General

$\mathrm{var}(AX) = A\,\mathrm{var}(X)A^T$

$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$

$P(X_1, .., X_n) = P(X_1)P(X_2|X_1)\cdots P(X_n|X_{1:n-1})$

Gaussian

$p(x) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}}\exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$

$\log p(x) = -\frac{1}{2}\log|\Sigma| - \frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu) + C$

Conditioning Gaussians

$X_A \mid X_B = x_B \sim \mathcal{N}(\mu_{A|B}, \Sigma_{A|B})$ where

$\mu_{A|B} = \mu_A + \Sigma_{AB}\Sigma_{BB}^{-1}(x_B - \mu_B)$,

$\Sigma_{A|B} = \Sigma_{AA} - \Sigma_{AB}\Sigma_{BB}^{-1}\Sigma_{BA}$.

KL Divergence

$KL(P\|Q) = \mathbb{E}_p[\log(\frac{p}{q})]$

If $P \sim \mathcal{N}(\mu_1, \Sigma_1)$ and $Q \sim \mathcal{N}(\mu_2, \Sigma_2)$, then

$KL(P\|Q) = \frac{1}{2}\Big(\mathrm{tr}\,\Sigma_2^{-1}\Sigma_1 + (\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1)$
$\quad - d + \log\frac{|\Sigma_2|}{|\Sigma_1|}\Big)$

Entropy

$H(p) = \mathbb{E}_p[-\log p]$

$X \sim \mathcal{N}(\mu, \Sigma):\ H(X) = \frac{1}{2}\log\left((2\pi e)^n|\Sigma|\right)$

Conditional Entropy

$H(X \mid Y) = \mathbb{E}_{p(x,y)}[-\log p(x \mid y)]$

$H(X, Y) = H(X) + H(Y \mid X)$

$H(S \mid T) \geq H(S \mid T, U)$

Mutual Information

$I(X;Y) = H(X) - H(X \mid Y) = I(Y;X) \geq 0$

$X \sim N(\mu, \Sigma),\ Y = X + \epsilon,\ \epsilon \sim N(0, \sigma^2 I)$

$\rightarrow I(X;Y) = \frac{1}{2}\log|I + \sigma^{-2}\Sigma|$

Bayesian Learning

Prediction:

$p(y|x, x_{1:n}, y_{1:n}) = \int p(y|x,\theta)p(\theta|x_{1:n}, y_{1:n})d\theta$

Convexity

$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$

f convex, g affine $\Rightarrow f \circ g$ convex

f non-decreasing, g convex $\Rightarrow f \circ g$ convex

Change of Variables

If $Y = g(X)$, then

$p_Y(y) = p_X(g^{-1}(y)) \cdot |\det Dg^{-1}(y)|$.

Complexity

Matrix mult. $A \in \mathbb{R}^{n\times k}, B \in \mathbb{R}^{k\times d}$ is $\Theta(nkd)$

1 Bayesian Linear Regression

$f = \mathbf{w}^T\mathbf{x},\ y = f + \epsilon,\ \epsilon \sim \mathcal{N}(0,\ \sigma_n^2)$

$p(\mathbf{w}) = \mathcal{N}(0,\ \sigma_p^2\mathbf{I})$

$p(\mathbf{w} \mid \mathbf{X}, \mathbf{y}) = \mathcal{N}(\overline{\mu},\ \overline{\Sigma})$, where

$\overline{\Sigma} = (\sigma_n^{-2}\mathbf{X}^T\mathbf{X} + \sigma_p^{-2}\mathbf{I})^{-1}$,

$\overline{\mu} = \sigma_n^{-2}\overline{\Sigma}\mathbf{X}^T\mathbf{y}$.

$p(f \mid \mathbf{X}, \mathbf{y}, \mathbf{x}) = \mathcal{N}(\mathbf{x}^T\overline{\mu},\ \mathbf{x}^T\overline{\Sigma}\mathbf{x})$

$p(y \mid \mathbf{X}, \mathbf{y}, \mathbf{x}) = \mathcal{N}(\mathbf{x}^T\overline{\mu},\ \mathbf{x}^T\overline{\Sigma}\mathbf{x} + \sigma_n^2)$

Epistemic: Uncertainty about model due to lack of data.

Aleatoric: Irreducible noise.

Recursive updates:

$\mathbf{X}_{t+1}^T\mathbf{X}_{t+1} = \mathbf{X}_t^T\mathbf{X}_t + x_{t+1}x_{t+1}^T$

$\mathbf{X}_{t+1}^T y_{t+1} = \mathbf{X}_t^T y_t + y_{t+1}x_{t+1}$

2 Bayesian Logistic Regression

$p(y_i \mid x_i, \theta) = \sigma(y_i w^T x_i) = \frac{1}{1+e^{-y_i w^T x_i}}$

3 Gaussian Processes

Process s.t. $\forall A \subseteq \mathcal{X}, A = \{x_1, .., x_m\}$ it holds $X_A = [X_{x_1}, .., X_{x_m}] \sim \mathcal{N}(\mu_A, K_{AA})$ where

$\mu_A^{(i)} = \mu(x_i)$ and $K_{AA}^{(ij)} = k(x_i, x_j)$

Kernel symmetric PSD

stationary if $k(x, x') = k(x - x')$

isotropic if $k(x, x') = k(\|x - x'\|_2)$

RBF: smooth

Exponential: cont. & nowhere differentiable

Matern: $\lceil\nu\rceil$-times differentiable

Prediction

$y_i = f(x_i) + \epsilon_i,\ \epsilon_i \sim \mathcal{N}(0, \sigma^2),\ A = \{x_1, .., x_m\}$.

Then $f \mid x_{1:m}, y_{1:m} \sim GP(\mu', k')$ where

$\mu'(x) = \mu(x) + \mathbf{k}_{x,A}(\mathbf{K}_{AA} + \sigma^2\mathbf{I})^{-1}(\mathbf{y}_A - \mu_A)$

$k'(x, x') = k(x, x') - \mathbf{k}_{x,A}(\mathbf{K}_{AA} + \sigma^2\mathbf{I})^{-1}\mathbf{k}_{x',A}^T$

$k_{x,A} = [k(x, x_1), \dots, k(x, x_m)]$

Predictive posterior:

$y^* \mid x_{1:m}, y_{1:m}, x^* \sim \mathcal{N}(\mu'(x^*), \sigma^2 + k'(x^*, x^*))$

Model selection

Marginal likelihood maximization

$\hat{\theta} = \arg\max_\theta p(y \mid X, \theta)$

$= \arg\max_\theta \int p(y \mid X, f)p(f \mid \theta)df$

Accelerating GPs

GP prediction has cost $\mathcal{O}(|A|^3)$

Kernel approximation: Find ϕ s.t. $k(x, x') \approx \phi(x)^T\phi(x')$, then do BLR

RFF: Stationary k has Fourier transf.:

$k(x, x') = \int_{\mathbb{R}^d} p(\omega)e^{j\omega^T(x-x')}d\omega$

$\approx \frac{1}{m}\sum_i z_{w^{(i)},b^{(i)}}(x)z_{w^{(i)},b^{(i)}}(x')$

$\rightarrow$ Set $\phi_i(x) = \frac{1}{\sqrt{m}}z_{w^{(i)},b^{(i)}}(x)$ where

$\omega \sim p(\omega), b \sim \mathcal{U}[0, 2\pi]$,

$z_{\omega,b}(x) = \sqrt{2}\cos(\omega^T x + b)$

4 Approximative Inference

Laplace Approximation

$p(\theta \mid y_{1:n}) \approx \mathcal{N}(\hat{\theta}, \Lambda^{-1}) =: q(\theta)$

$\hat{\theta} = \arg\max_\theta p(\theta \mid y),\ \Lambda = -\nabla^2\log p(\hat{\theta} \mid y)$

Prediction:

$p(y^* \mid x^*, x_{1:n}, y_{1:n}) \approx \int p(y^* \mid f^*)q(f^*)df^*$,

with $q(f^*) = \int p(f^* \mid \theta)q(\theta)d\theta$.

Variational Inference

$p(\theta \mid y) = \frac{1}{Z}p(\theta, y) \approx q_\lambda(\theta)$

$q_{bwd}^* \in \arg\min_q KL(q\|p)$: $q \approx p$ where q large

$q_{fwd}^* \in \arg\min_q KL(p\|q)$: $q \approx p$ where p large

$\arg\min_q KL(q\|p)$

$= \arg\max_q \mathbb{E}_{\theta\sim q}[\log p(y, \theta)] + H(q)$

$= \arg\max_q \mathbb{E}_{\theta\sim q}[\log p(y \mid \theta)] - KL(q\|p(\theta))$

$\leq \log p(y)$ (using Jensen)

5 MCMC

Approximate predictive distribution

$p(y^* \mid x^*, x_{1:n}, y_{1:n})$

$= \int p(y^* \mid x^*, \theta)p(\theta \mid (x,y)_{1:n})d\theta$

$= \mathbb{E}_{\theta\sim p(\theta|(x,y)_{1:n})}[f(\theta)] \approx \frac{1}{m}\sum_{i=1}^m f(\theta^{(i)})$,

with samples $\theta^{(i)} \sim p(\theta \mid (x,y)_{1:n})$ from MC with stationary distribution $p(\theta \mid (x,y)_{1:n})$.

Markov Chains

MC is ergodic if $\exists\ t$ s.t. every state is reachable from every state in *exactly* t steps.

A stationary ergodic MC has a unique and positive stationary distr. π, i.e. $\forall x$: $\lim_{t\to\infty} P(X_t = x) = \pi(x)$.

If MC satisfies **detailed balance**

i.e. $\forall x, x' : Q(x)P(x' \mid x) = Q(x')P(x \mid x')$,

then $\pi(x) = \frac{1}{Z}Q(x)$.

Hoeffding

If $f \in [0, C]$ and x_i are iid samples of X, then

$P(|\mathbb{E}f(X) - \frac{1}{N}\sum_{i=1}^N f(x_i)| > \epsilon) \leq 2exp(-2N\epsilon^2/C^2)$

Gibbs Sampling

Init x^0, fix observed RVs X_B to x_B

For $t = 1, ..$: Set $x^t = x^{t-1}$ and select $j \in [m] \setminus B$.

Update x_j^t by sampling from $P(X_j \mid x_{-i}^t)$.

Metropolis-Hastings

Needs proposal distr. $R(X' \mid X)$.

For $t = 1, ..$:

1) Sample $x \sim R(X' \mid X = x^{t-1})$

2) Set $x^t = \begin{cases} x, \text{with prob.} \min\{1, \frac{Q(x')R(x|x')}{Q(x)R(x'|x)}\} \\ x^{t-1}, \text{else} \end{cases}$.

MALA/LMC

MH with $R(x' \mid x) = \mathcal{N}(x - \tau\nabla f(x), 2\tau I)$.

SGLD

MALA with subsampling of data for gradient computation of the energy function.

6 Bayesian Neural Networks

MAP

$\hat{\theta} = \arg\max_\theta \log p(\theta) + \sum_i \log p(y_i|x_i, \theta)$

Variational Inference

SGD on ELBO to find approximate posterior q_λ. Draw samples $\theta^j \sim q_\lambda$ and approximate $p(y^* \mid x^*, x_{1:n}, y_{1:n}) \approx \frac{1}{m}\sum_j p(y^* \mid x^*, \theta^j)$.

7 Active Learning

Collect data maximally reducing uncertainty. Find $S \subseteq D$ maximizing mutual information

$I(f; y_S) = H(f) - H(f \mid y_S) \overset{G.P.}{=} \frac{1}{2}\log|I + \sigma^{-2}K_S|$.

Greedy Optimization

Given $S_t = \{x_1, .., x_t\}$, take

$x_{t+1} = \arg\max_x F(S_t + x) \overset{G.P.}{=} \arg\max_x \sigma_t^2(x)$.

Uncertainty Sampling:

$x_{t+1} = \arg\max_x \sigma_t^2(x)$

Heteroscedastic Noise:

$x_{t+1} \arg\max_x \sigma_t^2(x)/\sigma_n^2(x)$

8 Bayesian Optimization

Sequentially pick $x_1, .., x_T \in D$, get $y_t = f(x_t) + \epsilon_t$, find $\max_x f(x)$.

Cumulative Regret

$R_T = \sum_{t=1}^T \max_{x\in D} f(x) - f(x_t)$

GP-UCB

$x_t = \arg\max_{x\in D} \mu_{t-1}(x) + \beta_t\sigma_{t-1}(x)$

(upper confidence bound $\geq$ best lower bound)

Thm: $f \sim GP$, correct $\beta_t : \frac{1}{T}R_T = \mathcal{O}^*(\sqrt{\gamma_T/T})$,

$\gamma_T = \max_{|S|\leq T} I(f; y_S)$ (information gain).

EI

choose $x_t = \arg\max_x EI(x)$ where

$EI(x) = \int \max(0, y^* - y)p(y \mid x)dy$.

Thompson sampling

draw sample from GP $f \mid (x,y)_{1:t}$, select $x_{t+1} \in \arg\max_{x\in D}\tilde{f}(x)$.

9 Markov Decision Processes

A MDP is defined by States $X = \{1, .., n\}$, Actions $A = \{1, .., m\}$, Transition probabilities $P(x' \mid x, a)$, Reward function $r(x, a)$.

Policy

$\pi : X \rightarrow A$ or $\pi(a \mid x)$

Action-Value Function

$Q^\pi(x, a) = r(x, a) + \gamma\sum_{x'} P(x' \mid x, a)V^\pi(x')$

Value function

$V^\pi(x) = J(\pi \mid X_0 = x) = Q^\pi(x, \pi(x))$

$\quad = \mathbb{E}[\sum_{t=0}^\infty \gamma^t r(X_t, \pi(X_t)) \mid X_0 = x]$

$\overset{\pi det.}{=} r(x, \pi(x)) + \gamma\sum_{x'} P(x' \mid x, \pi(x))V^\pi(x')$

$\overset{\pi rand.}{=} \sum_a \pi(a|x)\Big[r(x, a) + \gamma\sum_{x'} P(x' \mid x, a)V^\pi(x')\Big]$

$\Leftrightarrow V^\pi = (I - \gamma T^\pi)^{-1}r^\pi$ with $V_i^\pi = V^\pi(i)$,

$r_i^\pi = r^\pi(i, \pi(i))$ and $T_{i,j}^\pi = P(j \mid i, \pi(i))$.

Fixed Point Iteration
Init V_0^π. For $t=1,..$ do:
$V_t^\pi = r^\pi + \gamma T^\pi V_{t-1}^\pi$ (contraction)

Greedy Policy w.r.t. V
V induces greedy policy $\pi_V(x) = \arg\max_a r(x,a) + \gamma \sum_{x'} P(x' \mid x,a) V(x')$
Thm: (Bellman) Optimal policy is greedy wrt. its own value function.

Policy Iteration
Init arbitrary policy π. Until converged: Compute $V^\pi(x)$; compute greedy policy π_G w.r.t. V^π; set $\pi \leftarrow \pi_G$.
PI monotonically improves all values $V^{\pi_{t+1}}(x) \geq V^{\pi_t}(x) \forall x$. Converges to exact solution in $\mathcal{O}(n^2 m/(1-\gamma))$ iterations.

Value Iteration
Init $V_0(x) = \max_a r(x,a)$. For $t=1,..$:
Set $Q_t(x,a) = r(x,a) + \gamma \sum_{x'} P(x' \mid x,a) V_{t-1}(x')$ and $V_t(x) = \max_a Q_t(x,a)$. Stop if $\|V_t - V_{t-1}\|_\infty \leq \epsilon$, then choose greedy policy w.r.t. V_t. Converges to ϵ-optimal policy in polynomially many iterations.

10 POMDP
Obtain only noisy observations Y_t of state X_t.
Solve by modelling $P(X_t \mid y_{1:t})$.

Belief States
POMDP as MDP where states $\equiv$ beliefs $P(X_t \mid y_{1:t})$ in the orig. POMDP.
Actions $\mathcal{A} = \{1,..,m\}$, Transitions: $P(Y_{t+1} = y \mid b_t, a_t) = \sum_{x,x'} b_t(x) P(x' \mid x, a_t) P(y \mid x')$; $b_{t+1}(x') = \frac{1}{Z} \sum_x b_t(x) P(X_{t+1} = x' \mid X_t = x, a_t) P(y_{t+1} \mid x')$
Reward: $r(b_t, a_t) = \sum_x b_t(x) r(x, a_t)$

11 Reinforcement Learning
Planning in unknown MDP.
- On-policy: agent has full control (actions)
- Off-policy: no control, only observational data

12 Model-Based RL
Learn MDP and use optimal π.
MLE estimate from path trajectory τ:
$P(X_{t+1} \mid X_t, A) \approx \frac{Cnt(X_{t+1}, X_t, A)}{Cnt(X_t, A)}$; $r(x,a) \approx N_{x,a}^{-1} \sum_{t: X_t = x, A_t = a} R_t$

ϵ_t-greedy:
Tradeoff exploration-exploitation W.p. ϵ_t: rand. action; w.p. $1-\epsilon_t$: best action. If ϵ_t satisfies RM $\implies$ converge to π^* w.p. 1.

Robbins-Monro (RM) $\sum_t \epsilon_t = \infty$, $\sum_t \epsilon_t^2 < \infty$

R_{max}-Algorithm
Assume $r(x,a) \in [0, R_{max}]$. Set unknown $r(x,a)$ to R_{max}, add fairy tale state x^*, set $P(x^* \mid x,a) = 1$, compute π. Repeat: run π while updating $r(x,a)$, $P(x' \mid x,a)$, then recompute π.
Thm: W.p. $1-\delta$, R_{max} will reach ϵ-opt policy in #steps polynomial in $|X|, |A|, T, 1/\epsilon, \log(1-\delta), R_{max}$.
Note: MDP is assumed ergodic.

Problems of Model-based RL:
- Memory required: $P(x' \mid x,a) \approx \mathcal{O}(|X|^2 |A|)$, $r(x,a) \approx \mathcal{O}(|X||A|)$
- Computation: repeatedly solve MDP

13 Model-Free RL
Directly estimate value function

TD-Learning (On)
Follow π, get (x,a,r,x').
$\hat{V}^\pi(x) \leftarrow (1-\alpha_t)\hat{V}^\pi(x) + \alpha_t(r + \gamma \hat{V}^\pi(x'))$
Thm: If α_t satisfies RM and all (x,a) are chosen infinitely often, then $\hat{V}$ converges to V^π a.s.

Q-learning (Off)
Init $Q(x,a) = \frac{R_{max}}{1-\gamma} \prod_{t=1}^{T_{init}} (1-\alpha_t)^{-1}$.
Pick a (e.g. ϵ_t greedy), get (x,a,r,x'): $Q(x,a) \leftarrow (1-\alpha_t) Q(x,a) + \alpha_t (r + \gamma \max_{a'} Q(x',a'))$
At convergence $\pi_G(x) = \arg\max_a Q(x,a)$.
Thm: If α_t satisfies RM and all (x,a) are chosen infinitely often, then Q converges to Q^* a.s. Same PAC guarantee as for R_{max}.
Computation time: $\mathcal{O}(|A|)$, Memory: $\mathcal{O}(|X||A|)$

14 RL via Function Approximation
Learn approximation of (action-)value function $V(x;\theta)$, $Q(x,a;\theta)$.

TD-learning as SGD
Tabular TD update is equivalent to SGD on the loss $l = \frac{1}{2}(V(x;\theta) - r - \gamma V(x';\theta_{old}))^2$.

Parametric Q-learning (Off)
SGD on the loss
$l = \frac{1}{2}(Q(x,a;\theta) - r - \gamma \max_{a'} Q(x',a';\theta))^2$.
DQN: Q-learning with NN as func. approx.
Use experience replay data D, cloned network to maintain constant NN across episode.
$L(\theta) = \sum_{(x,a,r,x') \in D} (r + \gamma \max_{a'} Q(x',a';\theta^{old}) - Q(x,a;\theta))^2$
Double DQN: Use new NN to evaluate $\max_a$; prevents maximization bias.
$L^{DDQN}(\theta) = \sum_{(x,a,r,x') \in D} [r + \gamma \max_{a'} Q(x',a^*(\theta);\theta^{old}) - Q(x,a;\theta)]^2$, $a^*(\theta) = \arg\max_{a'} Q(x',a';\theta)$
Finding $a_t = \arg\max_a Q(x_t,a;\theta)$ is intractable for $|A|$ large. In gradient, a^* is ignored.

15 Policy Gradient Methods
Maximize $J(\theta) = \mathbb{E}_{\pi_\theta}[\sum_{t=0}^T \gamma^t r(x_t,a_t)]$ by SGD.
$\nabla_\theta J(\theta) = \nabla_\theta \mathbb{E}_{\tau \sim \pi_\theta} r(\tau) = \mathbb{E}_{\tau \sim \pi_\theta}[r(\tau) \nabla_\theta \log \pi_\theta(\tau)]$
MDP $(\tau = (x,a,r,x')_{1:T})$: $\pi_\theta(\tau)$
$= p(x_0) \prod_{t=0}^T \pi(a_t \mid x_t;\theta) p(x_{t+1} \mid x_t, a_t)$
$\rightarrow \nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[r(\tau) \sum_{t=0}^T \nabla_\theta \log \pi(a_t \mid x_t;\theta)]$
Can reduce variance via baselines:
$\mathbb{E}_{\tau \sim \pi_\theta}[r(\tau) \nabla \log \pi_\theta(\tau)] = \mathbb{E}_{\tau \sim \pi_\theta}[(r(\tau) - b) \nabla \log \pi_\theta(\tau)]$
E.g. $\nabla J_T(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^T \gamma^t G_t \nabla_\theta \log \pi(a_t \mid x_t;\theta)]$
with $G_t = \sum_{m=0}^{T-t} \gamma^m r_{t+m}$.

REINFORCE (On)
Init $\pi(a \mid x;\theta)$. Repeat:
Generate episode $(x_i, a_i, r_i)_{i=0}^T$.
Compute G_t, update θ:
$\theta \leftarrow \theta + \eta \sum_{t=0}^T \gamma^t G_t \nabla_\theta \log \pi(a_t \mid x_t;\theta)$

Advantage Function
$A^\pi(x,a) = Q^\pi(x,a) - V^\pi(x)$
$\forall x, a, \pi: A^{\pi^*}(x,a) \leq 0$; $\max_a A^\pi(x,a) \geq 0$

16 Actor-Critic
Approximate both Q^π (or V^π) *and* π_θ.
Reinterpret score gradient: $\nabla_{\theta_\pi} J(\theta_\pi) = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{t=0}^\infty \gamma^t Q(x_t,a_t;\theta_Q) \nabla \log \pi(a_t \mid x_t;\theta_\pi)]$
$= \mathbb{E}_{(x,a) \sim \pi_\theta}[Q(x,a;\theta_Q) \nabla \log \pi(a \mid x;\theta_\pi)]$
Allows online updates:
$\theta_\pi \leftarrow \theta_\pi + \eta_t Q(x,a;\theta_Q) \nabla \log \pi(a \mid x;\theta_\pi)$
$\theta_Q \leftarrow \theta_Q - \eta_t \delta \nabla Q(x,a;\theta_Q)$
Variance reduction: replace with $Q(x,a;\theta_Q) - V(x;\theta_V) \rightarrow$ A2C.

17 Off-Policy Actor Critic
Replace $\max_{a'} Q(x',a';\theta^{old})$ in DQN loss by $\pi(x';\theta_\pi)$, where π should follow the greedy policy to model $\max_{a'}$. This is equivalent to:
$\theta_\pi^* \in \arg\max_\theta \mathbb{E}_{x \sim \mu}[Q(x, \pi(x;\theta);\theta_Q)]$,
where $\mu(x) > 0$ 'explores all states'.
Needs *deterministic* π. Inject additional action noise to encourage exploration.

Deep Deterministic Policy Gradient (DDPG)
Init θ_Q, θ_π. Repeat: Observe x, execute $a = \pi(x;\theta_\pi) + \epsilon$, observe r, x', store in D. If time to update: For some iterations: sample B from D, compute targets
$y = r + \gamma Q(x', \pi(x', \theta_\pi^{old}), \theta_Q^{old})$, update
Critic: $\theta_Q \leftarrow \theta_Q - \frac{\eta}{|B|} \sum_B \nabla (Q(x,a;\theta_Q) - y)^2$,
Actor: $\theta_\pi \leftarrow \theta_\pi + \frac{\eta}{|B|} \sum_B \nabla Q(x, \pi(x;\theta_\pi);\theta_Q)$,
Params: $\theta_j^{old} \leftarrow (1-\rho)\theta_j^{old} + \rho \theta_j$, $j \in \{\pi, Q\}$

Randomized policy DDPG: For Critic: sample $a' \sim \pi(x';\theta_\pi^{old})$ to get unbiased y estimates.
For Actor: consider $\nabla_{\theta_\pi} \mathbb{E}_{a \sim \pi(x;\theta_\pi)} Q(x,a;\theta_Q)$
Reparametrization trick: $a = \psi(x;\theta_\pi,\epsilon)$
$\nabla_{\theta_\pi} \mathbb{E}_{a \sim \pi_{\theta_\pi}} Q(x,a;\theta_Q) = \mathbb{E}_\epsilon \nabla_{\theta_\pi} Q(x, \psi(x;\theta_\pi,\epsilon);\theta_Q)$

18 Model-Based Deep RL
MPC (deterministic dynamics)
Given model $x_{t+1} = f(x_t, a_t)$, plan over finite horizon H. At each step t, maximize
$J_H(a_{t:t+H-1}) := \sum_{\tau=t:t+H-1} \gamma^{\tau-t} r_\tau(x_\tau(a_{t:\tau-1}), a_\tau)$
$x_\tau(a_{t:\tau-1}) = f(f(...(f(x_t,a_t),a_{t+1})..))$
then carry out a_t, then replan.
Optimize via gradient based methods (diff. r, f, cont. action) or via random shooting.

Random shooting
Sample $a_{t:t+H-1}^{(i)}$ and pick $\arg\max_i J_H(a_{t:t+H-1}^{(i)})$.

MPC with Value estimate
$J_H(a_{t:t+H-1}) := \sum_{\tau=t:t+H-1} \gamma^{\tau-t} r_\tau(x_\tau(a_{t:\tau-1}), a_\tau) + \gamma^H V(x_{t+H})$
$H=1 \rightarrow J_1(a_t) = Q(x_t,a_t)$; $\pi_G = \arg\max_a J_1(a)$

MPC (stochastic dynamics)
$\max_{a_{t:t+H-1}} \mathbb{E}_{x_{t+1:t+H}}[\sum_{\tau=t:t+H-1} \gamma^{\tau-t} r_\tau + \gamma^H V(x_{t+H}) \mid a_{t:t+H-1}]$

Parametrized policy
$J_H(\theta) = \mathbb{E}_{x_0 \sim \mu}[\sum_{\tau=0:H-1} \gamma^\tau r_\tau + \gamma^H Q(x_H, \pi(x_H,\theta)) \mid \theta]$
($H = 0 \Leftrightarrow$ DDPG obj.)

MPC (unknown dynamics)
follow π, learn f, r, Q off-policy from replay buf, replan π.
BUT: point estimates have poor performance, errors compound $\rightarrow$ use bayesian learning:
Model distribution over f (BNN, GP) and use (approximate) inference (exact, VI, MCMC,..).

Greedy exploitation for model-based RL:
1) $D = \{\}$, prior $P(f \mid \{\})$ 2) repeat: plan new π to maximize $\max_\pi \mathbb{E}_{f \sim P(\cdot \mid D)} J(\pi, f)$, rollout π, add new data to D, update posterior $P(f \mid D)$.

PETS algorithm:
Ensemble of NNs predicting cond. Gaussian transition distr., use MPC.

Thompson Sampling:
Like greedy but in 2) sample model $f \sim P(\cdot \mid D)$ and then $max_\pi J(\pi, f)$
Use epistemic noise to drive exploration.

Optimistic exploration:
Like greedy but in 2) $\max_\pi \max_{f \in M(D)} J(\pi, f)$; with $M(D)$ set of plausible models given D.