



# Reinforcement learning, les bienfaits de l'expérience

*Inria*

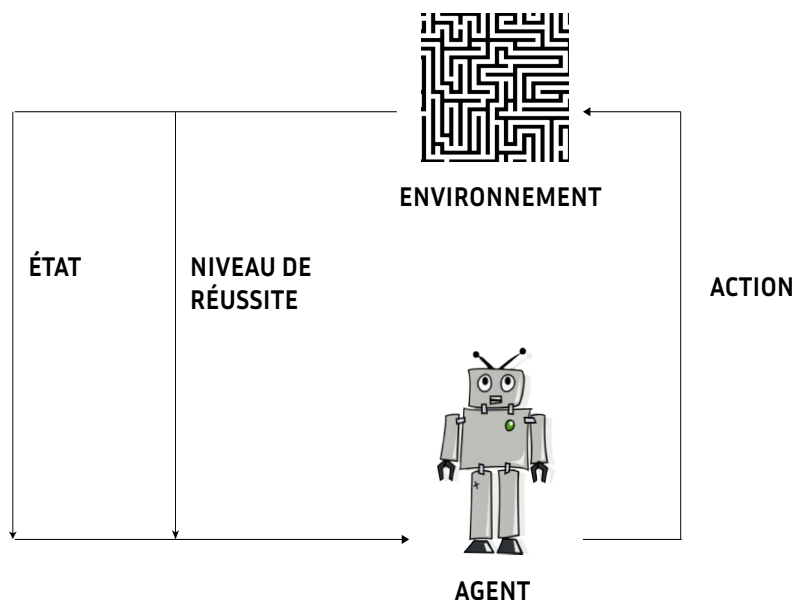
## PROCESSUS

Le reinforcement learning (apprentissage par renforcement, ou RL) est une branche de l'Intelligence Artificielle qui permet à un programme d'apprendre par l'expérience.

Le principe est le suivant : un algorithme (agent) interagit avec un environnement incertain ou inconnu pour apprendre quel comportement avoir, afin de réaliser une tâche le mieux possible (maximiser le niveau de réussite).

Une des particularités de cet algorithme est qu'il apprend tout seul. Lorsque l'algorithme est efficace, les résultats sont meilleurs que ceux obtenus par des humains sur les mêmes tâches.

L'expert humain peut donc s'inspirer de ce qui a été appris.



## CARACTÉRISTIQUES

### Besoins nécessaires

Pour fonctionner, il faut mettre en place un environnement (réel, simulateur ou modèle) pour extraire des cas d'apprentissage.

Il est également nécessaire de pouvoir extraire un signal qui note le niveau de réussite de l'algorithme selon le résultat attendu.

## USE CASES

Les domaines d'applications sont nombreux :

- déplacement autonome d'une machine (robotique),
- santé (stratégie de soins),
- gestion complexe (pilotage des centrales électriques, gestion d'investissement en bourse),
- découverte de stratégie (jeu de Go),
- logistique (optimisation de trajet), etc.

## FONCTIONNALITÉS GÉNÉRIQUES

À partir d'un environnement, d'une liste d'actions possibles et d'un objectif à résultat quantifiable, le système apprend le comportement qui maximisera l'objectif.

L'agent observe l'environnement pour savoir dans quel état il se trouve. En fonction de sa politique (expériences précédentes ou connaissances) et d'une possibilité d'exploration, une action est choisie.

Suite à cette action, l'environnement se met à jour et envoie à l'agent la récompense (positive ou négative) correspondant à son action. L'agent peut alors observer l'environnement pour découvrir son nouvel état.

Il utilise l'ancien état, l'action choisie, la récompense et le nouvel état pour mettre à jour sa politique. L'agent recommence une nouvelle action à partir du nouvel état et de ses connaissances.

**Exploration** : l'agent essaye des actions variées pour augmenter sa connaissance de l'environnement afin de trouver la nouvelle meilleure action.

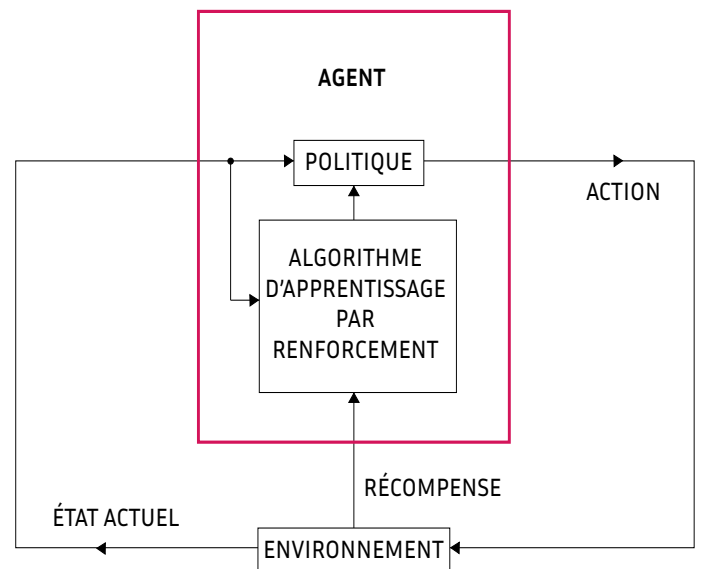
**Exploitation** : l'agent choisit la meilleure action selon ses connaissances actuelles pour maximiser sa récompense.

L'objectif est de trouver un compromis pour découvrir les meilleures actions, sans sacrifier trop de récompenses.



## FICHE IDENTITÉ

- **Langages de programmation** : Python (le plus utilisé), C, C++
- **Outils** : scikit-learn, tensorflow ou pytorch
- **Caractéristiques** : Modélisation de la tâche à optimiser par un problème de décision de Markov. Gestion du dilemme exploration/exploitation. Essai-erreur pour découvrir les meilleures solutions
- **Équipe projet** : Scol\* - <https://team.inria.fr/scool/>



## READ ME

<http://www.incompleteideas.net/book/RLbook2018.pdf>

Référent : Philippe PREUX

\* Scol est une équipe-projet commune à Inria et au laboratoire CRISTAL (Centrale Lille, CNRS, Université de Lille)





# Deep learning : apprendre avec des exemples

*Inria*

## CARACTÉRISTIQUES

Le Deep learning (apprentissage profond, ou DL) est une branche de l'Intelligence Artificielle qui permet à un programme d'apprendre un modèle à partir d'exemples.

### Données utilisées :

Le DL permet d'analyser de nombreux types de données : images, son, texte en langage naturel (langage humain tel qu'il est parlé), données numériques.

Pour permettre au système d'apprendre, il faut un grand nombre d'exemples. Ces exemples sont des données ayant chacune une étiquette associée.

### Exemple :

Donnée : une image avec un animal (photographie de chien, chat, etc.). La donnée peut être de différents types et composée de plusieurs éléments (texte, nombre, etc.)

Étiquette : le nom de l'animal présent sur l'image («chien» ou «chat»). L'étiquette peut être de différents types (nombre, phrase de légende, classification, etc.)

## USE CASES

Il y a beaucoup d'applications du DL :

- Reconnaissance d'objets ou de caractéristiques dans une image,
- Traitement de la langue naturelle (par exemple un assistant personnel qui répond à ce qu'on lui demande),
- Couplage de traitement d'images et de textes en langue naturelle (par exemple dans le but de fournir une audio-description automatique de sites web aux personnes malvoyantes).

## PROCESSUS

L'objectif est qu'à partir d'un ensemble de couples (donnée, étiquette), on induit un modèle qui permet, une fois le modèle appris, de prédire l'étiquette de n'importe quelle nouvelle donnée.

Le Deep learning fonctionne en deux grandes phases : la première où l'on présente des exemples au programme afin qu'il apprenne un modèle de la tâche à réaliser et la seconde où la méthode est utilisée sur de nouvelles données.

### Phase d'apprentissage :

L'apprentissage nécessite des exemples à analyser avec le résultat attendu. Si on veut détecter un objet dans une image, les exemples doivent contenir des couples : image et résultat attendu (le nom de l'objet).

### Phase de prédiction :

Il s'agit de présenter de nouvelles données afin que le programme prédise le résultat selon le modèle obtenu en phase d'apprentissage.

### QUELS AVANTAGES ?

- À la pointe de la technologie actuelle
- Généralisation à partir d'exemples



## FICHE IDENTITÉ

- Langages de programmation: Python (R, C, C++)
- Outils: tensorflow, pytorch
- Équipe projet: Scool\* - <https://team.inria.fr/scool/>

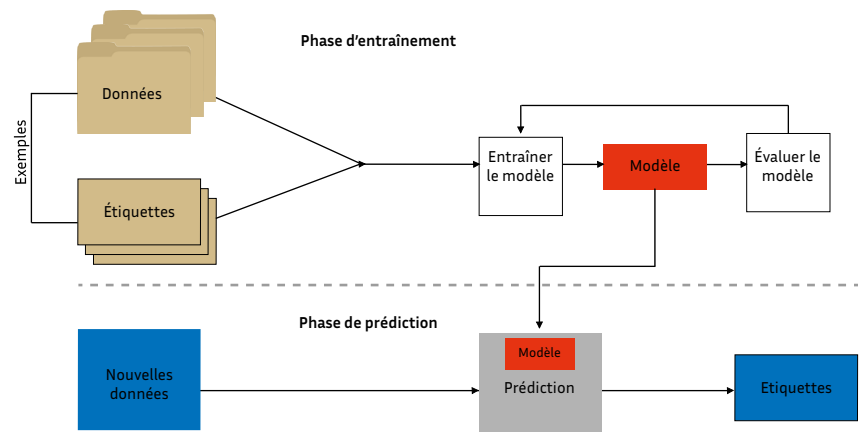
## FONCTIONNALITÉS GÉNÉRIQUES

À partir d'exemples, on peut identifier les étiquettes de nouvelles données.

1 - Avoir des éléments identifiés : un jeu d'exemples de « vérités terrain ».

2 – Entraîner et évaluer le modèle (réseau de neurones) à fournir des résultats proches de ces vérités terrain selon les exemples.

3 – Présenter des nouvelles données au modèle pour inférer leurs étiquettes.



Le Deep learning consiste à essayer de modéliser la relation entre des données d'entrée et le résultat attendu (étiquettes) à l'aide de transformations non-linéaires.

**Première étape :** création d'un réseau de neurones. C'est l'outil principal qui permet de réaliser ces transformations non-linéaires. Le choix du type de réseau de neurones dépend de plusieurs critères, principalement le type de données utilisées et l'objectif recherché.

**Deuxième étape :** entraînement du réseau de neurones avec les exemples. On présente les données au réseau de neurones et on observe les résultats obtenus. Ensuite, on compare ces résultats avec la vérité terrain des exemples et on modifie le réglage des neurones selon cette différence (par exemple via descente de gradient et rétropropagation). On recommence le processus jusqu'à ce que les différences entre les résultats du réseau et les «vérités terrain» soient stabilisées.

**Troisième étape :** utilisation du modèle. Les réglages du réseau ayant été gelés, on commence les inférences. Il suffit de lui présenter de nouvelles données et de regarder la sortie obtenue.

## READ ME

<http://deeplearningbook.org>

Référent : Julien TEIGNY

\* Scool est une équipe-projet commune à Inria et au laboratoire CRISTAL (Centrale Lille, CNRS, Université de Lille)





# Systemes de recommandation : un choix réduit et personnalisé

*Inria*

## PRÉSENTATION

Les systèmes de recommandation visent à filtrer les données de façon à ce que les éléments recommandés soient plus intéressants pour l'utilisateur.

Ces éléments sont de natures différentes : films, musiques, traitements médicaux, produits, etc.

Cette recommandation peut se baser sur les préférences et goûts des utilisateurs, leurs choix précédents, et/ou ceux d'utilisateurs similaires.

## EXEMPLES D'UTILISATION

La technologie peut être utilisée pour recommander des morceaux de musique ou des films dans un catalogue.

## CARACTÉRISTIQUES

Les données utilisées sont des retours sur la satisfaction des utilisateurs vis à vis des éléments proposés. Des informations sur le profil des utilisateurs et sur les éléments à recommander peuvent optimiser l'efficacité de la technologie.

Il existe différentes méthodes de recommandation utilisables selon les données disponibles, certaines seront détaillées au verso.

## QUELS AVANTAGES ?

- À la pointe de la technologie actuelle
- Permet de limiter le choix d'un contenu pour l'utilisateur
- Trouver la meilleure solution compte-tenu des caractéristiques de l'utilisateur

## USE CASES

Cette technologie peut s'appliquer à plusieurs domaines, notamment :

- le marketing : meilleure publicité à afficher,
- la santé : meilleur traitement à prendre,
- la vente : meilleur produit pour une utilisation particulière, etc.



## FICHE IDENTITÉ

- Langages de programmation : Python (le plus utilisé), R, C, C++
- Outils : scikit-learn
- Equipe projet : Scool\* - <https://team.inria.fr/scool/>

## FONCTIONNEMENT TECHNIQUE

### Type de recommandations :

#### Filtrage par contenu

Cet outil est utilisable lorsque l'on a des connaissances sur les éléments à recommander. On détermine des critères qui peuvent caractériser les éléments. Selon les éléments qui ont été aimés par un utilisateur, on peut faire une « carte d'identité » de ce dernier en fonction des critères aimés et détestés. Grace à cette carte d'identité et les caractéristiques des éléments non-testés, on peut estimer l'intérêt d'un utilisateur pour ces éléments.

Le système essaye de recommander des éléments similaires à ceux aimés dans le passé. Il est donc facile d'ajouter un nouvel élément tant que l'on connaît ses caractéristiques (pas de démarrage à froid coté élément). Il n'y a pas d'influence des autres utilisateurs : on peut récupérer la carte d'identité de l'utilisateur pour expliquer pourquoi on recommande tel élément (transparence).

#### Filtrage collaboratif

Ce système se base sur le fait que quand des personnes ont eu le même avis sur des mêmes éléments, on suppose qu'ils auront un avis similaire sur d'autres éléments. Pour fonctionner, il y a besoin de données d'entraînement (notes d'utilisateurs sur des éléments par exemple).

Ces informations sont souvent représentées sous forme de matrice. En détectant les similarités, le système fait des recommandations. Cet outil est utilisable lorsqu'on n'a aucune information ni sur les utilisateurs ni sur les éléments, excepté leurs identifiants (pour les différencier) et des retours (notes d'appréciation de l'utilisateur sur l'élément).

Il y a démarrage à froid lors d'un ajout d'un nouvel utilisateur ou d'un élément, ce qui rajoute une réelle complexité.

#### Bandit manchots

Le système se base sur l'apprentissage par renforcement (essai-erreur) et n'a besoin d'aucune donnée pour fonctionner, seulement d'un retour. Cet outil est utilisable lorsque l'on a une situation qui se répète plusieurs fois, avec différentes actions possibles.

Le problème est que l'on ne connaît pas a priori l'efficacité de ces actions, et cette efficacité n'est pas toujours la même (résultat stochastique).

Le but est de trouver un compromis exploration/exploitation pour améliorer les résultats, c'est-à-dire entre (re) tester une action pour améliorer nos connaissances sur son efficacité, et utiliser l'action la plus efficace en fonction des connaissances acquises.

### Problème du démarrage à froid

Cela peut se produire au lancement de l'algorithme, à l'ajout d'un utilisateur ou d'un nouvel élément. Le système ne peut pas faire d'inférences sur les éléments à recommander car il n'a pas assez d'informations.

### READ ME

<https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada>

Référent : Philippe PREUX

\* Scool est une équipe-projet commune à Inria et au laboratoire CRISTAL (Centrale Lille, CNRS, Université de Lille)





# Mangoes, géométrie des mots

*Inria*

## PROCESSUS

La librairie Mangoes fournit des outils pour créer, manipuler et évaluer des représentations vectorielles de mots (aussi appelées Word Embeddings).

Ces représentations encodent les mots d'un vocabulaire de telle façon que les vecteurs représentant des mots de sens similaires se retrouvent proches dans l'espace. Elles sont apprises de façon non supervisée à partir de texte brut.

Mangoes facilite la génération, l'évaluation et l'utilisation de ces représentations, en implémentant ou interfaçant différentes méthodes dans une même librairie : des traditionnelles méthodes statistiques au performant langage de modèle BERT, largement utilisé, qui apprend les représentations vectorielles de mots et de phrases.

Tous les outils sont largement configurables et modulaires:

- Le corpus d'entrée peut être du texte brut ou annoté (par exemple par les catégories grammaticales),
- de nombreux (hyper)paramètres sont disponibles: normalisation du texte, sélection du vocabulaire, échantillonnage du contexte, pondération des cooccurrences, choix du contexte,
- accès à de nombreux modèles et *tokenizers* BERT pré-entraînés,
- utiliser vos propres données pour adapter les embeddings BERT à différentes tâches telles que la classification de texte ou la réponse à des QCM.

## QUELS AVANTAGES ?

- utilisable sur des corpus en français,
- utilisable sur des corpus spécifiques à un domaine pour obtenir des représentations spécifiques à ce domaine,
- modulaire et configurable,
- extraction d'*embeddings* BERT facilitée,
- adaptation de BERT à une tâche facilitée.

## USE CASES

Les représentations vectorielles de mots sont une brique de base de nombreux outils de traitement du langage : ils peuvent donc être utilisés pour différentes tâches, allant de la classification de documents, de phrases ou de mots à des analyses sémantiques plus fines à des tâches de questions-réponses.



## FICHE IDENTITÉ

- **Langages de programmation** : Python
- **Caractéristiques** : Effectuer des tâches courantes sur un corpus : normalisation, comptage de mots, tokenisation ...
- **Équipe projet** : Magnet\* - <https://www.inria.fr/fr/magnet>

## FONCTIONNALITÉS GÉNÉRIQUES

### Création de vocabulaires, de matrices de comptes de cooccurrence et de représentations basées sur ceux-ci :

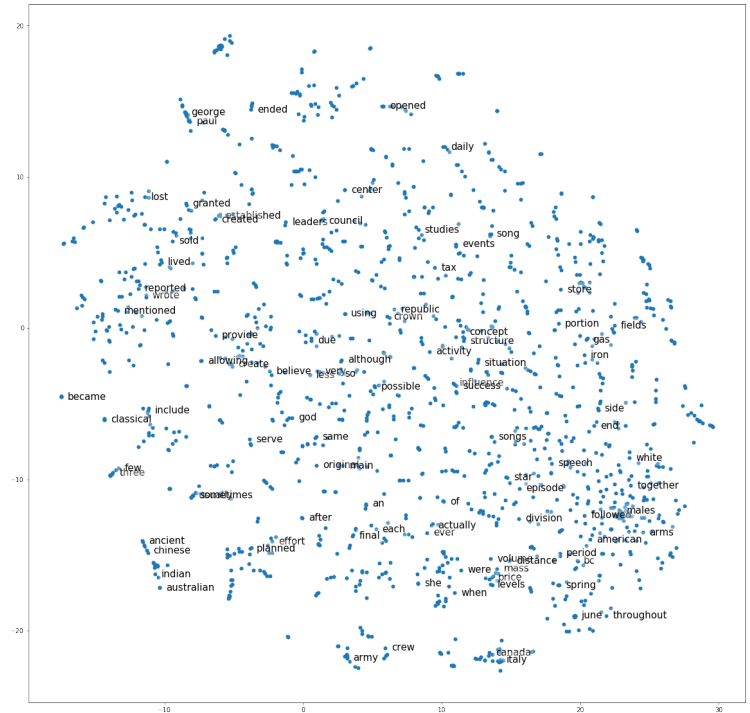
- à partir de textes bruts ou annotés,
- utilisation de différents contextes de comptage : phrase, fenêtre de mots, dépendance syntaxique,
- création de représentation à partir de cette matrice de comptes (PPMI, réduction de dimension, etc.).

### Interface facile à utiliser pour BERT :

- Extraction d'*embeddings* contextuels à partir d'un modèle pré-entraîné,
- Entraînement d'un modèle BERT à partir de zéro sur un grand corpus,
- Ajustement fin d'un modèle BERT pour des tâches spécifiques telles que l'analyse de sentiments, la classification de parties du discours, les questions à choix multiples, les réponses aux questions, etc.

### Évaluation des représentations :

- méthodes d'évaluation intrinsèques : similarités, analogies, détection d'intrus,
- propriétés statistiques,
- visualisation.



## READ ME

<https://gitlab.inria.fr/magnet/mangoes>

Documentation: <https://magnet.gitlabpages.inria.fr/mangoes/>

Référent : Joe Renner

\* Magnet est une équipe-projet commune à Inria et au laboratoire CRISTAL (Centrale Lille, CNRS, Université de Lille)







# Mangoes, geometry of words

*Inria*

## PROCESS

The Mangoes library provides tools for creating, manipulating, and evaluating vector representations of words (also called word embeddings).

These representations encode words of a vocabulary such that the words that are similar in meaning are expected to be closer in the vector space. They are learned in an unsupervised manner from raw texts.

Mangoes makes it easy to generate, evaluate and work with these representations, by implementing or interfacing different methods in a single library, from more traditional statistical embedding methods to the the widely-used BERT, which is a state of the art language modeling system which learns useful word and sentence embeddings from large corpora.

All the tools are largely configurable and modular:

- The input corpus can be plain or annotated text (e.g., in parts of speech),
- Many (hyper)parameters are available: text normalization, vocabulary selection, context sampling, reweighting of co-occurrences, contexts, etc.,
- Access to many pretrained BERT models and tokenizers,
- Use your own dataset to fine-tune BERT embeddings for various tasks such as text classification or multiple choice question answering.

## ADVANTAGES :

- usable on French corpora,
- usable on domain-specific corpora to get domain-specific representations,
- Modular and configurable,
- Easily extract BERT embeddings,
- Easy to fine tune BERT on downstream tasks.

## USE CASES

Word representations are a basic building block of any language processing tool: they can therefore be used for different tasks, ranging from document/sentence/token classification to finer semantic analyzes to question answering.

