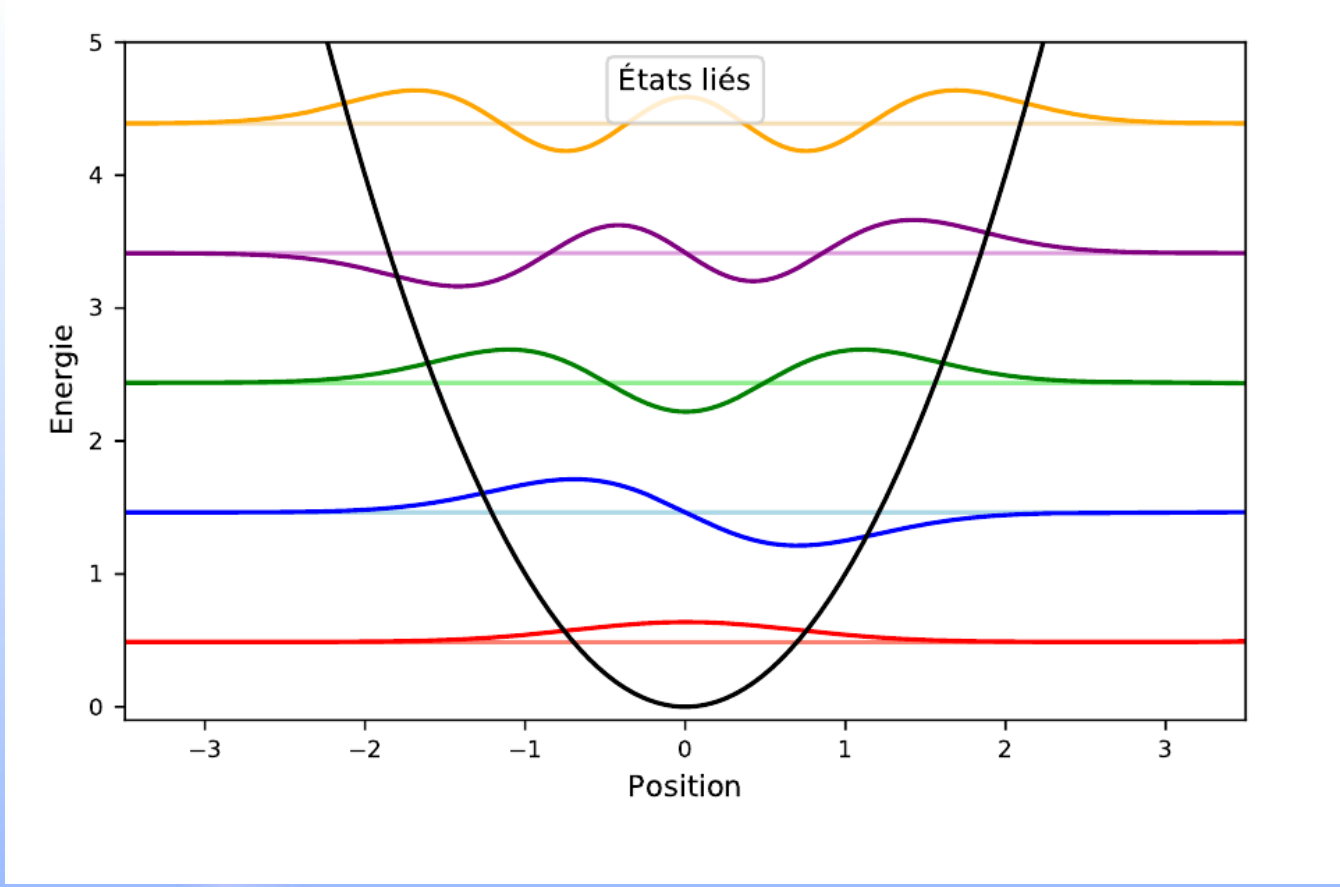
Machine learning pour la théorie

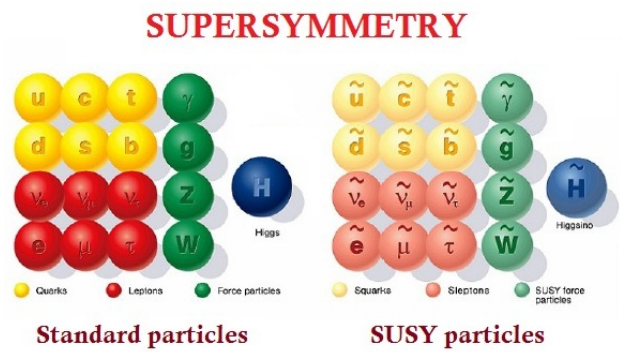
Supersymétrie, physique nucléaire, trouver les niveaux d'énergies d'un système quantique



Clément Lotteau, H. Hansen*, J. Margueron

Dans le groupe théorie

Deep Learning for SuperSymmetry



- **Supersymmetric model**
- **19 parameters** more than in the SM
- **~ 1s** to compare one scenario to observables

- => Parameter space **exploration challenging**

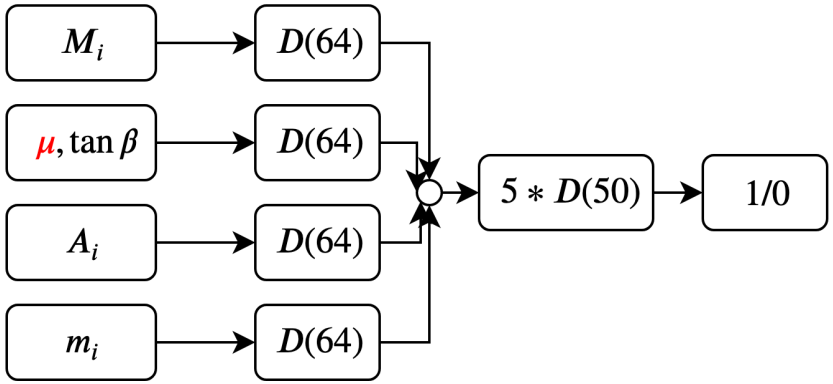
- Exploration guided by a **neural network**
- Learns to **recognize physically interesting scenarios**

- **Prefer false positive** to false positive: **overestimation** from the network:
  - **1%** false negative
  - **5%** false positive
- Exploration **50 times faster** with the MLP.

Groupe théorie,
A. Boussejra, N. Mahmoudhi, G. Ulrich

- Multi-Layer Perceptron (**MLP**)
- Fully connected layers (**Dense**)
- **Inputs**: parameters dispatched in 4 groups
- **Output**:
  - **1:** physically interesting scenario
  - **0:** not interesting, **we do not go further**

Fig. Network structure (D = Dense layer)



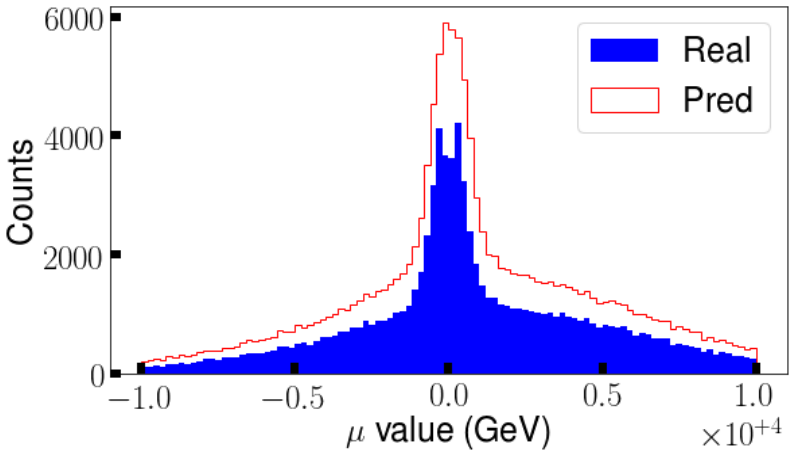


Fig. Predicted distribution of one parameter for physically interesting scenarios

Profiting from applied mathematics for machine learning

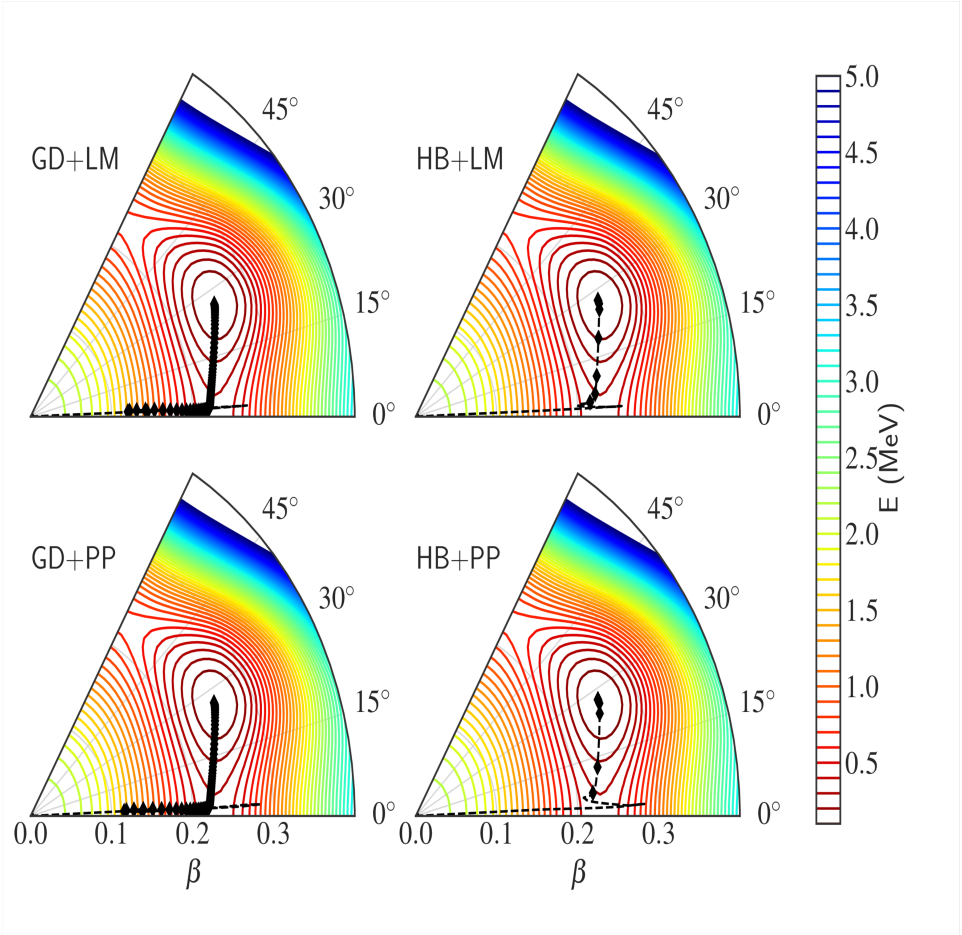
Problem: iterative construction of the lowest eigenstates of a large matrix in nuclear structure problems.

Similar to the the iterative solution of large systems of coupled equations as encountered for certain types of neural network.

Using the popular « heavy-ball » algorithm for that purpose (see e.g., N. Qian,*On the momentum term in gradient descent learning algorithms*, Neural Networks 12 (1999) 145) in our nuclear energy density functional context has several advantages:

- Reduction of the necessary number of iterations for typical nuclear energy density functional calculations by a factor 2-10 at same numerical cost per iteration.
- No significant change in necessary iteration number when changing the resolution scale in coordinate space
- Improved behaviour in pathological cases
- Possibility to automatically adaptat numerical parameters to a given resolution, box size, nucleus, state (no human case-by-case fine-tuning of parameters necessary!)

Ryssens, Bender, Heenen, EPJA 55 (2019) 130

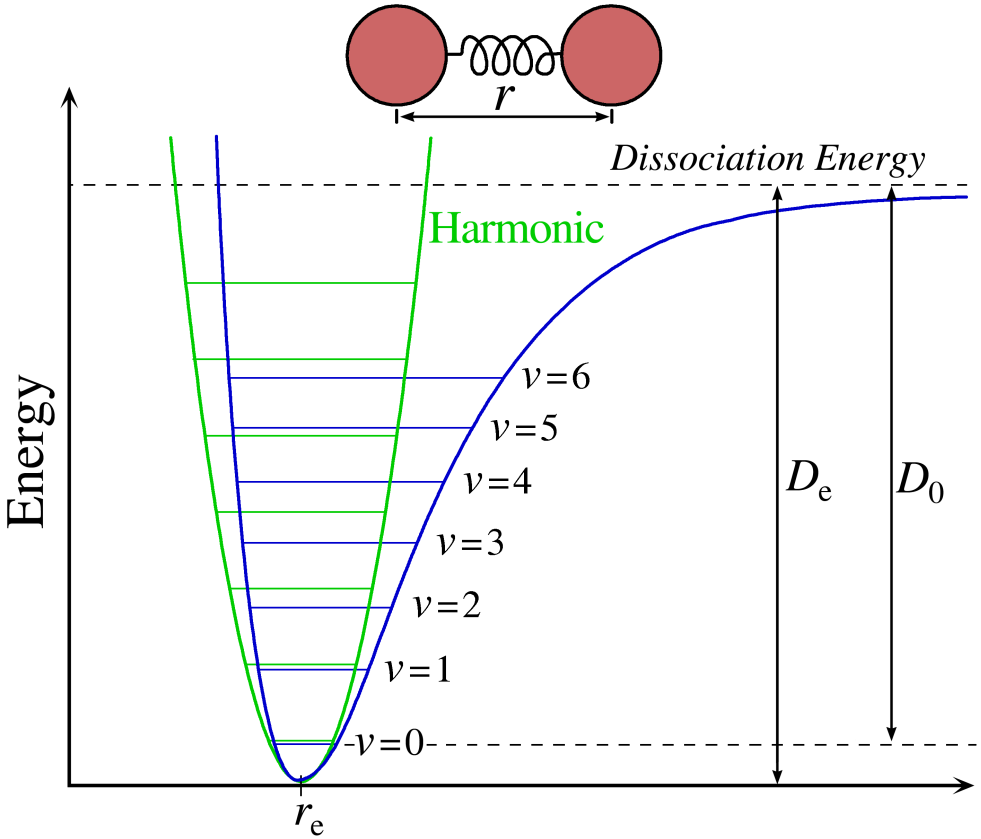


HB: heavy-ball
GD: standard gradient descend

Black dot every 100 iterations.

(LM & PP are two different schemes to achieve self-consistency that are irrelevant here)

Machine learning for nuclear many body problem



Problème : trouver les **niveaux d'énergie** et fonctions d'onde associées du potentiel nucléaire (bleu)

Méthode usuel : approximation par une **base de fonctions** d'un problème plus simple (harmonique, vert) et **minimisation** de l'énergie.

Groupe théorie,
H. Hansen, J. Margueron

Pourquoi du ML : Universal approximation theorem

- *représenter les fonctions d'ondes par un réseau avec l'objectif de minimiser l'énergie.*
- *Agnostique sur la forme des fonctions (pas besoin d'une base d'un problème connu)*
- *Compression de l'information*
- *Performant pour la minimisation*

Machine learning the deuteron,
J.W.T Keeble, A. Rios, arXiv:1911.1309 (Nov 2019)

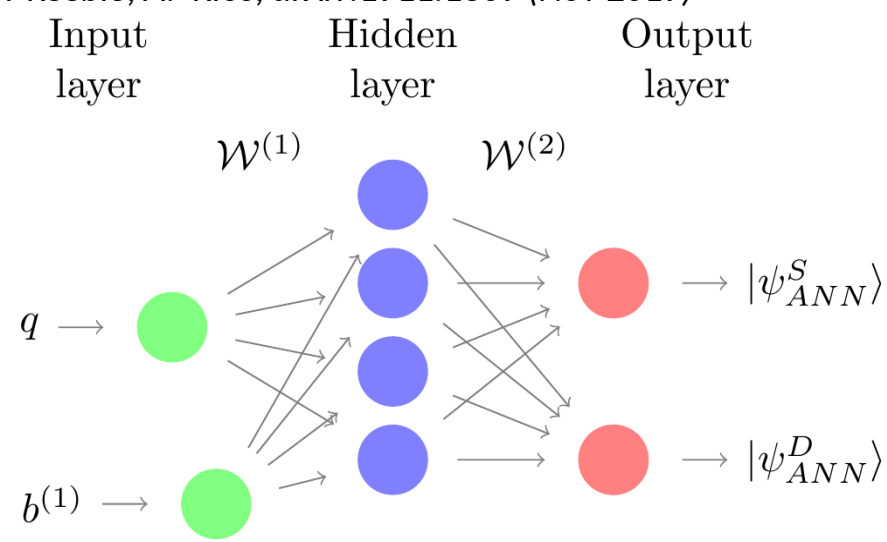
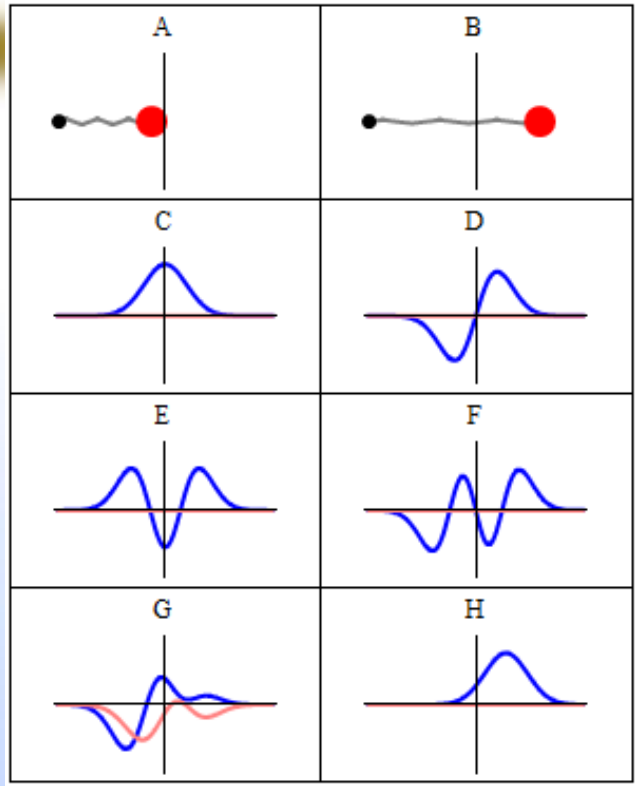


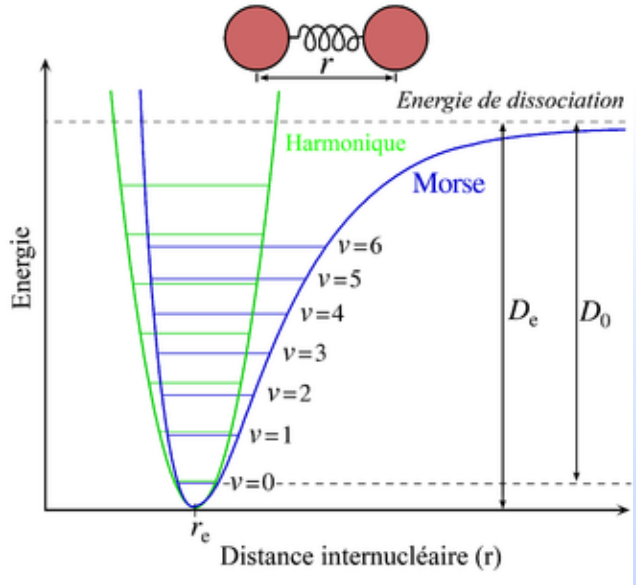
Figure 1: ANN architecture used in this work. The input is a single value of momentum, q, and the wave functions are modelled in terms of a single-layer network. In the example above, the number of hidden nodes is $N_{\text{hid}} = 4$. The ANN has two outputs, one for the S and one for the D state.

Projet :

- *Caractériser le gain de temps entre algorithme usuel et ML*
- *Extension au calcul des états excités*
- *Dynamique ?*

Problème spectral





(Wikipedia)

$$\left(-\frac{\hbar^2}{2m}\vec{\partial} + V(x)\right)\psi(x) = E\psi(x)$$

Déterminer E et $\psi(x)$ satisfaisant l'équation $\Longrightarrow$ **problème d'analyse**

Machine learning et NN adapté à ce problème ?!?

Réseaux de neurones et analyse

Théorème d'approximation universelle

Les NN peuvent approximer toutes fonctions continues (sur un compact)

Minimisation

Les NN sont adaptés pour les problèmes de minimisation

Méthode variationnelle :

Parmi toutes les fonctions, celle minimisant l'énergie est la solution de plus basse énergie.

Usuellement

Choix d'une certaine forme (base) pour les fonctions (pas agnostique, guidé par la physique) puis minimisation.

NN

$\Rightarrow$ on représente les fonctions par un NN (agnostique)

$\Rightarrow$ on donne l'objectif de minimiser l'énergie

Étude d'un problème connue

Objectif : apprendre au réseau à reproduire une solution connue ($E = 0.5$)

NN complètement agnostique sur la forme de la fonction

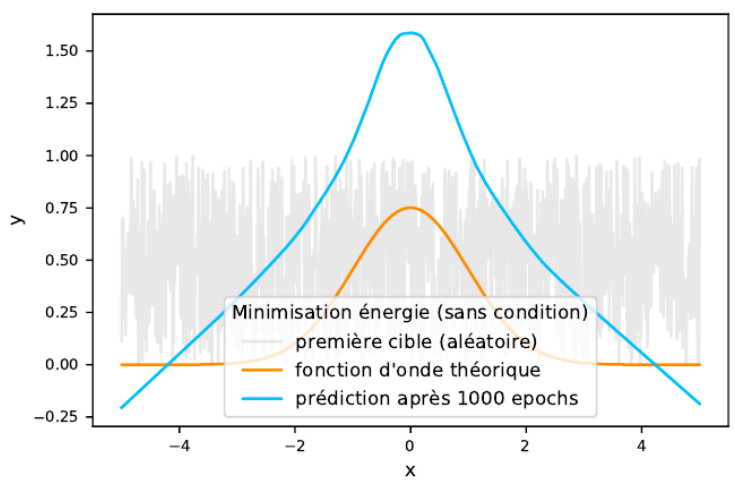


Fig. 5. Première version du programme. 1000 itérations, discrétisation sur 1000 points.

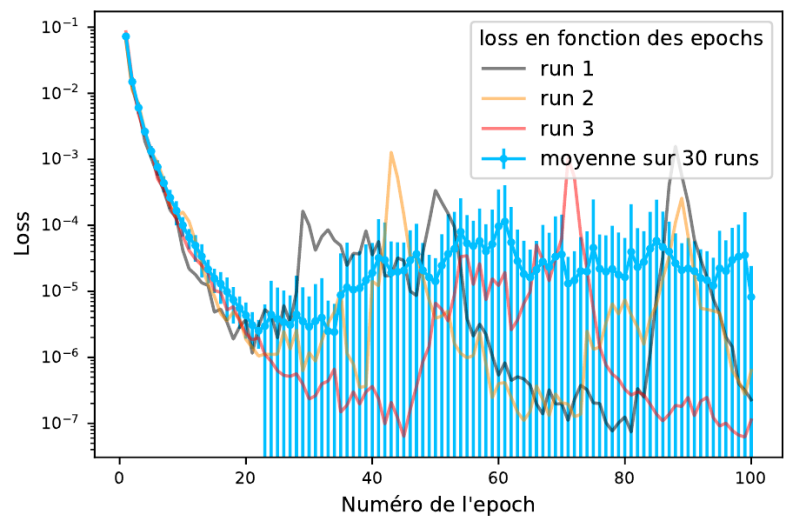


Fig. 3. Écart quadratique moyen (*loss*) entre ψ_{EF} et les prédictions du réseau (200x200x200) en fonction des epochs.

Algorithme

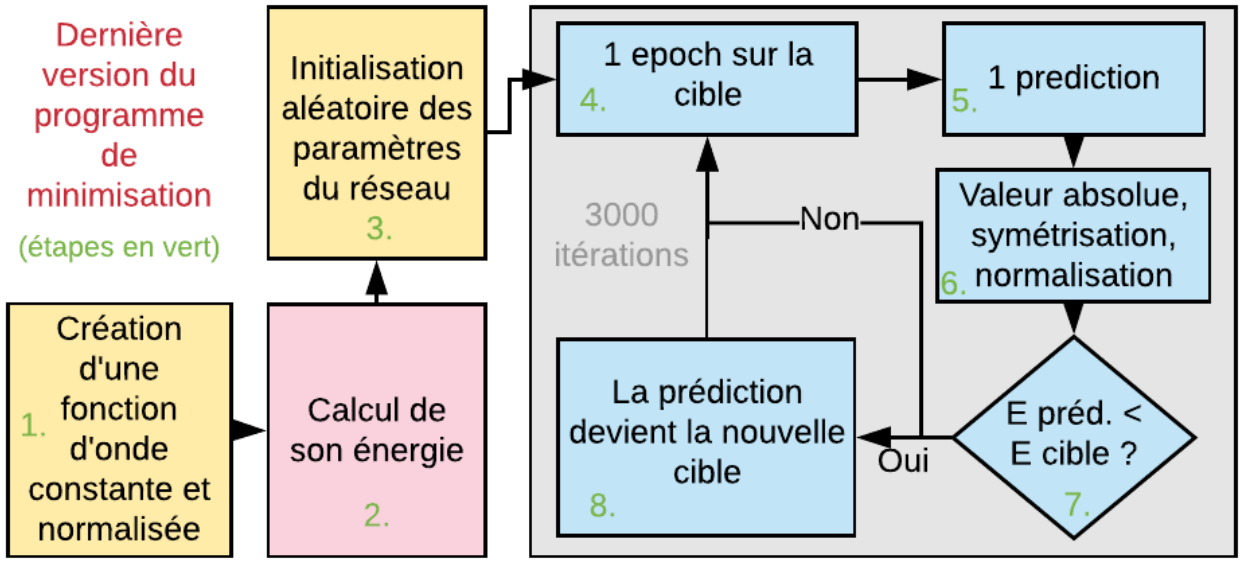


Fig. 6. La première cible est une fonction constante. On garde la prédiction avec l'énergie la plus faible en sortie de boucle.

Étude d'un problème connue

Objectif : apprendre au réseau à reproduire une solution connue ($E = 0.5$)

Ajout de quelques contraintes sur la fonction

Positive, symétrique, normée (quasi-agnostique)

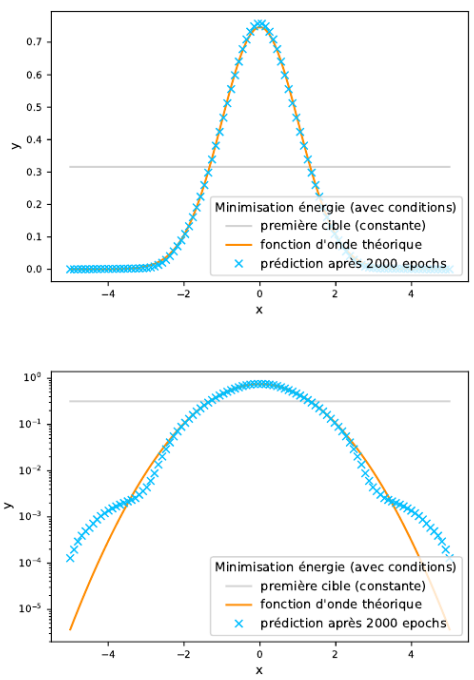


Fig. 7. Résultat moyen de la dernière version du programme en échelle linéaire puis logarithmique. 2000 itérations, discrétisation sur 100 points.

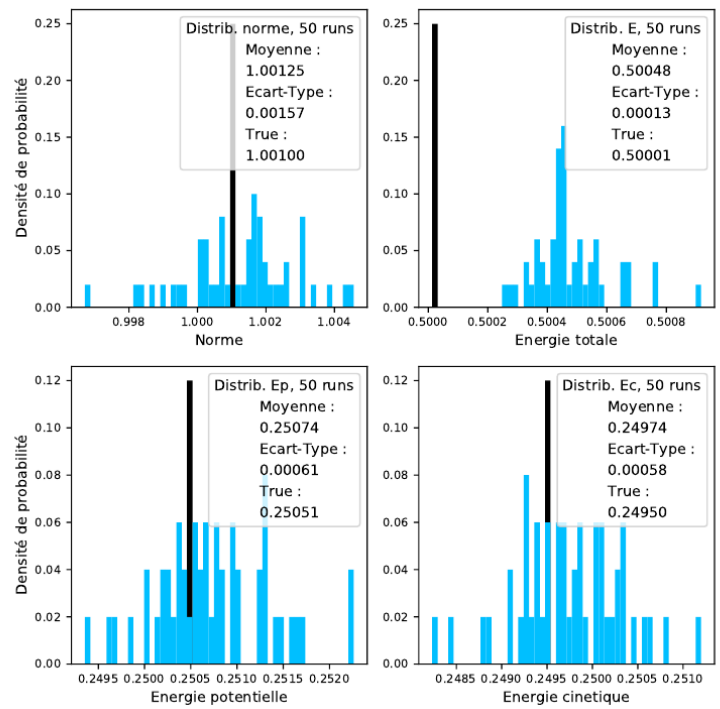
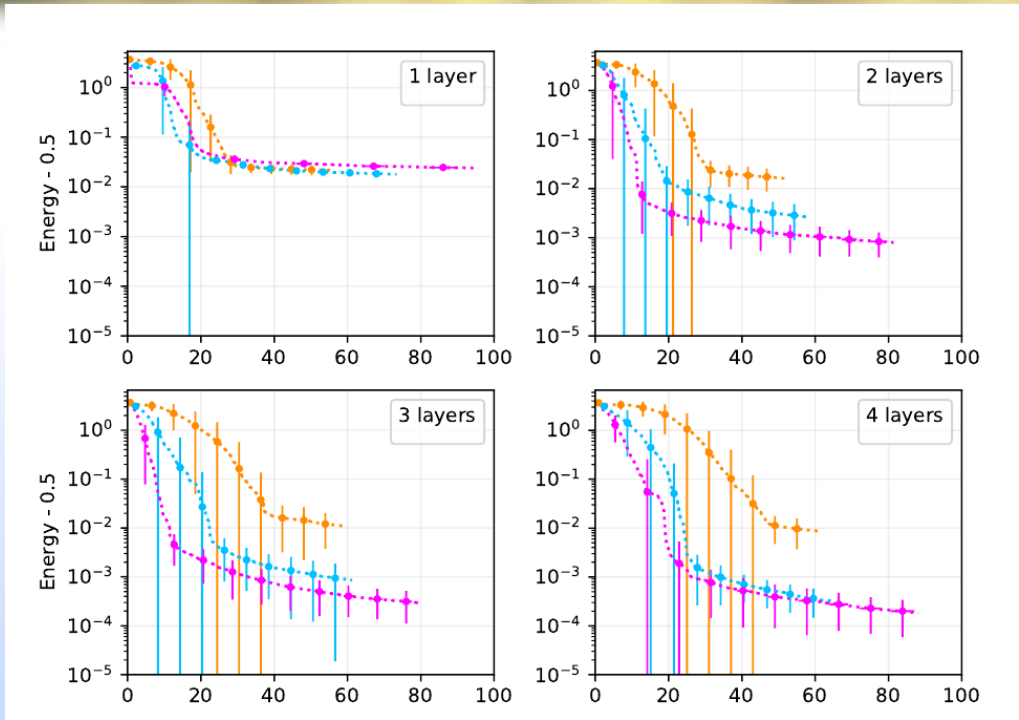
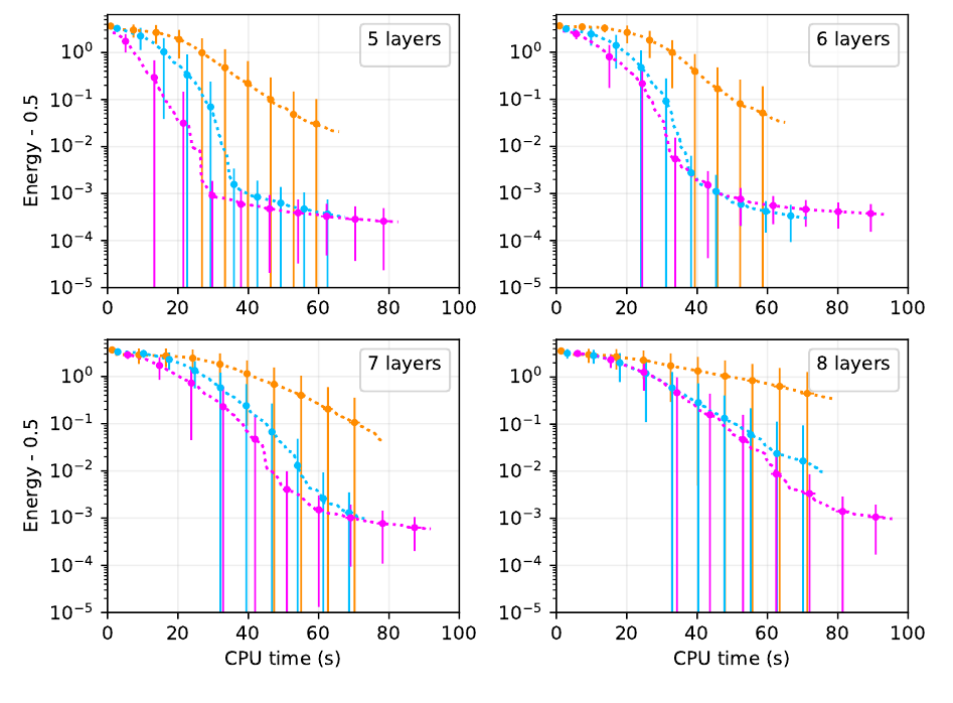


Fig. 4. Distribution des énergies et des normes de 50 prédictions du réseau. Les barres noires sont des repères visuels de la fonction à fitter.

Convergence





Orange : 500 paramètres ; Bleu : 5000 ; Rose : 30600. Les courbes s'arrêtent lorsque le programme a effectué 10000 itérations, à l'exception de "1 couche - rose" qui est coupée à 5000 itérations pour des raisons de lisibilité (200 secondes au total). La précision et le temps sont moyennés sur 30 runs.

Test avec un potentiel différent

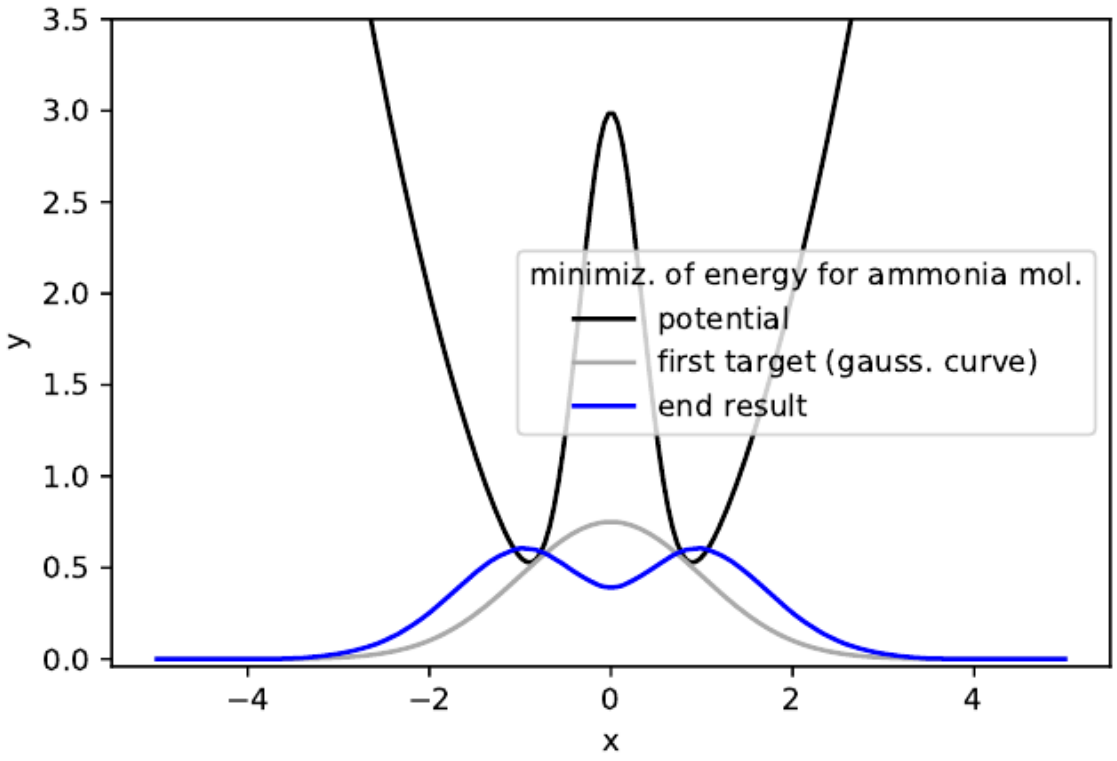


Fig. 10. Minimisationde l'énergie pour les potentiels de l'équation (7) Discrétisation de l'espace sur 200 points. 7000 itérations.