EEML Summer School – Day 1 01/07/19

Intro to Deep Learning – Razvan Pascanu (DeepMind)

- SGD intuition from Taylor expansion (1st order):

$$\arg\min_{\Delta\theta} L(\theta+\Delta\theta) \approx \arg\min_{\Delta\theta} \left[L(\theta) + \Delta\theta \frac{\partial L}{\partial\theta}\right] \quad \text{s.t. } \|\Delta\theta\| \le \epsilon$$

$$\Leftrightarrow \arg\min_{\Delta\theta} L(\theta) + \Delta\theta \frac{\partial L}{\partial\theta} + \lambda(\Delta\theta) I (\Delta\theta)^T$$

Lagrange M. rewrite
↳ form of learning rate!

- Black-Box opt. via Evolutionary algos ⇒ Mutation ⊕ Next Generation
  → allows for potentially global optimizers ⇒ problem of many params
  → More useful for hyperparams!

- Population-based Training ⇒ Train models in parallel / stop + evaluate afterwards only further develop best performing model
  → Between parametric and evolutionary optimization!

→ Parametric
↳ slow learning
↳ compression / fast exec

→ Non-Parametric
↳ Bad scaling / slow inference
↳ Fast learning

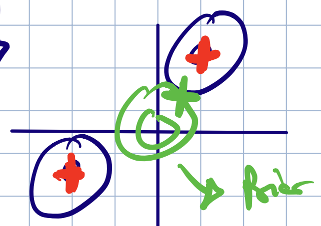
- $h: \mathcal{X} \to \mathcal{Y}$ ⇒ Predictor ⇔ Hypothesis
  (Classification / Regression / Structured Pred. (graph))
  ↳ $h: \Theta \times \mathcal{X} \to \mathcal{Y}$ ⇒ Fit Θ via Loss fct.
  ↳ ML / MAP perspective!

$$\arg\max_{\theta} P(\theta|D) = P(D|\theta)P(\theta)/P(D) \to \arg\max_{\theta} P(D|\theta)P(\theta)$$

→ Uniform prior in MAP formulation results in Max. Likelihood!
↳ together with iid assumption → split data into $\prod_{i=1}$ ⊕ log tricks

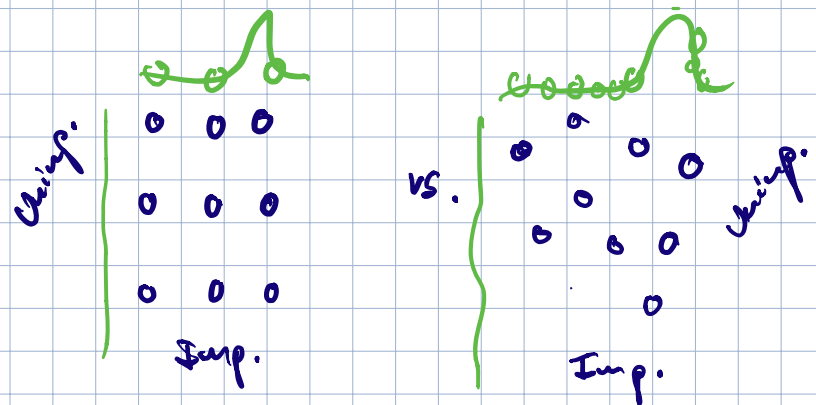
- Loss fct often times results from Bayesian / MLE perspective ⊕ assumptions on the data → E.g. MSE from MAP + Uniform Prior + Gaussian Data!
  ↳ Multi-Label Classification: Multinomial ⇔ neg. log LH.

- Regularize via not assuming $P(\Theta)$ being uniform! [Together w. Outward approx.]

→ 

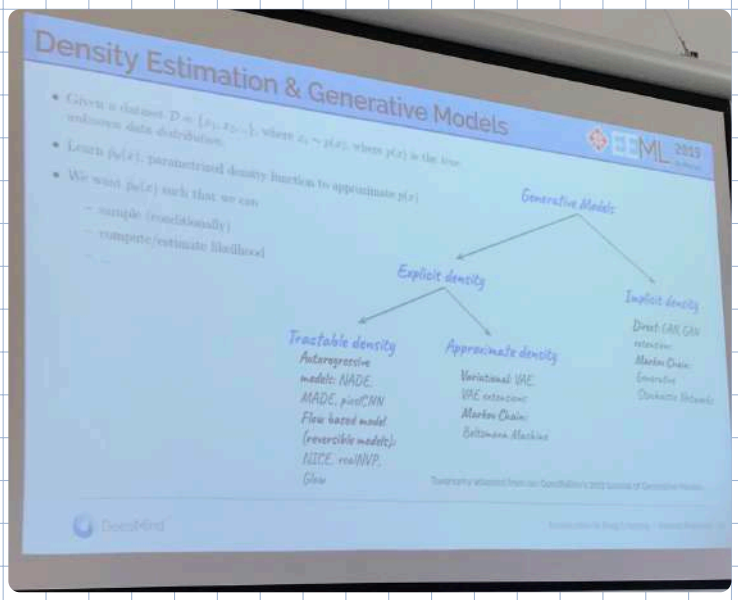
⇒ Reg. get rid of local min problem + by preferring one that is closer to 0 ⇒ +

- Hyperparameter Optimization: Random vs. Grid → Random covers more space!



↳ Be careful not to overfit test data!
↳ Unclear what global/local convergence is!

- Early-Stopping as a prior → Form of L_2 → See Bishop! Must Read?!
↳ Dumoulin et al 2016 - Early Stopping as Hyperparam. ✓

- 

Generative Models

→ Autoregressive: Decompose joint as product of conditionals
↳ Not necessarily clear how to do so if there is no temporal dim.
Eg. where to start sampling image
↳ Slow: Slow Sampling!

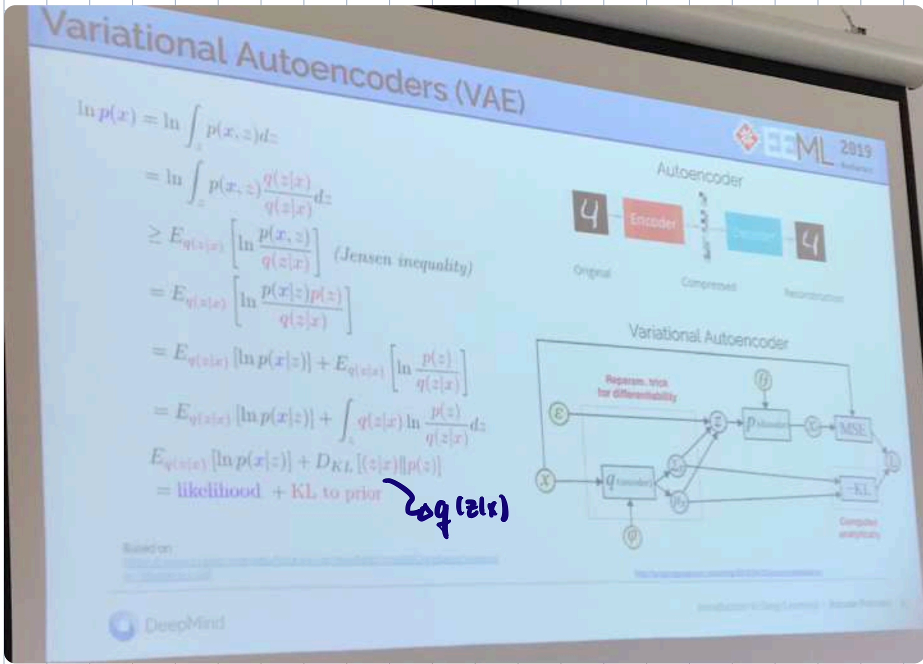
→ Flow-based / Reversible Models: Find $f_\theta^{-1}(z)$

→ Change of variables → Gaussian in latent space
↳ optimize easily and exactly

$$p_X(x) = p_Z(z) \left| \det \left(\frac{\partial x}{\partial z} \right) \right|^{-1}$$

↳ Not clear how to parametrize f_θ s.t. it is invertible!
↳ Easily sample based on sampling Gaussian z and then applying $f_\theta^{-1}(z)$

→ VAE → Restrict latent codes by having encoder output Gaussian parametrization → Optimize both Reconstruction as well as KL objective! ↳ Restriction allows for efficient sampling



↳ Gaussian comes in handy for KL computations as well as sampling

↳ Gaussian + Beta //

also new work on Gumbel Dist. and Multinomial

↳ Problem to get sharp color images!

↳ want VAE for latent interpolation!

→ Autoregressive models vs. GANs → Video GANs better, Audio AR

- ReLU activations: Splits space into linear regions → Can bound # → Learning
  - ↳ Montufar et al 2014 → NIPS
  - ↳ Form of optimal efficient folding of the space!
  - ↳ optimization such that decision boundary becomes simpler!
  - ↳ No overfitting: some regions are contracted to close points!
  - ⇒ Need high-dimensional data → Not necessarily better
  - ↳ 1d mixture of sinusoids hard to fit!

Not a lot known about restrictive properties!

- Loss Opt. Surfaces: Not necessarily crazy!
  - ↳ Yann LeCun: Simple random version of computer proto. of all dir. pointing up → very unlikely
  - ↳ Sharp vs. flat minima! → What do they mean? Lack of dependence
  - ↳ Can switch eigenvalue profile without changing function!

- RMSprop → Running Avg. Mean/Var. → larger step if small var
  - ↳ Approx. of Natural Gradient
  - ↳ Batch norm → Make curvature closer to identity → Fisher → NG!

- Convolution = Restriction of net → Need more inductive biases!
- Graph Nets = Do Conv not on locally defined pixels but on other local structure
- RNN → Exploding grad from product of Jacobians → Gradient Clipping
  ↳ LSTM → vanishing gradient combat via linear cell state
  But need to learn gates
  ↳ Interpret gates more as a form of low pass filters (power 0 or 1!)

Intro to RL - Doina Precup (DeepMind / McGill) → TD

- How to overcome sample efficiency problem with sparse rewards
  → self-play → curriculum acquisition // intrinsic motivation → surrogate reward
- Key features:
  * Trial-and-Error Search
  * Stochastic Env
  * Delayed reward → Temp. Credit Assignment
  * E-E Trade-off
- Learning with multiple output heads / auxiliary tasks
  ↳ Relationship to Ecol / bounded rationality ?!
- TD Gammon → (Tesauro et al, 1995) ⇒ "Early Alpha-Go"
- Imitation learning / behavioral cloning → Form of pretraining
  → Important to train in simulator with exploration to robustify!
- Different γ interpretations: Survival prob., bias-variance tradeoff, rightfulness
- MC methods ≈ Supervised → $V(S_t)_{t+1} \leftarrow V(S_t)_{t} + \alpha [G_t - V(S_t)_{t}]$
  → Update for each transition ⇒ not iid! Sequentially correlated
  ↳ Overcome via ER buffer for example!
- Bias - Variance Trade-Offs:
  * Function Approx. → Gradient Updates
  * Generalization across environments
  * Discount → Estimation of future rewards
  ↳ schedule of discount params ⇒ relate to TD as in PER?!

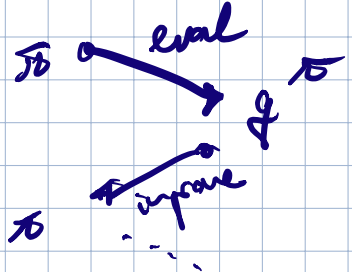
- Bellman Self-Consistency Equations: Set of linear equations!

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

  - But $v^*(s)$ is a non-linear set of equations due to argmax!
  - Can only be done if full MDP is known! ⇒ Estimate dynamics
  - Importance $\gamma < 1$: Contraction ⇒ Banach Fixed Point Theorem
  - Problem also with large state spaces ⇒ Storage is expensive!

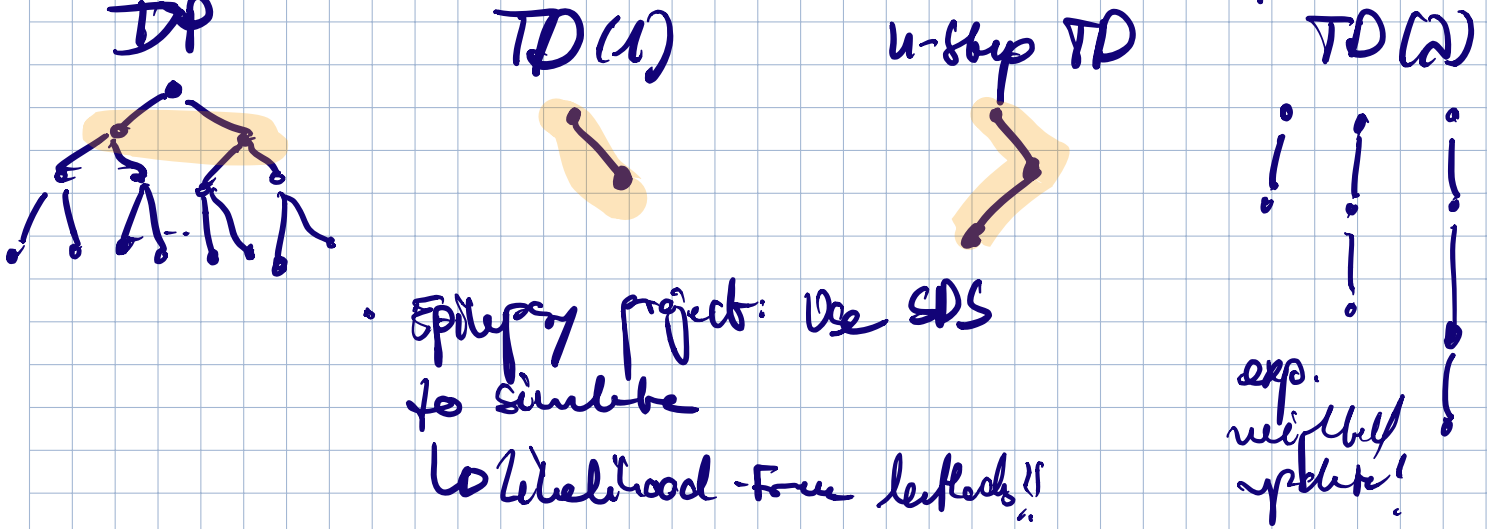
- Contraction is what makes bootstrapping work!
  - Would it make sense to start off with MC return backup and slowly move towards full bootstrap ≥ k-step where $k = T \rightarrow k = 1$

- Policy Improvement: Assumption of no local optima otherwise exploration needed



- Dyn. Pr.: Full-Width Backup
  - TD: In between MC and DP

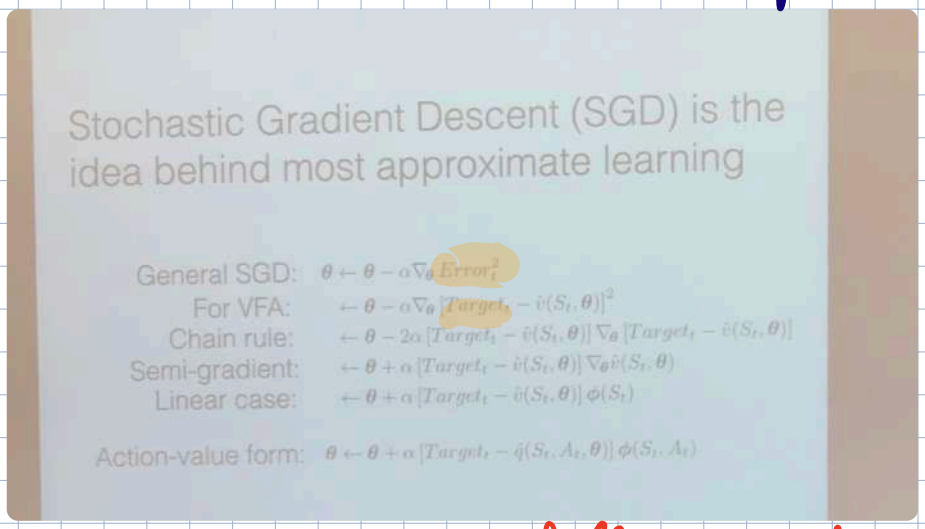
- Crucial assumption of Markovity for bootstrapping ⇒ BIAS
  - Introduce bias to reduce variance
  - MC not biased! ⇒ k-step TD interpolates!



- Epilepsy project: Use SDS
  - to simulate
    - Likelihood-Free methods!!

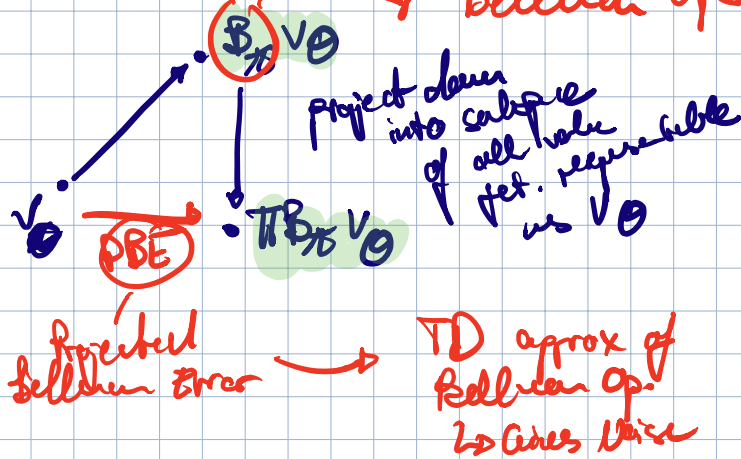
EEML Summer School - Day 2

Intro to RL II - Policy Gradient → Control

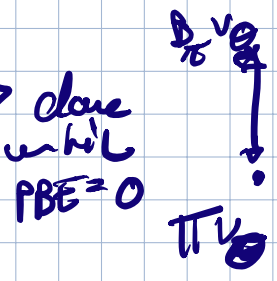


↳ Transform into form of supervised learning

↳ Assume that target does not depend on params
↳ Stability problems ⇒ Target networks
↳ Would not be a problem if full MC



$$\bullet\ (\mathcal{B}_\pi V)(s) := \sum_a \pi(s,a)\Big[r(s,a) + \gamma \sum_{s'} p(s'|s,a)\, V(s')\Big]$$

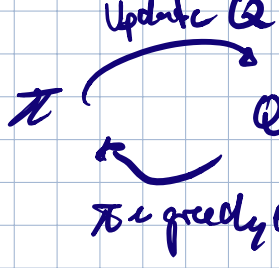


↳ Plain shifts can policy cliffs! ⇒ Need to constrain that shift
E.g. introducing trust region! → Clipping / PPO

↳ Linear fct. approx: $MSVE(\theta_{TD}) \leq \frac{1}{1-\gamma} MSVE(\theta_{MC})$ → bigger gives in non-linear!

↳ n-step backups: Trade-Off bias - variance ⇒ Full MC has large variance!
↳ of controls for TD(λ)

- Monte-Carlo Control:



⇒ Random exploring starts!
Learn vs off-policy

↳ Exploration: ε-greedy → distorts ⇒ problem: indep. of actual values
↳ In practice: exploration schedules are hard to implement

- Targets: $y_t = r_{t+1} + \gamma\, Q(s_{t+1}, a_{t+1}; \theta_t)$ [SARSA]

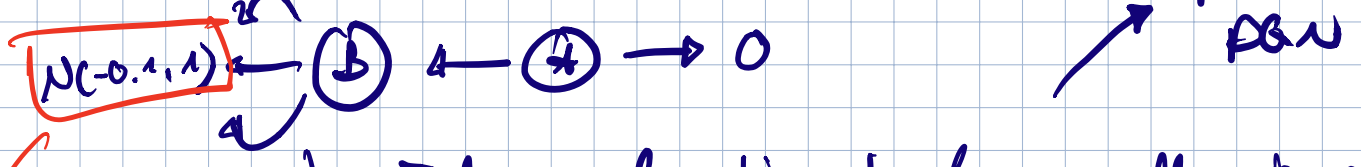
$y_t = r_{t+1} + \gamma \sum_a \pi(a|s_{t+1})\, Q(s_{t+1}, a; \theta_t)$ [Exp. SARSA]

$y_t = \sum_{s',r} p(s', r|s, a)\left[r + \gamma \sum_a \pi(a|s_{t+1})\, Q(\dots)\right]$

↳ SARSA: on-policy → next action is chosen at t to update
↳ skill explore?

- Q-Learning: Existing → Can theoretically converge even completely
  ↳ SARSA vs. Q-L: Safety vs. Optimality

- Maximization bias:



↳ Takes a long time to learn small values
↳ Use separate networks ⇒ Form of ensemble!
↳ Stochastic policy or stochasticity in environment
↳ Trade-off: More shifts → # iterations between targets and update
more iterations ⇒ larger shift but smaller variance!

- Rainbow: Orthogonal improvements ⇒ but maybe there is something better! → META RL!
- Need for different sets of optimizers in RL → different targets require different learning

- Policy gradient Methods:

$\pi(a|s, \theta)$ → Objective: $J(\theta) = v_{\pi_\theta}(s_0)$ → Initial state distr.
↳ $\theta_{t+1} := \theta_t + \alpha \nabla J(\theta_t)$ ↳ quality of policy

↳ $\nabla J(\theta) \propto \sum_s d_\pi(s) \sum_a q_\pi(s,a)\, \nabla_\theta \pi(a|s, \theta)$

↳ baseline subtract

Deriving REINFORCE from the PGT

$$\nabla\eta(\theta) = \sum_s d_\pi(s) \sum_a q_\pi(s,a) \nabla_\theta \pi(a|s,\theta),$$

$$= \mathbb{E}_\pi \left[\gamma^t \sum_a q_\pi(S_t,a) \nabla_\theta \pi(a|S_t,\theta) \right]$$

$$= \mathbb{E}_\pi \left[\gamma^t \sum_a \pi(a|S_t,\theta) q_\pi(S_t,a) \frac{\nabla_\theta \pi(a|S_t,\theta)}{\pi(a|S_t,\theta)} \right]$$

$$= \mathbb{E}_\pi \left[\gamma^t q_\pi(S_t,A_t) \frac{\nabla_\theta \pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)} \right] \quad \text{(replacing } a \text{ by the sample } A_t \sim \pi\text{)}$$

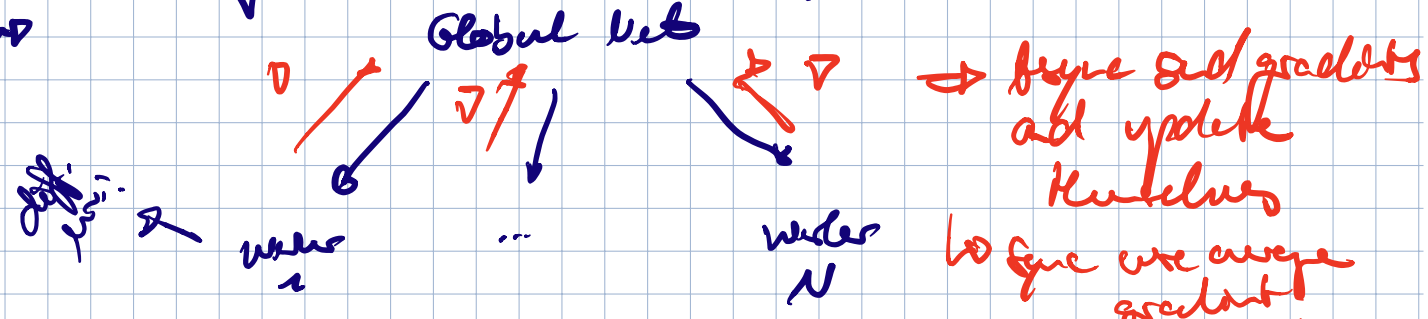
$$= \mathbb{E}_\pi \left[\gamma^t G_t \frac{\nabla_\theta \pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)} \right] \quad \text{(because } \mathbb{E}_\pi[G_t|S_t,A_t] = q_\pi(S_t,A_t)\text{)}$$

Thus

$$\theta_{t+1} \doteq \theta_t + \alpha \widehat{\nabla \eta(\theta_t)} \doteq \theta_t + \alpha \gamma^t G_t \frac{\nabla_\theta \pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)}$$

- No parametrization of value fct. needed
  - to use returns
  - to MC estimation
  - to get unbiased
  - lead to stabilize!
- AC → parametrize value fct!

- A3C → Random exploration by parallelization!
  - Pooling of experiences → Mean of gradients
  - to



→ Share and gradients and update Networks
to give one average gradient!

- TRPO / PPO → Enforce KL constraint as large!

Multi-agent RL – Shimon Whiteson (Oxford)

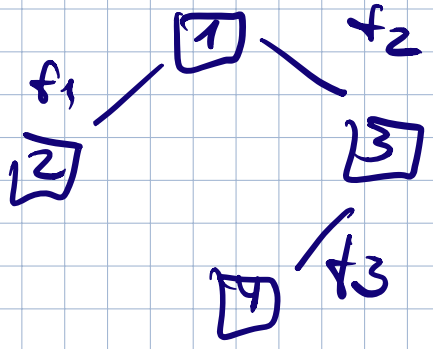
- Motivation: Classical RL abstracts many aspects
  → world is full of multi-agent systems

COOPERATIVE	COMPETITIVE	MIXED
→ Shared reward	→ Zero-Sum Game	→ General Sum
→ Coordination	→ Opposing rewards	→ Nash Eq.
	→ Minimax	→ QRE, Econ

- **Cooperative setting:** Absence of social conventions ⇒ risk of miscoordination
  → Exponential joint-action space. ⇒ grows very fast
  → Curse of real good ⇒ Coordination Graphs

$$Q(u) = f_1(u^1, u^2) + f_2(u^1, u^3) + \dots$$

→ agent 3



→ Probabilistic graphical model
⇒ Solutions: MAP estimation!
→ Encoding of cond. independence
⇒ No sequentiality!

→ Variable elimination:

$$\max_u Q(u) = \max_{u_2,u_3,u_4} \left[f_3(u^3, u^4) + \max_{u_1} \left[f_1(u^1, u^2) + f_2(u^1, u^3) \right] \right]$$

→ last is best response to agent 2,3

$f_2(u^2, u^3)$

→ form of cooperative World!
→ Computationally expensive the more connected the graph

→ Computes only optimal value of joint action ⇒ do backward pass to actually derive MAP action!
→ Need to know the graph / Cond. Indep. assumptions

→ Max-Plus ⇒ Vlassis et al (2004) → Message Passing
→ Two agent local payoff assumption

↳ Improving messages iteratively
↳ Convergence guarantee in acyclic graphs

→ MDP (+ Sequentiality!): Centralized Multi-Agent MDP
↳ All agents see state → Not really multi-agent: Simply large action/state space → simply comb. optimization

→ Independent Q-Learning: No attempt to model $Q(s,u)$
↳ each agent learns $Q(s_t, u_t^a)$! → Nonstationary if all agents learn → no convergence guarantee! → Tan et al (1993)

→ Coordinated Q-Learning: Guestrin et al (2002)

$$Q^{tot}(s,u) = \sum_{e=1}^{d} Q_e(s^e, u^e)$$

→ subsets over agents / ad factors

↳ Extremely strong assumption! → Transitions might be coupled
↳ Update each factor ⇒ graph is stationary!

- Partial observability → requires decentralized execution!
  ↳ Learning should be centralized! → Share params / Gradients
  ↳ Minimal vs. right assumptions!

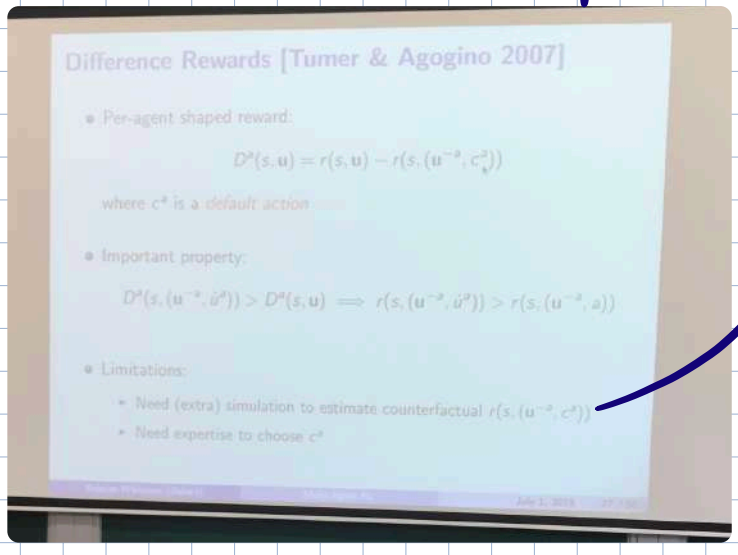
- Dec-POMDP
  ↳ $O(s,a): S \times A \to Z$ ↳ action-obs hist: $\tau^a \in T \equiv (Z \times U)^*$
  ↳ Decentralized policies: $\pi^a(u^a|\tau^a): T \times U \to [0,1]$
  ↳ Dilemma: Exploit private info vs. being predictable → Reproduce as reward fct

- Policy Gradient Methods for MARL:
  → PG useful if greedification is hard!
  → MAAC independent: Share parameters — still indep. agents + agent index
  ↳ nonstationary, hard to learn coordination, multi-agent coordinations.

- Counterfactual MADDPG → Foerster et al 2018 (COMA)

① Centralize Critic

② Wolpert & Tumer (2000) ⇒ Wonderful Life Utility
to reason about counterfactual reward if you would not have participated



→ need to go back into simulator!
to Counterfactual Baseline

$$A^a(s,u) = Q(s,u) - \sum_{u'^a} \pi^a(u'^a|\tau^a)\, Q(s,(u^{-a},u'^a))$$

to problem: as u^a becomes deterministic we can't get the counterfactual aspect

to COMA critic: output head gives all values

to still monolithic joint value fct.

- Value Decomposition Networks – Sunehag et al 2017

→ per agent: $Q_{tot}(\tau,u) = \sum_{a=1}^{n} Q_a(\tau^a, u^a; \theta^a)$

to decentralizes the agents ⇒ no longer local greedification

- QMIX: VDN only summable but mixing network

⇒ constrain to non-neg weights!

to can throw away after centralised learning!

SMAC framework → StarCraft Baseline!

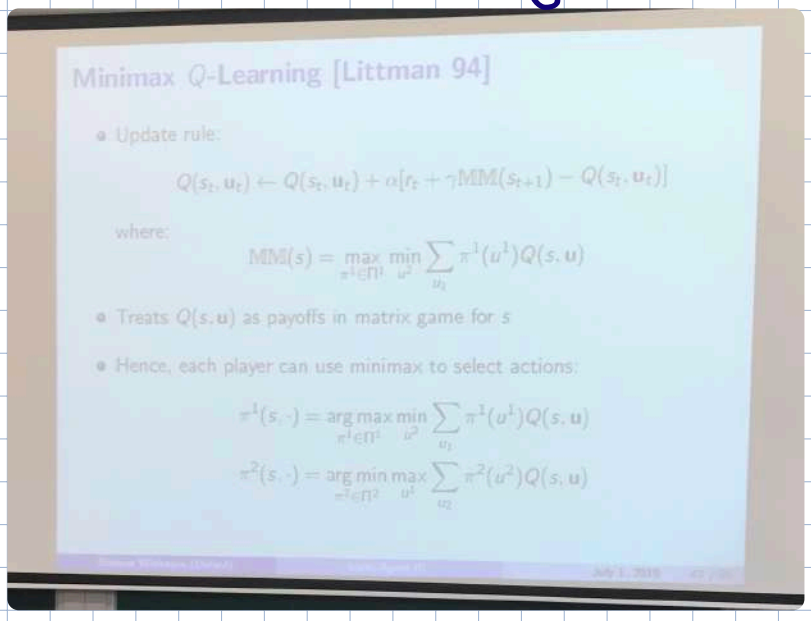
- Competitive setting: In competitive setting ⇒ one optimal policy exists → Nash: Stochasticity

→ Mixed strategies ⇔ Nonlinear Maximum

to Rational vs. strategic agents → Equilibrium ordering does not matter

to Inner player does not have to behave stochastically!

→ Minimax Q-Learning → Littman (1994)



→ Tabular approach: Learns about the opponent from data
↳ Ind. Q-Learning → implicit
↳ Explicit: Fictitious play
Rowan 2021
→ Opponent Modeling

→ Self-Play: Checkers → Arthur Samuel (1956)
↳ natural curriculum → can fail if game is not transitive
↳ learn generally learnable to beat opponents while beating strategies and → Alpha Star → League / Hall of Fame!

- Related: Nash Equilibrium → Brother!
↳ Nash Equilibrium Q-Learning → Hu & Wellman (1998)
→ almost never guaranteed to converge!
↳ "If multi-agent learning is the answer, what is the question?"
↳ Mechanism design! → Economics
↳ Friend-or-Foe Q-Learning