ICS 35.040
CCS L 71

NATIONAL STANDARD OF THE PEOPLE'S REPUBLIC OF CHINA

GB/T 42382.2-2026

Information technology - Neural network representation and model compression - Part 2: Large scale pre-trained models

信息技术 神经网络表示与模型压缩 第2部分：大规模预训练模型

Issued on: April 30, 2026 **Implemented on: November 1, 2026**

Issued by: State Administration for Market Regulation;
Standardization Administration of the PRC

Table of Contents

4 Abbreviations

The following abbreviations apply to this document. AI. Artificial Intelligence BERT. Bidirectional Encoder Representations from Transformers formers) FFN. Feedforward Network GAN. Generative Adversarial Network GPT. Generative Pre-trained Transformer HTTPS. Hyper-Text Transfer Protocol Secure LLM. Large Language Model MHA. Multi-Head Attention MLP. Multi-Layer Perception MSA. Multi-Head Self-Attention RAG. Retrieval-Augmented Generation REST. Representational State Transfer SGD. Stochastic Gradient Descent VAE. Variational Auto-Encoder ViT. Vision Transformer

5 Overview

Large-scale pre-trained models are interconnected in terms of representation, compression and adaptation, transmission and distribution, forming a complete ecosystem. The various stages are closely interconnected and span the entire lifecycle from model training to application. The overall architecture of each stage is shown in Figure 1.

6.1 Syntax Description

6.1.1 Principles This document defines the syntax for representing large-scale pre-trained models, from coarse-grained to fine-grained, that is, from model structure definition, computation graph definition, to... Node definitions, nested layer by layer, construct the

basic grammatical description of the entire large-scale pre-trained model. This representation grammar should be defined by a specific computing system. This is completed within a deep learning platform and related hardware and software, following these principles.

a) When implementing the computing system, it is necessary to adjust the syntactic elements according to actual needs, including but not limited to. 1) Keyword (parameter) naming; 2) Operator naming; 3) Data type.

b) The definition of necessary hierarchical levels needs to be considered, including. Model structure definition; 2) Definition of computational graph; 3) Basic data type definitions

6.1.2 Model Structure Definition The model structure represents the basic information and network architecture of the neural network model. The technical parameters describing the model structure are shown in Table 1.

6.1.4 Operation Node Definition The operation node definitions are shown in Table

3.The operation field defines the name of the operation node; input and output are the output parameters of the operation node. Input variable nodes and output variable nodes; attribute is the attribute of the operation node.

6.2.1 Representation of Operations

6.2.1.1 General Rules 6.2.1.1.1 Principles for Using Calculation Operations The computational operations contained in large-scale pre-trained models are implemented by specific computing systems and should follow the following usage principles.

a) Make adjustments as needed, including but not limited to. 1) Keyword (parameter) naming; 2) Operator naming; 3) Data type support range

b) Consider the elements involved in the operation, including. 1) Supported parameters; 2) Supported types. 6.2.1.1.2 Classification of Operations Operators for large-scale pre-trained models can be divided into the following three categories.

a) General operators for large-scale pre-trained model networking, including but not limited to embedding, layer normalization, linearity, and attention. Operators, whose definitions are shown in Tables 4 to 7.

b) General operators for neural network models, including but not limited to basic mathematical operators such as reshape and concat, and neural network operators. For detailed definitions of ReLU and Softmax, please refer to other relevant operator standards; they will not be repeated in this document. See also [link to relevant documentation]. GB/T 42382.1-2023

c) Communication operators for training large-scale pre-trained models, including but not limited to reduce, allreduce, reduce_scatter, etc. The operators allgather, broadcast, send,

and recv are defined as shown in Tables 8 to 14.

6.2.1.2 Definition of Basic Operations This section defines common computational operations used in large-scale pre-trained models. The specific definitions are as follows:

6.2.3 Multimodal Operation Representation

6.2.3.1 Overview 6.2.3.1.1 Core Components Multimodal large-scale pre-trained models are large models that integrate data from multiple modalities for understanding and generation. Their core components are... It includes an input layer, a modality-specific encoder, a modality fusion module, a modality alignment module, a decoding layer, and an output layer. 6.2.3.1.2 Input Layer Different modal inputs correspond to unique input processing modules. The data input processing module includes.

a) Text data. word segmentation and embedding processing;

b) Image data. Pixel value normalization and image enhancement.

c) Audio data. Convert to a spectrogram or other time-frequency representation. 6.2.3.1.3 Modality-Specific Encoder Modality-specific encoders include, but are not limited to.

a) Text encoder. Typically consists of a word embedding layer, a position embedding layer, and multiple Transformer encoders, capable of converting text data... Transformed into high-dimensional feature representations, including BERT, GPT, RoBERTa, etc.;

b) Image encoder. Typically contains convolutional layers (or self-attention layers) and pooling layers, capable of extracting multi-level features of the image, including... ResNet, Vision Transformer (ViT), etc.;

c) Audio encoder. Typically processes audio signals using convolutional layers or Transformer structures to extract meaningful audio features. Including Wav2Vec, Mel-Spectrogram, MFCC, etc.;

d) Video encoder. Typically, it extracts rich video features by processing temporal and spatial information in video data, including... C3D, I3D, etc. 6.2.3.1.4 Modal Fusion Module Multimodal data fusion strategies include.

a) Early fusion. Fusion is performed at the input layer or coding layer. Common methods include concatenation and weighted averaging.

b) Intermediate Fusion. Fusion is performed in the intermediate layers of the model, typically using a multi-layer cross-attention mechanism at different... Information exchange between modalities;

c) Late-stage fusion. Fusion is performed at higher levels of the model or at the output layer, typically using a separate decision module to process the data for each modality. Process the data and then merge the results. 6.2.3.1.5 Modal Alignment Module Modality alignment refers to mapping data from different modalities to the same feature space. Alignment

methods include, but are not limited to.

a) Co-occurrence analysis. Aligning data from different modalities using co-occurrence matrices or co-occurrence graphs.

b) Contrastive learning. By constructing positive and negative sample pairs, the similarity between features of the same modality and different modalities is maximized;

c) Alignment transformation. Using linear or nonlinear transformations, features of different modalities are aligned to the same representation space. 6.2.3.1.6 Decoder and Output Layer Decoders and output layers include, but are not limited to.

7 Compressed representation of large-scale pre-trained models

7.1 Overview Large-scale pre-trained models based on Transformer should include word vectors, multiple Transformer modules (each Transformer...). The module includes multi-head attention mechanisms, fully connected networks, regularization, skip connections, and task-related modules. This model's inference phase... The bottleneck lies in storing word vectors and weights in the Transformer module, while the bottleneck in computing the matrix lies in the multi-head attention mechanism and the fully connected network. Multiplication. Storage and computation directly impact inference latency and power consumption. To reduce storage and computation during inference, many methods... Word vectors and weight matrices have been compressed, or matrix multiplication has been accelerated and optimized. These are used for compressing word vectors and weight matrices. Methods for reducing, accelerating, and optimizing matrix multiplication include, but are not limited to.

a) Optimization of large-scale pre-trained model structure. reducing model complexity and computational requirements by adjusting the model structure;

b) Accelerated Compression of Large-Scale Pre-trained Models. This involves accelerating model inference and training through algorithmic and hardware optimizations, while minimizing compression. Storage requirements mainly include sparsity and quantization;

c) Large-scale pre-trained model transfer compression. By combining transfer learning and model compression, the complexity of the model on new tasks is reduced. Complexity and size, specifically including multimodal transfer and multitasking transfer.

7.2 Optimization of Large-Scale Pre-trained Model Structure

7.2.1 Nested Structure Nested structures utilize an outer Transformer and an inner Transformer to extract global and local features, respectively. The approach involves using an outer Transformer module to model the relationships between image patches, and an inner Transformer module to model the relationships between them. A technical solution for constructing rich visual representations is achieved by modeling the relationships between