

Notes sur le livre Machine Learning for Humans

Introduction & contexte

La recherche sur les intelligences artificielles existe depuis plus de 40 ans

→ L'application devient possible grâce aux performances grandissantes des calculateurs

2015 : L'AI peut faire office d'Helpdesk & communiquer avec un humain sur des thématiques simples et limitées sans que celui-ci comprenne qu'il échange avec une machine

2017 : OpenAI a créé des bots qui ont inventé leur propre langage, « plus efficace »

2017 : OpenAI a créé une AI capable de battre le world record à Dota2

AI déjà présente dans nos vies : Google Translate utilise l'appareil photo et traduit instantanément en la langue souhaitée (même les sinogrammes)

Le machine learning s'inclue dans l'AI :

Intelligence artificielle

Conception d'un agent intelligent qui analyse son environnement et réagit en conséquence pour maximiser ses chances d'atteindre son objectif

Machine Learning

Apprentissage autonome des machines, qui visent à améliorer leur stratégie en termes de planification, décision, conception et action

Apprentissage supervisé

Classification, régression

Apprentissage non-supervisé

Groupage, recommandation

Apprentissage renforcé

Décider à partir de l'expérience acquise de la marche à suivre pour optimiser une récompense quantitative

Le machine learning c'est permettre aux ordinateurs d'apprendre par eux-mêmes, identifier des éléments caractéristiques, construire des modèles descriptifs de l'environnement & prédire sans avoir de règle ou de modèle préalablement programmés.

Apprentissage supervisé

On a un dataset d'entraînement avec des marqueurs associés identifiant les éléments caractéristiques intéressants

Régression : Estimer une valeur (prédire une variable cible continue)

→ ex : pour combien va se vendre telle maison ?

Classification : Décider de l'appartenance à une classe (prédire une variable cible discrète)

→ ex : est-ce un chien ou un chat ?

Régression

$$Y = f(X) + \varepsilon$$

$$\bar{\varepsilon} = 0$$

$$X \in \mathbb{C}^{m \times n}, m, n > 0$$

Entraînement de la machine : ordinateur trouve fonction f à partir du dataset d'entraînement

Test : Ordinateur prédit Y à partir d'un dataset de test (sans marqueurs)

Régression linéaire ignorée car non-utilisée dans la reconnaissance d'images

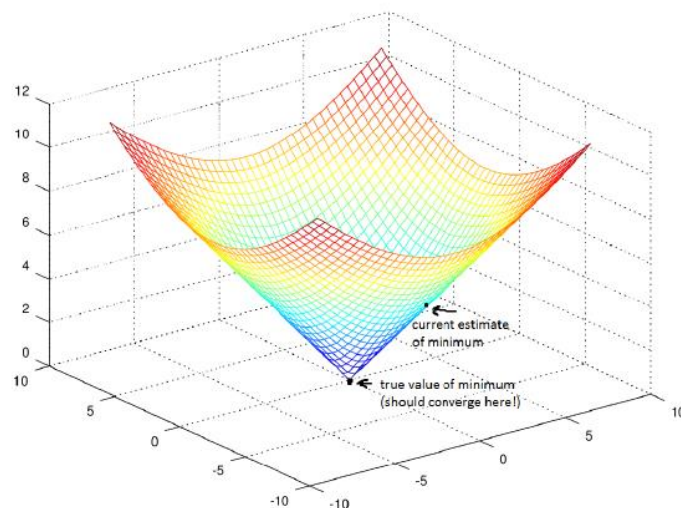
Diminution graduelle

On souhaite minimiser la fonction de perte en trouvant la meilleure approximation possible du modèle

→ ex : problème de l'aveugle qui cherche le fond de la vallée

$$Cost = \frac{\sum_{i=1}^n ((\beta_1 x_i + \beta_0) - y_i)^2}{2n}$$

Variables : β_0, β_1



Estimation de β_0 et β_1 « à l'aveugle » puis calcul des dérivées partielles de z pour estimer l'impact des variations des bêtas sur la perte.

Trouver le minimum revient à faire converger l'algorithme

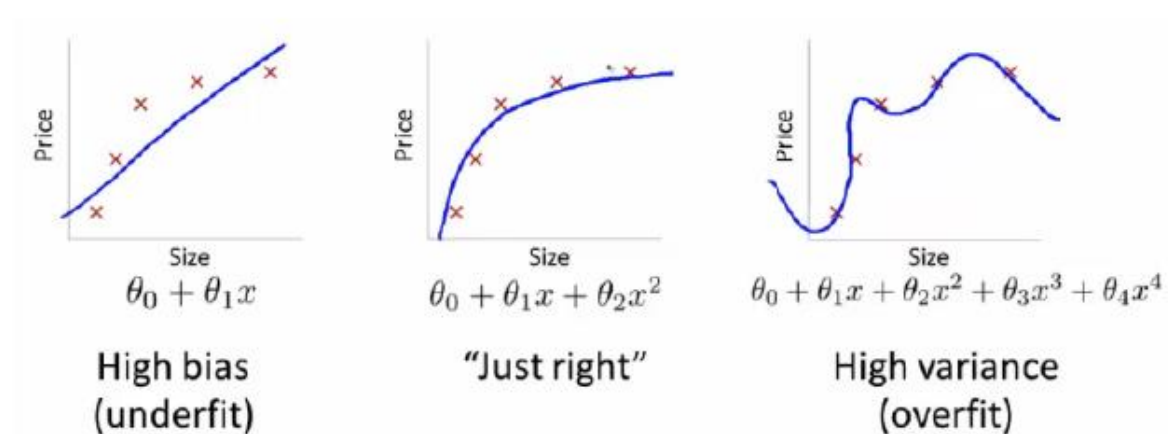
Dim graduelle intéressante pour trouver les paramètres optimaux des équations paramétriques

Surestimation / hyperstatisme (overfit)

Le modèle peut trouver une fonction qui explique parfaitement le dataset d'entraînement mais qui du coup est trop spécifique & ne peut pas s'appliquer au dataset de test

→ Il prend en compte les particularités du dataset d'entraînement, qui ne sont pas représentatives de l'environnement

→ Un bon modèle a un faible biais et une faible variance



Pour pallier l'hyperstatisme on peut utiliser un + gros dataset d'entraînement ou régulariser en ajoutant un terme à la fonction de coût qui pénalise les termes singularisant trop la fonction de coût

Classification

Pour classifier (variables discrètes)

Idée : associer un élément d'une image à une classe, l'ensemble des classes est déterminé par le dataset d'entraînement

En + de ça il donne la proba de certitude de classif (c'est un chat à 98%)

Véracité de classif : dépend efficacité algo, application, quantité d'information pertinente

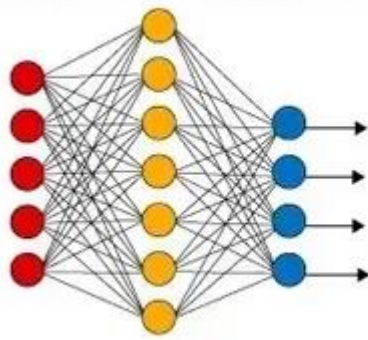
Apprentissage profond

On veut toujours apprendre f telle que $Y = f(X) + \varepsilon$

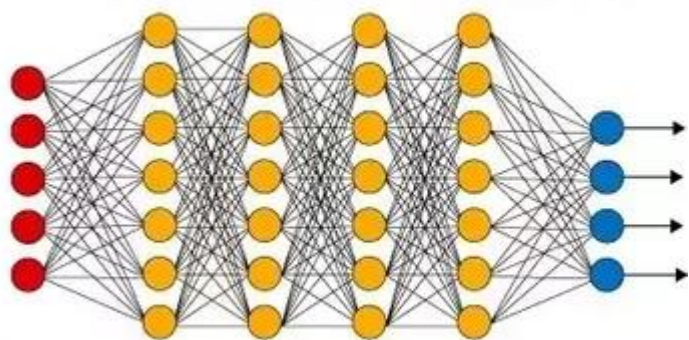
En image les pixels sont vecteur d'une quantité colossale d'informations

On ne sait pas comment des méthodes aussi simples que la régression linéaire réagiraient en présence d'éléments aussi complexes

Simple Neural Network



Deep Learning Neural Network

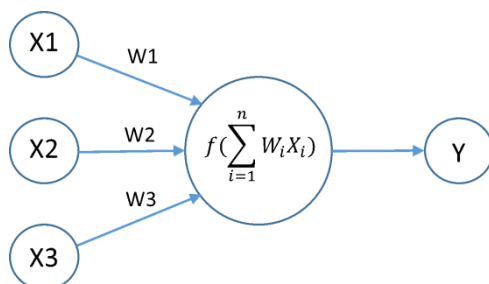
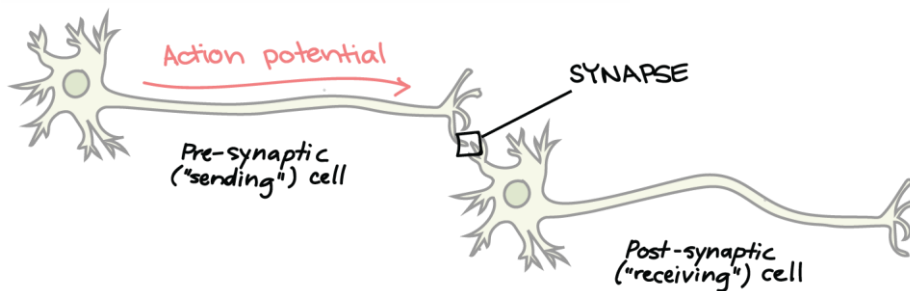
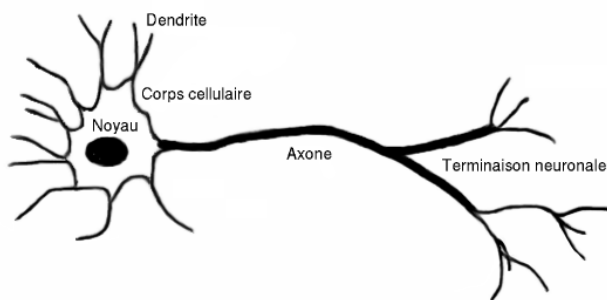


● Input Layer ● Hidden Layer ● Output Layer

+ haute probabilité : celle de l'objet recherché

Les couches intermédiaires font des multiplications matricielles (fonction d'activation) et on minimise les paramètres de la fonction de coût avec la diminution graduelle

Deep learning date de 2006



Les cerveaux animaux sont faits de neurones qui émettent des signaux électriques à d'autres neurones après avoir eux-mêmes été stimulés (activés). Les signaux reçus influencent différemment (plus ou moins fortement) l'activation dudit neurone en réception.

Neurones ont différents rôles : détectent soit des détails soit des objets plus abstraits

Modèles linéaires impossibles pour deep learning : supposons on fait une moyenne de toutes les images appartenant à la même classe : on se retrouve avec un cheval à 2 têtes, par exemple

En calculant ensuite la distance de chaque pixel non-classé via un algo de + proche voisin → ça ressemble à rien ton image moyennée (car pas de niveaux d'abstraction)

Les couches intermédiaires permettent de ne pas perdre le niveau d'abstraction → leurs neurones détectent les concepts abstraits les + utiles pour capturer le + d'infos & réduire l'erreur

→ C'est ce qui fait que c'est si difficile d'expliquer ce qu'il se passe dans les couches intermédiaires puisqu'on compare l'image d'entrée à des concepts abstraits donc c'est pas trop intelligible d'un point de vue humain

Plusieurs réseaux pourraient être testés dans notre étude dont certains sont entièrement dédiés à la reconnaissance de caractéristiques dans les images

Déjà utilisé : les voitures autonomes reposent sur le deep learning pour identifier & interpréter la signalétique et les obstacles sur la route

Apprentissage renforcé

Souris dans labyrinthe : elle détecte la source d'eau elle va l'exploiter à fond une fois la source découverte mais ne part pas à la recherche de source plus intéressante

Compromis expérience / exploitation : déterminer un % décrivant la fréquence à laquelle la souris emprunte un chemin différent (aléatoire) dans le but d'explorer ce qu'il se passe ailleurs (dans l'espoir d'une source plus intéressante, évidemment)

On parle de stratégie ϵ -radin(e) : epsilon vaut 20% ici

Comme la souris améliore sa connaissance du labyrinthe progressivement epsilon va tendre à diminuer (< 10%)