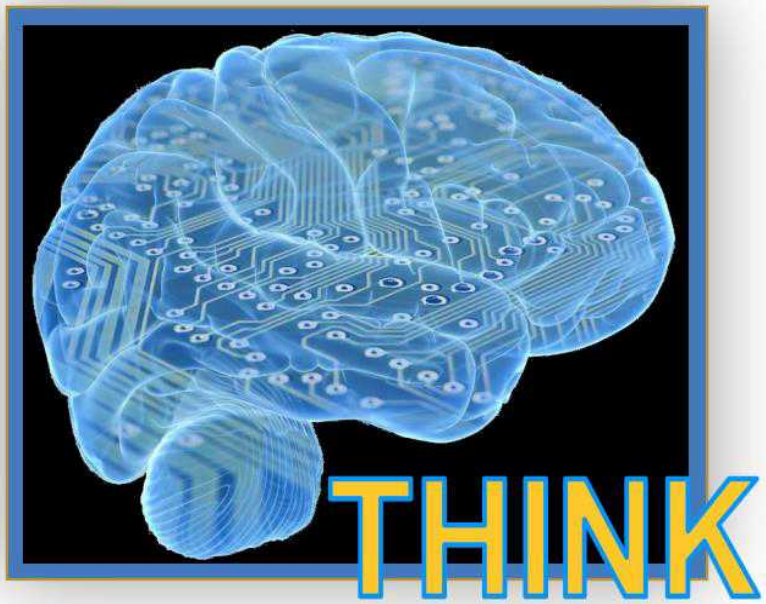
Le projet THINK
Testing Hardware Instantiations of Neural Kernels



J.-P. Cachemiche (CPPM)
V. Gligorov, O. Ledortz (LPNHE)
D. Etasse, J. Hommet (LPC Caen)
F. Bellachia, S. Lafrasse (LAPP
R. Bouet, F. Druillole, A. Rebii (CENBG)
F. Magniette, E. Sauvan (LLR)
G. Aad, Y. Boursier, T. Calvet, E. Fortin, E. Monnier (CPPM)
J. Frontera (CEA IRFU)

Motivation (1)

Evolution des détecteurs

- L’augmentation de la luminosité dans les détecteurs de physique, sous l’effet du bruit de fond et des phénomènes de pile-up, complexifie fortement la tâche des algorithmes de reconnaissance.

- D'une manière générale, ces détecteurs vont requérir plus d'intelligence pour filtrer efficacement les données.

- Cette problématique a donné naissances aux architectures triggerless dans lesquelles les données sont analysées par des mélanges complexes d'accélérateurs et de cartes GPUs.

- Une autre approche consiste à réduire les données au plus près de la source en injectant **plus d'intelligence dans la chaîne d'acquisition hardware**, éventuellement avec des techniques neuronales

Motivation (2)

Savoir faire actuel

- Techniques de bases neuronales et méthodologie connues par certains physiciens, mais peu connues par les ingénieurs

- Principalement implantées sur fermes de calcul, mais pratiquement pas dans les étages amonts, en particulier au niveau hardware

- Calcul neuronal implique la maîtrise de nombreux outils
  - Conda, jupyterlabs, python, matplotlib, pandas, caffe, scikit-learn, tensorflow, keras, pytorch, etc …
  - Langages de haut niveau pour FPGA : OpenCL, HLS, etc …
  - Nombreuses passerelles de translation entre deep learning et inférence, toutes hétérogènes :Vitis, HLS4ML, OpenVino, OneAPI, etc …

Motivation (3)

Besoin d'évaluation globale

- Hiérarchisation des performances
  - Un GPU est-il plus rapide qu'un FPGA ou qu'un MPPA pour une application donnée ?
  - Investissement très important pour maîtriser une technique
  - Entrées-Sorties disponibles
- Limites
  - Type et taille des réseaux implémentables
  - Quelles applications se prêtent à de telles techniques
- Coûts
  - Matériels, outillages
  - Mais aussi en temps de main d'œuvre
    - Learning curve des outils
    - Efficacité
    - Facilité d'usage
- Accessibilité des outils

➡ Déterminant pour choisir une architecture en début de projet

Objectifs du projet

Le projet se déroule en 7 étapes réparties sur 36 mois:

- Une phase de formation pour les ingénieurs et techniciens chargés de la mise en œuvre technique des différentes implantations.

- Dans une seconde phase, nous avons identifié deux applications typiques qui serviront de benchmark aux implémentations hardware.

- Une troisième phase consiste à définir une ou plusieurs structures de réseaux et à effectuer un apprentissage sur ces dernières. Cette phase peut se faire en simulation et ne dépend normalement pas de l'implémentation matérielle future.

- Quatre implémentation matérielles doivent être effectuées en parallèle sur respectivement FPGA, processeur MPPA, processeur neuromorphique et GPU.

- La phase suivante consistera à comparer les performances en terme de coût, de vitesse d'exécution, de consommation, etc … La facilité d'évolution algorithmique et par conséquent la facilité de mise en œuvre des outils de portage fera également l'objet d'une comparaison.

- Enfin le projet se terminera par une phase de diffusion du savoir éventuellement soutenue par plusieurs workshops, ainsi qu'une mise à disposition des outils ou des blocs utilisés dans un espace commun.

Applications en support du projet

- **Le projet Amidex OWEN** (Optimal Waveform recognition Electronic Node) qui consiste à développer un nouvel instrument pour traiter le signal venant d'un détecteur innovant, une TPC sphérique à haute pression. Son but est la recherche d'un phénomène rare tel que la détection directe de matière noire et l'observation de la décroissance double béta sans neutrino. Dans ce contexte, il s'agit de développer un système d'acquisition intégrant un algorithme de problème inverse basé sur les réseaux de neurones pour l'identification des formes d'ondes

- **Le projet RTA** (Real-Time Analysis) dans l'expérience LHCb qui consiste à traiter 40 Tb des donnees par seconde pour n'en garder que 80Gb/s pour une analyse plus profonde offline. Pour ce faire RTA doit à la fois utiliser efficacement les architectures modernes de calcul, et mettre en place des algorithmes avancés de tels que les réseaux neurones.

- **Le projet Amidex AIDAQ** qui consiste à implémenter des algorithmes de reconnaissance neuronale sur FPGA dans le calorimètre à argon liquide d'ATLAS pour réaliser les fonctions de trigger de premier niveau en environnement fortement bruité et avec des niveaux de pile-up variables.

- **Le projet HGCNN** qui consiste à développer des outils d'analyse neuronale pour les données des calorimètres à haute granularité (comme le futur calorimètre HGCal de CMS). Ces outils doivent être intégrés dans des FPGA et fournir des primitives de déclenchement avec des latences de l'ordre de la microseconde.

- **Le projet imXgam** d'imagerie médicale par tomographie. On se propose de débruiter les images par des techniques neuronales

Etat de l'art en physique des particules

- Compilateurs : ex HLS4ML
- Etages amont : Implémentations principalement sur FPGA
  - Applications : Identification de jets, muon trigger, calcul d'énergie, etc …
- Le plus souvent dans les étages aval : GPU, FPGA coprocesseurs

- Hardware/firmware
  - Fast inference of deep neural networks in FPGAs for particle physics [DOI]
  - Compressing deep neural networks on FPGAs to binary and ternary precision with HLS4ML [DOI]
  - Fast inference of Boosted Decision Trees in FPGAs for particle physics [DOI]
  - GPU coprocessors as a service for deep learning inference in high energy physics
  - Distance-Weighted Graph Neural Networks on FPGAs for Real-Time Particle Reconstruction in High Energy Physics [DOI]
  - Studying the potential of Graphcore IPUs for applications in Particle Physics [DOI]
  - PDFFlow: parton distribution functions on GPU
  - FPGAs-as-a-Service Toolkit (FaaST) [DOI]
  - Accelerated Charged Particle Tracking with Graph Neural Networks on FPGAs
  - PDFFlow: hardware accelerating parton density access [DOI]
  - Fast convolutional neural networks on FPGAs with hls4ml
  - Ps and Qs: Quantization-aware pruning for efficient low latency neural network inference
  - Sparse Deconvolution Methods for Online Energy Estimation in Calorimeters Operating in High Luminosity Conditions
  - Nanosecond machine learning event classification with boosted decision trees in FPGA for high energy physics
  - A reconfigurable neural network ASIC for detector front-end data compression at the HL-LHC
  - Muon trigger with fast Neural Networks on FPGA, a demonstrator
  - Autoencoders on FPGAs for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider
  - Graph Neural Networks for Charged Particle Tracking on FPGAs
  - Accelerating Deep Neural Networks for Real-time Data Selection for High-resolution Imaging Particle Detectors [DOI]
  - Ephemeral Learning – Augmenting Triggers with Online-Trained Normalizing Flows
  - Ultra-low latency recurrent neural network inference on FPGAs for physics applications with hls4ml
  - Nanosecond machine learning regression with deep boosted decision trees in FPGA for high energy physics

Organisation projet

Projet démarré en mars 2020

7 laboratoires impliqués

Responsabilités

- **LPC Caen** : Portage sur MPPA, éventuellement sur carte développée par le laboratoire
- **LAPP** : Portage sur processeur neuromorphique
- **LPNHE** : Portage sur FPGA et GPU
- **LP2IB**: Portage sur FPGA Xilinx
- **IRFU/AIM** : Aspects théoriques et formation
- **LLR** : Optimisation Bayesienne
- **CPPM** : coordination du projet, portage sur FPGA Intel et sur GPU

Supports hardwares envisagés

Cartes GPU

Rien d'innovant en première approche

- Déjà très utilisé dans les centres de calcul
- Plutôt utilisé en tant que référence pour les benchmarks
- Possibilité de s'appuyer sur les GPU du mésocentre MUST mis à disposition par le LAPP (cartes Tesla K80 et V100)

Cependant produits dérivés embarquables très intéressants

- Série Jetson de nVidia

Et bientôt ...

Jetson série AGX Orin		Kit de développement Jetson AGX Orin
Jetson AGX Orin 32 Go	Jetson AGX Orin 64 Go	
200 TOPs	275 TOPs	
GPU 1792 cœurs à architecture NVIDIA Ampere (avec 56 cœurs Tensor)	GPU 2048 cœurs à architecture NVIDIA Ampere (avec 64 cœurs Tensor)	

FPGAs

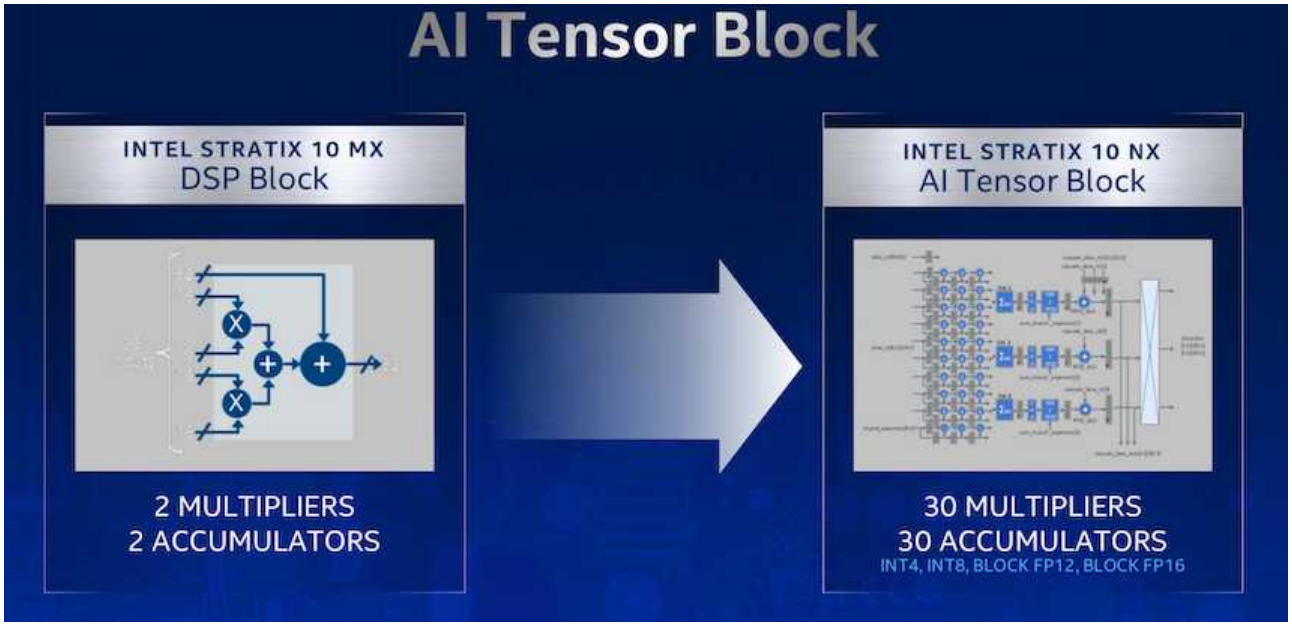
cf_FPN-resnet18_EDD_320_320_45.3G_1.3
cf_FPN-resnet18_Endov_240_320_13.75G_1.3
cf_SPnet_aichallenger_224_128_0.54G_1.3
cf_VPGnet_caltechlane_480_640_0.99_2.5G_1.3
cf_densebox_wider_320_320_0.49G_1.3
cf_densebox_wider_360_640_1.11G_1.3
cf_face-quality_80_60_61.68M_1.3
cf_facerec-resnet20_112_96_3.5G_1.3
cf_facerec-resnet64_112_96_11G_1.3
cf_fpn_cityscapes_256_512_8.9G_1.3
cf_hourglass-pe_mpii_256_256_10.2G_1.3

FPGA déjà adaptés par nature au calcul neuronal

- Nombreux outils ciblés avec bibliothèques des principaux réseaux (ResNet, Mobile net, Yolo, etc)

Sortie de produits plus ciblés :

- Stratix NX d'Intel
  - Tensor blocs intégrés
  - Intégration de larges blocs mémoires à haute vitesse

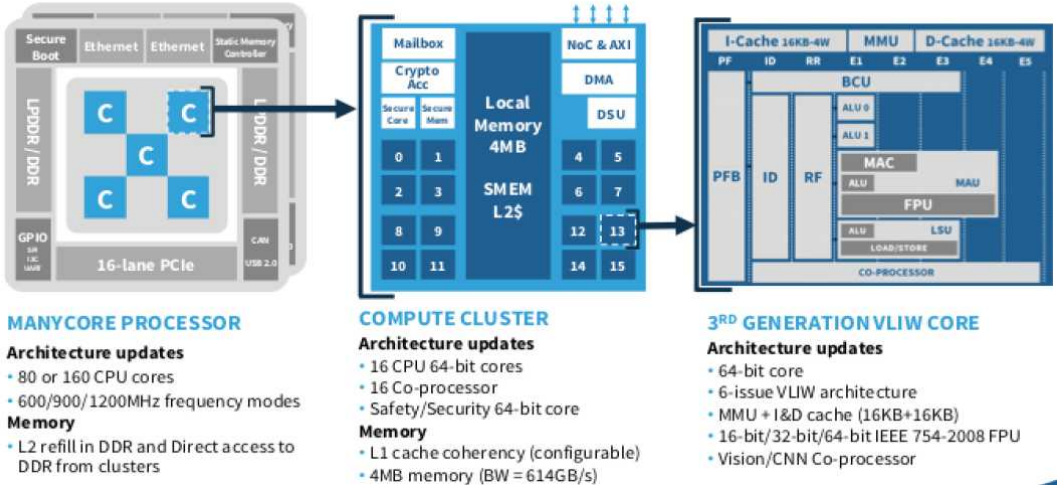


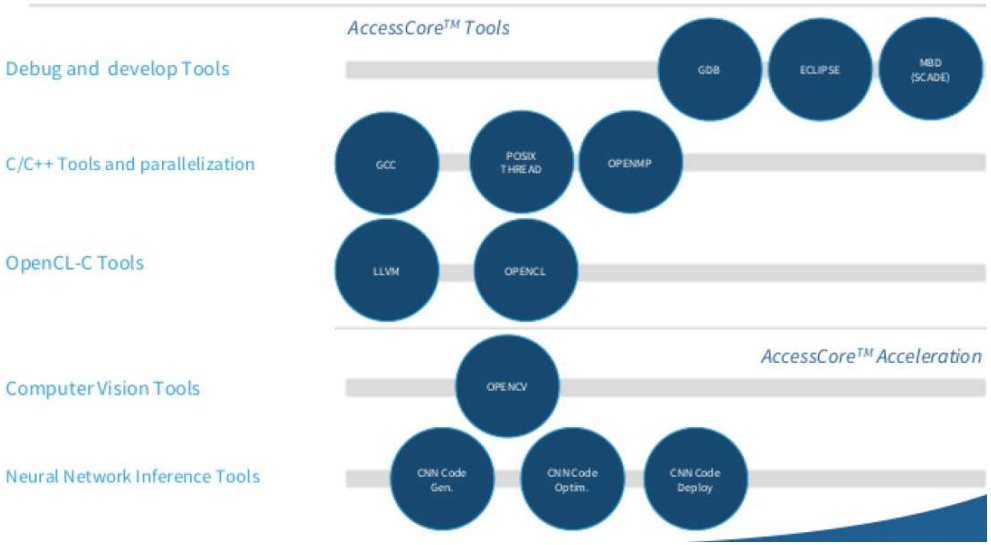
- « Intel says its Stratix 10 NX device is up to 2.3X faster than Nvidia V100 GPUs for BERT batch processing, 9.5X faster in LSTM batch processing, and 3.8X faster in ResNet50 batch processing »

MPPA (Massively Parallel Processor Array)

Coolidge de Kalray

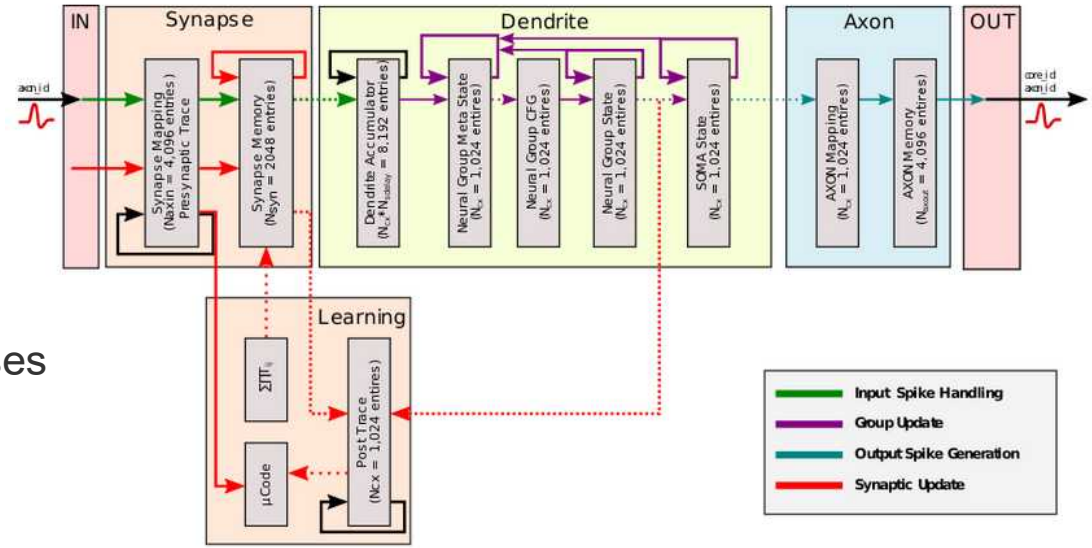
- 80 ou 160 processeurs sur un même chip
- Versions à 288 et 512 cœurs en préparation
- Outils de portage DNN basés sur OpenCL
- Jusqu'à 8 Tflops en FP16
  - Mais puissance conditionnée à l'usage des coprocesseurs (actuellement verrouillés)





		Coolidge-80 v1 @1.2 GHz	Coolidge -80 v2 @1.2 GHz	Coolidge -160 v2 @1.2 GHz	NVIDIA Xavier
INT8	Core	N/A	N/A	N/A	N/A
	Copro	24.6 TOPS	49.2 TOPS	98.4 TOPS	20 + 10
	TOTAL	**24.6 TOPS**	**49.2 TOPS**	**98.4 TOPS**	**30 TOPS**
INT16	Core	2 TOPS	2 TOPS	4 TOPS	
	Copro	12.3 TOPS	24.6 TOPS	49.2 TOPS	10 + 5
	TOTAL	**14.3 TOPS**	**26.6 TOPS**	**53.2 TOPS**	**15 TOPS**
FP16	Core	1.15 TFLOPS	1.15 TFLOPS	2.3 TFLOPS	
	Copro	3.05 TFLOPS	3.05 TFLOPS	6.1 TFLOPS	10 + 5
	TOTAL	**4.2 TFLOPS**	**4.2 TFLOPS**	**8.4 TFLOPS**	**15 TFLOPS**
FP32	Core	1.15 TFLOPS	1.15 TFLOPS	2.3 TFLOPS	
	Copro	N/A	N/A	N/A	1.3 TFLOPS
	TOTAL	**1.15 TFLOPS**	**1.15 TFLOPS**	**2.3 TFLOPS**	**1.3 TFLOPS**
Power		**25W**	**30W**	**60W**	**30W**

Chip neuromorphiques

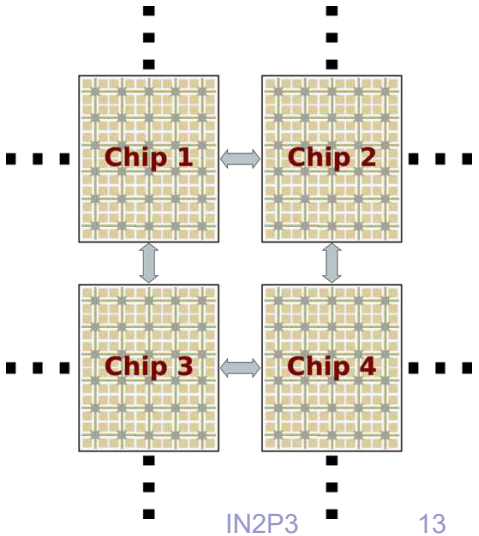


LoiHi d'Intel

- 128 neuromorphic cores + 3 x86 cores
- 130000 neurones, 130 millions de synapses
- Cascadable dans 4 directions
- Spiking Neural Network (SNN) :
  - Suite d'impulsions chronologiquement ordonnées
  - Règle mettant à jours les poids synaptiques en fonction des temps de spikes
  - Pas de descente de gradient.

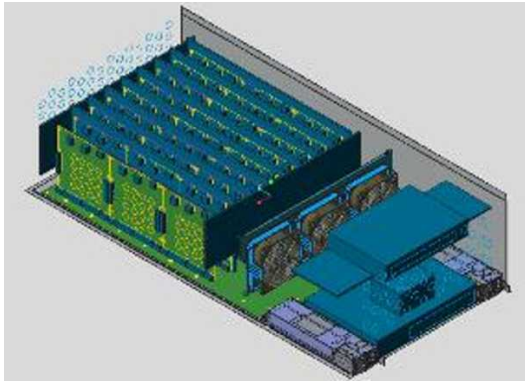
Difficultés

- Pratiquement aucune information d'Intel
- Devons soumettre un projet jugé intéressant pour avoir accès



Systèmes basés sur LoiHi

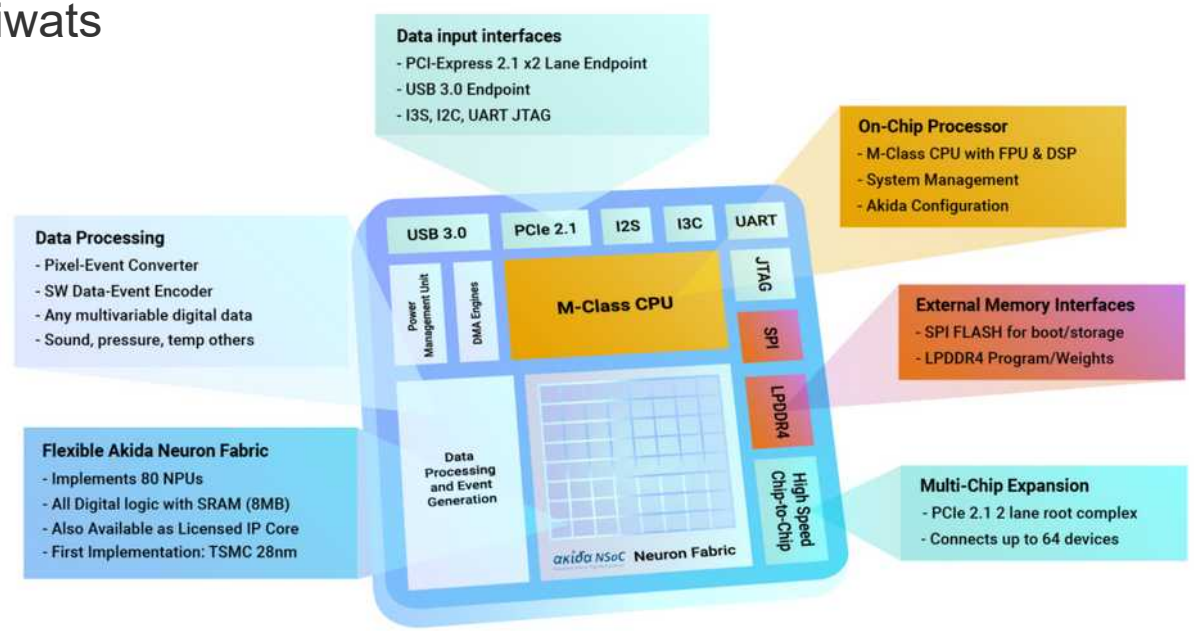
De la clé USB jusqu'à 768 chips interconnectés



Chips neuromorphiques

BrainChip d'Akida

- 1.2 million de neurones et 10 milliards de synapses
- Facteur 10 par rapport à la concurrence
- Très basse consommation : ~ milliwats



Actions menées jusqu'à présent

Cours théoriques

- 12 sessions organisées
- Disponibles sur l'Indico CERN
  - https://indico.cern.ch/category/12078
- Sessions enregistrées (sauf la première)
- Exercices et corrigés

Trainings

- 4 sessions avec industriels : Xilinx, nVidia, Kalray, Intel

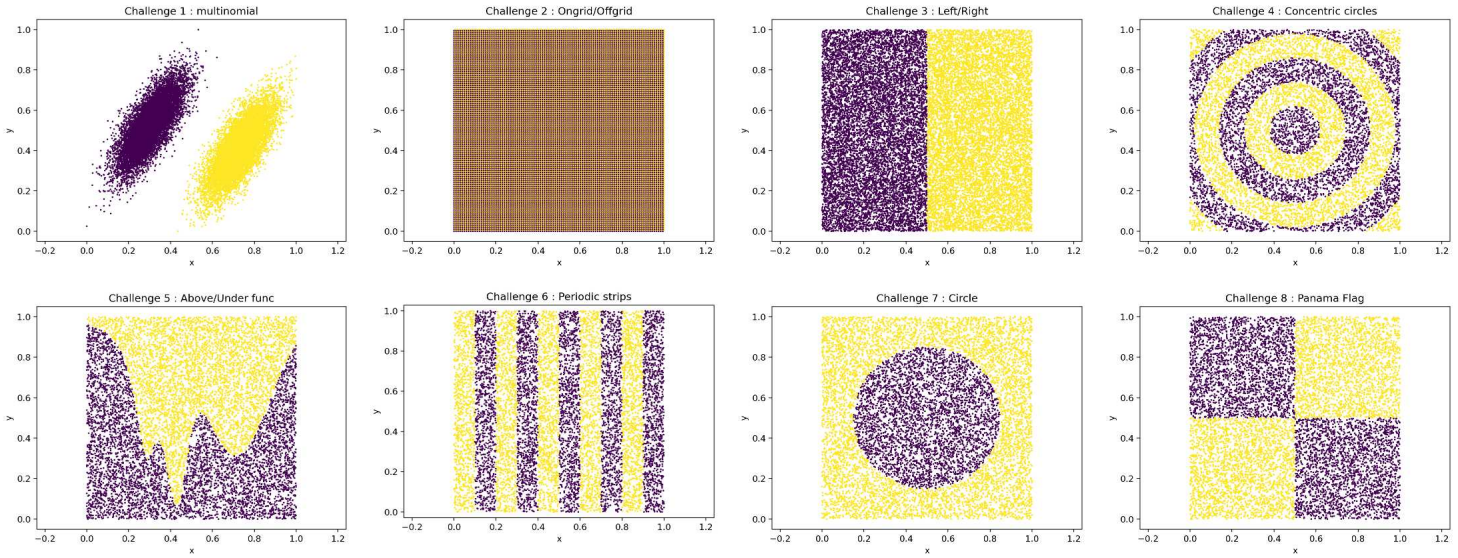
Définition de 2 benchmaks

- Calcul d'énergie pour le calorimètre d'ATLAS : Recursive neural network
- Débruitage d'images ImXgam : Autoencoder
  - Choix basé sur la disponibilité immédiate de données
- Besoin d'un benchmark plus simple pour faciliter la première implémentation cross plateforme
  - Les Challenges

Les Challenges

- Problèmes simples permettant de **mettre en place rapidement les chaînes de développement**

- 8 challenges proposés par Frédéric Magniette



CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8
33	156 101	24 582	659 002	156 101	156 101	156 101	156 101

Nombre de paramètres en fonction du réseau étudié

Premiers résultats

Intel FPGA

Exemple sur challenge 5

- **HLS** : implmentation discrète des couches densess
  - Vanilla : Une implémentation qui ne contient pas d'optimisations dans le code.
  - Pipeline : Une implémentation dans laquelle on pipeline le réseau.
  - Unroll : Une implémentation dans laquelle on ouvre la totalité des boucles.
  - U+p : Une implémentation dans laquelle on ouvre une partie des boucles (les boucles imbriquées) + pipeline du réseau.

	ALUTS	FFs	RAMs	MLABs	DSPs	Débit (FPS)
Ch5 vanilla	1 669	2 193	32	16	2.5	2 556
Ch5 pipeline	2 617	6 074	35	76	2.5	3 839
Ch5 unroll	569 249	196 051	6	40	0	240 000 000
Ch5 u+p	17 560	23 244	507	237	50.5	952 380

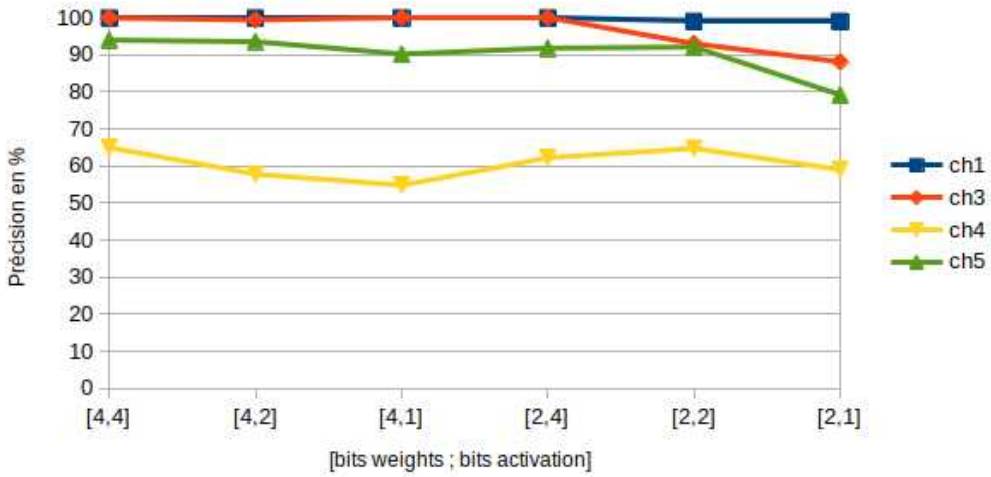
- **OpenAI** : processeur neuronal lisant un jeu d'instructions
  - Just-In-Time : Le réseau est compilé au moment de l'inférence.
  - Ahead-of-Time : Le réseau a été compilé au préalable.
  - -niter : Nombre d'entrées à traiter.
  - -nireq : Nombre d'exécutions en parallèle (sous multiple de niter).
  - -api : Mode dans lequel on utilise l'API (async ou sync).

-api	-niter	-nireq	FPS JIT	FPS AOT
async	12	6	18 662	7 498
async	12	4	13 506	6 903
async	12	3	13 450	6 281
async	12	2	11 075	5 852
async	12	1	6 317	4 116
sync	12	X	14 023	9 596

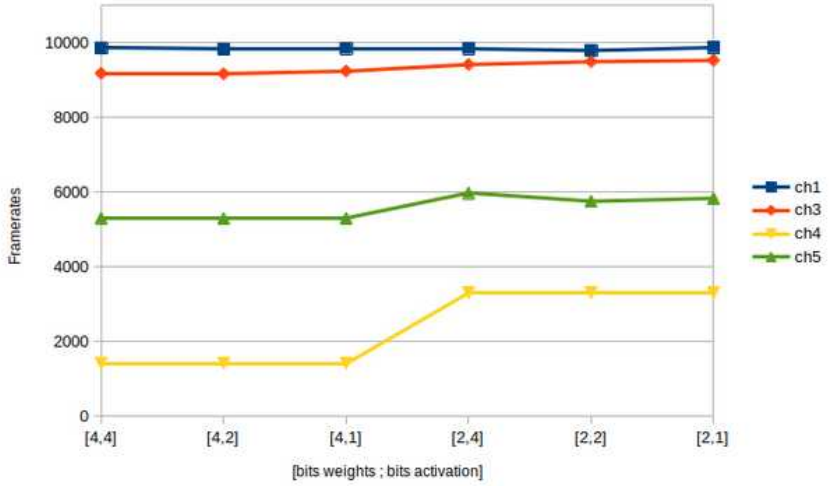
Brainchip

Exemple sur challenge 4

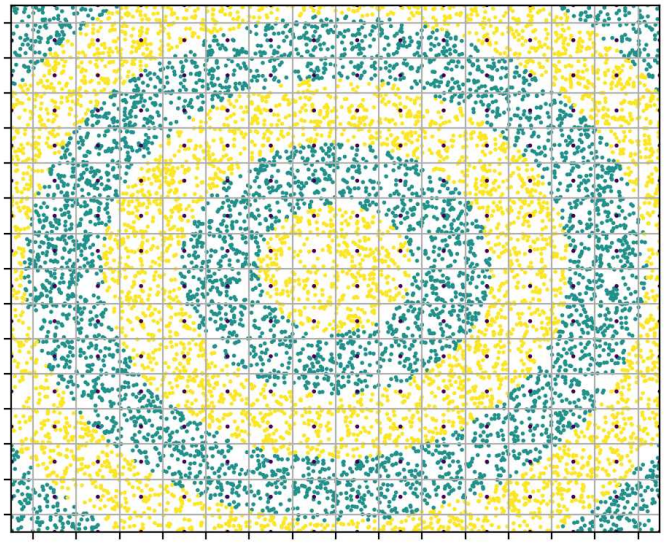
- Perte de précision importante : 70 %
- Lié aux limitations du chip
  - Quantification sur 4 bits
  - Erreur d'arrondi de position



- Performance en FPS similaire à celle d'un processeur neuronal sur FPGA



Challenge 4 : Concentric circles

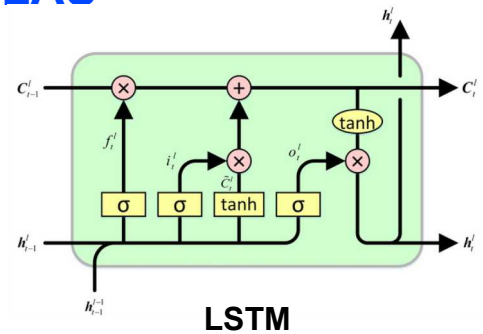


Peu d'avantage tiré du nombre de neurones car chaque couche consomme 1 NPU (57334 neurones) : 1 challenge comporte entre 2 et 6 couches – 80 NPU disponibles

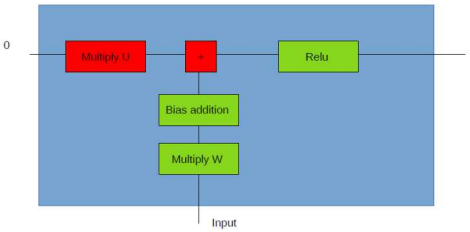
Résultats sur AIDAQ

RNN sur Stratix10 pour calorimètre ATLAS

- 384 channels à implémenter
  - Utilisation de LSTM
    - Fixed point precision
    - LUT for les fonctions d'activation
  - Trop encombrant



LSTM



Vanilla cells

  - Remplacement par Vanilla cells
    - Nombre maximal de channels pouvant être traités avec les ressources DSP :

Hidden Dimensions	Multiplexing														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	411	822	1234	1645	2057	2468	2880	3291	3702	4114	4525	4937	5348	5760	6171
3	213	426	640	853	1066	1280	1493	1706	1920	2133	2346	2560	2773	2986	3200
4	130	261	392	523	654	785	916	1047	1178	1309	1440	1570	1701	1832	1963
5	88	177	265	354	443	531	620	708	797	886	974	1063	1152	1240	1329
6	64	128	192	256	320	384	448	512	576	640	704	768	832	896	960
7	48	96	145	193	242	290	338	387	435	484	532	580	629	677	726
8	37	75	113	151	189	227	265	303	341	378	416	454	492	530	568
9	30	60	91	121	152	182	213	243	274	304	335	365	396	426	457
10	25	50	75	100	125	150	175	200	225	250	275	300	325	350	375
11	20	41	62	83	104	125	146	167	188	209	230	251	272	293	314
12	17	35	53	71	88	106	124	142	160	177	195	213	231	248	266
13	15	30	45	61	76	91	106	122	137	152	168	183	198	213	229
14	13	26	39	53	66	79	92	106	119	132	145	159	172	185	199

Résultats sur AIDAQ

HLS permet d'obtenir des résultats rapidement mais relativement peu optimisés

Travail d'optimisation manuel indispensable

- Nombre d'ALM :
  - Sur HLS : 226 % → 23 %
  - Sur VHDL : 23 % → 18 %
- Nombre de DSPs :
  - Sur HLS : 529 % → 100 %
  - Sur VHDL : 100 % → 66 %

RNN firmware results

- HLS allows fast development and optimisation of the firmware
  - Multiple developments and optimisations of RNN firmware in a short time
  - RNN for INTEL FPGAs implemented in HLS4ML for wider usage
- VHDL is needed to fine tune the design and fit the LAr requirements
- Vanilla RNN firmware produced and fit the requirements with Stratix 10
  - Better performance expected with the Agilex FPGA
  - However still need to test it within the full LASP firmware

	N networks x multiplexing	ALM	DSP	FMax	latency
target	**384 channels**	**30%***	**70%***	**-**	**125 ns**
HLS (no multiplexing)	384x1	226%	529%	-	322 ns
HLS optimized	37x10	23%	100%	414 MHz	302 ns
VHDL optimized	28x14	18%	66%	561 MHz	121 ns

Impact du projet sur choix d'architecture

Faster V3

- Initialement prévu avec un chip Kalray
- Actuellement en migration vers un Jetson Xavier NX

Carte low cost avec 2 FPGAs, un CPU 6 cœurs, et un GPU !

- Deep learning ready

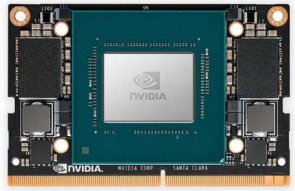
	NANO	TX2 NX	XAVIER NX	XAVIER
AI PERFORMANCE	472 GFLOP	1.33 TFOPS	21 TOPS	32 TOPS
GPU	128 Cores (Maxwell)	256 Cores (Pascal)	384 Cores (Volta) 48 Tensor Cores	512 Cores (Volta) 64 Tensor Cores
CPU	Quad-Cores A57	Quad-Cores A57 Dual-Cores Denver	6-cores Carmel	8-cores Carmel
DDR4	4 Go (64bits)	4 Go (128Bits)	8 Go (128 bits)	32Go (256 bits)
Stockage	16 Go	32 Go	16 Go	32 Go
PCIe	4x Gen2	2* Gen2 1* Gen2	4* Gen4 1* Gen3	8*, 4*, 2*,1*,1* Gen4
POWER	5/10 W	7.5/15 W	10/15 W	10/15/30 W
COST	129 $	199$	479$	

PROCESSOR MANY CORE

KALRAY MPPA COOLIDGE

- 600/900/1200 MHz frequency modes
- 5 or 10 Compute Cluster
- 4 MB -> 1 Cluster (20/40 MB)
- 16 CPU cores 64 bits -> 1 Cluster
- 80 or 160 CPU cores
- 3 or 6 TFLOPS
- 2 * 100 Gbe (x->10Gbe, y->1Gbe, w-> 40Gbe)
- 2 * 8 lane PCIe Gen4
- 5 - 15 W / 5-30 W
- 900 €

David Etasse, LPC Caen



Conclusion

Difficultés rencontrées

- Domaine extrêmement actif: très difficile de maintenir un état de l'art à jour
- Complexité d'appréhension des nombreux outils
- Accès à certains outils ou licences
- En particulier si on ne peut pas se référer à une application avec de gros volumes
- Problèmes de disponibilité de certains membres du projet : habituel dans projets transverses
  - Prolongation du projet d'un an

Premiers résultats

- Même si performances séduisantes pour les chips neuromorphiques, de nombreuses limitations sous-jacentes
- Les FPGA semblent le meilleur compromis en ratio performance /bande passante pour les étages amont
- Les passerelles ou langages de haut niveau ne dispensent pas d'un travail d'optimisation important pour tenir les contraintes temps réel ou de taux d'occupation.

Les résultats sont progressivement mis à disposition sur le site web du projet (think.in2p3.fr) et sur gitlab

- Exemples d'implémentations sur chaque hardware, Méthodologies
- Documentations outils de développement
- Mesures de performances, comparaisons, ...
- Liens utiles, etc ...

More information

Domaine bouillonnant : très difficile de maintenir un état de l'art à jour

CMS

CMS HGCAL neural network architectures and inputs

J.-B. Sauvan
LLR CNRS / École Polytechnique

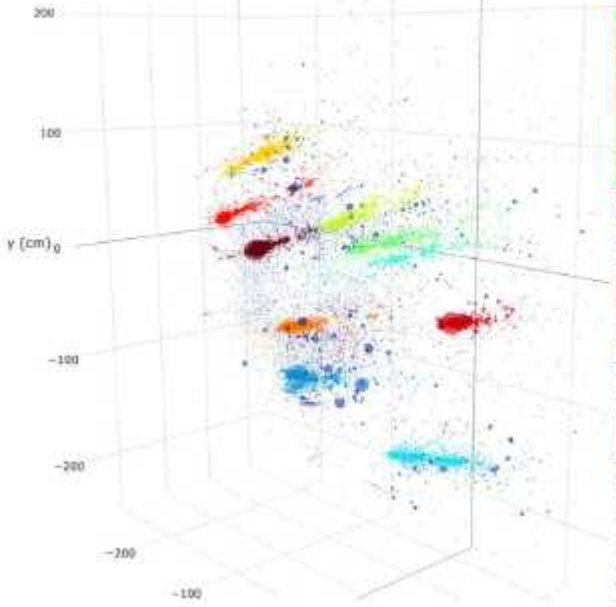
15/06/2020

CMS

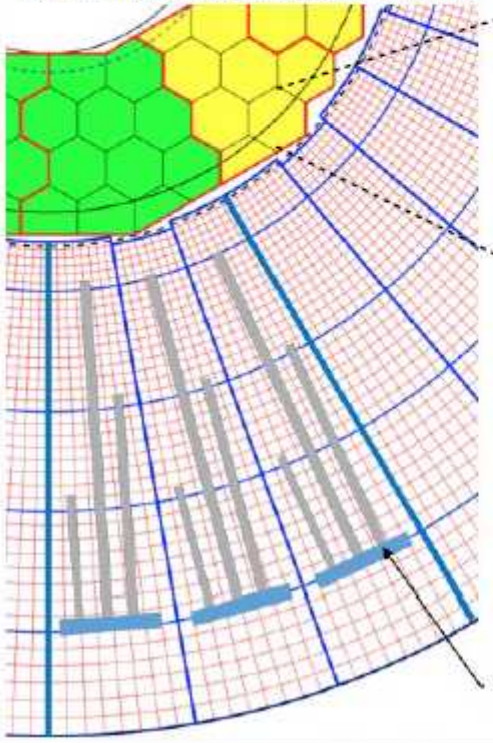
Inputs

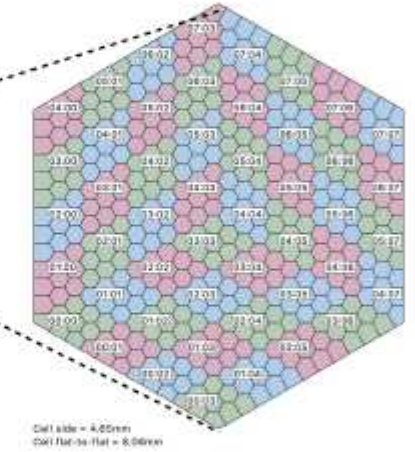
- Inputs are 3D "images"
  - Sensor cells can have various shapes and sizes in the same image
  - e.g., hexagons, trapezoids, diamonds
  - The 3rd dimension is made of consecutive layers separated with non-uniform distances

Pion showers in the HGCAL



Layer with different types of cells





Example of module

15/06/20

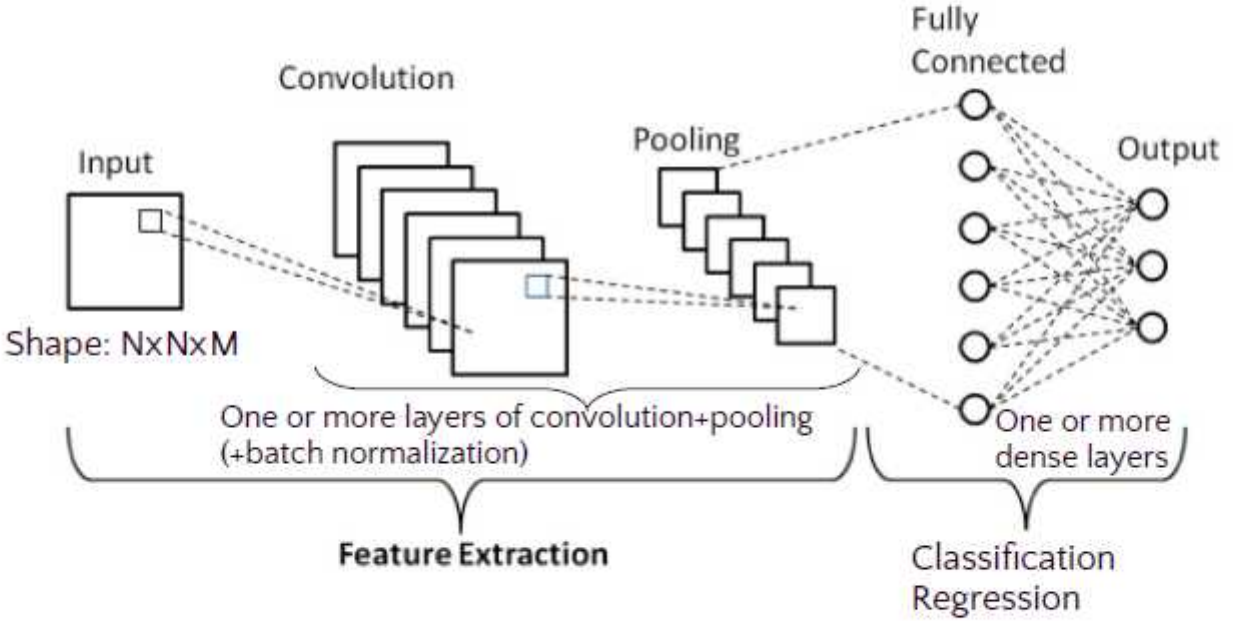
2

Jean-Baptiste Sauvan, Frédéric Magniette, LLR

CMS

Network structure

- If the data is kept with its heterogeneity, graph convolutional networks can be used
- But in a more standard approach, the data can also be mapped to a regular grid with squared pixels
  - Can be treated as 3D images, or 2D with M features corresponding to the 3rd dimension



- The task can be a classification or regression task, or more complex tasks like object detection and segmentation

15/06/20 3

ATLAS

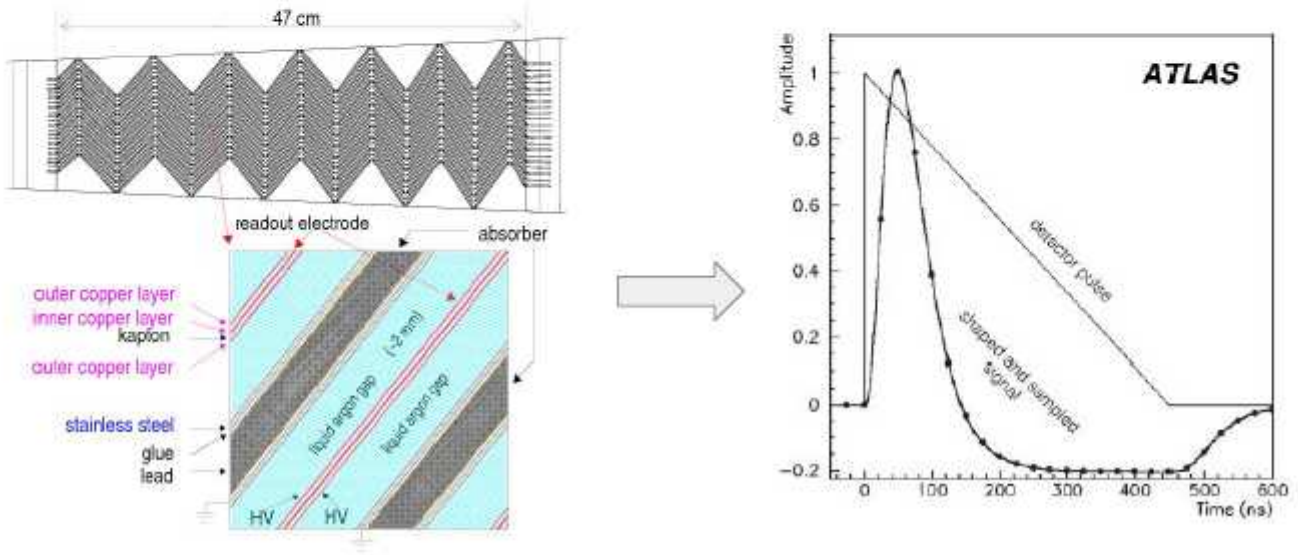
Computing the energy deposited in the LAr Calorimeter using a neural network

Georges Aad,
Thomas Calvet,
Emmanuel Monnier,
Etienne Fortin,
CPPM

ATLAS

LAr Calorimeter Pulse

- Energy deposited in calo cell leads to detector pulse
- Pulse shaped in a bipolar shape and digitized at 40 MHz
  - Pulse shape spans over 32 samples (800 ns)
- Need to compute the energy from the pulse shape
  - Should be done online at 40 MHz

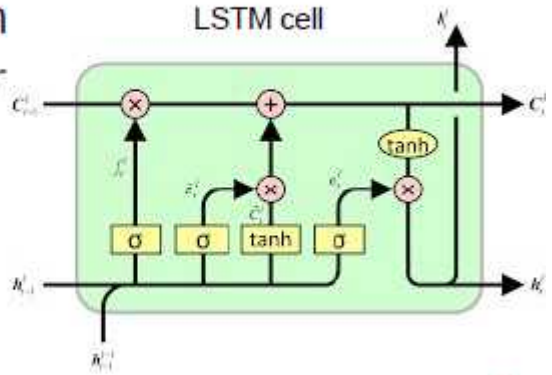


2

ATLAS

Neural Network Architecture

- Need a filter algorithm that produce an energy at each LHC bunch crossing using the information from the past digitized samples
  - Recursive NN (such as LSTM) are the obvious choice
  - Convolutional NNs using several samples are also investigated
- The algorithm should be robust against changing pileup conditions
  - Should learn from past deposits about the structure of pileup
- Typical constraints for this application
  - Large number (~500) of channels per hardware -> need small networks
  - Low latency algorithms (few 100 ns)
- However Recursive networks can be used in various types of application that can have different constraints



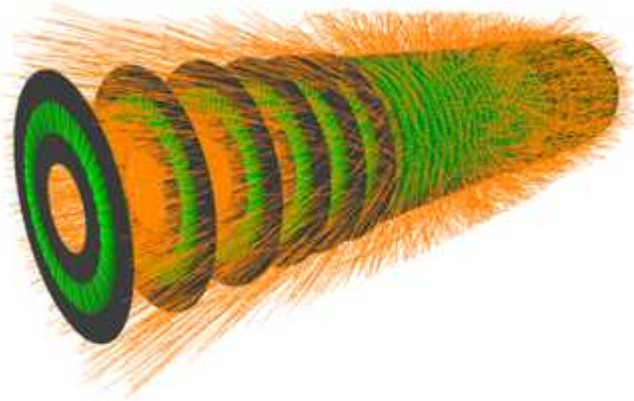
3

LHCb

Do not reinvent wheel

Use CERN TrackML contest

- Was announced in 2018
  - Result ?
- More info here



To explore what our universe is made of, scientists at CERN are colliding protons, essentially recreating mini big bangs, and meticulously observing these collisions with intricate silicon detectors.

While orchestrating the collisions and observations is already a massive scientific accomplishment, analyzing the enormous amounts of data produced from the experiments is becoming an overwhelming challenge.

Event rates have already reached hundreds of millions of collisions per second, meaning physicists must sift through tens of petabytes of data per year. And, as the resolution of detectors improve, ever better software is needed for real-time pre-processing and filtering of the most promising events, producing even more data.

To help address this problem, a team of Machine Learning experts and physics scientists working at CERN (the world largest high energy physics laboratory), has partnered with Kaggle and prestigious sponsors to answer the question: can machine learning assist high energy physics in discovering and characterizing new particles?

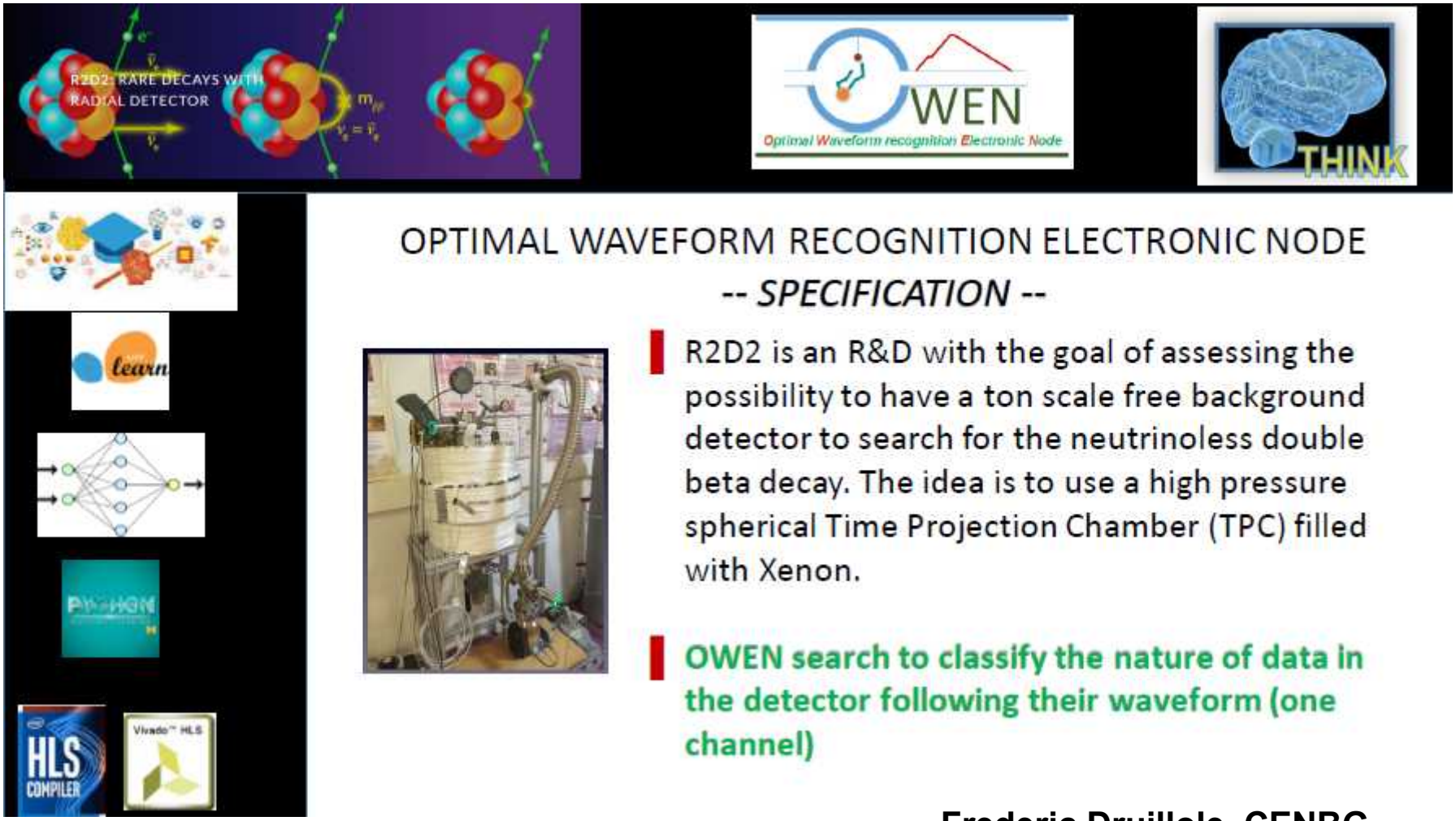
Specifically, in this competition, you're challenged to build an algorithm that quickly reconstructs particle tracks from 3D points left in the silicon detectors. This challenge consists of two phases:

- The Accuracy phase has run on Kaggle from May to 13th August 2018 (Winners to be announced by end September). Here we'll be focusing on the highest score, irrespective of the evaluation time. This phase is an official IEEE WCCI competition (Rio de Janeiro, Jul 2018).
- The Throughput phase will run on Codalab starting in September 2018. Participants will submit their software which is evaluated by the platform. Incentive is on the throughput (or speed) of the evaluation while reaching a good score. This phase is an official NIPS competition (Montreal, Dec 2018).

All the necessary information for the Accuracy phase is available here on Kaggle site. The overall TrackML challenge web site is there.

Vladimir Gligorov,
Olivier Le Dortz,
LPNHE

OWEN



OPTIMAL WAVEFORM RECOGNITION ELECTRONIC NODE

-- SPECIFICATION --

- R2D2 is an R&D with the goal of assessing the possibility to have a ton scale free background detector to search for the neutrinoless double beta decay. The idea is to use a high pressure spherical Time Projection Chamber (TPC) filled with Xenon.

- **OWEN search to classify the nature of data in the detector following their waveform (one channel)**

Frederic Druillole, CENBG
Anselmo Meregaglia

OWEN

OWEN : Apprentissage sur Ordinateur

Prototype R2D2

Simulateur R2D2

Source de données

Données d'entrainements

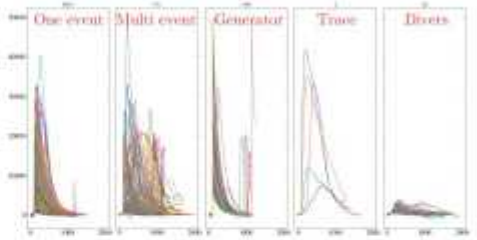
Données classées

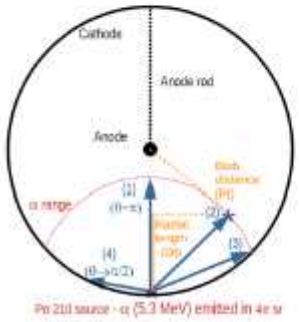
Données de Tests

Données classées de tests

Données réelles

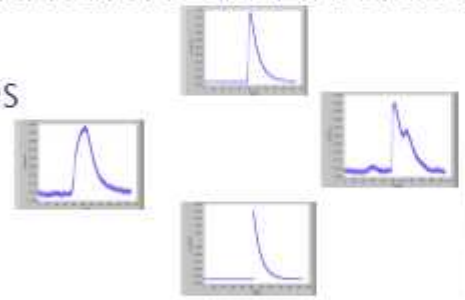
Labellisation automatique

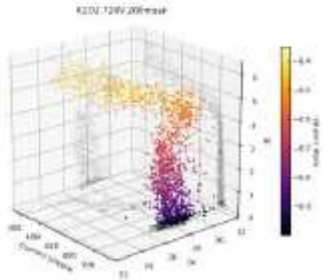




Objectf: determiner la nature de l'interaction:

- 1 particule
- Plusieurs particules
- Trace
- Générateurs
- Autres





Sélection en ligne de la recherche d'évènements rares (double bêta)

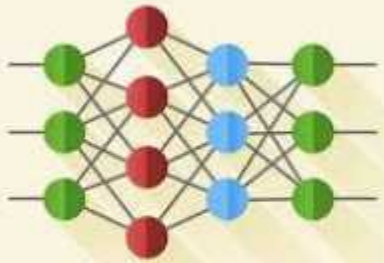
2

OWEN

OWEN : Implementation sur FPGA

1- Régression logistique

2- Réseau de Neurone Profond



- Objectif:
  - Extraire des caractéristiques des signaux
  - Classification des formes d'ondes

3

IMXGAM

Deep Learning challenges for tomographic imaging

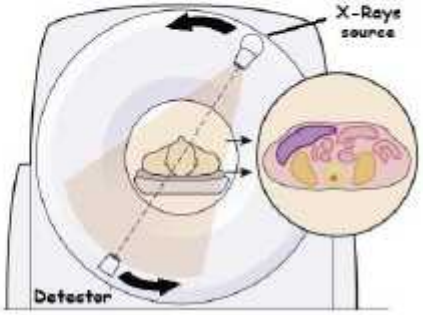
Two targeted challenges :

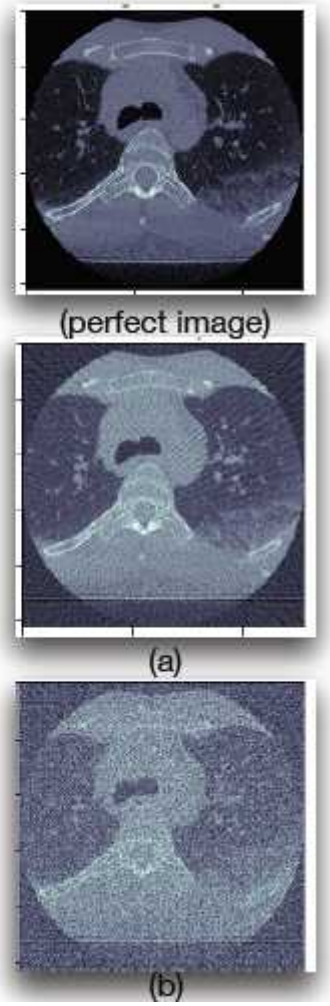
1 - Image denoising
2 - Source separation and parameter estimation

Yannick Boursier, CPPM

imXgam

Context : Tomographic reconstruction





- Reconstructed volumes encompasses two kinds of noise:

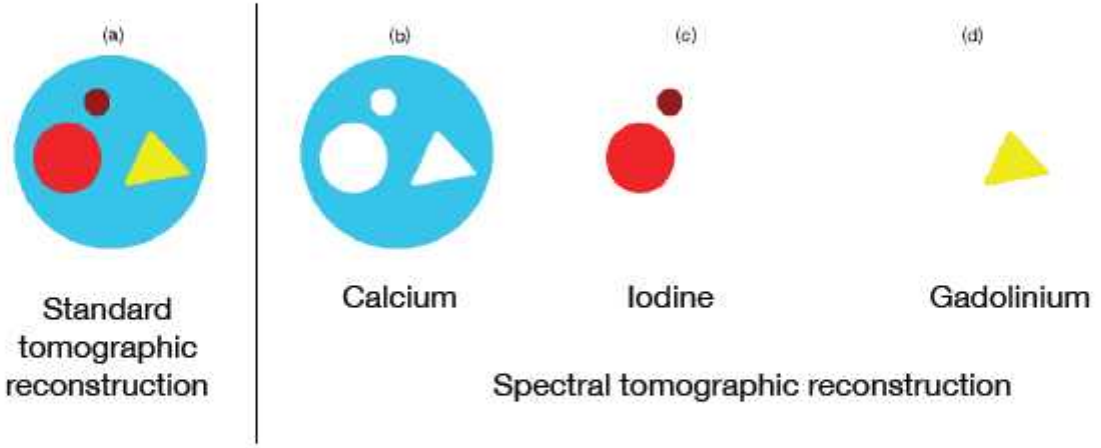
  - Structured Radon noise (streaks) (a)
  - Poisson noise (b)

- For the moment, treat volume slice by slice (thus images).
  If directly dealing with 3D volume is tractable, let's go !

imXgam

Context : Spectral tomography

- Multi-channels acquisition and multi-channel 3D reconstruction

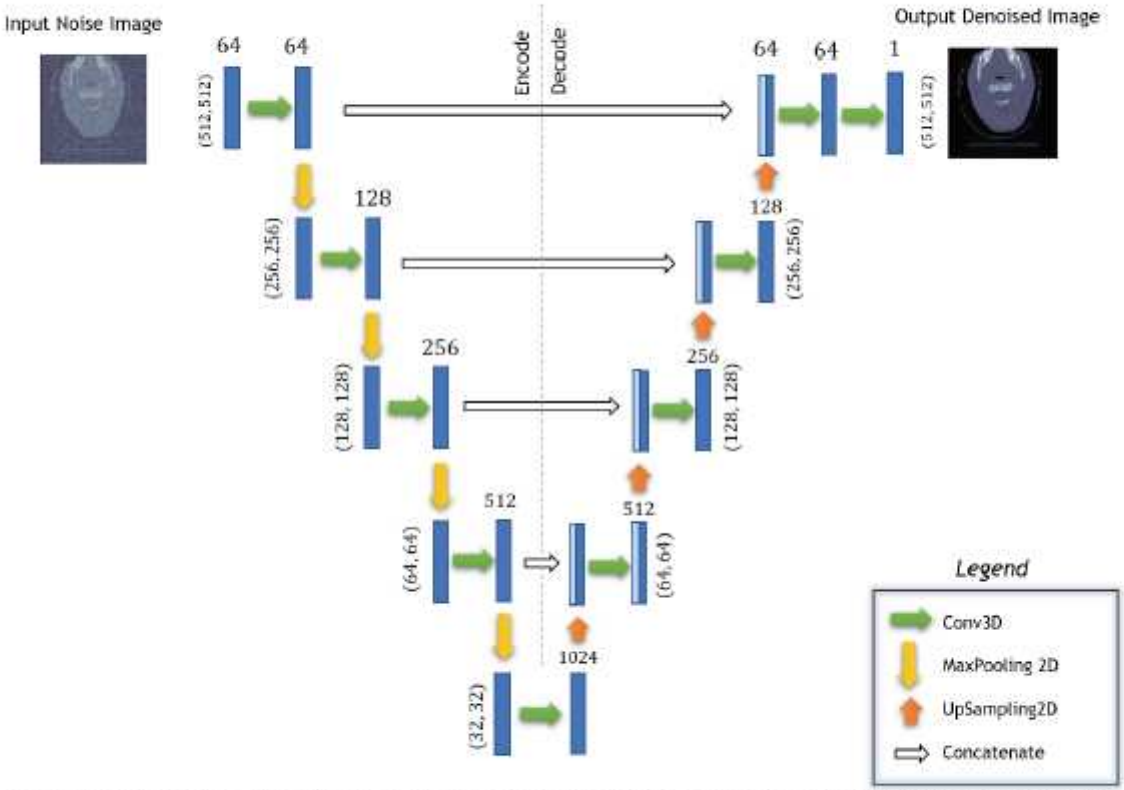


- Issue : reconstruct with high quality each component map.

  Hidden challenges : accurate and precise (denoising) identification (separation), quantification (regression) of each map.

imXgam

Architecture : the U-net CNN



By redefining the last layers (e.g. dense layers), we can achieve denoising, segmentation or classification.
Our first results on real data are in favor of a U-net-like architecture.

Synthèse

Applications du projet

- CMS : images 3D multi-tracks multi-sensor
  convolutive NN
  identification de particules
- ATLAS : images 1D single sensor évoluant dans le temps (pile-up)
  recursive NN, peut être ramené à réseau convolutif avec n dimensions
  calcul d'énergie
- TrackML: images 3D multi-track multi-sensors
  trajectographie
- OWEN : images 3D multi-track single sensor
  convolutive NN
  identification de particules
- IMXGAM : Images 2D bruitées
  convolutive NN, maxpooling
  débruitage

Challenge 4

Nombre de couches

2 entrées : les coordonnées X,Y.
Une couche Fully-Connected avec 1000 neurones + couche d'activation ReLU.
Une couche Fully-Connected avec 500 neurones + couche d'activation ReLU.
Une couche Fully-Connected avec 250 neurones + couche d'activation ReLU.
Une couche Fully-Connected avec 100 neurones + couche d'activation ReLU.
Une couche Fully-Connected avec 50 neurones + couche d'activation ReLU.
Une couche Fully-Connected avec 1 neurone + couche d'activation Sigmoid.

Au total, le réseau comporte 659 002 paramètres.