International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

F.748.12

(06/2021)

SERIES F: NON-TELEPHONE TELECOMMUNICATION SERVICES

Multimedia services

Deep learning software framework evaluation methodology

Recommendation ITU-T F.748.12

ITU-T F-SERIES RECOMMENDATIONS

NON-TELEPHONE TELECOMMUNICATION SERVICES

TELEGRAPH SERVICE	
Operating methods for the international public telegram service	F.1–F.19
The gentex network	F.20–F.29
Message switching	F.30–F.39
The international telemessage service	F.40–F.58
The international telex service	F.59–F.89
Statistics and publications on international telegraph services	F.90–F.99
Scheduled and leased communication services	F.100–F.104
Phototelegraph service	F.105–F.109
MOBILE SERVICE	
Mobile services and multidestination satellite services	F.110–F.159
TELEMATIC SERVICES	
Public facsimile service	F.160–F.199
Teletex service	F.200–F.299
Videotex service	F.300–F.349
General provisions for telematic services	F.350–F.399
MESSAGE HANDLING SERVICES	F.400–F.499
DIRECTORY SERVICES	F.500–F.549
DOCUMENT COMMUNICATION	
Document communication	F.550–F.579
Programming communication interfaces	F.580–F.599
DATA TRANSMISSION SERVICES	F.600–F.699
MULTIMEDIA SERVICES	**F.700–F.799**
ISDN SERVICES	F.800–F.849
UNIVERSAL PERSONAL TELECOMMUNICATION	F.850–F.899
ACCESSIBILITY AND HUMAN FACTORS	F.900–F.999

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T F.748.12

Deep learning software framework evaluation methodology

Summary

A deep learning software framework provides an easy and fast way for manufactures to develop their own artificial intelligence (AI) applications. However, different frameworks show different performances under different scenarios. Recommendation ITU-T F.748.12 helps to evaluate deep learning software frameworks to help manufactures take full advantage of certain frameworks and avoid the disadvantages of others.

History

Edition	Recommendation	Approval	Study Group	Unique ID[*]
1.0	ITU-T F.748.12	2021-06-13	16	11.1002/1000/14681

Keywords

Deep learning software framework, ease of use, ecological construction, evaluation, performance, security and stability, supported architecture, underlying optimization.

[*] To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Page

Recommendation ITU-T F.748.12

Deep learning software framework evaluation methodology

1 Scope

This Recommendation provides the evaluation methodology for deep learning software framework. It addresses the following subjects:

a) Ecological construction of deep learning software framework;

b) Ease of use of deep learning software framework;

c) Performance of deep learning software framework;

d) The supported architecture of deep learning software framework;

e) Underlying optimization of deep learning software framework;

f) Security and stability of deep learning software framework.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation does not use any terms defined elsewhere.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 deep learning: A representation learning method, used to model high-level abstractions in data through the use of model architectures, which are composed of multiple nonlinear transformations.

3.2.2 deep learning software framework: A tool that uses a set of pre-built and optimized components to define a model to achieve the encapsulation of artificial intelligence algorithms, data calls, and the use of computing resources.

3.2.3 deep learning model: A deep learning algorithm used to solve a specific task, usually refers to the computational graph structure information and parameter information used to represent the deep learning algorithm.

3.2.4 model compression: A mechanism to reduce the size of the model through algorithms such as pruning, quantification, regularization, knowledge distillation, and conditional calculation.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ASIC	Application-specific integrated circuit

CPU	Central processing unit
FPGA	Field-programmable gate array
GPU	Graphics processing unit
LLVM	Low level virtual machine
NNEF	Neural network exchange format
ONNX	Open neural network exchange
SGD	Stochastic gradient descent

5 Conventions

The following conventions are used in this Recommendation:

- The keywords "is required to" indicate a requirement that must be strictly followed and from which no deviation is permitted if conformance to this Recommendation is to be claimed.
- The keywords "is recommended" indicate a requirement that is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

6 Industrial realization of deep learning software framework in artificial intelligence

6.1 Application architecture based on deep learning software framework

Figure 6-1 illustrates the architecture of applications based on the deep learning software framework.

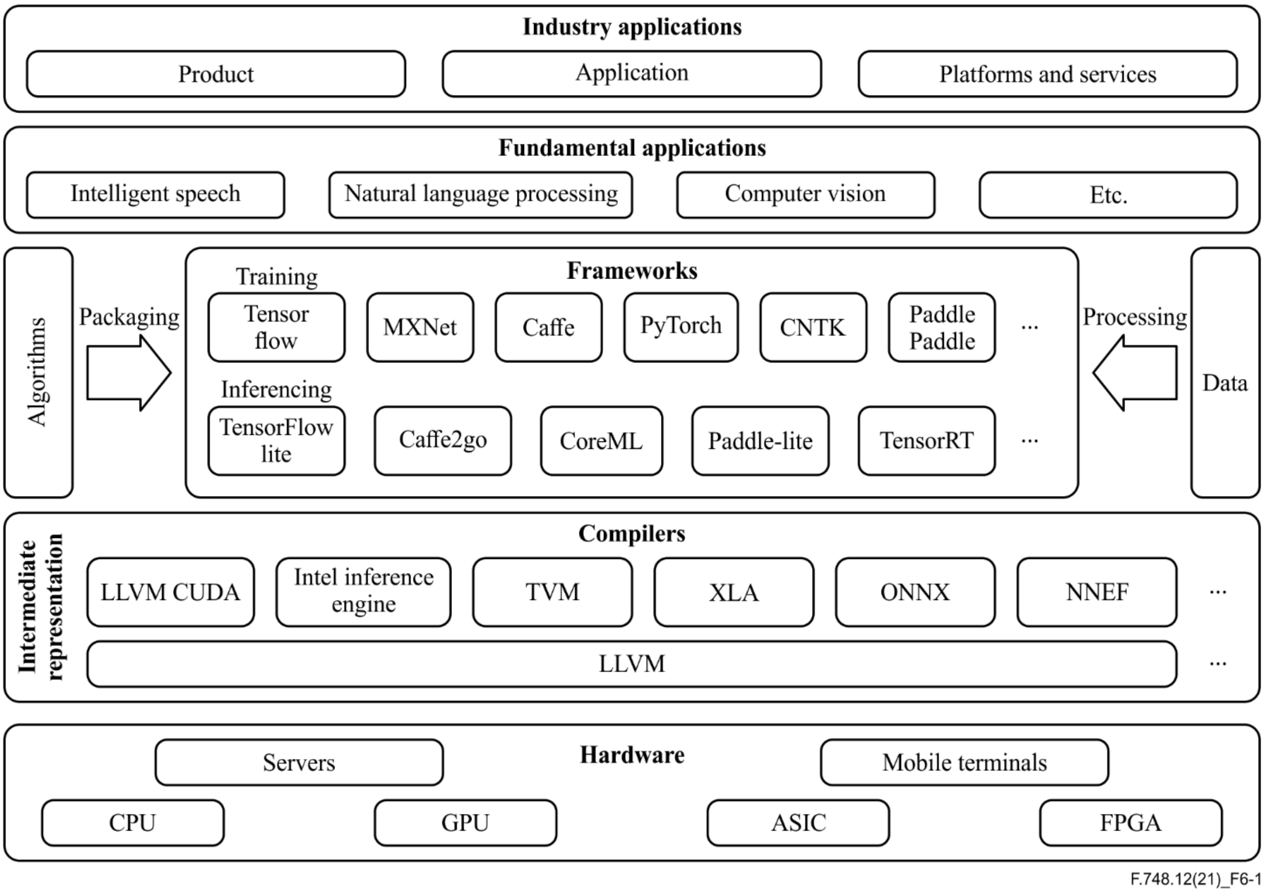

Figure 6-1 – Architecture of applications based on deep learning software framework

6.2 Hardware

Hardware includes central processing unit (CPU), graphics processing unit (GPU), application-specific integrated circuit (ASIC), field-programmable gate array (FPGA), and other chips, as well as servers, mobiles and embedded terminal devices constructed by these chips.

6.3 Compilers

Existing intermediate representations, including low level virtual machine (LLVM) CUDA, Intel Inference Engine, TVM, XLA, open neural network exchange (ONNX), neural network exchange format (NNEF), etc.

6.4 Frameworks

The frameworks are used to shield the underlying hardware and implement the AI algorithms, including the training framework and inference framework. The training framework includes TensorFlow, PyTorch, Caffe, PaddlePaddle, MXNet, etc. The inference framework includes Tensor RT, TensorFlow Lite, Caffe2go, Paddle Mobile, etc.

6.5 Fundamental applications

The fundamental applications mainly indicate the deep learning technology for each application scenario, such scenarios include computer vision, natural language processing, intelligent speech, etc.

6.6 Industry applications

The industry applications mean applying deep learning technology to specific vertical industries.

7 Indicating requirements and evaluation methods of deep learning software framework

7.1 Requirements overview

This Recommendation separately formulates related requirements for the training framework and inference framework of deep learning. For a training framework, it covers five aspects including ecological construction, ease of use, performance, supported architecture, security and stability. For inference framework, it covers four aspects including ease of use, performance, underlying optimization, security and stability.

The training framework indicating system includes the elements listed in Table 7-1.

Table 7-1 – Training framework indicating system and specific indicator items

Indicating system	Specific indicator item
Ecological construction	Interface
	Core developers and contributors
	The situation in which the issues are solved
Ease of use	Model building and conversion
	Secondary development based on high-level language
	Custom extension
	Cross-platform
	Model library support
	Tutorial, documentation, and training materials
	Dynamic graph and static graph

Table 7-1 – Training framework indicating system and specific indicator items

Indicating system	Specific indicator item
	Stability
	Debuggability
Performance	Model library operating behaviour
	Operating behaviour of a customized model
	Hardware acceleration support
Supported architecture	CPU/FPGA
	Single GPU/multi-GPU
	Distributed training
	Virtual environment support
	Operating system support
Security and stability	Usage of third-party library
	Data security

The inference framework indicating system includes the elements listed in Table 7-2.

Table 7-2 – Inference framework indicating system and specific indicator items

Indicating system	Specific indicator item
Ease of use	Model optimizing functionality
	Universal model representation
	Cross-platform
Performance	Inference speed
	Run-up time
	System resources occupation
	Energy consumption
Underlying optimization	Support for different underlying hardware
	Optimization for instruction set
Security and stability	Model encryption

7.2 Classic deep learning models

Evaluation for a deep learning software framework needs some classic test examples. Table 7-3 lists some classic deep learning models in different test scenarios and corresponding metrics, which are recommended when doing the evaluation.

Table 7-3 – Different test scenarios and classic models with related dataset and quality objectives

Test scenarios	Dataset	Quality objectives	Reference model
Image classification	ImageNet	74.90% classification	ResNet-50 v1.5
Object detection (light-weight)	COCO 2017	21.2% mAP	SSD (ResNet-34 backbone)

Table 7-3 – Different test scenarios and classic models with related dataset and quality objectives

Test scenarios	Dataset	Quality objectives	Reference model
Object detection (heavy-weight)	COCO 2017	0.377 Box min AP, 0.339 Mask min AP	Mask R-CNN
Pre-trained model	Wikipedia	Related to specific tasks	BERT
Translation (recurrent)	WMT English-German	21.8 BLEU	Neural machine translation
Translation (non-recurrent)	WMT English-German	25.0 BLEU	Transformer
Recommendation	MovieLens-20M	0.635 HR@10	Neural collaborative filtering
Reinforce learning	Pro games	40.00% move prediction	Mini Go

7.3 Specific indicating items and evaluation methods for training framework

7.3.1 Ecological construction

7.3.1.1 Interface

Refers to the programming languages that the training framework supports.

7.3.1.2 Core developers and contributors

Refers to the number of core developers and contributors, activeness of the framework, especially the number of watch, star, fork, pull request, core developers, and contributors in GitHub.

7.3.1.3 The situation in which issues are solved

Refers to the mechanism about solving issues and the operating situation.

7.3.2 Ease of use

7.3.2.1 Model building and conversion

Refers to the consistency of application programming interface (API), model definition structure, model storage format, and model converters.

7.3.2.2 Secondary development based on high-level language

Refers to high-level languages the training framework supports. Weighting the popularity of each language to calculate the average support for high-level languages of the training framework.

7.3.2.3 Custom extension

Refers to support and easy-used tools to develop extensive functionality, such as custom op, custom network layer, and added lines of code to realize one extensive functionality.

7.3.2.4 Cross-platform

Refers to different supported platforms of the framework, such as Linux, Windows, Android, iOS, cloud platform, and so on.

7.3.2.5 Model library support

Refers to supported deep learning models, including CNN, RNN, and models that are used in industrial scenarios. The number of models and the covered types of scenes are required to be included.

7.3.2.6 Tutorial documentation and training materials

Tutorial documentation and training materials are required to be declared, including whether or not official documents, community documents and material from cooperative training institutions exist. If they exist, the number of documents, completeness, updates, and languages that are used to write the documents are also required to be included.

7.3.2.7 Dynamic graph and static graph

Refers to code execution mechanism of the training framework, such as a dynamic or a static graph.

7.3.2.8 Stability

Refers to the mechanism of improving the stability of the training framework.

7.3.2.9 Debuggability

Refers to the flexibility and effectiveness of debugging mechanisms.

7.3.3 Performance

7.3.3.1 Model library operating performance

Refers to the execution time and memory usage of running classic models from the official model library with different batch sizes.

7.3.3.2 Operating performance of a customized model

Refers to concurrency, stability, and scale of features. For concurrency, evaluations are recommended to be taken to get the specific calculating quantities executed simultaneously and the interactive efficiency of the framework. For stability, evaluations are recommended to be taken to get statistic data of the failures during a-week-time by continuously executing the tasks based on the training framework. For a scale of features, evaluations are recommended to be taken to get the biggest throughput ability of the training framework and level it with its ability.

7.3.3.3 Hardware acceleration support

Refers to the optimization of the training framework for different underlying hardware, such as CPU, GPU, FPGA, dedicated AI accelerators, and so on.

7.3.4 Supported architecture

7.3.4.1 CPU/FPGA

Refers to the performance of classic models executed on CPU or FPGA, including training time and memory usage.

7.3.4.2 Single GPU/multi-GPU

Refers to the synchronized stochastic gradient descent (SGD) performance of classic models and other sophisticated model optimization methods executed on a single GPU and multi-GPU.

7.3.4.3 Distributed training

Refers to synchronized SGD behaviour of classic models and other sophisticated model optimization methods executed on the same machine and different machines.

7.3.4.4 Virtual environment support

Refers to the applicability of the training framework to a virtual environment, virtual machine, and so on.

7.3.4.5 Operating systems support

Refers to the applicability of the training framework to multiple operating systems such as Linux, Windows, and macOS.

7.3.5 Security and stability

7.3.5.1 Usage of third-party library

Refers to the third-party library used in the training framework.

7.3.5.2 Data security

Refers to whether encrypted data is supported in training and whether an encryption tool is provided for the framework.

7.4 Specific indicating items and evaluation methods for inference framework

7.4.1 Ease of use

7.4.1.1 Model optimizing functionality

Refers to the ability of model compression, accelerate algorithm components and hyper-parameter optimization components.

7.4.1.2 Universal model representation

Refers to whether the inference framework supports the universal format of the deep learning model, such as ONNX.

7.4.1.3 Cross-platform

Refers to the compatibility of framework to different platforms such as Linux, Windows, Android, iOS, cloud platform, etc.

7.4.2 Performance

7.4.2.1 Inference speed

Refers to the inference time by executing the classic deep learning models using inference framework under different energy consumption.

7.4.2.2 Run-up speed

Refers to the run-up time by executing the classic deep learning models using inference framework based on different scales of data.

7.4.2.3 System resources occupation

Refers to storage consumption, CPU memory usage, and GPU memory usage by executing deep learning models of different scales.

7.4.2.4 Energy consumption

Refers to cost energy in one hour by executing deep learning models of different scales.

7.4.3 Underlying optimization

7.4.3.1 Support for different underlying hardware

Refers to the ability of inference framework supporting different underlying hardware such as FPGA, ASIC, GPU, and so on.

7.4.3.2 Optimization for instruction set

Refers to optimization of the supported instruction set of the inference framework, such as advanced RISC machine (ARM), INTEL instruction set.

7.4.4 Security and stability

7.4.4.1 Model encryption

Refers to the encryption strategy of the inference framework for the trained inference model.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	**Non-telephone telecommunication services**
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2021