# Généralisation en apprentissage par renforcement

- Jusqu'à maintenant, on a supposé que les fonctions de valeur ou action-valeur étaient représentées sous forme de tableau

| $s$ | $V(s)$ |
|---|---|
| $s_0$ | -0.2 |
| $s_1$ | -0.1 |
| $s_2$ | 3 |
| ... | ... |

- Cela pose deux problèmes
  - pas pratique lorsque l'espace d'états $S$ est immense
  - impossible de **généraliser** à un état $s_i$ jamais visité jusqu'à maintenant

# Généralisation en apprentissage par renforcement

- Exemple: poker

| *s* | *V*(*s*) |
|---|---|
| « quatre As, un roi de pique » | 10.1 |
| « quatre As, un trois de coeur » | 9.9 |
| « quatre As, un dix de carreau » | ? |
| ... | ... |

- Le fait qu'on ait quatre As est le facteur déterminant de la valeur de sa main
  - on voudrait généraliser cette observation à toutes mains avec quatre As, quelque soit la cinquième carte

# Généralisation à l'aide d'approximateur de fonction

- Idée: remplacer la table $V(s)$ par une fonction linéaire $V_{\boldsymbol{\theta}}(s)$ avec
  - des **caractéristiques** $f_i(s)$ décrivant l'état $s$
  - des **paramètres** $\theta_i$ à apprendre

$$V_\theta(s) = \theta_1\, f_1(s) + \theta_2\, f_2(s) + \theta_3\, f_3(s) + \ldots + \theta_n\, f_n(s)$$

- Exemple de caractéristique pour le poker:

$$f_i(s) = \begin{cases} 1 & \text{si } s \text{ est une main avec quatre As} \\ 0 & \text{sinon} \end{cases}$$

- L'apprentissage vise à **trouver la valeur des paramètres $\boldsymbol{\theta}_i$ tels que $V_{\boldsymbol{\theta}}(s)$ est proche de $V(s)$**, pour tout état $s$

# Généralisation à l'aide d'approximateur de fonction

- **Application à l'apprentissage TD**

$$V(s) \leftarrow V(s) + \alpha\, \underbrace{(\, R(s) + \gamma\, V(s') - V(s)\,)}_{}$$

peut être vu comme $- \dfrac{\partial\, Loss}{\partial\, V(s)}$

- On dérive une règle d'apprentissage comme avec la règle de chaînage

$$\theta_i \leftarrow \theta_i + \alpha\, \underbrace{(\, R(s) + \gamma\, V_\theta(s') - V_\theta(s)\,)}_{- \frac{\partial\, Loss}{\partial\, V_\theta(s)}}\, \underbrace{f_i(s)}_{\frac{\partial V_\theta(s)}{\partial\, \theta_i}}$$

# Généralisation à l'aide d'approximateur de fonction

- **Application au Q-learning**
  - on utilise des caractéristiques de paires (état,action)

$$Q_\theta(s,a) = \theta_1\, f_1(s,a) + \theta_2\, f_2(s,a) + \theta_3\, f_3(s,a) + \ldots + \theta_n\, f_n(s,a)$$

ex.: $f_1(s,a)$ = 1 si main avec 4 As et action $a$ est « *all in* »

- On dérive la règle d'apprentissage similairement

$$\theta_i \leftarrow \theta_i + \alpha\, \underbrace{(\, R(s) + \gamma \max Q_\theta(s',a') - Q_\theta(s,a)\,)}_{-\frac{\partial\, Loss}{\partial\, Q_\theta(s,a)}}\, \underbrace{f_i(s,a)}_{\frac{\partial Q_\theta(s,a)}{\partial\, \theta_i}}$$

# Généralisation à l'aide d'approximateur de fonction

- **Application à l'approche PDA**
  - on utilise une fonction pour modéliser les probabilités $P(s'|s,a)$

$$P(s'|s,a) \propto \exp(\, \theta_1\, f_1(s',s,a) + \theta_2\, f_2(s',s,a) + \theta_3\, f_3(s',s,a) + \ldots + \theta_n\, f_n(s',s,a)\,)$$

  - possible de définir un coût approprié et d'en dériver un gradient
    - » c'est un problème d'apprentissage supervisé
    - » on doit prédire la cible *s'* étant donnée l'entrée constituée de *s* et de *a*

- Possible d'appliquer à la recherche de plan/politique
  - voir section 21.5 du livre

# d’approximateur de fonction

d’application: **IA pour le backgammon** (*TD-Gammon*)

- idée: modéliser la fonction de (action-)valeur **à l’aide d’un réseau de neurones**
  - plus puissant qu’une fonction linéaire
- première approche: demander à cibles à prédire
  - approche d’apprentissage supervisée, et non par renforcement
  - **problème**
  - **résultat**: n’est pas compétitif contre un humain
- deuxième approche: apprentissage TD actif avec réseau de neurones
  - le réseau de neurones **joue contre lui-même** pendant plusieurs jours
  - (300 000 parties!)
  - **résultat**: compétitif par rapport aux trois meilleurs joueurs


Hugo Larochelle et Froduald Kabanza