



Collaborations

L'AI appliquée aux détecteurs semi-conducteurs, LP2i Bordeaux, 1-2 juillet 2025

Discrimination gamma-neutron dans des scintillateurs avec des réseaux de neurones



O. Stézowski

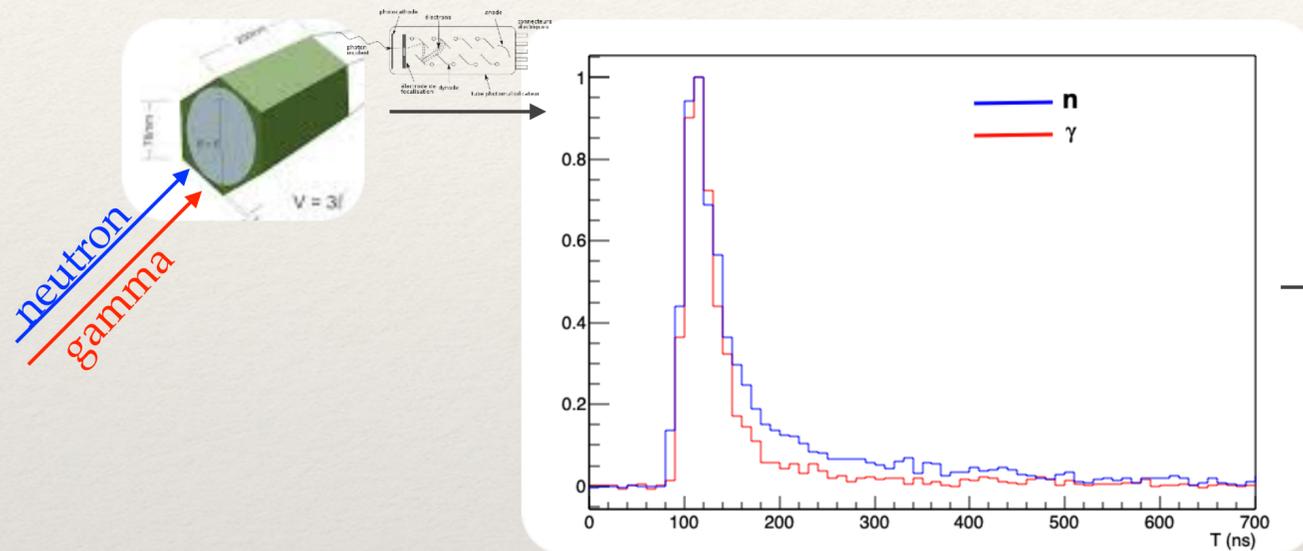
Application de réseaux de neurone à l'analyse de formes de signaux

- ➔ Études qui ont lancées l'application aux Germanium segmentés AGATA
- ➔ Voir présentation J.Dudouet

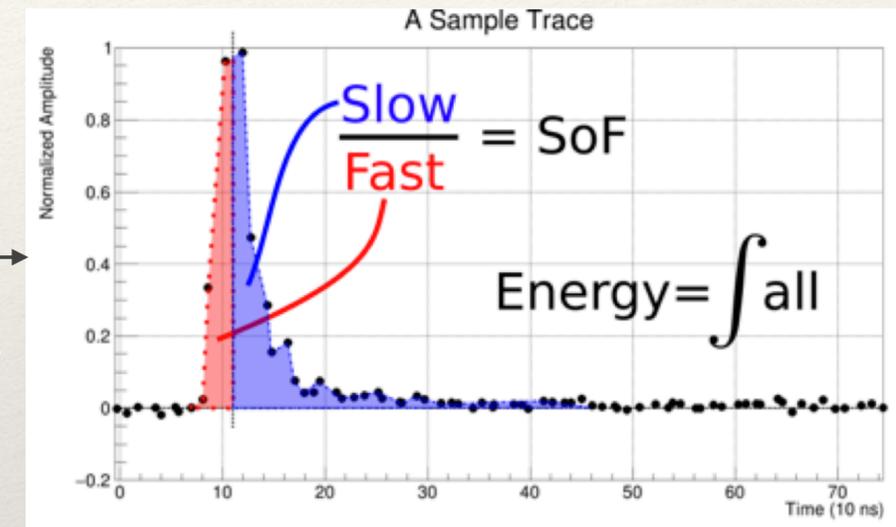
Les techniques utilisées sont exportables aux semi-conducteurs

Contexte : forme des signaux

Les signaux issus des scintillateurs ont des formes différentes en fonction de la nature de la particule

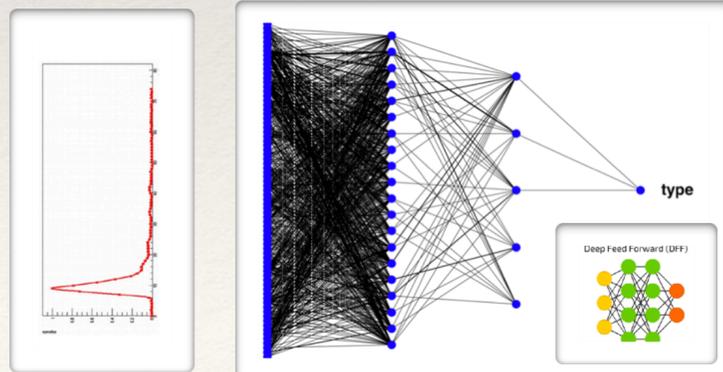


Traitement 'classique'
Traitement simple
Électroniques analogiques



SoF < seuil → γ
SoF > seuil → n

Avec l'électronique numérique, les signaux numérisés peuvent être traités sur ordinateur
→ discrimination à l'aide de réseaux de neurones, input = signal, output = n/γ

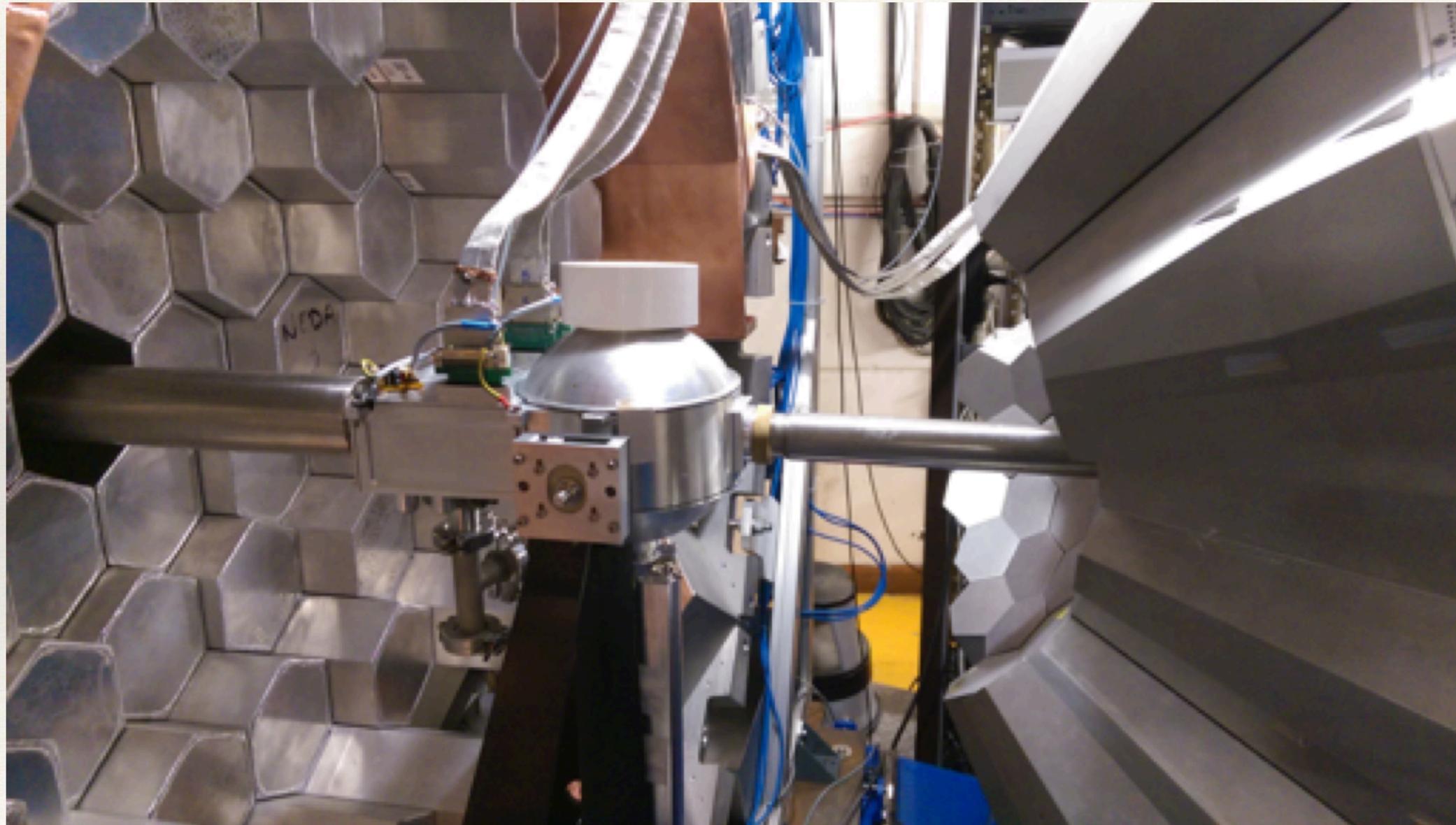


L'étude a été réalisée par la collaboration NEDA
→ setup dédié pour 'tagué' neutron-gamma !

Implementation avec ROOT (monthread / CPU)
Ils ont montré une meilleur discrimination basse énergie
Ronchi et al., A 610 (2009) 534-539

IP2I, pouvez-vous nous aider à mettre ceci dans notre acquisition ?

Contexte : expérience NEDA + AGATA



Contexte : expérience NEDA + AGATA

Modules NEDA



PSD

Artificial Neural Networks

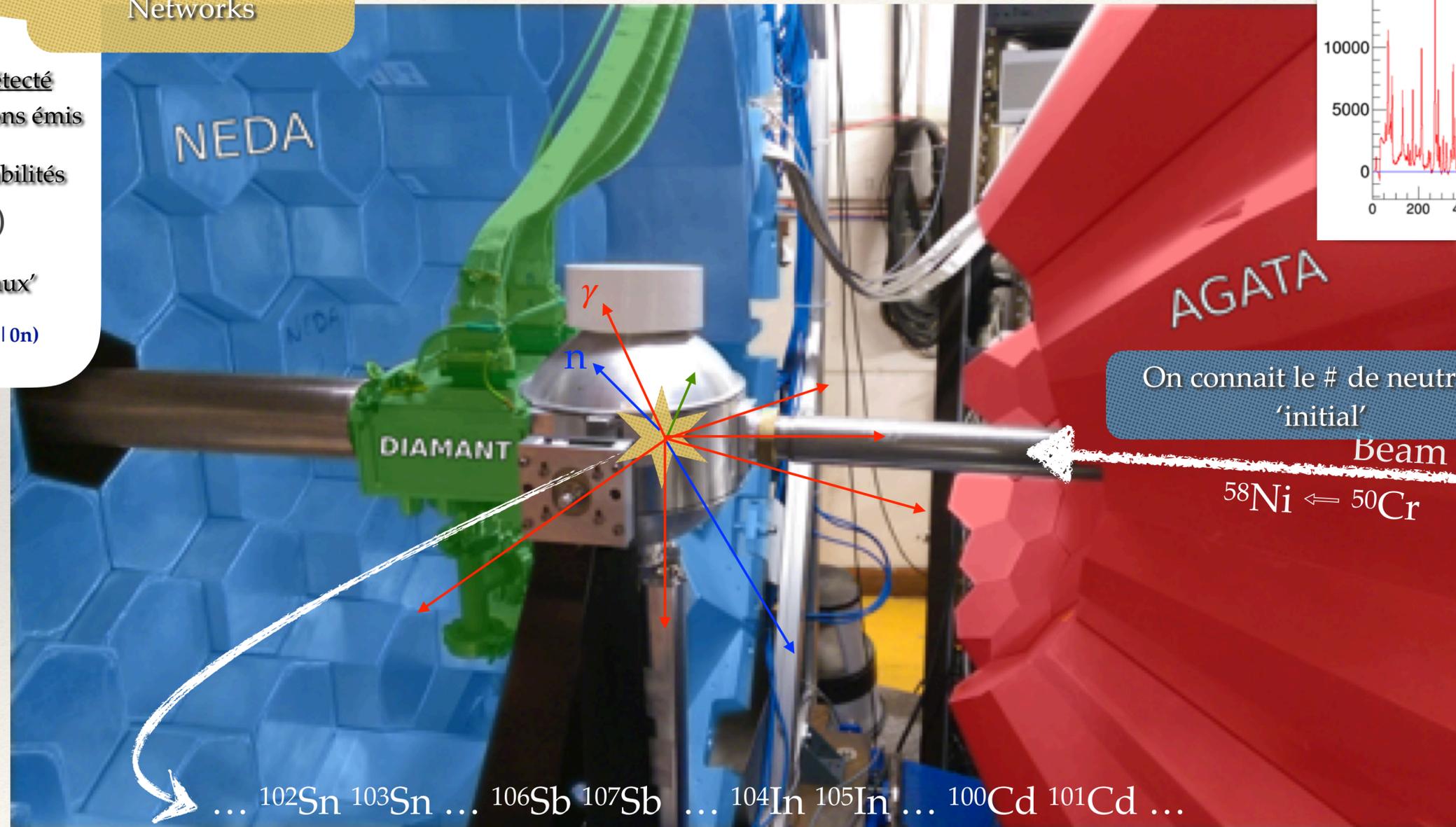
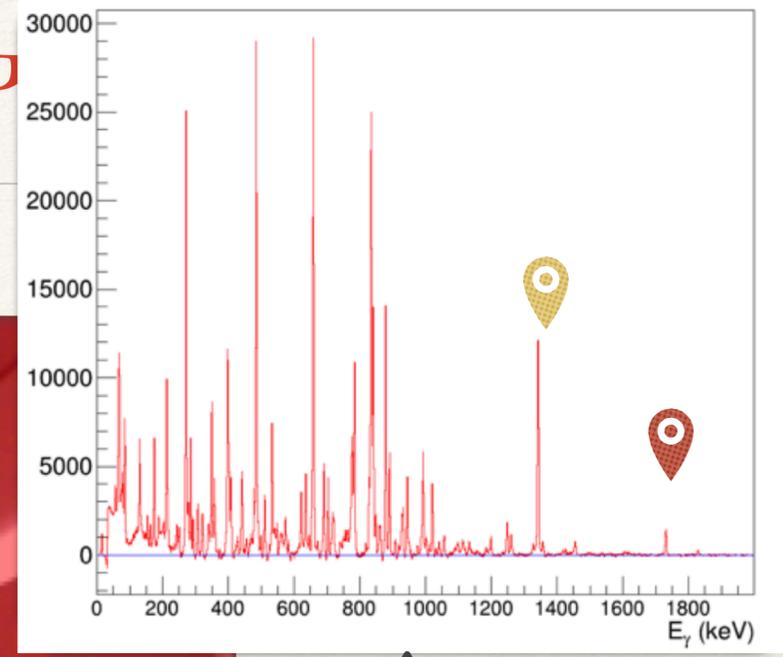
On a le # x de neutrons déTECTÉ
AGATA donne le 3 de neutrons émis

On peut calculer des probabilités

$$P(x_{\text{détecté}} | y_{\text{émis}})$$

On peut connaître les 'faux'

$$\text{Mislabel Probability} \equiv P(1n | 0n)$$



On connaît le # de neutrons 'initial'

... ^{102}Sn ^{103}Sn ... ^{106}Sb ^{107}Sb ... ^{104}In ^{105}In ... ^{100}Cd ^{101}Cd ...

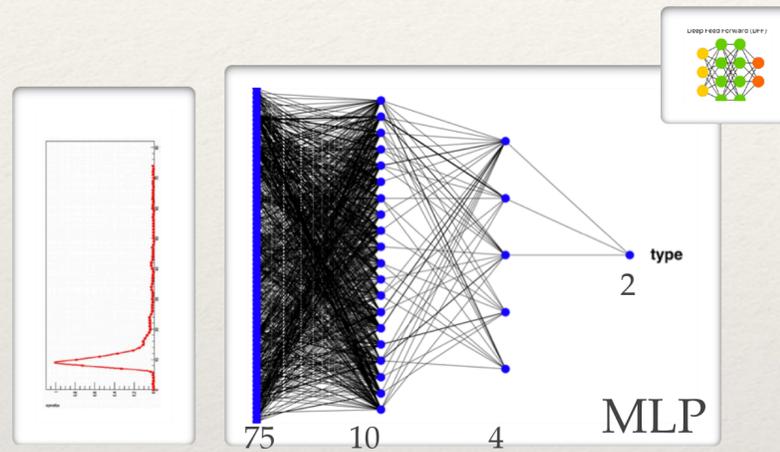
plusieurs résidus de réaction sont produits ... mais AGATA les identifie, on connaît le nombre 'final'

$$\text{Nombre émis} = \text{nombre initial} - \text{final}$$

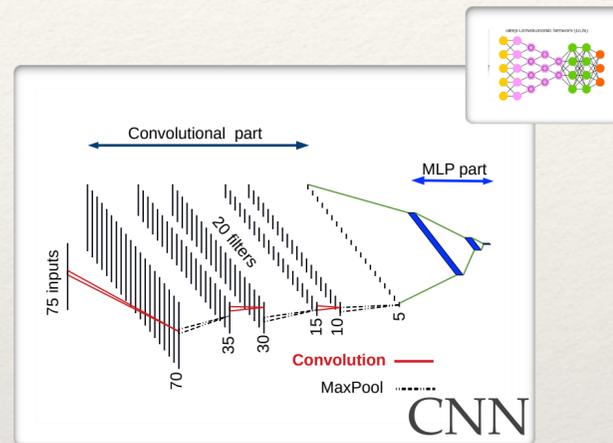
Les réseaux mise en oeuvre

NIMA Volume 986, 11 January 2021, 164750 - <https://doi.org/10.1016/j.nima.2020.164750>

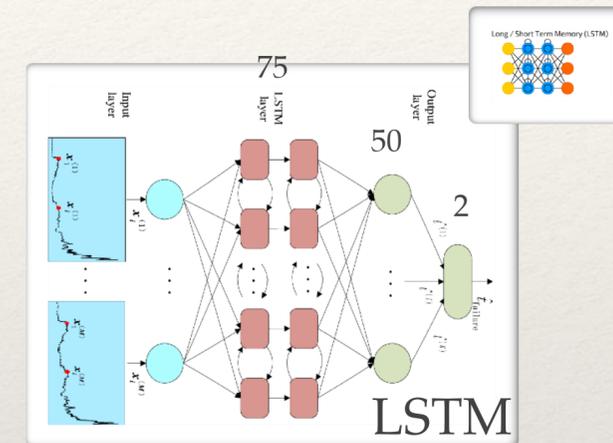
Nous avons utilisé Tensorflow / Keras - multi CPU / GPU  



Développé par la collaboration NEDA

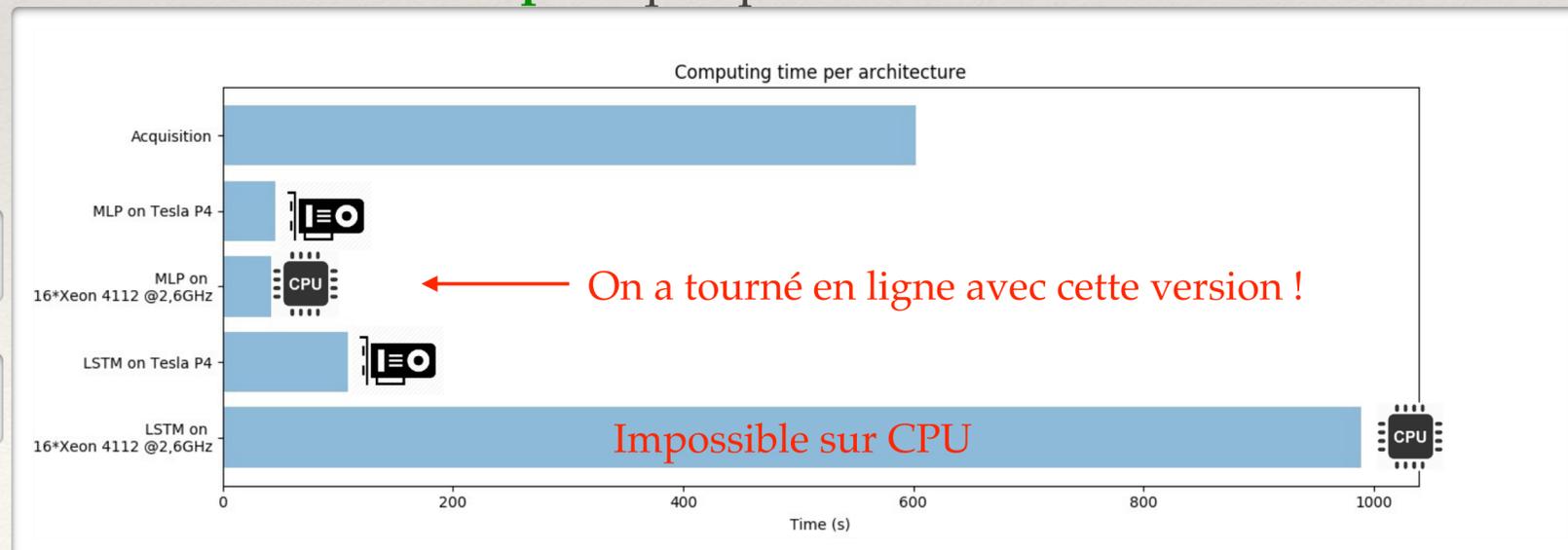


👍 bon pour la reconnaissance de pattern 👍 adapté pour les séries temporelles



Nombre de parametres
MLP: 814,
LSTM: 10502,
CNN: 7042

Temps requis pour l'inference



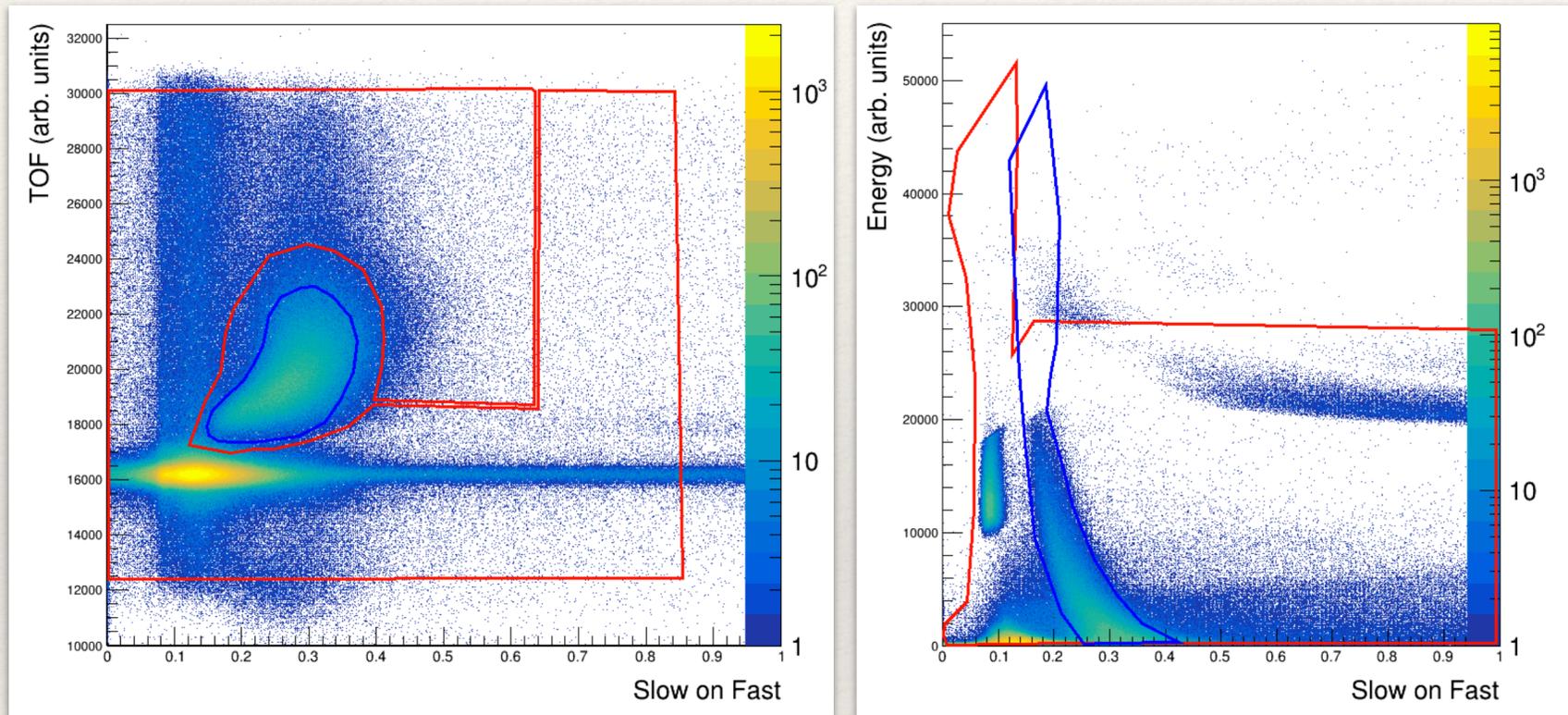
Network type	Structure	Activation functions
MLP	3 Dense layers (75 x 10 x 4 x 2)	Relu x ReLu x SoftMax
LSTM	75 x 1 LSTM layer (50 hidden units) x 1 Dense layer (50 x 2)	SoftMax
Convolution	75 x 3 (Conv1D+Max_Pooling) layers x 2 Dense layers (100 x 20 x 2)	ReLU x ReLu x ReLu x ReLu x SoftMax

Pour le jeu d'entraînement,
il faut une étiquette γ/n !

L'entraînement des réseaux

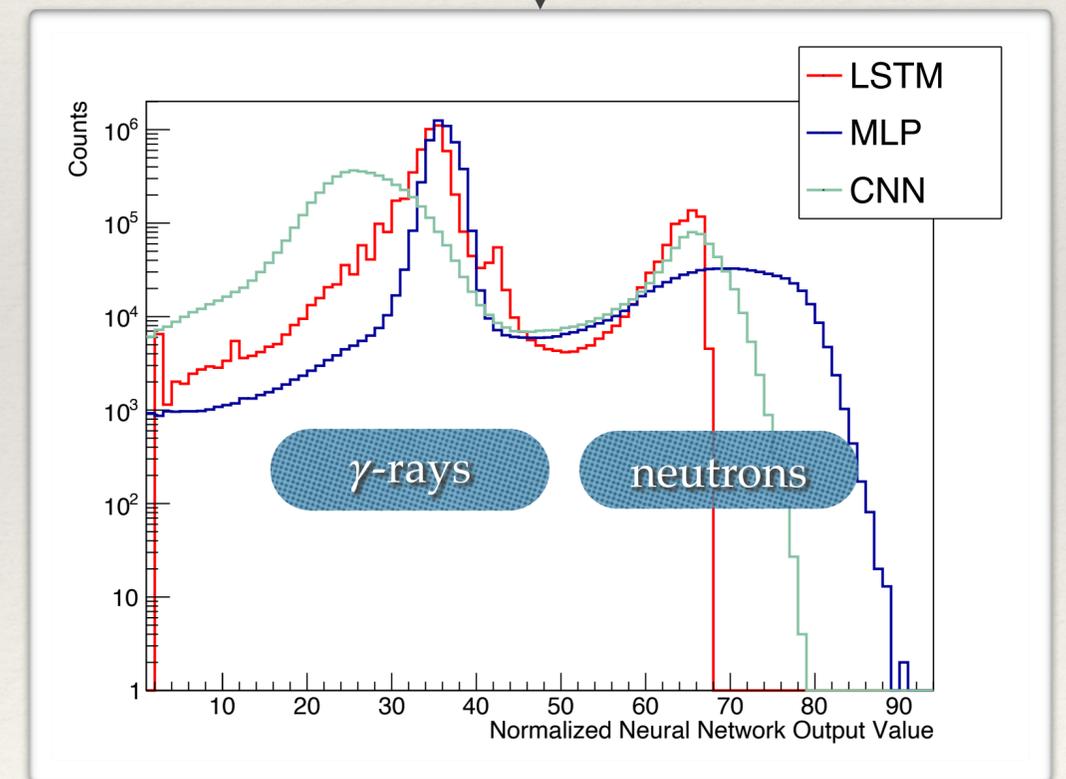
NIMA Volume 986, 11 January 2021, 164750 - <https://doi.org/10.1016/j.nima.2020.164750>

Nous avons utilisé 2 coupures dans deux bidims



Si dans les 2 contours bleus  neutron
Si dans les 2 contours rouges  gamma

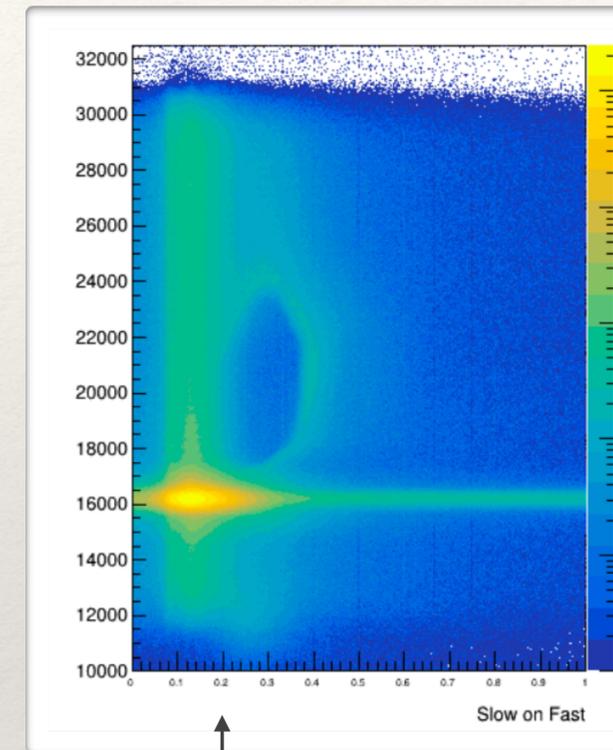
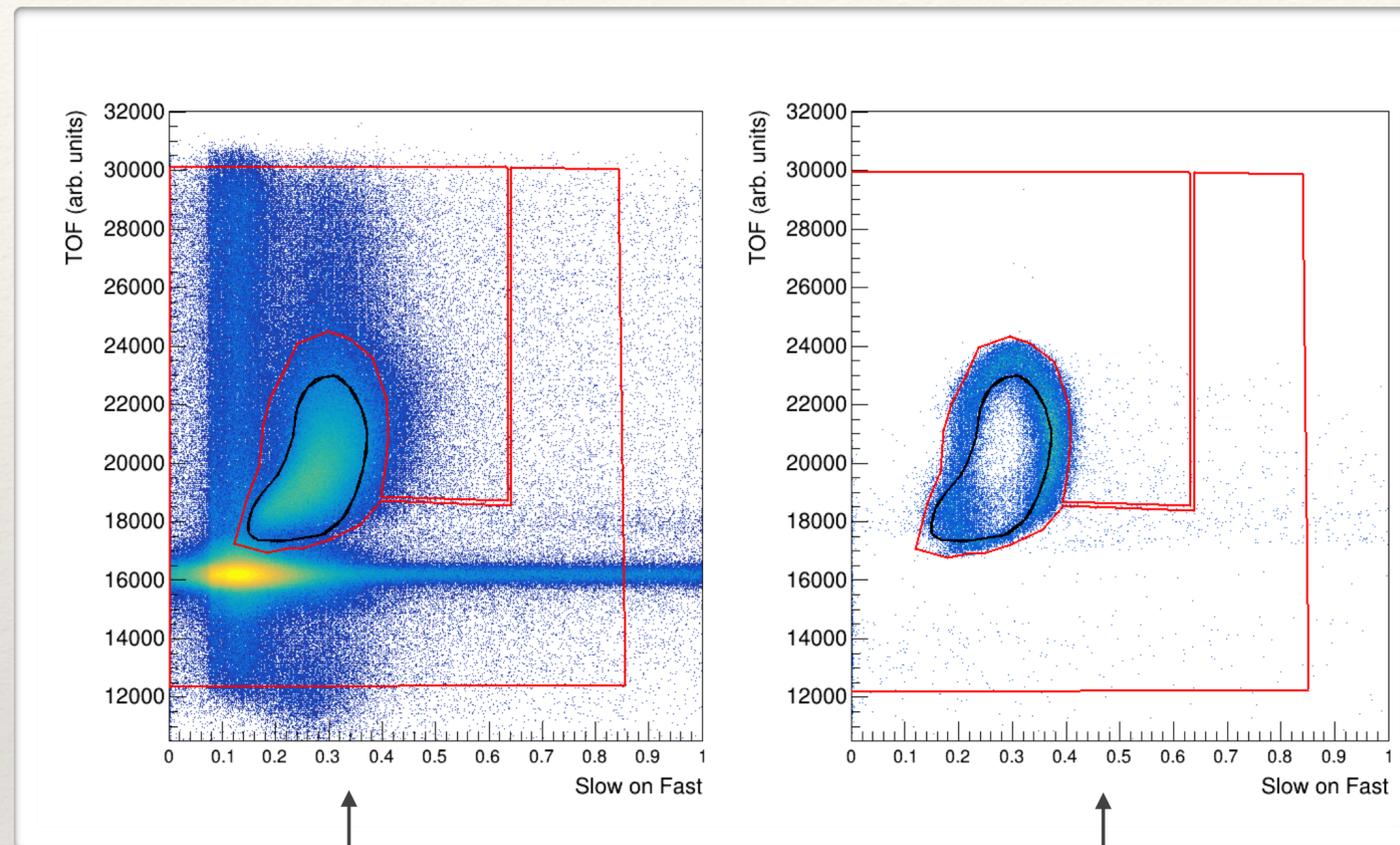
Après entraînement
valeur du neurone de sortie sur les données de test



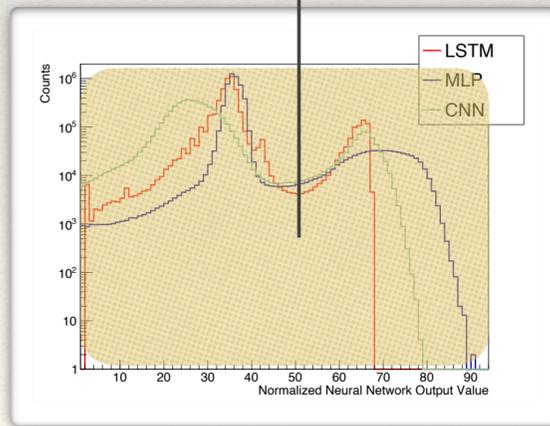
LSTM a des choix plus tranchés ?

Capacité d'interpolation sur les données non apprises ?

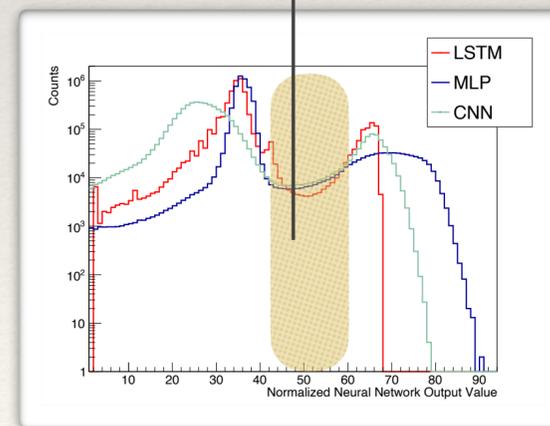
NIMA Volume 986, 11 January 2021, 164750 - <https://doi.org/10.1016/j.nima.2020.164750>



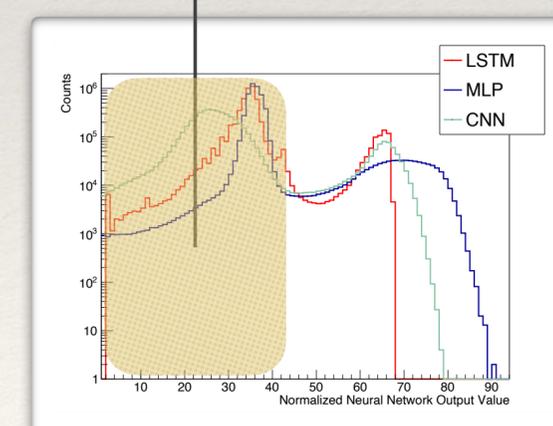
Fenêtre de sélection sur valeur output



On retrouve la forme de la distribution d'entraînement



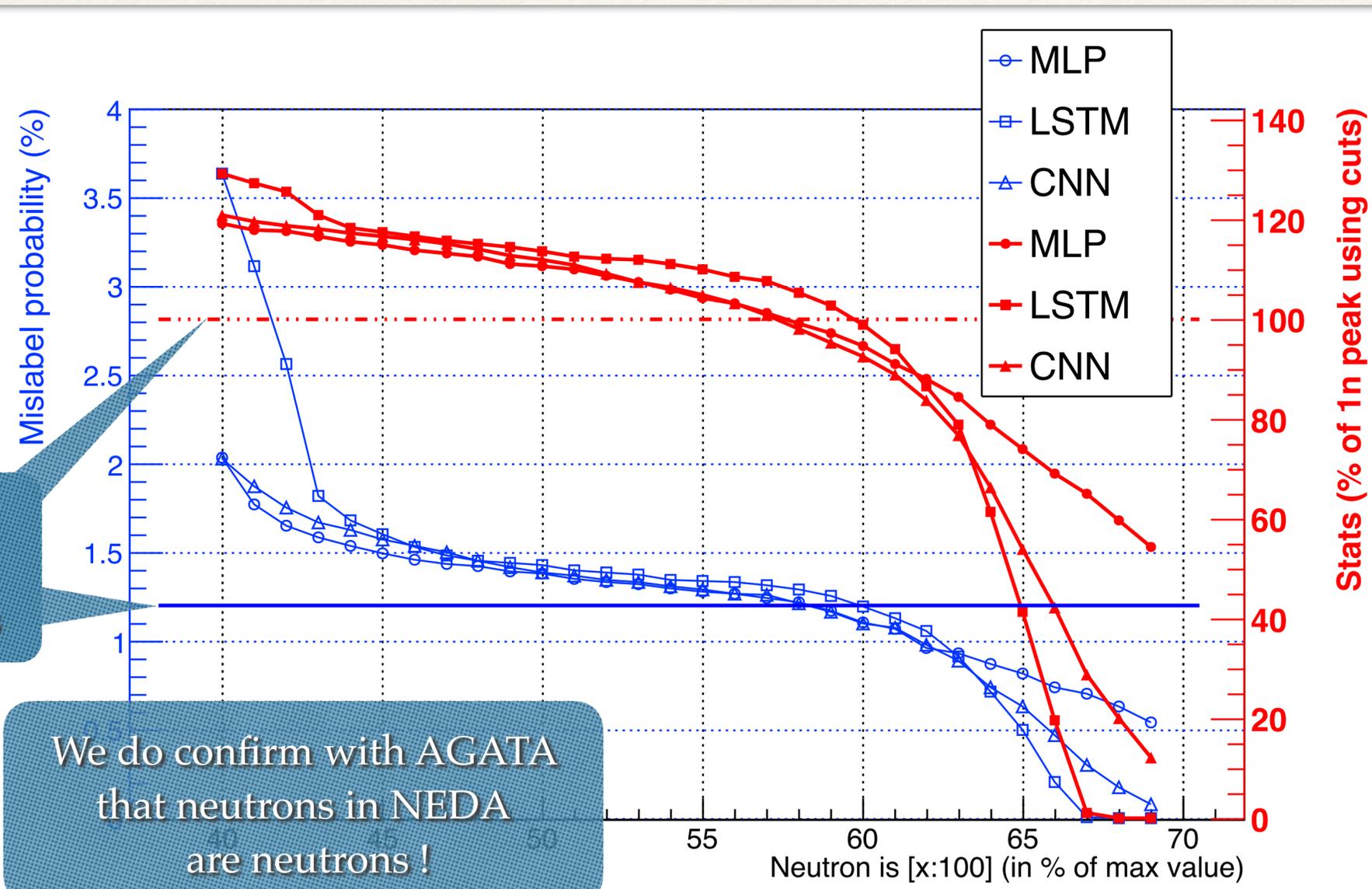
Entre gamma-neutron (non inclus dans l'entraînement)



Fond gamma

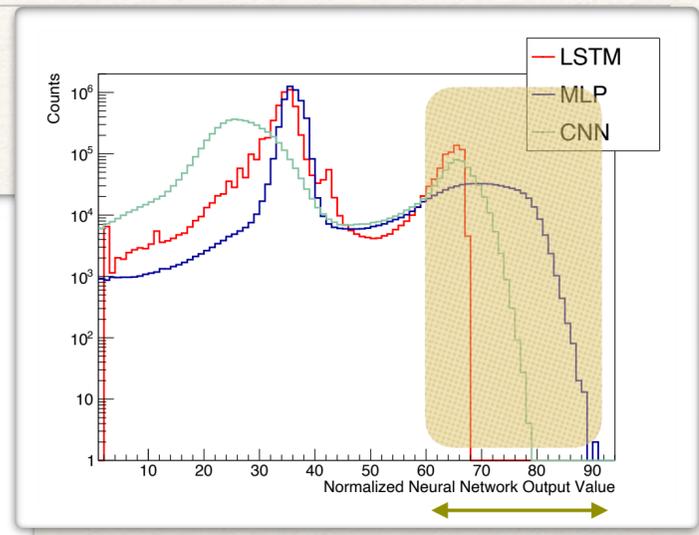
Quantification des erreurs de discrimination

Mislabel probability and impact on statistics in good events



obtained with training cuts

We do confirm with AGATA that neutrons in NEDA are neutrons !



On peut jouer entre qualité et statistique

Autres études, avec des données simulées

Nos données sont tagguées avec des coupures classiques qui ont leurs limites !

↳ les labels sur γ ou n ne sont pas 100 % sûrs !

Mais on sait que la forme des signaux peut être paramétrée de façon assez réaliste par

$$s(t) = \mathbf{A} [\exp(-t/\mathbf{td1}) - \exp(-t/\mathbf{tr}) + \mathbf{R}^*(\exp(-t/\mathbf{td2}) - \exp(-t/\mathbf{tr}))] \text{ si } t > \mathbf{T0}$$

↳ paramètres différents pour γ et n et on génère des événements à partir des 2 fonctions

Etude 1 : sensibilité au temps $\mathbf{T0}$

On sait que aligner les signaux expérimentaux est une procédure délicate

On a entraîné les réseaux avec des signaux dont $\mathbf{T0}$ suivait une distribution gaussienne avec $\sigma = 2$

On a ensuite testé avec des signaux ayant une distribution $\mathbf{T0}$ bien plus large σ de 2 à 40

Etude 2 : pouvoir des réseaux de neurones pour séparer deux signaux de pileup

On a travaillé avec deux signaux de pileup séparés par ΔT , ΔT ayant distribution random uniforme

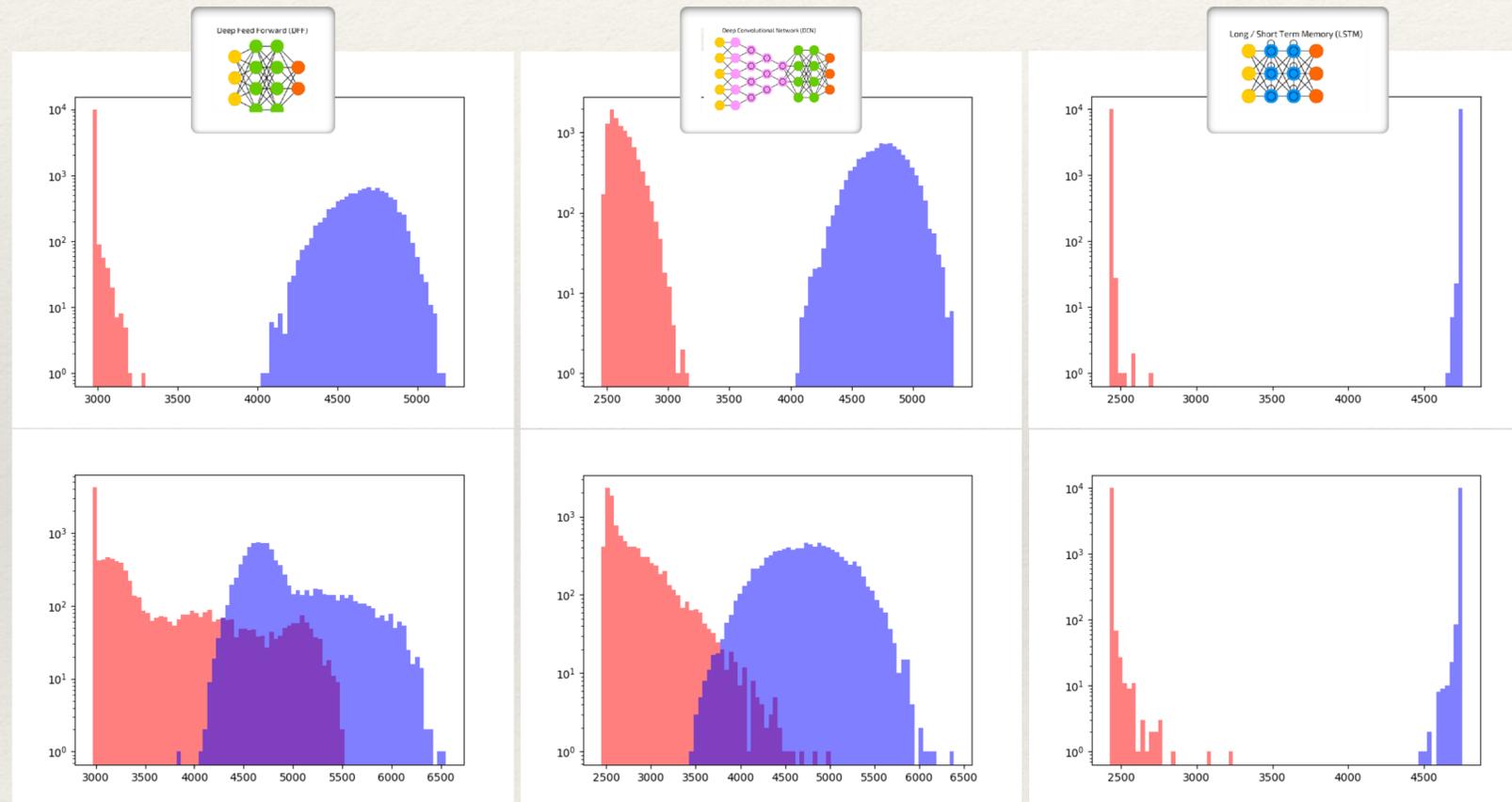
Etude 1: sensibilité au désalignement T0

MLP

CNN

LSTM

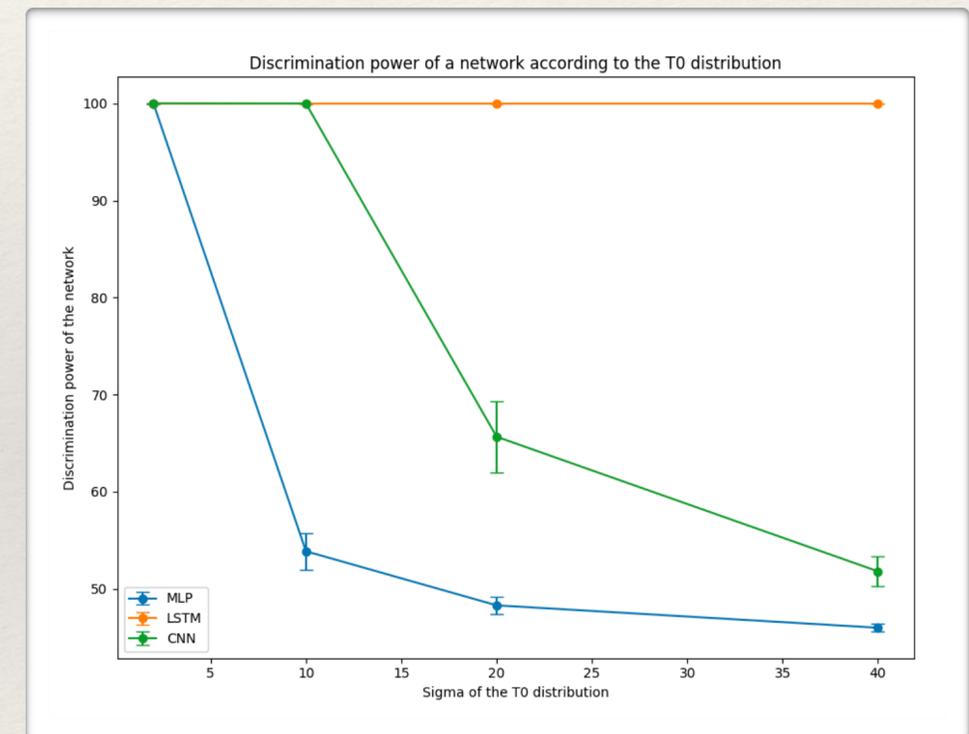
$\sigma = 2$



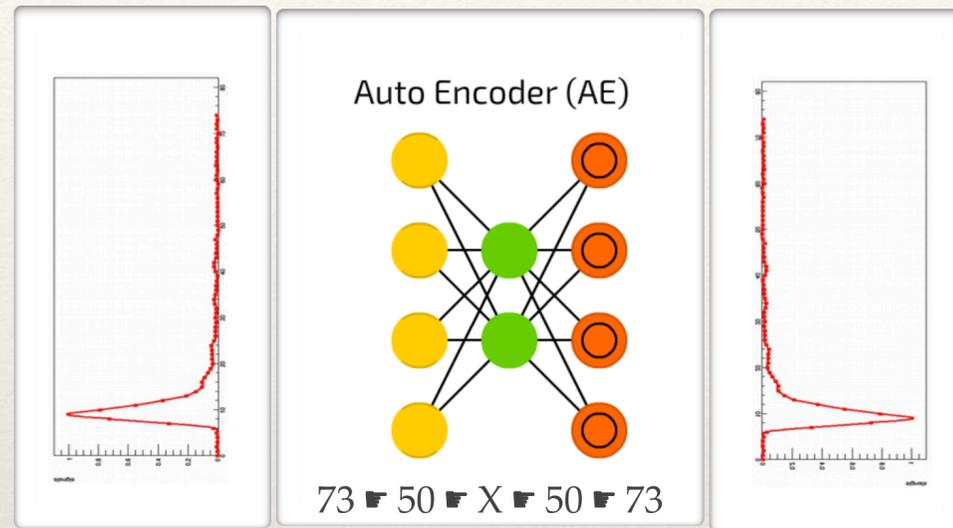
$\sigma = 20$

Valeur du neurone de sortie

LSTM le + robuste !



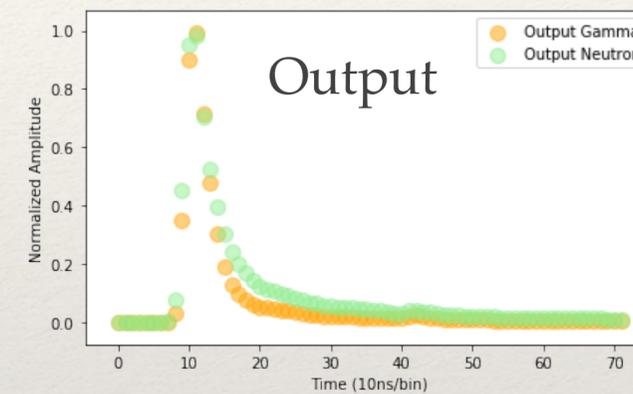
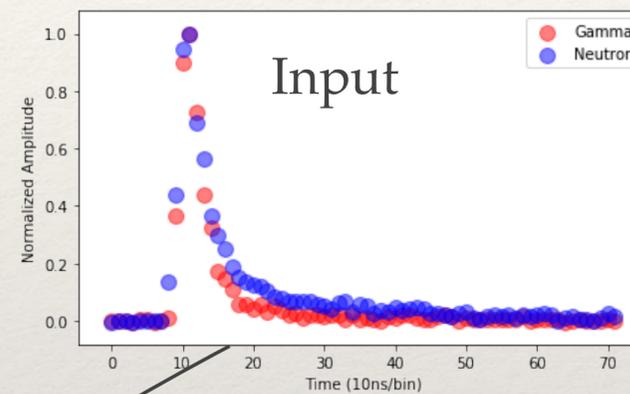
Etude 2: architecture auto-encodeur



Pour l'entraînement, sortie = entrée

Pas de supervision \Rightarrow auto apprentissage !

\Rightarrow Pas de pb de classification pour le jeux d'entraînement ...



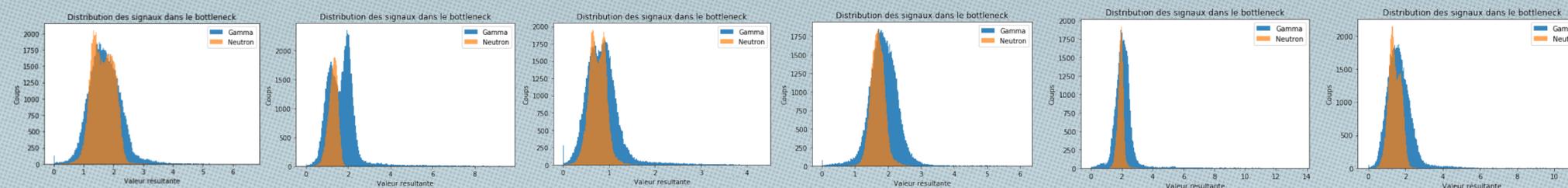
Interets :

Dé-bruiter les signaux
Réduction des données

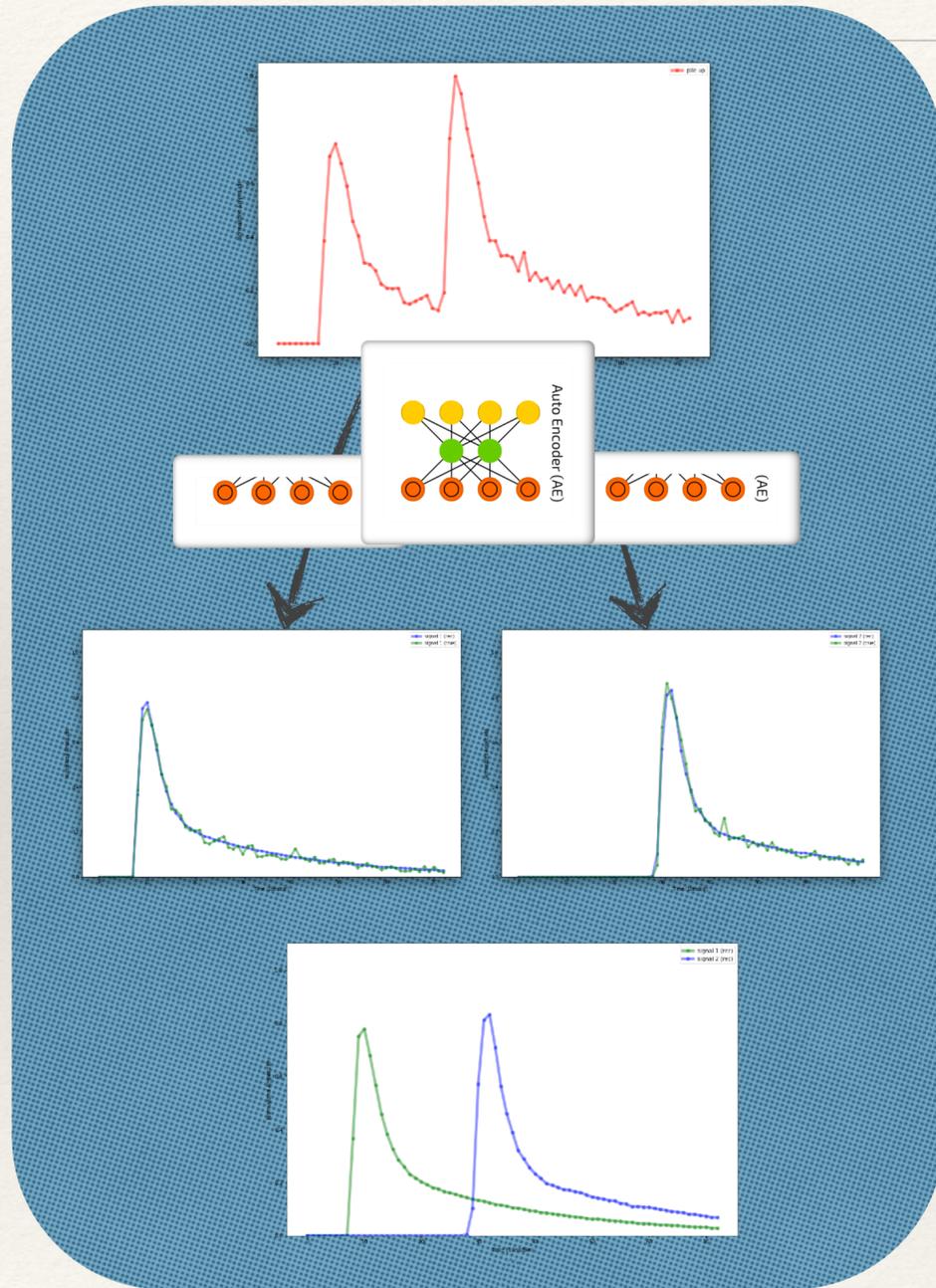
Identification des anomalies ?
Génération de signaux ?

Minimum 4 neurones centraux, lien avec
 $s(t) = A [\exp(-t/\tau d1) - \exp(-t/\tau r)] + R^*(\exp(-t/\tau d2) - \exp(-t/\tau r))$?

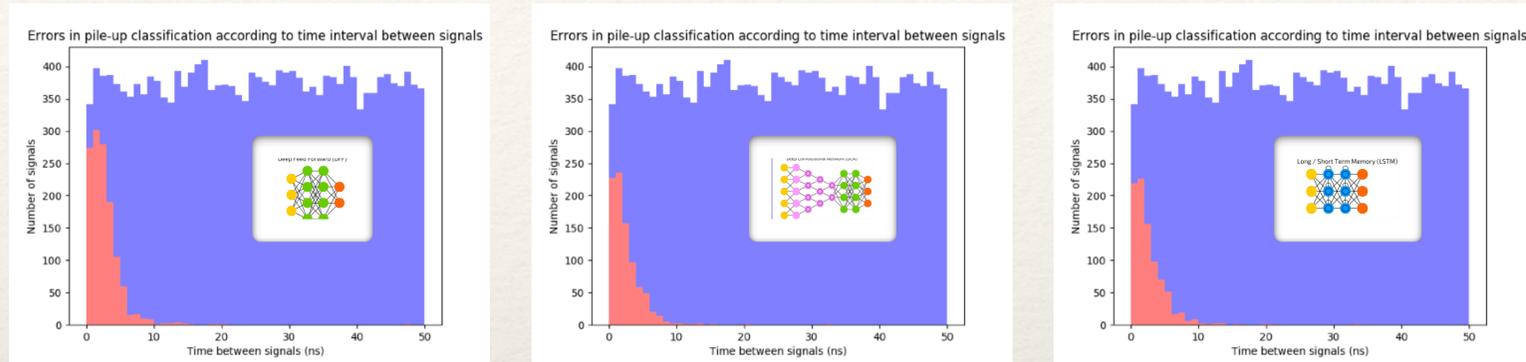
Distribution 6 neurones centraux



Etude 2: pileup déconvolution & identification par auto-encodeur



Erreur en fonction du ΔT entre les deux signaux ... seulement quand $\Delta T < 10$ ns, sinon presque 100 % efficace



Matrice de confusion globale

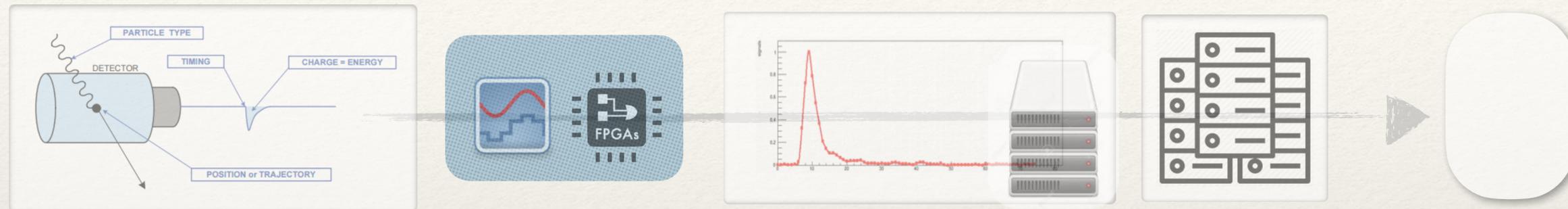
True label \ Predicted label	G	N	G/G	G/N	N/G	N/N
G	99.98	0.00	0.02	0.00	0.00	0.00
N	0.00	99.99	0.00	0.00	0.00	0.01
G/G	0.78	0.00	99.22	0.01	0.00	0.00
G/N	0.01	0.01	0.01	99.44	0.53	0.00
N/G	0.01	0.01	0.01	0.84	99.14	0.00
N/N	0.00	0.88	0.00	0.00	0.00	99.12

True label \ Predicted label	G	N	G/G	G/N	N/G	N/N
G	100.00	0.00	0.00	0.00	0.00	0.00
N	0.00	99.91	0.00	0.00	0.00	0.09
G/G	0.56	0.00	99.44	0.00	0.00	0.00
G/N	0.00	0.00	0.00	99.86	0.14	0.00
N/G	0.00	0.00	0.01	1.03	98.96	0.00
N/N	0.00	0.37	0.00	0.00	0.00	99.63

True label \ Predicted label	G	N	G/G	G/N	N/G	N/N
G	99.98	0.00	0.02	0.00	0.00	0.00
N	0.00	99.87	0.00	0.00	0.00	0.13
G/G	0.54	0.00	99.46	0.00	0.00	0.00
G/N	0.00	0.00	0.00	99.85	0.14	0.00
N/G	0.00	0.00	0.00	1.07	98.93	0.00
N/N	0.00	0.36	0.00	0.00	0.00	99.64

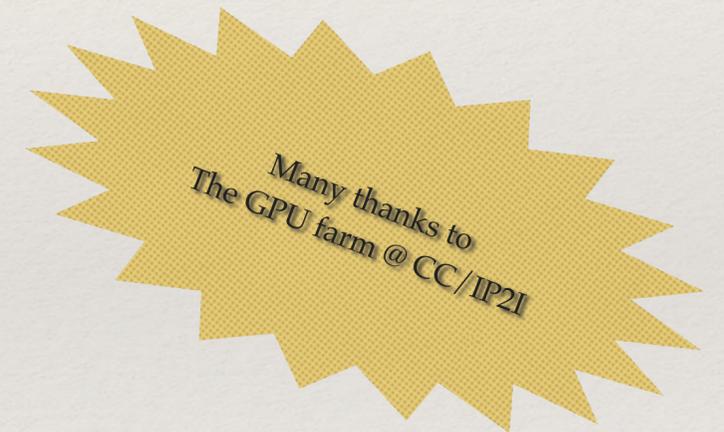
On a des erreurs mais rarement on confond gamma - neutron !

Conclusions / Perspectives



Données NEDA idéales pour nos premiers pas en ML, 3 architectures étudiées

- MLP a tourné en ligne sur CPU.
- Les données AGATA ont permis de qualifier la discrimination γ/n dans NEDA
- LSTM a quelques avantages, moins sensible à un désalignement temporel des signaux
- Auto encoders into the game for compression / de-noising



Rampe de lancement pour le traitement des signaux AGATA

- NEDA: Pulse Shape Discrimination (oui/non) sur un signal par module
- AGATA, beaucoup plus complexe : Pulse Shape Analysis \equiv regression 37 signaux par crystal
- Pas de modèle simple plus les pulses \rightarrow Simulations plus complexes, scanning 3D à mettre en oeuvre ...

Pistes non explorées avec les données NEDA

- compression / de-noising / detection d'anomalies