

# Récapitulatif du cours *Introduction à l'IA*

7 décembre 2023

## 1 Apprentissage supervisé

- Principes
- Espace d'états
- Algorithme KNN
- Réseaux de neurones

## 2 Apprentissage par renforcement

- Principes
- Q-learning
- Apprentissage par renforcement profond

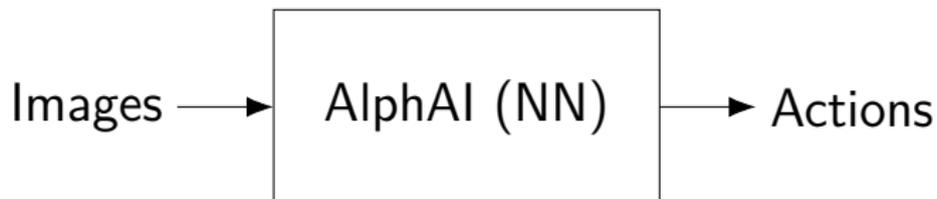
# Principes de l'apprentissage supervisé

## Modèle entraîné

Le résultat de l'apprentissage est un **modèle entraîné** qui fait de la **classification** de données :



Dans le cas du robot AlphaAI en mode autonome :

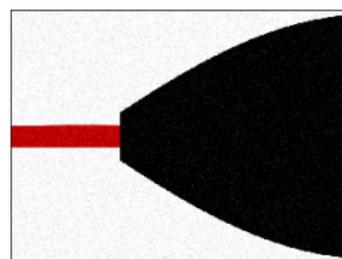
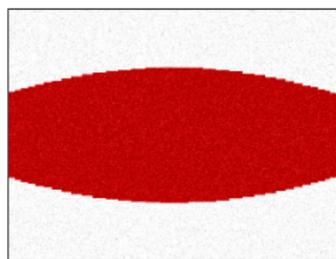
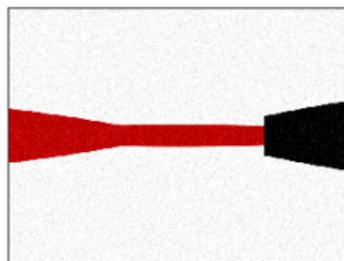


# Principes de l'apprentissage supervisé

## Données étiquetées

Pour entraîner un modèle, on a besoin de **données étiquetées** : il faut associer à chaque point de données la réponse qui est attendue.

Exemple : 3 points de données étiquetés pour la course de robots :



# Principes de l'apprentissage supervisé

## Phases de l'apprentissage

L'entraînement d'un modèle est généralement réalisé en 3 étapes :

- 1 **Phase d'entraînement** : on fournit au modèle des données étiquetées pour réaliser son entraînement. Exemple : piloter le robot AlphaAI.
- 2 **Phase de validation** : on fournit au modèle de **nouvelles** données étiquetées pour évaluer sa performance. Exemple : observation du robot en mode autonome.
- 3 **Phase d'utilisation** : le modèle traite des données non étiquetées, on s'intéresse au résultat de la classification. Exemple : course de robots autonomes.

# Principes de l'apprentissage supervisé

## Importance des données

La qualité du **modèle entraîné** dépend fortement de la qualité des **données d'entraînement**.

### 2 types de biais à éviter

**Biais explicites** : dûs à un mauvais étiquetage des données.

**Biais implicites** : les données ne sont pas assez exhaustives pour permettre la généralisation attendue.

### Solutions

**Biais explicites** : détecter et corriger les erreurs.

**Biais implicites** : ajouter des données plus variées.

# Principes de l'apprentissage supervisé

## Écueils de l'apprentissage

**sous-apprentissage** (*underfitting*) : le modèle n'est pas assez sophistiqué pour traiter les données.

**sur-apprentissage** (*overfitting*) : le modèle est trop complexe comparé à la quantité et la variété des données.



## 1 Apprentissage supervisé

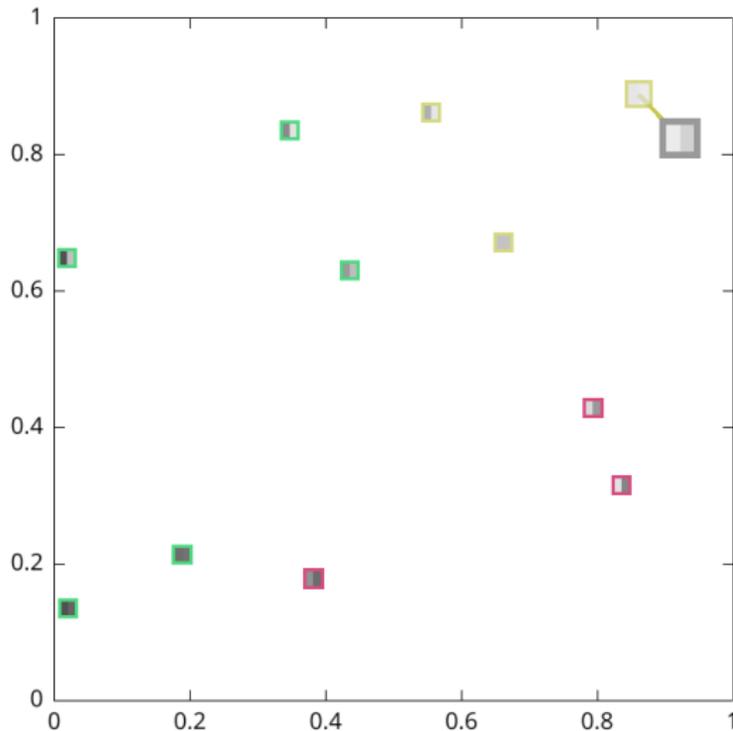
- Principes
- Espace d'états
- Algorithme KNN
- Réseaux de neurones

## 2 Apprentissage par renforcement

- Principes
- Q-learning
- Apprentissage par renforcement profond

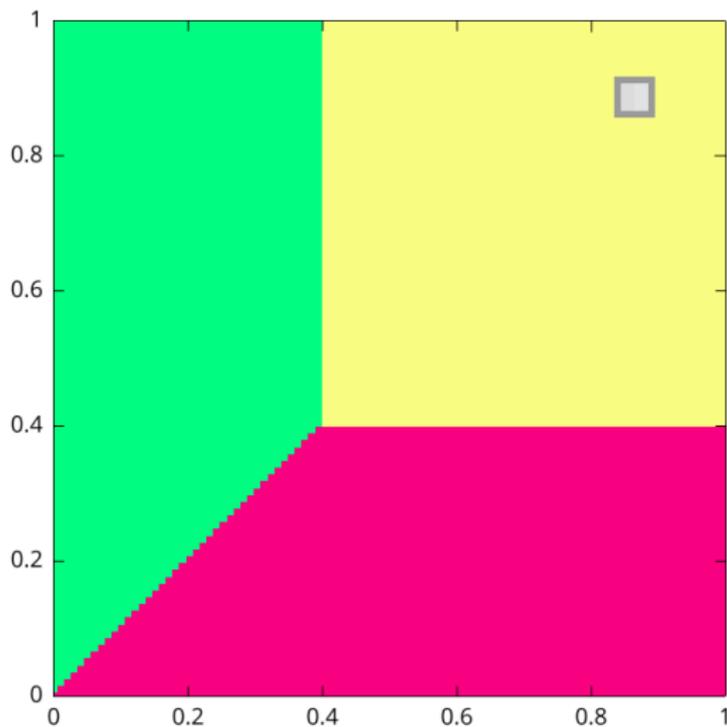
# Représentation de l'espace d'états

Les données étiquetées



# Représentation de l'espace d'états

L'algorithme de prise de décision



## 1 Apprentissage supervisé

- Principes
- Espace d'états
- Algorithme KNN
- Réseaux de neurones

## 2 Apprentissage par renforcement

- Principes
- Q-learning
- Apprentissage par renforcement profond

# Algorithme KNN

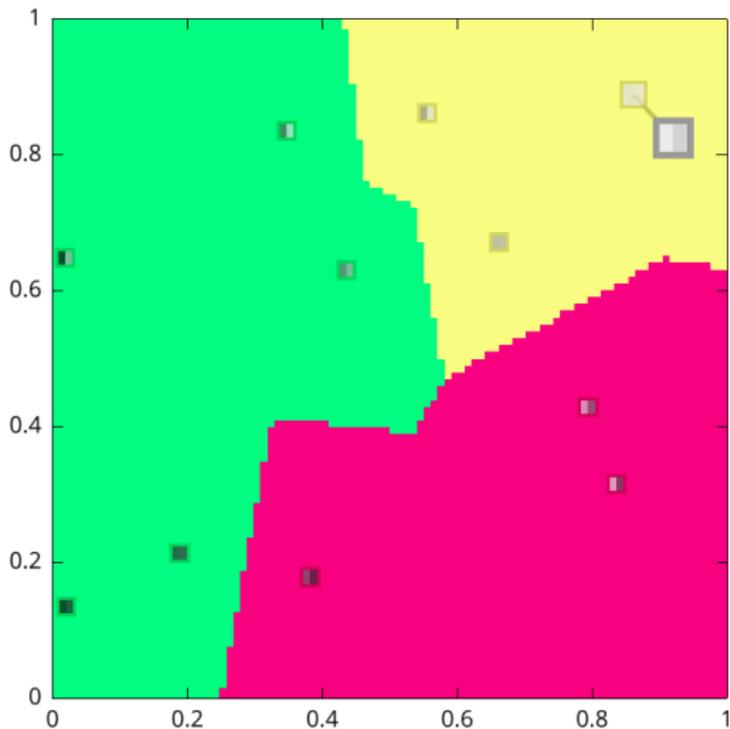
Un seul paramètre appelé  $k$  : le nombre de voisins.

## Prise de décision

- 1 Entrée : un état  $s$ .
- 2 Parmi les données mémorisées, on cherche les  $k$  points les plus proches de  $s$ .
- 3 On sélectionne l'étiquette  $a$  majoritaire parmi ces  $k$  points.
- 4 Sortie : la catégorie correspondant à l'étiquette  $a$ .

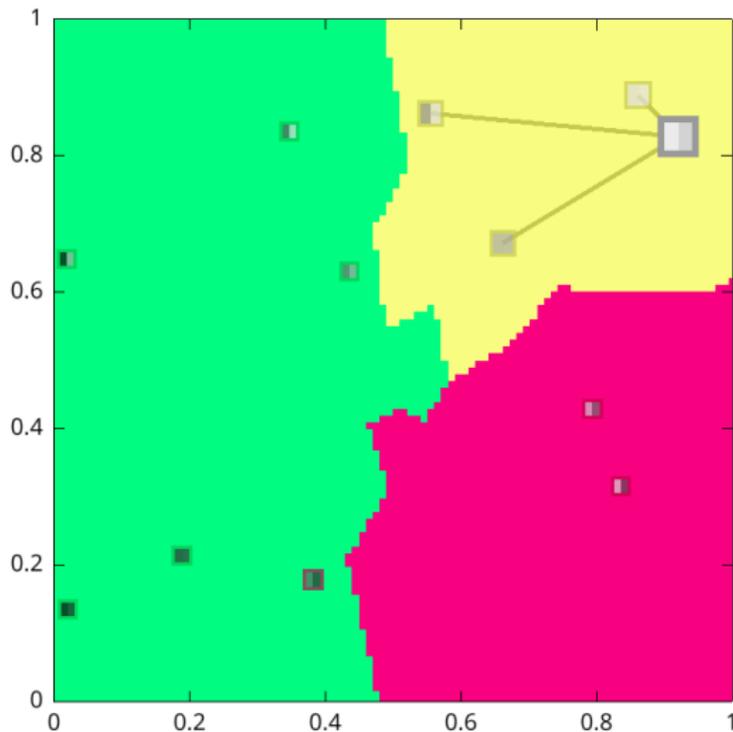
# Algorithme KNN

Résultat pour  $k = 1$



# Algorithme KNN

Résultat pour  $k = 3$



## 1 Apprentissage supervisé

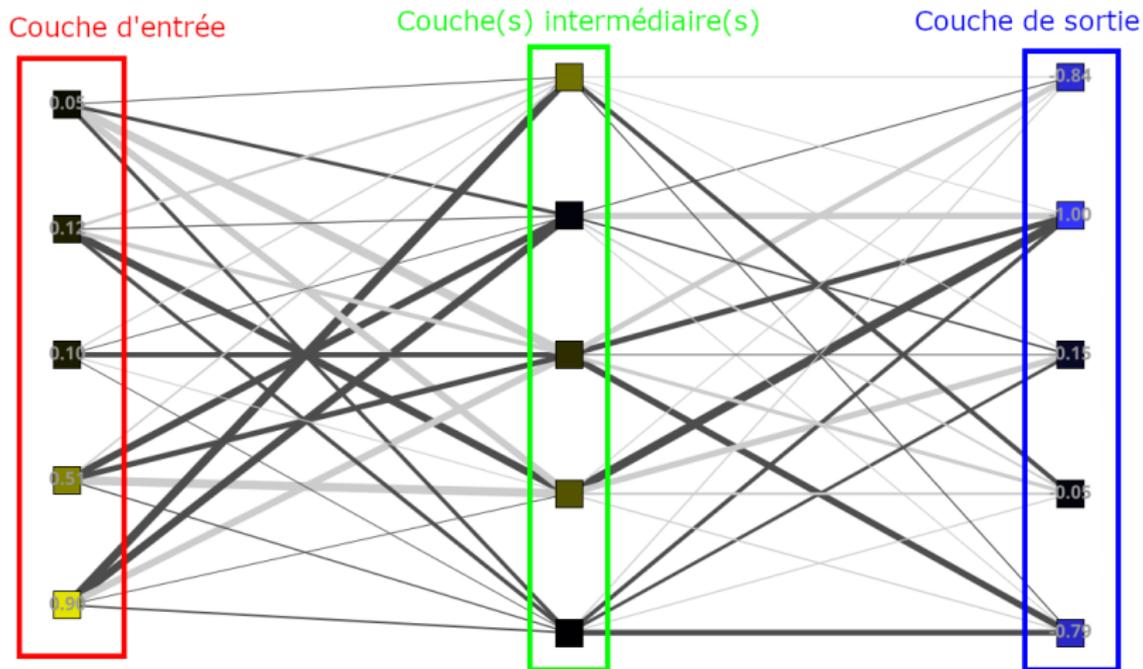
- Principes
- Espace d'états
- Algorithme KNN
- Réseaux de neurones

## 2 Apprentissage par renforcement

- Principes
- Q-learning
- Apprentissage par renforcement profond

# Les éléments d'un réseau de neurones

## Les couches de neurones

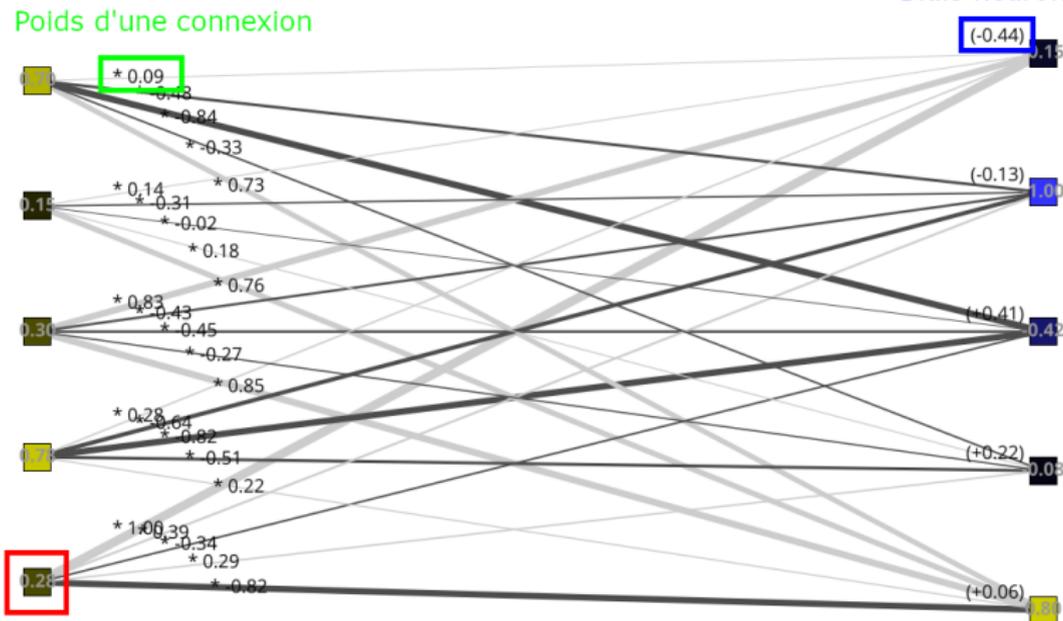


# Les éléments d'un réseau de neurones

## Niveau d'activation et paramètres

Poids d'une connexion

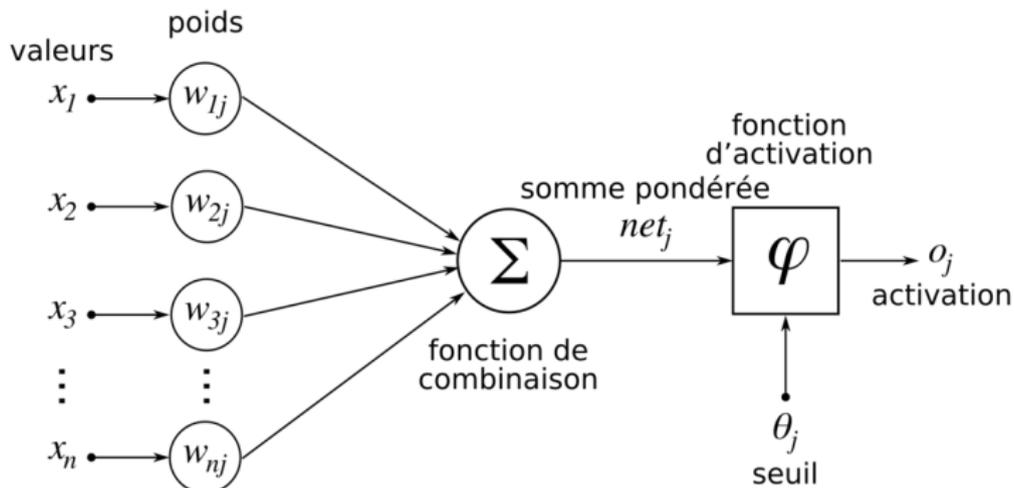
Biais neuronal



Neurone et son niveau d'activation

# Fonctionnement d'un réseau de neurones

## Calcul du niveau d'activation



$$y_j = f \left( b_j + \sum_{i=1}^n x_i w_{ij} \right)$$

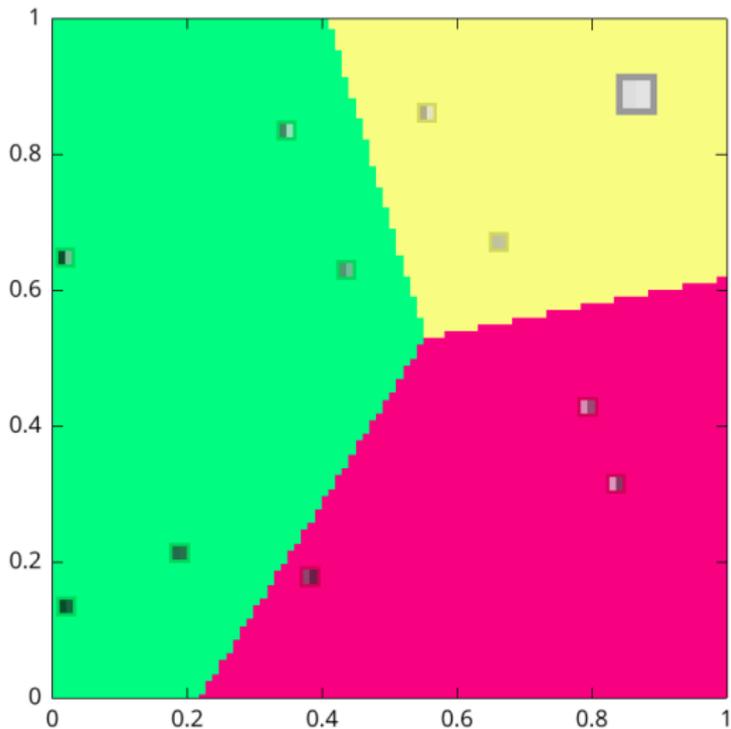
# Fonctionnement d'un réseau de neurones

## Entraînement

- Ce sont les valeurs des **poids des connexions** et des **biais des neurones** qui déterminent le comportement du réseau.
- L'entraînement d'un réseau de neurones consiste à **ajuster les poids et les biais** jusqu'à s'approcher autant que possible du résultat souhaité.
- Il s'agit d'un problème d'optimisation que l'on peut résoudre grâce à des algorithmes de **descente de gradient** (par exemple).

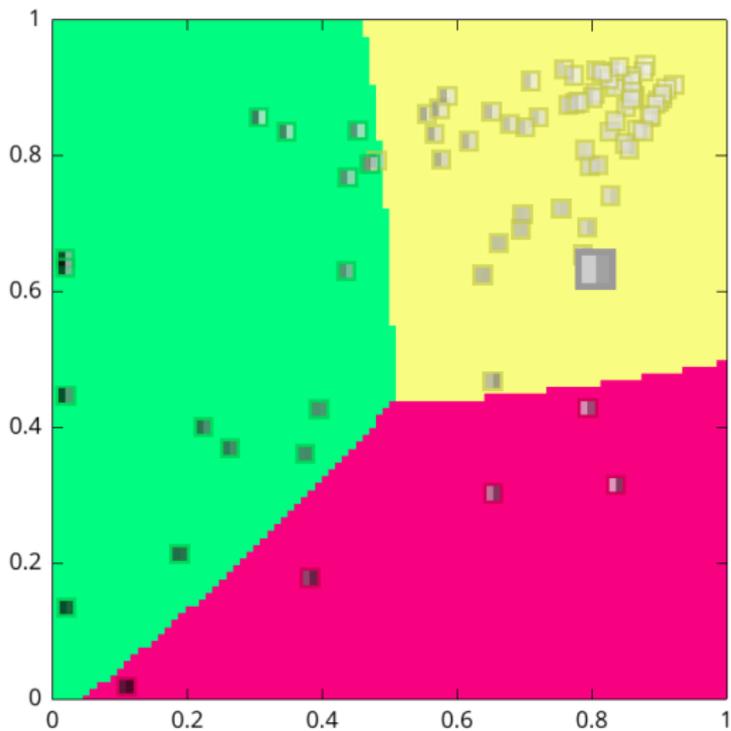
# Réseau de neurones

Résultat sans couche intermédiaire



# Réseau de neurones

Résultat avec couche intermédiaire



## 1 Apprentissage supervisé

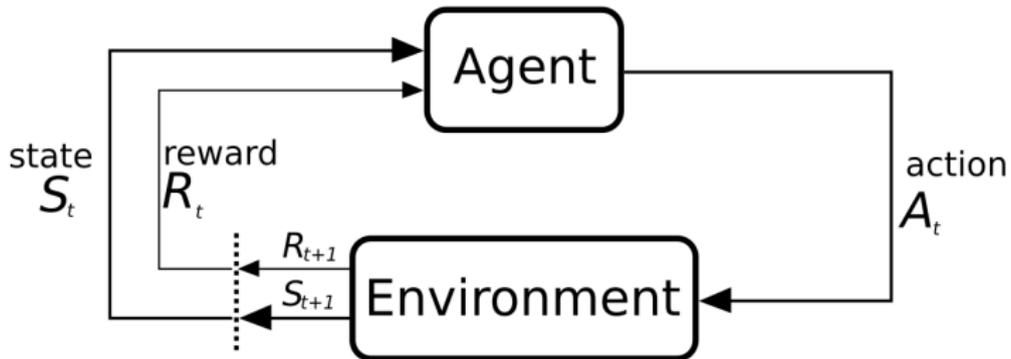
- Principes
- Espace d'états
- Algorithme KNN
- Réseaux de neurones

## 2 Apprentissage par renforcement

- Principes
- Q-learning
- Apprentissage par renforcement profond

# Apprentissage par renforcement

## Principe général



# Apprentissage par renforcement

## Différences avec l'apprentissage supervisé

Pas besoin de fournir des **données étiquetées** pour l'apprentissage. L'agent va générer ses propres données d'apprentissage. Pour cela, trois choses sont nécessaires :

- 1 Une **fonction de récompense** choisie avec soin en fonction de l'objectif recherché.
- 2 Une fonction de **prise de décision** incluant un certain degré d'**exploration** afin de ne pas passer à côté de nouvelles stratégies permettant d'améliorer la récompense.
- 3 L'algorithme d'apprentissage comporte un **aspect temporel** qui permet au modèle d'optimiser la moyenne des récompenses sur le **long terme**.

## 1 Apprentissage supervisé

- Principes
- Espace d'états
- Algorithme KNN
- Réseaux de neurones

## 2 Apprentissage par renforcement

- Principes
- Q-learning
- Apprentissage par renforcement profond

# Q-learning

## Q-table

Cet algorithme utilise un tableau de valeurs appelé *Q-table*. Le but de l'apprentissage est de faire converger ces valeurs vers une prédiction de la récompense moyenne associée à une action.

États	Actions		
	$a_1$	$\dots$	$a_m$
$s_1$	$Q(s_1, a_1)$	$\dots$	$Q(s_1, a_m)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$s_n$	$Q(s_n, a_1)$	$\dots$	$Q(s_n, a_m)$

# Q-learning

## Formule de mise à jour

$$\begin{cases} target &= (1 - \gamma)r + \gamma \max_a Q(s', a) \\ Q'(s, a) &= (1 - \alpha)Q(s, a) + \alpha target \end{cases}$$

- $s$  : état,  $a$  : action,  $r$  : récompense.
- $Q(s, a)$  : valeur de la cellule correspondant à l'état  $s$  et à l'action  $a$ .
- $\alpha$  : paramètre compris entre 0 et 1, appelé **vitesse d'apprentissage**.
- $\gamma$  : paramètre compris entre 0 et 1, appelé **facteur d'actualisation**.
- $target$  : une prédiction de la valeur moyenne de la récompense.

## 1 Apprentissage supervisé

- Principes
- Espace d'états
- Algorithme KNN
- Réseaux de neurones

## 2 Apprentissage par renforcement

- Principes
- Q-learning
- Apprentissage par renforcement profond

# Deep Q-learning

## Principe

- On utilise un réseau de neurones comme **modèle** à la place de la table de *Q-learning*.
- Les données générées par l'exploration sont utilisées pour entraîner le réseau de neurones à prédire la valeur de récompense associée à chaque prise de décision.