INTERNATIONAL STANDARD

ISO/IEC 24029-2

First edition
2023-08

Artificial intelligence (AI) — Assessment of the robustness of neural networks —

Part 2: Methodology for the use of formal methods

Intelligence artificielle (IA) — Evaluation de la robustesse de réseaux neuronaux —

Partie 2: Méthodologie pour l'utilisation de méthodes formelles

Reference number
ISO/IEC 24029-2:2023(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

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 not 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 42, *Artificial intelligence,* in collaboration with the European Committee for Standardization (CEN) Technical Committee CEN/CLC/JTC 21, *Artificial Intelligence,* in accordance with the Agreement on technical cooperation between ISO and CEN (Vienna Agreement).

A list of all parts in the ISO/IEC 24029 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

Neural networks are widely used to perform complex tasks in various contexts, such as image or natural language processing and predictive maintenance. AI system quality models comprise certain characteristics, including robustness. For example, ISO/IEC 25059:2023,[1] which extends the SQuaRE International Standards[2] to AI systems, considers in its quality model that robustness is a sub-characteristic of reliability. Demonstrating the ability of a system to maintain its level of performance under varying conditions can be done using statistical analysis, but proving it requires some form of formal analysis. In that regard formal methods can be complementary to other methods in order to increase trust in the robustness of the neural network.

Formal methods are mathematical techniques for rigorous specification and verification of software and hardware systems with the goal to prove their correctness. Formal methods can be used to formally reason about neural networks and prove whether they satisfy relevant robustness properties. For example, consider a neural network classifier that takes as input an image and outputs a label from a fixed set of classes (such as car or airplane). Such a classifier can be formalized as a mathematical function that takes the pixel intensities of an image as input, computes the probabilities for each possible class from the fixed set, and returns a label corresponding to the highest probability. This formal model can then be used to mathematically reason about the neural network when the input image is modified. For example, suppose when given a concrete image for which the neural network outputs the label "car" the following question can be asked: "does the network output a different label if the value of an arbitrary pixel in the image is modified?" This question can be formulated as a formal mathematical statement that is either true or false for a given neural network and image.

A classical approach to using formal methods consists of three main steps that are described in this document. First, the system to be analyzed is formally defined in a model that precisely captures all possible behaviours of the system. Then, a requirement is mathematically defined. Finally, a formal method, such as solver, abstract interpretation or model checking, is used to assess whether the system meets the given requirement, yielding either a proof, a counterexample or an inconclusive result.

This document covers several available formal method techniques. At each stage of the life cycle, the document presents criteria that are applicable to assess the robustness of neural networks and to establish how neural networks are verified by formal methods. Formal methods can have issues in terms of scalability, however, they are still applicable to all types of neural networks performing various tasks on several data types. While formal methods have long been used on traditional software systems, the use of formal methods on neural networks is fairly recent and is still an active field of investigation.

This document is aimed at helping AI developers who use neural networks and who are tasked with assessing their robustness throughout the appropriate stages of the AI life cycle. ISO/IEC TR 24029-1 provides a more detailed overview of the techniques available to assess the robustness of neural networks, beyond the formal methods described in this document.

Artificial intelligence (AI) — Assessment of the robustness of neural networks —

Part 2: Methodology for the use of formal methods

1 Scope

This document provides methodology for the use of formal methods to assess robustness properties of neural networks. The document focuses on how to select, apply and manage formal methods to prove robustness properties.

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 22989:2022, *Information technology — Artificial intelligence — Artificial intelligence concepts and terminology*

ISO/IEC 23053:2022, *Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 22989:2022, ISO/IEC 23053:2022 and the following 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
domain
set of possible inputs to a neural network characterized by attributes of the environment

EXAMPLE 1 A neural network performing a natural language processing task is manipulating texts composed of words. Even though the number of possible different texts is unbounded, the maximum length of each sentence is always bounded. An attribute describing this domain can therefore be the maximum length allowed for each sentence.

EXAMPLE 2 A face capture domain requirements can rely on attributes such as that the size of faces is at least 40 pixels by 40 pixels. That half-profile faces are detectable at a lower level of accuracy, provided most of the facial features are still visible. Similarly, partial occlusions are handled to some extent. Detection typically requires that more than 70 % of the face is visible. Views where the camera is the same height as the face perform best and performance degrades as the view moves above 30 degrees or below 20 degrees from straight on.

Note 1 to entry: An attribute is used to describe a bounded object even though the domain can be unbounded.

3.2
attribute
property or characteristic of an object that can be distinguished quantitatively or qualitatively by human or automated means

[SOURCE: ISO/IEC/IEEE 15939:2017, 3.2, modified — "entity" replaced with "object".]

3.3
bounded domain
set containing a finite number of objects

EXAMPLE 1 The domain of all valid 8-bit RGB images with n-pixels is bounded by its size which is at most $256^{3 \times n}$.

EXAMPLE 2 The number of all valid English sentences is infinite, therefore this domain is unbounded.

Note 1 to entry: The number of objects in an unbounded domain is infinite.

3.4
bounded object
object represented by a finite number of attributes

Note 1 to entry: Contrary to a bounded object, an unbounded object is represented with an infinite number of attributes.

3.5
stability
extent to which the output of a neural network remains the same when its inputs are changed

Note 1 to entry: A more stable neural network is less likely to change its output when input changes are noise.

3.6
sensitivity
extent to which the output of a neural network varies when its inputs are changed

Note 1 to entry: A more sensitive neural network is less likely to change its outputs when input changes are informative.

3.7
architecture
fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution

3.8
relevance
ordered relative importance of an input's impact on the output of a neural network as compared to all other inputs

3.9
criterion
rule on which a judgment or decision can be based, or by which a product, service, result, or process can be evaluated

[SOURCE: ISO/IEC/IEEE 15289:2019 3.1.6]

3.10
time series
sequence of values sampled at successive points in time

[SOURCE: ISO/IEC 19794-1:2011, 3.54]

3.11
reachability
property describing whether a set of states is possible to be reached by an AI agent in a given environment

3.12
piecewise linear neural network
neural network using piecewise linear activation functions

Note 1 to entry: Examples of linear activation functions are Rectify linear unit or MaxOut.

3.13
binarized neural network
neural network having parameters that are primarily binary

3.14
recurrent neural network
neural network maintaining an internal state which encodes what the neural network has learned after processing a subsequence of the input data

3.15
transformer neural network
transformer
neural network using a self-attention mechanism to weight the effect of different parts of the input data during processing

3.16
model checking
formal expression of a theory

3.17
structural-based testing
glass-box testing
white-box testing
structural testing
dynamic testing in which the tests are derived from an examination of the structure of the test item

Note 1 to entry: Structure-based testing is not restricted to use at component level and can be used at all levels, e.g. menu item coverage as part of a system test.

Note 2 to entry: Techniques include branch testing, decision testing, and statement testing.

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3.80]

3.18
closed-box testing
specification-based testing
black-box testing
testing in which the principal test basis is the external inputs and outputs of the test item, commonly based on a specification, rather than its implementation in source code or executable software

[SOURCE: ISO/IEC/IEEE 29119-1:2022, 3,75]

4 Abbreviated terms

AI artificial intelligence

BNN binarized neural networks

GNN graph neural networks

MILP mixed-integer linear programming

MRI magnetic resonance imaging

PLNN piecewise linear neural networks

ReLU rectified linear unit

RNN recurrent neural networks

SAR synthetic aperture radar

SMC satisfiability modulo convex

SMT satisfiability modulo theories

5 Robustness assessment

5.1 General

In the context of neural networks, robustness specifications typically represent different conditions that can naturally or adversarially change in the domain (see 5.2) in which the neural network is deployed.

EXAMPLE 1 Consider a neural network that processes medical images, where inputs fed to the neural network are collected with a medical device that scans patients. Taking multiple images of the same patient naturally does not produce identical images. This is because the orientation of the patient can slightly change, the lighting in the room can change, an object can be reflected or random noise can be added by image post-processing steps.

EXAMPLE 2 Consider a neural network that processes the outputs of sensors and onboard cameras of a self-driving vehicle. Due to the dynamic nature of the outside world, such as weather conditions, pollution and lighting conditions, the input to the neural network is expected to have wide variations of various attributes.

Importantly, these variations introduced by the environment are typically not expected to change the neural network's robustness. The robustness of the neural network can then be verified against changes to such environmental conditions by using relevant proxy specifications within the neural network's domain of use.

Robustness properties can be local or global.[10] It is more common to verify local robustness properties than global robustness properties, as the former are easier to specify. Local robustness properties are specified with respect to a sample input from the test dataset. For example, given an image correctly classified as a car, the local robustness property can specify that all images generated by rotating the original image within 5 degrees are also classified as a car. A drawback of verifying local robustness properties is that the guarantees are local to the provided test sample and do not extend to other samples in the dataset. In contrast, global robustness properties define guarantees that hold deterministically over all possible inputs.[11] For domains where input features have semantic meaning, for example, air traffic collision avoidance systems, the global properties can be specified by defining valid input values for the input features expected in a real-world deployment. Defining meaningful input values is more challenging in settings where the individual features have no semantic meaning. The set of robustness properties described in this clause is not exhaustive and it is possible that new robustness properties occur in the future.

5.2 Notion of domain

Most AI systems, including neural networks, are intended to operate in a particular environment where their performance characteristics can be defined and evaluated (typical metrics of evaluation can be found in ISO/IEC TR 24029-1:2021, Table 1). Robustness, being one of the key performance characteristics, is inseparable from the domain where a neural network is operating. The existence of a bounded domain is implicit in many neural network applications (e.g. image classification expects images of certain quality and in a certain format).

The agent paradigm shown in Figure 1 (reproduced from ISO/IEC 22989:2022, Figure 1) postulates that an agent senses its environment and acts on this environment towards achieving certain goals. The distinct concepts AI agent and environment are emphasized in this paradigm. The notion of domain captures the limitations of current technology where a neural network, being a particular type of AI agent, is technically capable of achieving its goal only if it is operating on appropriate inputs.

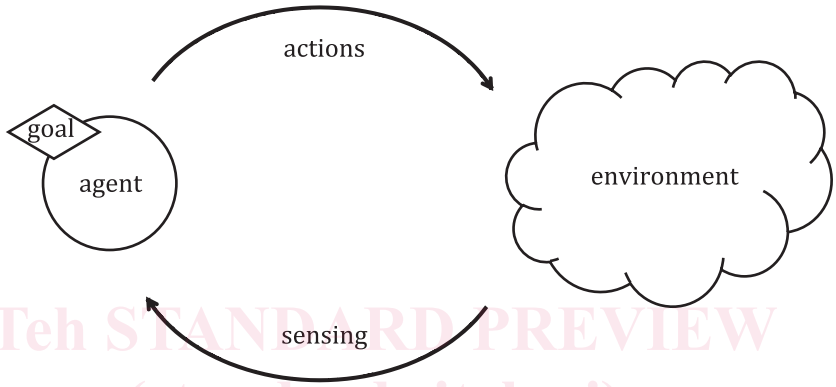

Figure 1 — The agent paradigm

The concept of domain rests on the following pillars:

— a domain shall be determined by a set of attributes which are clearly defined (i.e. the domain contains bounded objects);

— the specification of domain should be sufficient for the AI system to conduct one or more given tasks as intended;

— data used for training should be representative of data expected to be used for inference.

Establishing a domain involves specifying all data attributes essential for the neural network to be capable of achieving its goal.

Several popular domains of application of neural networks cover applications in vision, speech processing and robotics. To describe these domains, and more importantly their variability, the attributes used are generally numerical. Examples include the shape of an object in an image, the intensity of some pixels or the amplitude of an audio signal.

However, other domains can be expressed through non-numerical attributes including natural language processing, graph and Big Code (the use of automatically learning from existing code). In these cases, the attributes can be non-numerical, for example, the words in a sentence or the edges in a graph.

The attributes allow the AI developer to generate another instance in the domain from an existing instance. The attributes should be bounded in the robustness specification.

5.3 Stability

5.3.1 Stability property

A stability property expresses the extent to which a neural network output remains the same when its inputs vary over a specific domain. Checking the stability over a domain where the behaviour is supposed to hold allows for checking whether or not the performance can hold too. A stability property can be expressed either in a closed-end form (e.g. “is the variation under this threshold?”) or an open-ended form (e.g. “what is the largest stable domain?”).

In order to prove that a neural network remains performant in the presence of noisy inputs, a stability property shall be expressed. A stability property should be used on domains of uses which, in terms of expected behaviour, present some regularity properties. A stability property should not be used on a chaotic system as it is not relevant. However, even when the regularity of the domain is not easy to affirm (e.g. chaotic system), the stability property can be used to compare neural networks.

5.3.2 Stability criterion

A stability criterion establishes whether a stability property holds within a specific domain, not just for a specific set of examples or for a subset of the domain such as training or validation datasets. A stability criterion can be checked using formal methods described in 6.2.

A stability criterion shall define at least the domain value space and output value space on which it has been measured and the stability property expected.

A stability criterion may be used as one of the criteria to compare models.

For a comparison to be accurate, the following requirements shall be met:

— the neural networks perform the same task;

— the stability criterion is used on the same domain;

— the stability criterion proves the same objective.

For example, for a neural network doing classification, a stability criterion assesses whether or not a particular decision holds for every input in the domain. For a neural network doing regression, a stability criterion assesses whether or not the regression remains stable on the domain.

To be applicable, a stability criterion relies on pre-existing information of the expected output of the neural network. This information can be known by the AI developer or can be determined by another means (using simulation or solver systems). It is well-suited to assess the robustness over a domain where the expected answer is known to be similar. For this reason, a stability criterion is recommended for any decision-making process handled by a neural network (e.g. classification, identification).

5.4 Sensitivity

5.4.1 Sensitivity property

A sensitivity property on a neural network expresses the extent to which the output of a neural network varies when its inputs are changed. In order to assess the robustness on a domain, it is sometimes necessary to check the variability of a system. A sensitivity analysis can be carried out to determine how much the system varies and the inputs which can influence that variance. This analysis is then compared to a pre-existing understanding of the expected performance of the system.

When a sensitivity analysis is used to determine whether a neural network stays bounded, the sensitivity analysis shall be used over a domain. As is the case for the stability property, sensitivity analysis is more suited for domains of use which present some regularity properties.