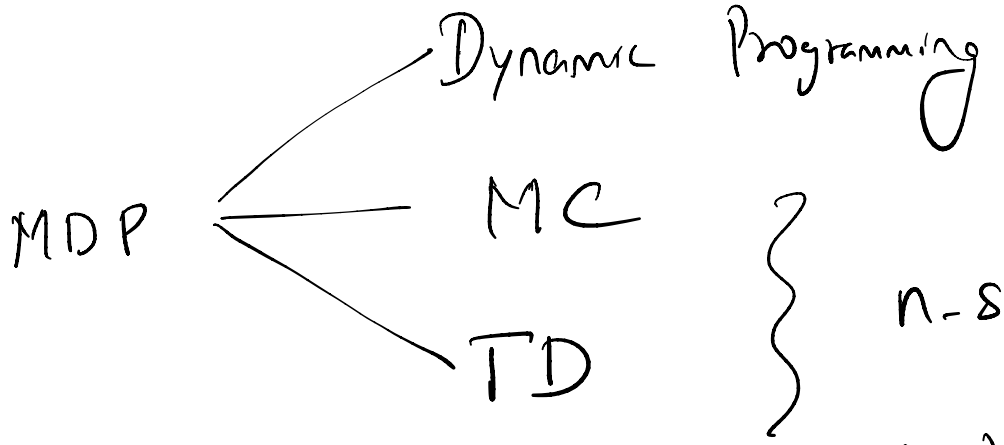
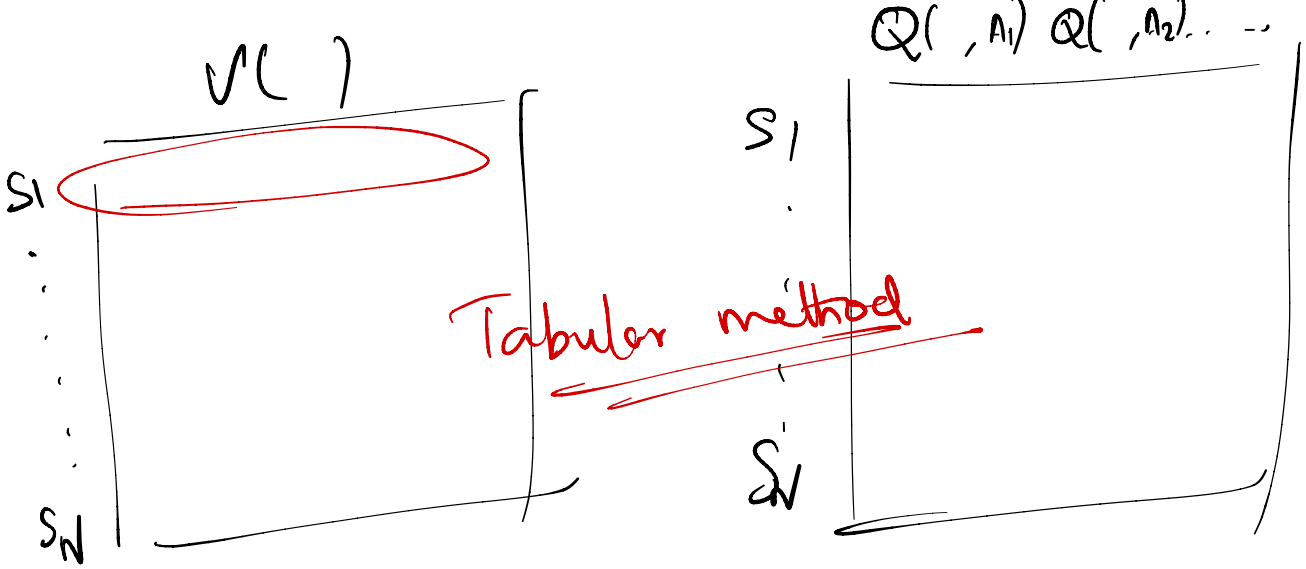
Lecture -07

Value Function Approximation



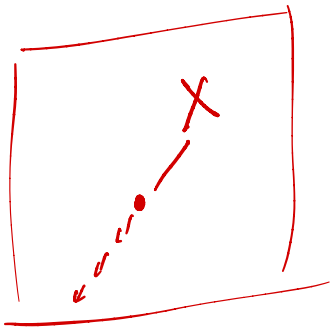


$S_t \longrightarrow x(S_t)$

$$\hat{v}(s;w) = w^T x$$

Non-linear function approximators :-

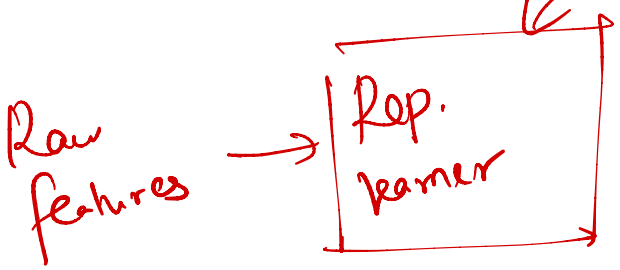
$$\hat{v}(s;\omega) = \omega^T x$$



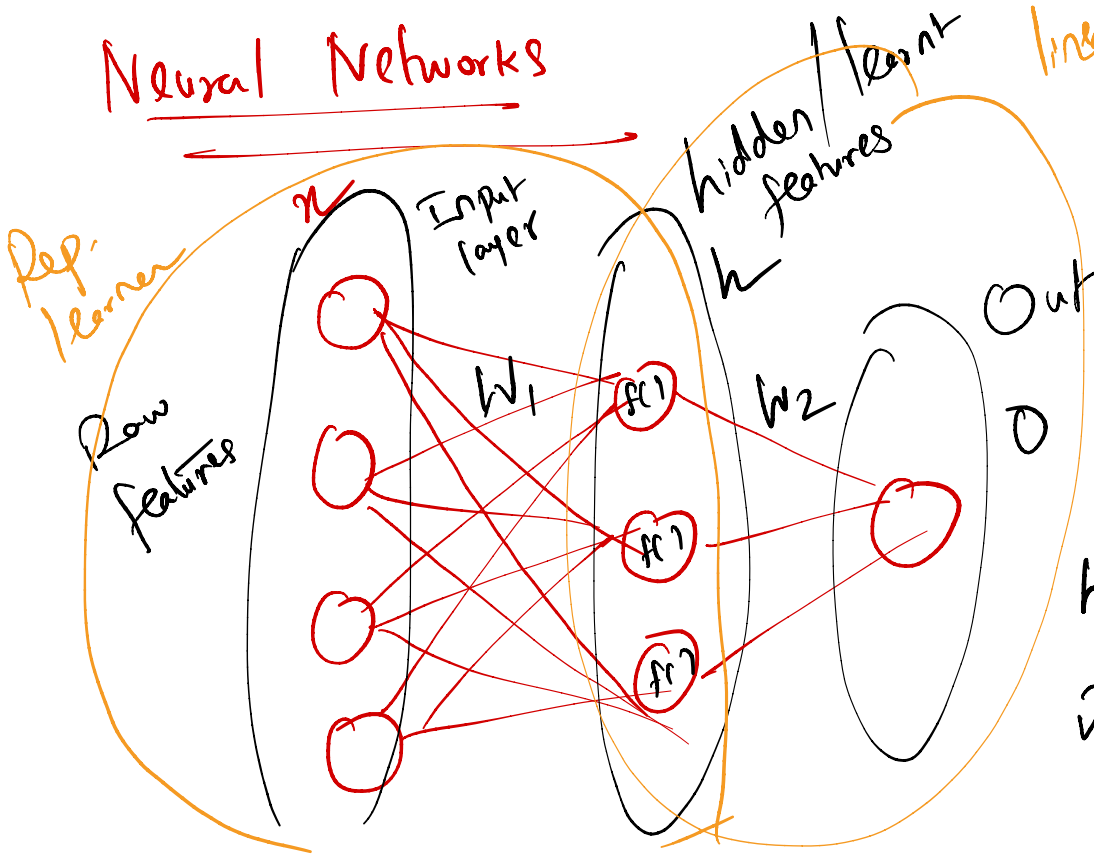
① Carefully handcrafted features.

② interaction between the features is limited.

Representation learning :- non-linear model.



Neural Networks



$$h = f(W_1 n + b_1)$$

$$\hat{v}(s;NN) = W_2 h + b_2$$

$$h = f(W_1 x + b_1)$$

$$\hat{v}(s;\theta) = W_2 h + b_2$$

$$\theta = (W_1, b_1, W_2, b_2)$$

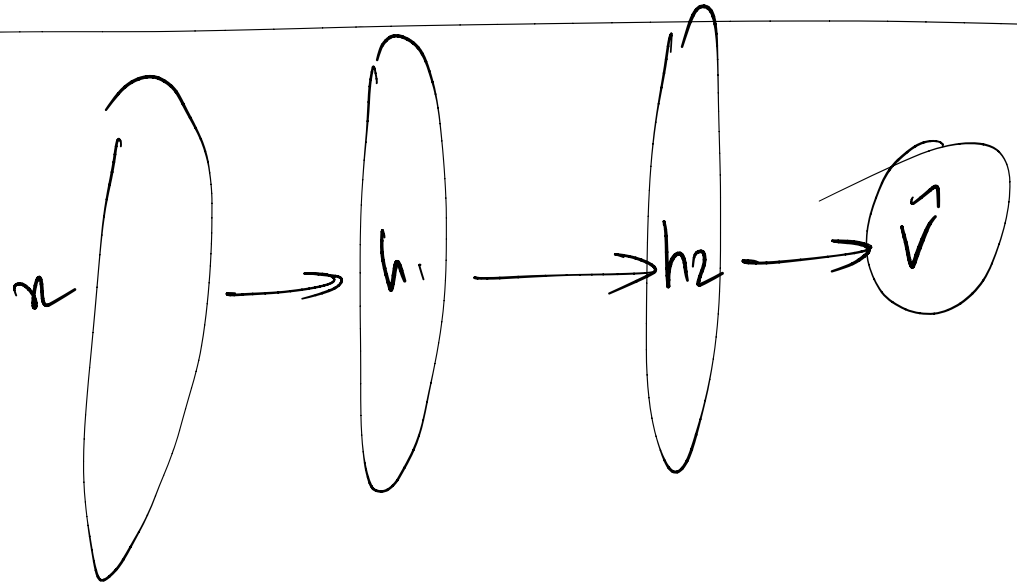
Non-linear activation fn:

1) Sigmoid $\sigma(a) = \frac{1}{1+e^{-a}}$

2) tanh $\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}}$

3) ReLU $(a) = \max(0, a)$

Rectified linear unit



Fully connected network:

Multi layer perceptron (MLP)

$$h_1 = \sigma(w_1 x + b_1)$$

$$h_2 = \sigma(w_2 h_1 + b_2)$$

$$o = w_3 h_2 + b_3$$

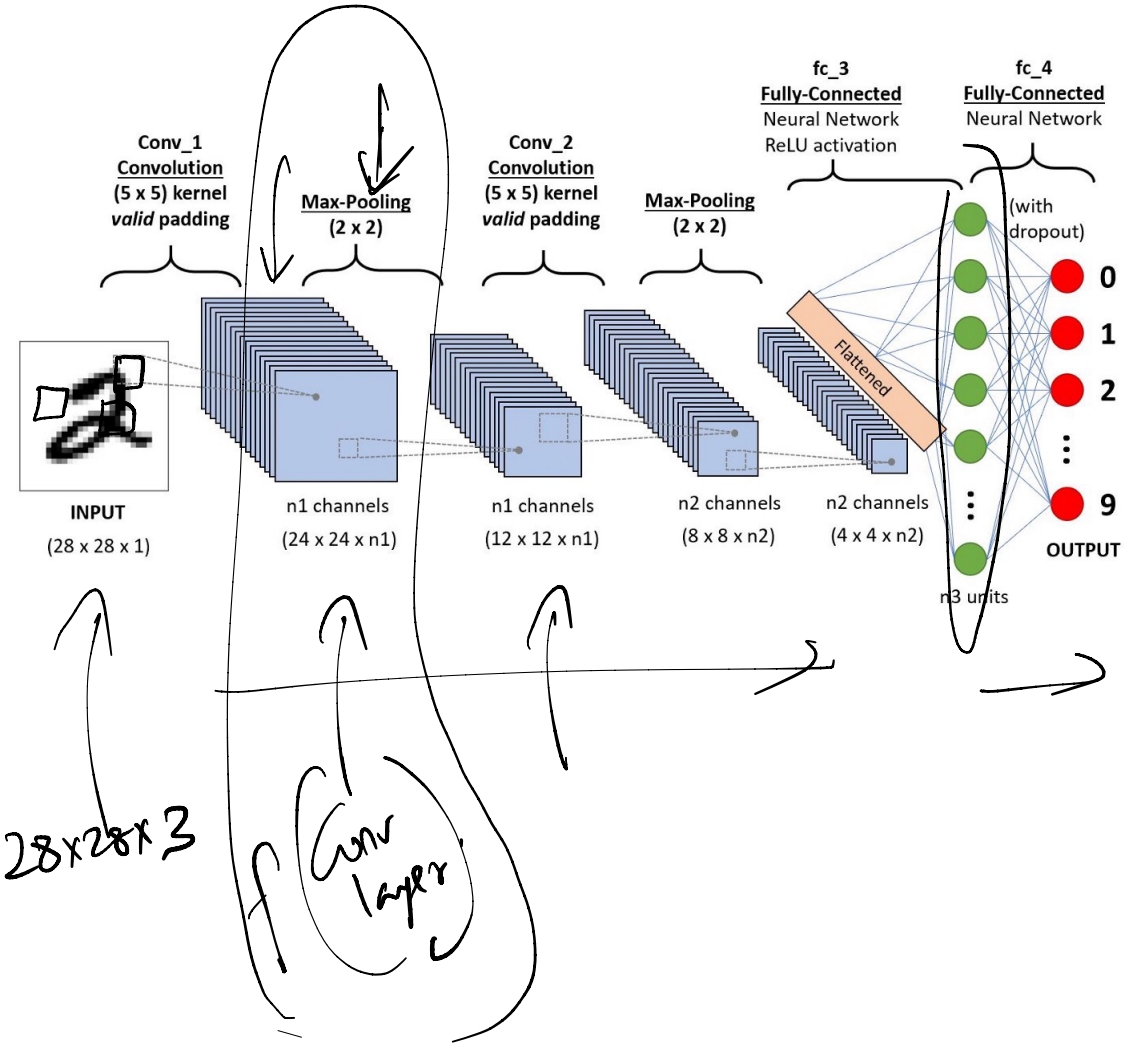
feed-forward.

$$\text{loss} = \frac{1}{2}(o - y_{(i)})^2$$

Backpropagation Algorithm

$$\frac{\partial \text{loss}}{\partial w_2} = \frac{\partial \text{loss}}{\partial o} \frac{\partial o}{\partial h_2} \frac{\partial h_2}{\partial w_2}$$

Convolutional Neural Networks


Conv_1
Convolution
(5 x 5) kernel
valid padding
Max-Pooling
(2 x 2)
Conv_2
Convolution
(5 x 5) kernel
valid padding
Max-Pooling
(2 x 2)
fc_3
Fully-Connected
Neural Network
ReLU activation
fc_4
Fully-Connected
Neural Network
(with
dropout)
Flattened
INPUT
(28 x 28 x 1)
n1 channels
(24 x 24 x n1)
n1 channels
(12 x 12 x n1)
n2 channels
(8 x 8 x n2)
n2 channels
(4 x 4 x n2)
n3 units
0
1
2
9
OUTPUT
28x28x3
Conv
layer

Tabular methods — linear FA — Non-linear FA

Off-Policy methods with FA:-

Off-Policy learning:-

goal: to learn a value for for a target policy π, given data due to a different behavior policy b.

Prediction: π, b are static and given.

Control: π, b are changing.

$\pi \rightarrow$ greedy poly wrt. $\hat{q}$.

$b \rightarrow \epsilon$-greedy.

2 challenges in off-Policy learning:

① target of the update [tabular, FA]

② distribution of updates [only in FA]

Semi-gradient methods:-

Per-step important Sampling ratio:

$$\rho_t = \rho_{t:t} = \frac{\pi(A_t|S_t)}{b(A_t|S_t)}$$

Semi-gradient off-Policy TD:-

-one-step, State-val. algo

$$w_{t+1} = w_t + \alpha \rho_t \delta_t \nabla \hat{v}(S_t; w_t)$$

where $\delta_t = R_{t+1} + \gamma \hat{v}(S_{t+1}; w_t) - \hat{v}(S_t; w_t)$

Semi-gradient expected Sarsa :-

— one-step, action-value algo

$$\omega_{t+1} = \omega_t + \alpha \, \delta_t \, \nabla \hat{q}(S_t, A_t ; \omega_t)$$

where $\delta_t = R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) \, \hat{q}(S_{t+1}, a ; \omega_t) - \hat{q}(S_t, A_t ; \omega_t)$

---

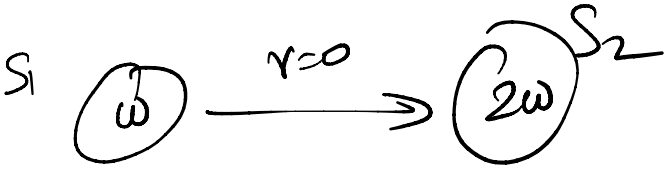
n-step semi-grad. expected Sarsa :-

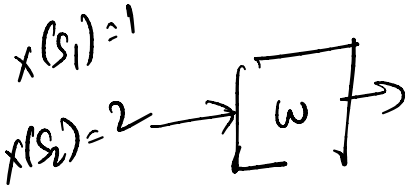
$$\omega_{t+n} = \omega_{t+n-1} + \alpha \, \rho_{t+1} \rho_{t+2} \cdots \rho_{t+n-1} \left[G_{t:t+n} - \hat{q}(S_t, A_t ; \omega_{t+n-1}) \right] \nabla \hat{q}(S_t, A_t ; \omega_{t+n-1})$$

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \hat{q}(S_{t+n}, A_{t+n} ; \omega_{t+n-1})$$

Examples for off-policy divergence :-

Example 1 :





$w = 10.$

$\hat{v}(s_1; w) = 10 \qquad \hat{v}(s_2; w) = 20$

$\gamma \sim 1 \qquad$ TD error $\sim 10 \qquad \alpha = 0.1$

$w = 11$

$\hat{v}(s_1; w) = 11 \qquad \hat{v}(s_2; w) = 22$

TD error $\sim 11 \qquad w \sim 12.1$

TD error :

$$\delta_t = R_{t+1} + \gamma \hat{v}(S_{t+1}; w_t) - \hat{v}(s_t; w_t)$$

$$= 0 + \gamma 2 w_t - w_t$$

$$\delta_t = (2\gamma - 1) w_t$$

off-Policy Semi-grad. TD update:

$$w_{t+1} = w_t + \alpha \rho_t \delta_t \nabla \hat{v}(s_t, w_t)$$

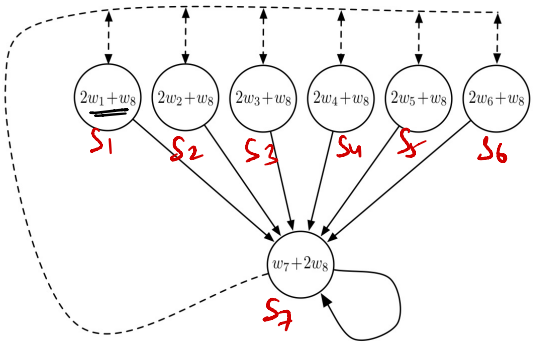
$$= w_t + \alpha \cdot 1 \cdot (2\gamma - 1) w_t \cdot 1$$

$$w_{t+1} = \left(1 + \alpha(2\gamma - 1)\right) w_t$$

if $\left(1 + \alpha(2\gamma - 1)\right) > 1$

$\gamma > 0.5$

Ex 2: Baird's counter example.



$\pi(\text{solid}|\cdot) = 1$

$b(\text{dashed}|\cdot) = 6/7$

$b(\text{solid}|\cdot) = 1/7$

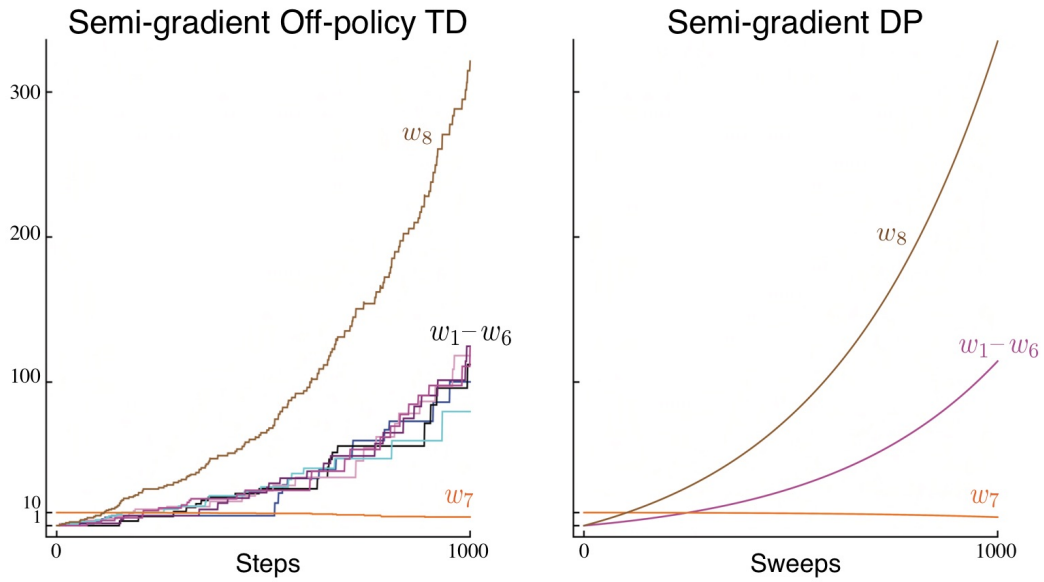
$\gamma = 0.99$

Reward is 0 on all transitions.

$w \in \mathbb{R}^8$ $\quad s_1 = 2w_1 + w_8$

$x(1) = (2, 0, 0, 0, 0, 0, 0, 1)^T$

$v_\pi(s) = 0$ for all $s \Rightarrow w = 0$



Semi-gradient DP

$$w_{k+1} = w_k + \frac{\alpha}{|S|} \sum_s \left(\mathbb{E}_\pi \left[R_{t+1} + \gamma \hat{v}(S_{t+1}, w_k) \mid S_t = s \right] - \hat{v}(s, w_k) \right) \nabla \hat{v}(s, w_k)$$

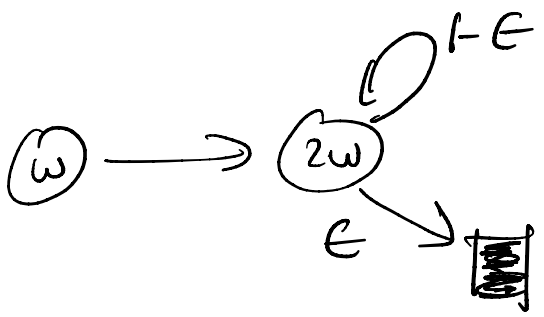
FA, Bootstrapping, off-Policy training.

→ Stochastic ? X

→ asynch. ? X

→ one-step return ? X

Ex 3: Tsitsiklis & Van Roy:



$$w_{k+1} = \underset{w \in \mathbb{R}}{\operatorname{argmin}} \sum_{s \in S} \left[\hat{v}(s;w) - \mathbb{E}_{\pi}\left[R_{t+1} + \gamma \hat{v}(S_{t+1}; w_k) \right] \right]^2$$

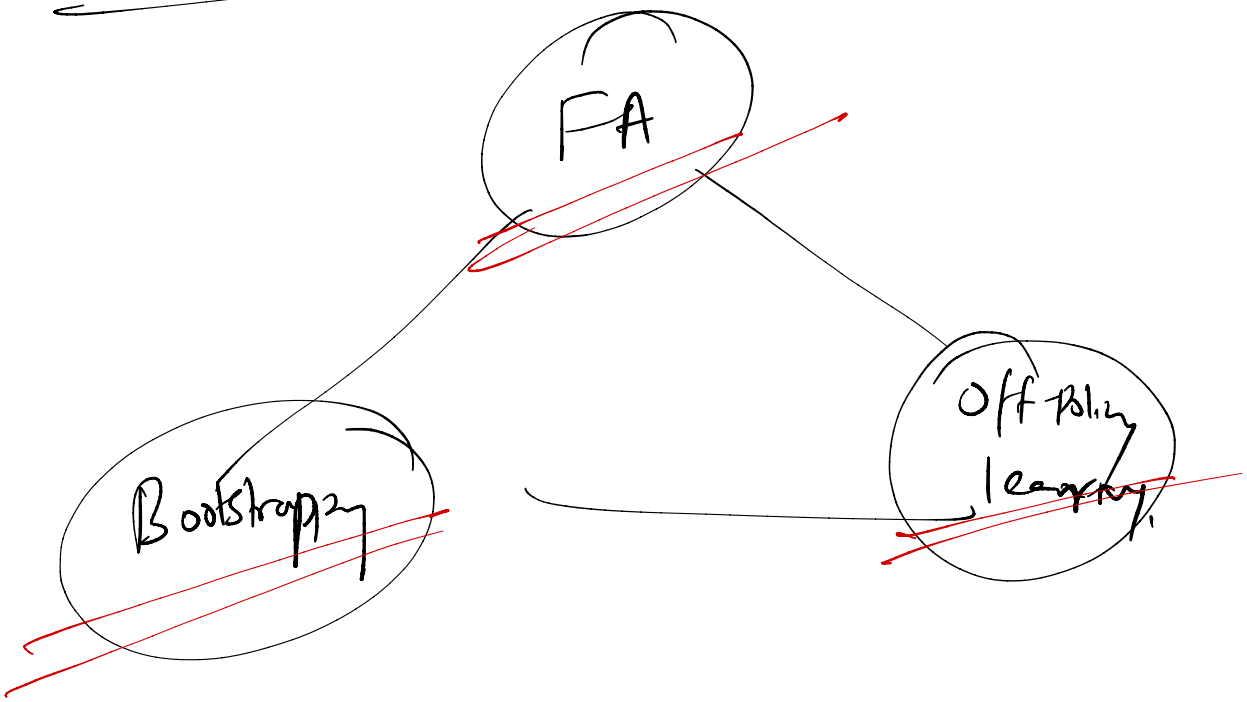
$$= \underset{w \in \mathbb{R}}{\operatorname{argmin}} \left(w - \gamma 2 w_k \right)^2 + \left(2w - (1-\epsilon)\gamma 2 w_k \right)^2$$

Quadratic

$$= \frac{6 - 4\epsilon}{5} \gamma w_k.$$

$\{w_k\} \longrightarrow$ diverge when $\gamma > \frac{5}{6-4\epsilon}$ and $w_0 \neq 0$.

the deadly triad:



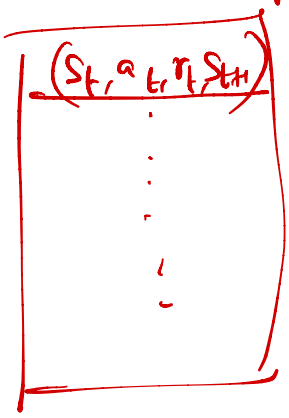
On/off-Policy	Algo	Tabular	linear FA	Non-linear FA
on-policy	MC	✓	✓	✓
	TD	✓	✓	X
off-policy	MC	✓	✓	✓
	TD	✓	X	X

Deep Q-Networks (DQN):

Q learning + NN FA.

① Experience Replay:

DQN →



FIFO buffer.

→ data efficiency

→ breaks correlation & reduces the variance in updat

→ averaging the behav. distrib

② Target Network:

$$\delta_t = R_{t+1} + \gamma \max_a Q(S_{t+1}, a; \theta_t^-) - Q(S_t, A_t; \theta_t)$$

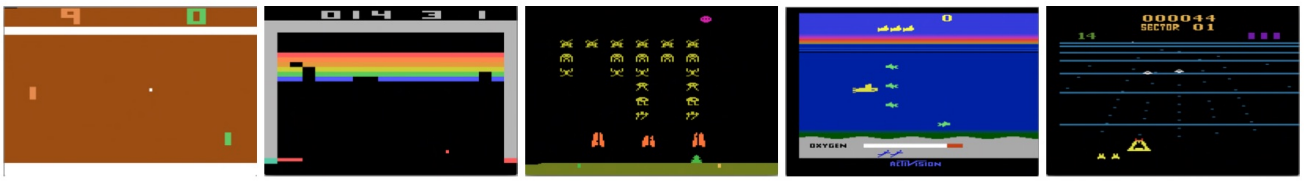
Current net. θ → θ_t

Target Nt θ^- → θ_t^-

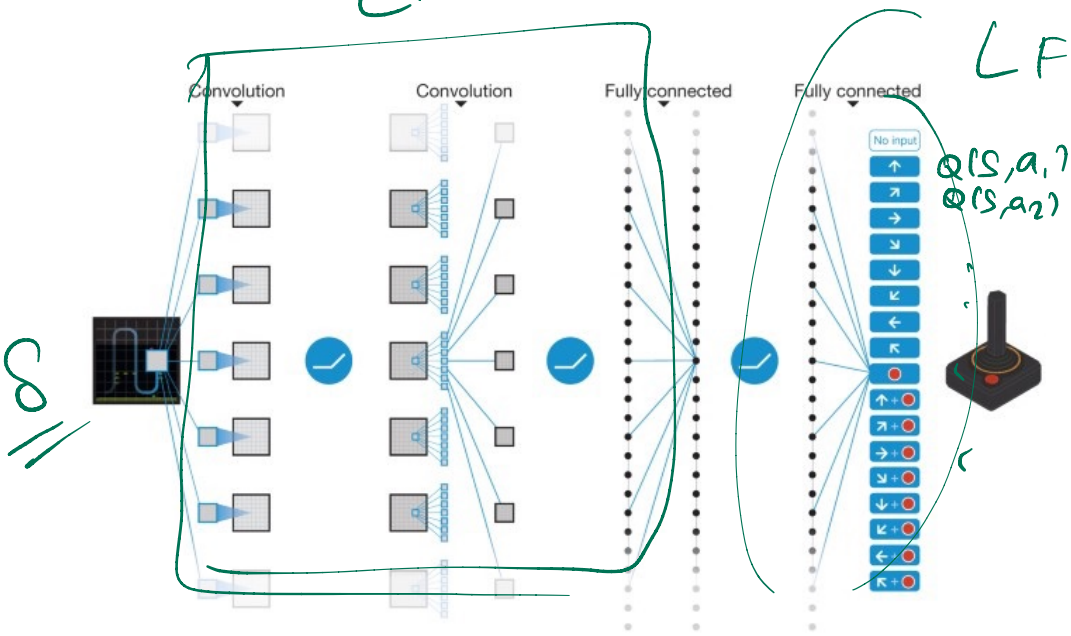
③ Reward clipping.

$(-1, 1)$

Atari :



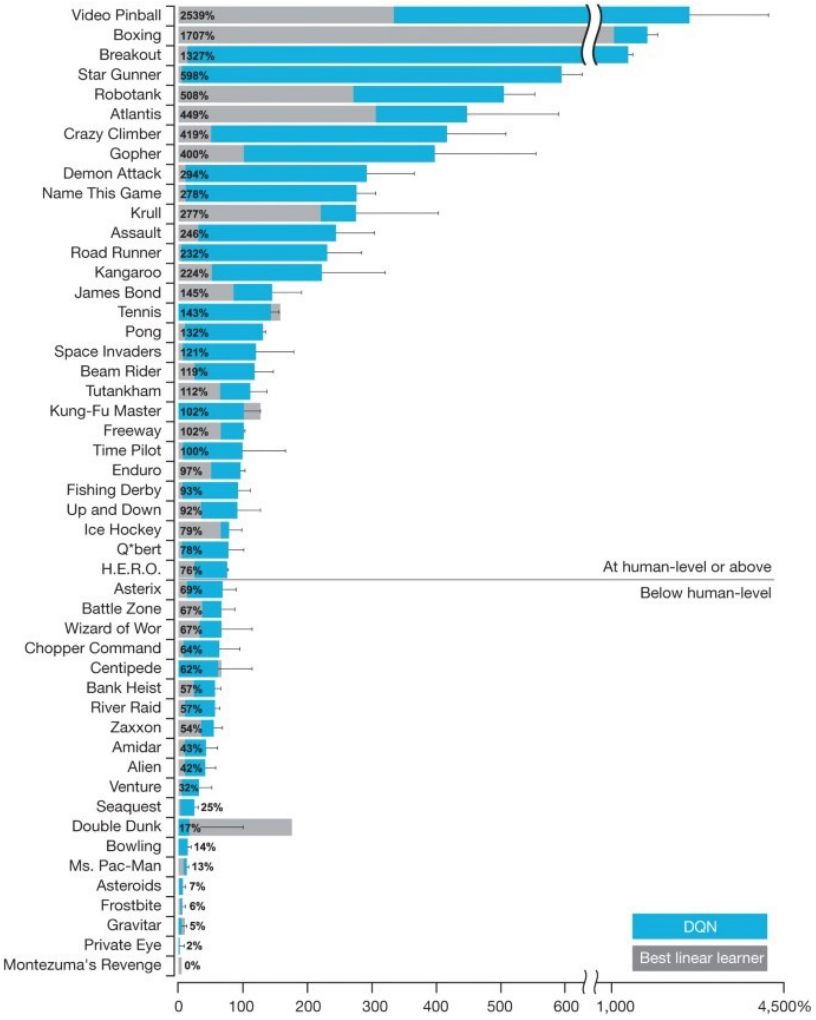
CNN

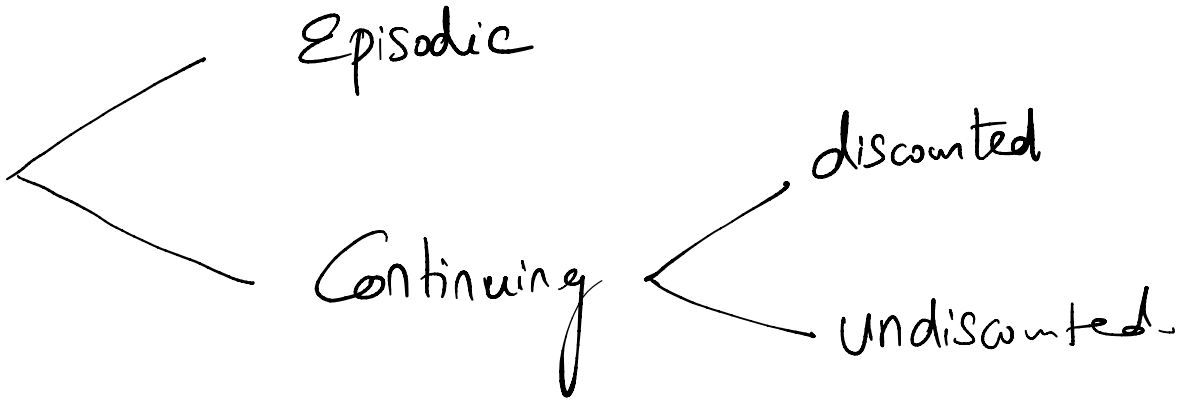


LFA:

Q(S,a,)
Q(S,a2)

S


Video Pinball 2539%
Boxing 1707%
Breakout 1327%
Star Gunner 598%
Robotank 508%
Atlantis 449%
Crazy Climber 419%
Gopher 400%
Demon Attack 294%
Name This Game 278%
Krull 277%
Assault 246%
Road Runner 232%
Kangaroo 224%
James Bond 145%
Tennis 143%
Pong 132%
Space Invaders 121%
Beam Rider 119%
Tutankham 112%
Kung-Fu Master 102%
Freeway 102%
Time Pilot 100%
Enduro 97%
Fishing Derby 93%
Up and Down 92%
Ice Hockey 79%
Q*bert 78%
H.E.R.O. 76%
Asterix 69%
Battle Zone 67%
Wizard of Wor 67%
Chopper Command 64%
Centipede 62%
Bank Heist 57%
River Raid 57%
Zaxxon 54%
Amidar 43%
Alien 42%
Venture 32%
Seaquest 25%
Double Dunk 17%
Bowling 14%
Ms. Pac-Man 13%
Asteroids 7%
Frostbite 6%
Gravitar 5%
Private Eye 2%
Montezuma's Revenge 0%
At human-level or above
Below human-level
DQN
Best linear learner
0
100
200
300
400
500
600
1,000
4,500%



Average reward setting:

$$r(\pi) = \lim_{h \to \infty} \frac{1}{h} \sum_{t=1}^{h} \mathbb{E}\left[R_t \mid S_0, A_{0:t-1} \sim \pi\right]$$

$$= \lim_{t \to \infty} \mathbb{E}\left[R_t \mid S_0, A_{0:t-1} \sim \pi\right]$$

$$= \sum_s \mu_\pi(s) \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a)\, r$$

$\mu_\pi(s)$ — steady state distrib.

$$\mu_\pi(s) = \lim_{t \to \infty} \Pr\{S_t = s \mid A_{0:t-1} \sim \pi\}$$

(Ergodic assumption)

Steady state dist

$$\sum_s \mu_\pi(s) \sum_a \pi(a|s) P(s'|s,a) = \mu_\pi(s')$$

Return

$$G_t = R_{t+1} - r(\pi) + R_{t+2} - r(\pi) + R_{t+3} - r(\pi) + \cdots$$

differential return.

Bellman eqn

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{rs'} P(s'r|s,a)\left[r - r(\pi) + v_\pi(s')\right]$$

$$q_\pi(s,a) = \sum_{rs'} P(s'r|s,a)\left[r - r(\pi) + \sum_{a'} \pi(a'|s') q_\pi(s',a')\right]$$

$$V_*(s) = \max_a \sum_{rs'} P(s'r|s,a)\left[r - \max_\pi r(\pi) + v_*(s')\right]$$

$$q_*(s,a) = \sum_{r,s'} P(s'r|s,a)\left[r - \max_\pi r(\pi) + \max_{a'} q_*(s',a')\right]$$

$$\delta_t = R_{t+1} - \bar{R}_{t+1} + \hat{v}(S_{t+1}; w_t) - \hat{v}(S_t; w_t)$$

Differential semi-gradient Sarsa for estimating $\hat{q} \approx q_*$

Input: a differentiable action-value function parameterization $\hat{q} : \mathcal{S} \times \mathcal{A} \times \mathbb{R}^d \to \mathbb{R}$
Algorithm parameters: step sizes $\alpha, \beta > 0$, small $\varepsilon > 0$
Initialize value-function weights $\mathbf{w} \in \mathbb{R}^d$ arbitrarily (e.g., $\mathbf{w} = \mathbf{0}$)
Initialize average reward estimate $\bar{R} \in \mathbb{R}$ arbitrarily (e.g., $\bar{R} = 0$)

Initialize state S, and action A
Loop for each step:
  Take action A, observe R, S'
  Choose A' as a function of $\hat{q}(S', \cdot, \mathbf{w})$ (e.g., ε-greedy)
  $\delta \leftarrow R - \bar{R} + \hat{q}(S', A', \mathbf{w}) - \hat{q}(S, A, \mathbf{w})$
  $\bar{R} \leftarrow \bar{R} + \beta\delta$
  $\mathbf{w} \leftarrow \mathbf{w} + \alpha\delta\nabla\hat{q}(S, A, \mathbf{w})$
  $S \leftarrow S'$
  $A \leftarrow A'$

Depreciating discounted setting :-

$\gamma(\pi)$ / $(1-\gamma)$

$$J(\pi) = \sum_s \mu_\pi(s)\, v_\pi^\gamma(s)$$

$$= \sum_s \mu_\pi(s) \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_\pi^\gamma(s')\right]$$

$$= r(\pi) + \sum_s \mu_\pi(s) \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\, \gamma v_\pi^\gamma(s')$$

$$= r(\pi) + \gamma \sum_{s'} v_\pi^\gamma(s') \sum_s \mu_\pi(s) \sum_a \pi(a|s) p(s'|s,a)$$

$$= r(\pi) + \gamma \sum_{s'} V_{\pi}^{\gamma}(s') \sum_{s} \mu_{\pi}(s) \sum_{a} \pi(a|s) p(s'|s,a)$$

$$= r(\pi) + \gamma \sum_{s'} V_{H}^{\gamma}(s') \mu_{\pi}(s')$$

$$= r(\pi) + \gamma J(\pi)$$

$$= r(\pi) + \gamma r(\pi) + \gamma^2 J(\pi)$$

$$= r(\pi) + \gamma r(\pi) + \gamma^2 r(\pi) + \cdots$$

$$= \frac{1}{1-\gamma} r(\pi)$$