

# Conformance Testing Follow-up

Bruce ([feng.dai@intel.com](mailto:feng.dai@intel.com))

Dec. 2021

# Agenda

- Status update
- Open
- Proposals
- Plan

# Status update

- Landed the PR [wpt#31201](#) of WebNN IDL Tests
- Landed the PR [webnn-polyfill#139](#) of JavaScript CompareULP() function
- Collected ULP distances of 48 supported ops based on WebNN-polyfill Op-level tests - issue [webnn-polyfill#144](#)
  - Use result of CPU backend ([tfjs-backend-cpu](#) based) as the baseline

# Current 'Max ULP distance' of WebNN Ops

Op	Device 1		Device 2		Device 3	
	Wasm	WebGL	Wasm	WebGL	Wasm	WebGL
cos	3	590	3	4	3	4
sin	0	670	0	14	0	4
tan	4	284	4	14	4	5
conv2d	1	1	1	0	1	2
conv2d (fused sigmoid)	4320708*	0	4320708*	0	4320708*	0
sigmoid	4320708*	0	4320708*	0	4320708*	0
relu	0	21474836482#	0	21474836482#	0	0
matmul	28	14	28	18	28	28
concat	0	0	0	0	0	0
gemm	1	1	1	1	1	1
instanceNorm	128	128	128	128	128	128
pow**0.5	0	2	0	2	0	4
pow**50	1	51	1	51	1	71

## Observations:

1. Same max distance on Wasm backend for each op cross three devices
2. Various max distance on WebGL backend for some ops cross three devices, e.g. cos, sin, etc.
3. Some ops have same max distance on both Wasm & WebGL cross three devices, e.g. concat, gemm, etc.
4. The max distance of those ops with activation option would be affected by activation op, e.g. conv2d, batchNormal
5. The output of *relu(negative number)* on Device 1 & 2 + WebGL backend is -0.0, distance is 2147483648 to the baseline 0.0

Note: \* ULP distance 4320708 is actual output 6.054601485195952e-39 against baseline 0.0.

# ULP distance 2147483648 is actual output -0.0 against baseline 0.0 with negative input, ULP distance will be 0 for non-negative input.

# Open

1. What's the strategy of defining acceptable ULP distance of ops?
  - May we select number 0 to be the acceptable ULP distance for tensor manipulation ops? E.g., concat, reshape, etc.
  - Do we need get maximum upper limit ULP distance data for those ops having various distance? Is there a reference?
  - Should we distinguish -0.0 and +0.0?
  - Reference devices
2. How to collect Top-K data of Model level tests?
  - Models
  - Input
  - Reference devices

# Proposals

1. Use result of WebNN-polyfill CPU backend ([tfjs-backend-cpu](#) based) as the baseline
  - CPU uses Double Precision
2. Add tensor manipulation op tests into WPT, e.g., concat etc.
3. Use real input data and parameters for op test
  - Artificial test data couldn't reflect real case
  - Real Image / Voice data as input
  - Model's trained weights, bias, etc. as parameters

# Plan

- Op level tests
  - Collect ULP distance on WebNN-native e.g. DML/OpenVINO backend
  - Real test data (depend on WG inputs)
- Model-level tests (depend on WG inputs)
  - Investigate Top-K retrieval algorithm
  - Collect Top-k test result for first wave models
- Contribute WPT WebNN Op-level tests (float32)
  - [Propose to add ULP based assertion function into WPT](#) referring to JavaScript CompareUlp() function
  - Add tests of tensor manipulation ops into WPT (depend on WG inputs)

Thanks