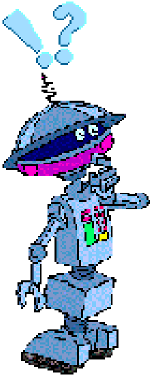


"*Deep Reinforcement Learning*" Sometimes it works, but more often it does not !!

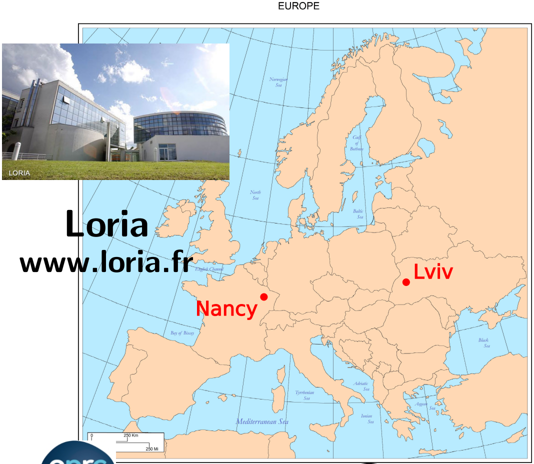
Alain DUTECH

INRIA - LORIA, Nancy

6th of July, 2019 - StatPhys 2019, Lviv

Loria
Laboratoire lorrain de recherche en informatique et ses applications

Loria Laboratory



Loria

www.loria.fr

Computer Sciences

- 28 research teams
- 204 researchers
  CNRS, Univ., INRIA
- 130 PhD students
- 35 post-doctorants
- 40 engineers
- 60+ admin/tech staff

Loria research topics

Organized in 5 departments

D1 **Algorithms, Computation, Images and Geometry**
ABC, ADAGIo, CARAMBA, MAGRIT, GAMBLE, PIXEL

D2 **Formal Methods**
CARTE, CARBONE, PESTO, DEDALE, MOSEL-VERIDIS, TYPES

D3 **Networks, Systems and Services**
COAST, MADYNES, OPTIMIST

D4 **Knowledge and Language Management**
CELLO, K, MULTISPEECH, ORPAILLEUR, READ, SMarT, SEMAGRAMME, SYNALP

D5 **Complex Systems, Artificial Intelligence and Robotic**
CAPSID, BISCUIT , KIWI, LARSEN, NEURORHYTHMS

Loria research topics

Organized in 5 departments

D1 **Algorithms, Computation, Images and Geometry**
ABC, ADAGIo, CARAMBA, MAGRIT, GAMBLE, PIXEL

D2 **Formal Methods**
CARTE, CARBONE, PESTO, DEDALE, MOSEL-VERIDIS, TYPES

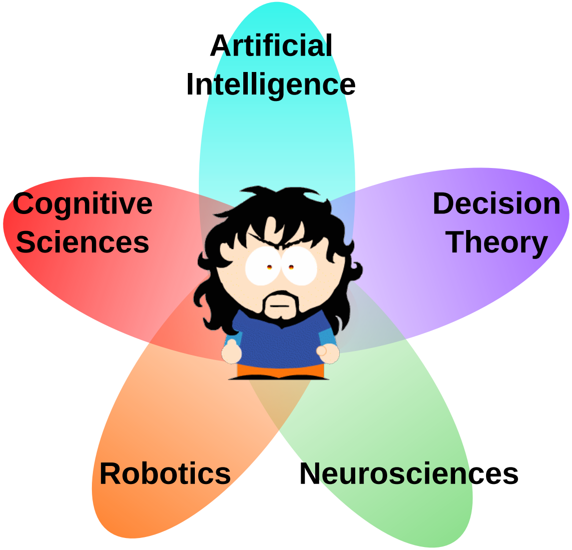
D3 **Networks, Systems and Services**
COAST, MADYNES, OPTIMIST

D4 **Knowledge and Language Management**
CELLO, K, MULTISPEECH, ORPAILLEUR, READ, SMarT, SEMAGRAMME, SYNALP

D5 **Complex Systems, Artificial Intelligence and Robotic**
CAPSID, BISCUIT , KIWI, LARSEN, NEURORHYTHMS

BISCUIT = Bio Inspired Situated Cellular Unconventionnal Information Technology.

Alain DUTECH

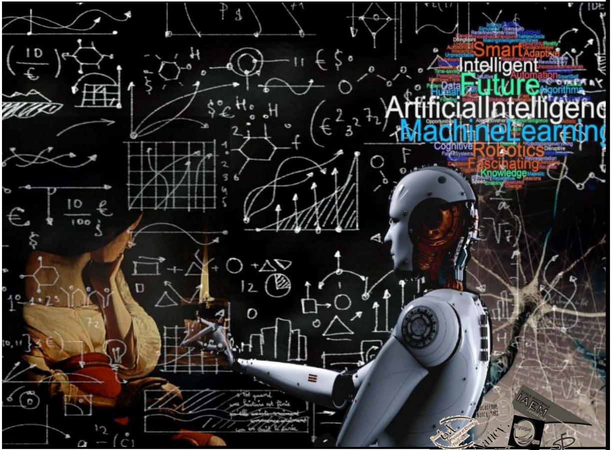


INRIA researcher at LORIA, Nancy, France...

One of my reason fore being here

5

Intelligences Artificielles



Colloque Cathy Dufour 2018

Site web : https://poincare.univ-lorraine.fr/fr/manifestations/colloque-cathy-dufour-2018

Nancy
Faculté des Sciences & Technologies
Amphi 5
Ouvert à tous

Jeudi 15 novembre

14h00
N. Fates : Intelligence artificielle : vers l'« ordination universelle » ?

14h40
M. Amblard : Calculer sur la langue mais qu'y comprendre ?

15h 15 - DISCUSSION

16h15
M. Rebuschi & M. Renauld : Interagir avec une machine ou faire-semblant ?

16h55
M. Clausel : Modélisation probabiliste et analyse de données textuelles : les approches de type topic modeling

17h35
A. Dutech : "Deep Reinforcement Learning" : des fois ça marche, souvent ça marche pas !

18h 10 - DISCUSSION

Vendredi 16 novembre

9h15
I. Marcovici : Automates cellulaires et phénomènes d'auto-organisation : le rôle de l'aléa

9h55
T. Boraud : Une histoire naturelle des aptitudes

11h00
F. Alexandre : L'Intelligence Artificielle apprend-elle de ses erreurs ?

11h 35 - DISCUSSION

Outline

Intro

Some context

RL

Reinforcement Learning (Q-Learning)

ANN

Artificial Neural Networks (+ Deep Learning)

DeepRL

Deep Reinforcement Learning

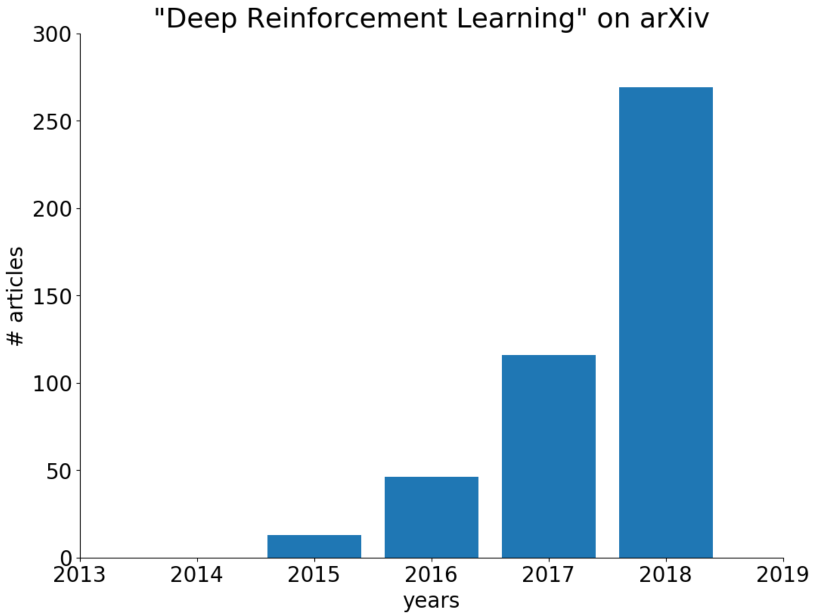
Conclusion

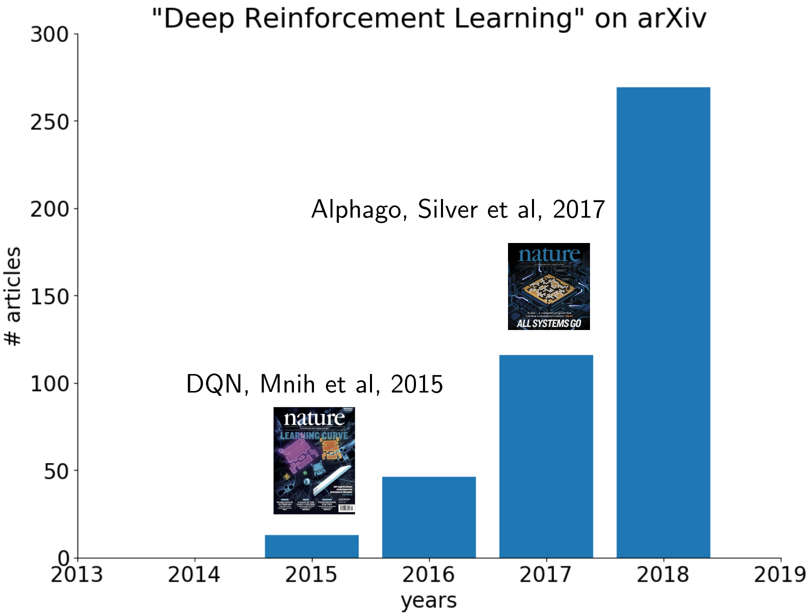
Can we really conclude anything ?

Outline


nature
THE INTERNATIONAL WEEKLY JOURNAL OF SCIENCE
At last — a computer program that can beat a champion Go player PAGE 484
ALL SYSTEMS GO

David Silver et al. (2017). "Mastering the game of Go without human knowledge". In: *Nature* 550.7676, p. 354


"Deep Reinforcement Learning" on arXiv
300
250
200
150
100
50
0
articles
2013
2014
2015
2016
2017
2018
2019
years


"Deep Reinforcement Learning" on arXiv
articles
0
50
100
150
200
250
300
years
2013
2014
2015
2016
2017
2018
2019
DQN, Mnih et al, 2015
nature
LEARNING CURVE
Alphago, Silver et al, 2017
nature
ALL SYSTEMS GO

Systems that act rationaly ?

"The exciting new effort to make computers think ... *machines with minds*, in the full and literal sense" (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning ..." (Bellman, 1978)	"The study of mental faculties through the use of computational models" (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act" (Winston, 1992)
"The art of creating machines that perform functions that require intelligence when performed by people" (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better" (Rich and Knight, 1991)	"A field of study that seeks to explain and emulate intelligent behavior in terms of computational processes" (Schalkoff, 1990) "The branch of computer science that is concerned with the automation of intelligent behavior" (Luger and Stubblefield, 1993)

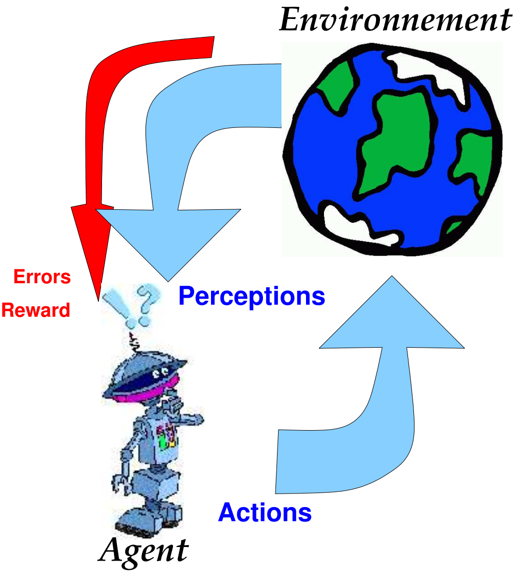
Figure 1.1 Some definitions of AI. They are organized into four categories:

Systems that think like humans.	Systems that think rationally.
Systems that act like humans.	Systems that act rationally.

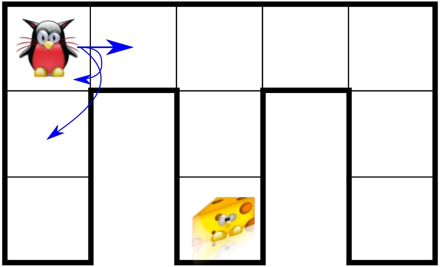
≪*Artificial Intelligence: A modern approach*≫, Russell and Norvig 1995

Outline

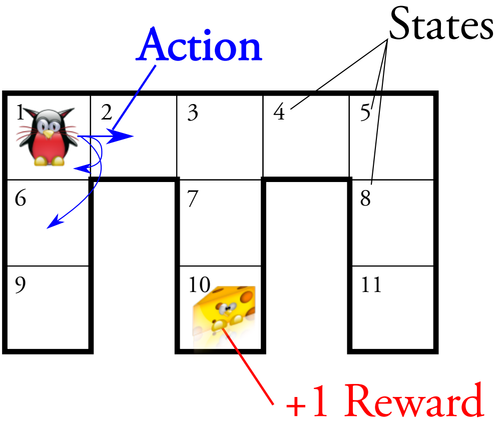
Model of the problem



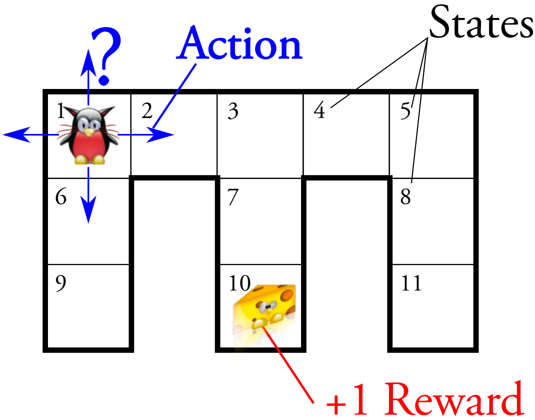
Example: find the cheese



Example: find the cheese

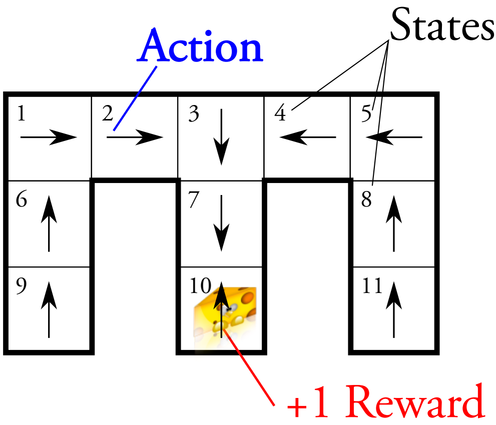


Example: find the cheese



Find a policy $\pi : \mathcal{S} \longrightarrow \mathcal{A}$ that maximizes $\mathbb{E}_{\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$

Example: find the cheese



Find a policy $\pi : \mathcal{S} \longrightarrow \mathcal{A}$ that maximizes $\mathbb{E}_{\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$

Markov Decision Process (Bellman 1957, Groupe PDMIA 2008)

Spaces

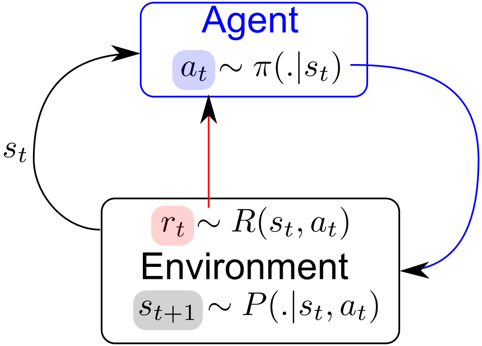
- $\mathcal{S}$: states
- $\mathcal{A}$: actions

Dynamics

- $P(s_{t+1}|s_t, a_t)$: transitions
- $R(s, a)$: reward

Agent

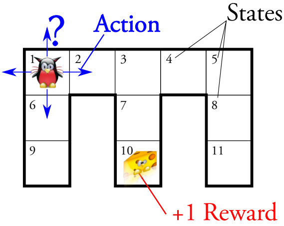
- $\pi(a_t|s_t)$: policy

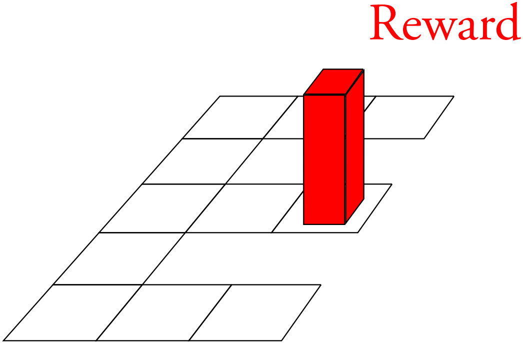


Criteria

Value Function: $V^\pi(s) = \mathbb{E}_{\sim\pi}\left[\sum_{t=1}^{T} \gamma^t r_t | s_0 = s\right]$, $\gamma \in [0, 1[$

Reward and Value Function

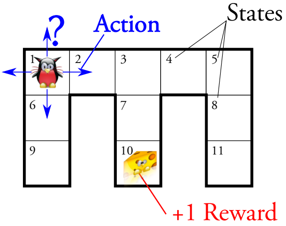


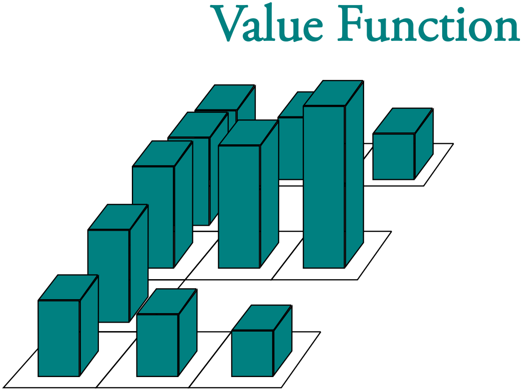


Criteria

Value Function: $V^{\pi}(s) = \mathbb{E}_{\sim\pi}\left[\sum_{t=1}^{T} \gamma^t r_t | s_0 = s\right], \qquad \gamma \in [0, 1[$

Reward and Value Function

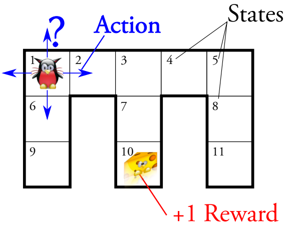




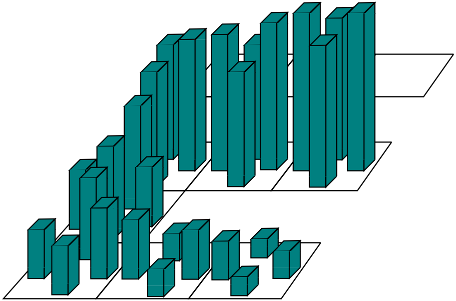
Criteria

Value Function: $V^{\pi}(s) = \mathbb{E}_{\sim\pi}\left[\sum_{t=1}^{T} \gamma^t r_t | s_0 = s\right], \qquad \gamma \in [0, 1[$

Reward and Value Function



Q-Value Function



Criteria

Q-Value: $Q^{\pi}(s,a) = \mathbb{E}_{\sim\pi}\left[\sum_{t=1}^{T}\gamma^t r_t \middle| s_0 = s, a_0 = a\right], \qquad \gamma \in [0,1[$

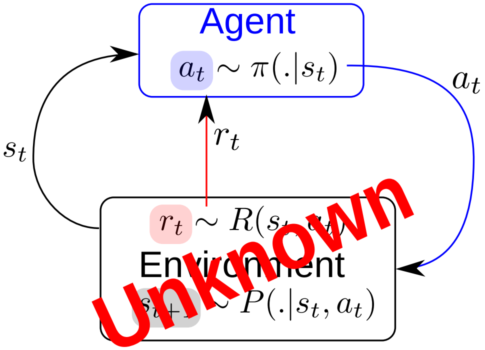
Learn Q, even without *Model/Dynamics*

Spaces

- $\mathcal{S}$: states
- $\mathcal{A}$: actions

Dynamics

- $\cancel{P(s_{t+1}|s_t, a_t)}$: transitions
- $\cancel{R(s, a)}$: reward



Agent

- $\pi(a_t|s_t)$: policy

Critère

Value Function: $V^\pi(s) = \mathbb{E}_{\sim\pi}\left[\sum_{t=1}^{T} \gamma^t r_t | s_0 = s\right]$, $\gamma \in [0, 1[$

Q-Learning

Optimal Q-Function Q^*

- By definition: $Q^*(s,a) = \max_{\pi in \Pi} \{\mathbb{E}_{\sim\pi}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s, a_0 = a]\}$
- Property: $Q^*(s,a) = \mathbb{E}_{\sim\pi}[R(s,a) + \gamma \max_{a' \in \mathcal{A}} Q^*(s', a')]$

Q-Learning (Watkins 1989)

1. Define a exploration policy π, Init $Q(s,a), \forall s, a$
2. Repeat until "convergence"
   2.1 In s_t, apply $\pi \rightsquigarrow (s_t, a_t, r_t, s_{t+1})$
   2.2 Update
   $Q(s_t, a_t) \longleftarrow Q(s_t, a_t) + \alpha_t [r_t + \gamma \max_{a' \in \mathcal{A}} Q(s_{t+1}, a') - Q(s_t, a_t)]$
3. Optimal Policy: $\pi^*(s) \longleftarrow \mathrm{argmax}_{a \in \mathcal{A}} Q(s,a)$, for all $s \in \mathcal{S}$

Sufficient Conditions for convergence

- Every (s,a) explored infinitely often
- $\sum \alpha_t = \infty$, $\sum (\alpha_t)^2 < \infty$

Reinforcement Learning

The framework of Markov Decision Processes ensure that:

Repeat until "convergence"

1. chose an action in state ($\rightsquigarrow r_t, s_{t+1}$)
2. update Q-valeur from previous state according to reward
   (avec $\Delta Q \approx [r_t + \gamma \max_{a' \in \mathcal{A}} Q(s_{t+1}, a') - Q(s_t, a_t)]$)

will lead to the optimal policy.

Reinforcement Learning

The framework of Markov Decision Processes ensure that:

Repeat until "convergence"

1. chose an action in state ($\rightsquigarrow r_t, s_{t+1}$)
2. update Q-valeur from previous state according to reward
   (avec $\Delta Q \approx [r_t + \gamma \max_{a' \in \mathcal{A}} Q(s_{t+1}, a') - Q(s_t, a_t)]$)

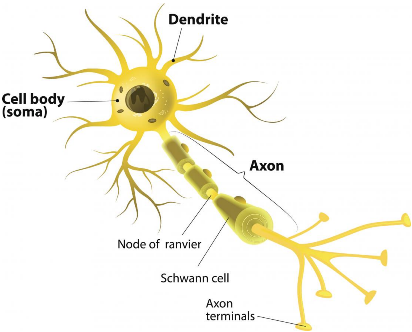
will lead to the optimal policy.

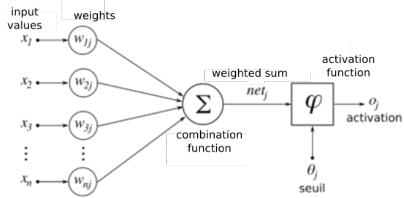
Problem

How can we represent/memorize this Q **function** when $\mathcal{S}$ is a continuous (or very large) space ?

Outline

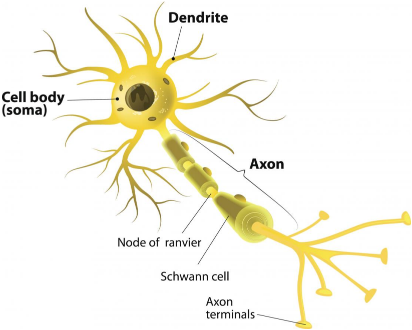
Formal Neuron

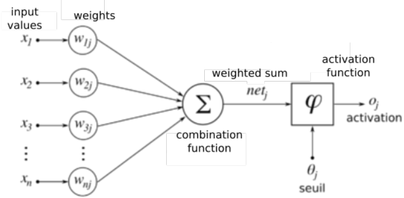




$$o_j = \varphi(x_1.w_{1j} + x_2.w_{2j} + \cdots + x_n.w_{nj} - \theta_j)$$

Formal Neuron





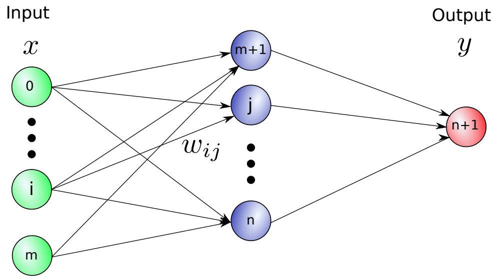
$$o_j = \varphi(x_1.w_{1j} + x_2.w_{2j} + \cdots + x_n.w_{nj} - \theta_j)$$

Universal Approximator

Theorem Univeral Approximation

Formal neural networks with at least 3 layers are **universal approximators** under rather weak hypothesis on the activation functions (non-polynomial).

Cybenko 1989; Hornik 1993; Scarselli and Tsoi 1998



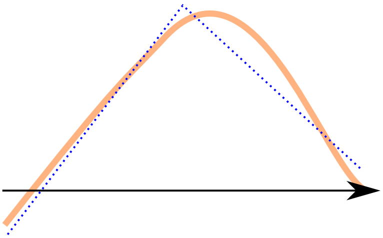
Supervised Learning

Gradient Descent using Back-Propagation

Examples with **labels** : $\left\{x^i, t^i = f(x^i)\right\}_{i \in 1,\ldots,N}$
Minimize error : $E = \frac{1}{2}\sum_N (y^i - t^i)^2$

Repeat

1. Example $x_i \overset{NN}{\rightsquigarrow} y_i$
2. Gradient error $\dfrac{\partial E}{\partial w_{ij}}$
3. Update $\Delta w_{ij} = -\alpha \times \dfrac{\partial E}{\partial w_{ij}}$



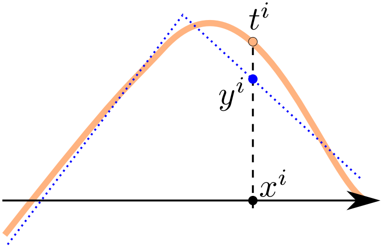
Supervised Learning

Gradient Descent using Back-Propagation

Examples with **labels** : $\left\{x^i, t^i = f(x^i)\right\}_{i \in 1,\ldots,N}$
Minimize error : $E = \frac{1}{2}\sum_N (y^i - t^i)^2$

Repeat

1. Example $x_i \overset{NN}{\rightsquigarrow} y_i$
2. Gradient error $\dfrac{\partial E}{\partial w_{ij}}$
3. Update
   $\Delta w_{ij} = -\alpha \times \dfrac{\partial E}{\partial w_{ij}}$



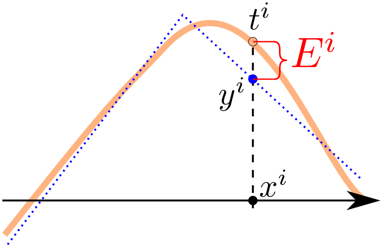
Supervised Learning

Gradient Descent using Back-Propagation

Examples with **labels** : $\left\{x^i, t^i = f(x^i)\right\}_{i \in 1,\ldots,N}$
Minimize error : $E = \frac{1}{2}\sum_N (y^i - t^i)^2$

Repeat

1. Example $x_i \overset{NN}{\rightsquigarrow} y_i$
2. Gradient error $\dfrac{\partial E}{\partial w_{ij}}$
3. Update
   $\Delta w_{ij} = -\alpha \times \dfrac{\partial E}{\partial w_{ij}}$



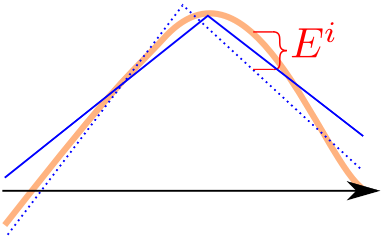
Supervised Learning

Gradient Descent using Back-Propagation

Examples with **labels** : $\{x^i, t^i = f(x^i)\}_{i \in 1,\ldots,N}$
Minimize error : $E = \frac{1}{2}\sum_N (y^i - t^i)^2$

Repeat

1. Example $x_i \overset{NN}{\rightsquigarrow} y_i$
2. Gradient error $\dfrac{\partial E}{\partial w_{ij}}$
3. Update
   $$\Delta w_{ij} = -\alpha \times \frac{\partial E}{\partial w_{ij}}$$



Supervised Learning

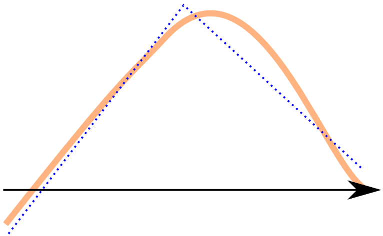
Gradient Descent using Back-Propagation

Examples with **labels** : $\left\{x^i, t^i = f(x^i)\right\}_{i\in 1,\dots,N}$

Minimize error : $E = \frac{1}{2}\sum_N (y^i - t^i)^2$

Repeat

1. Example $x_i \overset{NN}{\rightsquigarrow} y_i$
2. Gradient error $\dfrac{\partial E}{\partial w_{ij}}$
3. Update $\Delta w_{ij} = -\alpha \times \dfrac{\partial E}{\partial w_{ij}}$



Convolution Network

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

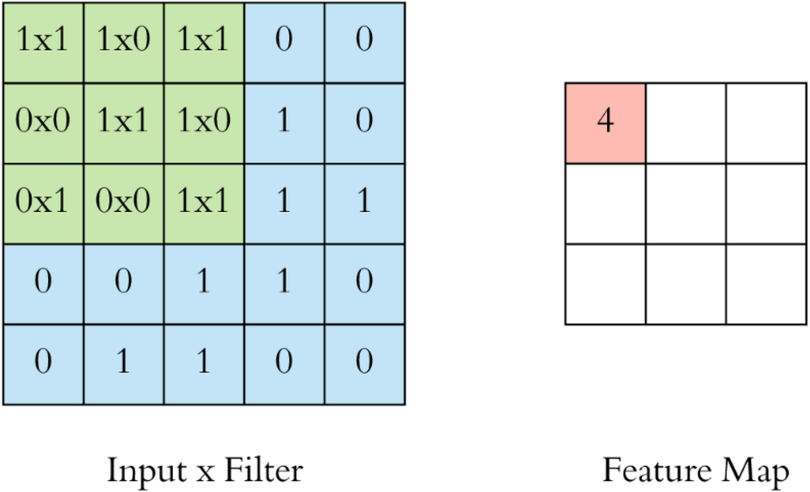
Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

Source : https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Convolution Network



Source : `https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2`

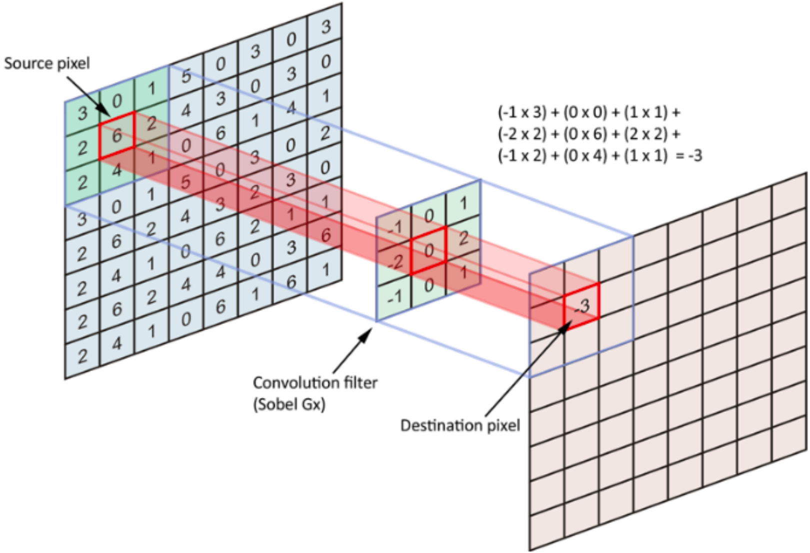
Convolution Network

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

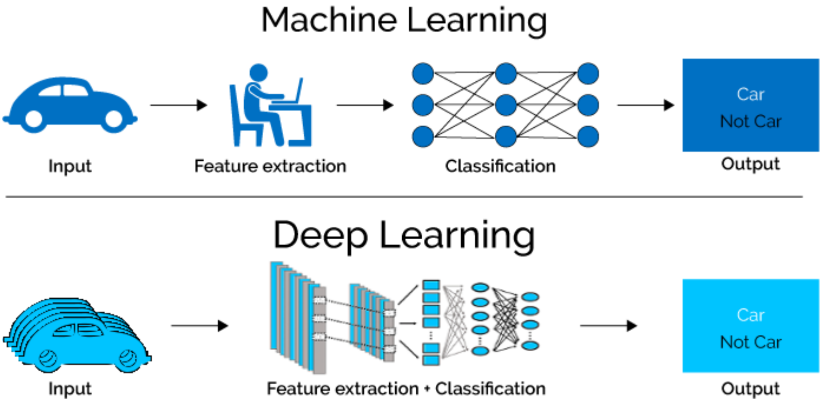
Source : `https://towardsdatascience.com/`

Convolution Network



Source : `https://towardsdatascience.com/`

Automatic Feature Extraction



Sources:
- https://medium.com/swlh/ill-tell-you-why-deep-learning-is-so-popular-and-in-demand-5aca72628780
- Hou, Adhikari, and Cheng 2018

Connexionism - Deep Learning

Reminder

Multi-layer regressor and **convolution** formal networks are only a **small** subpart of artificial neural networks.

- neural network with at least 3 layers can learn any function
- convolution networks: extract features
- deep learning: combine previous points
- no constructive theorem/algorithm but learning algorithm quite efficient
- need huge datasets

Connexionism - Deep Learning

Reminder

Multi-layer regressor and **convolution** formal networks are only a **small** subpart of artificial neural networks.

- neural network with at least 3 layers can learn any function
- convolution networks: extract features
- deep learning: combine previous points
- no constructive theorem/algorithm but learning algorithm quite efficient
- need huge datasets

Deep Reinforcement Learning

Represent the Q-function with a (deep) neural network

Outline

Intro

Some context

RL

Reinforcement Learning (Q-Learning)

ANN

Artificial Neural Networks (+ Deep Learning)

DeepRL

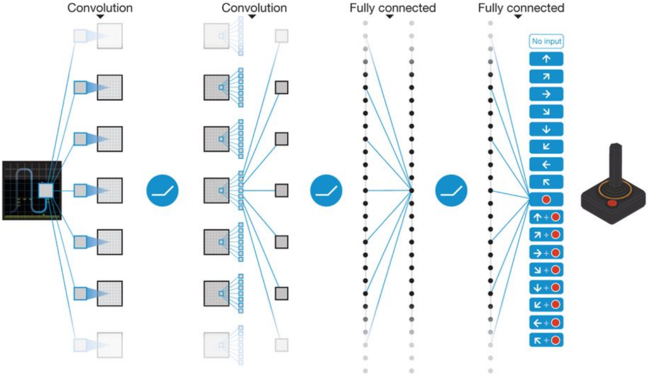
Deep Reinforcement Learning

Conclusion

Can we really conclude anything ?

Deep Reinforcement Learning

"breakthrough" : DQN (Deep Q-Network) Mnih et al. 2015

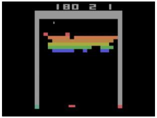


Deep Reinforcement Learning

Represent the Q-function with a (deep) neural network

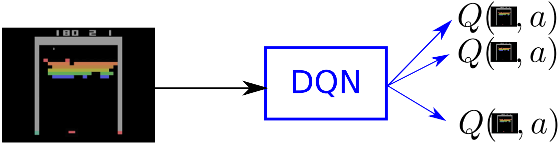
Deep Reinforcement Learning

"breakthrough" : DQN (Deep Q-Network) Mnih et al. 2015



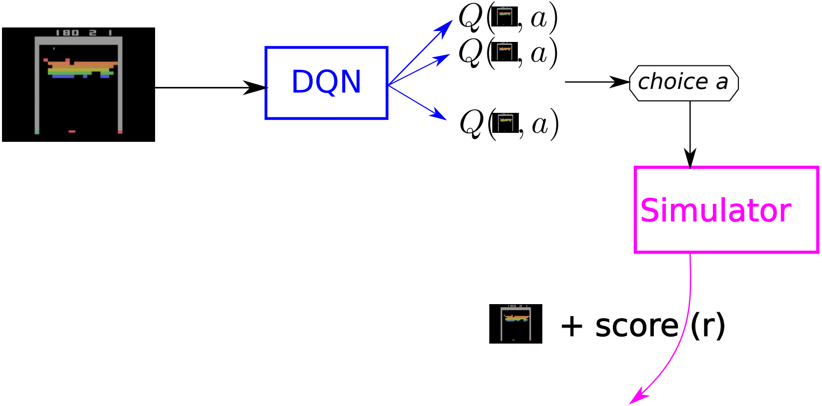
Deep Reinforcement Learning

"breakthrough" : DQN (Deep Q-Network) Mnih et al. 2015



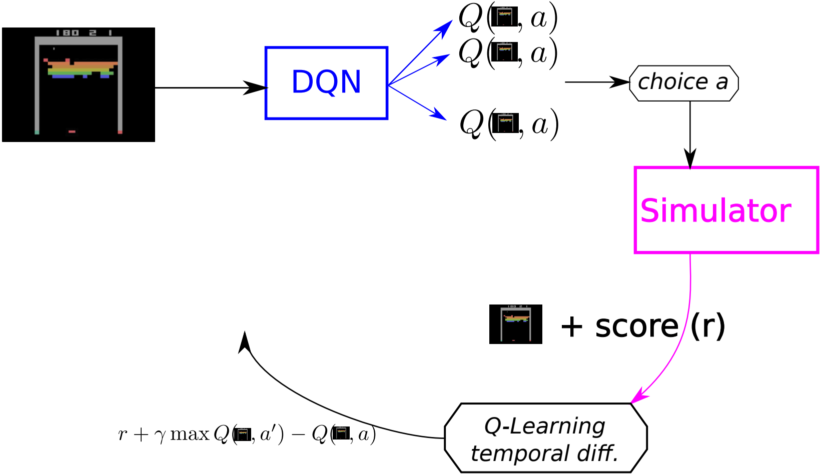
Deep Reinforcement Learning

"breakthrough" : DQN (Deep Q-Network) Mnih et al. 2015



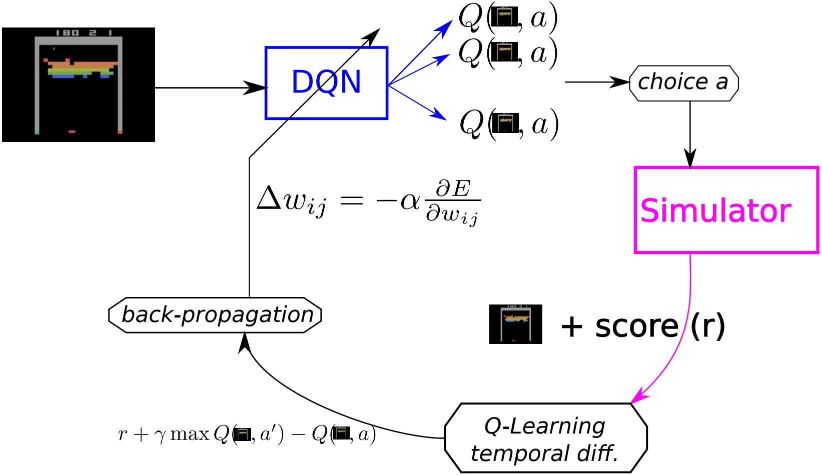
Deep Reinforcement Learning

"breakthrough" : DQN (Deep Q-Network) Mnih et al. 2015



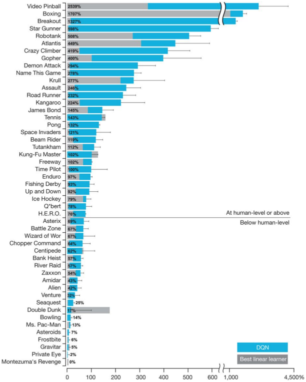
Deep Reinforcement Learning

"breakthrough" : DQN (Deep Q-Network) Mnih et al. 2015



Deep Reinforcement Learning

"breakthrough" : DQN (Deep Q-Network) Mnih et al. 2015



Learn autoamticaly to "play"

- State: 4 x images (84x84)
- Actions : joystick
- Reward : according to score (??) $([-1, 1])$
- 49 games
- number of iterations : **a lot** (70 million img??)

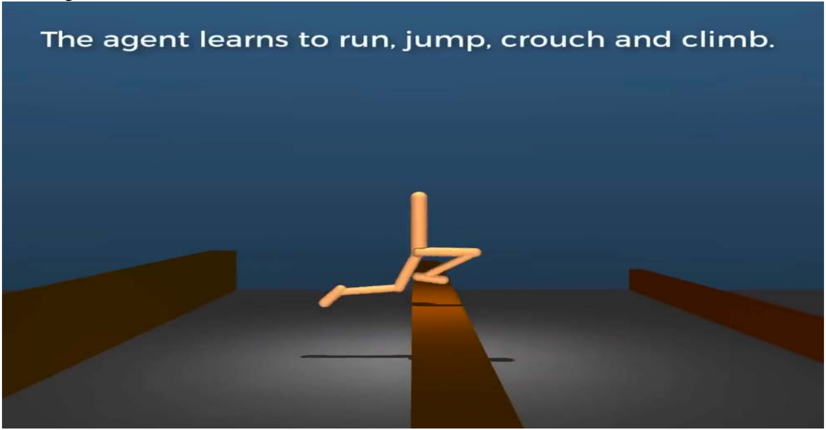
What about theory ?

not much can be **proved** nor **ensured** as we need:

- **Markovian** problem
- **infinite** number of trials
- approximation of Q function should be **linear**
- ANN can learn any function (but **what structure ?**)
- Backpropagation $\rightsquigarrow$ **local** optimum

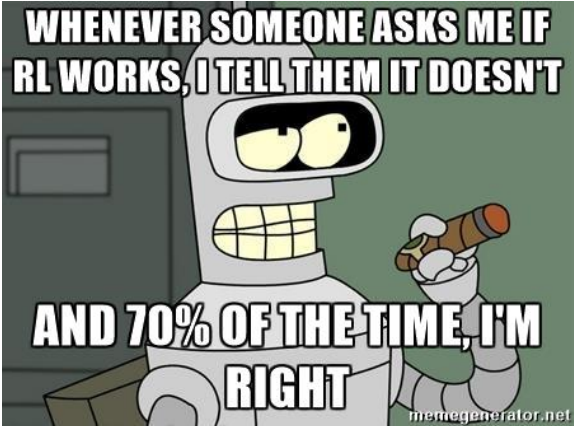
Sometimes it works

Emergence of Locomotion Behaviours in Rich Environments.mov



And also DQN, AlphaGo, AlphaZero, reduce energy consumption in large datacenters, AutoML, Dota 2, ...

… but more often it does not !!



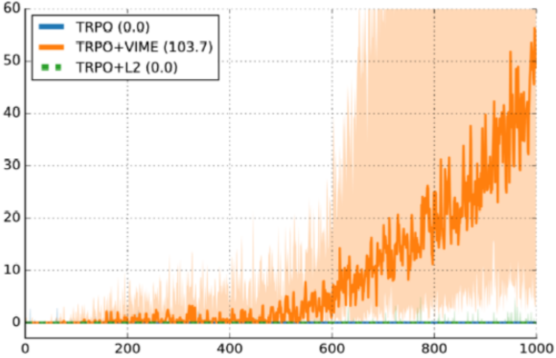
Irpan 2018

... but more often it does not !!

- **Not** data **efficient**
- (Often, not the best performance reached)
- Defining the reward is a very delicate task
- Local optima
- Generalization is a hard problem (vs **over-specialization**)
- Very unstable, many hyper-parameters, **very hard to reproduce**

Irpan 2018

… but more often it does not !!



Irpan 2018

What can help you ? (if you want to try)

- Easy to generate zillions of examples
  - Able to “self-play” or againt yourself.
- Exist simplified expression of the problem
- Clear and easy way to define the rewards
- Reward function can be shaped to give information very often
- (Already know good features to use)

Irpan 2018

Outline

Any "take home" message ?

Deep RL (Artificial Intelligence)

- can sometimes lead to spectacular (technical) achievements
- relies on "ancient" (grounded) knowledge
  (MDP, backpropagation, CNN)
- it looks like simple ideas but with solid theoretical grounding
- but theory is very limited: non-realistic conditions

- Sometimes, motivates and inspires real scientific progress

Any "take home" message ?

Deep RL (Artificial Intelligence)

- can sometimes lead to spectacular (technical) achievements
- relies on "ancient" (grounded) knowledge
  (MDP, backpropagation, CNN)
- it looks like simple ideas but with solid theoretical grounding
- but theory is very limited: non-realistic conditions

- Sometimes, motivates and inspires real scientific progress In Machine Learning (**vanishing gradient, exploration, goal generation, state representation, unsupervised learning, data efficience, ...**)

Any "take home" message ?

Deep RL (Artificial Intelligence)

- can sometimes lead to spectacular (technical) achievements
- relies on "ancient" (grounded) knowledge
  (MDP, backpropagation, CNN)
- it looks like simple ideas but with solid theoretical grounding
- but theory is very limited: non-realistic conditions

- Sometimes, motivates and inspires real scientific progress but also in other fields because of "**Ready to use toolkit**". Statistical physics ?

Références I

Bellman, R. (1957). *Dynamic programming.* Princeton University Press, Princeton, New-Jersey.

Cybenko, George (1989). "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems* 2.4, pp. 303–314.

Groupe PDMIA (2008). *Processus Décisionnels de Markov en Intelligence Artificielle. (Edité par Olivier Buffet et Olivier Sigaud)*. Vol. 1 & 2. Lavoisier - Hermes Science Publications.

Hornik, Kurt (1993). "Some new results on neural network approximation". In: *Neural networks* 6.8, pp. 1069–1072.

Hou, Jie, Badri Adhikari, and Jianlin Cheng (2018). "DeepSF: deep convolutional neural network for mapping protein sequences to folds". In: *Bioinformatics* 34.8, pp. 1295–1303.

Références II

Irpan, Alex (2018). *Deep Reinforcement Learning Doesn't Work Yet*. https://www.alexirpan.com/2018/02/14/rl-hard.html.

Mnih, Volodymyr et al. (2015). "Human-level control through deep reinforcement learning". In: *Nature* 518.7540, p. 529.

Russell, S. and P. Norvig (1995). *Artificial Intelligence: A modern approach*. Prentice Hall.

Scarselli, Franco and Ah Chung Tsoi (1998). "Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results". In: *Neural networks* 11.1, pp. 15–37.

Silver, David et al. (2017). "Mastering the game of Go without human knowledge". In: *Nature* 550.7676, p. 354.

Watkins, C. (1989). "Learning from delayed rewards.". PhD thesis. King's College of Cambridge, UK.