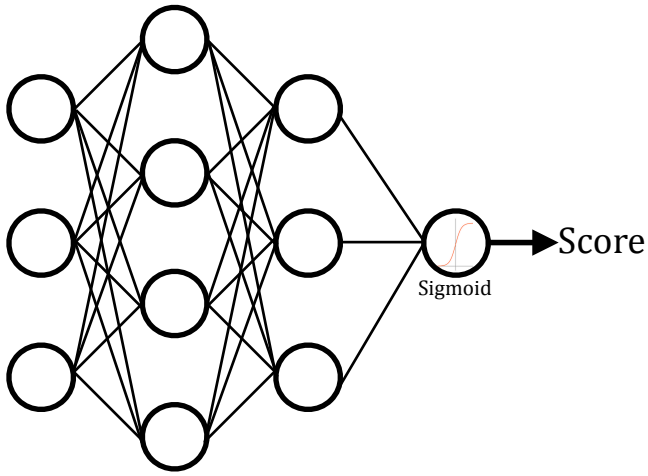
Optimisation des hyper-paramètres

Hyper-paramètres

Lors de l'entrainement d'un réseau de neurones de nombreux paramètre propre au réseaux peuvent être optimiser au niveau de l'architecture et de l'entrainement:

- Nombre de couche
- Nombre de neurones
- Nombre d'époques
- Dropout
- Paramètres libre lier à la loss
- …



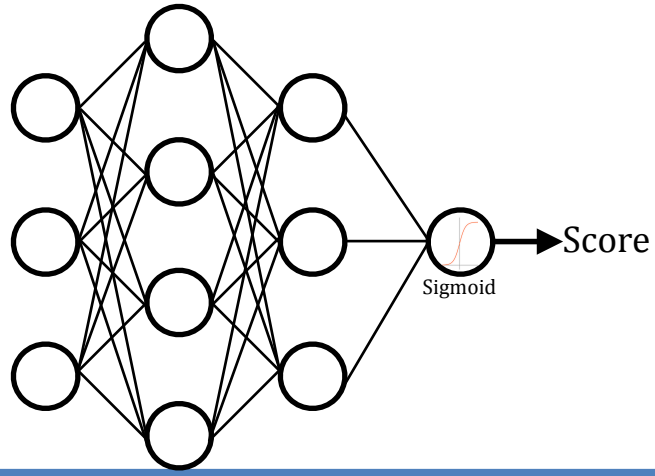
Comment les optimiser ?

Approche manuel

Essaie de différente combinaison manuellement :

Intuitions :

- Commencer par un « faible » nombre de couche puis augmenter
- Nombre de neurones adapter au nombre de paramètres
  ~~50 inputs -> 3 neurones~~ et ~~5 input -> 500 neurones~~
- Evolution de la loss avec les époques s'arrêter au plateau/sur-entrainement



Optimisation automatique

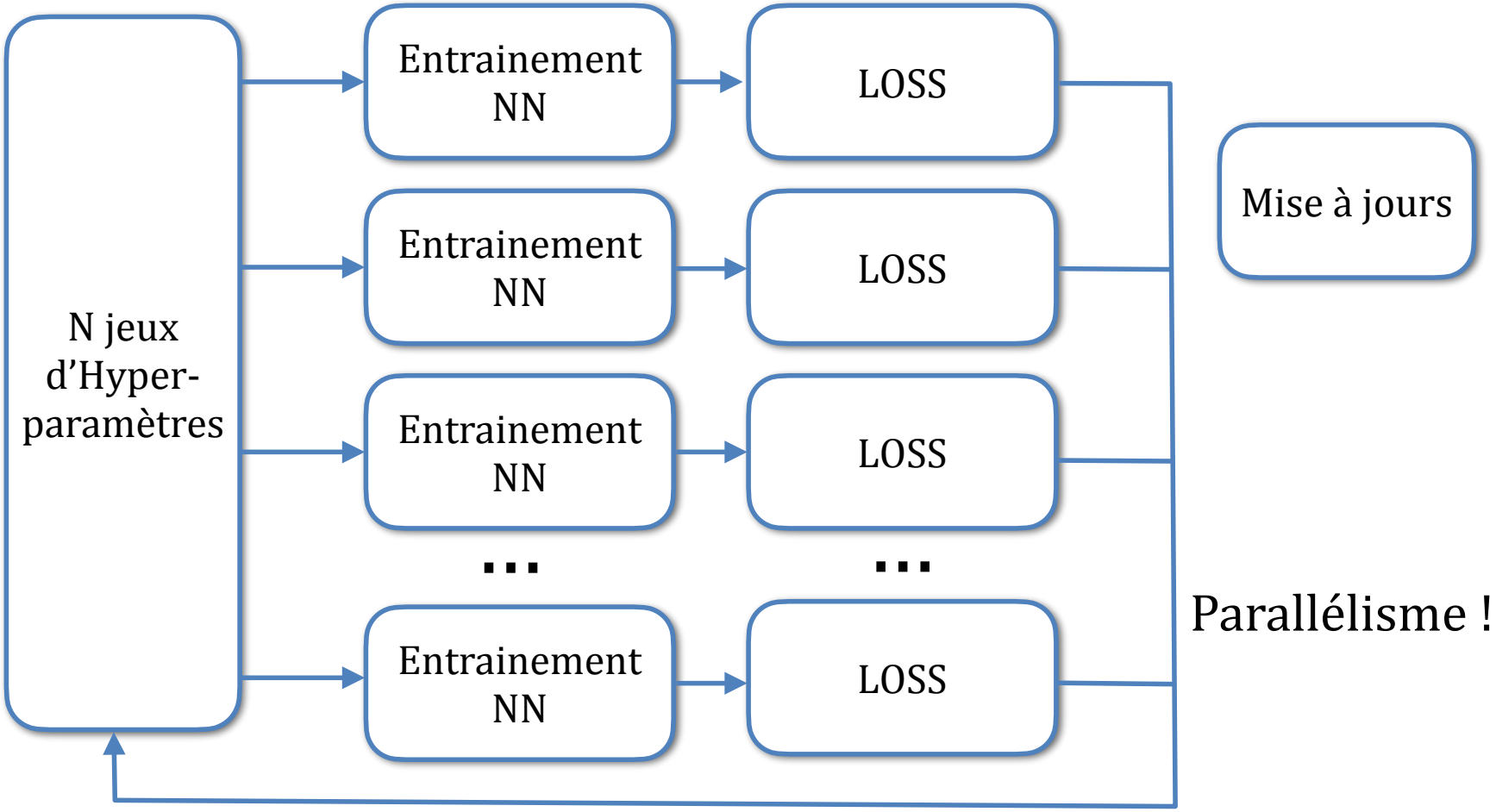
Enormément de bibliothèque pout l'optimisation automatique :

- Orìon
- Optuna
- **KerasTuner**

Différente bibliothèques indépendante + algorithme intégré à chaque bibliothèque de ML

Optimisation automatique

Minimisation d'une fonction de coût sans descente de gradient à partir de paramètres



Optimisation automatique

Différent algorithmes d'optimisation possible :

- Random search -> lent mais converge toujours vers la meilleurs solution
- Bayesian Optimisation
- Hyperband

} Utilise les précédent essaie pour determiner quels nouveaux essayer

- …

Chaque optimiser a ces propres algorithmes

KerasTuner

```
pip install keras-tuner
```

Interface directement avec Keras :

Déclare un object hyper-paramètre qui sera optimisé ensuite

Plus de detail dans le TP