

Composants électroniques pour l'IA embarquée

« Benchmark des performances des μ contrôleurs orientés IA »

S. ROUXEL / A. NEDJAI – 15/10/2026 – v1.0

Réf du document : SemCompIA_PPT_SR_IAembarq_v1p0_20261015

Le CRT CRESITT est soutenu par :

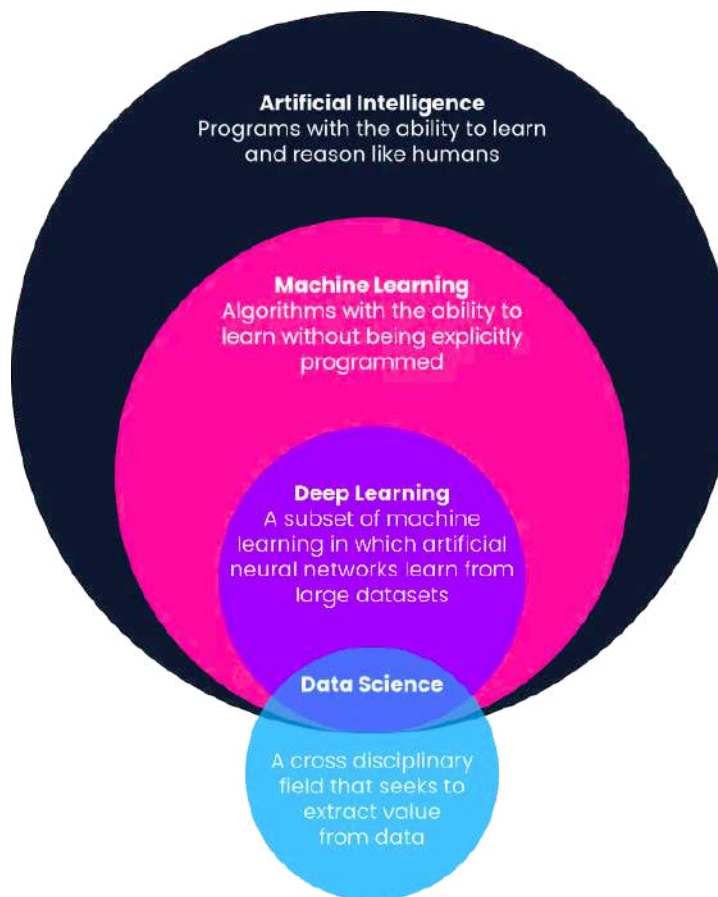


Cofinancé par
l'Union européenne



L'action de diffusion technologique est cofinancée par l'Union européenne.
L'Europe s'engage en région Centre-Val de Loire avec le Fonds européen de développement régional.

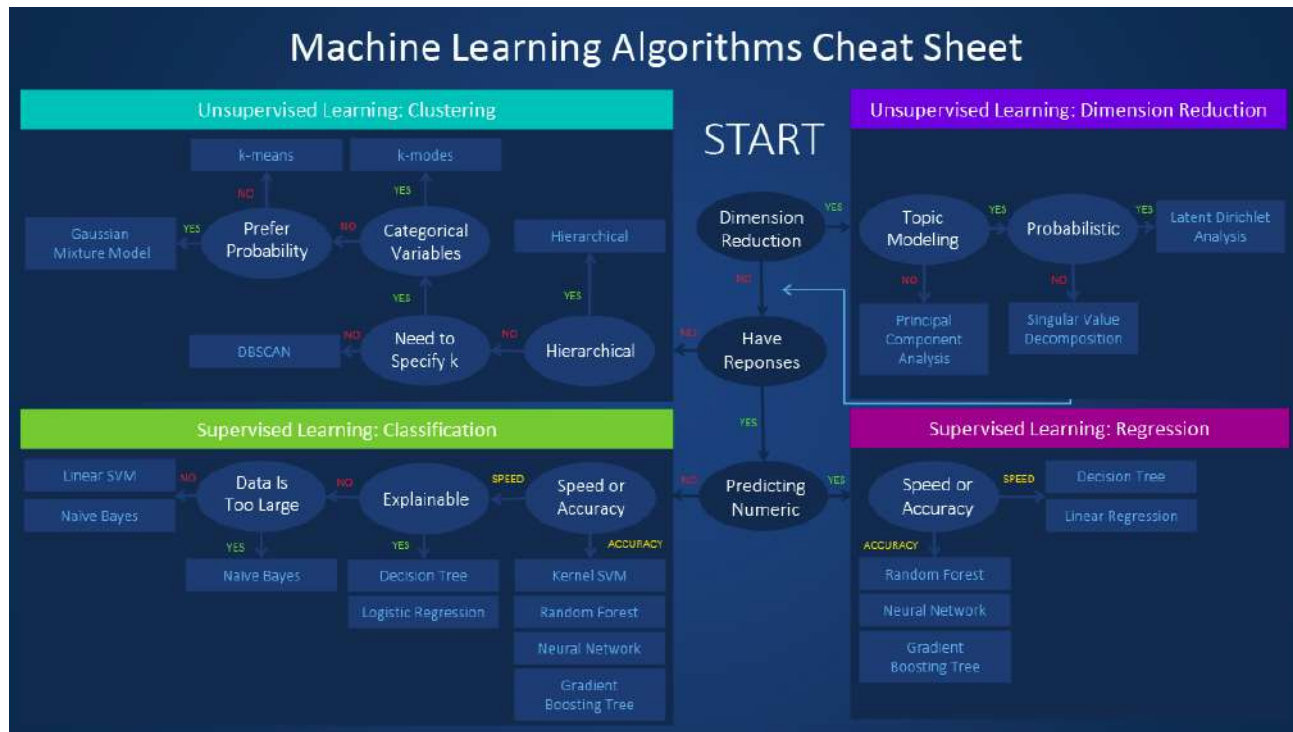
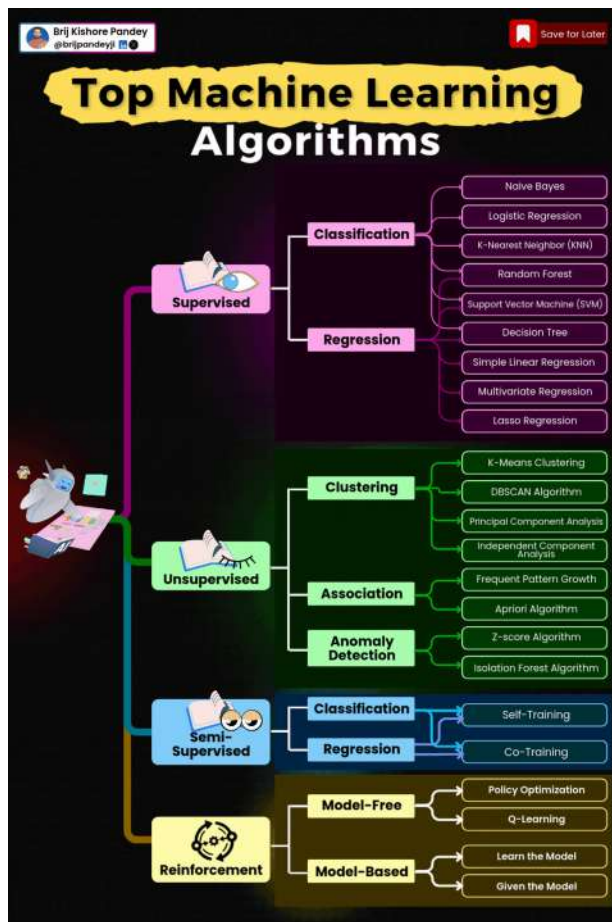
- Rappels IA
- Offre & ressources
- Retex : évaluation & performances



Src : <https://www.datacamp.com>

Processus en 2 phases

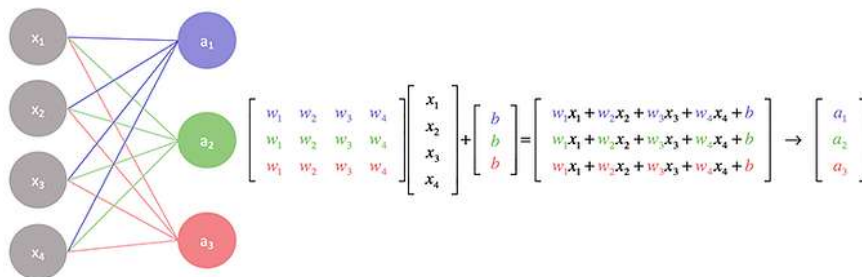
- 1) Apprentissage : un modèle apprend à reconnaître des tendances dans des données
- 2) Inférence : exécute un modèle entraîné pour générer des réponses ou faire des prédictions à partir de nouvelles données d'entrée



Src: devopsschool.com
www.kdnuggets.com

- Données = vecteurs , Transformations = matrices, Lignes = observations, Colonnes = caractéristiques

Type de Modèle	Opérations clés
Réseaux de neurones feedforward	Multiplication matricielle, ReLU, Softmax
CNN (Convolutional NN)	Convolution, Pooling, ReLU
RNN/LSTM	Multiplication matricielle récurrente
Transformers	Attention (Q·K^T), Softmax, LayerNorm
SVM	Produits scalaires, inversion de matrice
k-NN	Distance euclidienne/ cosinus
Reinforcement Learning	Q-Learning, Policy Gradient



Fonctions d'activation
ReLU, Tanh, sigmoïde, Softmax



$$f(x) = \max(0, x)$$

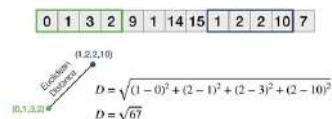
$$f(x) = (e^x - e^{-x}) / (e^x + e^{-x})$$

$$f(x) = 1 / (1 + e^{-x})$$

$$f(x) = e^{-x} / \sum_j e^{-x_j}$$

$$distance(I, C) = \sqrt{\sum_{j=1}^n (x_{c_j} - x_{I_j})^2}$$

Euclidean Distance

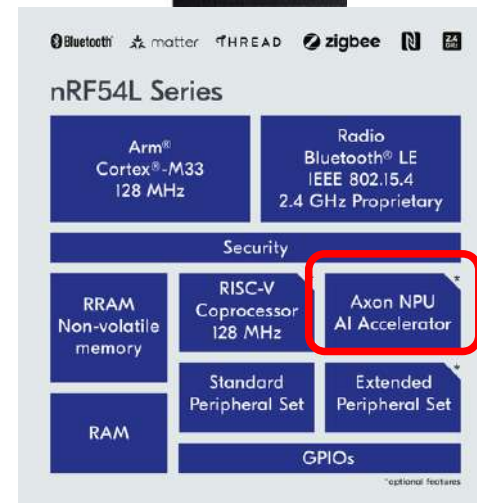


Src: www.jybaudot.fr,
pedrasecabegues.com
www.datacamp.com

- De la giga factory.....



- au microcontrôleur



Src : agentland.fr, www.nordicsemi.com

Offre disponible – Microcontrôleur + NPU, fonctions DSP++...

Fabricant	Réf	Micro	Jeu instr.	Freq (Mhz)	SRAM (Ko)	Flash (Ko)	NPU	npu name	NPU Freq	Memory Access (SRAM)	Gop/s int8	MAC /cycle
ST Microelectronic	STM32N647	arm® cortex M55	32	800	4200 + 8(backup)	externe	x	Neural-art accelerator	1Ghz		600	288
	STM32N657	arm® cortex M55	32	800	4200 + 8(backup)	externe					600	288
	STELLAR P3E	4 arm Cortex R52+ Cortex M4	32	up to 500 Up to 200	up to 1792	Up to 19500	x	Neural-art accelerator 11		NVRAM 384kB		
NXP	i.MX95	6x Arm® Cortex-A55 1x Arm® Cortex-M7 1x Arm® Cortex-M33	64 32 32		1376	externe	x	eiq neutron n3				
	i.MX94	4x Arm® Cortex-A55 2x Arm® Cortex-M7 2X Arm® Cortex-M33	64 32 32		3500	externe	x	eiq neutron				
	i.MX93	2x Arm® Cortex-A55 multicore complex 1x Arm Cortex-M33 1 arm Ethos U-65 microNPU	64 32 32	1700	640	externe	x	High-efficiency NPU Ethos U-65	1Ghz	104kB	1000 512 (FP16)	256
	MCX N94x	2xarm Cortex-M33	32	150	512	2000	x	eiq neutron n1-16			4.8	16

Offre disponible – Microcontrôleur + NPU, fonctions DSP++...

Fabricant	Réf	Micro	Jeu instr.	Freq (Mhz)	SRAM (Ko)	Flash (Ko)	NPU	npu name	NPU Freq	Memory Access (SRAM)	Gop/s int8	MAC /cycle
Nordic Semiconductor	nRF54LM20B	1 Arm® Cortex-M33 Risc-V	32	128	512	2000 NVM	x	Axon NPU	128M hz		600	288
Texas Instrument	TMS320F28P550	C2000	32	150		1100	x	TinyEngine NPU	150M hz		0.6 – 1.2	
	AM69A	8x Arm® Cortex-A72 8x Arm® Cortex-R5F	64	2000 1000	8000 + 128Kb LPDDR4			4x DSP C7x 4x MMV2	1Ghz	1MB Cache L2	32000	
	AM13Ex	Arm cortex M33	32	200	128	512	x	TinyEdge NPU				0.6
	MSPM0G5187	arm cortex M0+	32	80	32	128	x	TinyEngine NPU (CNN)			0.640	
Silicon Labs	EFR32MG24	arm cortex M33	32	78	256	1536		MVP(matrix vector Processor)				
Renesas	RA8P1	arm cortex-M85 Arm cortex-M33	64 32	1000 250	1792	1024	x	NPU Ethos-U55	500M hz		256	256
Espressif	ESP32-P4	Risc-V (dual core) Risc-V	32 32	360 40	preliminary doc			jeu instruction étendu (custom) Xai (DSP) pour algo AI DSP				

Offre disponible – Microcontrôleur + NPU, fonctions DSP++...

Fabricant	Réf	Micro	Jeu instr.	Freq (Mhz)	SRAM (Ko)	Flash (Ko)	NPU	npu name	NPU Freq	Memory Access (SRAM)	Gop/s int8	MAC /cycle
Espressif	ESP32-S3	Xtensa dual core LX7	32	240	512	384 + efuse (1792nits)		accélération logicielle instruction vectorielles				
Infineon (cypress)	edge E81 - E82	Cortex-M55 + Helium DSP Cortex-M33	32	200	5000		x	Ethos-u55-128	400Mhz			128
	edge E83 E84	Cortex-M55 + Helium DSP Cortex-M33	32	200	5000 6000		x	Ethos-u55-128	400Mhz			128
Himax	WiseEye2 AI Processor (WE2) CMOS imager + HX6538 WE2	2xCortex-M55	32	400 150	2000	512	x	Ethos-u55-256	400Mhz		50/204	256
Asygn	Colibri	Risc-V	32	300	128		x	Neural Network Accelerator		640	9.6	
Syntiant	NDP250	M0	32	48	500			Neural decision processor		6000	30	
Rockchip	RK3588	4xCortex-A76 + 4x Cortex-A55+ 3x cortex-M0	64 64 32	2200 1800	3000+1000 (share)		x	NPU	1Ghz		6000	
	RK3566	4x Cortex-A55		1800	64	32	x	NPU	1Ghz		1000	

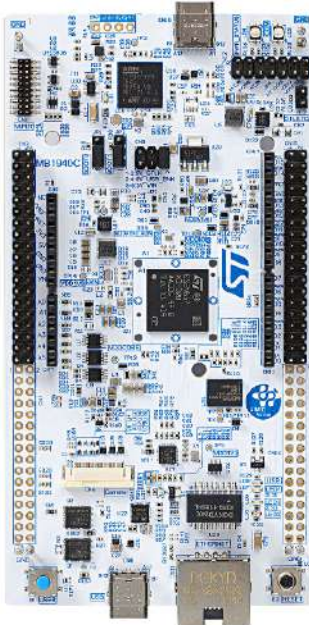
Offre disponible – Microcontrôleur + NPU, fonctions DSP++ ...

Fabricant	Réf	Micro	Jeu instr.	Freq (Mhz)	SRAM (Ko)	Flash (Ko)	NPU	npu name	NPU Freq	Memory Access (SRAM)	Gop/s int8	MAC /cycle
Analog device	Max78000	Cortex-M4 + RISC-V	32	100 60	128	512	x	NNA	up to 50Mhz	512		
Synaptics	Astra SR88	Cortex-M33 + 2dsp	32	250	3500		x	Synaptics NPU	500Mhz		50	
	Astra SR100	Cortex-M55 + Cortex-M4	64 32	400 100	Jusqu'à 3000	640	x	Ethos-U55 + Micro NPU				
Hailo	Hailo8-l	2x arm Cortex-M4	32	200	640	256(ROM)		NN Core x4			13000	

Choix des cartes :

- Critères :
 - Microcontrôleur pour l'embarqué : Cortex-M
 - Avec ressources de calcul dédiées aux algorithmes d'intelligence artificielle (inférence)
 - Architecture semblable avec niveau de performance gradué
 - Disponible

NUCLEO-N657X0-Q



50€
(à partir de ~22€/unité
la puce ~14€/500)

FRDM-MCXN947



23€
(~15€/unité
~9€/500)

EFR32xG24
Explorer Kit



35€
(~5€/unité
~3€/500)

Caractéristiques :

	STM32N657	NXP MCXN947	Silicon Labs EFR32MG24
Format de modèle accepté	TensorFlow Lite/ONNX	TensorFlow Lite	TensorFlow Lite
Microcontrôleur / CPU	Arm Cortex-M55	2 × Arm Cortex-M33	Arm Cortex-M33
Architecture CPU	32 bits	32 bits	32 bits
Fréquence CPU	800 MHz	150 MHz	78 MHz
Accélérateur IA	Neural-ART NPU	eIQ Neutron N1-16	MVP (Matrix Vector Processor)
Fréquence accélé- rateur	jusqu'à 1 GHz	150 MHz	–
Capacité de calcul INT8	600 GOPS	4.8 GOPS	non spécifiée en GOPS
RAM interne	4.2 MB	512 kB	256 kB
Flash interne	–	2 MB	1.5 MB
Mémoire externe	XSPI / FMC	FlexSPI	–
Accélération CPU complémentaire	MVE (Helium), FPU	SIMD, FPU, Power- Quad	DSP, FPU
Type d'accéléra- tion IA	NPU dédié	NPU dédié	accélérateur matri- ciel/vectorel

Opérations supportées :

NUCLEO-N657X0-Q *Neural-ART Accelerator*

Opérations supportées (NPU)	Opérations supportées partiellement (Fallback possible) NPU/CPU	Opérations Non supportées (Fallback CPU mais en int)	Opérations Non supportées (Fallback CPU en float à éviter)
CONV_2D, DEPTHWISE_CONV_2D, FULLY_CONNECTED, ADD, SUB, RELU, TANH, LOGISTIC, MAX_POOL_2D, AVERAGE_POOL_2D	MUL, SOFTMAX, PAD, TRANPOSE, REDUCE_MAX/MIN, RESIZE_NEARE ST_NEIGHBOR, BATCH_MATMUL	ARG_MAX, ARG_MIN, DIV, GATHER, RESIZE_BILINEAR	LOG, SQRT, MEAN, L2_NORMALIZATION, FLOOR, ROUND, SUM, DEQUANTIZE, QUANTIZE

EFR32xG24 Explorer Kit *Matrix Vector Processor (MVP)*

Opérations supportées
ADD FULLY_CONNECTED, FULLY_CONNECTED_INT8 AVERAGE_POOL_2D MAX_POOL_2D CONV_2D DEPTHWISE_CONV_2D TRANPOSE_CONV_2D

FRDM-MCXN947 *eIQ Neutron N1-16*

Opérations supportées (int8 uniquement)=> NPU	Opérations non supportées => DSP/CPU
ADD AVERAGE_POOL_2D CONV_2D DEPTHWISE_CONV_2D FULLY_CONNECTED LOGISTIC (Sigmoid) MAX_POOL_2D	MUL SOFTMAX LSTM(long short term memory) SVDF(Single value decomposition filter)

Choix des modèles évalués :

- Tests préliminaires sur Benchmark : **MLPerf Tiny**
<https://mlcommons.org/benchmarks/inference-tiny/>
<https://github.com/mlcommons/tiny>
<https://github.com/nxp-appcodehub/dm-mcxn947-npu-vs-tensorflm-benchmark>

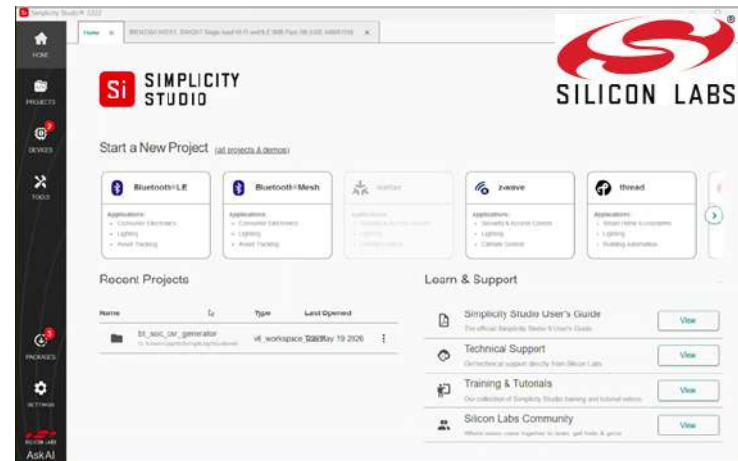
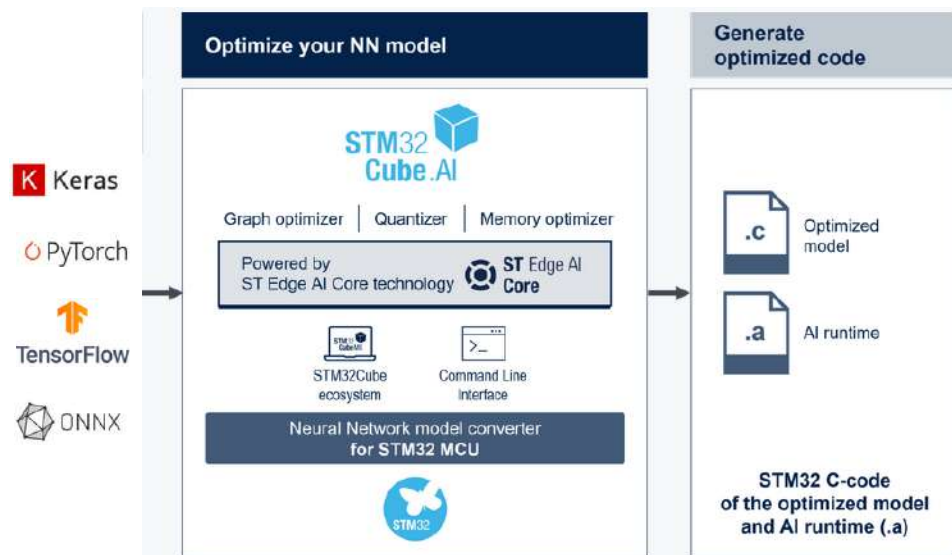


- MLPerf Tiny utilise Tensor Flow lite pour microcontrôleur (TFLM)
- Cas d'usage représentatifs du TinyML :
Keyword Spotting, détection d'anomalies,
classification d'images, Visual Wake Word

→ Utilisé comme base de départ pour valider la chaîne de déploiement et comparer les résultats NXP sur la carte FRDM-MCXN947

Choix des modèles évalués :

- Modèles dont la complexité est compatible avec les ressources matérielles des cartes ciblées.
- Possibilité de **convertir** les modèles par les outils des fabricants. => format commun **.tflite** en int8 (tensorflow lite)



+
AI/ML SDK



```
C:\Users\stagiaire\Desktop\NXP\neutron_sdk\bin>neutron-converter --dump-header-file-output --dump-header-file-input --target mcxn94
--input C:\Users\stagiaire\Desktop\NXP\model\model.tflite --output C:\Users\stagiaire\Desktop\NXP\model\model12.tflite
```

Choix des modèles évalués :

Deux types de données / Deux architectures

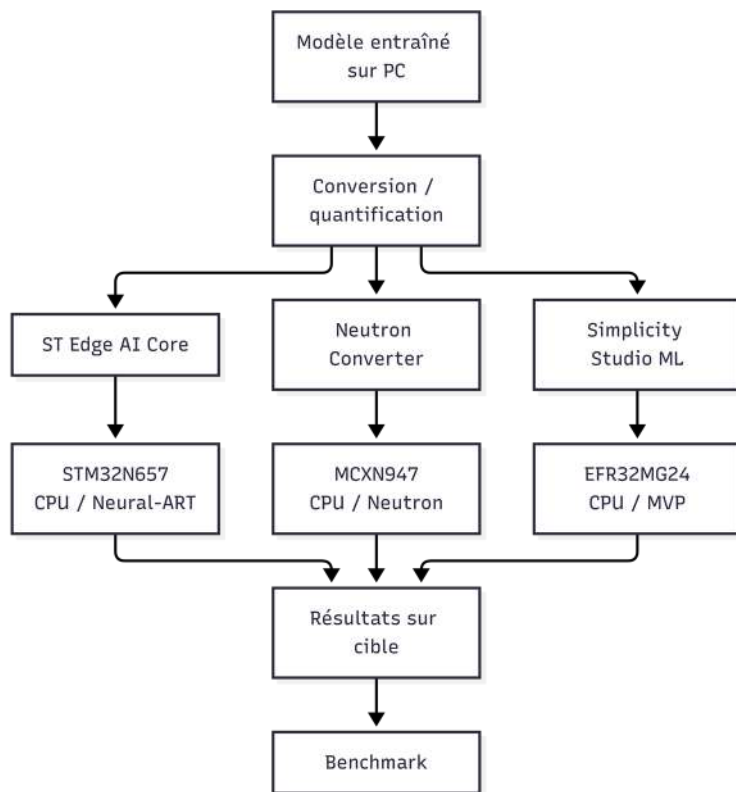
Détection d'anomalies

- Données **multi-capteurs**
- Température, humidité, CO, LPG, Fumée
- Modèle : **Autoencodeur**
- Détection par **erreur de reconstruction**

Reconnaissance des mots clés

- Données **audio**
- Google Speech Commands
- Modèle : **BC-Resnet**
- Classification de 12 classes

Scénario de développement



Obtention du modèle

Adaptation du modèle commun

Outils d'intégration des fabricants

Évaluation sur cibles

Modèle 1 - Détection d'anomalies

5 capteurs

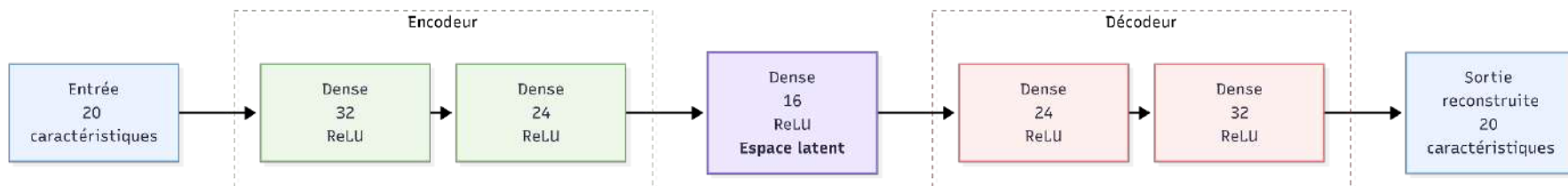
Température • Humidité • CO • LPG • Fumée

× 4 caractéristiques par capteur

Valeur instantanée -
 Δ mesure précédente
Moyenne glissante
Écart-type glissant

→ 20 caractéristiques d'entrée

Autoencodeur



Entraînement

- Train / Val / Test : 70 / 15 / 15 %
- Batch size : 256
- Early stopping : patience 5
- Arrêt : 17 époques

Principe

- Reconstruction de l'entrée
- Erreur quadratique moyenne -> score d'anomalie
- Seuil -> classification binaire

Quantification

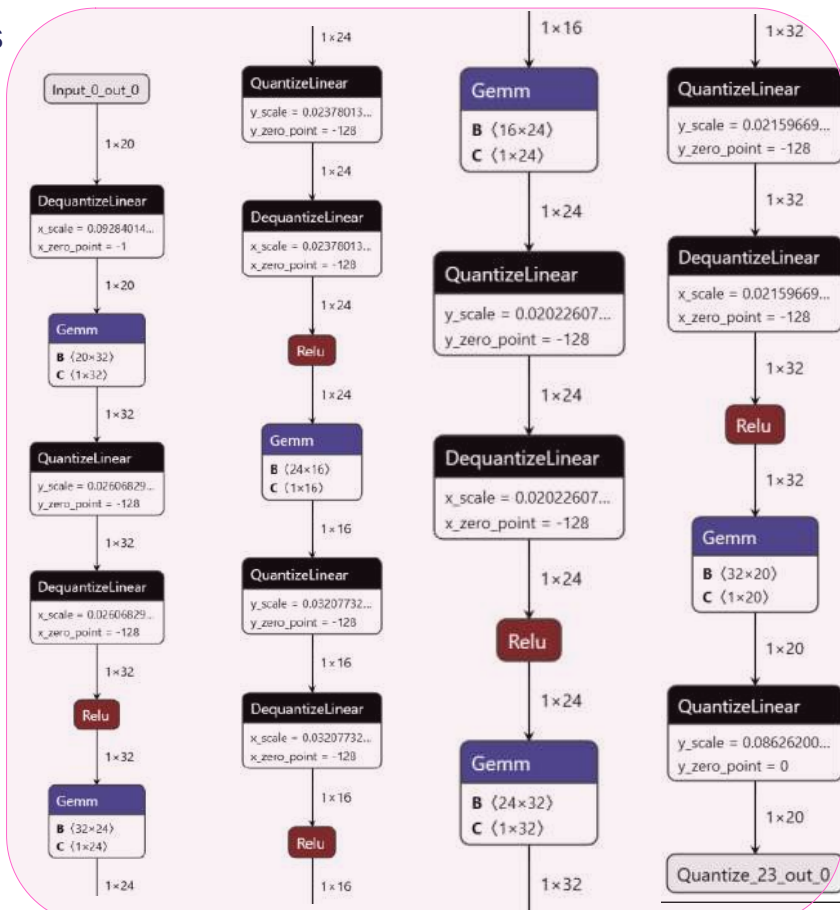
- Post-training-Quantization
- int8
- Représentatif dataset : 1000 données

limité à 1 dispositif du dataset initial (src : <https://www.kaggle.com/datasets/garystafford/environmental-sensor-data-132k/data>)

Cas ST : modèle autoencoder

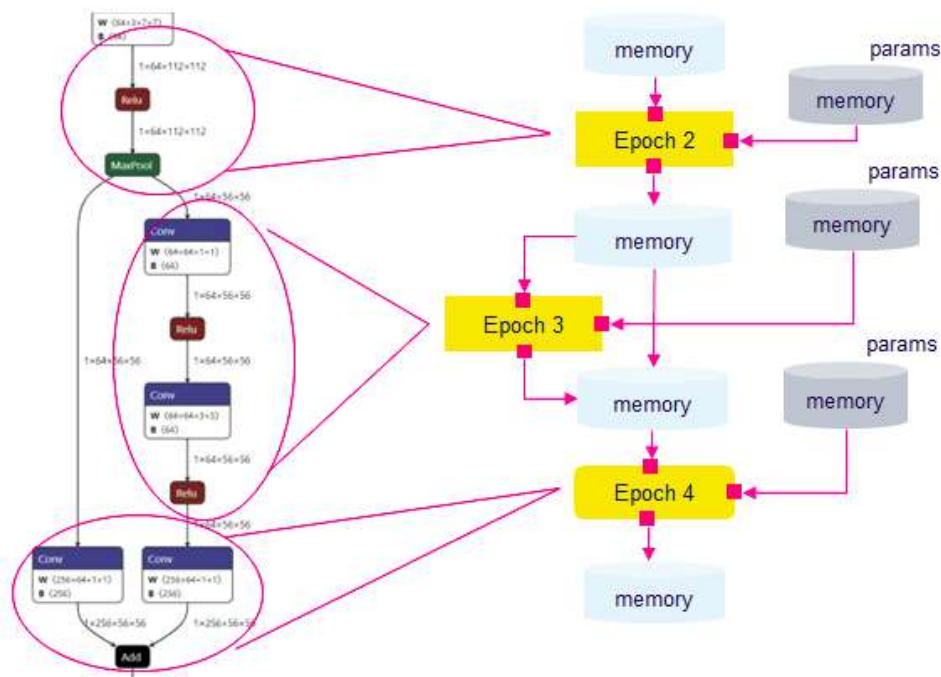


Modèle converti neural-ART NPU

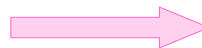


Gemm : General matrix multiplication

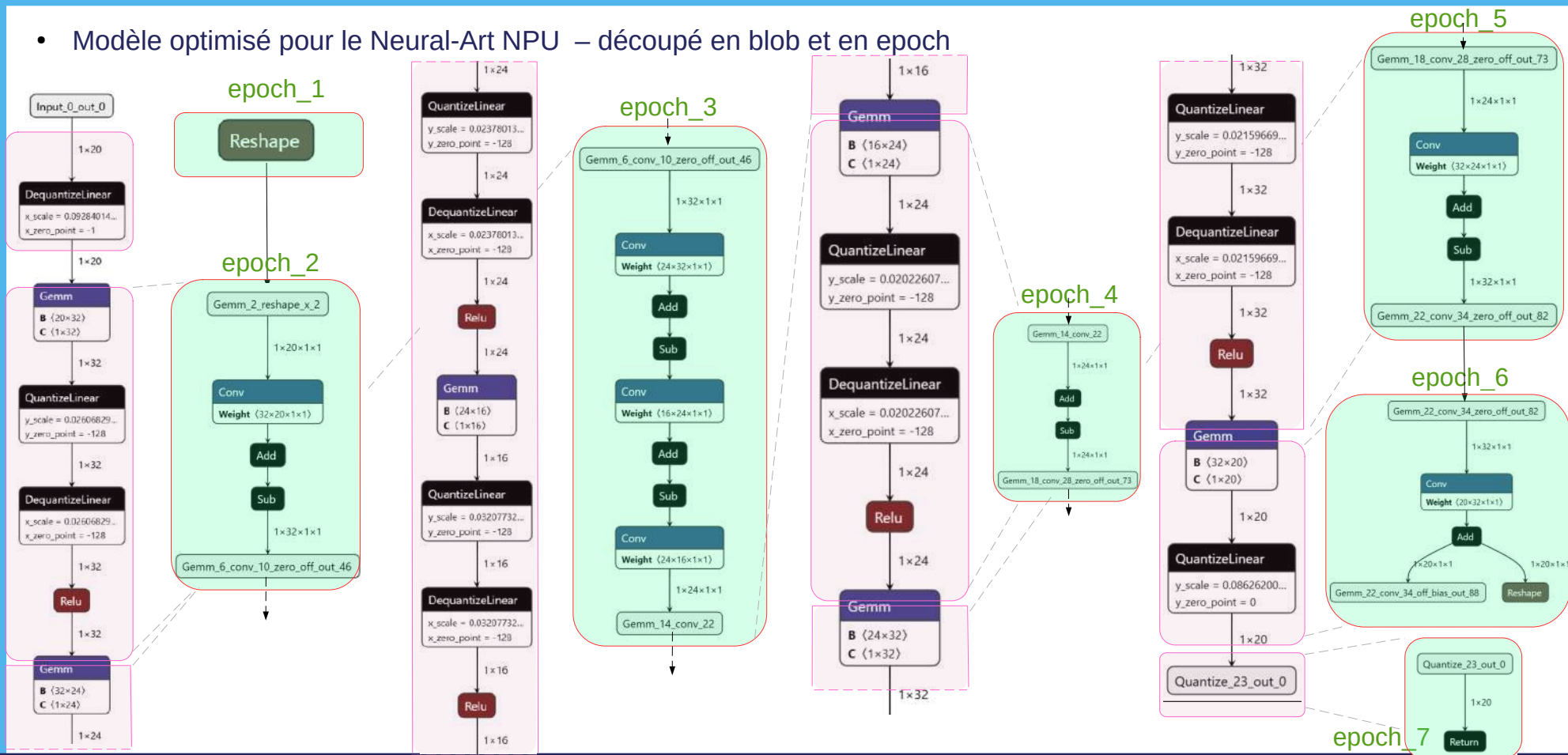
Modèle optimisé pour le Neural-Art NPU – découpé en blob et en epoch



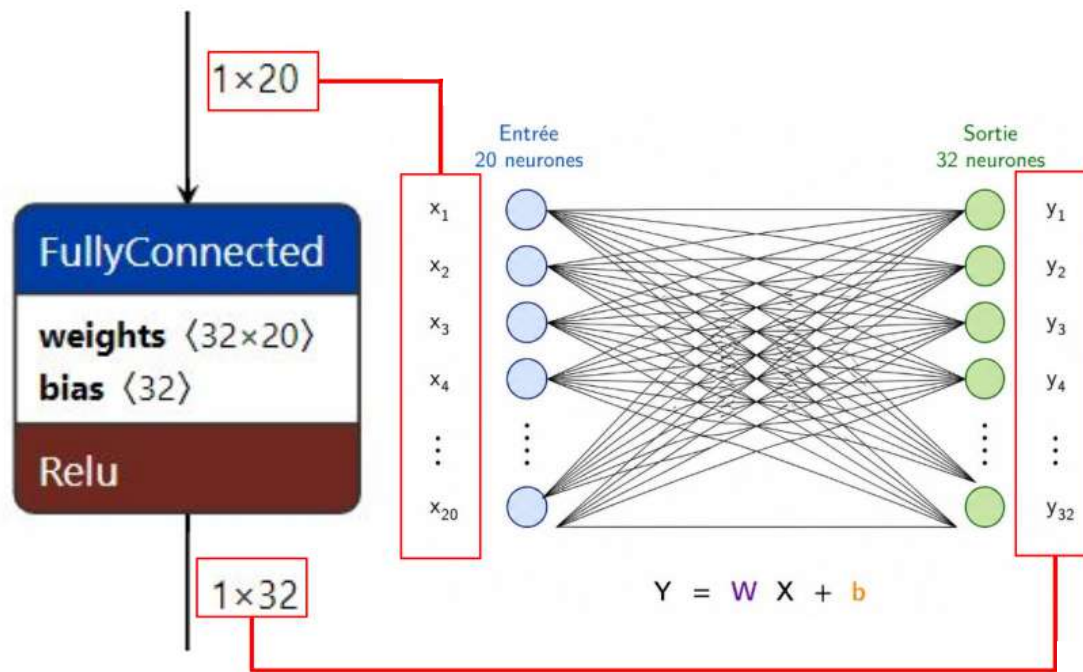
- Input_0_out_0 epoch_0 (NO_X) f Quantize_23_out_0



- Modèle optimisé pour le Neural-Art NPU – découpé en blob et en epoch



- Modèle optimisé pour le Neural-Art NPU



Multiplication accumulation

- Modèle optimisé pour le Neural-Art NPU

Fichier de reporting de conversion de modèle

```

model_name      : autoencoder_v3_int8
model_hash      : 0xf3b59367384741ae3296455ddae0c17d
params #        : 3,732 items (4.08 KiB)
-----
input 1/1       : 'Input_0_out_0', int8(1x20), 20 Bytes, QLinear(0.092840143,-1,int8), activations
output 1/1      : 'Quantize_23_out_0', int8(1x20), 20 Bytes, QLinear(0.086262000,0,int8), activations
macc            : 3,722
weights (rw)    : 4,561 B (4.45 KiB) (1 segment) / -10,367(-69.4%) vs float model
activations (rw): 80 B (80 B) (1 segment) *
ram (total)     : 80 B (80 B) = 80 + 0 + 0
  
```

Epochs details

```

-----
Total number of epochs      5
>> pure software (SW) epochs 0
>> hybrid epochs (using both software and hardware) 0
>> pure hardware (HW or EC) epochs 5 (implemented in 0 epoch controller blobs/meta epochs)
  
```

epoch ID	Type	Details
epoch_2	HW	
epoch_3	HW	
epoch_4	HW	
epoch_5	HW	
epoch_6	HW	

Fichier de reporting d'analyse du réseau

name	epoch	offset	size	shape	format
Input_0_out_0	0->2	0->20	20	[1,20]	STAI_FORMAT_S8
Gemm_2_reshape_x_2	0->2	0->20	20	[1,20,1,1]	STAI_FORMAT_S8
Gemm_6_conv_10_zero_off_out_46	2->3	48->80	32	[1,32,1,1]	STAI_FORMAT_U8
Gemm_14_conv_22	3->4	0->48	48	[1,24,1,1]	STAI_FORMAT_0 (16blits)
Gemm_18_conv_28_zero_off_out_73	4->5	48->72	24	[1,24,1,1]	STAI_FORMAT_U8
Gemm_22_conv_34_zero_off_out_82	5->6	0->32	32	[1,32,1,1]	STAI_FORMAT_U8
Quantize_23_out_0	6->7	32->52	20	[1,20]	STAI_FORMAT_S8
Gemm_22_conv_34_off_bias_out_88	6->7	32->52	20	[1,20,1,1]	STAI_FORMAT_S8

48B : Buffer d'activation le + grd : 24 valeurs en 16 bits

+32B : sorties intermédiaires

=>Double buffering avec alternance entre les 2 zones

epoch	macc	node
epoch_0	0	HW
epoch_1	0	HW
epoch_2	672	HW
epoch_3	1576	HW
epoch_4	24	HW
epoch_5	800	HW
epoch_6	650	HW
epoch_7	0	HW
macc=3722		

Modèle optimisé par le Neural-Art NPU : Adéquation des ressources

For the STM32N6xx device, the Neural-ART 14 configuration includes the following processing units to address a wide range of quantized convolutional neural network (CNN) models:

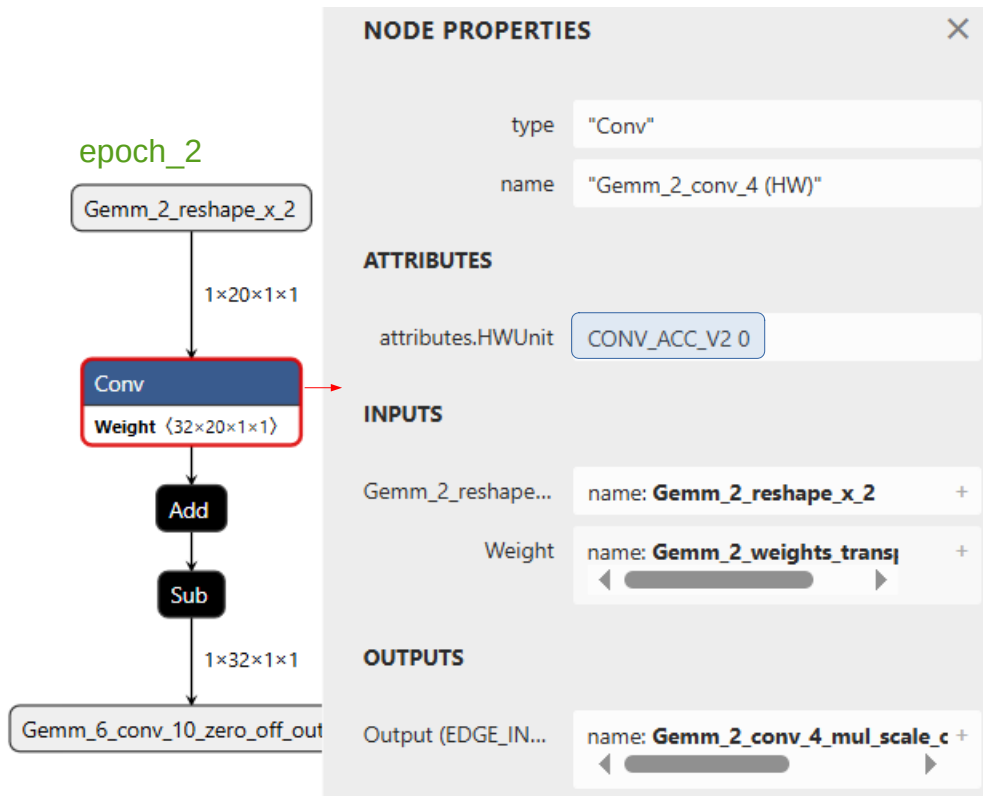
Processing unit	description
CONV_ACC (x4)	The main processing unit performing the convolution operations with up to 72 8x8 multiply accumulate operations per cycle (or 18 16x16 macc/cycle). This allows a theoretical peak processing of 72 x 4 x 2 (addition and multiplication) = 578+ GOPs @1GHz or 600+ GOPs including the operations from the ACTIV/ARITH/POOL units.
POOL_ACC (x2)	Perform the pooling operations like local 2D windowed (NxN), min, max, average pooling as well as global max, min, or average pooling.
ACTIV_ACC (x2)	Perform the activation functions associated with the convolutional neural networks: Logistic, TanH, ReLU, PReLU, etc.
ARITH_ACC (x4)	Perform the arithmetic operations: element-wise addition/subtraction/multiplication or any other affine operation.
STREAM_ENG (2x5)	5 per port, Key unit (smart half-DMA engine) to fetch/push data to/from memory subsystem.

- All processing units are based on integer arithmetic with 8-bit, 16-bit, and 24-bit data path widths. They support both signed and unsigned formats.
- They are designed to support a *fixed-point format*, enhanced by shifters with rounding and saturation capabilities to adjust values. There is no hardware support for floating-point arithmetic (no floating-point unit).
- The absence of floating-point support means that the NPU relies solely on fixed-point arithmetic. This is generally more efficient in terms of power consumption and computational speed.

Modèle optimisé par le Neural-Art NPU : Adéquation des ressources

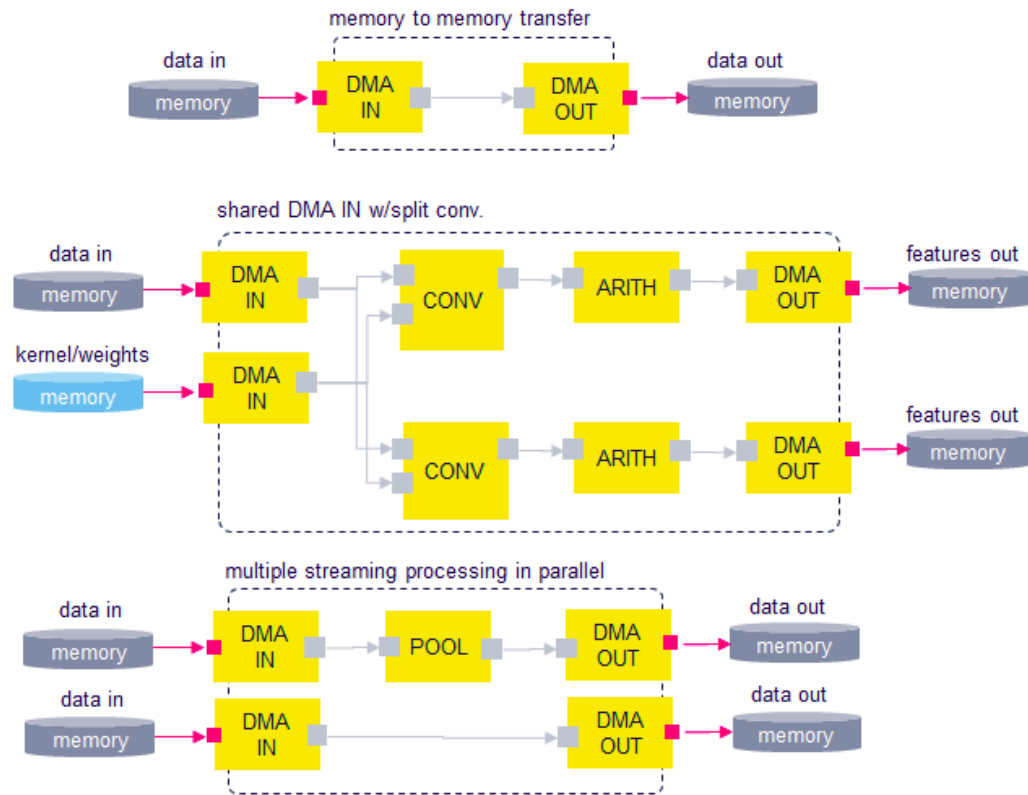
Specific unit	description
DEC_UNIT (x2)	Perform the decompression on-the-fly of the compressed kernel weights (Data are compressed offline).
RECBUF_UNIT (x1)	Internal unit to insert a partial buffer between two processing units avoiding potential deadlocks, starvation.
EPOCH_CTRL (x1)	A finite state machine which is able to decode simple microinstructions from a "command stream" (blob object) to execute a set of epochs (see " Epoch definition ").
DEBUG & TRACE UNIT (x1)	The debug unit allowing to collect various internal signals monitoring various internal signals (a set of specific 32b event counters).

Modèle optimisé par le Neural-Art NPU : Adéquation des unités de traitement



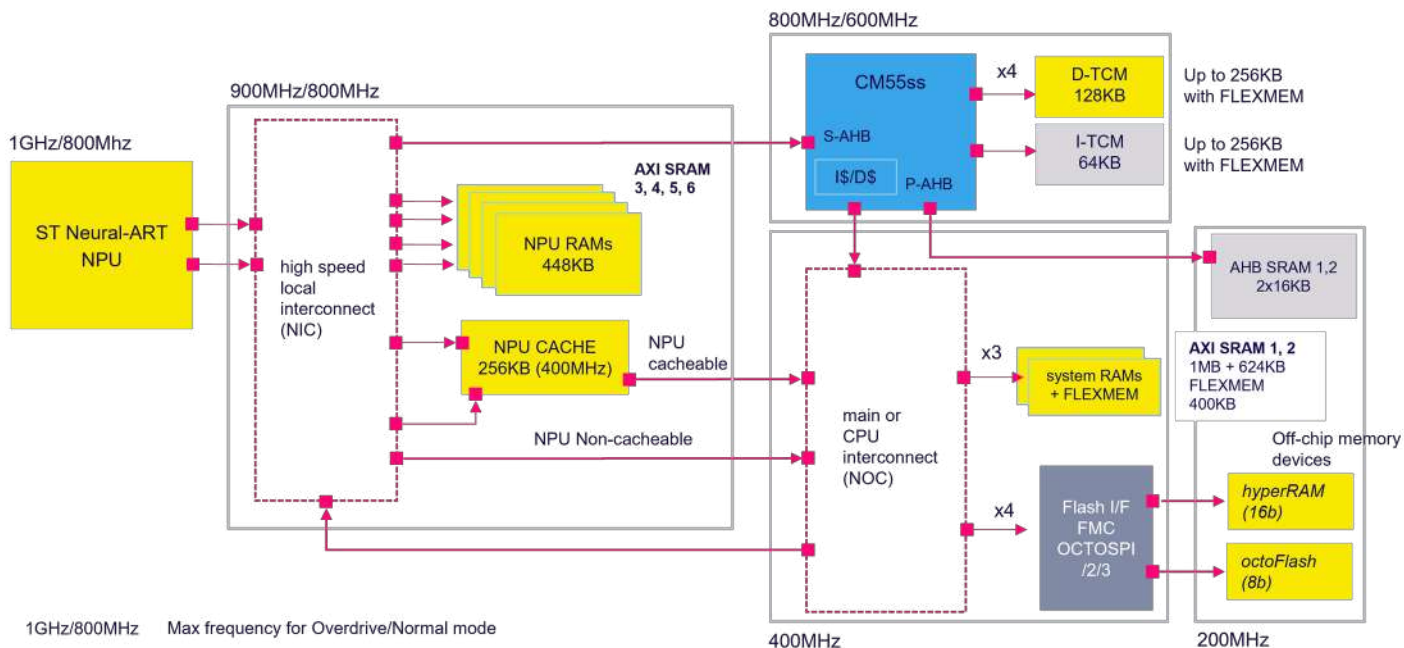
Src: stm32cube
ai studio

Modèle optimisé par le Neural-Art NPU : Adéquation des unités de traitement



Src: stedgeai-dc.st.com

Modèle optimisé par le Neural-Art NPU : Accès ressources mémoires



Src: stedgeai-dc.st.com

Modèle 1 - Détection d'anomalies - Test

2 scénarios

Jeux de données 1

- **310 instances d'anomalie**

- Fuite de gaz progressive



- Incohérence capteur



- Pic de température



Jeux de données 2

- **150 instances d'anomalie**

- Bruit aléatoire faible



- Simulation d'incendie



température, fumée, CO ↑ ; humidité ↓

Métriques d'évaluation



Qualité

- AUC
- Accuracy



Latence

- Temps d'inférence moyen



Mémoire

- Flash / poids
- RAM / activations

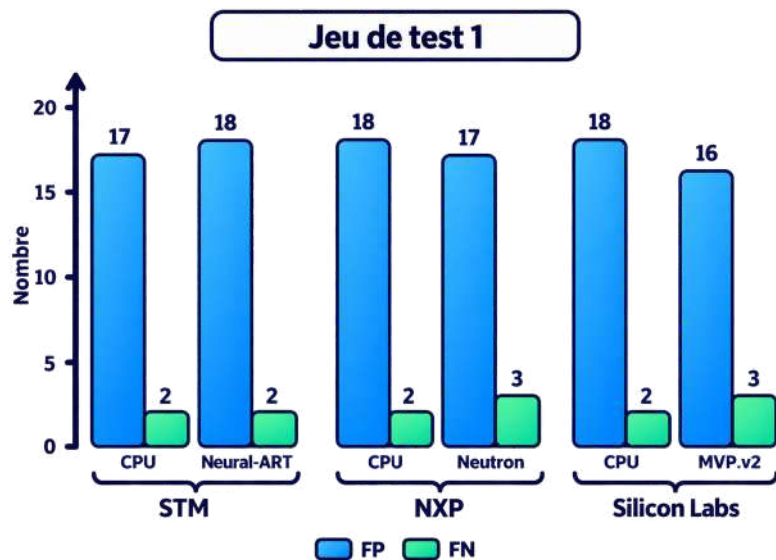


Énergie

- Énergie par inférence
- Mesure sur le domaine d'alimentation

Modèle 1 - Détection d'anomalies – Performances

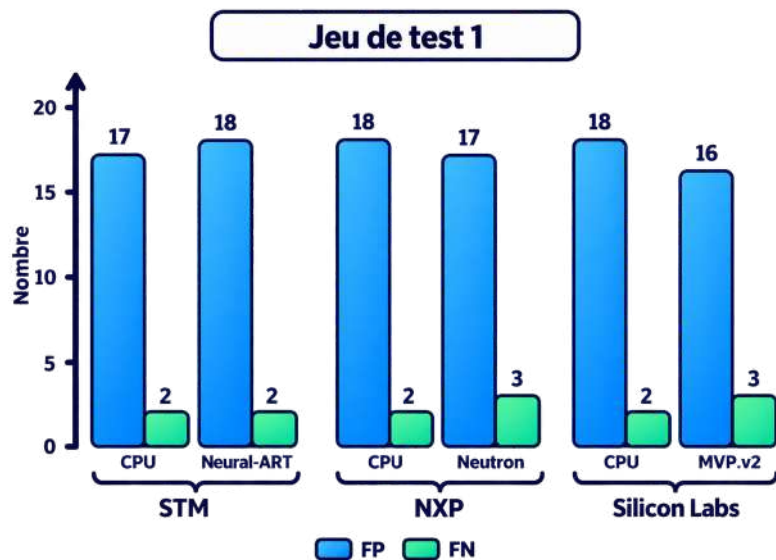
- QUALITÉ** – latence - mémoire - consommation



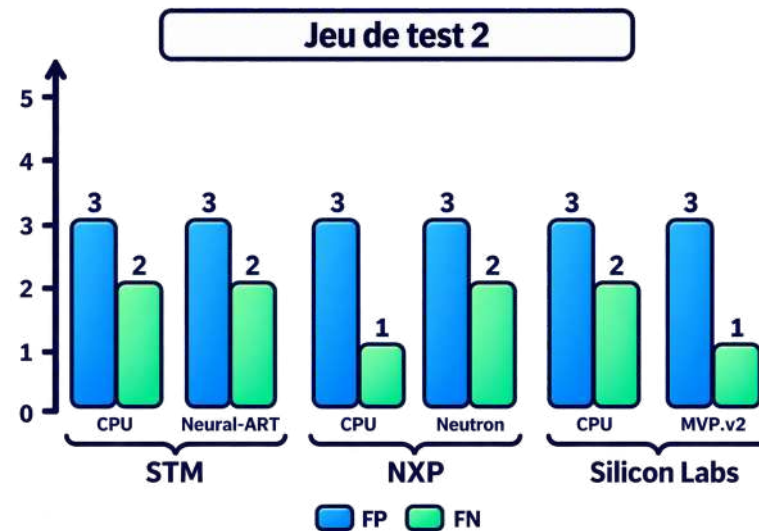
Validation des performances

Modèle 1 - Détection d'anomalies – Performances

- QUALITÉ** – latence - mémoire - consommation



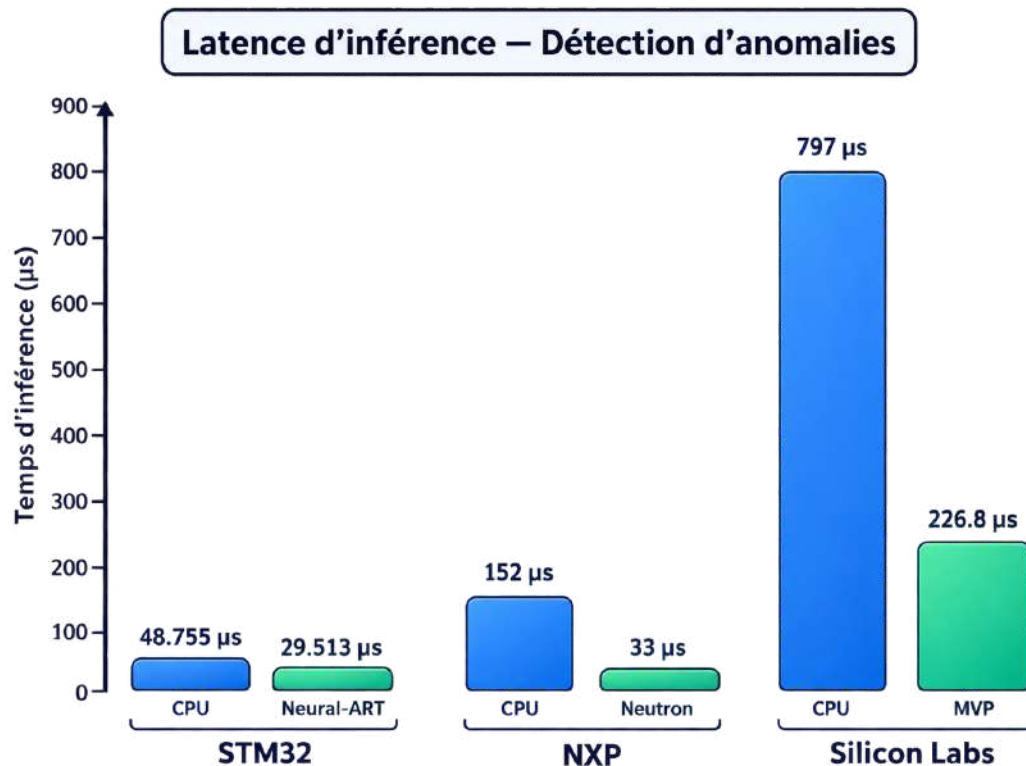
Validation des performances



Validation de la robustesse

Modèle 1 - Détection d'anomalies – Performances

- qualité – **LATENCE** – mémoire - consommation

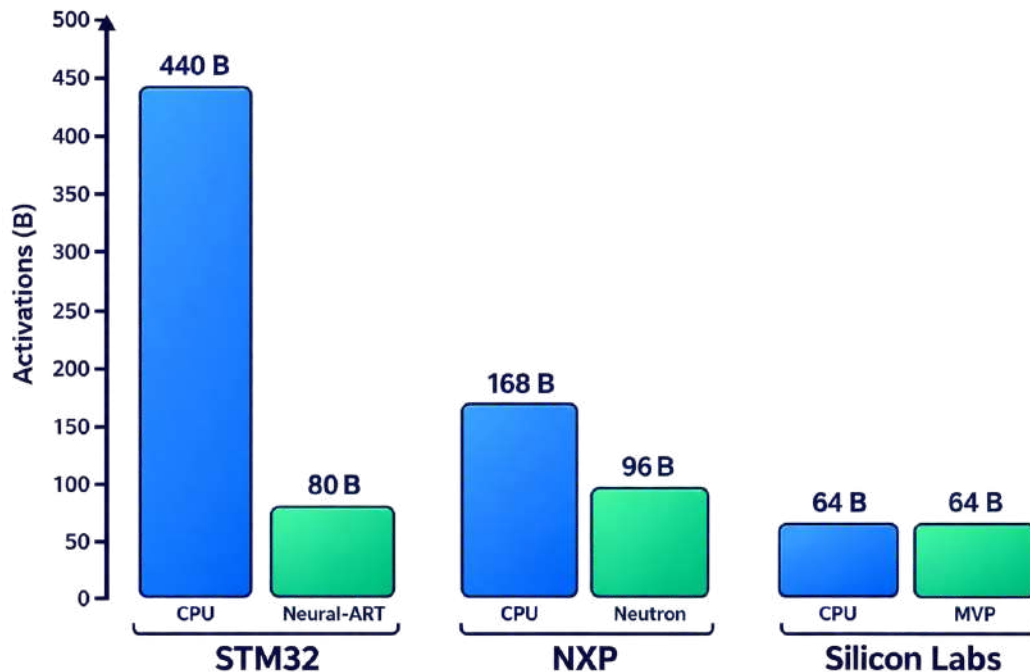


Meilleures performances avec l'accélérateur matériel

Modèle 1 - Détection d'anomalies – Performances

- qualité - latence – **MÉMOIRE RAM** – consommation

Activations — Détection d'anomalies



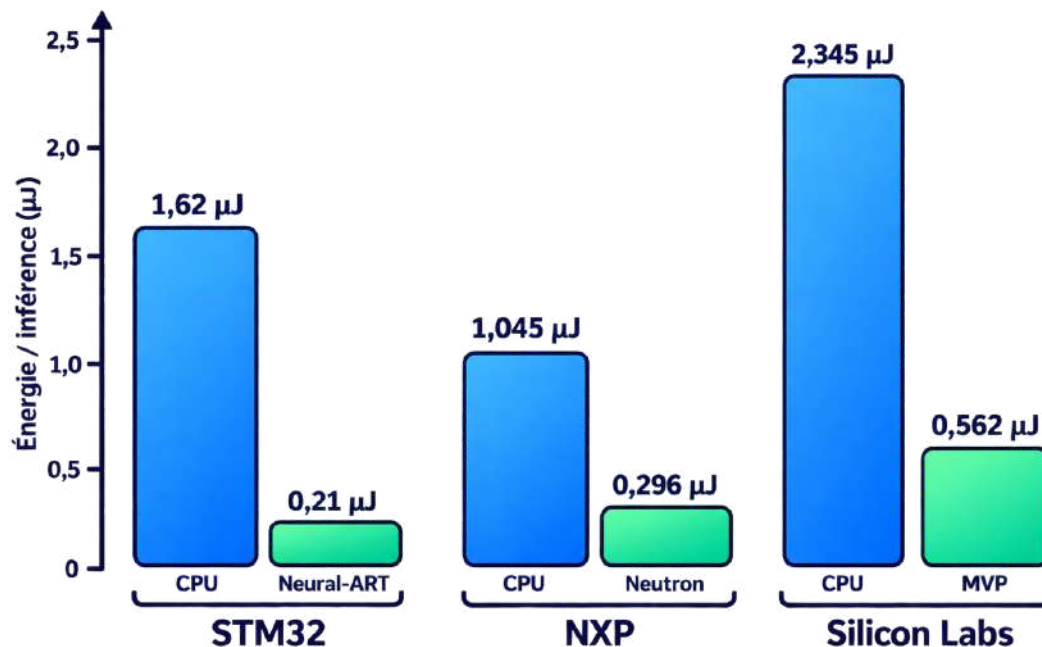
Gain / == avec
l'accélérateur
matériel

Issu des
rapports de
compilations

Modèle 1 - Détection d'anomalies – Performances

- qualité - latence - mémoire – **CONSOMMATION**

Énergie consommée – Détection d'anomalies

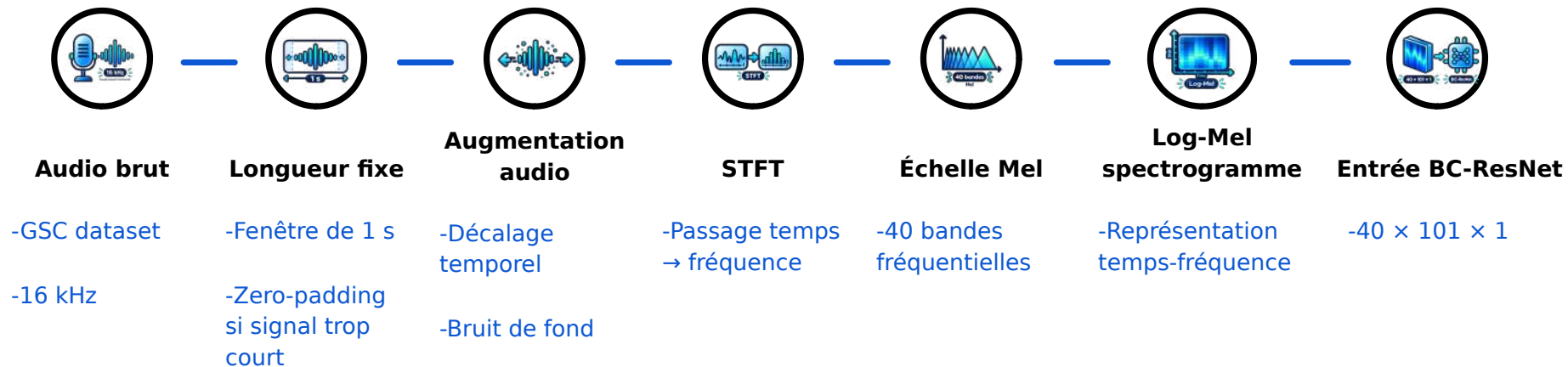


Corrélié avec la latence
Ramené à une détection anomalie

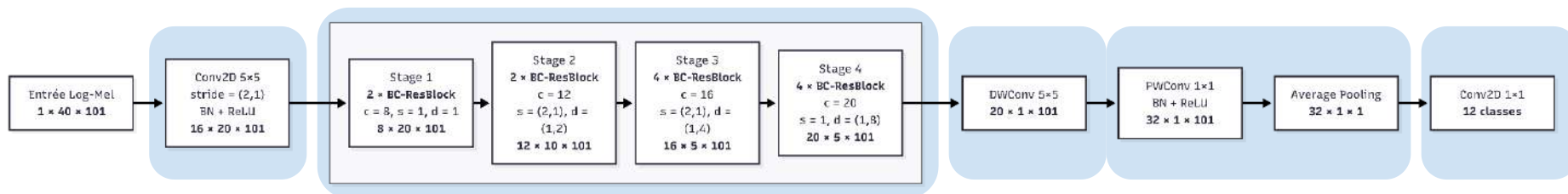
Issu de l'analyseur de puissance Keysight câblé aux cartes

Modèle 2 - Reconnaissance des mots clés BC-Resnet

Phase de pré-traitement des données : <https://github.com/Qualcomm-AI-research/bcresnet>
google speech commands v1.



Modèle 2 - Reconnaissance des mots clés BC-Resnet



BC-ResBlocks

Entraînement

- Loss : Cross-Entropy
- Warm-up + cosine decay
- 200 époques
- ≈ 9 h 30 d'entraînement

Principe

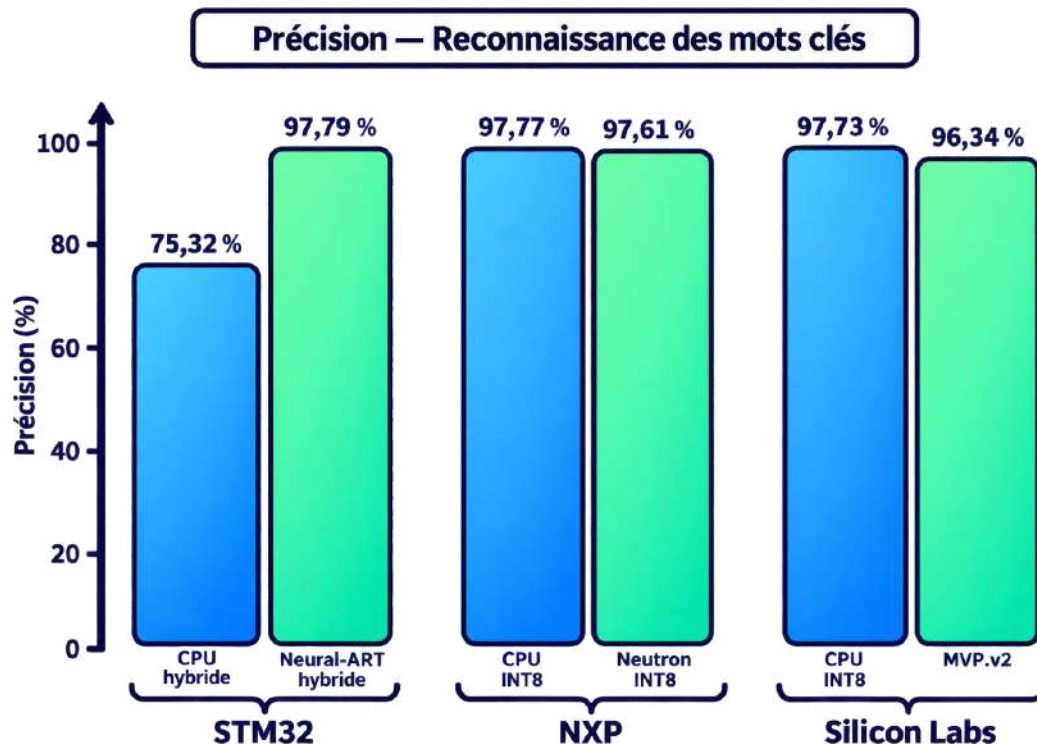
- Évaluation après chaque époque
- Critère principal : **accuracy validation**
- Critère secondaire : **validation loss**
- Checkpoint retenu : époque **190**

Conversion / Quantification

- ONNX → TFLite
- Post-training-Quantization
- Int8
- Représentatif dataset : 1000 données

Modèle 2 - Reconnaissance des mots clés – Performances

- QUALITÉ** – latence - mémoire - consommation

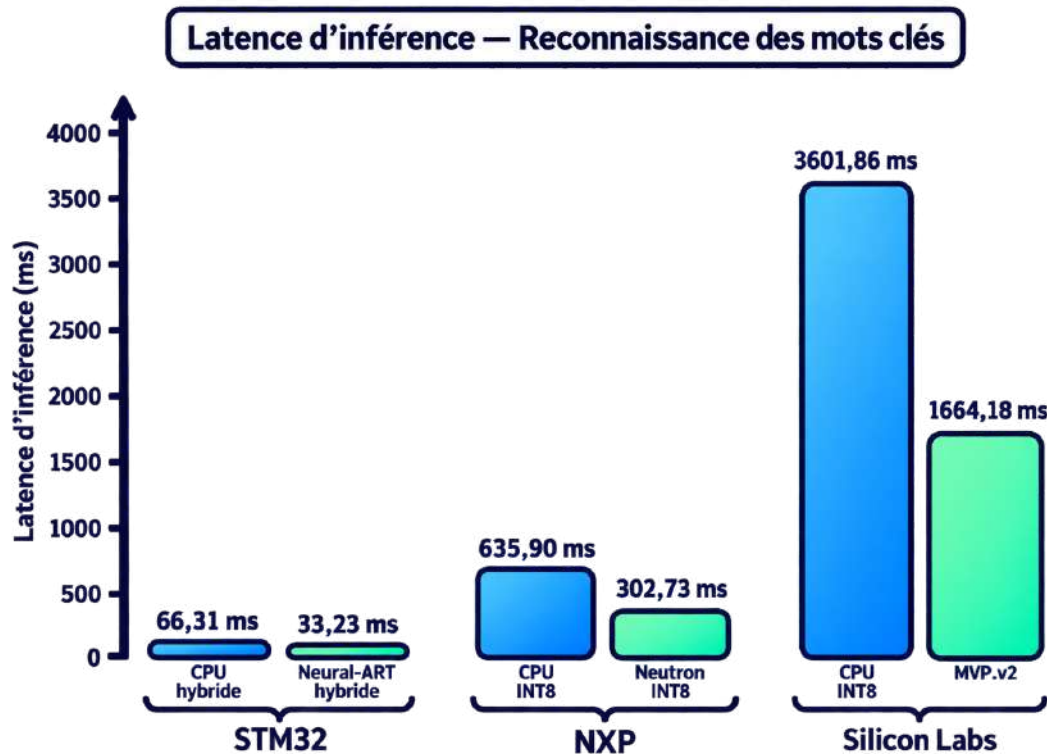


Conservation de la précision

STM32 : CPU – dégradation sur 3 classes inexpliquée

Modèle 2 - Reconnaissance des mots clés – Performances

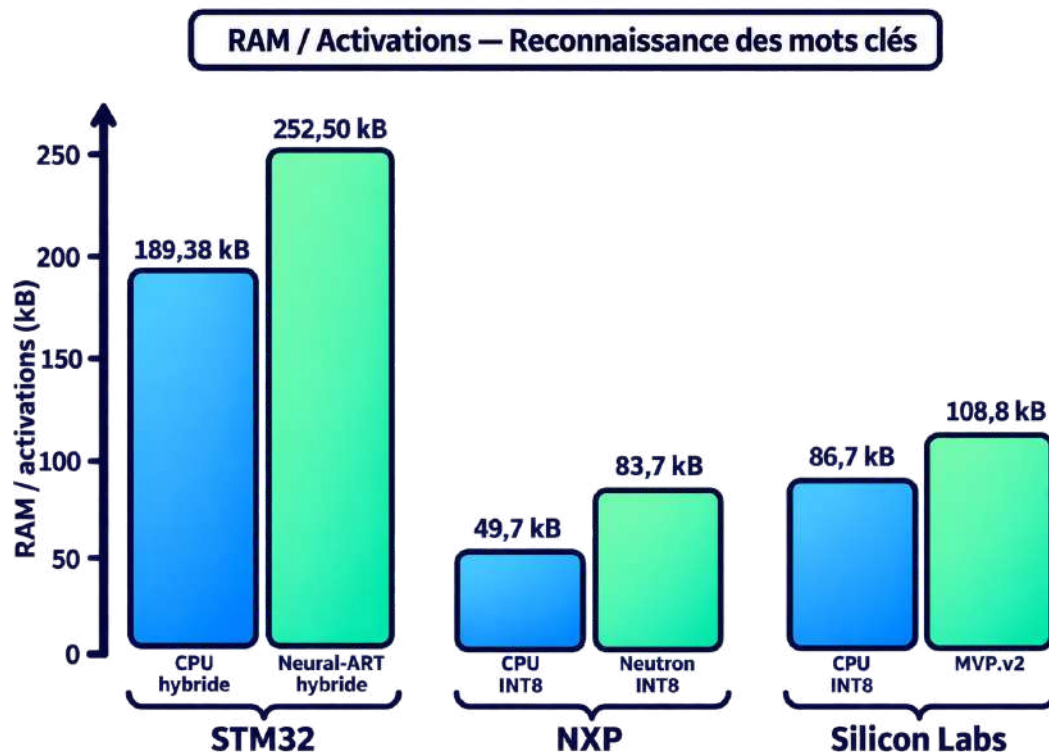
- qualité – **LATENCE** – mémoire - consommation



Gain x2
Écart de
performances
entre les
plateformes

Modèle 2 - Reconnaissance des mots clés – Performances

- qualité - latence – **MÉMOIRE RAM** – consommation



Augmentation
mémoire avec
l'utilisation des
accélérateurs

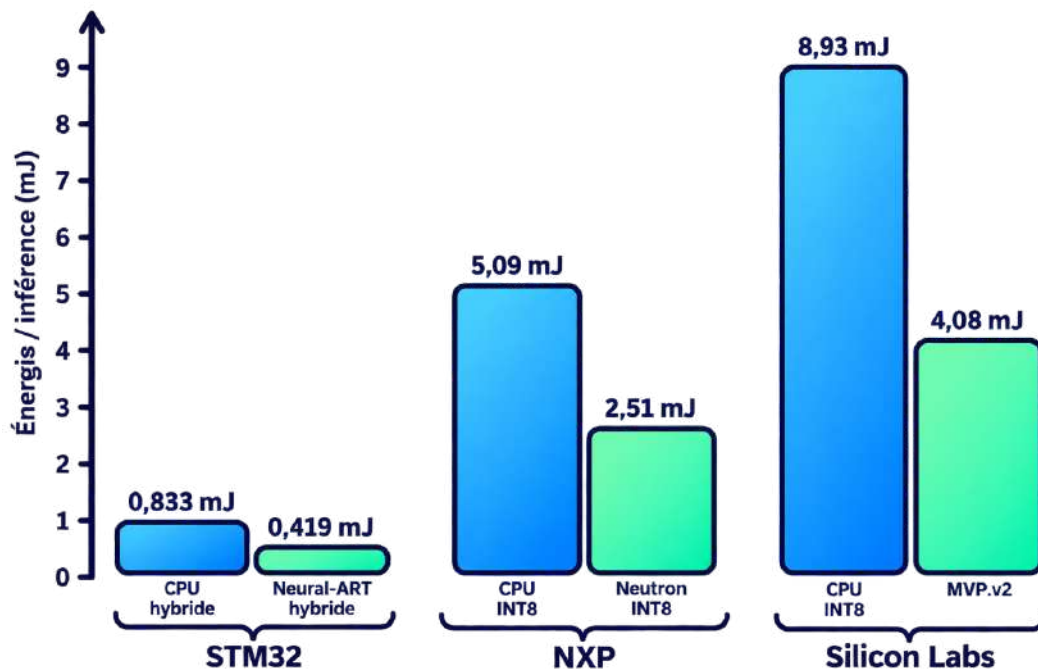
Doivent disposer
de valeurs en
mémoire avec
moins de
chargements à
la volée

Issu des
rapports de
compilations

Modèle 2 - Reconnaissance des mots clés – Performances

- qualité - latence - mémoire – **CONSOMMATION**

Énergie consommée – Reconnaissance des mots clés



Corrélié avec la
latence
Ramené à une
reconnaissance

Issu de
l'analyseur de
puissance
Keysight câblé
aux cartes

Synthèse :



STM32N657

- Faible latence
- Énergie par inférence la plus faible
- Neural-ART efficace mais déploiement plus complexe sur les modèles hétérogènes



MCXN947

- Compromis intermédiaire
- Neutron réduit latence et énergie
- Précision globalement conservée



EFR32MG24

- Latence plus élevé
- Faible puissance pendant l'exécution
- Modèles simples/applications moins contraintes en latence

Modèles perfectibles :

Adaptation nécessaire pour sa conversion optimale.

=> Autoencodeur => rajout de métriques au modèle

=> Forcer certaines allocations d'opérations au CPU et non au NPU.

 **STM32N657**

- Modèle hybride float / INT8
- Convolutions quantifiées

 **MCXN947**

- Modèle INT8
- Désactivation de certaines passes d'optimisation

 **EFR32MG24**

- Modèle INT8
- 3 couches exclues du chemin MVP

- L'exécution de modèles est possible sur les microcontrôleurs à ressources limitées sans détérioré la qualité.
- Adéquation à trouver :

Modèle – Accélérateur matériel embarqué – chaîne logicielle
(NPU, NNA, AI Engine, DSP custom, etc..)
- Outils :
 - Outils des fabricants permettent de savoir quelles seront les ressources allouées et donc si l'implémentation est possible. A voir ensuite les résultats
 - Mises à jour et amélioration continue
- Arrivée continue de nouveaux composants