Βαθιά Αυτοενισχυόμενη Μάθηση(Deep Reinforcement Learning)

Βαθιά Μηχανική Μάθηση

ΔΠΜΣ Επιστήμης Δεδομένων & Μηχανικής Μάθησης

Εθνικό Μετσόβιο Πολυτεχνείο

Γιώργος Αλεξανδρίδης

Αυτοενισχυόμενη Μάθηση

Reinforcement Learning – RL

Μοντέλα Μηχανικής Μάθησης

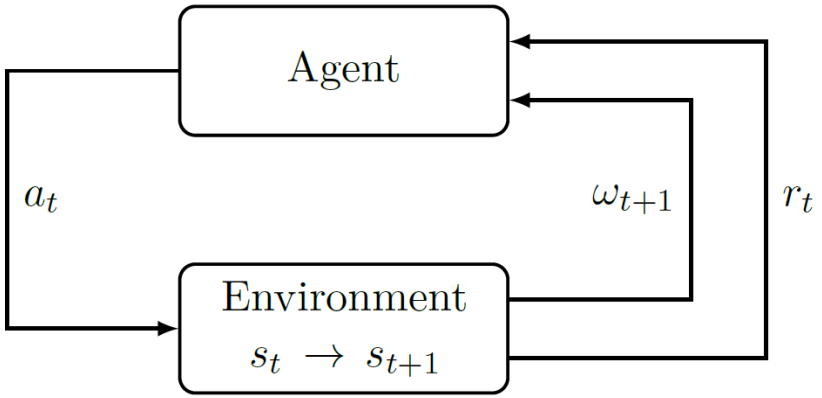
1. Επιβλεπόμενη Μάθηση
2. Μη-επιβλεπόμενη Μάθηση
3. Αυτοενισχυόμενη Μάθηση (RL)
   - Εμπνευσμένη από τον μπιχεβιορισμό (behaviorism) και τον τρόπο που μαθαίνουν οι βιολογικοί οργανισμοί
   - Αλληλεπίδραση με το περιβάλλον
     - Στοχαστικό ή όχι, πλήρως ή μερικώς παρατηρίσιμο, οι παρατηρήσεις μπορεί να είναι πολυδιάστατες, ο πράκτορας μπορεί να συλλέγει ελεύθερα εμπειρία από το περιβάλλον ή να είναι περιορισμένος,…
   - Βέλτιστη κατάσταση προσεγγίζεται με τη μορφή μικρών σωρευτικών ανταμοιβών (cumulative rewards)
   - Κατάλληλη για ακολουθιακά προβλήματα λήψης αποφάσεων (sequential decision-making problems), όπου το ορθό της απόφασης βασίζεται στην πρότερη εμπειρία (past experience)

Ευφυής δράστης/πράκτορας RL

- Μαθαίνει «καλές» συμπεριφορές
- Μάθηση μέσω δοκιμής και σφάλματος (trial and error)
- Δεν χρειάζεται να έχει πλήρη γνώση του περιβάλλοντος, αρκεί να μπορεί να αλληλεπιδρά με αυτό
  - Σε αντίθεση λχ με δυναμικό προγραμματισμό που απαιτεί πλήρη γνώση του περιβάλλοντος από πριν
- Online μάθηση
  - Ο πράκτορας μαθαίνει σε «πραγματικό» χρόνο ακολουθιακά, τη στιγμή που παρατηρεί το περιβάλλον
  - Μπορεί ωστόσο να συγκεντρώσει και την εμπειρία εκ των προτέρων $\Longrightarrow$ offline μάθηση, γνωστή και ως batch RL (μάθηση κατά δέσμες)

Τυπικός ορισμός RL μάθησης

- Ένα πρόβλημα RL ορίζεται ως μια στοχαστική διαδικασία ελέγχου διακριτού χρόνου, όπου ο πράκτορας:
  1. Τη χρονική στιγμή t επιλέγει τη δράση α_t
  2. Το περιβάλλον του πράκτορα μεταβαίνει από την κατάσταση s_t στην κατάσταση s_{t+1}
  3. λαμβάνει ανταμοιβή r_t
  4. κάνει την παρατήρηση ω_{t+1}
- Στην αρχή του χρόνου, ο πράκτορας βρίσκεται στην κατάσταση α_0 και κάνει την αρχική παρατήρηση ω_0



Μαρκοβιανή Ιδιότητα

- Μια στοχαστική διαδικασία ελέγχου διακριτού χρόνου έχει τη μαρκοβιανή ιδιότητα (markovian property) αν ισχύει
  - $\mathbb{P}(\omega_{t+1}|\omega_t, \alpha_t) = \mathbb{P}(\omega_{t+1}|\omega_t, \alpha_t, \dots, \omega_0, \alpha_0)$ και
  - $\mathbb{P}(r_t|\omega_t, \alpha_t) = \mathbb{P}(r_t|\omega_t, \alpha_t, \dots, \omega_0, \alpha_0)$
- Δηλαδή η επόμενη τιμή εξαρτάται μόνο από την τωρινή και όχι τις παρελθοντικές
  - Ιδιότητα «αμνησίας»
- Μια στοχαστική διαδικασία ελέγχου διακριτού χρόνου που έχει τη μαρκοβιανή ιδιότητα ονομάζεται μαρκοβιανή διαδικασία απόφασης (Markov Decision Process – MDP)

Μάθηση RL ως MDP

- Μια πλειάδα $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$ 5 στοιχείων, όπου
  1. $\mathcal{S}$ χώρος καταστάσεων
  2. $\mathcal{A}$ χώρος δράσεων
  3. $T: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \longrightarrow [0,1]$ η συνάρτηση μετάβασης (transition function)
     - Σύνολο υπό συνθήκη πιθανοτήτων μετάβασης μεταξύ των καταστάσεων
  4. $\mathcal{R}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \longrightarrow \mathcal{R}$ η συνάρτηση ανταμοιβής (reward function)
     - $\mathcal{R}$ συνεχής σε εύρος $[0, R_{max}]$, όπου $R_{max} \in \mathbb{R}^+$
  5. $\gamma \in [0,1)$
     - Συντελεστής ελάττωσης (discount factor) ανταμοιβής
- Το σύστημα είναι πλήρως παρατηρήσιμο $\Longrightarrow \omega_t = s_t$

Μάθηση πράκτορα RL

- Εύρεση πολιτικής $\pi \in \Pi$ μετάβασης από τη μια κατάσταση στην άλλη
- Μπορεί να πραγματοποιηθεί με συνδυασμό κάποιων (ή όλων) από τους παρακάτω τρόπους
  1. Με απευθείας εκτίμηση της $\pi(s)$ ή της $\pi(s, a)$
  2. Με τη χρήση συνάρτησης αποτίμησης κατάστασης (state value function) που εκτιμά πόσο επωφελής είναι μια κατάσταση ή ένα ζεύγος κατάστασης-δράσης για τον πράκτορα
  3. Με την κατασκευή μοντέλου (model) για το περιβάλλον
- Συνδυασμός περιπτώσεων 1 και 2⇒ RL χωρίς μοντέλο (model-free RL)
- Περίπτωση 3 ⇒ RL βασισμένη σε μοντέλο (model-based RL)

Τεχνικές εκτίμησης πολιτικής

1. Στάσιμες (stationary) ή Μη-στάσιμες (non-stationary)
   - Στη δεύτερη περίπτωση, η πιθανότητα μετάβασης εξαρτάται από t
2. Ντετερμινιστικές ή Στοχαστικές
   - Ντετερμινιστικές
     - $\pi(s): \mathcal{S} \to \mathcal{A}$
   - Στοχαστικές
     - $\pi(s, a): \mathcal{S} \times \mathcal{A} \longrightarrow [0,1]$
     - $\pi(s, a)$ η περίπτωση να επιλεγεί η δράση α όταν ο πράκτορας βρίσκεται στην κατάσταση s

Συνάρτηση αποτίμησης κατάστασης

- Πόσο ωφέλιμη είναι μια κατάσταση s για τον πράκτορα;
  - Επιστρεφόμενη τιμή (Return value): $\mathrm{R} = \sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s$
  - Άθροισμα γεωμετρικής σειράς
    - Η ανταμοιβή δεν τείνει στο άπειρο, όσο περνάει ο χρόνος
    - Ο πράκτορας προτιμά τις πιο «άμεσες» ανταμοιβές
- Αναμενόμενη επιστρεφόμενη τιμή
  - $V_\pi(s) = \mathbb{E}[\mathrm{R}]$

Συνάρτηση Τιμής (Value function)

- Ποια από τις διαθέσιμες πολιτικές $\pi(s) \in \Pi$ μεγιστοποιεί την αναμενόμενη τιμή επιστροφής R σε μια κατάσταση s;
  - $V^{\pi}(s) = \mathbb{E}[\mathrm{R}|\pi] = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t | s_0 = s\,, \pi]$, με $V^{\pi}(s): \mathcal{S} \to \mathbb{R}$
- Η $V^{\pi}(s)$ καλείται συνάρτηση V-value
  - Από τον ορισμό της μεγιστοποιείται εκεί που $V^{*}(s) = \max_{\pi(s)\in\Pi} V^{\pi}(s)$
  - Βέλτιστη πολιτική π^{*} που μεγιστοποιεί V-value
    - $\pi^{*} = arg\max_{\pi} V^{\pi}(s)$

Συνάρτηση Q-value

- Παρότι η αποτίμηση καταστάσεων αρκεί για τους πράκτορες RL, σε πολλές περιπτώσεις η αποτίμηση ζευγών κατάστασης-δράσης μπορεί να είναι ιδιαίτερα βοηθητική
- Ποια από τις διαθέσιμες πολιτικές $\pi(s,\alpha) \in \Pi$ μεγιστοποιεί την αναμενόμενη τιμή επιστροφής R ζεύγους κατάστασης-δράση (s,α);
  - $Q^{\pi}(s,a) = \mathbb{E}[\mathrm{R}|\pi] = \mathbb{E}[\sum_{t=0}^{\infty}\gamma^{t}r_{t}|s_{0}=s,\alpha_{0}=\alpha\,,\pi]$, με $Q^{\pi}(s,a)\colon \mathcal{S}\times\mathcal{A}\to\mathbb{R}$
- Η $Q^{\pi}(s,a)$ καλείται συνάρτηση Q-value
  - Από τον ορισμό της μεγιστοποιείται εκεί που $Q^{*}(s,\alpha) = \max_{\pi(s,\alpha)\in\Pi} Q^{\pi}(s,\alpha)$
  - Βέλτιστη πολιτική π^{*} που μεγιστοποιεί Q-value
    - $\pi^{*}(s) = arg\max_{a\in\mathcal{A}} Q^{*}(s,\alpha)$

Στρατηγικές προσδιορισμού βέλτιστης πολιτικής

1. Brute force
   - Δοκίμασε δειγματοληπτικά κάθε δυνατή πολιτική
   - Επέλεξε εκείνη με τη μεγαλύτερη αναμενόμενη επιστροφή
   - Προβληματική προσέγγιση σε μεγάλους χώρους καταστάσεων ή/και ανταμοιβών
2. Επανάληψη τιμών (value iteration)
3. Επανάληψη πολιτικών (policy iteration)

Επανάληψη τιμών

- Προτάθηκε το 1957 από τον Bellman
- Καλείται και προς τα πίσω επαγωγή (backward induction)
  - Βέλτιστη τιμή $V^*(s)$ για βέλτιστη στρατηγική π^* αναλύεται σε
    - $V^*(s) = r_0 + \gamma \max_{s_0} r_1 + \gamma^2 \max_{s_1} r_2 + \cdots = \mathbb{E}_{s'}[r + \gamma V(s')]$
    - Ξεκινάω από εκτίμηση V_0 και υπολογίζω επαναληπτικά εκτιμήσεις V_i για όλες τις καταστάσεις μέχρι η σειρά να συγκλίνει επαναληπτικά στην αναμενόμενη τιμή του δεξιότερου τμήματος της Σχέσης που καλείται Εξίσωση Bellman
    - Η πολιτική ενσωματώνεται μέσα στη V-value
  - Με αντίστοιχο τρόπο προσδιορίζεται και η $Q^*(s, \alpha)$
  - Η διαδικασία αυτή εγγυημένα συγκλίνει σε βέλτιστες τιμές

Επανάληψη πολιτικών

- Προτάθηκε το 1960 από τον Howard
- Η πολιτική δεν ενσωματώνεται στις τιμές των $V^{\pi}(s)$, $Q^{\pi}(s,a)$, όπως προηγουμένως
- Ομοιάζει, ως προς την φιλοσοφία, με αλγόριθμο expectation-maximization
  - Εκτίμηση π' για βέλτιστη πολιτική
  - Προσέγγιση βέλτιστων τιμών V, Q για τη συγκεκριμένη εκτίμηση
  - Ενημέρωση εκτίμησης π στη βάση των τιμών που υπολογίστηκαν στο προηγούμενο βήμα
  - Σύγκλιση όταν π δεν μεταβάλλεται πλέον μεταξύ των επαναλήψεων
- Πιο αργή σύγκληση σε σύγκριση με επανάληψη τιμών

Q-learning

- RL χωρίς μοντέλο
- $\mathcal{R}$ συνάρτηση ανταμοιβής $\Longrightarrow$ Πίνακας ανταμοιβής (reward table) R
  - Γνωστός εκ των προτέρων και σταθερός
  - Στο διπλανό παράδειγμα με -1 συμβολίζονται τα μη-δυνατά ζεύγη κατάστασης-δράσης
- $Q(s,a) \Longrightarrow$ πίνακας (Q-table)
  - Αρχικά κενός
  - Ενσωματώνει «εμπειρία» πράκτορα καθώς μαθαίνει

$$R = \begin{array}{c} \\ \text{State} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} \multicolumn{6}{c}{\text{Action}} \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Q-learning: Κανόνας μετάβασης

- $Q(s_t, \alpha_t) = R(s,a) + \gamma \max_{\alpha_{t+1}} Q(s_{t+1}, \alpha_{t+1})$
- Έστω ότι αρχικά βρίσκομαι στην κατάσταση 1
- Μέσω επανάληψης τιμών είτε επανάληψης πολιτικών επιλέγω τη δράση 5
- $Q(1,5) = R(1,5) + 0.8 * \max\{Q(5,1), Q(5,4), Q(5,5)\} = 100 + 0.8 * 0 = 100$

$$R = \begin{array}{c} \\ \text{State} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} \multicolumn{6}{c}{\text{Action}} \\ 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 & -1 & 100 \\ -1 & -1 & -1 & 0 & -1 & -1 \\ -1 & 0 & 0 & -1 & 0 & -1 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ -1 & 0 & -1 & -1 & 0 & 100 \end{bmatrix} \end{array}$$

$$Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \Rightarrow Q = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Q-learning: Κανόνας μετάβασης (συνέχεια)

- $Q(s_t, \alpha_t) = R(s,a) + \gamma \max_{\alpha_{t+1}} Q(s_{t+1}, \alpha_{t+1})$
- Έστω ότι επιλέγοντας τη δράση 5 (προηγούμενη διαφάνεια), βρέθηκα στην κατάσταση 3
- Μέσω επανάληψης τιμών είτε επανάληψης πολιτικών επιλέγω τη δράση 1
- $Q(3,1) = R(3,1) + 0.8 * \max\{Q(1,3), Q(1,5)\} = 0 + 0.8 * 100 = 80$

Action

State	0	1	2	3	4	5
0	−1	−1	−1	−1	0	−1
1	−1	−1	−1	0	−1	100
2	−1	−1	−1	0	−1	−1
3	−1	0	0	−1	0	−1
4	0	−1	−1	0	−1	100
5	−1	0	−1	−1	0	100

$R =$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	100
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0

$Q =$

$Q =$

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	100
2	0	0	0	0	0	0
3	0	80	0	0	0	0
4	0	0	0	0	0	0
5	0	0	0	0	0	0

Q-learning: Αλγόριθμος μάθησης

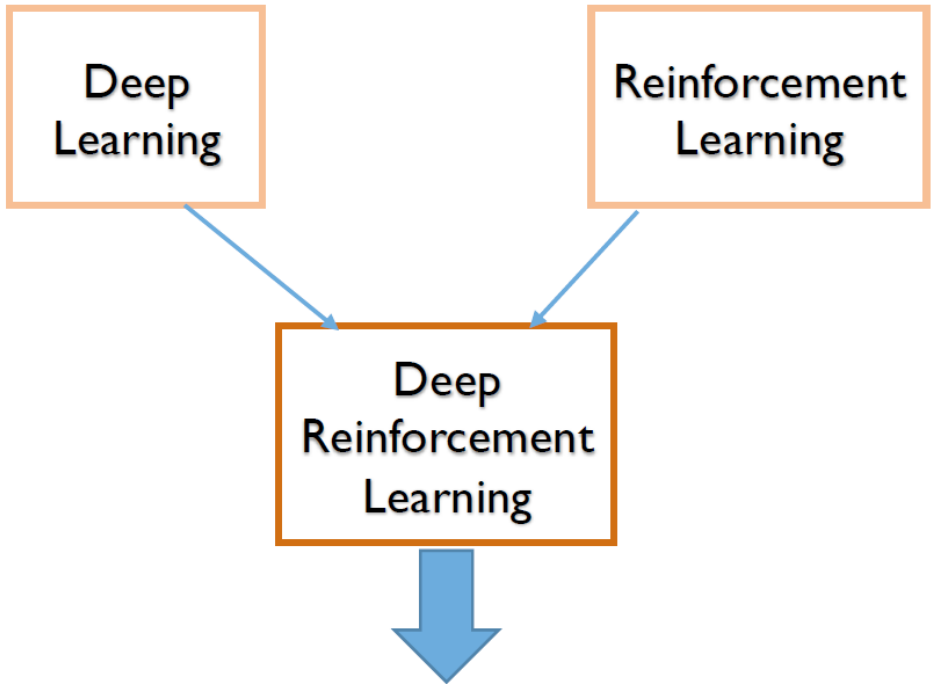
- Αλγόριθμος μάθησης
  1. Αρχικοποίηση πίνακα $Q(s,a)$
  2. Για όλα τα επεισόδια (εποχές) εκπαίδευσης
     i. Επέλεξε αρχική κατάσταση s
     ii. Επανέλαβε, μέχρις ότου να φτάσεις σε τελική κατάσταση
        a. Επέλεξε δράση α (πχ μέσω επανάληψης τιμών ή πολιτικών)
        b. Λάβε ανταμοιβή r και εξέτασε τη νέα κατάσταση s'
        c. $\hat{Q}(s_t, \alpha_t) \leftarrow \hat{Q}(s_t, \alpha_t) + \eta\left(R(s,a) + \gamma \max_{\alpha_{t+1}} \hat{Q}(s_{t+1}, \alpha_{t+1}) - Q(s_t, \alpha_t)\right)$
- η: ρυθμός μάθησης
- Κανόνας ανταμοιβής $R(s,a) + \gamma \max_{\alpha_{t+1}} \hat{Q}(s_{t+1}, \alpha_{t+1})$
  - Μπορεί να θεωρηθεί ως σύνολο δειγμάτων ζευγών κατάστασης, δράσης
  - Μέσω του κανόνα της μάθησης θέλουμε το $\hat{Q}(s_t, \alpha_t)$ να συγκλίνει στη μέση τιμή του

Βαθιά Αυτοενισχυόμενη Μάθηση

Deep Reinforcement Learning – DRL

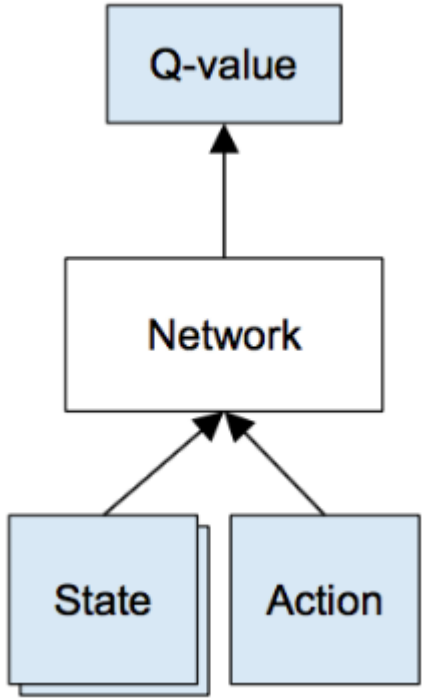
Βαθιά Αυτοενισχυόμενη Μάθηση

- Χρήση βαθιού δικτύου για την αναπαράσταση τιμής συνάρτησης, πολιτικής ή μοντέλου
- Βελτιστοποίηση τιμής συνάρτησης, πολιτικής ή μοντέλου
- Χρήση τεχνικών κατάβασης κλίσης



Deep Q-learning (DQN)

- Προσέγγιση συνάρτησης που καθορίζει τιμές Q-table
  - Χρήση (βαθιών) νευρωνικών δικτύων ως προσεγγιστών συναρτήσεων (function approximators)
- Παραλλαγές
  - Double DQN
    - Για τη διόρθωση υπερ-εκτιμήσεων
  - Delayed Q-learning with PAC
    - Χρήση online μάθησης



Μάθηση DQN

- Συνάρτηση απώλειας
  - $L = \frac{1}{2}\Big[\underbrace{r + \max_{a'} Q(s', a')}_{\text{στόχος}} - \underbrace{Q(s, a)}_{\text{πρόβλεψη}}\Big]^2$

- Αλγόριθμος ενημέρωσης παραμέτρων δικτύου
  1. Προς τα εμπρός διάσχιση του δικτύου (forward pass) για πρόβλεψη τιμών Q για όλες τις δράσεις
  2. Προς τα εμπρός διάσχιση του δικτύου για την επόμενη κατάσταση s' και υπολογισμός $\max_{a'} Q(s', a')$
  3. Για τη δράση a με το βέλτιστο αποτέλεσμα, ενημερώνουμε θέτοντας $Q(s, a) = r + \max_{a'} Q(s', a')$
     - Οι υπόλοιπες μένουν ως έχουν
  4. Ενημέρωση βαρών μέσω της προς τα πίσω διάδοσης του σφάλματος (back-propagation)

Επανάληψη Τιμών

- Αναπαράσταση συνάρτησης τιμής από ένα βαθύ Q δίκτυο με βάρη w

$$Q(s, a, w) \approx Q^{\pi}(s, a)$$

- Ορισμός αντικειμενικής συνάρτησης μέσου τετραγωνικού σφάλματος ως προς τις τιμές Q

$$\mathcal{L}(w) = \mathbb{E}\left[\left(\underbrace{r + \gamma \max_{a'} Q(s', a', w)}_{\text{στόχος}} - \underbrace{Q(s, a, w)}_{\text{πρόβλεψη}}\right)^2\right]$$

- Κλίση αντικειμενικής συνάρτησης

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \mathbb{E}\left[\left(r + \gamma \max_{a'} Q(s', a', w) - Q(s, a, w)\right) \frac{\partial Q(s, a, w)}{\partial w}\right]$$

- Χρήση κατάβασης κλίσης για την ενημέρωση των βαρών

Επανάληψη Πολιτικών

- Αναπαράσταση συνάρτησης τιμής από ένα βαθύ Q δίκτυο με βάρη w

$$Q(s, a, w) \approx Q^{\pi}(s, a)$$

- Ορισμός αντικειμενικής συνάρτησης μέσου τετραγωνικού σφάλματος ως προς τις τιμές Q

$$\mathcal{L}(w) = \mathbb{E}[\underbrace{(r + \gamma Q(s', a', w)}_{\text{στόχος}} - \underbrace{Q(s, a, w))}_{\text{πρόβλεψη}}^2]$$

- Κλίση αντικειμενικής συνάρτησης

$$\frac{\partial \mathcal{L}(w)}{\partial w} = \mathbb{E}\left[(r + \gamma Q(s', a', w) - Q(s, a, w)) \quad \frac{\partial Q(s, a, w)}{\partial w}\right]$$

- Χρήση κατάβασης κλίσης για την ενημέρωση των βαρών

Ζητήματα ευστάθειας

- Ο απλός Q-learning αλγόριθμος εμφανίζει φαινόμενα ταλαντώσεων ή αποκλίσεων όταν εφαρμόζεται στα νευρωνικά δίκτυα
  1. Δεδομένα είναι ακολουθιακά
     - Τα νευρωνικά δίκτυα προϋποθέτουν ότι οι εισόδοι τους είναι i.i.d και όχι συσχετισμένες
  2. Μικρές αλλαγές στα Q-values οδηγούν σε αλλαγές πολιτικών
  3. Εύρος ανταμοιβών και Q-values όχι γνωστό πριν την εκπαίδευση
     - Οι κλίσεις μπορούν να γίνουν ασταθείς κατά τη φάση της προς τα πίσω διάδοσης του σφάλματος
- Τα προβλήματα αυτά αντιμεωπίζονται από τον DQN
  - ευσταθής αλγόριθμος για DRL δίκτυα που βασίζονται στην εκτίμηση τιμών

Αλγόριθμος DQN

- Προσφέρει σταθερή λύση στις βαθιές μεθόδους RL βασισμένες σε τιμές
  1. Επανάληψη εμπειρίας
     - «Σπάσιμο» των συσχετίσεων στα δεδομένα, επαναφορά των i.i.d συνθηκών
     - Εκμάθηση από όλες τις προηγούμενες πολιτικές
  2. Πάγωμα του Q-δικτύου
     - Αποφυγή ταλαντώσεων
     - «Σπάσιμο» συσχετίσεων μεταξύ του δικτύου Q και του στόχου της μάθησης
  3. Περικοπή των ανταμοιβών ή κανονικοποίηση της προσαρμογής του δικτύου εντός λογικών ορίων
     - «Εύρωστες» (robust) κλίσης

Επανάληψη Εμπειρίας

- Experience Replay
- Για την αφαίρεση συσχετίσεων, κατασκευή συλλογής δεδομένων από την εμπειρία του ίδιου του πράκτορα
  - Επιλογή δράσης a_t, στη βάση «άπληστης» πολιτικής
  - Αποθήκευση μετάβασης $(s_t, a_t, r_{t+1}, s_{t+1})$ στη μνήμη επανάληψης D
  - Δειγματοληψία μικρο-δεσμών (mini-batches) μεταβάσεων $(s, a, r\ , s')$ από το D
  - Βελτίστοποίηση μέσου τετραγωνικού σφάλματος μεταξύ δικτύου Q και στόχων μάθησης Q.

$$\mathcal{L}(w) = \mathbb{E}_{s,a,r,s' \sim D}\left[\left(r + \gamma \max_{a'} Q(s', a', w) - Q(s, a, w)\right)^2\right]$$

«Πάγωμα» Q-δικτύου στόχου

- Για την αποφυγή ταλαντώσεων, «πάγωμα» των παραμέτρων που χρησιμοποιούνται για το Q-learning
  - Υπολογισμός μετρικών Q-learning με τις παλιές (προηγούμενες) παραμέτρους w^-

$$r + \gamma \max_{a'} Q(s', a', w^-)$$

  - Βελτιστοποίηση μέσου τετραγωνικού σφάλματος μεταξύ δικτύου Q και μάθησης Q

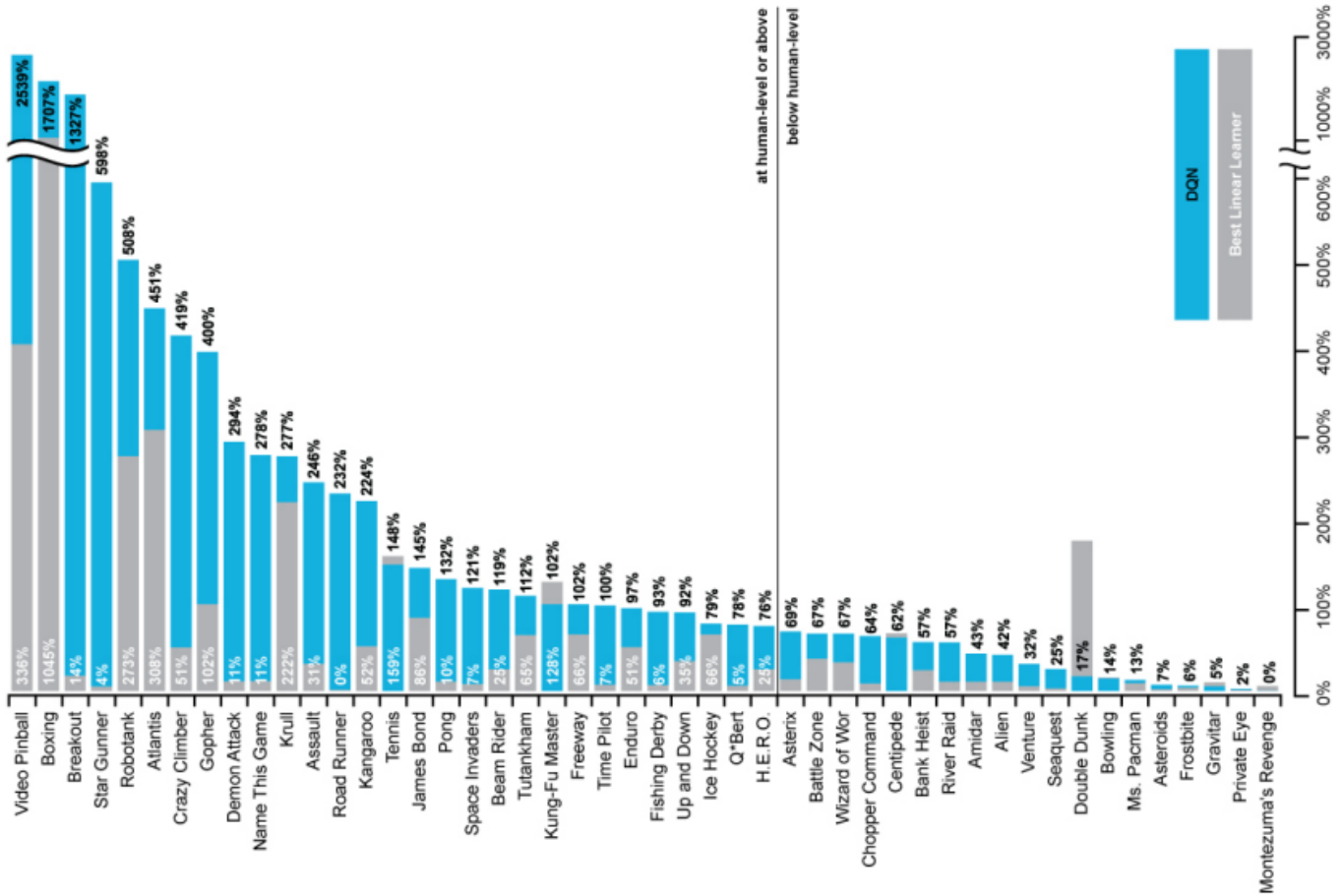
$$\mathcal{L}(w) = \mathbb{E}_{s,a,r,s' \sim D}\left[\left(r + \gamma \max_{a'} Q(s', a', w^-) - Q(s, a, w^-)\right)^2\right]$$

  - Ανά περιόδους, ενημέρωση σταθερών παραμέτρων $w^- \longleftarrow w$

Περικοπή Ανταμοιβών

- Περικοπή ανταμοιβών από το DQN στο [-1,+1]
- Κατ' αυτόν τον τρόπο, οι τιμές Q δεν γίνονται ποτέ πολύ υψηλές
- Οι συνεπαγόμενες κλίσεις είναι σαφώς ορισμένες

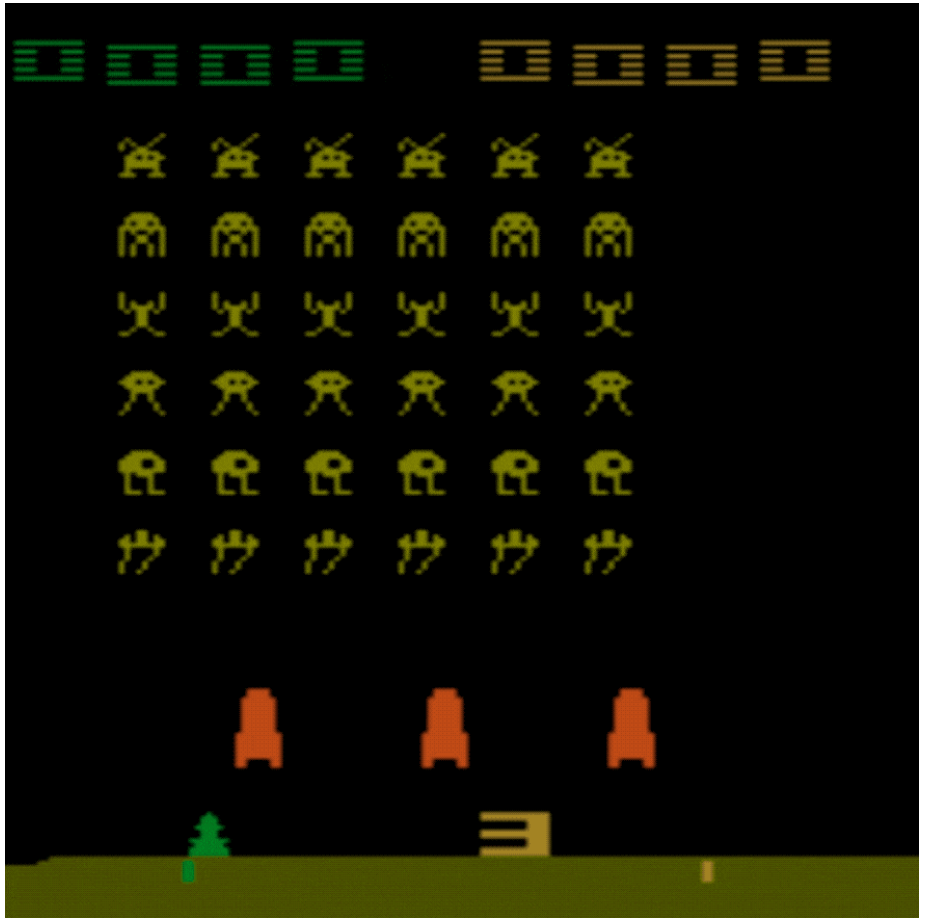
Αποτελέσματα εφαρμογής DQN στο Atari



Εφαρμογές

Atari (2015)

- Εκμάθηση τρόπου παιξίματος παιχνιδιών κονσόλας Atari εξετάζοντας μόνο τα pixels τους
- Αλγόριθμος DDQN – Double Q-Learning
  - https://www.deepmind.com/open-source/dqn
- Η εργασία δημοσιεύτηκε στο περιοδικό Nature το 2015
  - https://www.nature.com/articles/nature14236



AlphaGo (2016)

- Χρήση DRL και Monte Carlo Tree Search για κατασκευή ευφυούς πράκτορα για το παιχνίδι Go
- Νίκησε τον Ευρωπαίο πρωταθλητή του Go Fan Hui και στις 5 παρτίδες που παίχτηκαν μεταξύ 5-9 Οκτωβρίου 2015.
- Η εργασία δημοσιεύτηκε στο περιοδικό Nature το 2016

Βιβλιογραφία

- Mnih et al (2015) – Human-level control through deep reinforcement learning
  - Αλγόριθμος DQN, Experience Replay, «Πάγωμα» Q-δικτύου, Reward clipping
- Tamar et al (2016) – Value Iteration Networks
  - Value Iteration Networks
- Silver et al (2015) – Mastering the game of Go with deep neural networks and tree search
  - Deep Reinforcement Learning & Monte Carlo Tree Search