Designing a Pipelined Architecture to Accelerate the Execution of a Neural Network

Ali Oudrhiri Alix Munier-Kordon [2]

[2]LIP6, Sorbonne Université

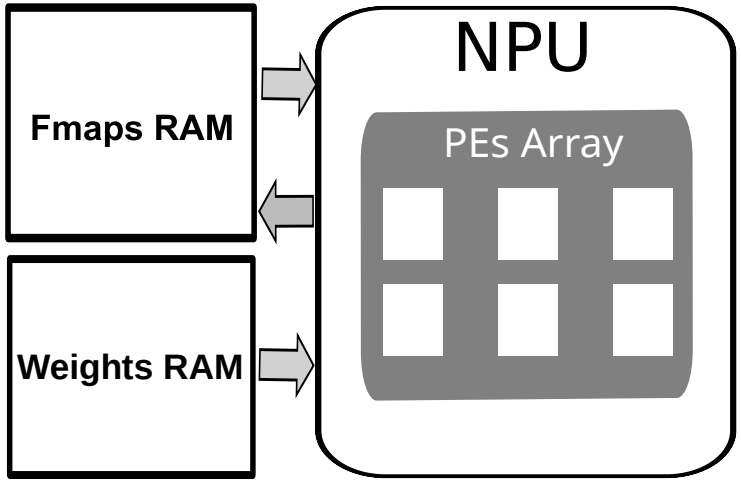
Aussois, June 24, 2024

Presentation Outline

Description of the NPU (Neural Processing Unit)

A prototype AI accelerator designed by STMicroelectronics:

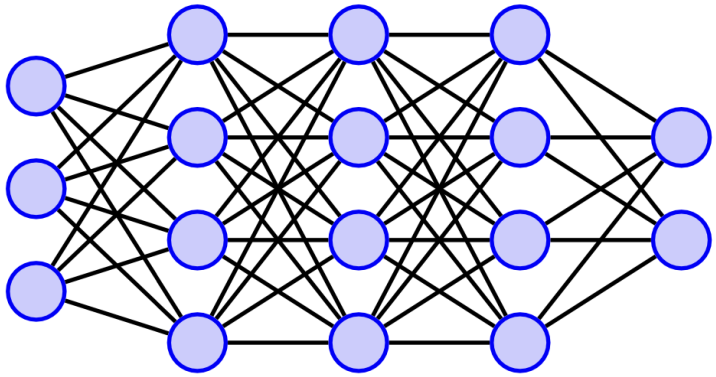
- for inference;
- energy efficient;
- in 40nm CMOS technology;
- aimed at embedded systems.



- Fmaps RAM to store successive images;
- Weights RAM to store weights;
- *N* Processing Elements (PE) units organized as a matrix;
- An evaluation policy for the different layers of a neural network (convolution, Maxpool, fully connected, etc.).

Feed-Forward Neural Networks as Input

- Consists of $\ell \geq 1$ successive layers $\mathcal{L} = [L_0, L_{\ell-1}]$;
- Each node corresponds to a calculation, then a diffusion of the obtained value;
- For all $j \in [1, \ell - 1]$, s_j is the size of layer L_j.



$L_1 \rightarrow L_2 \rightarrow L_3 \rightarrow L_4 \rightarrow L_5$

- [IFMAP,OFMAP] denotes the input and output images of the network;
- For all $j \in [1, \ell - 1]$, I_{j-1} is the image of size s_j that transits from L_{j-1} to L_j.

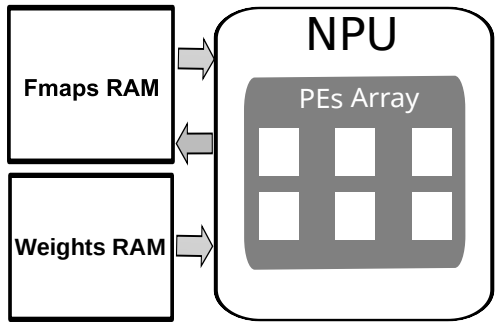
Considered neural networks:

- MobileNet (2017): 27 layers (mostly depthwise, convolution) on images of size $224 \times 224 \times 3$;
- VGG-like, extracted from VGG-16: 11 layers (convolution, maxpool, and fully connected) on images of size $128 \times 128 \times 1$;
- P-Net: 7 layers (convolution, maxpool, and fully connected) on images of size $32 \times 32 \times 1$.

NPU Operation

1. No parallelism between neural networks or successive layers of the same neural network;
2. The execution of a layer (calculations and memory writes) is time-optimized according to the size and type of the layer.

- At startup, the image to be processed (IFMAP) is stored in the Fmaps RAM, and the weights in the Weights RAM;
- When a layer is evaluated, the Fmaps RAM contains both the image being processed and the resulting image;
- Once all layers are evaluated, the resulting image (OFMAP) is in the Fmaps RAM.



Measuring and Evaluating NPU Performance

Two starting points:

1. Description of the execution of a layer;
2. An industrial synthesis tool (Mentor) to measure circuit performance.

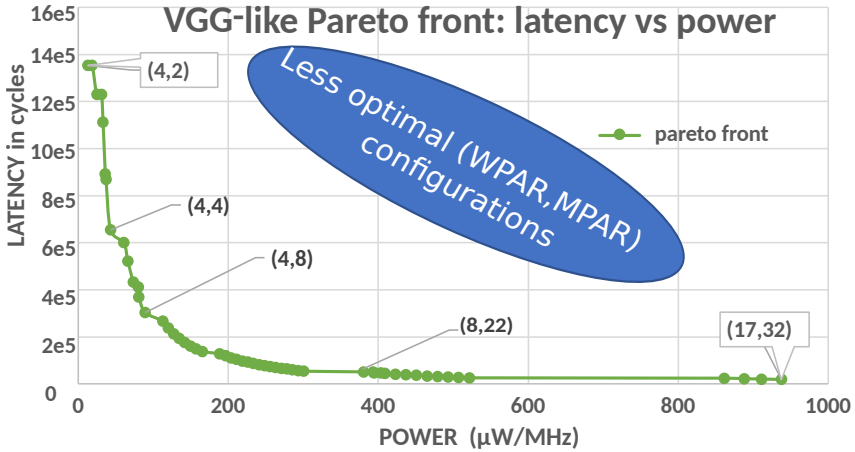
Several performance indicators:

1. **Execution Time** (or latency) of a layer L for an NPU composed of N processing elements: $y(L, N) = \lceil \frac{f(L)}{N} \rceil + c_1$;
2. **Area** function $a(N)$ increasing with N;
3. **Power** The total power $P(L, N) = P_D(L, N) + P_L(N)$
   - Leakage $P_L(N)$ is an increasing function of N;
   - Dynamic power $P_D(L, N)$ is an increasing function of the size of L and N.

Initial Conclusions and Questions

Study of NPU performance alone as a function of $N = WPAR \times MPAR$

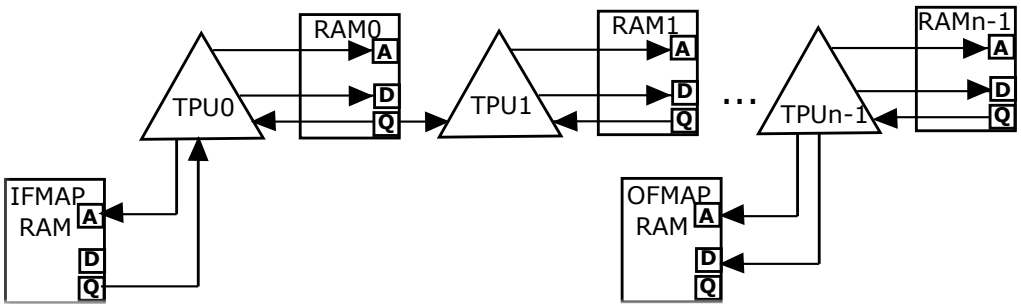
- When N increases, the execution time decreases, and the power consumption increases;
- Beyond a certain value, the execution time decreases very slowly;



1. Very slow decrease in execution time beyond a certain circuit size, at the cost of a significant increase in power and size;
2. Neural networks are evaluated one by one by an NPU: when the images to be processed arrive in a cascade, how can the throughput be improved?

Design and Use of a Pipeline

- n NPUs $G_0, \ldots, G_{n-1}$, composed of $N_0, \ldots N_{n-1}$ processing elements;
- $n + 2$ RAMs composed of:
  1. IFMAP and OFMAP RAMs for input and output images;
  2. Intermediate RAMs $R_0, \ldots, R_{n-1}$ with respective capacities $K_0, \ldots, K_{n-1}$.



The problem then consists of setting the pipeline parameters and placing an application such that:

1. The memory capacities are sufficient to store the intermediate images;
2. The power, area, or energy is minimized while ensuring a certain number of images processed per second.

Design and Use of a Pipeline (continued)

We consider as input:

- A neural network $L_0, \ldots, L_{\ell-1}$;
- A pipeline of n NPUs $G_0, \ldots, G_{n-1}$ and $n+2$ RAM units.

Any placement of the layers π: $\{0, \ldots, \ell-1\} \to \{0, \ldots, n-1\}$ onto the NPUs satisfies:

- Layers L_0 and $L_{\ell-1}$ are executed by NPUs G_0 and G_{n-1};
- Each NPU must execute at least one layer;
- The placement is in increasing order.

Throughput associated with a placement π:

- If the NPU G_i executes layers $[L_g, \ldots, L_d]$, the utilization time of G_i satisfies $p_{i,[g,h]} = \sum_{j=g}^{h} y(L_j, N_i) + c(h-g)$.
- The associated period is $P = \max_{i \in [0,n-1]} p_{i,\pi}$.

Formal Description of the Problem

Input:

- A neural network $L_0, \ldots, L_{\ell-1}$;
- A description of a configurable NPU;
- A maximum period $P^\star$ for executing the neural network;
- An objective function φ to minimize.

Output:

- A description of a pipeline;
- A placement π of the layers on the NPUs

The objective is minimized for an execution period of the neural network on the architecture bounded by $P^\star$.

Sizing the NPU

Using a binary search, the following theorem can be demonstrated:

Theorem

Given a pair of integers (g, d) such that the NPU G executes layers $L_g, \ldots, L_d$, and given $P^\star$ as the upper bound of the period, the minimum number $\widehat{N}$ of processing elements required for G can be calculated in $\mathcal{O}(\log N_{\max})$ time, where $N_{\max}$ is an upper bound on the number of processing elements.

This theorem allows for sizing the NPU based on both the upper bound $P^\star$ and the placement π.

Sizing the RAM

An image I_j is stored in R_i if layer L_j is evaluated by G_i.

Lemma

For any pair $(i, j) \in [0, n-1] \times [1, \ell - 1]$ and any placement π, the intermediate image I_j is stored in R_i if and only if $\pi(j-1) = i$.

Theorem

Given a placement π and an index $i \in [0, n-1]$, let $\bar{j}$ (resp. $\underline{j}$) be the maximum (resp. minimum) index $j \in [1, \ell - 1]$ such that $\pi(j-1) = i$. Then, $\widehat{K}_i(\pi) = \max(s_{\underline{j}}, \max_{j \in [\underline{j}, \bar{j}-1]}(s_j + s_{j+1}))$ is the minimum capacity required for RAM R_i.

The IFMAP and OFMAP RAM must be able to contain the input and output images, respectively.

Exact Resolution of the Problem

- This is an Assembly Line Balancing problem [Boysen et al. 2022];
- Solved by [Held et al. 1963] using a polynomial dynamic programming algorithm.

Construction of a state graph $\mathcal{H} = (V, E, w)$:

- $V = \{s, p\} \cup V_1$ for $V_1 = \{[g, d], 0 \leq g \leq d \leq \ell - 1\}$;
- $E = E_1 \cup E_s \cup E_p$ with
  - $E_1 = \{a = (u, u') \in V_1^2, u = [g, d] \text{ and } u' = [d + 1, m]\}$;
  - $E_s = \{(s, u), u = [0, d] \in V_1\}$;
  - $E_p = \{(u, p), u = [g, \ell - 1] \in V_1\}$.
- $w \mapsto \mathbb{N}$ is defined by:
  - For every arc $a = (s, u) \in E_s$, $w(a) = 0$;
  - For every arc $a = (u, v) \in E_p \cup E_1$ with $u = [g, d]$, $w(a)$ is the restriction of φ to the layers $[L_g, L_d]$ realized on the same layer.

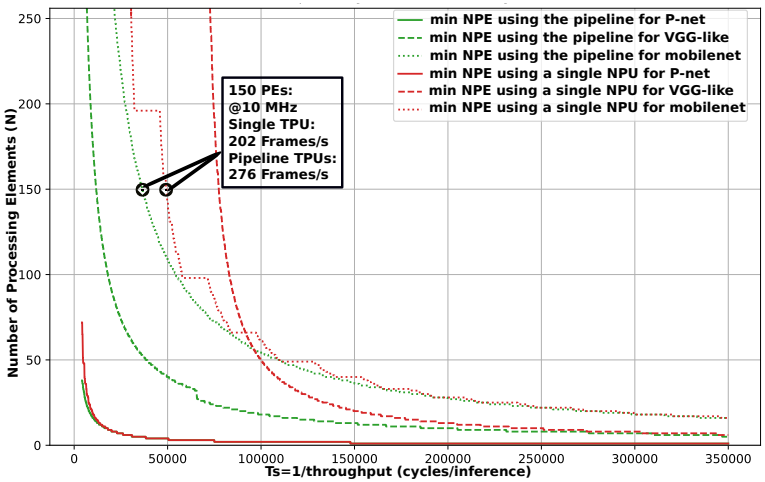
The shortest path from s to p then allows constructing both a pipeline and a placement of layers with a period of at most $P^\star$ that minimizes φ.

Experimental Results: Minimum Period of an NPU and a Pipeline

		Pipeline Solution	
NN	Single NPU	Min. P^*	Corresp. Lat.
Mobilenet	26810	8400	176400
VGG-like	117890	34000	102000
P-Net	2720	1470	5880

- For a single NPU, the latency is equal to the period;
- Throughput is increased by 3.2 times, 3.5 times, and 1.85 times for the 3 neural networks;
- This increase comes at the expense of latency (6.6 times higher for Mobilenet).

Experimental Results: Minimization of the Number of Processing Elements as a Function of the Period

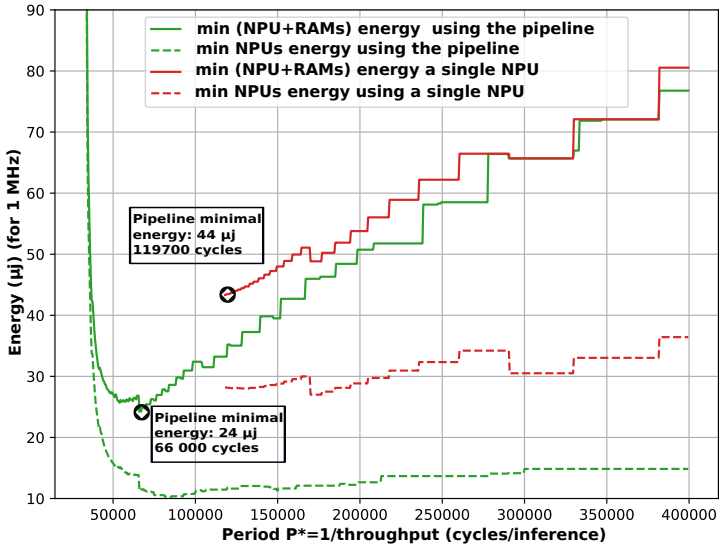


- For all instances, the total number of processing elements is always lower for the pipeline solution;
- For a single NPU, the processing elements are underutilized for layers with few vertices.

For example, the periods achieved for the execution of MobileNet using 150 processing elements are:

- 275 FPS for the pipeline;
- 202 FPS for a single NPU.

Experimental Results: Energy Minimization for VGG-like



- The single NPU always consumes more than the pipeline, with or without considering the RAM;
- The lowest energy values are obtained with the pipeline.

The two configurations that minimize the NPU+RAM energy are:

- 24μJ for a period $P^\star = 66000$ for the pipeline;
- 44μJ for a period $P^\star = 119700$ for a single NPU.

Conclusions and Perspectives

Conclusions:

- Development of a generic polynomial complexity methodology to optimize performance criteria under pipeline throughput constraints, consisting of AI accelerators to execute a network in inference;
- Connection with a classic combinatorial problem;
- An experimental study demonstrated the method's interest.

Perspectives:

- Consideration of multiple neural networks with different or equal periods;
- Consideration of multiple criteria simultaneously.