Computer Science and Application 计算机科学与应用, 2024, 14(2), 517-526
Published Online February 2024 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2024.142051

Hans汉斯

基于多头注意力与双向门控循环单元的数据无损压缩方法

鄂　驰，胡　潇，刘小康，张尚军，熊小舟

国网江西省电力有限公司信息通信分公司，江西　南昌

收稿日期：2024年1月26日；录用日期：2024年2月22日；发布日期：2024年2月29日

摘　要

为解决数据归档存储场景中出现的物理存储成本增长和数据库内存紧张等问题，本文提出一种基于注意力机制与双向门控循环单元的数据无损压缩方法，采用Transformer和双向门控循环单元作为概率预测器，输出数据流的条件概率分布，结合自适应算术编码器对数据进行压缩。实验对比结果表明，本文所提方法相较于算术编码和基于字典模型的LZW这两种传统无损压缩方法，压缩率分别平均提升约28.8%和7.8%；相较于Cmix v19和NNCP两种深度学习方法，平均压缩率分别降低0.4%和0.2%，但平均压缩时间分别约为其5.1%和39.4%。

关键词

数据无损压缩，双向门控循环单元，Transformer

Lossless Data Compression Method Based on Multi-Head Attention and Bidirectional Gated Recurrent Unit

Chi E, Xiao Hu, Xiaokang Liu, Shangjun Zhang, Xiaozhou Xiong

Information and Communication Branch, State Grid Jiangxi Electric Power Co., LTD., Nanchang Jiangxi

Received: Jan. 26th, 2024; accepted: Feb. 22nd, 2024; published: Feb. 29th, 2024

Abstract

To solve the problems of rising physical storage cost and memory of database limitation in the archiving and storing scenario of data, a lossless data compression method based on multi-head

文章引用: 鄂驰, 胡潇, 刘小康, 张尚军, 熊小舟. 基于多头注意力与双向门控循环单元的数据无损压缩方法[J]. 计算机科学与应用, 2024, 14(2): 517-526. DOI: 10.12677/csa.2024.142051

attention and bidirectional gated recurrent unit is proposed in this paper. Transformer and bidirectional gated cycle unit are used as probability predictors to output conditional probability distribution of data flow and an adaptive arithmetic encoder is combined to compress the data. The experimental results show that compared with the two traditional lossless compression methods, arithmetic coding and LZW based on the dictionary model, the compression ratio of the proposed method is improved by 28.8% and 7.8% respectively. Compared with Cmix v19 and NNCP, the average compression rate is reduced by 0.4% and 0.2%, respectively, but the average compression time is about 5.1% and 39.4%, respectively.

Keywords

Lossless Data Compression, Bidirectional Gated Recurrent Unit, Transformer

1. 引言

用电信息是电力消费情况的数据，反映信息采集区域内用户真实用电情况，这些业务数据具重要有保存价值。这些业务数据中，新产生数据被频繁读取或更新，历史数据很少被访问。例如，目前国家电网业务范围内拥有海量用户，数据库系统中积累大量访问频率低的历史数据，其中仅有部分关键信息具有存储价值。因此，对关键历史数据进行归档与管理，是各个电网公司的通用做法[1] [2] [3]。随着用电数据的积累，数据量呈直线上升趋势，将不可避免造成存储硬件成本急剧增长，是亟需解决的行业问题。

数据压缩算法对关键历史数据进行归档压缩，是降低数据存储硬件开销的有效方式。存储对象是关键历史数据，具有一定研究价值，需要采用无损压缩的方式。目前，主流的无损压缩是基于字典的 LZ 系列算法[4] [5]，但其压缩过程是基于模式匹配进行，当处理变化程度较大的业务数据，特别是浮点数据时，压缩率较低。

目前，深度学习方法凭借其端到端、对数据潜在特征有效挖掘等优势，成为计算机视觉、数字图像处理和自然语言处理等领域的研究热点。其中，循环神经网络，已应用于图像压缩[6]、轨迹压缩[7]；Transformer 已应用于字节对编码[8]、压缩成像[9]、压缩感知磁共振重建[10]。在具备学习复杂关系映射并形成泛用近似函数能力的神经网络模型中，Transformer [11] [12]作为其中一种高效的全局依赖模型，它的编码器层(Encoder Layer)采用多头注意力(Multi-Head Attention)和前馈网络(Feed Forward Network)，能将每个输入的令牌向量表示为对应的上下文向量，为原始输入嵌入丰富上下文信息。门控循环神经网络(Gated Recurrent Neural Network, GRNN) [13] [14]具有建立长期记忆的能力，可更好捕捉时间序列中时间步距离较大的依赖关系。本文采用 Transformer 和门控循环神经网络相结合的网络模型作为序列数据的条件概率预测器，经过样本训练，保存神经网络权值。编码时，载入网络权重构建概率预测器，计算当前数据流中字符分布情况，将预测结果与当前字符送入自适应算术编码器中，更新概率区间并将字符编码输出。同样，解码时，将预测器输出的概率分布送入自适应算术编码器内，输出解压后字符。本文提出的方法和 LZW 编码[4]、算术编码[15]两种传统压缩方法，以及 NNCP [16]、Cmix [17]两种基于深度学习的压缩方法进行对比。实验结果表明，本文提出的方法在压缩率和压缩时间上取得较好的平衡，适应于解决实际工程中的数据无损压缩问题。本文的主要贡献概括如下：

1) 设计基于 Transformer 和双向门控循环单元的概率预测模型，低运算复杂度下输出高精度预测结果。
2) 提出的数据压缩模型与自适应算术编码结合，提升了数据无损压缩算法的局部自适应特性。
3) 提出一种训练加速算法，在基本不损失压缩精度的情况下，加快模型训练速度，节省数据压缩时间。

2. 数据无损压缩网络模型

2.1. Transformer-GRU 概率预测器

采用单编码器层后接双向门控循环单元(Bidirectional Gated Recurrent Unit, BGRU)作为预测器的网络主干，在保证压缩、解压速率情况下融合上下文信息，同时防止梯度消失和长程依赖减弱等问题的出现。概率预测器模型架构如图 1 所示。

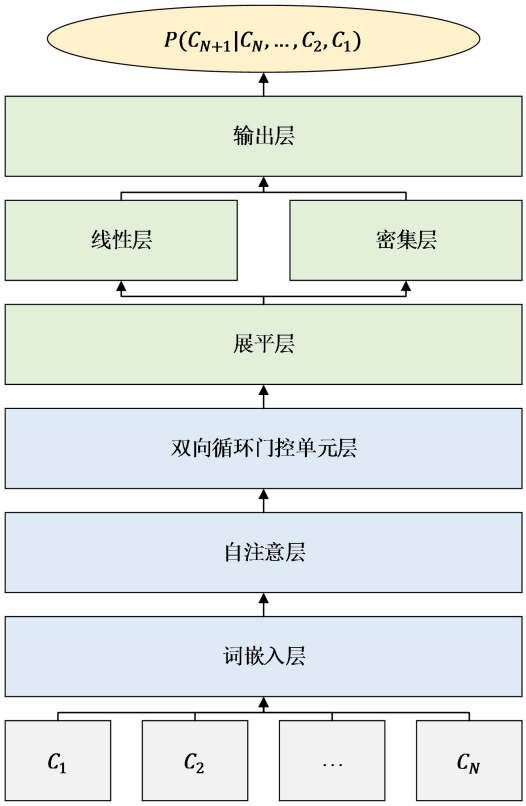


Figure 1. Network structure of the Transformer-GRU probability predictor
图 1. Transformer-BGRU 概率预测器网络架构

概率预测器模型输入为长度为 N 的字符串。首先，字符串输入词嵌入层，该层包括词嵌入和绝对位置嵌入，字符串中每个字符根据其在字典中的位置，进行词嵌入；字典包含待压缩文件中出现的字符词向量，词向量大小为 M/2，即特征维度的一半。同时，字符根据自身位置进行绝对位置嵌入，加入位置信息，与词嵌入进行连接，使相互独立的输入字符转换为在词义上、位置上具有内在联系的特征向量。通过自注意力机制的编码，这些特征向量包含上下文信息，同时特征维度不变。随后，特征向量进入双向门控循环单元层，该层门控循环单元数量为 2N，即输入序列长度的两倍；经过计算，输出融合双向上下文的特征向量(N, 2M)。进入展平层，对特征向量进行级联前，进行等间隔采样，采样间隔设置为 K，

得到(1, N/K, 2M)维向量。级联得到(1, N*2M/K)维向量，通过一个线性层和一个密集层，输出维度均设置为字典大小，来预测字典中每一个字符出现概率，最后将来自两个不同特征维度的预测结果相加通过LogSoftmax 函数输出。模型经过以上步骤输出最终预测结果，用于下一步自适应算术编码计算。

2.2. 编码层(Encoder Layer)

编码层由多头注意力和前馈网络经过残差连接构成，如公式(1)示。

$$Encoder(x) = FFN(Attention(x) + x) + Attention(x) + x \tag{1}$$

前馈网络是 Transformer 结构的一个重要子组件，如公式(2)所示。

$$FFN(x) = \mathrm{Re}LU(xW^1)W_2 \tag{2}$$

其中，W^1 将 MHA 的输出映射到高维空间，W^2 将通过激活函数的高维向量投影回原始向量空间。

注意力函数 $Attention(x)$中包含三个矩阵 W^Q、W^K、W^V，通过矩阵乘法，将输入 x 通过线性投影转化为三个分量：查询 Query、键 Key、值 Value。之后，查询 Q 和键 K 相乘以计算相关性的度量，即注意力矩阵，再将注意力矩阵和值 V 相乘，得到每个输入上下文向量表示。注意力函数如公式(3)所示。

$$Attention(x) = SoftMax\left(\frac{xW^Q * xW^{K^T}}{\sqrt{h_K}}\right)xW^V \tag{3}$$

其中，h_K 是隐藏维度，作为缩放因子。

与只计算一次注意力不同，多头机制以并行方式进行多个独立注意力计算。每个注意力头的输出级联起来，通过线性变换转换为标准维度。多头注意力函数如公式(4)和(5)所示。

$$MuitlHead(Q,K,V) = [head_1, head_2, \cdots, head_n]W \tag{4}$$

$$head_i = Attention(xW_i^Q, xW_i^K, xW_i^V) \tag{5}$$

其中W^Q、W^K、W^V和W是可以学习的参数矩阵。

2.3. 双向门控循环单元

双向门控循环单元(BGRU) [13]分别以顺序和逆序依次处理输入序列，并将每个时间步门控循环单元(GRU) [14]的输出拼接成最终输出层。与 GRU 相比，每个时间步的输出节点，都包含输入序列中当前时刻过去和未来的上下文信息。GRU 由更新门和重置门组成，结构如图 2 所示。

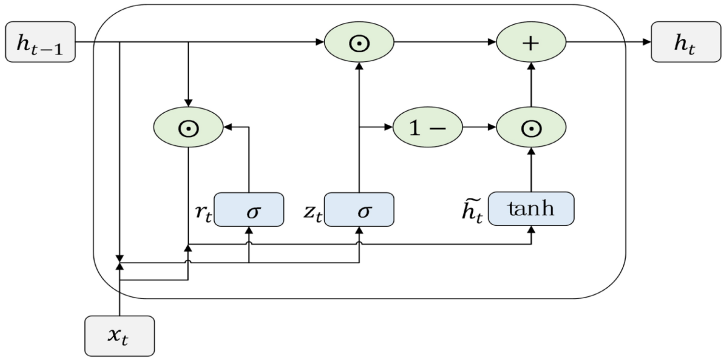


Figure 2. Gated recurrent unit
图 2. 门控循环单元

其中，更新门 z 决定是否使用新的隐藏层状态 $\tilde{h}_t$ 更新隐藏层状态 h，重置门 r 决定是否将之前的隐藏层状态 h 遗忘。重置门 r_t 将当前输入字符特征与上一个时间步的隐藏层状态分别进行线性变换后相加，如公式(6)所示。

$$r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \tag{6}$$

类似地，更新门 z_t 计算如公式(7)所示。

$$z_t = \sigma\left(W_z x_t + U_z h_{z-1} + b_z\right) \tag{7}$$

重置门决定当前隐藏状态 $\tilde{h}_t$，如公式(8)所示。

$$\tilde{h}_t = \tanh\left(W_h x_t + U_h\left(r_t \odot h_{t-1}\right) + b_h\right) \tag{8}$$

更新门决定之前隐藏状态和当前隐藏状态以何种方式结合为最终输出，如公式(9)所示。

$$h_t = z_t \odot h_{t-1} + \left(1 - z_t\right) \odot \tilde{h}_t \tag{9}$$

其中 σ 为 sigmoid 函数，将输出变换到 0~1 之间，x_t 为当前 t 时刻的输入，h_{t-1} 为 t 时刻前一时刻 $t-1$ 的隐藏层状态，$\odot$ 为同或运算，W 和 U 为可学习权重矩阵。

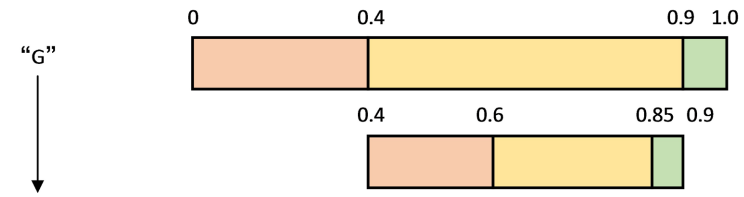
2.4. 算术编码

算术编码[18]是统计编码的一种，通过算法将整个输入流转换为一个很长的数值进行编码。该算法初始时产生一个概率表，表中记录每一个字符出现概率和该字符对应取值区间，总区间为[0, 1]。图 3 为算术编码概率表。

符号	概率	取值区间
R	0.4	[0, 0.4]
G	0.5	[0.4, 0.9]
B	0.1	[0.9, 1]

Figure 3. Arithmetic coding probability table
图 3. 算术编码概率表

首先读入一个字符，按照概率表中字符出现概率，选取读入字符所在取值区间为总区间进行重新划分，此时总区间范围缩小，将每个字符所分配到的新取值区间填入概率表中。如图 4 所示。重复上述操作，直到所有字符读取完毕，在取值区间内选择一个用最少二进制位表示的数作为编码结果。



符号	概率	取值区间
R	0.4	[0.4, 0.6]
G	0.5	[0.6, 0.85]
B	0.1	[0.85, 0.9]

Figure 4. The change after reading characters in the probability table
图 4. 概率表读入字符后的变化

算法在初始化时产生一个概率表，读取过程中字符出现概率不变。因此，编码之前需要遍历整个输入流，统计字符的全局概率，神经网络模型无法做到一次将全部数据输入，这样会带来巨大的计算量，严重影响压缩效率。算术编码虽然能为出现频率较大的符号赋予较短码字，为出现概率较低的符号赋予较长码字，但当某一字符在局部出现频率较高，全局出现频率较低时若再为其赋予较长码字会降低压缩效果。以上特性使得深度学习难以适用在算术编码的概率估计中，且压缩效果较差。

2.5. 自适应算术编码

自适应算术编码[19]是算术编码的改进，在根据当前读入字符重新划分取值区间后，概率表会根据读入的字符重新统计出现概率。相比于固定概率表的编码，具有良好的局部自适应特性，在字符分布情况复杂的数据流上也能取得更高的压缩率。同时不需要提前统计字符的全局概率，使得深度学习能够较好地应用在概率估计中。

3. 模型训练方法

对于输入序列$C^N=\{C_1,C_2,\cdots,C_N\}$，本文提出的模型基于对前 N 个字符的观察，做出对第 N+1 个字符C_r的条件概率$\hat{P}\left(C_r\middle|C_{r-1},C_{r-2},\cdots,C_{r-N}\right)$的预测。

读取待压缩文件，得到字符流为长度为 L 的序列，由于序列长度过长，不能直接用来训练模型。因此，要将一维的序列转化为满足训练要求二维的矩阵，即有重叠的分段矩阵。字符流的前 N+1 个字符作为该矩阵的首行，包含预测字符的前置序列和实际出现的字符，接着将下标 2 至 N+2 的字符作为次行，以此类推，直至满足该要求的最后一行，至此作为训练集的重叠分段矩阵 M 构建完成。

训练时，将 M 打乱，分为大小为 2048 的 Batch。实际训练中 N 取 64，epoch 设置为 2，采用 Adam 优化器，为防止过拟合，对 Adam 使用衰减的学习率，模型的预期输出为若干在不同上下文情况下对所有出现字符的归一化预测结果。使用 NLLLoss 作为损失函数计算每一个预测样本的负对数似然，评估预测结果。NLLoss 如公式(10)所示。

$$NLL\left(\log\left(softmax\left(input\right)\right),target\right)=-\sum_{i=0}^{n}OneHot\left(target\right)_i*\log\left(softmax\left(input\right)_i\right) \tag{10}$$

其中 *input* 表示对 *n* 类的预测。训练完成后，网络模型权重被保存，用于实际压缩与解压。

4. 压缩与解压方法

4.1. 压缩方法

在实际压缩过程中，由于前 N 个字符的前置字符串长度不足 N，因此对于前 N 个字符，使用均匀条件概率分布的自适应算术编码器对其编码。之后，将每个字符的前置字符串送入已加载训练权重的网络内部，字符本身配合输出的预测条件概率分布作为输入进入自适应算术编码器进行编码，同时自适应算术编码器调整概率表，重新划分取值区间。重复以上操作直至最后一个字符，输出最终编码结果，将所有编码写入到压缩文件中，完成压缩。如图 5 所示为本文所提方法的压缩过程。

算法在初始化时产生一个概率表，在读取过程中字符的出现概率不会变化，因此在编码之前需要遍历整个输入流，统计字符的全局概率，神经网络模型无法做到一次将全部数据输入，因为这样会带来巨大的计算量，严重影响压缩效率。算术编码虽然能为出现频率较大的符号赋予较短的码字，为出现概率较低的符号赋予较长码字，但是当某一字符在局部出现频率较高，全局出现频率较低时若再为其赋予较长码字会降低压缩效果。以上特性使得深度学习难以适用在算术编码的概率估计中，且压缩效果往往较差。

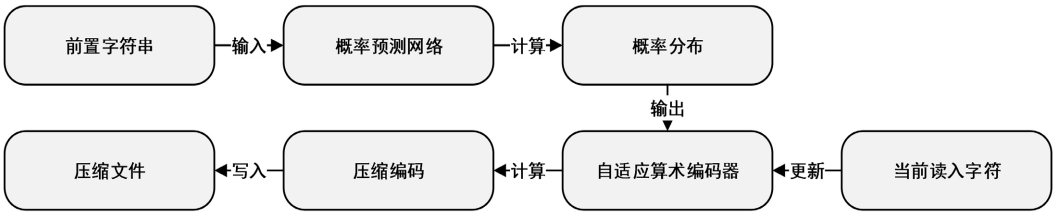


Figure 5. The compression process of proposed method
图 5. 本文方法压缩过程

4.2. 解压方法

解压过程为上述操作的逆操作，使用均匀条件概率分布的自适应算术编码器，利用编码对照取值区间，获得首字符，更新概率表，重新划分取值区间，重复得到前 N 个字符。之后使用同样的方式预测当前字符的条件概率分布，在解码得到字符后，更新概率表，重新划分取值区间，直到解码所有字符。

4.3. 训练加速方法

不同数据文件，其内部数据的种类及分布不同。基于此，本文提出的基于的数据压缩算法针对每一个压缩对象训练对应的权重。在实际使用时，由于网络的训练往往需要较长的时间，上述特性会严重影响数据压缩算法的效率。因此，如何在不损失精度的情况下，最大限度地提高训练速度，成为了提高压缩速率的关键。

模型的训练过程中，反向传播更新权重实际上就是在模拟当前数据流中数据的分布，该操作在模型训练时间中占有较大比重。在数据流的读取过程中，会出现数据分布极为接近甚至是相同的情况，在这种情况下，进行反向传播反而会使压缩效果变差。因此，本文提出一种避免在上述情况下进行反向传播的算法，从而提高模型训练速度。具体步骤为开辟长度为 k 的缓存，存储当前时刻 t 的前 k 个时间步上的 loss 值，比较当前预测结果的 loss 值，若大于前 k 个时间步的 loss 值均值，则认为需要当前对数据分布不准确，需要进行反向传播更新权重；否则不进行反向传播。最后，将缓存中最早的 loss 值剔除，将当前时间步的上的 loss 值写入。具体形式如下：

算法 1. 基于反向传播控制的训练加速算法

输入：当前时间步 loss 值 l_t，缓冲区长度 k
输出：当前反向传播控制值 c
a) if 缓冲区未满
b)　　将 l_t 输入缓冲区中；
c) else
d)　　计算缓冲区 loss 值均值 l_k；
e)　　if $l_t > l_k$
f)　　　　c = True；//需要进行反向传播
g)　　else
h)　　　　c = False；//无需进行反向传播
i) l_t 替代缓冲区最先进入的 loss 值

5. 实验与分析

5.1. 数据集

用电采集数据与常见的文本数据不同，大部分数据重复性大，且有相同的结构、格式。另一小部分

数据浮动范围较大，几乎不重复。造成这种情况的原因是实际情况下数据库内不同的数据表有不同的数据项，例如，在电流信息表中有多达 43 个数据项，对于不同的采集样本，某些数据项下具有相同的值，而另一些数据项下则差异较大，例如采集日期和用电值。

对于关键历史数据的选取，需具备一定代表性与研究价值。对于数量众多的数据项，本文选取两日内较为关键部分构成训练数据集以供模型训练与压缩，如表 1 所示。

Table 1. Training set sample description
表 1. 训练集样本说明

样本数	训练集大小/kB	数据项
1,098,772	512,000	103
1,098,700	512,000	103

由于训练数据集由具有代表性的关键信息构成，因此大部分数据项下的数值重复率均较低，主要由整型与浮点型数据构成。按照采集日期的不同，其数值会出现较大浮动。具有此特性的该类数据是本文在实际应用场景中压缩算法主要压缩的对象，因此在构建训练数据集时，尽可能多的涵盖较多样本，验证本文提出方法在实际应用场景下的有效性。

5.2. 评价标准

为了评估算法的精度与时效性，在与传统算法的比较部分，本文使用压缩率(Compression rate)作为评估标准，将模型作为压缩文件的一部分；在与主流基于深度学习的压缩算法比较部分，使用压缩率与压缩时间(Compression time)作为评估标准，由于其中部分方法采用多个模型，因此仅考虑压缩文件的大小；在是否使用训练加速方法比较部分，使用压缩率与训练时间作为评估标准，同样仅考虑压缩文件的大小。压缩率和压缩时间可定义为式(11)和(12)所示：

$$\text{Compression rate} = 1 - \frac{x_o}{x_c} \tag{11}$$

$$\text{Compression time} = t_i - t_o \tag{12}$$

其中，x_o、x_c 分别表示源文件大小、压缩文件大小，t_i、t_o 分别表示压缩完成的时间点与压缩开始的时间点。

5.3. 实验环境与参数设置

实验平台硬件配置为 16 核 Intel 酷睿 i9 12900K CPU、RTX3090-024G 独立显卡以及 64GB 内存；软件配置为 Ubuntu20.04 系统、Pytorch11.3 深度学习框架和 CUDA11.7。

训练迭代次数为 2 次，批大小为 2048，初始学习率设置为 0.001，进程数为 0，使用主线程，自适应矩估计作为优化器。

5.4. 对比实验分析

基于两日的用电采集数据，本文提出方法和两种传统压缩算法、两种深度学习压缩算法进行对比分析。本文方法和两种传统算法压缩率的对比结果如表 2 所示。表 2 结果表明，本文所提方法相较于算术编码和基于字典模型的 LZW 这两种传统无损压缩方法，压缩率分别平均提升约 28.8%和 7.8%。算术编码中符号出现概率是固定的，全局出现概率较低的字符在局部频繁出现，这种情况下算法为其分配较长

码字，算术编码没有利用重复模式，导致基于统计模型的算术编码压缩率最低。基于字典模型的 LZW 算法是一种自适应编码，根据局部字符出现频率调整其概率，可以有效处理以上情况，压缩率有所提升，但由于字典长度有限，无法进一步取得更优压缩率。本文提出的方法既解决了字符局部匹配问题，又带来预测精度的提升。

Table 2. Compression ratio of the proposed algorithm compared with that of two traditional algorithms
表 2. 提出算法和两种传统算法压缩率的对比结果

原始大小/	LZW [4]		算术编码[15]		本文方法	
kB	压缩大小/kB	压缩率/%	压缩大小/kB	压缩率/%	压缩大小/kB	压缩率/%
512,000	56,053	89.05	163,900	67.99	15,995	96.88
512,000	55,957	89.07	163,737	68.02	15,938	96.89

本文方法和两种深度学习算法压缩率和压缩时间的对比结果如表 3 所示。表 3 结果表明，本文提方法相较于 Cmix v19 和 NNCP 两种深度学习方法，平均压缩率分别降低 0.4%和 0.2%，但平均压缩时间分别约为其 5.1%和 39.4%。Cmix v19 是混合了多个不同尺度、不同方法模型的数据压缩器，在多个模型的庞大架构下，能取得最好的压缩效果，但压缩时间太长，难以应用在实际工程中。NNCP 采用基于 Transformer-xl 模型，transformer-xl 具有优秀的上下文预测能力，压缩率比本文方法略高；完全基于 transformer 架构计算量庞大，相比于本文方法，压缩率虽然高了 0.20%，但压缩时间是本文提出方法的 2.54 倍。本文提出方法采用双层多尺度架构，保持低复杂度的条件下充分利用多尺度信息进行预测，更好的平衡了压缩率和压缩时间，具有更强的实用性。

Table 3. Compression ratio and compression time of the proposed algorithm compared with that of two deep learning algorithms
表 3. 提出算法和两种传统算法压缩率的对比结果

原始大小/	Cmix v19 [17]		NNCP [16]		本文方法	
kB	压缩时间/s	压缩率/%	压缩时间/s	压缩率/%	压缩时间/s	压缩率/%
512,000	734847.74	97.93%	96029.94	97.69	37795.35	97.50
512,000	734242.43	97.96%	96537.69	97.71	38010.77	97.51

5.5. 训练加速实验对比

基于两日的用电采集数据，以是否采用训练加速策略为对比条件，计算不同条件下压缩率和训练时间，对比结果如表 4 所示。表 4 结果表明，若采取本文提出的训练加速方法，模型训练时间平均减少 15.06%，精度损失不到 0.01%。本文提出的训练加速方法能够有效减少无效的反向传播，提高模型训练速度。

Table 4. Effect of training acceleration strategy on compression ratio and training time
表 4. 训练加速策略对压缩率与训练时间的影响

训练集大小/kB	是否采用加速策略	压缩大小/kB	压缩率/%	训练时间/s
512,000	否	12,751	97.50	30316.80
512,000	是	12,808	97.50	25745.46
512,000	否	12,728	97.51	30532.77
512,000	是	12,682	97.52	25942.65

6. 结束语

本文结合 Transformer、BGRU 和自适应算术编码，提出一种基于多头注意力与双向门控循环单元的数据无损压缩方法。实验结果表明，提出的算法相较于算术编码和基于字典模型的 LZW 这两种传统无损压缩方法具有更高的压缩率，相较于 Cmix v19 和 NNCP 两种深度学习方法，节省了压缩时间。

参考文献

[1] 嵇月强. 工业历史数据库的研究[D]: [硕士学位论文]. 杭州: 浙江大学, 2007.

[2] 刘媛媛, 何文春, 王妍, 李江涛, 白金婷. 气象大数据云平台归档系统设计及实现[J]. 气象科技, 2021, 49(5): 697-706.

[3] 赵力帅, 李耕宇, 武振宇, 刘鹏. 基于分层异构数据库系统的历史数据归档技术[J]. 电信科学, 2018, 34(S1): 174-178.

[4] 王平. LZW 无损压缩算法的实现与研究[J]. 计算机工程, 2002(7): 98-99+150.

[5] 徐慧. 实时数据库中数据压缩算法的研究[D]: [硕士学位论文]. 杭州: 浙江大学, 2006.

[6] 于恒, 梅红岩, 许晓明, 等. 基于深度学习的图像压缩算法研究综述[J]. 计算机工程与应用, 2020, 56(15): 15-23.

[7] 励益韬, 孙未未. 基于循环神经网络的轨迹压缩算法[J]. 计算机科学, 2020, 47(10): 102-107.

[8] 刘文顺. 基于字节对编码的无损压缩算法[D]: [硕士学位论文]. 杭州: 浙江大学, 2023.

[9] 苏月明. 基于深度学习的压缩成像算法研究[D]: [博士学位论文]. 秦皇岛: 燕山大学, 2022.

[10] 闫赛然. 基于深度学习的压缩感知磁共振重建[D]: [硕士学位论文]. 北京: 北京交通大学, 2022.

[11] 任欢, 王旭光. 注意力机制综述[J]. 计算机应用, 2021, 41(S1): 1-6.

[12] 尹航, 范文婷. 基于 Transformer 目标检测研究综述[J]. 现代信息科技, 2021, 5(7): 14-17.

[13] 牛哲文, 余泽远, 李波, 唐文虎. 基于深度门控循环单元神经网络的短期风功率预测模型[J]. 电力自动化设备, 2018, 38(5): 36-42.

[14] 桑海峰, 陈紫珍. 基于双向门控循环单元的 3D 人体运动预测[J]. 电子与信息学报, 2019, 41(9): 2256-2263.

[15] 李雷定, 马铁华, 尤文斌. 常用数据无损压缩算法分析[J]. 电子设计工程, 2009, 17(1): 49-50+53.

[16] Bellard, F. (2019) Lossless Data Compression with Neural Networks. https://bellard.org/nncp/nncp.pdf

[17] Knoll, B. (2014) CMIX. https://www.byronknoll.com/cmix.html

[18] 吴晓云. 算术编码算法在图像压缩中的研究[J]. 计算机与数字工程, 2017, 45(9): 1863-1865.

[19] 刘尧, 蒋林, 李远成, 山蕊. 自适应二进制算术编码的动态可重构实现研究[J]. 电子测量技术, 2022, 45(19): 50-55.