web neural network

FEDAY

WEBNN, WEB 端侧推理的未来

胡宁馨 ningxin.hu@intel.com
张敏 belem.zhang@intel.com

英特尔 SATG Web 平台工程
2023 年 11 月

WebML 客户端推理的优势

隐私

摄像头、麦克风等传感器数据保留在设备中

离线

初始资源缓存并离线后，不再依赖网络

延迟

无云端网络问题，浏览器实时推理

成本

无需云端算力支持

0 安装

浏览器中运行，无需额外安装，并易于共享

跨平台

在几乎所有平台上运行 AI 应用

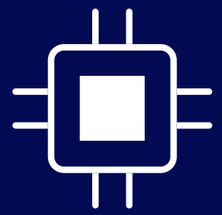
WebML 客户端推理

突发的
延迟敏感

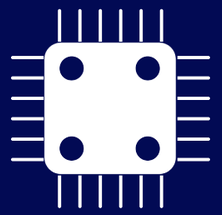
持续的
电量敏感

周期的
吞吐量敏感

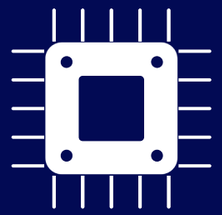
多样的客户端 AI 场景, 多种满足需求的计算单元



CPU
无处不在
低延迟, 单一推理任务



GPU
高并行性, 高 batch size
与 3D/渲染/媒体管道集成



NPU
专用低功耗AI加速器
高能耗比, 提升电源效率

Web 开发者的需求

> The web needs its own neural networks specification to leverage Apple Silicon, Tensor Cores, and others.

> Delighted to find the working drafts of WebNN. Incredible new power unlocked for the free, open and competitive Web!

> Native Tensor support! Would be amazing to have Tensor objects and ops built into Chrome, and available as an “ML API”.

> Although some scientific computing libraries exist for JS/TS, having built-in support would be far more desirable!

> If go through the code of utils, maths, audio, tensor in JS, it is annoying that I had to implement these ops myself in JS.

> llama2-7b in the browser – using WebNN – is going to be 🔥🔥 on-device, local ML 💪 cc @xenovacom

WebNN 简介

新兴的 W3C Web 标准 API

神经网络的统一抽象

通过原生 ML API 访问 AI 硬件加速器

接近原生的 AI 推理性能和结果的可靠性

目前在 Chrome 和 Edge Canary 中可用（runtime flag）

WebNN 标准规范

W3C Candidate Recommendation Draft

TABLE OF CONTENTS

Web Neural Network API

W3C Candidate Recommendation Draft, 6 June 2023

▼ More details about this document

This version:
https://www.w3.org/TR/2023/CRD-webnn-20230606/

Latest published version:
https://www.w3.org/TR/webnn/

Editor's Draft:
https://webmachinelearning.github.io/webnn/

Previous Versions:
https://www.w3.org/TR/2023/CRD-webnn-20230519/

History:
https://www.w3.org/standards/history/webnn

Implementation Report:
https://wpt.fyi/results/webnn?label=master&label=experimental&aligned&q=webnn

Test Suite:
https://github.com/web-platform-tests/wpt/tree/master/webnn

Feedback:
GitHub
Inline In Spec

Editors:
Ningxin Hu (Intel Corporation)
Chai Chaoweeraprasit (Microsoft Corporation)

Explainer:
explainer.md

Polyfill:
webnn-polyfill / webnn-samples

Abstract

This document describes a dedicated low-level API for neural network inference hardware acceleration.

WebNN 标准规范进展

已交付

- 2023 年 3 月: W3C CR
- 60 个 CNN/RNN 运算, float16/32, int32/uint32, int8/uint8
- 图像分类: SqueezeNet, MobileNet, ResNet
- 物体检测: TinyYOLO
- 噪声抑制: RNNoise, NSNet

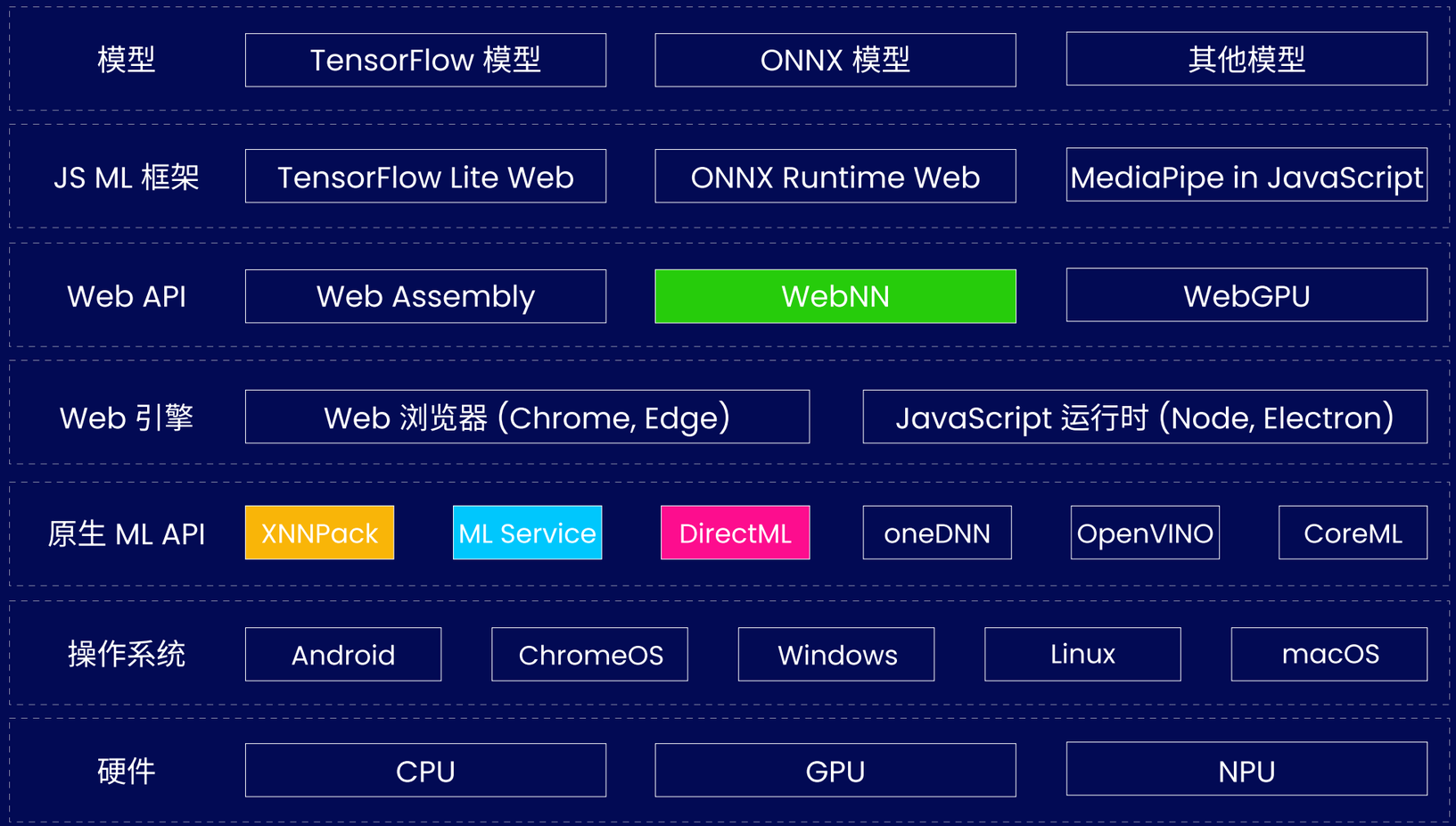
进行中

- 2023 年末: W3C 候选推荐更新
- 18~22 个 Transformer ops, int64/uint64, NPU 和量化 (TBD)

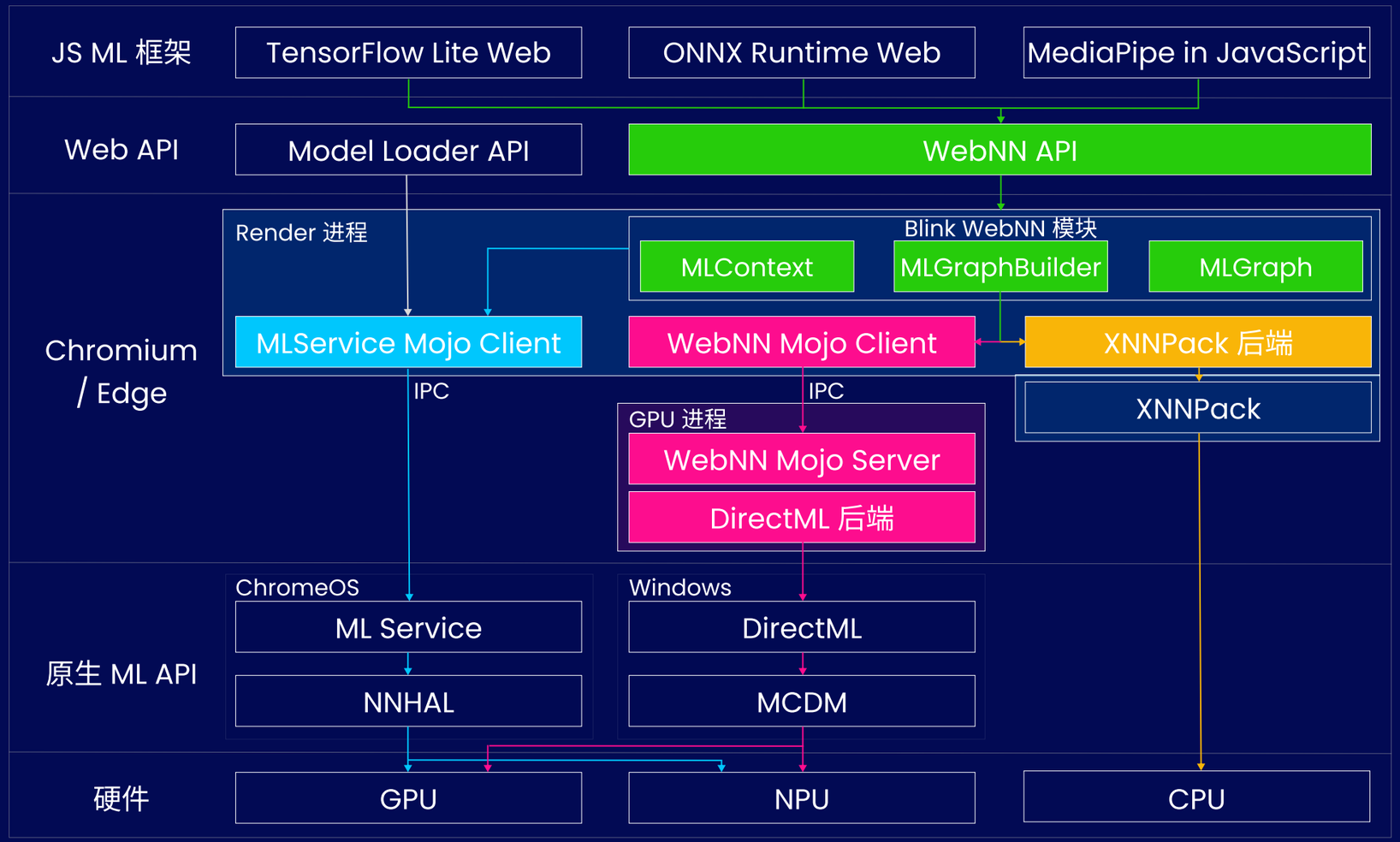
目标模型

- Text-to-image: Stable Diffusion unet/VAE/text encoder
- Image segmentation: Segment Everything decoder
- Speech-to-text: Whisper Tiny
- Text-to-text 生成 (encoder-decoder): T5 及 M2M100
- Text-generation (decoder): LLaMA

WebNN 架构


模型
TensorFlow 模型
ONNX 模型
其他模型
JS ML 框架
TensorFlow Lite Web
ONNX Runtime Web
MediaPipe in JavaScript
Web API
Web Assembly
WebNN
WebGPU
Web 引擎
Web 浏览器 (Chrome, Edge)
JavaScript 运行时 (Node, Electron)
原生 ML API
XNNPack
ML Service
DirectML
oneDNN
OpenVINO
CoreML
操作系统
Android
ChromeOS
Windows
Linux
macOS
硬件
CPU
GPU
NPU

WebNN 在 Chromium 中的实现



WebNN 操作符的实现状态（部分）

WebNN Spec	Web Platform Tests	XNNPack/CPU backend	External Delegate	Execution Provider
clamp	📈 🧪	✅ clamp ✅ Relu6	✅ ReluN1To1	✅ Clip
concat	📈 🧪	✅ concatenate2 ✅ concatenate3 ✅ concatenate4	✅ Concatenation	✅ Concat
conv2d	📈 🧪	✅ convolution_2d	✅ Conv2d ✅ DepthwiseConv2d	✅ Conv
convTranspose2d	📈 🧪	✅ deconvolution_2d	✅ TransposeConv ✅ Convolution2DTransposeBias	✅ ConvTranspose
add element-wise binary	📈 🧪	✅ add2	✅ Add	✅ Add
sub element-wise binary	📈 🧪	✅ subtract	✅ Sub	✅ Sub
mul element-wise binary	📈 🧪	✅ multiply2	✅ Mul	✅ Mul
div element-wise binary	📈 🧪	✅ divide	✅ Div	✅ Div
max element-wise binary	📈 🧪	✅ maximum2	✅ Maximum	✅ Max
min element-wise binary	📈 🧪	✅ minimum2	✅ Minimum	✅ Min
abs element-wise unary	📈 🧪	✅ abs	✅ Abs	✅ Abs
ceil element-wise unary	📈 🧪	✅ ceiling	✅ Ceil	✅ Ceil
floor element-wise unary	📈 🧪	✅ floor	✅ Floor	✅ Floor
neg element-wise unary	📈 🧪	✅ negate	✅ Neg	✅ Neg
elu	📈 🧪	✅ elu	✅ Elu	✅ Elu

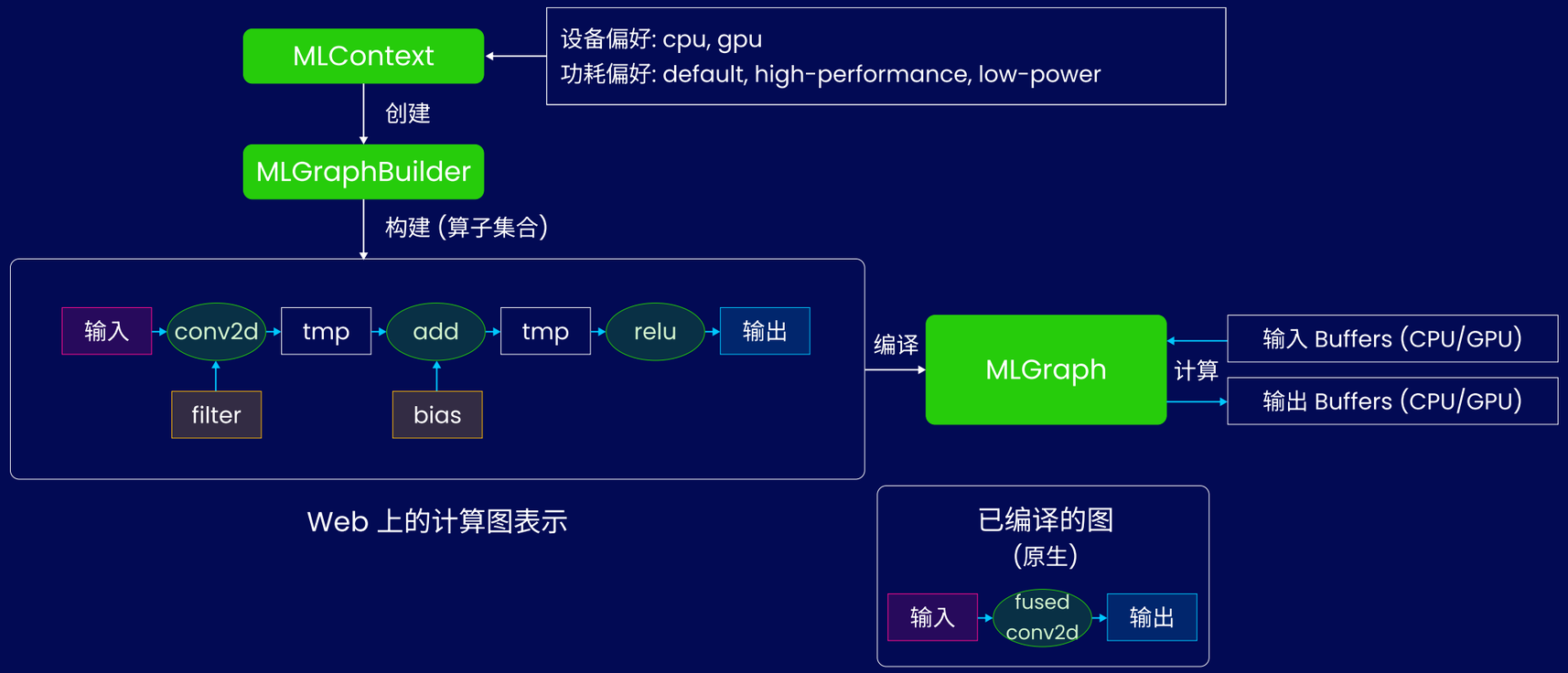
WebNN 操作符的实现状态（部分）

WebNN Spec	Web Platform Tests	XNNPack/CPU backend	External Delegate TensorFlow Lite for TensorFlow.js	Execution Provider ONNX RUNTIME
hardSwish	📈 🧪	✅ hardswish	✅ HardSwish	✅ HardSwish
leakyRelu	📈 🧪	✅ leaky_relu	✅ LeakyRelu	✅ LeakyRelu
pad	📈 🧪	✅ static_constant_pad	✅ Pad	✅ Pad
averagePool2d pooling	📈 🧪	✅ average_pooling_2d	✅ AveragePool2d ✅ Mean	✅ GlobalAveragePool ✅ AveragePool
maxPool2d pooling	📈 🧪	✅ max_pooling_2d	✅ MaxPool2d	✅ GlobalMaxPool ✅ MaxPool
prelu	📈 🧪	✅ prelu	✅ Prelu	✅ Prelu
relu	📈 🧪	✅ clamp	✅ Relu	✅ Relu
resample2d	🚀 🚀	✅ static_resize_bilinear_2d	✅ ResizeBilinear	✅ Resize
reshape	📈 🧪	✅ static_reshape	✅ Reshape	✅ Reshape
sigmoid	📈 🧪	✅ sigmoid	✅ Logistic	✅ Sigmoid
split	📈 🧪	✅ even_split2 ✅ even_split3 ✅ even_split4 ✅ static_slice (uneven split)	✅ Split	✅ Split
slice	📈 🧪	✅ static_slice	✅ Slice ✅ StridedSlice	✅ Slice
softmax	📈 🧪	✅ softmax	✅ Softmax	✅ Softmax
transpose	📈 🧪	✅ static_transpose	✅ Transpose	✅ Transpose
78	29	35	34	68

WebNN 的实现状态（DirectML）

- 目前已经支持 40 个 ops
- Transfomer 的 ops 也正与 spec 同步开发中
- 正在为 WebNN NPU 支持作出适配

WebNN 编程模型

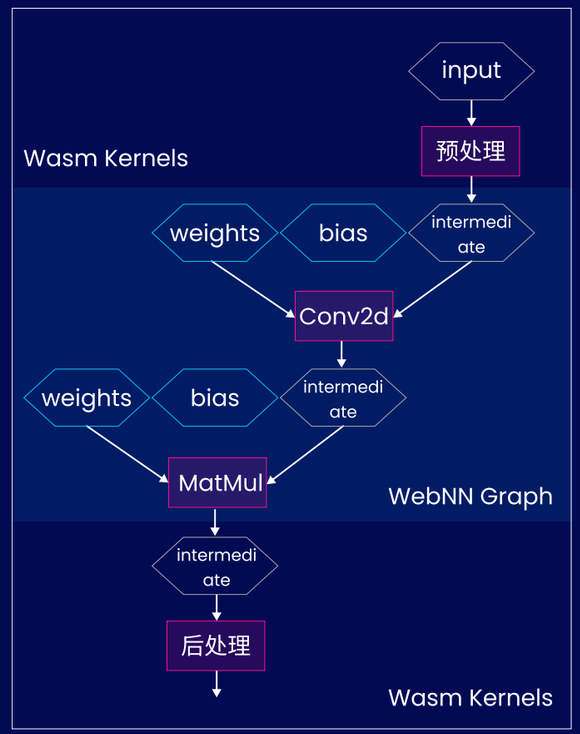

MLContext
设备偏好: cpu, gpu
功耗偏好: default, high-performance, low-power
创建
MLGraphBuilder
构建 (算子集合)
输入
conv2d
tmp
add
tmp
relu
输出
filter
bias
编译
MLGraph
输入 Buffers (CPU/GPU)
计算
输出 Buffers (CPU/GPU)
Web 上的计算图表示
已编译的图
(原生)
输入
fused
conv2d
输出
计算图图例
输入操作数
常量操作数
输出操作数
中间操作数
算子
WebNN API
其他 Web API
数据流
调用流

WebNN 代码示例

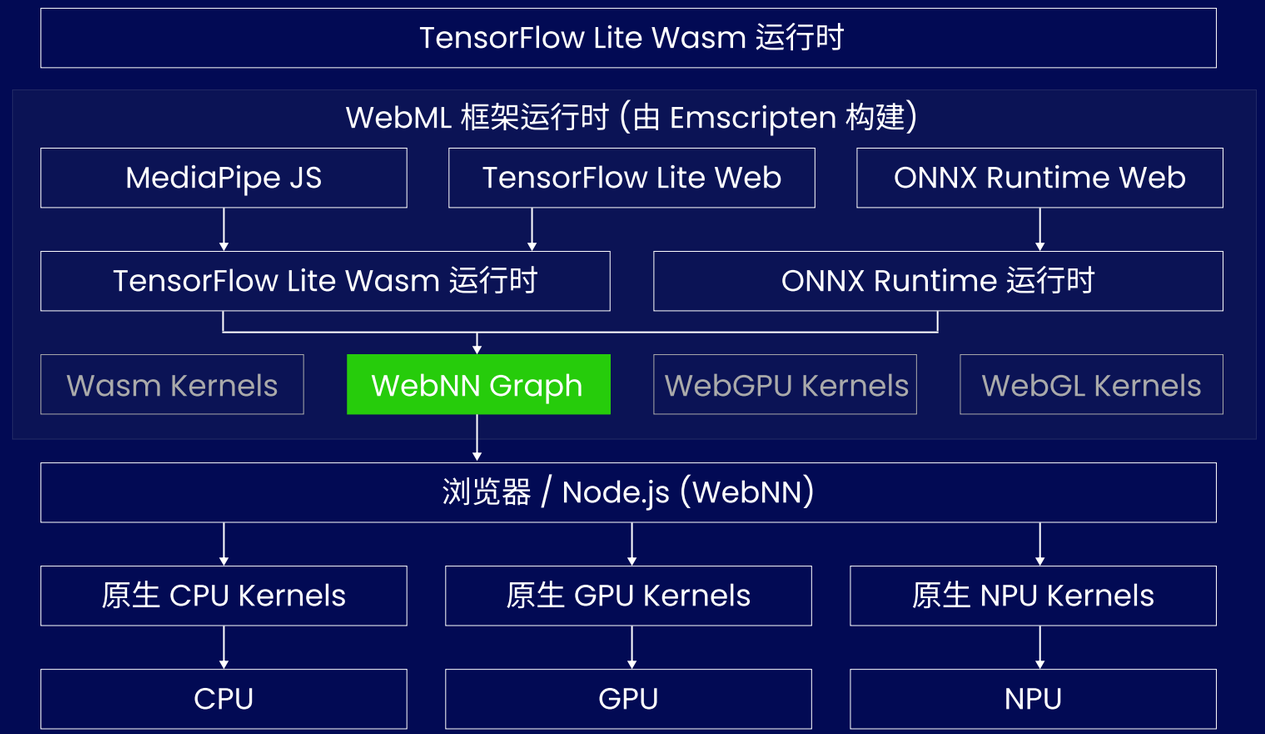
```
const context = await navigator.ml.createContext({powerPreference: 'low-power'})

// The following code builds a graph as:
// constant1 ---+
//              +--- Add ---> intermediateOutput1 ---+
// input1    ---+                                    |
//                                                   +--- Mul---> output
// constant2 ---+                                    |
//              +--- Add ---> intermediateOutput2 ---+
// input2    ---+

// Use tensors in 4 dimensions.
const TENSOR_DIMS = [1, 2, 2, 2];
const TENSOR_SIZE = 8;

const builder = new MLGraphBuilder(context);

// Create MLOperandDescriptor object.
const desc = {dataType: 'float32', dimensions: TENSOR_DIMS};

// constant1 is a constant MLOperand with the value 0.5.
const constantBuffer1 = new Float32Array(TENSOR_SIZE).fill(0.5);
const constant1 = builder.constant(desc, constantBuffer1);

// input1 is one of the input MLOperands. Its value will be set before execution
const input1 = builder.input('input1', desc);

// constant2 is another constant MLOperand with the value 0.5.
const constantBuffer2 = new Float32Array(TENSOR_SIZE).fill(0.5);
const constant2 = builder.constant(desc, constantBuffer2);
```

```
// input2 is another input MLOperand. Its value will be set before execution.
const input2 = builder.input('input2', desc);

// intermediateOutput1 is the output of the first Add operation.
const intermediateOutput1 = builder.add(constant1, input1);

// intermediateOutput2 is the output of the second Add operation.
const intermediateOutput2 = builder.add(constant2, input2);

// output is the output MLOperand of the Mul operation.
const output = builder.mul(intermediateOutput1, intermediateOutput2);

// Compile the constructed graph.
const graph = await builder.build({'output': output});

// Setup the input buffers with value 1.
const inputBuffer1 = new Float32Array(TENSOR_SIZE).fill(1);
const inputBuffer2 = new Float32Array(TENSOR_SIZE).fill(1);
const outputBuffer = new Float32Array(TENSOR_SIZE);

// Execute the compiled graph with the specified inputs.
const inputs = {
  'input1': inputBuffer1,
  'input2': inputBuffer2,
};
const outputs = {'output': outputBuffer};
const results = await context.compute(graph, inputs, outputs);

console.log('Output value: ' + results.outputs.output);
// Output value: 2.25,2.25,2.25,2.25,2.25,2.25,2.25,2.25
```

WebNN 和主流 JavaScript ML 框架的集成



解释器核心 Interpreter Core



WebNN 与 ONNXRuntime Web 集成的代码示例

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

// ...

// Initialize the ONNX model
const initModel = async () => {
  ort.env.wasm.numThreads = 1; // 4
  ort.env.wasm.simd = true;
  ort.env.wasm.proxy = true;

  const options: InferenceSession.SessionOptions = {
    // provider name: wasm, webnn
    // deviceType: cpu, gpu
    // powerPreference: default, high-performance
    executionProviders:
      [{ name: "wasm"}], // WebAssembly CPU
  };

  // ...
};

const results = await model.run(feeds);
const output = results[model.outputNames[0]];
```

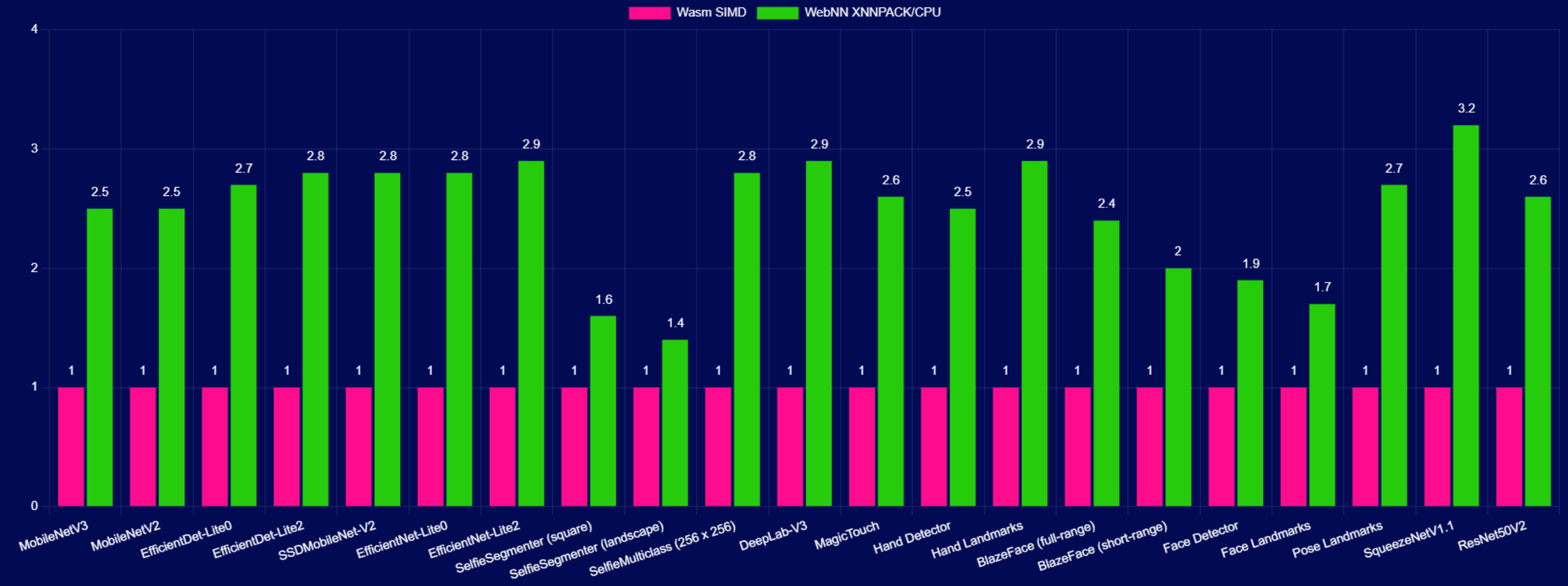
WebAssembly 后端

```
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
    // powerPreference: default, high-performance
    executionProviders:
      [{ name: "webnn", deviceType: "gpu", powerPreference: 'default' }],
  };

  // ...
};

const results = await model.run(feeds);
const output = results[model.outputNames[0]];
```

WebNN 后端

WebNN XNNPack/CPU 性能数据（标准化）

MediaPipe 模型, 越高越好



Web AI Benchmark alpha

Home Tests About

Backend: CPU | Wasm | Wasm 4 | WebNN | WebNN 4 | GPU | WebGL | WebGPU | WebNN | NPU | WebNN

Data Type: FP32 | INT64 | FP16 | INT8

Model Type: ONNX | TensorFlow Lite | NumPy | PyTorch

Model: DenseNet 121 | EfficientNet Lite 4 | MobileNet v2_10 | MobileNet v2_12 | ResNet50 v1 | ResNet50 v2 | SqueezeNet 1.1 | DeepLab v3 | Selfie Segmentation Gen… | Selfie Segmentation Lan… | Emotion FERPlus | FNS Candy | Tiny YOLO v2 | Stable Diffusion 2.1 V… | Stable Diffusion 2.1 V… | Segment Anything | T5 Small Decoder | Whisper Tiny Encoder

100 Runs: 100

PERFORMANCE (MS)

Compilation | First Inference | Average Inference | Median Inference | Best Inference

Model	Size	Type	Data	Wasm 1T	Wasm 4T	WebNN CPU 1T	WebNN CPU 4T	WebGL
MobileNet v2_12	13.32 MB		fp32	27.38	18.16	6.64	3.85	29.44
DenseNet 121	31.2 MB		fp32	200.32	74.83	65.79	34.33	81.17
EfficientNet Lite 4	49.54 MB		fp32	129.50	85.59	29.95	11.50	34.95
MobileNet v2_10	13.32 MB		fp32	28.57	19.04	6.74	3.38	27.88
ResNet50 v1	97.8 MB		fp32	251.24	93.34	69.77	24.42	78.79
ResNet50 v2	97.7 MB		fp32	284.46	122.64	90.76	36.98	96.33
SqueezeNet 1.1	4.72 MB		fp32	26.24	13.33	8.90	3.50	13.44
DeepLab v3	8.07 MB		fp32	753.34	418.62	197.68	95.49	
Tiny YOLO v2	60.54 MB		fp32	242.54	101.57	108.11	52.94	203.14

45/45 100.00%

Run Tests | Show Logs

ONNX Runtime Web Versions: Wasm v1.16.1 Public | WebGPU v1.17 Internal Nov 13 | WebNN v1.17 Internal Nov 08

x86-64 16 Logical Cores | Nominal CPU Pressure | Intel Graphicsgfx-driver-ci-master-15093 RI Direct3D11 | 6.02 MB | 395 MB 0.13% Storage Used | 3G | 100% AC

Windows 10 | Chrome 121.0.0.0 | Cross Origin Isolated

 WebNN Supported

Web AI Benchmark alpha

Home Tests About

Backend: CPU | Wasm | Wasm 4 | WebNN | WebNN 4 | GPU | WebGL | WebGPU | WebNN | NPU | WebNN

Data Type: FP32 | INT64 | FP16 | INT8

Model Type: ONNX | TensorFlow Lite | NumPy | PyTorch

Model:

DenseNet 121	EfficientNet Lite 4	MobileNet v2_10	MobileNet v2_12	MobileNet v2	MobileNet v2_12
ResNet50 v1	ResNet50 v2	SqueezeNet 1.1	SqueezeNet 1.0	DeepLab v3	Selfie Segmentation Gen…
Selfie Segmentation Lan…	Emotion FERPlus	FNS Candy	Tiny YOLO v2	ALBERT Base v2	ALBERT Base v2
BART Large CNN Encoder	BART Large CNN Encoder	BERT Base Cased	BERT Base Cased	BERT Base Uncased	BERT Base Uncased
BERT Base Multilingual…	BERT Base Multilingual…	CodeGen Mono 350M	CLIP ViT Base	DETR w/i ResNet-50	DINO ViT
DINO ViT	Distilbart CNN 6-6 Enc…	Distilbart CNN 6-6 Dec…	DistilBERT Base Cased …	DistilBERT Base Uncase…	DistilGPT2 Decoder
Distil-Whisper Decoder	GPT-2 Decoder	M2M100 418M Decoder	M2M100 418M Encoder	Stable Diffusion 1.5 U…	Stable Diffusion 1.5 V…
Stable Diffusion 2.1 V…	SAM B Decoder	SAM B Encoder	Segment Anything	Segment Anything	T5 Small Decoder
T5 Small Decoder	T5 Small Encoder	T5 Small Encoder	Vision Transformer (Vi…	ViT GPT2 Image Caption…	ViT GPT2 Image Caption…
Whisper Tiny Decoder	Whisper Tiny Decoder	Whisper Tiny Encoder	Whisper Tiny Encoder		

100 Runs: 100

PERFORMANCE (MS)

Compilation | First Inference | Average Inference | Median Inference | Best Inference

Model	Size	Type	Data	Wasm 4T	WebNN CPU 4T	WebGL	WebNN GPU
EfficientNet Lite 4	49.54 MB		fp32	74.54	9.52	23.65	5.45
MobileNet v2_10	13.32 MB		fp32	15.67	2.52	9.88	2.32
MobileNet v2_12	13.32 MB		fp32	15.14	2.49	10.01	2.36
ResNet50 v1	97.8 MB		fp32	101.75	20.25	23.79	4.77
SqueezeNet 1.1	4.72 MB		fp32	7.54	2.36	6.92	1.74
Emotion FERPlus	33.42 MB		fp32	17.04	4.33	11.24	2.19

24/24 100.00%

Run Tests | Show Logs

ONNX Runtime Web Versions: Wasm v1.16.1 Public | WebGPU v1.17 Internal Nov 13 | WebNN v1.17 Internal Nov 08

W3C Machine Learning for the Web

社区组

讨论和探索新想法，孵化机器学习推理的新提案

39 个组织代表, 126 名参与者

Adobe · agora · HUAWEI · moz://a · SAMSUNG · SAP

工作组

基于社区组孵化的提案，标准化机器学习推理的 Web API

17 个组织代表, 43 名参与者 (3 名特邀专家)

BBC · ByteDance · BERKELEY LAB · NVIDIA

Alibaba Group 阿里巴巴集团 · Baidu 百度 · Google · intel · Microsoft

Local Peer-to-Peer API

- 通过近距离通信传输消息或文件
- 隐藏各底层点对点技术的复杂性
- 于 2023 年 11 月 10 日进入 W3C Web Incubator CG 孵化
- 基于 Open Screen, QUICHE 等实现

谢谢！

web
neural network

https://webnn.dev

WebNN 交流群

张敏（Belem）