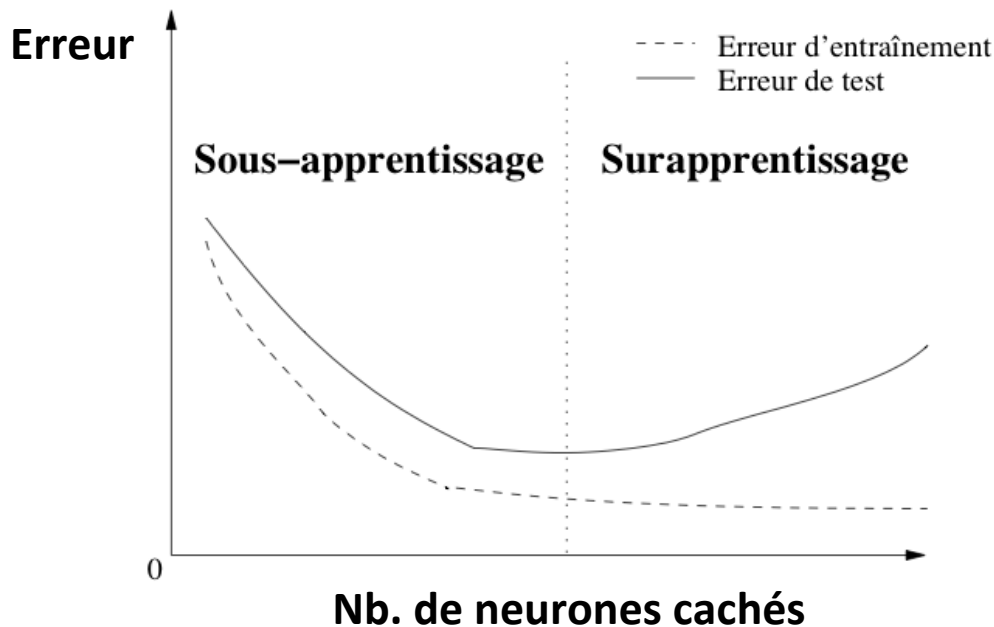


Mesure de la performance d'un algorithme d'apprentissage

- Comment évaluer le succès d'un algorithme?
 - ◆ on pourrait regarder l'erreur moyenne commise sur les exemples d'entraînement, mais cette erreur sera nécessairement optimiste
 - » h a déjà vu la bonne réponse pour ces exemples!
 - » on mesure donc seulement la capacité de l'algorithme à **mémoriser**
 - » dans le cas 1 plus proche voisin, l'erreur sera de 0!
- Ce qui nous intéresse vraiment, c'est la capacité de l'algorithme à **généraliser** sur de **nouveaux exemples**
 - ◆ reflète mieux le contexte dans lequel on va utiliser h
- Pour mesurer la généralisation, on met de côté des exemples étiquetés, qui seront utilisés seulement à la toute fin, pour calculer l'erreur
 - ◆ on l'appelle l'**ensemble de test**

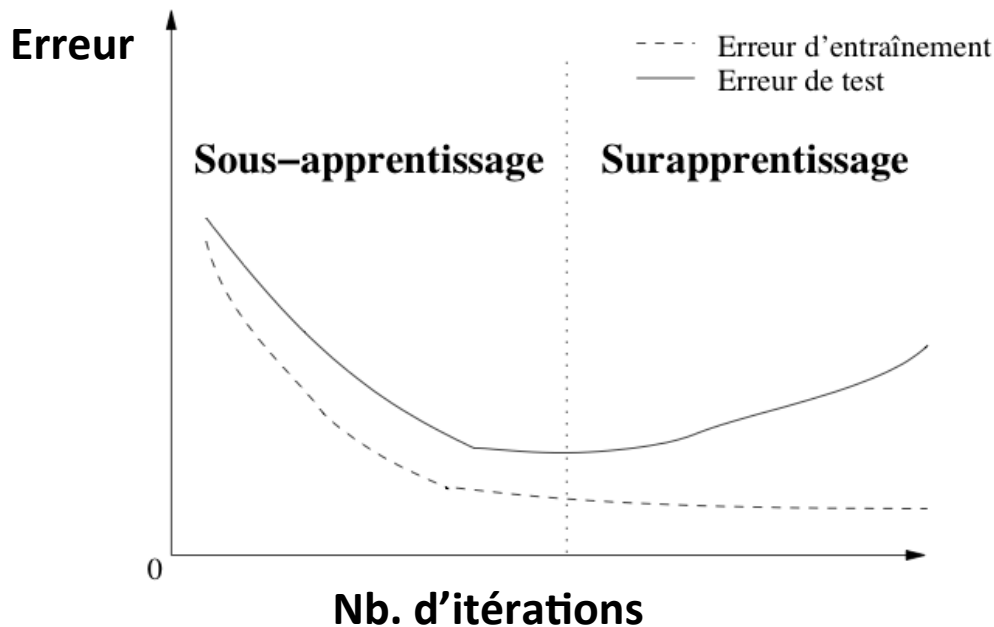
Retour sur la notion de généralisation

- Quelle est la relation entre l'erreur d'entraînement et de test ?



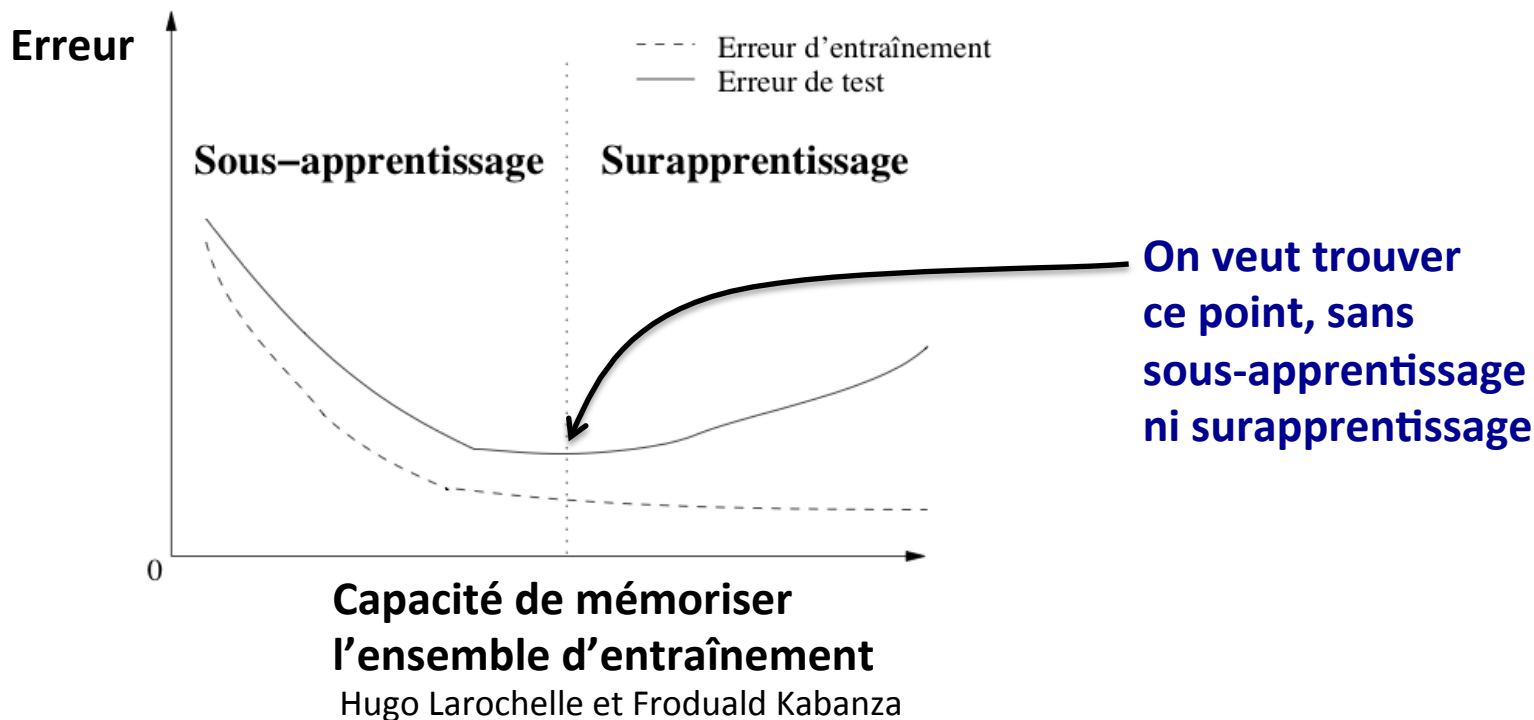
Retour sur la notion de généralisation

- Quelle est la relation entre l'erreur d'entraînement et de test ?



Retour sur la notion de généralisation

- Quelle est la relation entre l'erreur d'entraînement et de test ?



Hyper-paramètres

- Les algorithmes d'apprentissage ont normalement des « options » à déterminer
 - ◆ k plus proche voisins: la valeur de « k »
 - ◆ Perceptron et régression logistique: le taux d'apprentissage α , nb. itérations N
 - ◆ réseau de neurones: taux d'apprentissage, nb. d'itérations, nombre de neurones cachés, fonction d'activation $g(\cdot)$
- On appelle ces « options » des **hyper-paramètres**
 - ◆ ne pas choisir en fonction de l'erreur d'entraînement (mène à du surapprentissage)
 - » pour le k plus proche voisin, l'optimal sera toujours $k=1$
 - ◆ on ne peut pas utiliser l'ensemble de test non plus, **ça serait tricher!**
 - ◆ on garde plutôt un autre ensemble de côté, l'**ensemble de validation**
- Sélectionner les valeurs d'hyper-paramètres est une forme d'apprentissage

Procédure d'évaluation complète

- Utilisation typique d'un algorithme d'apprentissage
 - ◆ séparer nos données en 3 ensembles: entraînement (70%), validation (15%) et test (15%)
 - ◆ faire une liste de valeurs des hyper-paramètres à essayer
 - ◆ pour chaque élément de cette liste, lancer l'algorithme d'apprentissage sur l'ensemble d'entraînement et mesurer la performance sur l'ensemble de validation
 - ◆ réutiliser la valeur des hyper-paramètres avec la meilleure performance en validation, pour calculer la performance sur l'ensemble de test
- La performance sur l'ensemble de test est alors une **estimation non-biaisée** (non-optimiste) de la performance de généralisation de l'algorithme
- On peut utiliser la performance pour comparer des algorithmes d'apprentissage différents