Reinforcement Learning Paper Summaries

Jaskirat Singh
College of Engineering and Computer Science
Australian National University
ACT
jaskirat.singh@anu.edu.au

Abstract

This document is intended to be a short reference document to various reinforcement learning papers, I encounter during my research. For each paper, I record a brief copy of the problem statement, objectives, key ideas and a summary of results. This document is supposed to be a go-to resource for fast referencing during paper write-up, or may also be used to as a quick refresher of reinforcement learning research.

Contents

Preprint. Under review.

1 Generalization

1.1 Quantifying Generalization in Reinforcement Learning

Conference. ICML, Cobbe et al. (2018)

Objectives. The paper focused on analysing the factors leading to over-fitting in deep RL. More specifically, the paper states the following main objectives/ contributions:

1. Show that the number of training environments required for good generalization is much larger than the number used by prior work on transfer in RL.
2. Propose use of separate training and test sets. For procgen environments, 500 or fixed, finite training set vs unbounded testing set (2^{32} levels) provides good critria for measuring generalization performance.
3. Evaluate the impact of different convolutional architectures and forms of regularization, finding that these choices can significantly improve generalization performance.

Key Ideas.

- Use of separate training and test sets.
- Using supervised learning techniques like L2 regularization, data augmentation, batch normalization, **stochasticity** helps in the better generalization.
- Just as in supervised learning, with careful design, deeper CNNs perform better.

Results. All results are quite informative and should be referred to from the original paper.

Conclusion. *Excellent Paper*. It is an essential read while working on generalization in reinforcement learning. In addition to the said analysis, it provides food for thought and information about several factors affecting over-fitting.

Investigation Points.

- Read "A study on overfitting in Deep RL", Zhang et al. (2018).
- Learning faster has no correlation with generalization
- The reason for not deploying regularization techniques from supervised learning is the presence of same train and test set.
- Read "Improved regularization of convolutional neural networks with cutout".
- Investigate other papers on attaining generalization using stochasticity. The paper analysis 2 ways, ϵ greedy policy and entropy bonus like in PPO. Also check the theoretical basis for why ϵ greedy is suitable for countering overfitting, apart from the intuitive reason (refer Mnih et al. (2013)).
- Different generalization schemes address similar concerns while countering overfitting? What concerns? How?
- Increase in training performance beyond a certain number of training levels. Points to learning of more generalizing features even on the training dataset. Does supervised learning performance decrease after a certain point in terms of training data?

1.2 Assessing Generalization in Deep Reinforcement Learning

Conference. Arxiv, Packer et al. (2018)

Objectives. Provide a standard method to evaluate both in-distribution (interpolation) and out-distribution (extrapolation) generalization in deepRL agents.

Key Ideas.

- Clarify difference between in-distribution (interpolation) and out-distribution (extrapolation) generalization.

- There are two main approaches to generalization in RL: learning policies that are robust to environment variations and learning policies that adapt to such variations.
- Success rate might be better metric than total reward for environments with different dynamics. This holds true in cases where the total reward is dependent on the environment parameters. E.g. reduced reward for more timesteps on a longer game level.

Results.

- Packer et al. (2018) claims that, "vanilla" deep RL algorithms trained on a varied distribution of MDPs generalize better than specialized schemes that were proposed specifically to tackle generalization. However, the statement needs to be verified and might only hold for EPOpt (Rajeswaran et al., 2016) and RL^2 (Duan et al., 2016) models.

Conclusion. Good paper. *Excellent relevant work section*, which is great for building a strong foundation on current (till date) work on generalization. The paper is also good read for a theoretical view on generalization as training over a distribution of MDPs. Finally, the paper consists of some dubious claims which might be worth investigating.

Investigation Points.

- Consider how interpolation vs extrapolation distinction affects the generalization research in deepRL.
- Further investigate the 3 main approaches to generalization:
  1. Learning policies that are robust to environment variations. A popular approach to learn a robust policy is to maximize a risk-sensitive objective, such as the conditional value at risk over a distribution of environments (refer Tamar et al. (2015); Rajeswaran et al. (2016)), though this may lead the agent to sacrifice performance on many environments just to perform well on a few tough ones.
  2. Adapting using the environment trajectories. Use environment trajectories to learn an environment embedding which is then used for policy and value networks (refer Duan et al. (2016)).
  3. Finetuning at test time, like meta learning (refer Finn et al. (2017)).
- Revisit the correctional term for recurrent networks in policy gradients while using the entire (s_t, a_t, r_t) pair as input to the RNN, instead of just state s_t.
- Implement EPOpt on ProcGen (Rajeswaran et al., 2016).
- Using total reward vs something like success rate (independent of reward shaping) for ProcGen.
- For continuous action spaces, can we do better by not assuming a diagonal covariance matrix.
- Is EPOpt only good for continuous action spaces?
- More variance training environments (maybe synthetic) vs better algorithms to generalize from a limited set of environments.
- EPOpt by choosing a subsample of s,a pairs needs to be adapted to follow the policy gradient theorem.
- Is model-based RL more suited for generalization? Since it already learns the environment dynamics, the corresponding variation in these dynamics can be modelled directly in the network.

2 Meta-RL

2.1 Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks (MAML)

Conference. ICML 2017, Finn et al. (2017)

Objectives. For any general collection of tasks, in RL or supervised learning, find an initialization of parameters θ such that gradient descent updates from that point on a new task lead to quick learning/adaptation.

Key Ideas. For a network f with parameter θ, and a distribution of tasks τ_i sampled from $p(\tau)$, and task-specific loss $\mathcal{L}_i$,

$$\theta_i' \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_i(\theta) \tag{1}$$

$$\mathcal{L}(\theta) = \sum_i \mathcal{L}_i(\theta_i') \tag{2}$$

$$\theta' \leftarrow \theta - \beta \nabla_\theta \mathcal{L}(\theta) \tag{3}$$

On further analysis,

$$\mathcal{L}(\theta) = \sum_i \mathcal{L}_i(\theta_i') \tag{4}$$

$$= \sum_i \mathcal{L}_i(\theta - \alpha \nabla_\theta \mathcal{L}_i(\theta)) \tag{5}$$

$$\approx \sum_i \mathcal{L}_i(\theta) - \alpha (\nabla_\theta \mathcal{L}_i(\theta))^2 \tag{6}$$

So, the overall loss tries to minimize the sum of task-specific losses (which is the usual method), and in-addition aim to maximize the sensitivity of the losses to the changes in θ, which is expressed in the second term as the square of gradient of the task-specific loss.

$$\nabla_\theta \mathcal{L} = \sum_i \nabla_\theta \mathcal{L}_i \left[1 - \alpha \nabla_\theta^2 \mathcal{L}_i\right] \tag{7}$$

Results. From an RL-perspective, shows some convincing results with small modifications in-between tasks e.g. different target velocities for a walker, different starting/goal positions for a 2D maze etc.

Conclusion. *Excellent paper and idea.* A very impressive and general meta-RL idea for both supervised learning and RL.

Investigation Points.

- Investigate the use of trust-region methods like PPO for computation of final loss gradient $\nabla_\theta \mathcal{L}$, without sampling new trajectories with the updated task-specific parameter θ_i'
- Think of application of MAML to different scenes from ProcGen.
- Other ways of easy computation of Hessian.
- Meta-learning as learning how to learn the model-parameter.
- Is there a contrast with RL^2, Duan et al. (2016)?
- What could be other applications *e.g.* Wortsman et al. (2019)?
- What about application to tasks from different domains? not just different scenes?
- Qualitative analysis on how the final θ differs from just training on a collection of tasks? Compare the train-only (no-finetuning to new tasks) performance on train set and test set.
- How does the theory change for including more number of task-specific updates before computing final loss?
- K-shot learning refers to adaptation to a new task given only K-samples from the task. Read more in both RL and supervised learning contexts.

2.2 RL^2 Fast reinforcement learning via slow reinforcement learning

Conference. Arxiv, Duan et al. (2016)

Objectives. Counter low sample complexity (not clear if met)

Key Ideas. The key idea is to use an RNN for learning both the policy and value function.

Results. Shows almost similar performance as previous state of the art, for multi-armed bandit problems. Though, the visual navigation experiment is interesting.

Conclusion. Good paper, but key takeaway is to use an RNN for learning both policy and value function. Duan et al. (2016) also argue that "Rather than designing a "fast" reinforcement learning algorithm, we propose to represent it as a recurrent neural network (RNN) and learn it from data. In our proposed method, RL2, the algorithm is encoded in the weights of the RNN, which are learned slowly through a general-purpose ("slow") RL algorithm." However this assertion is questionable as using an RNN can be interpreted as using a more abstract state h_{t-1}, rather than just s_{t-1}, where the abstract state h_{t-1} is learned from past trajectory $\tau^{t-} : \{s_0, a_0, r_0, ..., r_{t-1}\}$.

Investigation Points.

- Is an LSTM equipped to provide a good balance between unigram previous state s_{t-1} and general prior state h_{t-1}? ... Seems so, but analyse how hindering/ increasing the carry over of long term memory in LSTMs effects performance.

3 Visual Navigation

3.1 Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning

Conference. ICRA 2017, Zhu et al. (2017)

Objectives.

- Use deep reinforcement learning for the task of visual navigation. (first of its kind)
- Introduce use of AI2-THOR framework
- Policy as a function of both current state and the goal (both as images), hence the term target driven.

Key Ideas.

- There are 3 types of generalization to consider in visual navigation
  - **Target generalization.** Generalization to new unseen targets without prior training.
  - **Scene Generalization.** Generalization to unseen scenes.
  - **Real World Generalization.** Generalization to real world settings, albeit allowing for some short finetuning
- Generalization across targets achieved through a deep siamese network which takes as input both the current state and the target image and leads to a joint embedding.
- This embedding is then used for learning actor / critic parameters through scene-specific layers. (should not need them though)
- The sharing of weights between target and the current state for all scenes is good for generalization and improves convergence speed across unseen scenes.

Results.

- A3C benefits from use of more threads.
- More training scenes lead to better generalization performance.

Conclusion. *Essential paper.* It was the first to purpose the use of deep reinforcement learning for visual navigation.

Investigation Points.

- More scenes lead to better train performance?
- Separate scene specific layers should not be required.
- Read more about A3C Mnih et al. (2016). Can it be used for any RL setting? What does asynchronous mean?
- Look at the previous works for VN, e.g. map-based methods.
- Analyse how the next papers cover the shortcomings of this method.

3.2 Visual Semantic Navigation using Scene Priors

Conference. Arxiv. 2018, Yang et al. (2018)

Objectives. Incorporate semantic knowledge graphs within the deep reinforcement learning framework for visual navigation using GCNs, Kipf & Welling (2016).

Key Ideas.

- Semantic relationship and learning of priors between objects categories is essential for visual navigation. *For e.g.* a remote is usually found near a television etc.
- Incorporate semantic knowledge through relational graphs, using GCNs.
- Input to the policy and critic networks is a concatenation of,
  - Resnet features for the current RGB image view (512-d).
  - Fast-text word embedding for the target object followed by an fc-layer (512-d).
  - Input to GCN is concatenation of visual features and embedding features. The GCN generates an output of dimension 512-D.

Results. The experiment and ablation results show the importance of semantic information for navigation, but it should be noted that the paper uses a different task setting, object types etc than Wortsman et al. (2019).

Conclusion. *Good paper.* Shows the importance of scene-priors and demonstrates a way of incorporating that information graph using a GCN, Kipf & Welling (2016).

Investigation Points.

- Read more about a GCN. Kipf & Welling (2016)
- The task setting in Wortsman et al. (2019), chooses object instances to ensure maximum visibility and avoids objects hidden within other objects, like a cup inside a cupboard. How can we address the case of partially observable room environment or a multi-room environment, where the agent is unable to see all objects from the current position (even after rotating).
- Can we add semantic relationship between the edges. like a distribution of recommended actions. For eg. laptop is mostly on top of a table, so the corresponding edge can store a distribution continuous/discrete with a maximum for the "top" relation.
- Instead of an object relation-graph, is it possible to create a graph for the current scene by looking around, then updating the graph as we move along. In a it is analogous to a map creation, but within the DRL framework.
- Advanced trajectory planning. At any given position, the agent decides to reach a certain point and creates a series of actions for the same. If the trajectory planning network is accurate, then it greatly increases the possibility of avoiding trajectories which may lead to deadlocks. The agent chooses a trajectory if the expected value at the end state is the highest.

3.3 Learning to Learn How to Learn: Self-Adaptive Visual Navigation using Meta-Learning

Conference. CVPR 2019, Wortsman et al. (2019)

Objectives. Learn a loss function for performing fast adaptation to new scenes in a manner similar to MAML.

Key Ideas.

- Want to perform fast adaptation to new scenes at test time in a manner similar to MAML.
- But, the loss function is not available during test times, *e.g.* from simulation to the real world.
- Need to learn a loss function from that mimics the original navigation loss. (might not be the correct way to do things), but RL loss can only be computed after collecting samples, which is not a real possibility for testing.

- So learn a loss function (interaction-loss) that can mimics the gradients of navigation loss from θ, and train θ using MAML.
- The following describe some ways to achieve this:
  1. Train θ at train time using MAML and navigation loss, and learn a loss function which mimics gradients from navigation loss.
  2. Learn a loss function which mimics gradients from navigation loss, and train θ at train time using MAML with interaction loss alone.
  3. Learn a loss function which mimics gradients from navigation loss, and train θ at train time using MAML with both navigation and interaction loss.
  4. The method suggested in paper: Train θ with MAML, but compute new θ_i' using interaction loss. It has the combined effect of minimizing expected navigation loss, while trying to maximize the difference inner product between gradients of navigation and interaction loss. At inference, perform one or more updates using interaction loss.

Results. Improved performance as a result of fast adaptation at inference times.

Conclusion. *Excellent idea* of using MAML in context of adapting to new scenes at test times.

Investigation Points.

- Think of other methods for learning the interaction loss and performing MAML, Finn et al. (2017).
- Investigate the case when ground object truth is available.
- Compare the training performances alone as well, without parameter updates. If it is improving, then the update is doing much more than just focus on the
- Read more about the handcrafted interaction losses to be used at test time.
- Are the inference updated θ, carried on during training? What about the same for human learning?
- Not training or studying is equivalent to not updating the loss function, but still learning very slowly due to the already learned loss function. Analyse further.

3.4 Learning Object Relation Graph and Tentative Policy for Visual Navigation

Conference. ECCV 2020, Du et al. (2020)

Objectives. Improve performance of deep RL agent on the visual navigation performance.

Key Ideas.

- Learn a object relational graph (ORG) to learn spatial and semantic similarities of different object types. For *e.g.* a remote is likely placed near a TV.
- Use supervision on optimal action (possible because of dividing the state space into grid) for deadlock states to help the agent learn faster while avoiding wasteful states.
- Since, the agent does not learn to get out of deadlock states automatically using RL, we train a separate TPN (tentative policy network) network for leaning the optimal action (given deadlock state) to get out of deadlock state during test time.

Results. Very big improvement over the previous SAVN-baseline Wortsman et al. (2019).

Conclusion. *Good paper*, with 2 main concepts. First, object relations are important. Second, we can improve learning efficiency by providing oracle supervision for wasteful states like deadlock states.

Investigation Points.

- Study separate affects of ORG vs TPN vs not giving fasttext Bojanowski et al. (2017) labels for target object, which are lack distinctiveness for objects of related object categories. e.g. the agent may get confused whether the target category is a table or chair.
- Can we use supervision on optimal action in a more general way, rather than just using it for deadlock states. Care must be taken to avoid overfitting.

- Maybe the agent can use active learning, *i.e.* the agent can learn a metric that tells whether it requires supervision. Also, reduce the supervision gradually during training. Or start injecting noise in the optimal action supervision labels.

References

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. ISSN 2307-387X.

Cobbe, K., Klimov, O., Hesse, C., Kim, T., and Schulman, J. Quantifying generalization in reinforcement learning. *arXiv preprint arXiv:1812.02341*, 2018.

Du, H., Yu, X., and Zheng, L. Learning object relation graph and tentative policy for visual navigation. *arXiv preprint arXiv:2007.11018*, 2020.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. Rl^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

Packer, C., Gao, K., Kos, J., Krähenbühl, P., Koltun, V., and Song, D. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.

Rajeswaran, A., Ghotra, S., Ravindran, B., and Levine, S. Epopt: Learning robust neural network policies using model ensembles. *arXiv preprint arXiv:1610.01283*, 2016.

Tamar, A., Glassner, Y., and Mannor, S. Optimizing the cvar via sampling. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Wortsman, M., Ehsani, K., Rastegari, M., Farhadi, A., and Mottaghi, R. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6750–6759, 2019.

Yang, W., Wang, X., Farhadi, A., Gupta, A., and Mottaghi, R. Visual semantic navigation using scene priors. *arXiv preprint arXiv:1810.06543*, 2018.

Zhang, C., Vinyals, O., Munos, R., and Bengio, S. A study on overfitting in deep reinforcement learning. *arXiv preprint arXiv:1804.06893*, 2018.

Zhu, Y., Mottaghi, R., Kolve, E., Lim, J. J., Gupta, A., Fei-Fei, L., and Farhadi, A. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3357–3364. IEEE, 2017.