

Open Neural Network Exchange (ONNX) — Part 3: Conformance test package

Preparatory Standard

Warning for drafts

This document is a draft and is not a standard. It is distributed for review and comment, is subject to change without notice, and may not be referred to as a standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Table of contents

Foreword.....	iv
Introduction.....	v
1. Scope.....	1
2. Normative references.....	1
3. Terms and definitions.....	1
4. Test vector format.....	1
4.1. General.....	1
4.2. Model.....	1
4.3. Values.....	2
4.4. Identification.....	2
5. Test selection by profile.....	2
5.1. General.....	2
5.2. Selection rules.....	2
5.3. Unsupported operators.....	2
6. Application of tolerances.....	2
6.1. General.....	2
6.2. Source of the tolerance.....	2
6.3. Comparison.....	3
7. Reporting.....	3
7.1. General.....	3
7.2. Content.....	3
7.3. Relationship to the conformance statement.....	3
Annex A Test vectors.....	4
Bibliography.....	5

Foreword

This document has been prepared by the ONNX Standardization Working Group.

It is Part 3 of ONNX 1; Part 1 specifies the core and Part 2 the operator sets.

Status of this document —

This is an unofficial working draft. It has no standing within the ONNX project, the Linux Foundation, IEC, ISO or any other standards development organization, and it MUST NOT be cited as a normative reference.

Introduction

A conformance requirement that cannot be checked is not a requirement. Part 1 states what an implementation must do; this part supplies the vectors that decide whether it did.

It is a separate part because it tracks Part 2: a vector exists for an operator at an operator set version, so the package moves on the operator sets' cadence rather than the core's.

It is also the part most likely to be embedded in implementations, as test suites usually are, which may argue for a licence different from that of the prose. That question is open; see Annex C of Part 1.

Open Neural Network Exchange (ONNX) — Part 3: Conformance test package

1. Scope

This document specifies the conformance test package for the Open Neural Network Exchange (ONNX) format: the format of a test vector, the rules for selecting vectors against a claimed conformance class or profile, the application of numerical tolerances, and the form of a conformance report.

This document does not specify:

- the requirements being tested, which are in [ONNX 1-1](#);
- the operator definitions the vectors exercise, which are in [ONNX 1-2](#).

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.

ONNX 1-1, *Open Neural Network Exchange (ONNX) — Part 1: Core*

ONNX 1-2, *Open Neural Network Exchange (ONNX) — Part 2: Operator sets*

3. Terms and definitions

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

The terms and definitions given in [ONNX 1-1](#) also apply.

3.1.

outcome

result of applying one [test vector](#) to an implementation, being one of *pass*, *fail*, *unsupported* or *error*

3.2.

test vector

model, together with input values, expected output values and the tolerance against which they are compared

4. Test vector format

4.1. General

A test vector consists of a model, one or more sets of input values, the corresponding expected output values, and the tolerance to be applied.

4.2. Model

The model of a test vector SHALL be a conforming model under [ONNX 1-1](#) and SHALL declare the operator set version the vector exercises.

4.3. Values

Editorial note —

Specify the encoding of input and expected output values. The reference implementation stores them as serialized tensor messages in a directory per test case; that is a workable choice and has the advantage of reusing the encoding the standard already specifies, but it must be stated normatively rather than inherited.

Specify also how values for the reduced-precision element types are stored, since those are the types for which Part 1 has no stated tolerance.

4.4. Identification

Each vector SHALL be identified by domain, operator, operator set version and a name unique within that triple.

NOTE The worked example in Clause 4 of [ONNX 1-2](#) cites a vector in this form, as `ai.onnx/Relu/14`.

5. Test selection by profile

5.1. General

An implementation is tested against the vectors implied by the conformance class and profile it claims under Clause 4 of [ONNX 1-1](#).

5.2. Selection rules

Editorial note —

State, for each conformance class, which vectors apply:

A producer is not exercised by input and output values at all; its vectors are models it emits, checked for conformance to Part 1. This may mean producer vectors are a different kind of artefact from evaluating-consumer vectors, and the format clause has to accommodate both or the classes need separate packages.

A validating consumer is exercised by deliberately non-conforming models, with the expected outcome being a specific diagnostic. No such corpus exists upstream; it has to be written, and it is the part of this package that would most improve the state of the art.

An evaluating consumer is exercised by the vectors of the operator sets it claims.

5.3. Unsupported operators

A vector exercising an operator the implementation declares unsupported SHALL be recorded with the outcome *unsupported*, and SHALL NOT be recorded as a failure.

6. Application of tolerances

6.1. General

A vector passes when every expected output value is reproduced within the tolerance stated for the vector.

6.2. Source of the tolerance

The tolerance is that of Clause 11 of [ONNX 1-1](#) for the element type concerned, unless the vector states a tighter one.

Editorial note —

This clause cannot be completed before Clause 11 of [ONNX 1-1](#) states a tolerance model; Annex C of that part records the absence as the largest blocking gap in the standard, and this clause is why it blocks. Without a tolerance there is no predicate that decides pass from fail, and a test package with no predicate is a collection of examples.

6.3. Comparison

Editorial note —

Specify the comparison itself: whether relative and absolute tolerances are combined conjunctively or disjunctively, how NaN compares to NaN, how the infinities compare, and whether an exact match is required for the integer and Boolean element types. These are small decisions individually and decide the outcome of vectors collectively.

7. Reporting

7.1. General

A conformance report records the outcome of every vector selected under Clause 5.

7.2. Content

A conformance report SHALL state:

- the identity and version of the implementation;
- the version of this package used;
- the conformance class and profile claimed;
- for each vector, its identifier and its [outcome](#);
- for each vector with the outcome *fail*, the achieved deviation.

7.3. Relationship to the conformance statement

The conformance statement required by Annex A of [ONNX 1-1](#) SHALL cite the report produced under this clause.

Annex A

(normative)

Test vectors

The vectors themselves are published as a separate artefact, in the format specified in [Clause 4](#), and are normative.

Editorial note —

The vectors are data, not prose, and are not reproduced in this document. This annex is to state where they are published, how a version of the package is identified, and what guarantees hold across versions — in particular whether a vector, once published, may change.

The obvious starting corpus is the reference implementation's backend test data, which is vendored in documentation form at `upstream/onnx/docs/OnnxBackendTest.md`. Adopting it means deciding what its existing tolerances mean normatively, which returns to the open question in Clause 6.

Bibliography

- [1] ONNX, *Open Neural Network Exchange*, <https://onnx.ai>