心理学报 2026, Vol. 58, No. 3, 558−568 https://doi.org/10.3724/SP.J.1041.2026.0558
Acta Psychologica Sinica https://cstr.cn/32110.14.2026.0558

基于长短期记忆网络的探索性因子分析因子保留方法*

郭 磊[1,2] 秦海江[1]

([1]西南大学心理学部; [2]中国基础教育质量监测协同创新中心西南大学分中心, 重庆 400715)

摘 要 心理学研究中, 确定心理特质的维度及其特征极为重要。探索性因子分析(EFA)是识别潜在维度的一种重要统计方法。准确识别因子数量是 EFA 的关键技术之一, 低估或者高估因子数量都会带来不良后果。为准确识别因子数量, 本研究将特征根视作序列数据, 采用长短期记忆(LSTM)网络构建的深度神经网络的各项评估指标(准确率、精确率、召回率、*F*1、*Kappa*)均在 83%以上。通过大规模的模拟实验及实证研究, 验证了 LSTM 在不同数据条件中的性能。结果表明: LSTM 比 CDF、EKC 和 PA 方法具有更高的准确率, 平均提升率为 48.50%, 最大提升率高达 171.09%。而且, LSTM 比 CDF、EKC 和 PA 方法具有更小偏差, 表现出更好稳健性。研究者可使用 R 包 *LSTMfactors* 调用本研究所训练的 LSTM 分析实证数据。

关键词 探索性因子分析, 长短期记忆, 因子保留, 深度学习

分类号 B841

1 引言

心理学研究关注个体的潜在变量(latent variables), 如智力、人格等, 对其进行描述和解释。然而, 心理品质具有抽象性和不可观测性, 使研究者只有通过可观察的外显指标去推测。探索性因子分析(Exploratory factor analysis, EFA)是心理量表开发中最常用的技术之一, 用于缺少特定理论模型强有力的先验信息时, 对一个或多个潜在变量与一组外显指标之间的关联进行建模。

EFA 中的一个关键问题是如何确定正确的因子数量(Zwick & Velicer, 1986)。低估因子数量会忽略掉某个因子及与其他因子之间的关联, 从而遗漏了理论上较为重要的心理结构或子维度, 造成重要信息的丢失, 也会增大所有因子载荷的估计误差(Wood et al., 1996), 还会增大因子分数的估计误差(Fava & Velicer, 1996)。高估因子数量会导致因子分裂(factor splitting), 例如在因子旋转之后, 外显变量在某个因子上的主要载荷将会分裂在多个因子上, 从而减弱外显变量与该因子的关联(Wood et al., 1996)。此外, 还可能得到一个不够简洁的模型, 其中包括几乎没有解释价值的结构(de Winter & Dodou, 2012)。因此, 明确各潜在变量的含义以及准确保留因子数量是可靠的心理评估和有效的测量工具在临床中使用的重要先决条件。

为了尽可能准确的保留因子, 研究者们提出了许多方法。Goretzko (2025)综述了目前因子保留的方法, 大致分为以下几类: (1)简单或图形化方法: 以简单规则来保留因子, 或者注重不同因子数量的图形变化, 例如特征根大于 1 原则(凯撒准则; Kaiser, 1960)、碎石图检验、Hull 方法(Lorenzo-Seva et al., 2011)等。(2)序列检验方法: 通过卡方检验或相对拟合指标(如 AIC、BIC 等)从多个可能的因子数量中挑选出最优方案。(3)公式化方法: 通过特定化的公式计算出特征根的参考值或其他特定指标, 然后根据计算出的参考值或特定指标来确定因子

收稿日期: 2025-04-02

* 中央高校基本科研业务费专项资金(SWU2109222; SWU-XJLJ202307); 西南大学 2035 先导计划项目(SWUPilotPlan006)。

通信作者: 郭磊, E-mail: happygl1229@swu.edu.cn

数量(还可以同时增加凯撒准则作为额外约束)。例如经验凯撒准则(empirical Kaiser criterion, EKC; Braeken & van Assen, 2017), 最小平均偏相关检验(Minimum Average Partial Test; Velicer, 1976)等。(4)模拟方法: 通过模拟随机或具有特定因子数量的数据集, 将之与实际数据集进行比较后保留因子, 例如比较数据(Comparison Data, CD; Ruscio & Roche, 2012)方法、平行分析(Parallel Analysis, PA; Horn, 1965)等。(5)机器学习方法: 利用训练好的机器学习模型保留因子, 例如因子森林(Factor Forest, FF; Goretzko & Bühner, 2020, 2022)方法, 比较数据森林(Comparison Data Forest, CDF; Goretzko & Ruscio, 2024)方法等。(6)正则化方法: 通过正则化将一些微小(不必要)因子调整为零, 从而实现因子保留。例如正则化的 EFA (即在似然函数中增加对载荷矩阵与因子间相关矩阵的惩罚)、探索图分析(对图形的边, 即两个观测变量的偏相关进行惩罚)等。

上述多数方法都基于特征根开展, 例如 PA、EKC、CDF 等。实际上, 特征根可以被视作序列数据, 从而以一个新的视角实现因子保留。因为利用 EFA 进行维度探索, 起始于对基于作答数据的相关系数矩阵 $\boldsymbol{\Sigma}$ 的特征分解, 从而得到特征根向量 $\boldsymbol{\lambda}$ 与特征向量矩阵 $\boldsymbol{\Lambda}_{J\times J}$, 三者存在如下关系:

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\lambda}\boldsymbol{\Lambda}^{T} \tag{1}$$

$$\sum_{j=1}^{J}\lambda_j = \mathrm{tr}(\boldsymbol{\Sigma}) = J \tag{2}$$

其中, $\mathrm{tr}(\boldsymbol{\Sigma})$ 为 $\boldsymbol{\Sigma}$ 的迹, 由于相关系数矩阵(即标准化的协方差矩阵)对角线元素均为 1, 因此迹等于观测变量个数 J, 总特征根之和也固定等于 J。通常将 $\boldsymbol{\lambda}$ 中的元素从大到小排列, 即有: $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_j \geqslant \cdots \geqslant \lambda_J$。显然, 若排序在前的特征根取值大, 则排序在后的特征根取值必然变小, 这充分反映出了特征根之间的相互依赖性, 赋予了特征根序列属性(Braeken & van Assen, 2017; Li et al., 2020)。

鉴于人工智能技术在 EFA 因子保留上展现出的准确性优势(如 FF 与 CDF), 且长短期记忆(Long Short-Term Memory, LSTM)网络是处理序列数据效果较佳(参见本文 3.1 部分的简介以及网络版附录 4 中的详细介绍)的经典人工智能技术, 为了验证以序列视角实现因子保留的可能性, 本研究拟: ①将用于 EFA 所抽取的特征根视作时序数据, 同时引入 LSTM 处理序列特征, 以此学习特征间的相互依赖关系并用于因子保留; ②对 LSTM 进行超参数调优, 以寻找最佳超参数组合; ③将 LSTM 与传统方法进行全面比较以验证其性能。

在传统方法中, PA 在大量模拟研究中均有较好表现, 如在不同的数据分析上具有较强的稳健性, 在各种数据条件下(如样本量在 30 至 360 之间, 观测指标在 9 至 72 之间)具有良好性能(Dinno, 2009; Humphreys & Montanelli, 1975; Peres-Neto et al., 2005; Zwick & Velicer, 1986), 因此被看作为金标准。同时, EKC 方法注重特征根的序列变化(即某一参考特征根的取值会受排序在其之前特征根大小的影响), 这与本研究的序列视角相似。此外, CDF 是结合了 FF 与 CD 的最新方法, 与 LSTM 同属于机器学习方法, 且在因子数量较多时表现优于 EKC。因此, 本研究将 PA、EKC 和 CDF 这三种现有方法纳入比较。

本研究按如下结构进行组织: 第二部分简要介绍本文用于比较的传统方法: PA、EKC 和 CDF。第三部分介绍 LSTM, 及其用于 EFA 构建的全过程, 包括: 生成训练数据集, 训练特征的抽取, 模型训练及验证, 超参数调整等内容。第四部分是在现有 EFA 领域中的典型数据条件中, 对 LSTM 进行模拟实验, 并和传统方法比较, 以验证 LSTM 的优势。第五部分是实证研究, 用于演示现实测验情景中 LSTM 的具体使用。最后是本研究的讨论和结论。

2　PA、EKC 与 CDF 简介

Horn (1965)提出的 PA 方法, 时至今日被部分研究者看作是“金标准” (Auerswald & Moshagen, 2019; Goretzko, 2025)。PA 方法的原理是通过比较多次(例如 100 次; Auerswald & Moshagen, 2019)随机数据的特征根与实证数据的特征根之间的大小来决定是否保留某个因子。大于参考特征根的特征根数量, 即为 PA 保留因子的数量。PA 的具体步骤请参见网络版附录 1。Auerswald 和 Moshagen (2019)的研究指出, 使用主成分分析计算特征根并采用 95%的百分位数作为参考特征根可以提供最高的估计准确性。因此, 本文也将采用该设置。

经验凯撒准则(Braeken & van Assen, 2017)是对传统凯撒准则的改进。与凯撒准则一样, EKC 也使用临界值 1 的准则来保留因子。然而, EKC 还考虑了来自随机样本中各特征根的变异性。在零模型假设下, 基于随机数据计算所得相关系数矩阵的特征根分布应当渐近服从 Marčenko-Pastur 分布(Marčenko & Pastur, 1967)。因此, Braeken 等(2017)通过计算服从该分布的参考特征根(详细过程参见

网络版附录 2), 并与实证数据的特征根进行比较, 那些大于 1 且大于参考特征根的特征根数量, 即为 EKC 保留因子的数量。

Goretzko 和 Ruscio (2024)提出的 CDF 是结合了 CD 和因子森林的最新方法。CDF 与 PA 类似, 也需要模拟数据集, 但模拟方式与 PA 略有区别。PA 的模拟数据是随机数据, 而 CDF 是基于实证数据计算得到的相关系数矩阵 $\boldsymbol{\Sigma}$ 进行的模拟, 该方法以迭代方式获得具有特定因子数量, 但在数值上近似 $\boldsymbol{\Sigma}$ 的相关系数矩阵, 再以该模拟的相关系数矩阵生成模拟作答数据, 具体过程请参见网络版附录 3 或 Ruscio 和 Kaczetow (2008)的研究。此外, PA 通过参考特征根与实证特征根之间的大小关系来决定因子数量, 而 CDF 则是通过比较不同因子数量的模拟数据与实证数据的匹配程度来决定因子数量。

3 LSTM 构建与训练

3.1 LSTM 简介

LSTM 是一种特殊的循环神经网络(Recurrent Neural Network), 专门设计用于解决传统循环神经网络在处理序列数据时遇到的梯度消失和梯度爆炸问题。LSTM 通过引入独特的门控机制, 能够有效捕捉序列数据中的依赖关系。一个完整的 LSTM 单元通常由一个记忆元(Memory Cell)和三个控制门组成: 输入门(Input Gate)、遗忘门(Forget Gate)和输出门(Output Gate) (Hochreiter & Schmidhuber, 1997)。记忆元承担了信息的长期存储功能, 而各个门结构通过引入可学习的权重参数, 控制信息的选择性更新, 从而有效缓解了传统循环神经网络在长期依赖学习中的梯度消失与爆炸问题。具体而言, 遗忘门决定当前时刻应当丢弃记忆元中哪些信息; 输入门控制当前输入信息对记忆元状态的更新程度; 输出门决定从记忆元中提取哪些信息作为当前时刻的输出。这种门控机制使得 LSTM 能够在序列建模任务中捕捉长期依赖结构, 广泛应用于语言建模、时间序列预测、序列标注等领域。

通常, 为了进一步提高模型性能, LSTM 会后接一个或多个全连接层(Fully Connected Layer, 也称为密集层)。全连接层的作用是将 LSTM 提取的高维特征向量映射到所需的输出维度(Hochreiter & Schmidhuber, 1997)。该映射有助于减少模型复杂度, 并且使模型能够更好地适应不同类型的任务需求(Goodfellow, et al., 2016; Brownlee, 2018)。此外, 批量标准化层(Batch Normalization Layer; Ioffe & Szegedy, 2015; Lange et al., 2022)能够在训练过程中对特征进行小批量标准化从而使神经网络更快稳定, 还能在一定程度上避免过拟合, 因此可以考虑在全连接层后接入批量标准化层。这种结构组合不仅能够充分利用 LSTM 对序列数据的强大建模能力, 还能通过全连接层实现高效的特征映射和最终预测。对本研究所使用的神经网络原理、数学表达形式及具体网络结构感兴趣的读者可参见网络版附录 4。

3.2 LSTM 模型构建

本研究构建机器学习模型的步骤与 Goretzko 和 Bühner (2020)文中图 1 步骤一致, 详细内容请参见该文。根据 Chen 等(2017)研究指出, 用于构建机器学习模型的数据集越大模型表现越好。同时参考 Goretzko 和 Bühner (2020)的研究, 本研究将生成 1000000 样本数据集用于训练 LSTM。每一份数据集的生成方式参考 Auerswald 和 Moshagen (2019)以及 Goretzko 和 Bühner (2020)的研究: 因子数量从均匀分布 $U(1,10)$ 中抽取, 每个因子下的题目数量从均匀分布 $U(3,10)$ 中抽取, 主要载荷从均匀分布 $U(0.35, 0.80)$ 中抽取, 交叉载荷从均匀分布 $U(-0.20, 0.20)$ 中抽取, 因子间相关从均匀分布 $U(0.00, 0.50)$ 中抽取, 样本量从均匀分布 $U(100,1000)$ 中抽取。

具体而言, 为了生成每个数据集, 需要首先计算相关系数矩阵:

$$\boldsymbol{\Sigma} = \boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^T + \boldsymbol{\Psi}^2 \tag{3}$$

公式(3)中, $\boldsymbol{\Lambda}$ 表示因子载荷矩阵(同时包含主要载荷和交叉载荷), $\boldsymbol{\Phi}$ 表示各因子间的相关系数矩阵, $\boldsymbol{\Psi}^2$ 是一个对角矩阵, 其元素为 $1-diag(\boldsymbol{\Lambda}\boldsymbol{\Phi}\boldsymbol{\Lambda}^T)$, 其作用是确保 $\boldsymbol{\Sigma}$ 的对角元素等于 1, $\boldsymbol{\Sigma}$ 便是根据上述数据条件而产生的相关系数矩阵。为了更贴合现实测验情境, 本研究采用 Auerswald 和 Moshagen (2019)以及 Goretzko 和 Bühner (2020)使用的多元正态分布来模拟作答数据, 即:

$$X_j = L_j + \varepsilon_j, 1 \leqslant j \leqslant J \tag{4}$$

其中, L_j 服从多元正态分布 $N(0, \boldsymbol{\Sigma})$, 用于表示潜在因子的贡献。$\varepsilon_j$ 是服从标准正态分布的残差项。值得注意的是, 根据 Auerswald 和 Moshagen (2019)以及 Goretzko 和 Bühner (2020)所提出的方法, 为使模拟生成的作答数据 $\boldsymbol{X}$ 的相关系数矩阵与 $\boldsymbol{\Sigma}$ 尽可能接近, 需要限制 L_j 与 ε_j 不相关, ε_j 与 ε_j 不相关。

3.3　训练特征提取

本研究将围绕序列数据 $\boldsymbol{\lambda}$ 来训练 LSTM, 以更准确地估计因子数量。根据 $\boldsymbol{\lambda}$ 归纳出了两类训练特征用于模型训练: 第一类为传统的基于主成分分析得到的特征根 λ, 第二类为基于主成分分析得到的特征根 λ 与 PA 得到的参考特征根 λ^{ref} 之差, 即 $\lambda_j - \lambda_j^{ref}$。第一类训练特征容易理解, 沿用了目前大部分方法的参考依据。第二类训练特征受到金标准 PA 的启发, 考虑实证数据方差与随机数据方差的差异。然而, PA 法简单地以 $I(\lambda_j - \lambda_j^{ref} > 0)$ 来判断是否应该保留第 j 个因子, 很可能受随机误差的影响。因此, 本研究避免以 0 为固定的判断标准, 而是通过训练深度神经网络的方式使 LSTM 实现动态判断。

总之, 本研究将使用两类(即 λ_j 和 $\lambda_j - \lambda_j^{ref}$)序列数据来训练最佳的 LSTM, 以实现对因子数量的准确估计。

3.4　模型训练与评估

LSTM 训练采用 R (4.5.0; R Core Team, 2025) + Python (3.13.3; https://www.python.org/downloads/release/python-3133/)的混合编程实现。具体而言, 利用 R 包 *reticulate* (Kalinowski, et al., 2025)和 Python 库 *pyper* (https://github.com/pyper-dev/pyper)实现 R 与 Python 的相互调用, 以 Python 库 *PyTorch* (Paszke et al., 2019; https://pytorch.org/)搭建 LSTM 并训练, 以 Cuda 12.06 实现 GPU 加速神经网络训练, 将最终训练好的 LSTM 转化为不依赖于 *PyTorch* 的".onnx"文件, 从而实现长期存储及跨平台运行(借助 *onnxruntime*; https://onnxruntime.ai/), 以 Python 库 *scikit-learn* (Pedregosa et al., 2011; https://scikit-learn.org/stable/)实现对 LSTM 评估指标的计算, 上述所有代码已公开在 https://osf.io/au9vd/网站。训练设备包括 Nvidia RTX 4060 8GB 显存 GPU, 64 GB 内存和 Intel i7-14700KF CPU.

基于所生成的 1000000 个样本数据集, 根据 Goretzko 和 Bühner (2020)构建机器学习模型的步骤, 抽取出每个数据集的特征用于 LSTM 训练。为保证所构建 LSTM 的有效性以及避免过拟合问题, 需要对其进行超参数调优, 其超参数设置如下: ①LSTM 的长短期记忆层为 1~2 层 (Hochreiter & Schmidhuber, 1997; LeCun et al., 2015), 每层节点数为 1~40 个(Heaton, 2008)。②全连接层的层数为 1~5 层(LeCun et al., 2015), 每层节点数为 1~40 个(Heaton, 2008)。全连接层后接入批量标准化层能够增加神经网络的鲁棒性、加速训练, 由于批量标准化层的神经节点数量与全连接层需要一致, 因此只需调整全连接层参数即可。③激活函数包括: ReLu, Tanh 和 Sigmod (Nair & Hinton, 2010; LeCun et al., 2015)。④学习率为: 0.1, 0.01, 0.001, 0.0001 和 0.00001 (Kingma & Ba, 2014)。

对于有多个隐藏层(全连接层和 LSTM 层都属于隐藏层)的深度神经网络而言, 由于参数数量过大且许多参数对网络性能的影响很小(Bergstra & Bengio, 2012), 导致网格搜索(Grid Search, 即穷举)的效率非常低, 而随机搜索是通过在一定迭代次数内(例如 1000 次)随机组合不同的超参数来筛选最优的超参数组合, 能够尽可能快速而有效地完成深度神经网络的训练(Bergstra & Bengio, 2012)。因此, 本研究使用 1000 次迭代的随机搜索进行超参数调整。在训练模型过程中, 采用 7 : 3 比例随机划分训练集与测试集(Qin & Guo, 2024)。

对于用于训练 LSTM 的特征根序列长度而言, 序列越长, 对训练算力要求越高。根据被广泛验证的传统因子保留方法, 如 PA、凯撒准则、EKC 等, 前 F 个特征根是决定是否保留 F 个因子的决策依据, 而后续特征根作用甚微。因此, 本研究合理地假设, 若讨论的因子数量范围在 1 至 10 之间, 特征序列长度只需等于 10 即可较好地训练 LSTM。若短于 10 则可能由于决策依据不足而导致训练精度下降, 若长于 10 则可能由于训练数据冗余而出现训练耗时大幅增加但精度并不会提升的情况。为了证明该假设的合理性, 我们以 5、10 和 20 三种序列长度分别训练了 LSTM。三种序列长度各自的超参数随机搜索(均为 1000 次)与分类准确率结果见图 1 (详细结果可从 https://osf.io/au9vd/的"results/study I/acc.xlsx"获取)。图 2 呈现了不同序列长度 LSTM 的训练耗时以及最佳 LSTM 的准确率结果。

很明显, 图 1 与图 2 的训练结果支持了本研究的理论假设。当序列长度超过 10 后训练精度不再明显提升: 长度由 10 增长至 20 时, LSTM 准确率仅提升了 0.97%, 但训练耗时却增加了 80522 秒(约 22.37 小时)。显然, 序列过长没有带来更大的精度提升, 徒增了训练成本。因此, 为了兼顾精度与实际应用, 本研究接下来都将使用序列长度为 10 的 LSTM, 并对最佳 LSTM 进行模型评估。

模型评估选取常用的评价指标: 准确率 accuracy、精确率 precision、召回率 recall、$F1$ 分数和 *Kappa* 系数(评价指标的计算参见网络版附录 5)。

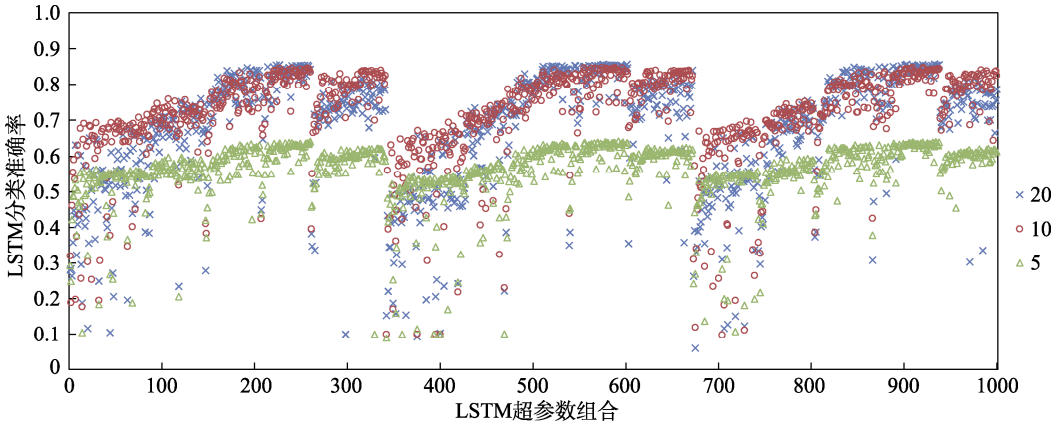

图 1 不同序列长度的 LSTM 超参数随机搜索结果。相同横坐标表示超参数组合相同

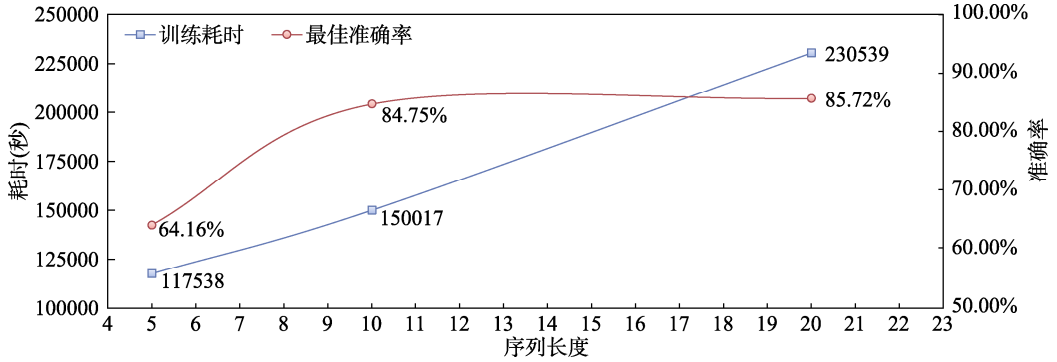

图 2 不同序列长度 LSTM 的训练耗时及最佳准确率

这 5 个指标的值越高，表明模型训练越成功。由表 1 可知, LSTM 的最优超参数组合为: 学习率为 0.01, 激活函数为 Tanh 等。此时，模型准确率达 0.847，其余指标也均高于 0.831。

表 1 超参数调优结果

最优的超参数组合		最优模型评价指标	
超参数	值	指标	值
学习率	0.01	Accuracy	0.847
LSTM 层数	2	Precision	0.847
LSTM 每层节点数	32, 38	Recall	0.847
全连接网络层数	5	*F1*-Score	0.847
全连接层每层节点数	38, 39, 34, 31, 27	*Kappa*	0.831
激活函数	Tanh		

4 模拟研究

4.1 研究设计

模拟研究将在不同实验条件下比较 LSTM 和三种方法的性能，PA、EKC 与 CDF 均可利用 R 包 *EFAfactors* 实现(Qin & Guo, 2025a), LSTM 可使用已训练完成的最佳 LSTM 实现(请见上文“3.4 模型训练与评估”部分), R 包 *LSTMfactors* (Qin & Guo, 2025b)提供了调用该模型的函数 LSTM()。实验共包括 7 个自变量，分别是: 因子数量: 1, 2, 4, 6, 8, 10; 每个因子下的题目数量: 4, 7, 10; 主要载荷: 高载荷 $U(0.65, 0.80)$，中等载荷 $U(0.50, 0.65)$，低载荷 $U(0.35, 0.50)$；交叉载荷: 高载荷 $U(-0.20, -0.10) \cup U(0.10, 0.20)$，低载荷 $U(-0.10, 0.10)$；因子间相关: 0.0, 0.25, 0.50, 0.75; 样本量: 100, 200, 500, 1000; 因子保留方法: LSTM, PA, EKC, CDF。每个实验条件重复 500 次，共需 $6\times3\times3\times2\times4\times4\times4\times500 = 3456000$ 次实验。

4.2 评价指标

采用因子数量识别的准确率(accuracy)和偏差(bias)作为评价指标，计算方式如下:

$$acc = \frac{\sum_{l=1}^{L} I(\hat{F}_l = F_l)}{L} \tag{5}$$

$$bias = \frac{\sum_{l=1}^{L} (\hat{F}_l - F_l)}{L} \tag{6}$$

其中，L 表示特定数据条件下的总实验次数，F_l 为第 l 次实验中的因子数量真值，$\hat{F}_l$ 为第 l 次实验中估计的因子数量，$I(\cdot)$ 为指示函数。acc 取值范

围为 0~1, 值越高越好。bias 取值越接近 0 越好。bias 大于 0 表示高估因子数量, 小于 0 表示低估因子数量。

4.3 结果

表 2 呈现了在不同数据条件下 4 种因子保留方法的准确率结果。

表 2　各数据条件下不同方法的因子数量保留准确率

数据条件	水平	准确率 acc				
		CDF	EKC	PA	LSTM	提升率
因子数量	1	0.677	0.922	0.944	**0.969**	2.58%
	2	0.722	0.715	0.677	**0.879**	21.69%
	4	0.558	0.516	0.493	**0.732**	31.08%
	6	0.450	0.419	0.394	**0.523**	16.17%
	8	0.373	0.358	0.330	**0.636**	70.62%
	10	0.313	0.319	0.280	**0.865**	171.09%
因子间相关	0.00	0.686	0.724	0.761	**0.901**	18.29%
	0.25	0.619	0.650	0.662	**0.867**	31.00%
	0.50	0.485	0.495	0.450	**0.775**	56.44%
	0.75	0.273	0.298	0.206	**0.527**	76.96%
每因子题目数量	4	0.370	0.416	0.381	**0.772**	85.47%
	7	0.547	0.568	0.547	**0.765**	34.51%
	10	0.630	0.640	0.631	**0.765**	19.54%
主要载荷	*L*	0.332	0.347	0.376	**0.687**	82.60%
	M	0.525	0.564	0.540	**0.784**	39.14%
	H	0.690	0.714	0.643	**0.831**	16.25%
交叉载荷	*L*	0.486	0.519	0.508	**0.750**	44.59%
	H	0.545	0.565	0.532	**0.785**	38.99%
样本量	100	0.286	0.299	0.309	**0.637**	106.58%
	200	0.429	0.461	0.454	**0.755**	63.70%
	500	0.617	0.657	0.614	**0.823**	25.26%
	1000	0.731	0.750	0.703	**0.854**	13.94%

注: 加粗数据为同等实验条件下的最佳结果。提升率计算方式为 $\frac{acc_{LSTM}-\max(acc_{CDF},acc_{EKC},acc_{PA})}{\max(acc_{CDF},acc_{EKC},acc_{PA})}$, 表示 LSTM 在准确率上相对于 CDF、EKC 和 PA 中最高准确率的提升程度。*L* 表示低载荷, *M* 表示中等载荷, *H* 表示高载荷。

由表 2 结果可知, LSTM 的准确率在所有数据条件下均显著优于传统方法, 其优势在因子数量较多、因子间相关性较高、题目数量较少或样本量较小时尤为突出, 而这些条件均是现实测验中偏于“极端”的情景。

具体而言: 随着因子数量的增加, 传统方法的准确率显著下降。如当因子数为 10 时, CDF、EKC 和 PA 的准确率仅为 0.313、0.319 和 0.280, 而 LSTM 仍高达 0.865, 提升幅度达到 171.09%。这表明传统方法在复杂因子结构下的识别能力有限, 而 LSTM 能够更有效地学习并应用于高维潜结构。当因子间相关性逐渐增强时, 4 种方法的准确率均呈下降趋势, 但 LSTM 的表现依然最佳。如在最高相关性下, LSTM 的准确率为 0.527, 仍显著高于 CDF 的 0.273、EKC 的 0.298 与 PA 的 0.206, 比 EKC 的精度提高了 76.96%。当每因子题目数量较少时, 传统方法难以准确估计因子数量, 如题目数为 4 时, 三种传统方法准确率均低于 0.42, 而 LSTM 仍维持在 0.772, 提升达 85.47%。该结果突显了 LSTM 对简式量表情境的强大适应能力。当主要载荷和交叉载荷水平变动时, LSTM 准确率仍优于传统方法, 尤其在低主要载荷和高交叉载荷条件下, 准确率分别达到 0.687 和 0.785, 表明其在处理弱结构或高噪声数据时的稳健性。最后, 4 种方法均呈现样本量越大准确率越高的趋势。然而, LSTM 即使在最小样本量时也可达到 0.637 的准确率, 远高于 CDF 的 0.286、EKC 的 0.299 和 PA 的 0.309, 比 PA 的精度提高了 106.58%。这进一步显示出 LSTM 在小样本测验情景中的优势。

表 3 呈现了不同数据条件下 4 种因子保留方法的偏差均值结果。

在偏差方面, LSTM 亦表现出显著优势: 整体上其估计偏差最小, 在高估或低估因子数量方面更为平衡。在因子数量为 2 或 4 时, 传统方法均倾向低估(bias 为负), 如 PA 在因子数为 4 时 bias 为 −1.131, 而 LSTM 偏差为 0.524, 虽然略有高估, 但偏差值远小于 PA, 反映其估计更接近真值。类似地, 当因子间相关性增强时, 传统方法偏差普遍为负, 且幅度在增大(如 EKC 在相关性为 0.75 时 bias 为 −2.422), 而 LSTM 保持在正向偏差小于 0.3 的范围内。在题目数量和载荷条件上, 传统方法普遍低估因子数, 尤其在低载荷和少题目条件下, CDF 和 PA 的偏差均超过−2。而 LSTM 即使在这些恶劣条件下, 偏差值也能维持在正负 0.5 之间, 表明其在“极端测验条件”中具有较强稳健性。此外, 所有方法的偏差随样本量增加而减小, 但 LSTM 即使在最小样本量下的偏差也仅为 0.270, 相较于 CDF 的−2.609 和 PA 的−2.695, 优势明显, 表明其在小样本情景时仍能稳定估计因子数量。

综上所述, LSTM 在准确率与偏差两个指标上均优于 CDF、EKC 和 PA, 尤其是在因子结构复杂、数据质量较差(高因子间相关、高交叉载荷、低主要载荷、小样本、每因子下题目数少)等极端条件

下, 表现出更强的稳定性。LSTM 作为基于深度学习的新方法, 更能从复杂数据中提取潜在结构。

表 3 各数据条件下不同方法的因子数量保留偏差均值

数据条件	水平	偏差 bias			
		CDF	EKC	PA	LSTM
因子数量	1	0.363	−0.063	−0.038	**0.031**
	2	−0.053	−0.290	−0.326	**0.010**
	4	−0.892	−0.936	−1.131	**0.524**
	6	−1.826	−1.689	−2.061	**0.669**
	8	−2.851	−2.489	−3.055	**0.164**
	10	−3.919	−3.309	−4.095	**−0.406**
因子间相关	0.00	−0.616	−0.824	−0.476	**0.088**
	0.25	−0.893	−1.049	−0.858	**0.117**
	0.50	−1.596	−1.555	−2.005	**0.242**
	0.75	−3.014	−2.422	−3.798	**0.215**
每因子题目数量	4	−1.853	−2.133	−2.371	**−0.269**
	7	−1.476	−1.334	−1.671	**0.208**
	10	−1.261	−0.921	−1.311	**0.558**
主要载荷	*L*	−2.466	−2.462	−2.279	**0.464**
	M	−1.447	−1.365	−1.727	**0.151**
	H	−0.676	−0.561	−1.347	**−0.118**
交叉载荷	*L*	−1.838	−1.889	−1.963	**0.176**
	H	−1.222	−1.037	−1.605	**0.155**
样本量	100	−2.609	−2.722	−2.695	**0.270**
	200	−1.856	−1.899	−2.022	**0.129**
	500	−1.036	−0.908	−1.383	**0.150**
	1000	−0.618	−0.321	−1.037	**0.113**

注: 加粗数据为同等实验条件下的最佳结果。

5 实证研究

本研究拟使用一批实证数据作为教程, 充分演示如何使用 LSTM 进行因子保留, 并与 PA、EKC 和 CDF 方法进行比较。该实证数据来自 2022 年施测于某市普通高中群体的父母心理控制量表所得数据, 共包括 987 名高中生, 男生 406 人(占比 41.1%), 女生 581 人, 总体平均年龄为 15.823 岁 (*SD* 为 0.793)。该数据已收录于 R 包 *LSTMfactors* 中。父母心理控制量表最初由 Soenens 和 Vansteenkiste (2010)编制, 包括 4 个维度: 独立−父母消极反应、依赖−父母积极反应、低成就−父母消极反应、高成就−父母积极反应, 共计 20 题。反应方式为 5 点利克特式, 1 表示非常不赞成, 5 表示非常赞成。后由邓衍鹤等(2019)进行本土化, 并检验了该量表在我国青少年群体中的信效度。本研究中, 总量表的 Cronbach's α 系数为 0.923, 4 个维度的 Cronbach's α 系数在 0.817~0.889 之间, 信度良好。描述性结果如表 4 所示。

通过验证性因子分析(Confirmatory Factor Analysis, CFA)检验了父母心理控制量表在该数据中的结构效度, 以保证因子数量及其结构的有效性, 也为 4 种方法的估计结果提供参考依据。CFA 结果如图 3 所示, 其中 CFI 等拟合指标均大于 0.9, RMSEA 和 SRMR 均低于 0.08, 因子载荷范围在 0.49 至 0.93 之间, 这些结果表明父母心理控制量表结构效度较好。

接下来, 分别使用 LSTM、PA、EKC 和 CDF 对这批实证数据进行因子保留分析, 示例代码见于网络版附录 6, 完整代码请见于 https://osf.io/au9vd/ 的"main/realdata/realdata.R"文件。与传统方法相比, LSTM 保留的因子数量为 4, 与理论以及 CFA 所得结构一致。图 4 进一步提供了 4 种方法的可视化结果。LSTM (图 4A)和 CDF (图 4B)均是执行分类任务的机器学习方法, 可以直接输出因子数量的概率分布结果, PA (图 4C)与 EKC (图 4D)则是通过对比实际特征值与参考特征值之间的差异来决断因子数量。从 CDF 所得因子数量概率分布来看, 其将父母心理控制量表的因子数量估计为 7 (最大概率为 0.447), 且出现了高估因子数量的情况(概率值主要分布在因子数量为 6 至 10 之间)。PA 与 EKC 估计得到的因子数量均为 3, 表现为低估因子数量。相对地, LSTM 的决策要比其余方法更清晰, 其输出的概率值集中在 3 和 4 个因子上, 且 4 个因子的概率值最高, 表明 LSTM 相比 PA 与 CDF 更加集中与准确, 这与模拟研究的表现一致。

表 4 父母心理控制量表的描述性统计结果(*M*±*SD*)与 *α* 系数

	样本量	年龄	独立−消极 *F*1	依赖−积极 *F*2	低成就−消极 *F*3	高成就−积极 *F*4	总量表
男生	406	15.823±0.736	11.421±4.450	14.244±4.557	10.739±4.550	15.778±5.299	52.182±15.043
女生	581	15.823±0.831	10.859±4.206	13.274±4.473	9.644±4.338	15.279±5.552	49.055±14.764
总体	987	15.823±0.793	11.090±4.315	13.673±4.530	10.094±4.457	15.484±5.452	50.341±14.951
α 系数			0.857	0.817	0.885	0.889	0.923

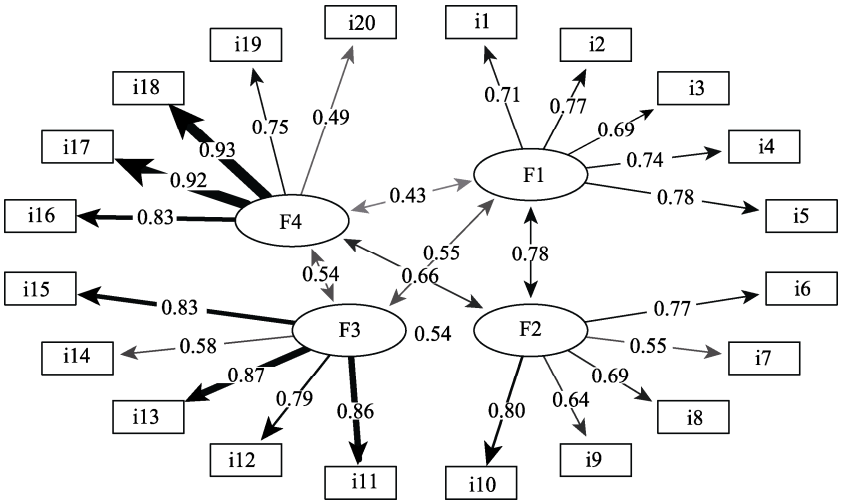

图 3　父母心理控制量表的 CFA 结构

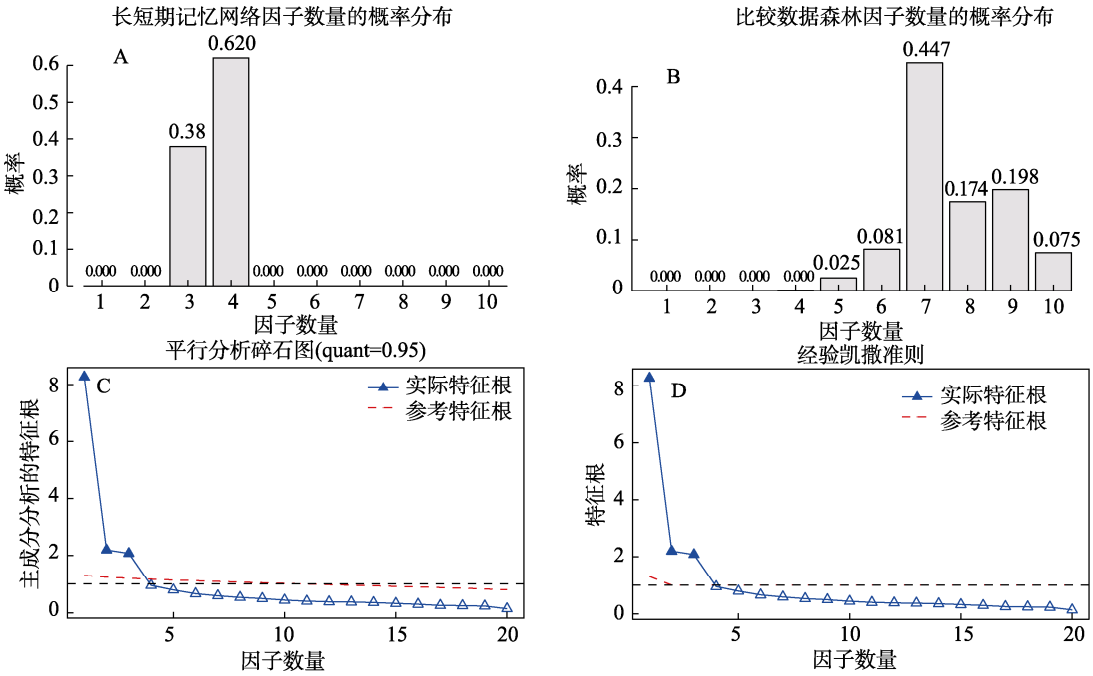

图 4　LSTM、CDF、PA 和 EKC 四种因子保留方法的可视化结果

6　讨论

EFA 可探索数据中的潜在结构，是心理与教育等领域不可或缺的统计技术。正确估计潜在因子数量是正确探索潜在结构的基础，是 EFA 最开始也是最重要的步骤(Zwick & Velicer, 1986)。然而，现有因子保留方法难以在广泛数据条件下保持准确率(Auerswald & Moshagen, 2019; Goretzko & Bühner, 2020)，且在实际应用中，许多过时方法(如碎石图、Kaiser 法)仍被普遍使用，国内对于因子保留方法研究更是稀少。本研究基于 EFA 理论基础，将特征根 λ 看作具有多个时间步的序列数据以训练由 LSTM 所构建而成的深度神经网络，各项评估指标均达 83%以上。通过模拟研究，该网络在广泛数据条件下均取得了相较于传统方法更高的准确率，且具有良好稳健性。在实证研究中，LSTM 也展现出了良好的生态效度。同时，本研究已将训练好的 LSTM 封装为 R 包 *LSTMfactors*，读者可以便捷调用 LSTM，具备良好的实践价值。

在 EFA 中，多种因素都会影响因子数量估计准确率，如因子数量、主要载荷水平与样本量等。具体而言，随着因子数量的增加、主要载荷水平的降低以及样本量的减少，大多数因子保留方法的准确率均呈下降趋势，这与既有研究结果(如

Auerswald & Moshagen, 2019; Goretzko & Bühner, 2020)相一致。原因在于, 因子数量越多, 数据的潜在结构越复杂; 主要载荷越低, 因子对观测变量的贡献越低, 导致因子难以识别; 而样本量不足会因随机误差而模糊潜在结构。此外, 诸如交叉载荷水平、因子间相关程度以及每个因子所包含的题目数量等因素也会对准确率产生影响。较高的交叉载荷可能导致某些变量在多个因子上同时具有较大载荷, 从而加剧潜在结构的复杂性; 因子间相关性过高可能引起因子同质化, 使多个因子被误判为一个; 每个因子下题目数量过少相当于减少了观测指标, 进一步增加了因子识别的困难。但 LSTM 对于这些恶劣条件却展示出了较强的稳健性。

值得注意的是, 相较于传统方法随着因子数量的增多其因子保留准确率持续下降不同的是, LSTM 的准确率表现出类似于 U 型变化的过程(见表 2), 即准确率随着因子数量增加而降低, 但当因子数量持续增加时准确率又表现出回升现象。我们认为, 这可能是因为 LSTM 相较于传统方法更容易估计出较大因子数量而导致。根据 bias 结果(见表 3)可知, 当因子数量持续增加时, 传统方法得到了持续增大的负偏差, 意味着低估越来越严重, 估计出的因子数量严重偏小。而在因子数量为 1 至 6 时, LSTM 的偏差值是持续增大的正值, 当因子数量为 8 和 10 时, bias 下降为负值。这意味着 LSTM 处理因子数量较少的数据时, 相较传统方法容易得到更大的因子数量, 但因子数量持续增加时, LSTM 得到的结果相较于真实因子数量而言差异在变小, 因此出现 bias 降低、准确率提升的现象。Goretzko (2025)认为, 低估因子数量比高估因子数量更加危险, 低估常常导致因子合并、载荷矩阵扭曲等一系列可能导致错误理论的后果, 而高估因子还可以通过后期人工对方差解释、理论框架的调整而排除冗余因子。从这一角度来看, LSTM 正好体现了其优势。

此外, 不同方法背后的理论机制在应对复杂数据结构时展现出了显著差异, 表现在准确率差异上。传统方法如 PA、EKC 和 CDF 多依赖特定的统计计算(如参考特征根的计算方式、随机数据的模拟方式等), 这使它们在因子数量较少、样本量充足、因子间相关较低、高信噪比(主要载荷高、交叉载荷低)等情况下具有一定有效性; 然而在因子数量较多、样本量不足、因子间相关较高、低信噪比等情况下, 这些方法往往无法充分捕捉特征根的复杂依赖关系, 从而导致因子数量的系统性低估。而 LSTM 作为一种深度学习技术, 通过将特征根视作序列数据能够更有效地学习复杂的关联, 尤其适用于样本规模小、载荷分布不均(主要载荷不高、交叉载荷不低)等复杂的测验数据。因此, LSTM 的优势不仅体现在准确率更高上, 更体现在极端测验情景中表现出来的较强稳定性, 从而拓展了因子保留方法在高复杂测验情景下的适用边界。

然而, 本研究也存在一些不足, 未来仍有许多内容值得探索, 以下是针对本研究的建议与展望: ① LSTM 是基于固定数据集训练得到的模型, 其适用范围不可避免的受到训练数据条件约束, 这是基于预训练的人工智能模型进行因子保留的局限(Goretzko, 2025)。但值得注意的是, 尽管模拟研究中已表明 LSTM 在一定范围内具有较强的稳健性, 如训练数据集的因子间相关范围为 $U(0.00, 0.50)$, 而 LSTM 在因子间相关达到 0.75 时(不在训练范围内)仍然获得了比传统方法更高的准确率。但训练数据集以外的数据范围“广阔无边”, 难以被一一验证。出于谨慎, 仅建议将本研究所训练的 LSTM 应用到因子数量在 1~10 之间, 每个因子下的题目数量在 3~10 之间, 主要载荷大于 0.35, 交叉载荷不高于 0.20, 因子间相关不高于 0.50, 样本量在 100~1000 之间的数据, 但这些条件范围已经足以覆盖大部分研究的需求。对于主要载荷、交叉载荷、因子间相关等不可被直接观测到的条件, 我们建议可以在 LSTM 估计因子数量后, 再去估计此时的因子载荷、因子间相关, 最后对照上述参考条件来判断是否适用。②本研究所训练的 LSTM 数据条件主要参考了 Goretzko 和 Bühner (2020)的研究, 并在其基础上进一步扩展了模型覆盖的数据条件范围, 取得了较好表现。这表明依靠人工智能模型实现更准确的因子数量估计方式可行且可靠。然而, 在少数情况下, 实证数据条件仍有可能超出 LSTM 的训练范围, 此时可以使用传统 EFA 因子保留方法(如 PA 法)。此外, 鉴于当下大模型的发展潮流(DeepSeek-AI et al., 2024), 神经网络参数规模越大、训练数据规模越大, 则最终模型性能以及在不同场景下的通用性越好, 未来可以依据本研究的经验来训练“因子保留专用大模型”, 以实现“One Model to Rule Them All” (Goretzko & Bühner, 2020)。③本研究训练数据与模拟研究均使用符合正态分布假设的标准化数据, 当应用到符合正态分布的 Likert 式等级数据和连续数据时, 只需要进行标准化即可无缝接入 LSTM。但对于称名数据、偏态分布数据等缺乏

验证与讨论, 未来可进一步对不同数据类型与分布条件进行探索验证。

参 考 文 献

Auerswald, M., & Moshagen, M. (2019). How to determine the number of factors to retain in exploratory factor analysis: A comparison of extraction methods under realistic conditions. *Psychological Methods*, *24*(4), 468−491.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-arameter optimization. *Journal of Machine Learning Research, 13*, 281−305.

Braeken, J., & van Assen, M. A. (2017). An empirical Kaiser criterion. *Psychological Methods, 22*(3), 450−466.

Brownlee, J. (2018). *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery.

Chen, S., Abhinav, S., Saurabh, S., & Abhinav, G. (2017). Revisiting unreasonable effectiveness of data in deep learning era. *arXiv preprint*:1707.02968.

DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., ... Pan, Z. (2024). DeepSeek-V3 technical report. *arXiv preprint*: 2412.19437.

Deng, Y., Gao, X., Xu, C., Sun, Z., Yue, Y., & Liu, X. (2019). Reliability and validity test of Dependency-Oriented and Achievement-Oriented Psychological Control Scale in Chinese adolescents. *Chinese Journal of Clinical Psychology, 27*(2), 253−257.

[邓衍鹤, 高芯芸, 徐陈晰, 孙治英, 岳艳春, 刘翔平. (2019). 依赖导向与成就导向心理控制量表修订版在我国青少年中的信效度检验. *中国临床心理学杂志, 27*(2), 253−257.]

de Winter, J. C., & Dodou, D. (2012). Factor recovery by principal axis factoring and maximum likelihood factor analysis as a function of factor pattern and sample size. *Journal of Applied Statistics, 39*(4), 695−710.

Dinno, A. (2009). Exploring the sensitivity of Horn's parallel analysis to the distributional form of random data. *Multivariate Behavioral Research, 44*(3), 362−388.

Fava, J. L., & Velicer, W. F. (1996). The effects of underextraction in factor and component analyses. *Educational and Psychological Measurement, 56*(6), 907−929.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Goretzko, D. (2025). How many factors to retain in exploratory factor analysis? A critical overview of factor retention methods. *Psychological Methods*, https://doi.org/0.1037/met0000733.

Goretzko, D., & Bühner, M. (2020). One model to rule them all? Using machine learning algorithms to determine the number of factors in exploratory factor analysis. *Psychological Methods*, *25*(6), 776−786.

Goretzko, D., & Bühner, M. (2022). Factor retention using machine learning with ordinal data. *Applied Psychological Measurement, 46*(5), 406−421.

Goretzko, D., & Ruscio, J. (2024). The comparison data forest: A new comparison data approach to determine the number of factors in exploratory factor analysis. *Behavior Research Methods*, *56*(3), 1838−1851.

Heaton, J. (2008). *Introduction to neural networks with Java* (2nd ed., pp. 143–172). Heaton Research, Inc.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735−1780.

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. *Psychometrika, 30*(2), 179−185.

Humphreys, L. G., & Montanelli, R. G. (1975). An investigation of the parallel analysis criterion for determining the number of common factors. *Multivariate Behavioral Research, 10*(2), 193−205.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint*:1502.03167.

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement, 20*, 141−151.

Kalinowski, T., Ushey, K., Allaire, J. J., RStudio, Tang, Y., Eddelbuettel, D., ... Geelnard, M. (2025). reticulate: Interface to Python. *R package version 1.42.0*. https://ran.r-project.org/web/packages/reticulate/index.html

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint*: 1412.6980.

Lange, S., Helfrich, K., & Ye, Q. (2022). Batch normalization preconditioning for neural network training. *Journal of Machine Learning Research*, *23*(1), 3118−3158.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature, 521, 436−444*.

Li, Y., Wen, Z., Hau, K.-T., Yuan, K.-H., & Peng, Y. (2020). Effects of cross-loadings on determining the number of factors to retain. *Structural Equation Modeling: A Multidisciplinary Journal*, *27*(6), 841−863.

Lorenzo-Seva, U., Timmerman, M. E., & Kiers, H. A. (2011). The Hull method for selecting the number of common factors. *Multivariate Behavioral Research, 46*(2), 340−364.

Marčenko, V. A., & Pastur, L. A. (1967). Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik, 1*, 457−483.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)* (pp. 807−814). Omnipress.

Peres-Neto, P. R., Jackson, D. A., & Somers, K. M. (2005). How many principal components? Stopping rules for determining the number of non-trivial axes revisited. *Computational Statistics & Data Analysis, 49*(4), 974−997.

Qin, H., & Guo, L. (2024). Using machine learning to improve Q-matrix validation. *Behavior Research Methods*, *56*(3), 1916−1935.

Qin, H., & Guo, L. (2025a). EFAfactors: Determining the number of factors in exploratory factor analysis. *R package version 1.2.1*.

Qin, H., & Guo, L. (2025b). LSTMfactors: Determining the number of factors in exploratory factor analysis by LSTM. *R package version 1.0.0*.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems* (pp. Article 721). Curran Associates Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825−2830.

R Core Team. (2025). *R: A language and environment for statistical computing*. R Foun dation for Statistical Computing, Vienna, Austria. https://www.R-project.org

Ruscio, J., & Kaczetow, W. (2008). Simulating multivariate nonnormal data using an iterative algorithm. *Multivariate Behavioral Research*, *43*(3), 355−381.

Ruscio, J., & Roche, B. (2012). Determining the number of factors to retain in an exploratory factor analysis using

comparison data of known factorial structure. *Psychological Assessment, 24*(2), 282−292.
Soenens, B., & Vansteenkiste, M. (2010). A theoretical upgrade of the concept of parental psychological control: Proposing new insights on the basis of self-determination theory. *Developmental Review, 30*(1), 74−99.
Velicer, W. F. (1976). Determining the number of components from the matrix of partial correlations. *Psychometrika, 41*(3), 321−327.
Wood, J. M., Tataryn, D. J., & Gorsuch, R. L. (1996). Effects of under- and overextraction on principal axis factor analysis with varimax rotation. *Psychological Methods, 1*(4), 354−365.
Zwick, W. R., & Velicer, W. F. (1986). Comparison of five rules for determining the number of components to retain. *Psychological Bulletin, 99*(3), 432−442.

Factor retention in exploratory factor analysis based on LSTM

GUO Lei[1,2], QIN Haijiang[1]

([1] *Faculty of Psychology, Southwest University*; [2]*Southwest University Branch, Collaborative Innovation Center of Assessment toward Basic Education Quality, Chongqing* 400715, *China*)

Abstract

Psychological research focuses on the latent traits of individuals, necessitating clear operational definitions to delineate the constructs of interest. Following this, the exploration and description of the dimensions and characteristics of these traits are essential. Exploratory Factor Analysis (EFA) is a pivotal statistical method for identifying these latent dimensions, widely utilized, especially in the development of psychological scales and instruments.

A critical aspect of employing EFA is the accurate determination of the number of factors. Underestimating the number of factors may result in the omission of theoretically significant psychological structures or sub-dimensions, leading to the loss of critical information, increased estimation errors in factor loadings, and diminished accuracy of factor scores. Conversely, overestimating the number of factors may lead to factors splitting, where the primary loadings of manifest variables are dispersed across multiple factors, thereby weakening the association between the manifest variables and the intended factor. Moreover, this may result in a model characterized by undue complexity and structures of limited practical or theoretical utility. To address these challenges, researchers have proposed various methods, including the Kaiser criterion (i.e., eigenvalues greater than one), the empirical Kaiser criterion, Parallel Analysis, the Hull method, Comparison Data, Factor Forest, and Comparison Data Forest. With the rapid advancement of machine learning, its application in EFA has begun to attract attention. This study introduces an innovative approach by treating eigenvalues as sequential data and leveraging Long Short-Term Memory (LSTM) networks to construct a predictive model. The performance of the LSTM-based method was subsequently evaluated through extensive simulations and empirical studies under diverse data conditions, demonstrating its robustness and applicability.

The findings of the study indicate that: (1) After hyperparameter tuning, an optimal combination was identified, enabling the LSTM model to achieve excellent performance across accuracy, precision, and other evaluation metrics, demonstrating high classification capability. (2) In the simulation study, the LSTM model significantly outperformed Comparison Data Forest, the Empirical Kaiser Criterion, and Parallel Analysis under nearly all data conditions, with an average improvement in estimation accuracy of 48.50% and a maximum improvement of 171.09%.

Furthermore, an empirical study was conducted using data from a parental psychological control scale administered to a cohort of 987 high school students in a city in 2022. Both traditional methods and the LSTM approach were employed to assess ecological validity. The results demonstrated that the LSTM provided the most accurate estimation of the number of factors, while the CDF method exhibited a significant tendency to overestimate. Overall, the LSTM proposed in this study demonstrates strong practical value and is worthy of broader adoption. Researchers can use the R package *LSTMfactors* to call the LSTM trained in this study to analyze empirical data.

Keywords exploratory factor analysis, LSTM, factor retention, deep learning

附录

1　平行分析

利用 PA 进行因子保留的具体步骤如下：

(1)使用主成分分析或因子分析方法计算实证作答数据矩阵 $\boldsymbol{X}_{N\times J}$ 的特征根，作为真实特征根向量 $\boldsymbol{\lambda}$ ($\boldsymbol{\lambda}=\{\lambda_1,\lambda_2,\cdots,\lambda_J\}, 1\leqslant j\leqslant J$)，N 为样本量，I 表示观测变量的总数，也等于特征根序列长度。

(2)抽取 M 次(本研究取 100 次，与 Auerswald 和 Moshage (2019)设置一致)与实证作答数据矩阵同样规格但元素随机的模拟作答数据矩阵 $\boldsymbol{X}_{N\times J}^{ref_m}, m=1,2,\cdots,M$ 。对于 $\boldsymbol{X}_{N\times J}^{ref_m}$ 模拟步骤如下：

a)对于观测变量 i，以 $\boldsymbol{X}_{j|N\times 1}(\boldsymbol{X}_{j|N\times 1}\in\boldsymbol{X}_{N\times J})$，其中 $\boldsymbol{X}_{j|N\times 1}$ 表示矩阵 $\boldsymbol{X}_{N\times J}$ 中的第 j 列数据作为抽样池，采用自助抽样方式有放回地随机抽取矩阵 $\boldsymbol{X}_{N\times J}$ 中第 j 列的 N 个数据，从而形成模拟作答向量 $\boldsymbol{X}_{j|N\times 1}^{ref_m}$。

b)遍历所有观测变量 j，得到 J 个 $\boldsymbol{X}_{j|N\times 1}^{ref_m}$，将所有 J 个 $\boldsymbol{X}_{j|N\times 1}^{ref_m}$ 进行列合并，形成模拟作答数据 $\boldsymbol{X}_{N\times J}^{ref_m}$ 。

c)使用主成分分析或因子分析方法计算模拟作答数据 $\boldsymbol{X}_{N\times J}^{ref_m}$ 的相关系数矩阵的特征根，作为第 m 次模拟作答数据的特征根向量 $\boldsymbol{\lambda}_{1\times J}^{ref_m}$，它是 1 行 J 列的向量，表示第 m 次模拟作答数据的 J 个特征根。

(3)将 M 次模拟得到的特征根向量 $\boldsymbol{\lambda}_{1\times J}^{ref_m}$ 按行合并，形成模拟作答数据的特征根矩阵 $\boldsymbol{\lambda}_{M\times J}^{ref}$ (M 行 J 列的矩阵，包含 M 次模拟作答数据的 J 个特征根)。随后，逐列 ($j=1,2,\cdots,J$) 计算其平均值或 95%的百分位数，记作参考特征根 $\boldsymbol{\lambda}_j^{ref}$ 。

(4)逐一比较真实特征根向量 $\boldsymbol{\lambda}$ 与参考特征根向量 $\boldsymbol{\lambda}^{ref}$ 对应位置的特征根大小。设 F 表示需要保留的因子数量，以 $I(\cdot)$ 为指示函数，若括号内条件为真则取值 1，否则为 0。F 的计算公式如下：

$$F=\sum_{j=1}^{J} I(\lambda_j>\lambda_j^{ref}) \tag{1}$$

2　经验凯撒准则

在零模型下，随机数据的相关系数矩阵的特征根渐近服从 Marčenko-Pastur 分布，以该分布的上限作为第一个参考特征根 λ_1^{ref}，即有：

$$\lambda_1^{ref}=\left(1+\sqrt{\frac{J}{N}}\right)^2 \tag{2}$$

其中，N 是样本量，J 是观测变量的数量。后续参考特征根的值使用下式计算：

$$\lambda_j^{ref}=max\left[\frac{J-\sum_{k=1}^{j-1}\lambda_k}{J-j+1}\left(1+\sqrt{\frac{J}{N}}\right)^2,1\right] \tag{3}$$

其中，λ_j^{ref} 表示第 j 个参考特征根值，λ_k 是实证作答数据的第 k 个真实特征根。EKC 同时遵循凯撒准则，即保留公式(2)中 $\frac{J-\sum_{k=1}^{j-1}\lambda_k}{J-j+1}\left(1+\sqrt{\frac{J}{N}}\right)^2$ 与 1 两者中的最大值。EKC 通过比较实证作答数据的第 j 个特征根 λ_j 与其参考特征根值 λ_j^{ref} 来决定是否保留第 j 个因子，因子数量的计算公式同公式(1)。

3　比较数据森林法

CDF 使用与 CD 方法相同的模拟数据方法，该模拟方法可以参见 Ruscio 和 Kaczetow (2008)的研究。为了方便读者理解，下文中的“通常”一词均指代 Auerswald 和 Moshagen (2019)与 Goretzko 和 Bühner (2020)研究中的取值，这两项研究对于数据模拟的参数设置是相同的。

CDF 方法的具体步骤为：

(1)基于实证作答数据矩阵 $\boldsymbol{X}_{N\times J}$ 计算得到相关系数矩阵 $\boldsymbol{\Sigma}$，生成因子数量为 f ($f=1,2,\cdots,F$) 的模拟作答数据矩阵(通常包含 $P=10000$ 个被试)，记作 $\boldsymbol{X}_{P\times J}^{f}$，表示具有 f 个潜在因子的总体。其中，F 表示拟讨论的最大因子数量，本研究取 $F=10$ 。具体而言，模拟 $\boldsymbol{X}_{P\times J}^{f}$ 的过程如下：

a)以实证作答数据计算所得的相关系数矩阵 $\boldsymbol{\Sigma}$ 作为目标相关系数矩阵。

b)随机生成 P 个被试在 f 个因子上的公因子得分 $\boldsymbol{S}_{P\times f}$ (P 行 f 列的矩阵)，以及 P 个被试在 J 个观测变量上的残差得分 $\boldsymbol{U}_{P\times J}$ (P 行 J 列的矩阵)，每一个公因子得分与残差得分均服从独立的标准正态分布：

$$\boldsymbol{S}_{P\times f}\sim N(0,1) \tag{4}$$

$$\boldsymbol{U}_{P\times I}\sim N(0,1) \tag{5}$$

c)初始化 $\boldsymbol{\Sigma}^a=\boldsymbol{\Sigma}$ 。初始化均方根残差 $\mathrm{RMSR}^a=Inf$，表示无穷大。初始化计数器 $t=0$ 。

d)采用主轴因子法从 $\boldsymbol{\Sigma}^a$ 中抽取 f 个因子，得到因子载荷矩阵 $\boldsymbol{\Lambda}_{J\times f}^{shared}$，表示 J 行 f 列的矩阵。

e)计算特殊因子矩阵 $\boldsymbol{\Lambda}_{J\times 1}^{unique}$，表示 J 行 1 列的向量，其中元素 j 的计算公式如下：

$$\Lambda_j^{unique}=\sqrt{1-\sum_{k=1}^{f}(\Lambda_{j,k}^{shared})^2} \tag{6}$$

f)模拟出作答数据矩阵 $\boldsymbol{X}_{P\times J}^{f}$，其中第 $p(p=1,2,\cdots,P)$ 个被试在观测变量 j 上的模拟作答是：

$$X_{p,j}^{f}=\boldsymbol{S}_{p,1:f}(\boldsymbol{\Lambda}_{j,1:f}^{shared})^T+U_{p,j}\Lambda_j^{unique} \tag{7}$$

g)计算模拟作答数据矩阵 $\boldsymbol{X}_{P\times J}^{f}$ 的相关系数矩阵 $\boldsymbol{\Sigma}^b$，以及残差相关矩阵 $\boldsymbol{\Sigma}_{Res}=\boldsymbol{\Sigma}-\boldsymbol{\Sigma}^b$，将残差相关矩阵 $\boldsymbol{\Sigma}_{Res}$ 的下三角中的元素记作 Res，从而可以计算得到 RMSR ：

$$\mathrm{RMSR}=\sqrt{\frac{\sum Res^2}{0.5\times J\times(J-1)}} \tag{8}$$

h) 若 $\mathrm{RMSR}<\mathrm{RMSR}^a$，则更新 $\boldsymbol{\Sigma}^a=\boldsymbol{\Sigma}^a+r\times\boldsymbol{\Sigma}_{Res}$（$r$ 为学习率，通常取 1），$\mathrm{RMSR}^a=\mathrm{RMSR}$，并重置计数器 $t=0$。否则，$\boldsymbol{\Sigma}^a=\boldsymbol{\Sigma}^a+0.5\times t\times r\times\boldsymbol{\Sigma}_{Res}$，并更新计数器 $t=t+1$。

i)重复步骤 d)至步骤 h)，直到 $t>t_{\max}$（$t_{\max}$ 通常为 5)。基于所输出的模拟作答数据矩阵 $\boldsymbol{X}_{P\times J}^{f}$ 所计算得到的相关系数矩阵将会近似于目标相关系数矩阵 $\boldsymbol{\Sigma}$，且 $\boldsymbol{X}_{P\times J}^{f}$ 具有 f 个潜在因子。

(2)采用自助法从 $\boldsymbol{X}_{P\times J}^{f}$ 数据集中抽取出 K 次与原始作答数据矩阵 $\boldsymbol{X}_{N\times J}$ 规格相同的模拟作答数据，第 k 次模拟作答数据记作 $\boldsymbol{X}_{N\times J}^{f,k}$，其中 $k=1,2,\cdots,K$ (通常取 5000)。

(3)采用与 FF 相同的特征工程，即计算每个样本数据矩阵 $\boldsymbol{X}_{N\times J}^{f,k}$ 的 181 个训练特征，具体的 181 个训练特征请参见 Goretzko 和 Bühner (2020)的研究。通过该方式便可形成用于构建因子森林的训练数据集 *TD*。

(4)以训练数据集 *TD* 中的因子数量 f 为标签，训练执行多分类任务(如 1 至 10 个因子)的随机森林。参考 Goretzko 和 Ruscio (2024)的设置，森林的树数量取 500，每棵树的最大深度为 $\sqrt{181}\approx13$。

(5)将训练好的 CDF 模型用于预测实证作答数据的因子数量。

4　LSTM 原理及结构

一个典型的 LSTM 记忆元如图 A1 所示。

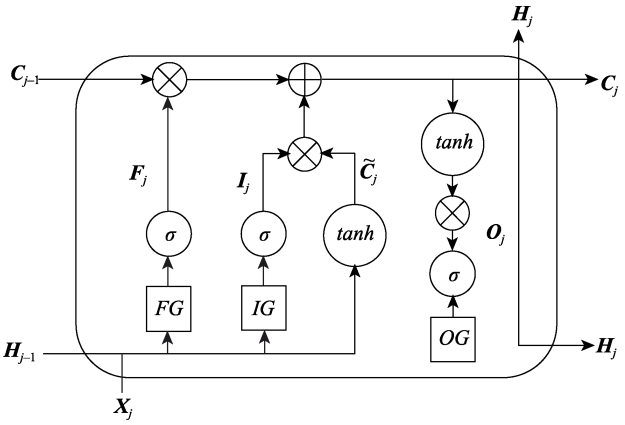

图 A1　LSTM 记忆元示意图

图 A1 中，$\boldsymbol{X}_j$ 为时间步为 j 的输入特征(在本研究中即为训练特征 $\boldsymbol{X}_j=\{\lambda_j,\lambda_j-\lambda_j^{ref}\}$)，$\tilde{C}_j$ 为候选记忆元(Candidate Memory Cell)，$\boldsymbol{C}_{j-1}$ 和 $\boldsymbol{C}_j$ 分别为前一个和当前时间步的记忆元(Memory Cell)，$\boldsymbol{H}_{j-1}$ 和 $\boldsymbol{H}_j$ 分别为前一个和当前时间步的隐藏状态(Hidden State)，σ 为 Sigmoid 激活函数，*tanh* 是另一种激活函数，⊗ 表示元素积，⊕ 表示元素和，*FG* 表示遗忘门(Forget Gate)，*IG* 表示输入门(Input Gate)，*OG* 表示输出门(Output Gate)，F_j、I_j 和 O_j 分别是遗忘门、输入门和输出门在当前时间步 j 的输出。三个门的输出以及 $\tilde{C}_j$ 的计算如下列公式所示：

$$\boldsymbol{F}_j=\sigma(\boldsymbol{X}_j\boldsymbol{W}_{x,FG}+\boldsymbol{H}_{j-1}\boldsymbol{W}_{h,FG}+\boldsymbol{b}_{FG}) \tag{9}$$

$$\boldsymbol{I}_j=\sigma(\boldsymbol{X}_j\boldsymbol{W}_{x,IG}+\boldsymbol{H}_{j-1}\boldsymbol{W}_{h,IG}+\boldsymbol{b}_{IG}) \tag{10}$$

$$\boldsymbol{O}_j=\sigma(\boldsymbol{X}_j\boldsymbol{W}_{x,OG}+\boldsymbol{H}_{j-1}\boldsymbol{W}_{h,OG}+\boldsymbol{b}_{OG}) \tag{11}$$

$$\tilde{\boldsymbol{C}}_j=tanh(\boldsymbol{X}_j\boldsymbol{W}_{x,\tilde{C}}+\boldsymbol{H}_{j-1}\boldsymbol{W}_{h,\tilde{C}}+\boldsymbol{b}_{\tilde{C}}) \tag{12}$$

其中，$\boldsymbol{W}_{x,FG}$、$\boldsymbol{W}_{x,IG}$、$\boldsymbol{W}_{x,OG}$ 和 $\boldsymbol{W}_{x,\tilde{C}}$ 分别是遗忘门、输入门、输出门和候选记忆元中针对输入特征 $\boldsymbol{X}_j$ 的权重参数，$\boldsymbol{W}_{h,FG}$、$\boldsymbol{W}_{h,IG}$、$\boldsymbol{W}_{h,OG}$ 和 $\boldsymbol{W}_{h,\tilde{C}}$ 分别是遗忘门、输入门、输出门和候选记忆元中针对隐藏状态 $\boldsymbol{H}_{j-1}$ 的权重参数，$\boldsymbol{b}_{FG}$、$\boldsymbol{b}_{IG}$、$\boldsymbol{b}_{OG}$ 和 $\boldsymbol{b}_{\tilde{C}}$ 分别是遗忘门、输入门、输出门和候选记忆元的偏置参数。LSTM 通过遗忘门决定丢弃哪些旧信息、通过输入门决定添加哪些新信息，两者共同决定当前的记忆元：$\boldsymbol{C}_j=\boldsymbol{F}_j\otimes\boldsymbol{C}_{j-1}+\boldsymbol{I}_j\otimes\tilde{\boldsymbol{C}}_j$，输出门则决定输出哪些信息：$\boldsymbol{H}_j=\boldsymbol{O}_j\otimes tanh(\boldsymbol{C}_j)$。在一个 LSTM 层中，可以有多个完整的记忆单元，这些记忆单元将共享同样的权重参数，但它们可以具备不同的遗忘门、输入门、输出门等，因此也将学习捕捉输入序列中不同方面的特征，增强 LSTM 表征复杂序列的能力。

全连接层与批量标准化层的结构相对于LSTM层而言更为简单。全连接层只需要考虑权重与偏置参数，即若输入 $\boldsymbol{X}$，则输出 $\boldsymbol{Y}=AF(\boldsymbol{X}\boldsymbol{W}+\boldsymbol{b})$，$AF$ 为任意激活函数(例如 ReLU、sigmoid 或 tanh 等)。批量标准化层的作用是在网络训练过程中对每一小批量(mini-batch)的输入信息进行标准化，以缓解内部协变量偏移(Internal Covariate Shift)问题，从而加快模型收敛速度并提高泛化能力。具体而言，对于给定的小批量输入 $\boldsymbol{X}=\{X_1,X_2,\cdots,X_m,\cdots,X_M\}$，批量标准化层首先计算该批次的均值 $\mu=\frac{1}{M}\sum X_m$ 与方差 $\sigma^2=\frac{1}{M}\sum(X_m-\mu)^2$，接着对每个样本进行标准化处理：

$$\hat{X}_m=\frac{X_m-\mu}{\sqrt{\sigma^2+\epsilon}} \tag{13}$$

其中，ϵ 是一个防止除以 0 的微小常数，最后引入可学习的缩放系数 γ 和偏移量 β，以恢复网络的表达能力：

$$Y_m=\gamma\hat{X}_m+\beta \tag{14}$$

在批量标准化层中，$\hat{X}_m$ 与 Y_m 仅是为了稳定训练过程而出现的中间状态，此外还具有轻微的正则化作用，能在一定程度上避免过拟合。

图 A2 呈现了本研究所使用的 LSTM 层+全连接层+批量标准化层的神经网络基本结构示例。该结构包含了两个 LSTM 层，用于处理步长为 I 的序列数据 $\boldsymbol{X}$、1 个

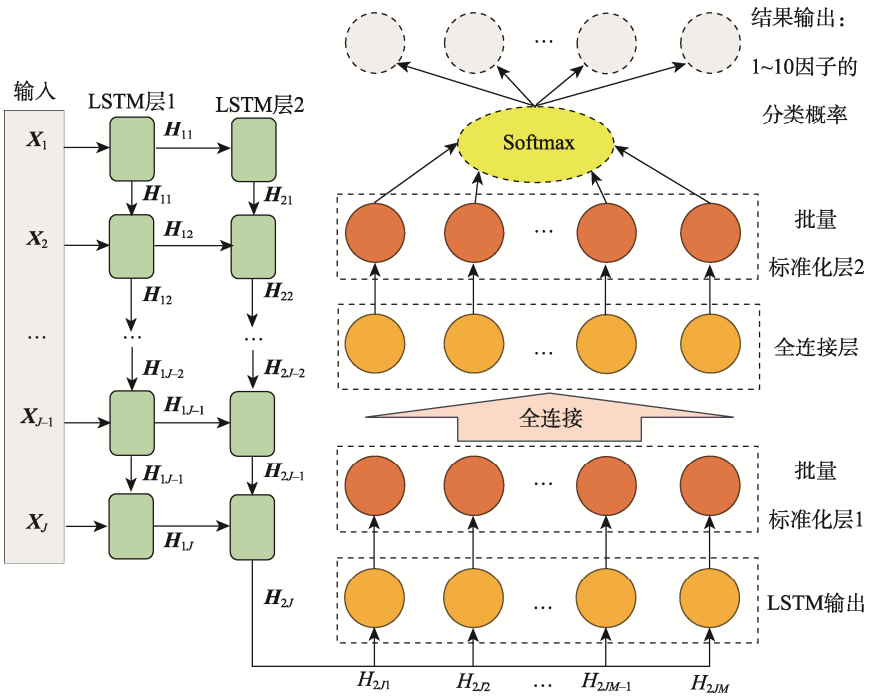

图 A2　LSTM+全连接层+批量标准化层的神经网络基本结构示例

全连接层以及两个批量标准化层,并且以 Softmax 作为输出层的激活函数以执行多分类任务。

5　多分类任务的混淆矩阵与评估指标

对于执行多分类任务的模型，可以使用混淆矩阵计算其宏平均的准确率(*Accuracy*)、精确率(*Precision*)、召回率(*Recall*)、*F*1 分数和 *Kappa* 系数，这些指标可以使用 Python 库 *sklearn* 便捷计算。以三分类任务为例，其混淆矩阵可表示如下：

表 A1　三分类任务的混淆矩阵示例

		预测类别			总计
	类别	C1	C2	C3	
真实类别	C1	*a*	*g*	*h*	*j*
	C2	*d*	*b*	*i*	*k*
	C3	*e*	*f*	*c*	*l*
总计		*m*	*n*	*o*	*p*

各指标计算公式如下：

$$Accuracy=\frac{a+b+c}{p} \tag{15}$$

$$Precision=\frac{\frac{a}{m}+\frac{b}{n}+\frac{c}{o}}{3} \tag{16}$$

$$Recall=\frac{\frac{a}{j}+\frac{b}{k}+\frac{c}{l}}{3} \tag{17}$$

$$F1=\frac{2}{\frac{1}{Precision}+\frac{1}{Recall}} \tag{18}$$

$$Kappa=\frac{P_0-P_e}{1-P_e} \tag{19}$$

$$P_0=Accuracy \tag{20}$$

$$P_e=\sum\frac{row_x\times col_x}{\omega^2}=\frac{jm+kn+lo}{p^2} \tag{21}$$

6　实证研究示例代码

PA、EKC 和 CDF 均可通过 R 包 *EFAfactors* 实现，执行 R 代码如下(代码以“R>”开头，输出结果紧跟代码呈现):

```
R> set.seed(112)        ## 固定随机数种子
R> library(EFAfactors)    ## 导入 EFAfactors 库
EFAfactors R Package (version 1.2.1; 2025-02-15)
R> PA.obj <- PA(response = response, plot = TRUE)
## 运行 PA
The number of factors suggested by PA (quant=0.95) is 3 .
R> EKC.obj <- EKC(response = response, plot =TRUE)
## 运行 EKC
The number of factors suggested by EKC is 3 .
R> CDF.obj <- CDF (response = response, nfact. max = 10,
                   N.pop = 10000, N.Samples =
                   5000, mtry = "sqrt")   ## 运行 CDF
CDF is simulating data: nfact=10/10 - N_rep= 5000/5000
The number of factors suggested by CDF is 7 .
```

LSTM 可以通过 R 包 *LSTMfactors* 的 LSTM( )函数调用，执行 R 代码如下：

```
R> library(LSTMfactors)   ## 导入 LSTMfactors 库
LSTMfactors R Package (version 1.0.0; 2025-06-25)
R> LSTM.obj <- LSTM(response = response, plot =
TRUE)   ## 运行 LSTM
The number of factors suggested by LSTM is 4.
```