International Standard

ISO/IEC 15938-17

Information technology — Multimedia content description interface —

Part 17: **Compression of neural networks for multimedia content description and analysis**

Second edition
2024-01

Technologies de l'information — Interface de description du contenu multimédia —
Partie 17: Compression des réseaux neuronaux pour la description et l'analyse du contenu multimédia

Reference number
ISO/IEC 15938-17:2024(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This second edition cancels and replaces the first edition (ISO/IEC 15938-17:2022), which has been technically revised.

The main changes are as follows:

- Support for incremental compression of updates of neural networks respective to a base model,
- Additional sparsification tools,
- Additional entropy coding tools, leveraging dependencies in incremental updates,
- Additional quantization tools, including representation as residuals of updates, and
- Additional high-level syntax, covering the new coding tools as well as more metadata (e.g. performance metrics).

A list of all parts in the ISO/IEC 15938 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

Artificial neural networks have been adopted for a broad range of tasks in multimedia analysis and processing, media coding, data analytics and many other fields. Their recent success is based on the feasibility of processing much larger and complex neural networks (deep neural networks, DNNs) than in the past, and the availability of large-scale training data sets. As a consequence, trained neural networks contain a large number of parameters and weights, resulting in a quite large size (e.g. several hundred MBs). Many applications require the deployment of a particular trained network instance, potentially to a larger number of devices, which may have limitations in terms of processing power and memory (e.g. mobile devices or smart cameras), and also in terms of communication bandwidth. Any use case, in which a trained neural network (or its updates) needs to be deployed to a number of devices thus benefits from a standard for the compressed representation of neural networks.

Considering the fact that compression of neural networks is likely to have a hardware dependent and hardware independent component, this document is designed as a toolbox of compression technologies. Some of these technologies require specific representations in an exchange format (i.e. sparse representations, adaptive quantization), and thus a normative specification for representing outputs of these technologies is defined. Others do not at all materialize in a serialized representation (e.g. pruning), however, also for the latter ones required metadata is specified. This document is independent of a particular neural network exchange format, and interoperability with common formats is described in the annexes.

This document thus defines a high-level syntax that specifies required metadata elements and related semantics. In cases where the structure of binary data is to be specified (e.g. decomposed matrices) this document also specifies the actual bitstream syntax of the respective block. Annexes to the document specify the requirements and constraints of compressed neural network representations; as defined in this document; and how they are applied.

- Annex A specifies the implementation of this document with the Neural Network Exchange Format (NNEF[1]), defining the use of NNEF to represent network topologies in a compressed neural network bitstream.
- Annex B provides recommendations for the implementation of this document with the Open Neural Network Exchange Format (ONNX®)[2], defining the use of ONNX to represent network topologies in a compressed neural network bitstream.
- Annex C provides recommendations for the implementation of this document with the PyTorch®[3] format, defining the reference to PyTorch elements in the network topology description of a compressed neural network bitstream.
- Annex D provides recommendations for the implementation of this document with the Tensorflow®[4] format, defining the reference to Tensorflow elements in the network topology description of a compressed neural network bitstream.
- Annex E provides recommendations for the carriage of tensors compressed according to this document in third party container formats.
- Annex F provides recommendations for the naming of common performance metrics to specify the metric that was used for validation.

[1] NNEF is the trademark of a product owned by The Khronos® Group. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO/IEC of the product named.

[2] ONNX is the trademark of a product owned by LF PROJECTS, LLC. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO/IEC of the product named.

[3] PyTorch is the trademark of a product supplied by Facebook, Inc. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO/IEC of the product named.

[4] TensorFlow is the trademark of a product supplied by Google LLC. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO/IEC of the product named.

— Annex G provides recommendations for implementing the encoding side of some of the compression tools.

The compression tools described in this document have been selected and evaluated for neural networks used in applications for multimedia description, analysis and processing. However, they may be useful for the compression of neural networks used in other applications and applied to other types of data.

Information technology — Multimedia content description interface —

Part 17: Compression of neural networks for multimedia content description and analysis

1 Scope

This document specifies Neural Network Coding (NNC) as a compressed representation of the parameters/weights of a trained neural network and a decoding process for the compressed representation, complementing the description of the network topology in existing (exchange) formats for neural networks. It establishes a toolbox of compression methods, specifying (where applicable) the resulting elements of the compressed bitstream. Most of these tools can be applied to the compression of entire neural networks, and some of them can also be applied to the compression of differential updates of neural networks with respect to a base network. Such differential updates are for example useful when models are redistributed after fine-tuning or transfer learning, or when providing versions of a neural network with different compression ratios.

This document does not specify a complete protocol for the transmission of neural networks, but focuses on compression of network parameters. Only the syntax format, semantics, associated decoding process requirements, parameter sparsification, parameter transformation methods, parameter quantization, entropy coding method and integration/signalling within existing exchange formats are specified, while other matters such as pre-processing, system signalling and multiplexing, data loss recovery and post-processing are considered to be outside the scope of this document. Additionally, the internal processing steps performed within a decoder are also considered to be outside the scope of this document; only the externally observable output behaviour is required to conform to the specifications of this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 10646, *Information technology — Universal coded character set (UCS)*

ISO/IEC 60559, *Information technology — Microprocessor Systems — Floating-Point arithmetic*

IETF RFC 1950, *ZLIB Compressed Data Format Specification version 3.3*

NNEF-v1.0.3[5], *Neural Network Exchange Format*, The Khronos NNEF Working Group, Version 1.0.3, 2020

FIPS PUB 180-4:2015, *Secure Hash Standard (SHS)*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

[5] Available from: https://www.khronos.org/registry/NNEF/specs/1.0/nnef-1.0.3.pdf

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at https://www.iso.org/obp
- IEC Electropedia: available at https://www.electropedia.org/

3.1
aggregate NNR unit
NNR unit which carries multiple NNR units in its payload

3.2
base neural network
neural network serving as reference for a differential update

3.3
compressed neural network representation
NNR
representation of a neural network with model parameters encoded using compression tools

3.4
decomposition
transformation to express a tensor as product of two tensors

3.5
hyperparameter
parameter whose value is used to control the learning process

3.6
layer
collection of nodes operating together at a specific depth within a neural network

3.7
model parameter
coefficients of the neural network model such as weights and biases

3.8
NNR unit
data structure for carrying (compressed or uncompressed) neural network data and related metadata

3.9
parameter identifier
value that uniquely identifies a parameter throughout different incremental updates

Note 1 to entry: Parameters having the same parameter identifier are at the same position in the same tensor in different incremental updates. This means they are co-located.

3.10
pruning
reduction of parameters in (a part of) the neural network

3.11
sparsification
increase of the number of zero-valued entries of a tensor

3.12
tensor
multidimensional structure grouping related model parameters

3.13
updated neural network
neural network resulting from modifying the base neural network

Note 1 to entry: The updated neural network is reconstructed by applying the differential update to the base neural network.

4 Abbreviated terms, conventions and symbols

4.1 General

This subclause contains the definition of operators, notations, functions, textual conventions and processes used throughout this document.

The mathematical operators used in this document are similar to those used in the C programming language. However, the results of integer division and arithmetic shift operations are specified more precisely, and additional operations are specified, such as exponentiation and real-valued division. Numbering and counting conventions generally begin from 0, e.g. "the first" is equivalent to the 0-th, "the second" is equivalent to the 1-th, etc.

4.2 Abbreviated terms

DeepCABAC	Context-adaptive binary arithmetic coding for deep neural networks
LDR	Low displacement rank
LPS	Layer parameter set
LR	Low-rank
LSA	Local scaling adaptation
LSB	Least significant bit
MPS	Model parameter set
MSB	Most significant bit
MSE	Mean square error
NN	Neural network
NNC	Neural network coding
NDU	NNR compressed data unit
NNEF	Neural network exchange format
QP	Quantization parameter
PRE	Predictive residual encoding
SBT	Stochastic binary-ternary quantization
SVD	Singular value decomposition

4.3 List of symbols

This document defines the following symbols:

A	Input tensor
B	Output tensor
B_{jl}^{k}	Block in superblock *j* of layer *k*.
b	Bias parameter
C_{i}	Number of input channels of a convolutional layer
C_{o}	Number of output channels of a convolutional layer
c_{j}^{k}	Number of channels in dimension *j* of tensor in layer *k*
$c_{j}^{k'}$	Derived number of channels in dimension *j* of tensor in layer *k*
d_{j}^{k}	Depth dimension of tensor at layer *k*
e	Parameter of f-circulant matrix Z_e
F	Parameter tensor of a convolutional layer
f	Parameter of f-circulant matrix *Zf*
G_k	Left-hand side matrix of Low Rank decomposed representation of matrix W_k
H_k	Right-hand side matrix of Low Rank decomposed representation of matrix W_k
h_{j}^{k}	Height dimension of tensor for layer *k*
K	Dimension of a convolutional kernel
L	Loss function
L_{c}	Compressibility loss
L_{d}	Diversity loss
L_{s}	Task loss
L_{t}	Training loss
M	Feature matrix
M_k	Pruning mask for layer *k*
m	Sparsification hyperparameter
m_i	*i*-th row of feature matrix M
n_{j}^{k}	Kernel size of tensor at layer *k*.
n^{k}	Dimension resulting from a product over n_{j}^{k}
P	Stochastic transition matrix
p	Pruning ratio hyperparameter
p_{ij}	Elements of transition matrix P
q	Sparsification ratio hyperparameter

q_b	Binary quantization
q_t	Ternary quantization
S	Importance of parameters for pruning
S_j^k	Superblock j in layer k
s	Local scaling factors
s_j^k	Size of superblock j in layer k
T	Topology element
T^q	Quantizable topology element
u	Unification ratio hyperparameter
W	Parameter tensor
ΔW	Difference of parameter tensor
W_l	Weight tensor of l-th layer
W_k	Parameter tensor of layer k
$\hat{W}_k$	Low Rank approximation of W_k
w	Parameter vector
$w_{l,i}$	Vector of weights for the i-th filter in the l-th layer
$w'_{l,i}$	Vector of normalized weights for the i-th filter in the l-th layer
y, y_{ref}	Coding performance, reference coding performance
y_d	Coding performance difference
X	Input to a batch-normalization layer
Z_e	f-circulant matrix
Z_f	f-circulant matrix
α	Folded batch normalization parameter
α'	Combined value for folded batch normalization parameter and local scaling factors
β	Batch normalization parameter
β_u	Updated batch normalization parameter
γ_c	Compressibility loss multiplier
γ	Batch normalization parameter
γ_u	Updated batch normalization parameter
δ	Folded batch normalization parameter
δ_f	Sparsification threshold (mean of filter means)

δ_s	Scaling factor for sparsification
ϵ	Scalar close to zero to avoid division by zero in batch normalization
λ	Eigenvector
λ_c	Compressibility loss weight
λ_d	Diversity loss weight
μ	Batch normalization parameter
v_j^k	Width dimension of tensor for layer *k.*
π	Equilibrium probability of P
π_t	Probability of applying ternary quantization
ρ	Parameter
σ	Batch normalization parameter
τ	Threshold (sparsification, ternary-binary quantization)
θ_ρ	Weight magnitude threshold
φ	Smoothing factor

4.4 Number formats and computation conventions

This document defines the following number formats:

integer	Integer number which may be arbitrarily small or large. Integers are also referred to as signed integers.
unsigned integer	Unsigned integer that may be zero or arbitrarily large.
float	Floating point number according to ISO/IEC 60559.

If not specified otherwise, outcomes of all operators and mathematical functions are mathematically exact. Whenever an outcome shall be a float, it is explicitly specified.

4.5 Arithmetic operators

The following arithmetic operators are defined:

+	Addition
−	Subtraction (as a two-argument operator) or negation (as a unary prefix operator)
*	Multiplication, including matrix multiplication
$\circ$	Element-wise multiplication of two transposed vectors or element-wise multiplication of a transposed vector with rows of a matrix or Hadamard product of two matrices with identical dimensions
x^y	Exponentiation. Specifies *x* to the power of *y*. In other contexts, such notation is used for superscripting not intended for interpretation as exponentiation.