WebNN Overview and Status Update

Ningxin Hu
Belem Zhang
May 2024
Ningxin Hu, Intel Principal Engineer, initiator and co-editor of the W3C Web Neural Network (WebNN) specification, Chromium committer and co-owner of the Chromium WebNN component

Belem Zhang, Engineering Manager leading Intel WebNN team for spec, Chromium and ONNX Runtime WebNN EP development, author of WebNN developer preview
Executive Summary

- Three AI hardware engines of AI PC: CPU, GPU and NPU
- WebNN brings a unified abstraction of neural networks to the web
- Accesses AI hardware acceleration through native OS ML API
- Delivers near-native performance and the next gen use cases

Status:
- Spec:
  - CNN/RNN - CR published Q2’23
  - Transformers/GenAI – CR refresh published Q2’24
- Implementation:
  - DirectML GPU on Win: Announcing developer preview
  - DirectML NPU on Win: Coming soon
  - CoreML on MacOS: WIP
  - TFLite on Android/ChromeOS: WIP
Age of the AI PC

Heterogenous execution of AI workloads embraces the best practices in AI software design.

Three AI Engines

GPU
High Throughput
Ideal for AI-accelerated digital content creation

NPU
Low Power
Ideal for sustained AI workloads and AI offload for battery life

CPU
Fast Response
Ideal for low-latency AI workloads
Hardware-Accelerated Web AI Overview

Use cases:
- Image Classification
- Object Detection
- Background Segmentation
- Noise Suppression
- Natural Language

Frameworks:
- TensorFlow.js
- ONNXRuntime Web
- MediaPipe Web
- Transformers.js

Web API:
- WebAssembly
- WebGPU
- WebNN
- API extensions

Web Engines:
- Web Browser (e.g., Chrome/Edge)
- JavaScript Runtime (e.g., Electron/Node.js)

Native ML APIs:
- TFLite
- DirectML
- CoreML
- Windows Studio Effects
- Other ML OS APIs

Hardware:
- CPU
- GPU
- NPU
WebNN Programming Model

WebNN brings a unified abstraction of neural networks to Web
Hello Tensors

Browser-native tensor operations targeting CPU, GPU and NPU

```javascript
1 // Step 0: Create a context and graph builder for 'gpu', 'cpu' or 'npu'.
2 const context = await navigator.ml.createContext({deviceType: 'gpu'});
3 const builder = new MLGraphBuilder(context);
4 // Step 1: Create a computational graph calculating `c = a * b`.
5 const a = builder.input('a', {dataType: 'float32', dimensions: [3, 4]});
6 const b = builder.input('b', {dataType: 'float32', dimensions: [4, 3]});
7 const c = builder.matmul(a, b);
8 // Step 2: Compile it into an executable graph.
9 const graph = await builder.build([c]);
10 // Step 3: Bind input and output buffers to the graph and execute.
11 const bufferA = new Float32Array(3*4).fill(1.0);
12 const bufferB = new Float32Array(4*3).fill(0.8);
13 const bufferC = new Float32Array(3*3);
14 const results = await context.compute(graph, {'a': bufferA, 'b': bufferB}, {'c': bufferC});
15 // Step 4: Retrieve the results.
16 console.log('values: ${results.outputs.c}');
```
WebNN Browser Implementation

Apps/ Frameworks

Web Application

JS ML Frameworks

Chromium

MLContext
MLGraphBuilder
MLGraph

WebNN Mojo Client

WebNN Mojo Server

CoreML Backend
DirectML Backend
TFLite Backend

Native ML APIs

macOS
Windows
Android/ChromeOS/Linux

CoreML
DirectML
TFLite

BNNS/MPS
MCDM
XNNPACK/Delegate

Hardware

GPU
NPU
CPU

OS Drivers

Chromium

Native ML APIs
Mainstream ML frameworks are integrating WebNN.
ONNX Runtime Web Code Samples with WebNN

Switching to WebNN can be done by modifying a single line of code

WebAssembly backend

```javascript
import { InferenceSession } from "onnxruntime-web";

// Initialize the ONNX model
const initModel = async () => {
  env.wasm.numThreads = 1; // 4
  env.wasm.simd = true;
  env.wasm.proxy = true;
  const options: InferenceSession.SessionOptions = {
    // provider name: wasm, webnn
    // deviceType: cpu, gpu, npu
    // powerPreference: default, high-performance
    executionProviders: [{ name: "wasm"}, // WebAssembly CPU
                       {
                         [name: "webnn", deviceType: "gpu", powerPreference: 'default']
                       }]
  };
  // ...
};
```

WebNN backend

```javascript
import { InferenceSession } from "onnxruntime-web";

// Initialize the ONNX model
const initModel = async () => {
  env.wasm.numThreads = 1; // 4
  env.wasm.simd = true;
  env.wasm.proxy = true;
  const options: InferenceSession.SessionOptions = {
    // provider name: wasm, webnn
    // deviceType: cpu, gpu, npu
    // powerPreference: default, high-performance
    executionProviders: [{ name: "webnn", deviceType: "gpu", powerPreference: 'default' }]
  };
  // ...
};
```
“Near-Native” Performance of WebNN on CPU

The average performance of listed 15 models on WebNN on CPU is about **93%** of native XNNPack

- **Browser**: Chrome Canary 118.0.5943.0
- **DUT**: Dell/Linux/i7-1260P, single p-core
- **Workloads**: MediaPipe solution models (FP32, batch=1)
"Near-Native" Performance of WebNN on GPU

The average performance of listed 26 models on WebNN DirectML is about **83%** of native DML on MTL iGPU
“Near-Native” Performance of WebNN on NPU

The average performance of listed 4 models on WebNN DirectML is about 80% of native DML on MTL NPU

- Browser: Chrome Canary 126.0.6459.0
- OS: Windows 11 Pro 23H2
- DUT: Asus Zenbook
- CPU: Intel(R) Core(TM) Ultra 7 155H 3.80 GHz
- NPU: Intel(R) AI Boost
- NPU Driver: 32.0.100.2381
WebNN Developer Preview

Run ONNX models in the browser with WebNN. The developer preview unlocks interactive ML on the web that benefits from reduced latency, enhanced privacy and security, and GPU acceleration from DirectML.

Stable Diffusion 1.5
Stable Diffusion Turbo
Segment Anything
Whisper Base
Image Classification
WebNN Execution Provider of ONNX Runtime Web with GPU acceleration from DirectML. Running on Intel® Core™ Ultra 7 processor 155H with integrated Arc™ GPU.

WebNN Developer Preview

Run ONNX models in the browser with WebNN. The developer preview unlocks interactive ML on the web that benefits from reduced latency, enhanced privacy and security, and GPU acceleration from DirectML.
WebNN Execution Provider of ONNX Runtime Web with GPU acceleration from DirectML. Running on Intel® Core™ Ultra 7 processor 155H with integrated Arc™ GPU.

WebNN Developer Preview

Run ONNX models in the browser with WebNN. The developer preview unlocks interactive ML on the web that benefits from reduced latency, enhanced privacy and security, and GPU acceleration from DirectML.

Stable Diffusion 1.5
Text-to-Image

Stable Diffusion Turbo
Text-to-Image

Segment Anything
Image Segmentation

Whisper Base
Automatic Speech Recognition
Vanilla JavaScript (plain JavaScript) use of WebNN API, with NPU acceleration from DirectML. Running on Intel® Core™ Ultra 7 processor 155H with integrated Intel® AI Boost NPU.

Object Detection
Detecting instances of semantic objects of a certain class in digital images and videos.

Semantic Segmentation
Partitioning image into semantically meaningful parts to classify each part into pre-determined class.

Facial Landmark Detection
Detecting facial landmarks like eyes, nose, mouth, etc., which can be used for web-based try-on simulator of online store.

Face Recognition
Detecting faces of participants using object detection and checking whether each face was present or not.
Call to Action

- Try WebNN on Microsoft Edge Dev channel and Google Chrome Dev channel
  - Navigate to about://flags in the URL bar and turn on “Enables WebNN API”

  WebNN Developer Preview  WebNN Samples
  https://aka.ms/webnn  https://webmachinelearning.github.io/webnn-samples/

- Join Intel, Microsoft, Google, Hugging Face, and other industry leaders and shape WebNN definition and development
References

- **WebNN Spec:** [https://www.w3.org/TR/webnn/](https://www.w3.org/TR/webnn/)
- **WebNN Implementation Status:** [https://webmachinelearning.github.io/webnn-status/](https://webmachinelearning.github.io/webnn-status/)
- **Awesome WebNN:** [https://github.com/webmachinelearning/awesome-webnn](https://github.com/webmachinelearning/awesome-webnn)
- **WebNN Dev Preview:** [https://microsoft.github.io/webnn-developer-preview/](https://microsoft.github.io/webnn-developer-preview/)
- **WebNN Samples:** [https://webmachinelearning.github.io/webnn-samples/](https://webmachinelearning.github.io/webnn-samples/)
- **ONNX Runtime WebNN Execution Provider:** [https://github.com/microsoft/onnxruntime/tree/main/onnxruntime/core/providers/webnn](https://github.com/microsoft/onnxruntime/tree/main/onnxruntime/core/providers/webnn)