Part 3 - Proximal Policy Optimization - Optional

Runar Fosse

May 9, 2024

Exercise 3 - Entropy

1. *Compute the expected value of the random variable X with the distributions above, that is, $E_{p1}[X]$, $E_{p2}[X]$, $E_{p3}[X]$, $E_{p4}[X]$.*

$$\begin{aligned} E_{p1}[X] &= 0.2 \cdot 0 + 0.2 \cdot 1 + 0.2 \cdot 2 + 0.2 \cdot 3 + 0.2 \cdot 4 \\ &= 2 \end{aligned}$$

$$\begin{aligned} E_{p2}[X] &= 0.1 \cdot 0 + 0.2 \cdot 1 + 0.4 \cdot 2 + 0.2 \cdot 3 + 0.1 \cdot 4 \\ &= 2 \end{aligned}$$

$$\begin{aligned} E_{p3}[X] &= 0.1 \cdot 0.0 + 0.2 \cdot 0.2 + 0.4 \cdot 0.4 + 0.2 \cdot 0.6 + 0.1 \cdot 0.8 \\ &= 0.4 \end{aligned}$$

$$\begin{aligned} E_{p4}[X] &= 0.1 \cdot 0.0 + 0.2 \cdot 0.2 + 0.2 \cdot 0.4 + 0.4 \cdot 0.6 + 0.1 \cdot 0.8 \\ &= 0.44 \end{aligned}$$

2. *Which distributions above are undistinguishable when you summarize them with the expected value?*

   $E_{p1}[X] = E_{p2}[X]$, therefore, summarizing the whole distributions using only expected value, $p1$ and $p2$ would be indistinguishable.

3. *Compute the variance of the random variable X with the distributions above, that is, $Var_{p1}[X]$, $Var_{p2}[X]$, $Var_{p3}[X]$, $Var_{p4}[X]$.*

$$\begin{aligned}\mathrm{Var}_{p1}[X] &= (0-2)^2 + (1-2)^2 + (2-2)^2 + (3-2)^2 + (4-2)^2 \\ &= 4 + 1 + 0 + 1 + 4 = 10\end{aligned}$$

$$\begin{aligned}\mathrm{Var}_{p2}[X] &= (0-2)^2 + (1-2)^2 + (2-2)^2 + (3-2)^2 + (4-2)^2 \\ &= 4 + 1 + 0 + 1 + 4 = 10\end{aligned}$$

$$\begin{aligned}\mathrm{Var}_{p3}[X] &= (0.0-0.4)^2 + (0.2-0.4)^2 + (0.4-0.4)^2 + (0.6-0.4)^2 + (0.8 - \\ &= 0.4\end{aligned}$$

$$\begin{aligned}\mathrm{Var}_{p4}[X] &= (0.0-0.44)^2 + (0.2-0.44)^2 + (0.4-0.44)^2 + (0.6-0.44)^2 + (\\ &= 0.408\end{aligned}$$

4. *How does the variance help with the problem of discrimination?*

   Variance adds a second source of information to summarize a distribution, which could give extra information distinguishing different distributions which otherwise would look equal. More specifically, variance helps give information about how a distribution spans the domain. However, this is not always sufficient, as we can see that since $\mathrm{Var}_{p1}[X] = \mathrm{Var}_{p2}[X]$, we would still classify $p1$ as equal to $p2$, even though they are not.

5. *Why could we be interested only in the shape of a distribution?*

   Only caring about the shape of a distribution would allow us to identify similar/equal distributions without regarding information such as scale or position. E.g., a normal distribution centered at 0.0, with standard deviation of 1.0, and a normal distribution centered at 1000, spanning with a standard deviation of 100. Even though these are vastly different in terms of which values would be sampled from them, they still both are ordinary normal distributions.

   Another argument could be that the shape of the distributions act as another discriminant when it comes to summarizing different distributions. E.g., currently using only expectation value and variance, we would conclude that $p1 = p2$. However, we can see that $p1$ is a uniform distribution whilst $p2$ is a normal distribution. Taking distribution into account, we would successfully distinguish these as different.

6. *Compute the entropy of the random variable X with the distributions above, that is, $H_{p1}[X]$, $H_{p2}[X]$, $H_{p3}[X]$, $H_{p4}[X]$.*

$$H_{p1}[X] = -(0.2 \cdot ln(0.2) + 0.2 \cdot ln(0.2)+$$
$$0.2 \cdot ln(0.2) + 0.2 \cdot ln(0.2) + 0.2 \cdot ln(0.2)) \approx 1.28755$$

$$H_{p2}[X] = -(0.1 \cdot ln(0.1) + 0.2 \cdot ln(0.2)+$$
$$0.4 \cdot ln(0.4) + 0.2 \cdot ln(0.2) + 0.1 \cdot ln(0.1)) \approx 1.47081$$

$$H_{p3}[X] = -(0.1 \cdot ln(0.1) + 0.2 \cdot ln(0.2)+$$
$$0.4 \cdot ln(0.4) + 0.2 \cdot ln(0.2) + 0.1 \cdot ln(0.1)) \approx 1.47081$$

$$H_{p4}[X] = -(0.1 \cdot ln(0.1) + 0.2 \cdot ln(0.2)+$$
$$0.2 \cdot ln(0.2) + 0.4 \cdot ln(0.4) + 0.1 \cdot ln(0.1)) \approx 1.47081$$

7. *Which distributions are now undistinguishable?*

   As $H_{p2}[X] = H_{p3}[X] = H_{p4}[X]$, $p2$, $p3$ and $p4$ are indistinguishable.

8. *Can you explain in which sense* $p4$ *has the same shape as* $p2$ *or* $p3$*?*

   $p2$, $p3$ and $p4$ are all normal distributions, therefore they would have the same "shape". However, this "shape" disregards any skew (which $p4$) has. Therefore, they all get classified as equal.

9. *Are indeed the values of the domains* Ω_1, Ω_2 *used in the computation of the entropy?*

   No, they are not. The shape of the distribution is calculated disregarding the domain.

10. *How does the entropy relate to the spread of the distribution? Does entropy increase or decrease if the distribution becomes more concentrated?*

    The closer any sampled probability gets to $p(x) = 1$, the smaller $log(p(x))$ will become. Therefore, a concentrated distribution would have a much lower entropy than a spread out distribution. Using a hypothetical extremely concentrated distribution (a constant distribution) $p = [1.0], \Omega_p = [1.0]$ we would have $H_p = -1.0 \cdot ln(1.0) = 0$.

11. *Why might we want to promote entropy in RL? What would happen if the entropy is minimized? What advantage would you get from promoting entropy?*

    We have concluded that entropy could be used as a measure of how spread out a distribution is, independent of its domain. Therefore, entropy could be a good choice for RL where we want to encourage exploration whilst trying to optimize an agent to perform the most rewarding actions.

    If entropy is minimized, we would converge toward a constant (deterministic) policy. Promoting bigger values of entropy would be equivalent to promoting an agents exploration. This is very useful within reinforcement learning, where we want an agent to explore the environment to a degree where it finds an optimal policy.

Exercise 4 - KL

1. *Can you explain why $KL(p;q)$ is asymmetric in p, q?*

   $KL(p;q)$ is asymmetric in p, q as $KL(p;q) \neq KL(q;p)$ given that $p \neq q$.

2. *Compute $KL(p1;p1)$. Does the results make sense?*

$$\begin{aligned} KL(p1;p1) &= 0.2 \cdot \frac{ln(0.2)}{ln(0.2)} + 0.2 \cdot \frac{ln(0.2)}{ln(0.2)} + 0.2 \cdot \frac{ln(0.2)}{ln(0.2)} \\ &+ 0.2 \cdot \frac{ln(0.2)}{ln(0.2)} + 0.2 \cdot \frac{ln(0.2)}{ln(0.2)} = 5 \cdot 0.2 \\ &= 1 \end{aligned}$$

   Yes, the results make sense. As they are equal distributions we would expect the divergence to be 1.

3. *Compute $KL(p1;p2)$. Does the results make sense?*

$$\begin{aligned} KL(p1;p2) &= 0.2 \cdot \frac{ln(0.2)}{ln(0.1)} + 0.2 \cdot \frac{ln(0.2)}{ln(0.2)} + 0.2 \cdot \frac{ln(0.2)}{ln(0.4)} \\ &+ 0.2 \cdot \frac{ln(0.2)}{ln(0.2)} + 0.2 \cdot \frac{ln(0.2)}{ln(0.1)} = 5 \cdot 0.2 \\ &\approx 1.03088 \end{aligned}$$

   Yes, the results also make sense here. These are similar-ish distributions over the same domain. Therefore we would expect the divergence to be small.

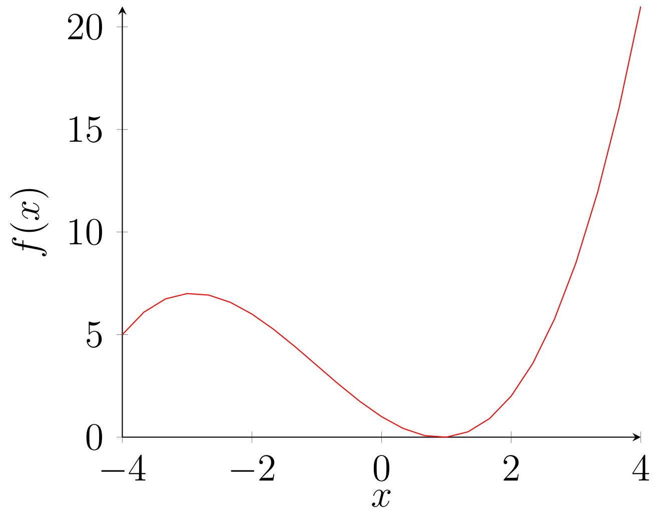
4. *Compute $KL(p1;p3)$. Does the results make sense?*

$$KL(p1;p3) = 0.2 \cdot \frac{ln(0.2)}{ln(0.0)} + 0.2 \cdot \frac{ln(0.2)}{ln(0.0)} + 0.2 \cdot \frac{ln(0.2)}{ln(0.0)} + 0.2 \cdot \frac{ln(0.2)}{ln(0.0)} + 0.2 \cdot \frac{ln(0.2)}{ln(0.0)}$$

No, the results do not make sense. We have several occurrences of $ln(0.0)$. This is because $p1$ and $p3$ are distributions over different domains. KL divergence is only defined for distributions over the same domain.

Exercise 5 - First- and second-order optimization, and trust region optimization

1. *Plot the function* $f(x) = \frac{1}{4}x^3 + \frac{3}{4}x^2 - 2x + 1$ *over the domain* $x \in [-4, 4]$.



2. *Assume that we are currently at value $x = 2$. What is the value of $f(x)$ at $x = 2$?*

$$f(2) = \frac{8}{4} + \frac{12}{4} - 4 + 1 = 2$$

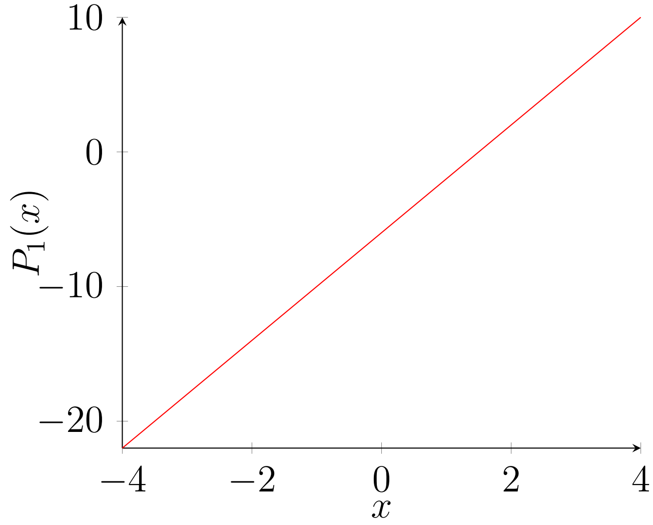
3. *First, compute the first derivative $f'(x)$.*

$$f'(x) = \frac{3}{4}x^2 + \frac{6}{4}x - 2$$

4. *Then, compute the Taylor approximation of first-order around $x = 2$.*

$$\begin{aligned} P_1(x) &= f(2) + f'(2) \cdot (x - 2) \\ &= 2 + 4 \cdot (x - 2) \\ &= 4x - 6 \end{aligned}$$

5. *Plot the first-order Taylor approximation.*



6. *What information is the first-order Taylor approximation providing and how would you use it to perform gradient descent?*

   The first-order Taylor approximation provides us with a rough estimation of how the function grows if represented linearly. Using gradient descent we could see how the local minimum lies in comparison to $p = 2$. As the function is more negative the more to the left we go, we could imagine there is a local minimum there, so we perform gradient descent moving in that direction (negative gradient).

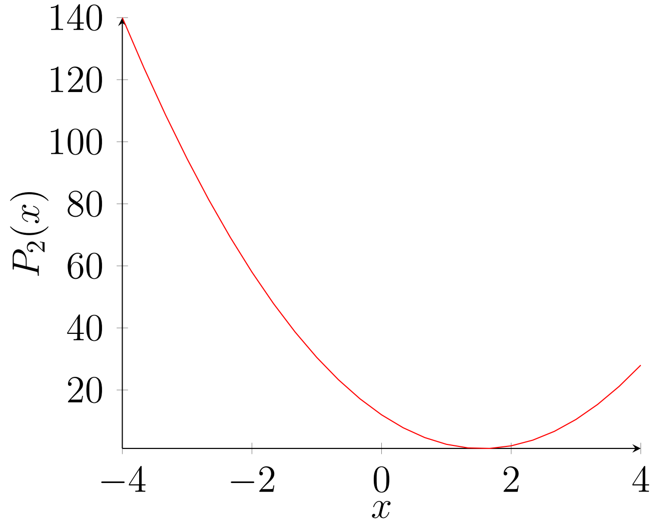
7. *Compute now also the second derivative $f''(x)$.*

$$f''(x) = \frac{6}{4}x + \frac{6}{4}$$

8. *Compute the Taylor approximation of second-order around $x = 2$.*

$$\begin{aligned}P_2(x) &= f(2) + f'(2) \cdot (x-2) + f''(2) \cdot (x-2)^2 \\ &= 2 + 4 \cdot (x-2) + \frac{9}{2} \cdot (x^2 - 4x + 4) \\ &= \frac{9}{2}x^2 - 14x + 12\end{aligned}$$

9. *Plot the second-order Taylor approximation.*



10. *How would you use the additional information of the second-order Taylor approximation to perform gradient descent?*

    Now we finally have a quadratic function, meaning we have an approximation of a minima. We could use this information to gradient descent down to this minima, approximating the minima of the function $f(x)$.

11. *If there is more information in the second derivative, why are not second-order optimization methods more widely used in machine learning?*

    To use second-order optimization techniques, we need to first approximate the second-order derivative from samples. This introduces a lot more complexity within the optimization methods, and is another approximation. This could lead to very complex methods where the increase in complexity is not necessarily proportional to the increase in result quality.

Exercise 6 - PPO

1. ***Challenges:*** *what are the challenges and the limitations in the field detected by the authors?*

   There were 3 main challenges and limitations in the field detected by the authors. These are scalability, data efficiency and robustness.

2. ***Aim:*** *what do the authors want to achieve with their work?*

   They want to improve the current state of affairs by introducing an algorithm which attains data efficiency and robustness without being overly complex.

3. ***Contributions:*** *what are the contributions of this paper to the field?*

   They introduced a well-performing and revolutionary deep reinforcement learning algorithm PPO (proximal policy optimization), and compare its performance to already existing, well performant DRL algorithms.

4. ***Evaluation:*** *how do the authors assess the quality of their results?*

   As mentioned above they compare PPO's performance to other algorithms (like actor-critic) on reinforcement learning problems. There were two main performance measures, speed of learning and final performance.

5. *Relate Equation (1) to the corresponding formula we used in the course.*

Equation (1) is defined as

$$\hat{g} = \hat{\mathbb{E}}[\nabla_\theta \mathrm{log} \pi_\theta(a_t|s_t)\hat{A}_t]$$

whilst the equation we've previously seen in the course is defined as

$$\mathbf{E}_{f_\pi(a,s;\mathbf{w}_\pi)}[\delta_v \nabla_{\mathbf{w}_\pi} \mathrm{log} f_\pi(a,s;\mathbf{w}_\pi)]$$

.

6. *In the course, we used the notation $f_\pi(a, s|\mathbf{w}_\pi)$ and $\mathbf{w}_\pi$ for a policy function estimator and its parameters. What notation does the paper use?*

   The paper uses the symbol π_θ for its policy function estimator, and θ for its parameters.

7. *Explain what the paper means by* ***using automatic differentiation software.***

   This would be using libraries like PyTorch which handle gradient updates and backpropagation for you, automatically, as long as a loss function has been defined.

8. *Consider how the paper goes from Equation (1) to Equation (2). It goes in the opposite way compared to what we did in the course. Can you make sense why it does that?*

   Yes. They first want to give us an intuitive understanding of how they've defined their policy gradient estimation, and then provide information on how to use that gradient estimation to learn models using an automatic differentiation software.

9. *Express policy gradient as the maximization of the objective in Equation (2).*

   It could be expressed as

   $$\max_{\theta} L^{PG}(\theta)$$

10. *What is the meaning and role of a surrogate objective function?*

    The purpose and role of the surrogate function is to improve stability and prevent too large of gradient updates.

11. *What is the meaning of $\pi_{\theta_{old}}$?*

    This is the policy before updating the weights.

12. *What is the role of $\pi_{\theta_{old}}$ in the maximization of Equation (3)? How would $\pi_{\theta_{old}}$ behave if we were to take the gradient ∇_θ of the expectation in the Equation (3)?*

    It is there to provide a way to calculate the ratio between if an action is more or less likely to be selected after policy updates.

    As π_{old} is fixed, it could easily be derived if one were to compute ∇_θ.

13. *What do you expect is the effect of the ratio $r(\theta) = \frac{\pi_\theta}{\pi_{\theta_{old}}}$ in Equation (3)?*

    $r(\theta) > 1$ if action is more likely after update, and respectively $r(\theta) < 1$ if action is less likely.

14. *What does the constraint of Equation (4) add?*

This constraints the size of the policy update, preventing too large of a policy update. I.e., new policy distribution after the update shouldn't be too different from the old policy distribution before.

15. *How does the objective overall change from Equation (2) to Equation (3-4)?*

    Now, instead of trying to maximizing the expected reward given the policy itself, we want to maximize the expected reward given a change in policy. One could say that this explicitly values learning.

16. *Two TRPO maximization problems are presented, a hard version with constraints in Equations (3-4) and a soft version with a trade-off parameter β in Equation (5). Which maximization problem is encoded in standard TRPO?*

    Equation (3-4) is the "standard" TRPO.

17. *Why does the standard TRPO use one maximization problem instead of the other?*

    Standard TRPO uses Equation (3-4) as using the soft version with the parameter β could be hard. Choosing a β which works over several problems, or even within a single problem is hard, and generally, using Equation (3-4) performs better, as it provides a lower (pessimistic) bound on the performance of the policy π_θ.

18. *Does the solver for standard TRPO use first-order or second-order optimization?*

TRPO uses the Hessian (second-order differentiation) matrix. Therefore it uses second-order optimization.

19. *How does Equation (6) relate to Equation (3-4)?*

    Equation (6) is exactly the same as Equation (3), omitting the constraint Equation (4).

20. *What is the limitation of Equation (6)?*

    Without any constraint, policy updates could become excessively large.

21. *How does Equation (7) relate to Equation (3-4)?*

    Instead of having an explicit constraint Equation (4), they clip/clamp the ratio to the interval $[1-\epsilon, 1+\epsilon]$. This prevents the large policy updates by preventing ratios far away from $\pi_{\theta_{\text{old}}}$ from appearing.

22. *Consider Figure 1. Explain the meaning of the axes.*

    The y-axis is the value of the loss (expectation value), and the x-axis is the value of the ratio $r(\theta)$.

23. *Consider Figure 1. Why are there two plots? Can L^{CLIP} be negative for $A > 0$, or L^{CLIP} be positive for $A < 0$?*

    There are two plots to provide visual information on how the loss reacts to $A > 0$ and $A < 0$, as they change the function in different ways, and using a 3D plot would not be efficient in a paper.

    No, L^{CLIP} cannot be negative for $A > 0$ and L^{CLIP} cannot be positive for $A < 0$. This is because

$$r(\theta) > 0, \forall\theta$$

24. *Consider Figure 1. What value does π_θ assume at the red dot?*

    At the red dot, $\pi_\theta = \pi_{\theta_{\text{old}}}$. This is because $r(\theta) = 1$, and this only happens given the case above.

25. ***Critical questions?*** *would you have any question to ask the authors about this algorithm?*

    No, not currently.

26. *Recall the soft formulation of the TRPO problem from Section 2.2. What was the main difficulty in implementing this objective?*

    That it is hard to find a suitable value for β.

27. *How do the authors suggest to tackle that challenge now?*

    They now use an adaptive value for β, which is being automatically updated based on the KL divergence of the old and new policy.

28. *How many hyperparameters does their approach introduce?*

    Only β, however this is not that important to tune initially, as it automatically will be tuned when training.

29. *What is the role of the Adaptive KL penalty coefficient loss?*

    The role is to prevent too large policy updates within a few updates.

30. ***Critical questions:*** *would you have any question to ask the authors about this algorithm?*

How were scores in the adaptive updates decided on?

31. *How do the authors suggest you would compute the gradient of L^{CLIP} or L^{KLPEN}? How would that work?*

    They suggest using automatic differentiation software, and constructing the loss value of L^{CLIP} or L^{KLPEN} instead of L^{PG}. Then we can perform gradient updating and backpropagation on this automatically (using e.g. PyTorch).

32. *Why are the authors considering the idea of computing $V(s)$?*

    By computing $V(s)$ we can sample and approximate the advantage function $\hat{A}$ over several timesteps T.

33. *What class of RL algorithms would PPO belong to?*

    PPO would belong to the class of actor-critic algorithms. This is because we use both an estimation of the policy, as well as of the state-value function.

34. *We considered a few methods in class for estimating $V(S)$, however the authors proposed different solutions. What are they suggesting?*

    One could either approximate $V(S)$ in the same network as the policy π_θ, or use a seperate network altogether.

35. *The authors consider the possibility of a network sharing parameters between the policy and the value function. Where did we see in class a similar architecture?*

    The duelling DQN used a similar approach.

36. *What is the rationale for adding an entropy bonus?*

    An entropy bonus in the loss function would promote and ensure sufficient exploration, and prevent a policy from stagnating in a sub-optimal behaviour. We learned in Exercise 3 that entropy is low when a distribution is close to deterministic (constant), but high when stochastic.

37. *Consider Equation (9). Explain the role of each term (L^{CLIP}, L^{VF}, S) in making up the overall loss ($L^{CLIP+VF+S}$).*

    L^{CLIP} is considers the expected reward given a change in the policy π_θ. L^{VF} considers the loss regarding the state-value function, and ensures that the approximation is a correct estimate of the actual value function. (This is supposed to be handled as an actual loss (wants to be minimized), and is therefore negated in the equation). And S is the entropy bonus, promoting that the agent explores its environment.

38. *Consider Equation (9). What is the role of c_1 and c_2?*

    These are coefficients weighting the "importance" of the different terms.

39. *Consider Equation (9). How would you practically deal with the expected value $\mathbb{E}$ when you implement your algorithm?*

    We would add up all the different terms/losses, and assume that this represents our expectation value. Then we would use e.g. PyTorch to handle gradient updates and backpropagation.

40. *The authors follow the approach of running T timesteps and then compute the estimator in Equation (10). What approach does this remind you of?*

    This is very similar to TD (temporal difference methods). More specifically, TD(T).

41. *What approach instead would you have if you were to run episodes until the end instead of T timesteps?*

    This would be a Monte-Carlo approach (MC).

42. *Equation (11) provides a generalization of Equation (10). Verify that indeed Equation (11) reduces to Equation (10) when $\lambda = 1$.*

    This is a given by inspecting the declared variable δ_t.

43. *The final PPO algorithm relies on another couple of tricks: parallelization and mini-batches. Explain them.*

    Parallelization is gotten from running several instances of the same agent at the same time, before updating.

    Mini-batches mean running through several sub-samples of the total samples.

44. ***Critical questions:*** *would you have any question to ask the authors about this algorithm?*

    How was this practically implement, e.g. in PyTorch?