036] 步骤 S2:应用预处理好的数据,训练堆叠自编码器深度网络,得到最优网络结构和对应参数,生成最优堆叠自编码器深度网络。该过程流程图如图 2 所示,其中 AE 表示自动编码器(Auto-Encoders),具体包含以下步骤:

[0037] 步骤 S21,将归一化处理后的数据分为训练集和测试集。

[0038] 如果数据量足够大,可以取 80% 的数据为训练集,20% 的数据为测试集。这里的 20%、80% 划分只是推荐值,实际应用时应该根据具体情况采取合适的划分。如果数据量太少,可以采用 N 折交叉验证方法划分训练集和测试集以充分利用数据。

[0039] 步骤 S22,设置堆叠自编码器深度网络的网络结构:包括组成堆叠自编码器的自编码器个数 L ,以及每一个自编码器隐藏层节点个数 n_1 。

[0040] 步骤 S23,利用训练数据集,非监督地逐层训练每个自编码器的参数;“逐层”意味着将前一个自编码器的编码输出用作后一个自编码器的输入,第一个自编码器的输入来自训练集数据。

[0041] 自编码器是一种输入节点和输出节点个数相同的神经网络,最简单的自编码器只有一个隐藏层,它的结构如图 3 所示。第 k 层的直接输入 $z^{(k)}$ (加粗体表示向量或矩阵,下同)通过函数 $f(\cdot)$ 变换为激活值 $a^{(k)}$, $k = 1, 2, 3$ 。其中第一层即输入层不需要激活变换,且 $z^{(1)}$ 等于输入 x ,因此 $x = z^{(1)} = a^{(1)}$ 。函数 $f(\cdot)$ 称为激活函数,取 sigmoid 函数,如公式 (2) 所示。

$$[0042] \quad f(z) = \frac{1}{1 + \exp(-z)} \quad (2)$$

[0043] 一个向量的函数 $f(z)$ 仍为一个向量,它表示对向量的每一个元素分别求函数值

后组成的向量,即 $f(z) = [z_1, z_2, \dots, z_D]^T$, D 为向量 z 的维数。

[0044] 后一层的直接输入 $z^{(k)}$ 由前一层的激活值 $a^{(k-1)}$ 通过线性组合得到,线性组合参数存放在矩阵 $W^{(k-1)}$ 和向量 $b^{(k-1)}$ 中。模型的输出 \hat{x} 等于最后一层的激活值 $a^{(3)}$ 。

[0045] 以上为神经网络的前向传播过程,用向量和矩阵的形式表示为式 (3)、(4)、(5) :

$$[0046] \quad x = z^{(1)} = a^{(1)} \quad (3)$$

$$[0047] \quad a^{(2)} = f(z^{(2)}) = f(W^{(1)}a^{(1)} + b^{(1)}) \quad (4)$$

$$[0048] \quad \hat{x} = a^{(3)} = f(z^{(3)}) = f(W^{(2)}a^{(2)} + b^{(2)}) \quad (5)$$

[0049] 其中, $W^{(1)}$ 和 $W^{(2)}$ 分别为 $n_1 \times M$ 和 $M \times n_1$ 的矩阵, $b^{(1)}$ 和 $b^{(2)}$ 分别为 n_1 和 M 维的列向量, M 表示输入向量的维数, n_1 表示第一个自编码器隐藏层节点的个数。

[0050] 自编码器不同于普通神经网络之处在于,它通过对输入数据编码再解码重构,从而找到重新表达输入数据的形式。因此,训练自编码器的目标是最小化输入和输出之间的误差,加上正则化项(目的是避免过拟合)后的最小化目标可表示为式 (6) :

$$[0051] \quad J(W, b) = \left[\frac{1}{m} \sum_{q=1}^m \left(\frac{1}{2} \|\hat{x}^{(q)} - x^{(q)}\|^2 \right) \right] + \frac{\lambda}{2} \sum_{i=1}^M \sum_{j=1}^{n_1} \left[(W_{ji}^{(1)})^2 + (W_{ij}^{(2)})^2 \right] \quad (6)$$

[0052] 其中 $W = \{W^{(1)}, W^{(2)}\}$, $b = \{b^{(1)}, b^{(2)}\}$, m 为训练样本个数, λ 为权重参数,用于调整目标函数中重构误差与正则化项的权重。

[0053] 当隐藏层节点个数大于输入节点个数时,自编码器训练结果往往不唯一,且结果严重依赖于初始条件。为了解决这一问题,还需要加入稀疏性限制。我们使用 $a_j^{(2)}(x)$ 表示在给定输入为 x 的情况下,自编码器隐藏层节点 j 的激活度,因此定义隐藏层节点 j 的平均激活度如式 (7) 所示

$$[0054] \quad \hat{\rho}_j = \frac{1}{m} \sum_{q=1}^m a_j^{(2)}(x^{(q)}) \quad (7)$$

[0055] 我们希望平均激活度与 ρ 越接近越好,其中 ρ 是一个接近 0 的正常数,称为稀疏性参数。 $\hat{\rho}_j$ 与 ρ 的接近程度可以用 KL 距离表示,因此加上稀疏性约束的目标函数如式 (8) 所示 :

$$[0056] \quad J_{\text{sparse}}(W, b) = J(W, b) + \beta \sum_{j=1}^{n_1} \text{KL}(\rho \parallel \hat{\rho}_j) \quad (8)$$

[0057] 其中, $\text{KL}(\rho \parallel \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$, β 为权重参数,用于调整 $J(W, b)$ 和稀疏性限制的权重。利用神经网络后向传播算法最小化 $J_{\text{sparse}}(W, b)$ 即可求得参数 W 和 b 。

[0058] 当第一个自编码器训练好后,将其隐藏层节点的输出即编码输出作为新的输入,用同样的方法训练第二个自编码器。依次类推,直到 L 个自编码器全部训练完成。

[0059] 步骤 S24,将步骤 S23 中训练好的自编码器堆叠起来,并在顶层加入预测器,再次利用训练数据,有监督地微调深度网络各层参数 ;

[0060] 将前一个自编码器的隐藏层输出直接连接到后一个自编码器的隐藏层输入即构成堆叠自编码器。加入预测器的堆叠自编码器如图 4 所示,图中, x 为某一个输入样本; $a_j^{(2)}$

为第 1 个自编码器的编码输出, $l = 1, 2, \dots, L$; $\mathbf{W}_l^{(k)}$ 、 $\mathbf{b}_l^{(k)}$ 为第 1 个自编码器的第 k 组参数, $k = 1, 2$; u_0 和 u 为预测器的参数; \hat{y} 为反归一化后的预测器输出。

[0061] 本发明预测器采用逻辑斯蒂 (Logistic) 回归模型, 表示为式 (9):

$$[0062] \quad \sigma(\mathbf{a}_L^{(2)}) = \frac{1}{1 + \exp(-\mathbf{u}_0 - \mathbf{u}^T \mathbf{a}_L^{(2)})} \quad (9)$$

[0063] 其中 $\mathbf{a}_L^{(2)}$ 为堆叠自编码器中最后一个自编码器的编码输出。参数 u_0 和 u 需要通过训练获得。此处再次利用训练数据, 采用有监督的学习方法进一步微调深度网络包括预测器的参数, 依然采用后向传播算法。由于堆叠自编码器已经经过预训练得到了合适的初始值, 因此此处的有监督学习会很快收敛。至此, 深度网络的训练过程全部完成。

[0064] 步骤 S25, 利用测试数据集, 在训练好的深度网络上进行前向计算, 得到测试集上的平均绝对预测误差;

[0065] 前向计算只需在步骤 S23 所述前向传播过程的基础上稍作扩展即可, 并将最后一个自编码器的隐藏层输出带入逻辑斯蒂回归模型, 即可求得输入样本的预测值, 该预测值为归一化值, 记为 \hat{y}_{norm} , 反归一化后得到其真实值 $\hat{y} = \hat{y}_{norm}(y_{max} - y_{min})$ 。因此, 测试集上的平均绝对预测误差如式 (10) 所示:

$$[0066] \quad MAE = \frac{1}{m'} \sum_{p=1}^{m'} |y^{(p)} - \hat{y}^{(p)}| \quad (10)$$

[0067] 其中 m' 为测试集的样本个数。

[0068] 步骤 S26, 在一定的范围内改变自编码器个数 L 以及每一个自编码器隐藏层节点个数 n_1 的数值, 重复步骤 S22 ~ S25; 选取最小误差对应的网络结构和参数作为堆叠自编码器深度网络的最优网络结构和参数。

[0069] 堆叠自编码器可能的结构数随着 L 的增大呈指数级增长, 为了避免“组合爆炸”问题, 当 L 很大时, 不必遍历每一种可能的结构, 而是随机产生一定数目 (例如 1000 个) 的结构, 在这些结构的深度网络上进行测试验证即可。

[0070] 步骤 S3: 调用训练好的最优堆叠自编码器深度网络预测车辆短程行驶时间。该步骤具体包括:

[0071] 步骤 S31, 当新的车辆到达时, 采集需要的输入数据并以同样的参数做归一化处理。

[0072] 数据采集和归一化方法如前所述, 归一化参数由步骤 S1 得到。

[0073] 步骤 S32, 将归一化后的输入数据带入训练好的深度网络进行前向计算, 得到对应的归一化预测值。

[0074] 前向计算只需在步骤 S23 所述前向传播过程的基础上稍作扩展即可, 并将最后一个自编码器的隐藏层输出带入逻辑斯蒂回归模型, 即可求得输入样本的归一化预测值。

[0075] 步骤 S33, 将步骤 S32 得到的归一化预测值反归一化, 得到车辆在预测起止点之间的预测行驶时间。

[0076] 反归一化过程如步骤 S25 所述。

[0077] 以上所述, 仅为本发明中的具体实施方式, 但本发明的保护范围并不局限于此, 任

何熟悉该技术的人在本发明所揭露的技术范围内,可理解想到的变换或替换,都应涵盖在本发明的包含范围之内。因此,本发明的保护范围应该以权利要求书的保护范围为准。

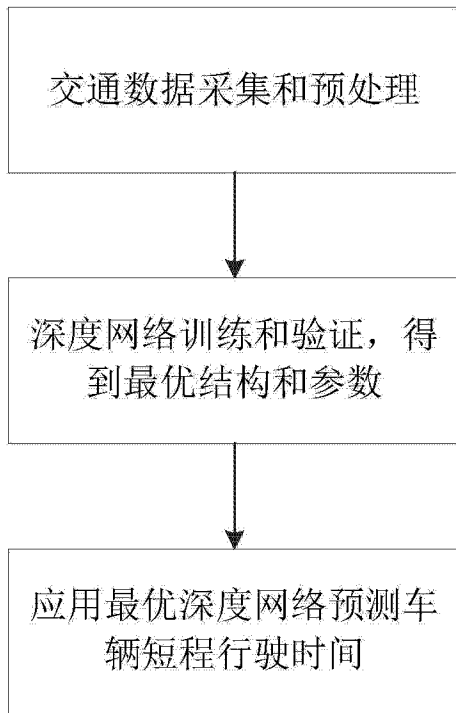


图 1

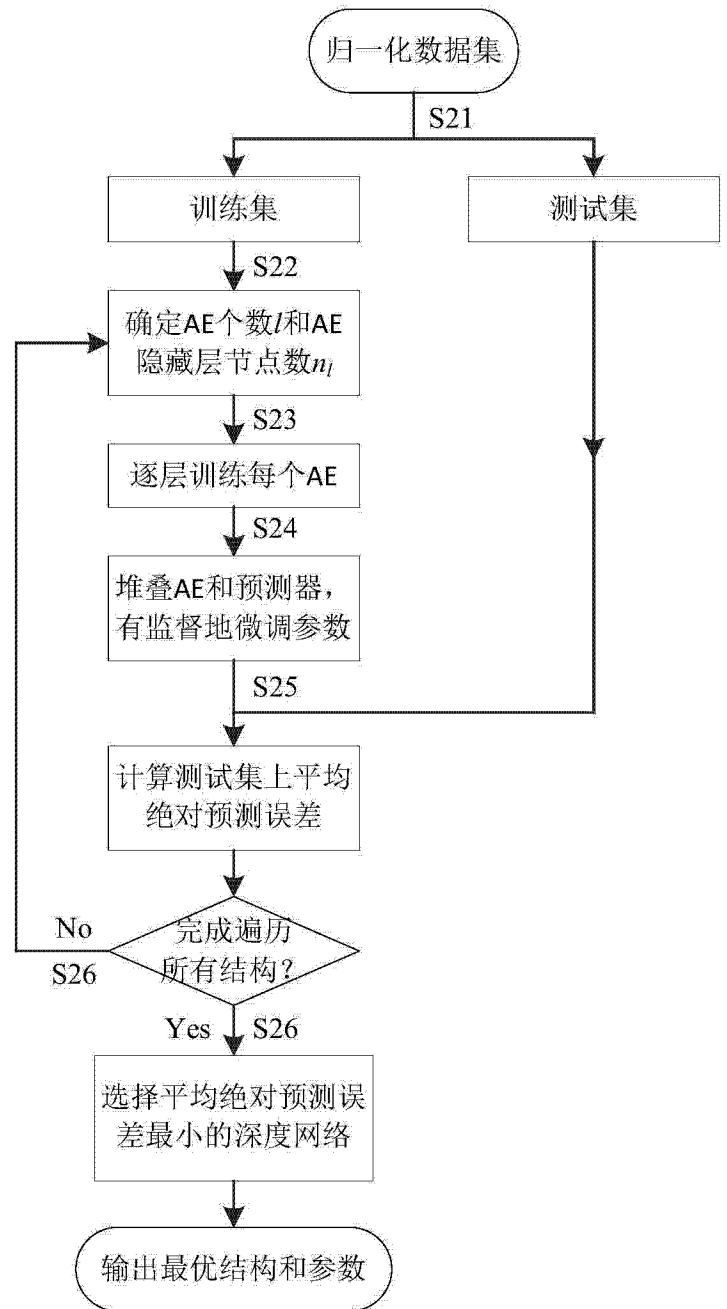


图 2

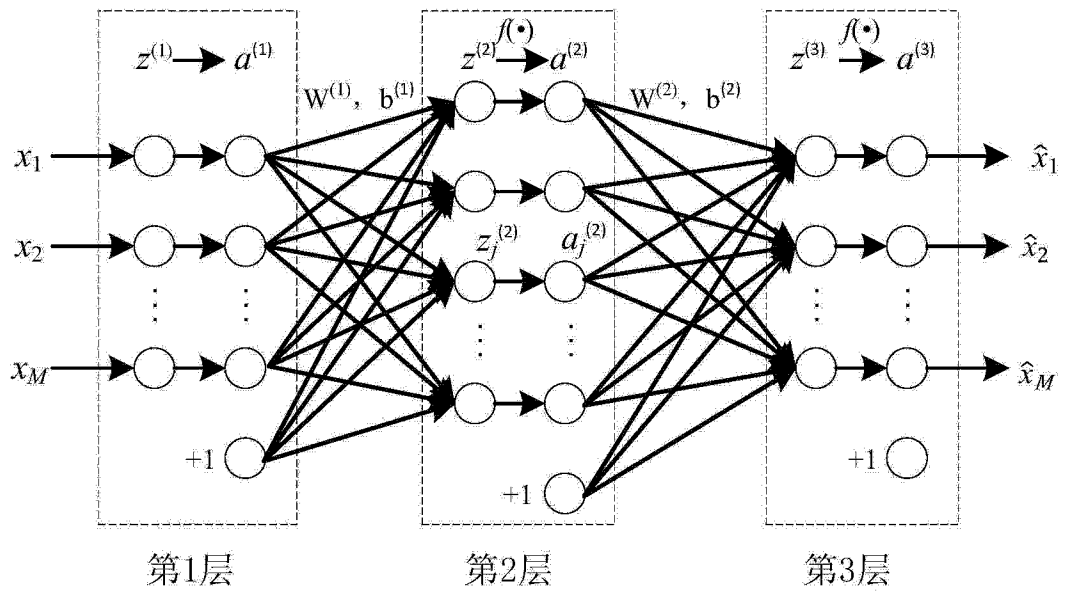


图 3

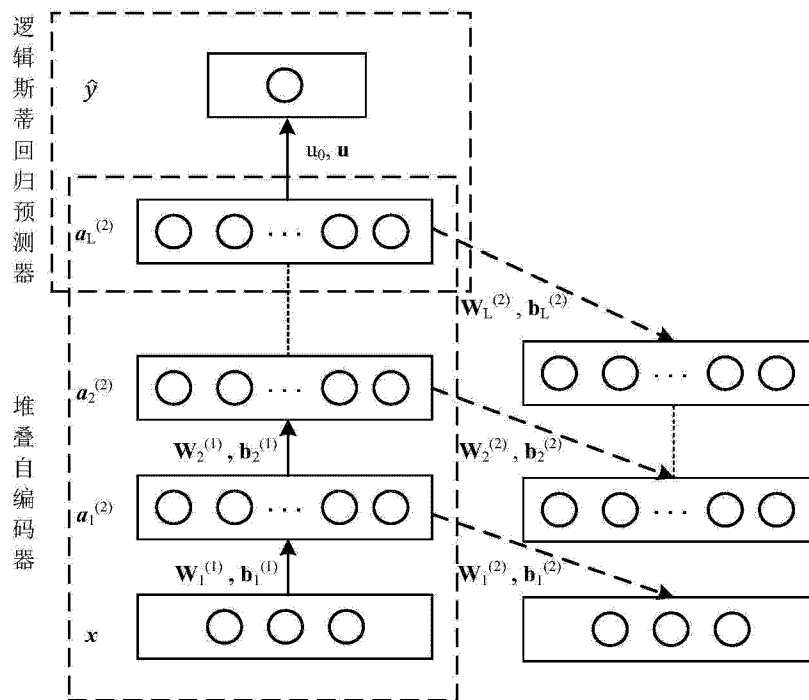


图 4