Reading Group : Episodic Exploration for Deep Deterministic Policies

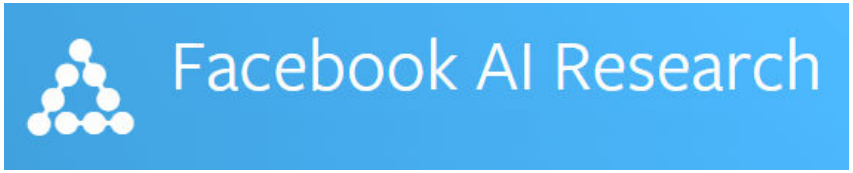
Florian Richoux

Nantes Machine Learning Meetup
12 juin 2017

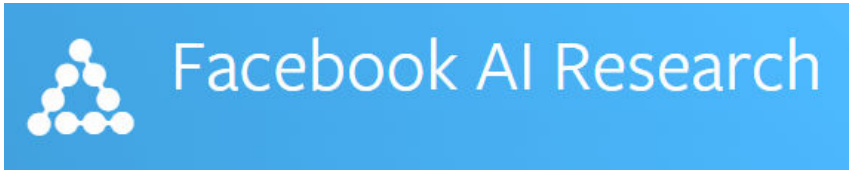
Outline

Sommaire

À propos de ce papier



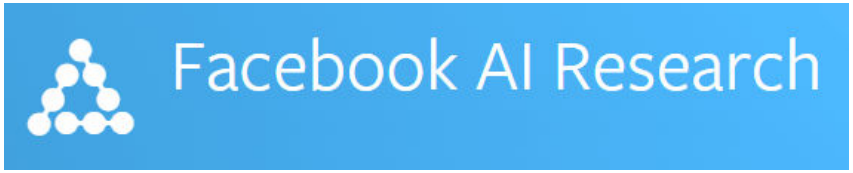
À propos de ce papier



Ce qu'est ce papier

- Le premier papier sur du deep learning avec StarCraft.

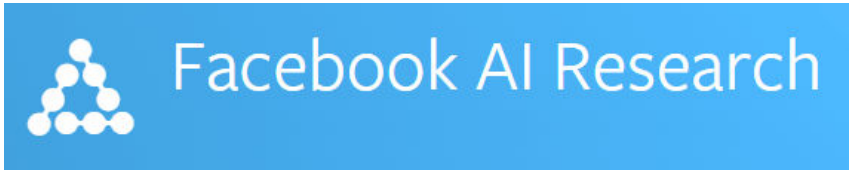
À propos de ce papier



Ce qu'est ce papier

- Le premier papier sur du deep learning avec StarCraft.
- Une proposition de scénarii de micro-gestion.

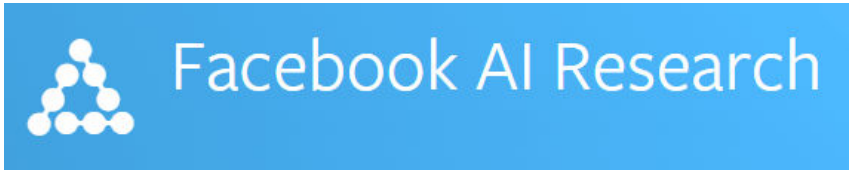
À propos de ce papier



Ce qu'est ce papier

- Le premier papier sur du deep learning avec StarCraft.
- Une proposition de scénarii de micro-gestion.
- Un nouvel algo *gradient-free* d'apprentissage par renforcement.

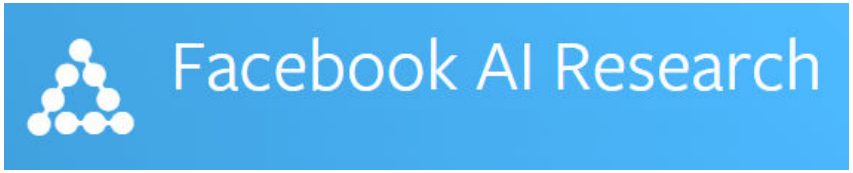
À propos de ce papier



Ce qu'est ce papier

- Le premier papier sur du deep learning avec StarCraft.
- Une proposition de scénarii de micro-gestion.
- Un nouvel algo *gradient-free* d'apprentissage par renforcement.
- Beaucoup de sueur (TorchCraft arxiv.org/abs/1611.00625).

À propos de ce papier



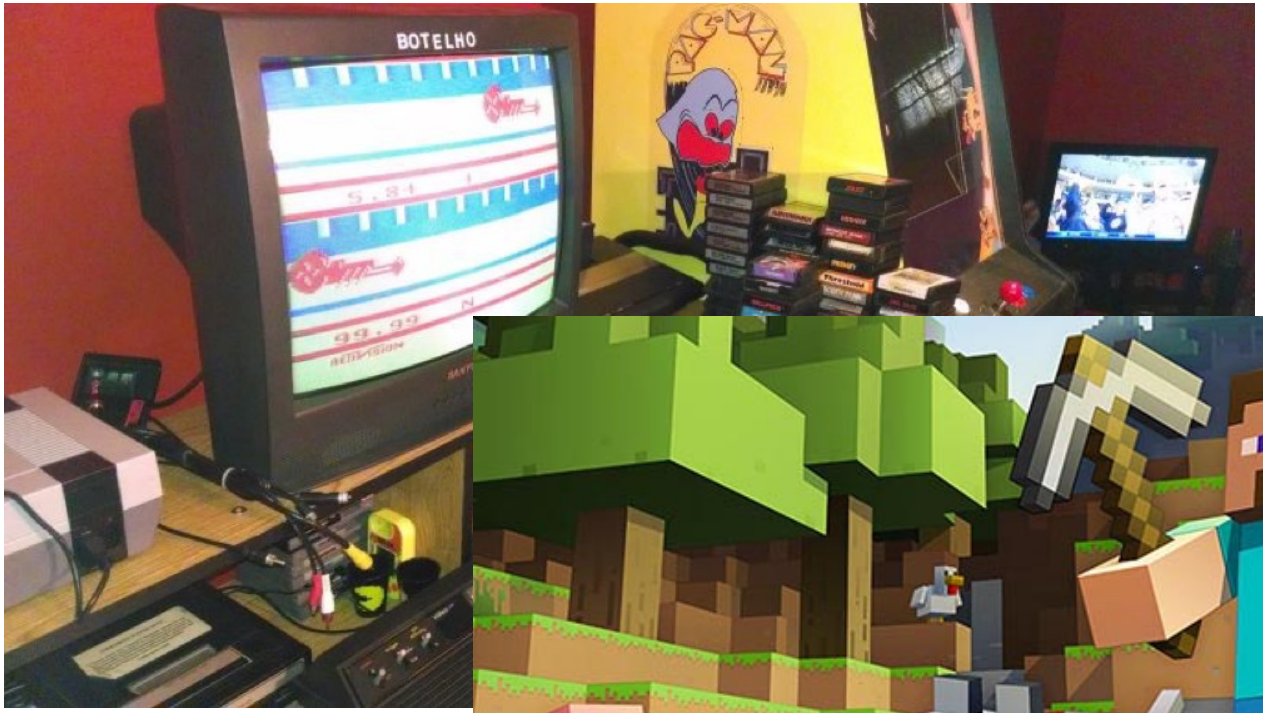
Ce qu'est ce papier

- Le premier papier sur du deep learning avec StarCraft.
- Une proposition de scénarii de micro-gestion.
- Un nouvel algo *gradient-free* d'apprentissage par renforcement.
- Beaucoup de sueur (TorchCraft arxiv.org/abs/1611.00625).

Ce que n'est pas ce papier

- Un début d'IA jouant à StarCraft basé sur du deep learning (holistique).

Challenges de la micro-gestion dans StarCraft



Atari

Minecraft

Challenges de la micro-gestion dans StarCraft

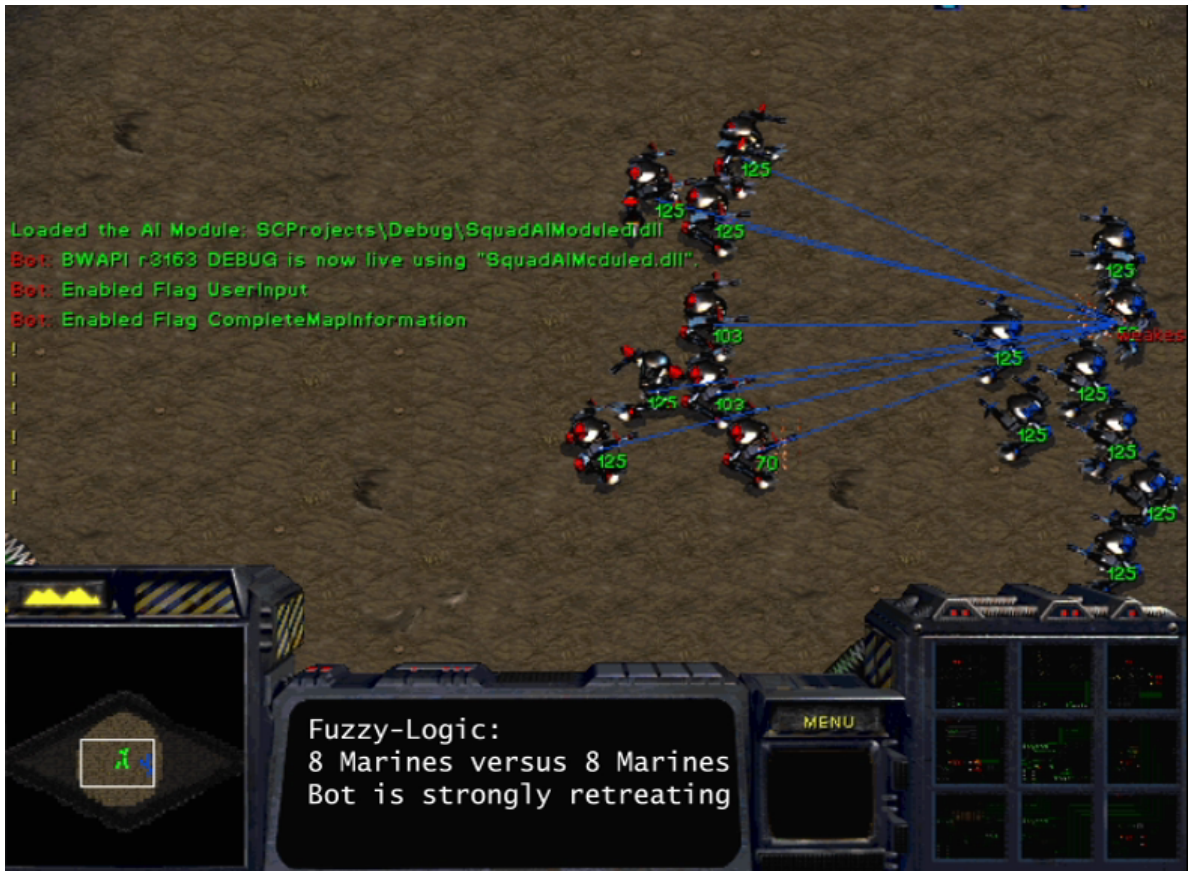
Contrôle de plusieurs agents en même temps

- Actions duratives et interdépendantes.
- Évaluation plus chaotique des stratégies.

Challenges de la micro-gestion dans StarCraft

Explorer en tirant au hasard une action casse tout équilibre

- Désorganisation des agents.
- Défaite assuré dont aucune leçon ne peut être tirée.

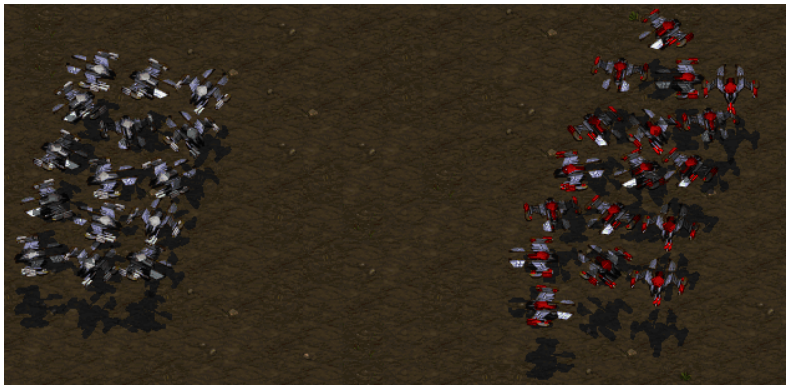


Sommaire

Scénarii de micro-gestion

5 marines vs 5 (m5v5)

15 marines vs 16 (m15v16)

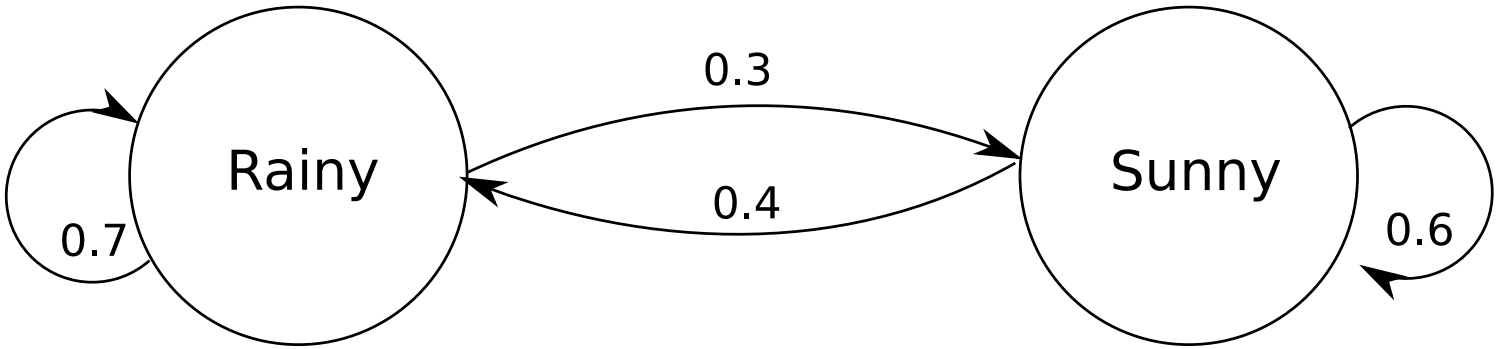


15 wraith vs 17 (w15v17)

mix 3 zealots + 2 dragoons

Sommaire

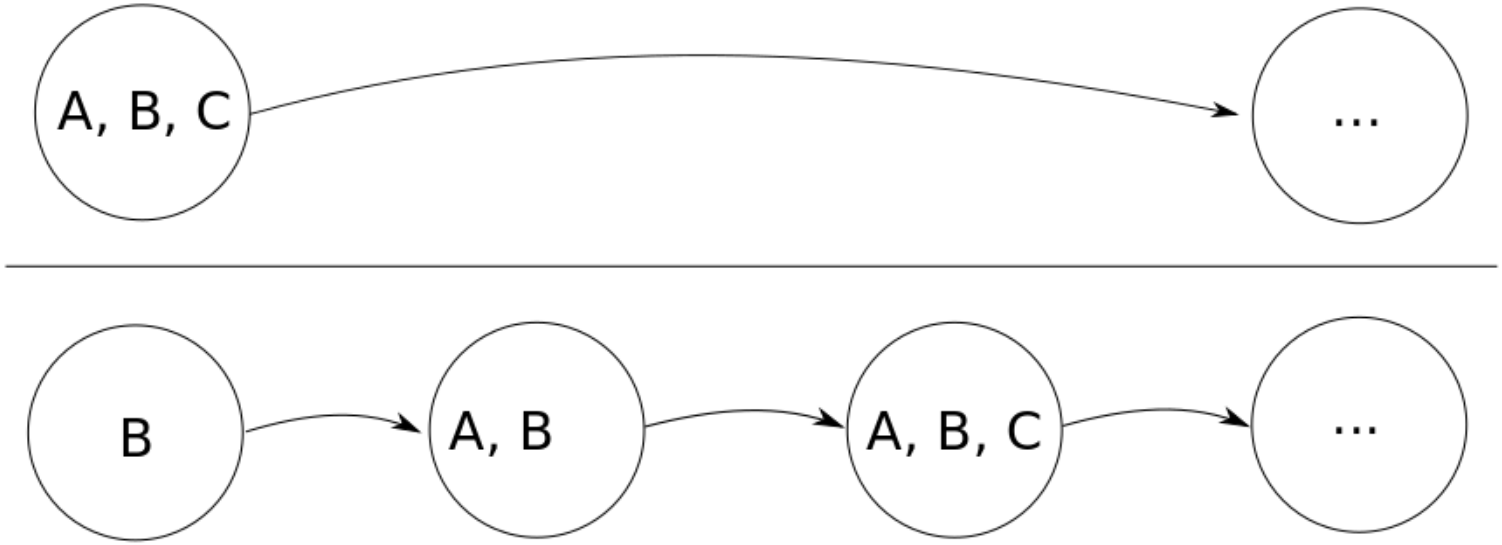
Le modèle MDP de base



À chaque instant t

- À l'état s^t, "choisir" l'action a^t.
- Aller à l'état s^{t+1} avec la récompense r^t

Le modèle MDP greedy



À chaque instant t

- Créer autant d'états intermédiaires que d'unités.
- Chaque unité tour à tour "choisit" une action.
- Quand la dernière joue, aller à l'état suivant et recevoir la récompense.

Q-learning (DQN)

- Off-policy
- Mise à jour des paramètres θ par descente de gradient :

$$\theta_{t+1} = \theta_t + \eta \left(r^t + \gamma \max_{a \in \mathcal{A}} Q_{\theta_t}(s^{t+1}, a) - Q_{\theta_t}(s^t, a^t) \right) \nabla_{\theta_t} Q_{\theta_t}(s^t, a^t)$$

- Phase d'entraînement stochastique via ϵ-greedy.
- Phase de test déterministe.

REINFORCE (PG)

- On-policy
- Apprentissage sur les traces $(s^t, a^t, s^{t+1}, r^{t+1})_{t=1,\dots,T-1}$ générées
- Mise à jour des paramètres θ par descente de gradient :

$$\theta_{k+1} = \theta_k + \eta \sum_{t=1}^{T} R^t \nabla_{\theta_k} [log\ \pi_{\theta_k}(a^t \mid s^t)]$$

- Gibbs policy :

$$\pi_\theta(a \mid s) = \frac{exp(\phi_\theta(a,s)/\tau)}{\sum_{b \in \mathcal{A}(s)} exp(\phi_\theta(b,s)/\tau)}$$

- ϕ_θ est le réseau de neurones de paramètres θ.

De l'apprentissage par renforcement sans gradient ?

Gradient-free en mode brutasse

- Soit une politique déterministe.
- Soit $R(\theta)$ sa récompense cumulative.
- Optimisation stochastique *gradient-free* :

$$\theta_{k+1} = \theta_k + \eta_k R(\theta_k + \delta u_k) u_k$$

avec u_k un vecteur random sur la sphère unité.

L'intuition derrière une optimisation *gradient-free*

L'idée

- Sans aléatoire, pas d'exploration avec une politique déterministe.
- On ajoute donc du bruit δu_k dans les paramètres θ.

L'intuition derrière une optimisation *gradient-free*

L'idée

- Sans aléatoire, pas d'exploration avec une politique déterministe.
- On ajoute donc du bruit δu_k dans les paramètres θ.

Passage d'un problème d'exploration à un autre

Espace d'action $\Rightarrow$ Espace de politique

L'intuition derrière une optimisation *gradient-free*

L'idée

- Sans aléatoire, pas d'exploration avec une politique déterministe.
- On ajoute donc du bruit δu_k dans les paramètres θ.

Passage d'un problème d'exploration à un autre

Espace d'action $\Rightarrow$ Espace de politique

Problème

- Dans un deep NN, il y a trop de paramètres θ : ce n'est pas gérable !
- Donc on ne fait ça que pour la dernière couche du NN.

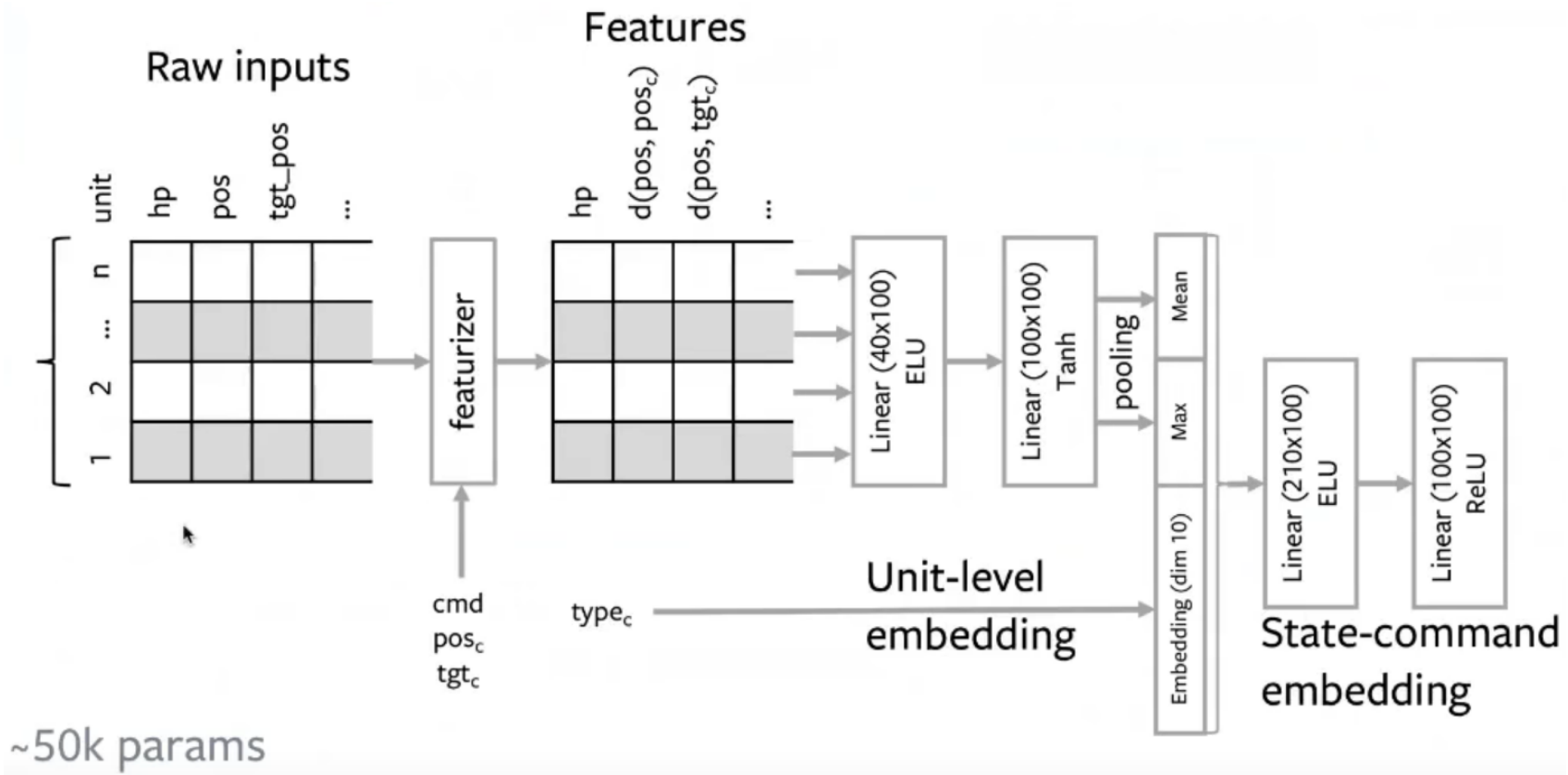
Sommaire

Le modèle

Les inputs

- Matrice où chaque ligne représente une unité.
- 17 features par unité : camp, type, HP, ... + 9 distances

Le modèle


Raw inputs
unit
hp
pos
tgt_pos
...
n
...
2
1
featurizer
Features
hp
d(pos, pos_c)
d(pos, tgt_c)
...
Linear (40x100) ELU
Linear (100x100) Tanh
pooling
Mean
Max
Embedding (dim 10)
Linear (210x100) ELU
Linear (100x100) ReLU
cmd
pos_c
tgt_c
type_c
Unit-level embedding
State-command embedding
~50k params

Sommaire

Backprop et zero-order gradient

Paramètres (θ, w)

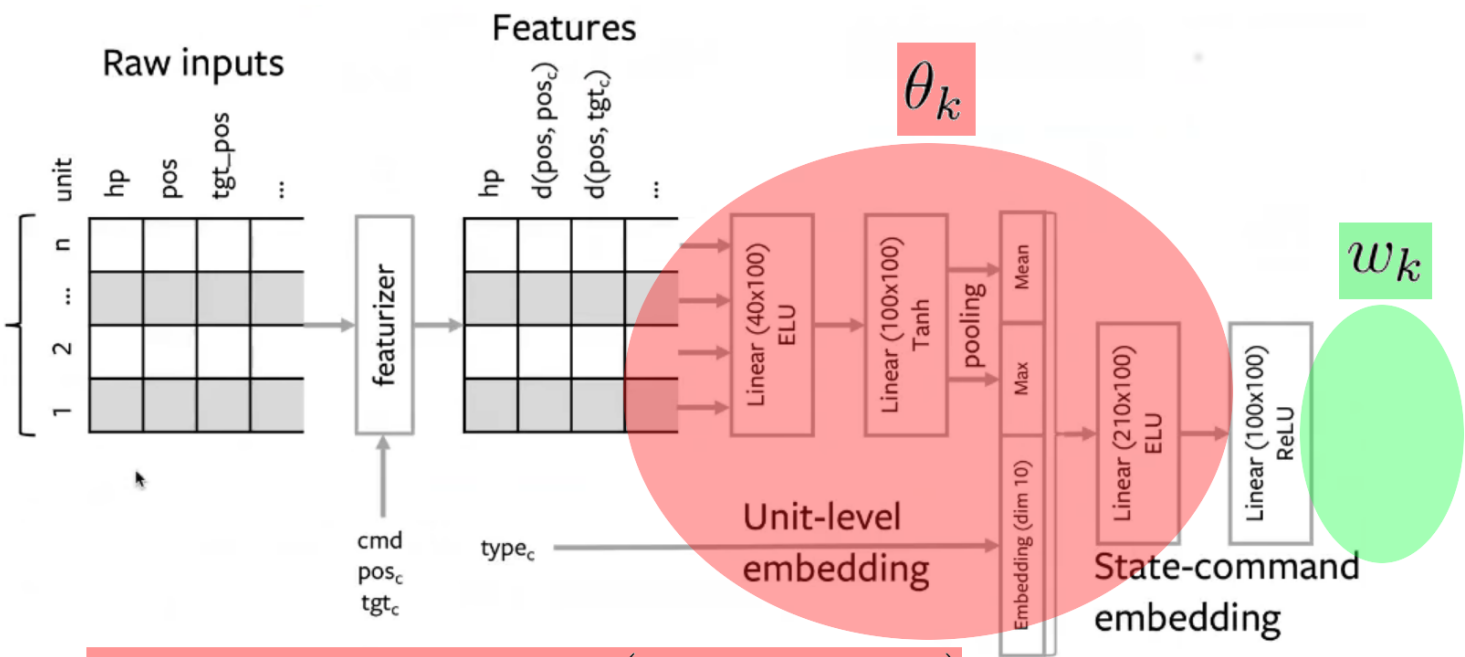
Deux types de paramètres :

- θ pour le NN (50k),
- w pour la dernière couche (100).

Zero-order backprop (ZO)

- Zero-order : $w_{k+1} = w_k + \frac{R}{t} u_k$
- Backprop : $\theta_{k+1} = \theta_k + backprop_{\phi_{\theta_k}} \left(\frac{R}{t} u_k \odot sign \ \frac{w}{\phi_{\theta_k}(s^t, a^t)} \right)$

Intuitivement



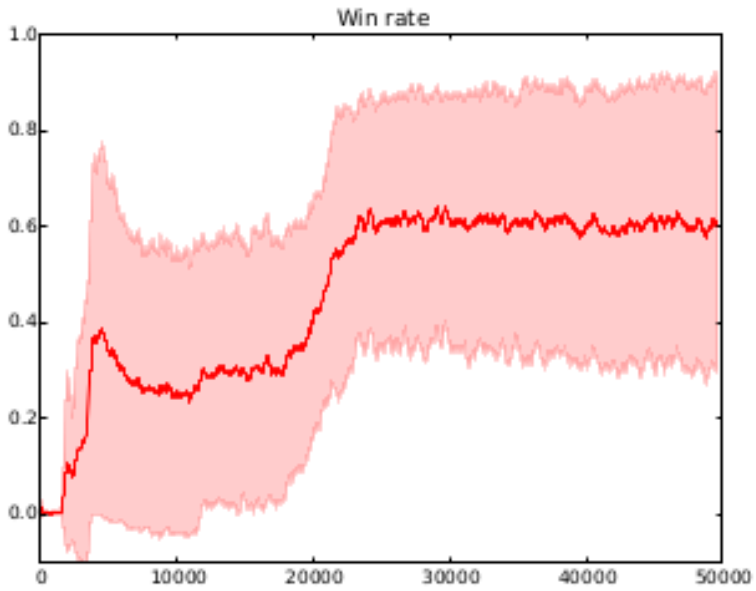
Backprop : $\theta_{k+1} = \theta_k + backprop_{\phi_{\theta_k}} \left(\frac{R}{t} u_k \odot sign \frac{w}{\phi_{\theta_k}(s^t, a^t)} \right)$

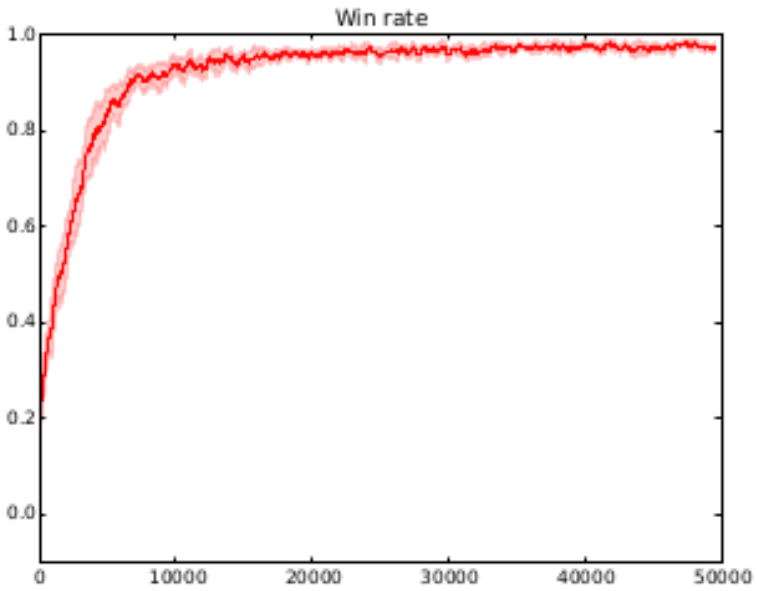
Zero-order : $w_{k+1} = w_k + \frac{R}{t} u_k$

Sommaire

Résultats expérimentaux

Figure 1: Example of the training uncertainty (one standard deviation) on 5 different initialization for DQN (left) and zero-order (right) on the m5v5 scenario.





Résultats expérimentaux

Table 1: Test win rates over 1000 battles for the training scenarios, for all methods and for heuristics baselines. The best result for a given map is in bold.

	heuristics					RL		
map	rand_nc	noop	c	wc	nok_nc	DQN	PG	ZO
dragoons_zealots	.14	.49	.67	.83	.50	.61	.69	**.90**
m5v5	.49	.84	.94	.96	.83	**.99**	.92	**1.**
m15v16	.00	**.81**	**.81**	.10	.68	.13	.19	**.79**
w15v17	.19	.10	.20	.02	.12	.16	.14	**.49**

Résultats expérimentaux

train map	test map	best heuristic	DQN	PG	ZO
m15v16	m5v5	**.96** (wc/c)	***.96***	.79	.80
	m15v15	**.97** (c)	.27	.16	*.80*
	m18v18	**.98** (c/noop)	.18	.25	*.82*
	m18v20	**.63** (noop)	.00	.01	*.17*
w15v17	w5v5	**.78** (c)	.70	.70	*.74*
	w15v13	1. (rand_nc/c)	1.	.99	1.
	w15v15	.95 (c)	.87	.61	***.99***
	w18v18	**.99** (c)	.92	.56	***1.***
	w18v20	.71 (c)	.31	.24	***.76***

Résultats expérimentaux

Table 4: Win rates over 2000 games against each other.

trained on	dragoons_zealots	m15v16		m5v5	w15v15		w15v17	
tested on	dragoons_zealots	m15v15	m18v18	m5v5	w15v15	w18v18	w15v15	w18v18
PG > DQN	.74	.46	.47	.49	.61	.69	.09	.04
ZO > PG	.76	.82	.79	.44	.82	.77	.98	.99
ZO > DQN	.93	.85	.86	.39	.88	.90	.79	.80

Sommaire

Perspectives

- ConvNet 2D pour apprendre les formes.
- Fusion de RL et d'exemples humains.
- Hierarchical RL pour apprendre des concepts (fuites, ...)