INTERNATIONAL STANDARD

ISO/IEC 15938-17

First edition
2022-08

Information technology — Multimedia content description interface —

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

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 15938-17:2022(E)

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 v

Introduction vi

1 Scope 1

2 Normative references 1

3 Terms and definitions 1

4 Abbreviated terms, conventions and symbols 2

4.1 General 2

4.2 Abbreviated terms 2

4.3 List of symbols 3

4.4 Number formats and computation conventions 5

4.5 Arithmetic operators 5

4.6 Logical operators 6

4.7 Relational operators 6

4.8 Bit-wise operators 6

4.9 Assignment operators 7

4.10 Range notation 7

4.11 Mathematical functions 7

4.12 Array functions 8

4.13 Order of operation precedence 9

4.14 Variables, syntax elements and tables 10

5 Overview 11

5.1 General 11

5.2 Compression tools 11

5.3 Creating encoding pipelines 12

6 Syntax and semantics 13

6.1 Specification of syntax and semantics 13

6.1.1 Method of specifying syntax in tabular form 13

6.1.2 Bit ordering 14

6.1.3 Specification of syntax functions and data types 14

6.1.4 Semantics 16

6.2 General bitstream syntax elements 17

6.2.1 NNR unit 17

6.2.2 Aggregate NNR unit 17

6.2.3 Composition of NNR bitstream 18

6.3 NNR bitstream syntax 18

6.3.1 NNR unit syntax 18

6.3.2 NNR unit size syntax 19

6.3.3 NNR unit header syntax 19

6.3.4 NNR unit payload syntax 24

6.3.5 Byte alignment syntax 29

6.4 Semantics 29

6.4.1 General 29

6.4.2 NNR unit size semantics 29

6.4.3 NNR unit header semantics 29

6.4.4 NNR unit payload semantics 36

7 Decoding process 41

7.1 General 41

7.2 NNR decompressed data formats 42

7.3 Decoding methods 42

7.3.1 General 42

7.3.2 Decoding method for NNR compressed payloads of type NNR_PT_INT 43

7.3.3 Decoding method for NNR compressed payloads of type NNR_PT_FLOAT 43

7.3.4 Decoding method for NNR compressed payloads of type NNR_PT_RAW_FLOAT 43
7.3.5 Decoding method for NNR compressed payloads of type NNR_PT_BLOCK 43
7.3.6 Decoding process for an integer weight tensor 45

8 Parameter reduction 46
8.1 General 46
8.2 Methods 46
8.2.1 Sparsification using compressibility loss 46
8.2.2 Sparsification using micro-structured pruning 46
8.2.3 Combined pruning and sparsification 47
8.2.4 Parameter unification 49
8.2.5 Low rank/low displacement rank for convolutional and fully connected layers 50
8.2.6 Batchnorm folding 50
8.2.7 Local scaling adaptation 51
8.3 Syntax and semantics 52
8.3.1 Sparsification using compressibility loss 52
8.3.2 Sparsification using micro-structured pruning 52
8.3.3 Combined pruning and sparsification 52
8.3.4 Weight unification 53
8.3.5 Low rank/low displacement rank for convolutional and fully connected layers 53
8.3.6 Batchnorm folding 53
8.3.7 Local scaling 54

9 Parameter quantization 54
9.1 Methods 54
9.1.1 Uniform quantization method 54
9.1.2 Codebook-based method 54
9.1.3 Dependent scalar quantization method 54
9.2 Syntax and semantics 54
9.2.1 Uniform quantization method 54
9.2.2 Codebook-based method 55
9.2.3 Dependent scalar quantization method 55

10 Entropy coding 55
10.1 Methods 55
10.1.1 DeepCABAC 55
10.2 Syntax and semantics 56
10.2.1 DeepCABAC syntax 56
10.3 Entropy decoding process 59
10.3.1 General 59
10.3.2 Initialization process 60
10.3.3 Binarization process 61
10.3.4 Decoding process flow 61

Annex A (normative) **Implementation for NNEF 67**

Annex B (informative) **Implementation for ONNX® 69**

Annex C (informative) **Implementation for PyTorch® 71**

Annex D (informative) **Implementation for TensorFlow® 73**

Annex E (informative) **Recommendation for carriage of NNR bitstreams in other containers 75**

Bibliography 77

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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

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*.

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

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

[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.

be useful for the compression of neural networks used in other applications and applied to other types of data.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of patents.

ISO and IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and IEC. Information may be obtained from the patent database available at www.iso.org/patents.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

INTERNATIONAL STANDARD **ISO/IEC 15938-17:2022(E)**

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.

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, 1996*

NNEF-v1.0.3, Neural Network Exchange Format, The Khronos NNEF Working Group, Version 1.0.3, 2020-06-12 (https://www.khronos.org/registry/NNEF/specs/1.0/nnef-1.0.3.pdf)

3 Terms and definitions

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

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

3.3
decomposition
transformation to express a tensor as product of two tensors

3.4
hyperparameter
parameter whose value is used to control the learning process

3.5
layer
collection of nodes operating together at a specific depth within a neural network

3.6
model parameter
coefficients of the neural network model such as weights and biases

3.7
NNR unit
data structure for carrying (compressed or uncompressed) neural network data and related metadata

3.8
pruning
reduction of parameters in (a part of) the neural network

3.9
sparsification
increase of the number of zero-valued entries of a tensor

3.10
tensor
multidimensional structure grouping related model parameters

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
LSB	Least significant bit
MPS	Model parameter set
MSB	Most significant bit
NN	Neural network
NNEF	Neural network exchange format
NNC	Neural network coding
NNR	Compressed neural network representation
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 of tensor in dimension *j* and in layer *k*
$c_j^{k'}$	Derived number of channels of tensor in dimension *j* and 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 a product of n_j^k
P	Stochastic transition matrix
p	Pruning ratio hyperparameter
p_{ij}	Elements of transition matrix P
q	Sparsification ratio hyperparameter
S	Importance of parameters for pruning
S_j^k	Superblock in layer *k*
s	Local scaling factors
s_j^k	Size of superblock in layer *k*
u	Unification ratio hyperparameter
W	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
v_j^k	Width dimension of tensor for layer *k*.
$w_{l,i}$	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
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
γ_c	Compressibility loss multiplier
γ	Batch normalization parameter
δ	Folded batch normalization parameter
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
π	Equilibrium probability of P
τ	Sparsification pruning 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.
/	Integer division with truncation of the result toward zero. For example, 7 / 4 and −7 / −4 are truncated to 1 and −7 / 4 and 7 / −4 are truncated to −1.

÷ Used to denote division in mathematical equations where no truncation or rounding is intended.

$\frac{x}{y}$ Used to denote division in mathematical equations where no truncation or rounding is intended, including element-wise division of two transposed vectors or element-wise division of a transposed vector with rows of a matrix.

$\sum_{i=x}^{y} f(i)$ The summation of *f*(*i*) with *i* taking all integer values from *x* up to and including *y*.

$\prod_{i=x}^{y} f(i)$ The product of *f*(*i*) with *i* taking all integer values from *x* up to and including *y*.

x % *y* Modulus. Remainder of *x* divided by *y*, defined only for integers *x* and *y* with $x \geq 0$ and $y > 0$.

4.6 Logical operators

The following logical operators are defined:

x && *y*	Boolean logical "and" of *x* and *y*
x || *y*	Boolean logical "or" of *x* and *y*
!	Boolean logical "not"
x ? *y* : *z*	If *x* is TRUE or not equal to 0, evaluates to the value of *y*; otherwise, evaluates to the value of z.

4.7 Relational operators

The following relational operators are defined as follows:

>	Greater than
≥	Greater than or equal to
<	Less than
≤	Less than or equal to
==	Equal to
!=	Not equal to

When a relational operator is applied to a syntax element or variable that has been assigned the value "na" (not applicable), the value "na" is treated as a distinct value for the syntax element or variable. The value "na" is considered not to be equal to any other value.

4.8 Bit-wise operators

The following bit-wise operators are defined as follows:

& Bit-wise "and". When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

$|$ Bit-wise "or". When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

^ Bit-wise "exclusive or". When operating on integer arguments, operates on a two's complement representation of the integer value. When operating on a binary argument that contains fewer bits than another argument, the shorter argument is extended by adding more significant bits equal to 0.

$x >> y$ Arithmetic right shift of a two's complement integer representation of *x* by *y* binary digits. This function is defined only for non-negative integer values of *y*. Bits shifted into the MSBs as a result of the right shift have a value equal to the MSB of *x* prior to the shift operation.

$x << y$ Arithmetic left shift of a two's complement integer representation of *x* by *y* binary digits. This function is defined only for non-negative integer values of *y*. Bits shifted into the LSBs as a result of the left shift have a value equal to 0.

! Bit-wise not operator returning 1 if applied to 0 and 0 if applied to 1.

4.9 Assignment operators

The following arithmetic operators are defined as follows:

= Assignment operator

++ Increment, i.e., $x{+}{+}$ is equivalent to $x = x + 1$; when used in an array index, evaluates to the value of the variable prior to the increment operation.

−− Decrement, i.e., $x{-}{-}$ is equivalent to $x = x - 1$; when used in an array index, evaluates to the value of the variable prior to the decrement operation.

+= Increment by amount specified, i.e., x += 3 is equivalent to $x = x + 3$, and x += (−3) is equivalent to $x = x + (-3)$.

−= Decrement by amount specified, i.e., x −= 3 is equivalent to $x = x - 3$, and x −= (−3) is equivalent to $x = x - (-3)$.

4.10 Range notation

The following notation is used to specify a range of values:

$x = y..z$ *x* takes on integer values starting from *y* to *z*, inclusive, with *x*, *y*, and *z* being integer numbers and *z* being greater than y.

array[*x*, *y*] a sub-array containing the elements of array comprised between position *x* and *y* included. If *x* is greater than *y*, the resulting sub-array is empty.

4.11 Mathematical functions

The following mathematical functions are defined:

Ceil(*x*) the smallest integer greater than or equal to *x*

Floor(*x*) the largest integer less than or equal to x

Log2(*x*) the base-2 logarithm of *x*