Watch your Weight Reinforcement Learning

Robert Müller [1]

Abstract

Deep Neural Networks (DNN) have been a crucial driver behind the recent successes of deep reinforcement learning (RL). While allowing for a tremendous scale-up to high dimensional domains like robot-control or GO they have also introduced various difficulties into the training process, that have been addressed by methods like double Q-learning to stabilize learning. However, the role of the weights is normally limited to a storage object. This work connects the information in the weights via a PAC-Bayes bound to generalisation across a distribution of environments and showcases maximum-entropy and exploration via noise injection as places to exploit the connection of information in weights and activations.

1. Introduction

Deep reinforcement learning has recently seen successes on a variety of tasks (Mnih et al., 2015; Vinyals et al., 2019). DNNs are a common building block across all these works. Their common use is restricted as being the function approximator that approximates the value-function or policy in an algorithm. The use of DNNs for RL comes at the expense of additional challenges in training combated for example by double Q-learning (van Hasselt et al., 2015). Empirical evidence relates the flatness of minima to the generalisation properties in supervised learning (Hinton and van Camp, 1993; Hochreiter and Schmidhuber, 1997). Recently (Achille et al., 2019) relate the information encoded in the weights of a DNN via a PAC-Bayes bound to its generalisation properties. This results in a training objective that is at the same time an upper bound on the test error. Inspired from this connection of information in the weights of a DNN and the generalisation properties we extend this analysis to RL. We begin with related work before introducing different information types in weights and activation's. Subsequently we define the complexity of a policy for a given environment and give PAC-Bayes generalisation bounds for distributions of tasks. We conclude by using this perspective to take a look at max-ent RL and exploration by adding noise schemes.

2. Related Work

The bottleneck theory, of learning an encoding with as few bits about the input as possible while retaining good performance, was introduced by (Tishby et al., 2000), who pointed out in (Tishby and Zaslavsky, 2015) that the bottleneck principle is also applicable to deep neural networks. While optimizing the IB objective directly is intractable, (Alemi et al., 2016) introduced the variational information bottleneck (VIB) to obtain a tractable lower bound, which was shown to improve the robustness of image classification models. The variational autoencoder (Kingma and Welling, 2013) can likewise be interpreted as an autoencoder equipped with a VIB.

(Tirumala et al., 2019) apply the IB in activation's in RL and develop a hierarchical learning schema. (Igl et al., 2019) use the IB in activation's as regularisation and study the impact on generalisation. (Goyal et al., 2019) use the IB to discover options. (Rakelly et al., 2019) train a task inference network with an IB that is at meta-test-time used for task-inference.

(Achille et al., 2019) connect information in weights and activation's of a DNN relate in addition information in weights to generalisation via PAC-Bayes bounds. We extend their analysis to the RL case. (Majumdar et al., 2018) consider policy learning across a distribution of environment's and derive an algorithm based on directly optimizing the PAC-Bayes bound resulting from a reduction of policy learning to supervised learning. In comparison to their work we explicitly draw the connection between the PAC-Bayes bound and the information in the weights as well as a extension to structured task-families.

Max-ent RL is a paradigm maximizing instead of the normal reward an augmented reward $r(s,a) + H(\pi(\cdot|s))$ (Ziebart et al., 2008; Haarnoja et al., 2018). A topic of ongoing research is why and when max-ent RL works well. (Eysenbach and Levine, 2019) conjecture that it helps in cases where the reward function is chosen in an adversarial way

[1] Department of Mathematics, Technical University of Munich, Germany. Correspondence to: Robert Müller <2robert.mueller@gmail.com>.

Inductive Biases, Invariances and Generalization in RL (BIG) Workshop at the 37^{th} International Conference on Machine Learning, Vienna, Austria, 2020. Copyright 2020 by the author(s).

from within a certain class of reward functions.

3. Background

3.1. Reinforcement Learning

A Markov Decision Process (MDP, (Bellman, 1957)) is a quintuple $(\mathcal{S}, \mathcal{A}, \gamma, \mathrm{P}, \mathrm{R})$. Here $\mathcal{S}$ and $\mathcal{A}$ stand for the sets of states and actions respectively, such that for $s_t, s_{t+1} \in \mathcal{S}$ and $a_t \in \mathcal{A}$: $\mathrm{P}(s_{t+1}|s_t, a_t)$ is the probability that the system/agent transitions from s_t to s_{t+1} given action a_t and $\mathrm{R}(a_t, s_t, s_{t+1})$ is a reward obtained by an agent transitioning from s_t to s_{t+1} via a_t and γ the discount factor.

A policy $\pi_\theta : \mathcal{S} \to \mathcal{A}$ is a (possibly randomized) mapping (parameterized by $\theta \in \mathbb{R}^d$, in our case weights of a DNN) from $\mathcal{S}$ to $\mathcal{A}$. The goal of RL is to find parameters θ of a policy π_θ such that an agent applying it in the environment given by a fixed MDP maximizes total discounted reward $\mathbb{E}_\pi \sum_{t=1}^{H} \gamma^t r$ over given horizon H. In this paper we consider MDPs with finite horizons.

In **multi-task learning RL (MT-RL)** we are given a distribution of tasks $\mathcal{E}$ that are either given in form of a dataset of environments $\{E_1, \ldots, E_N\}$ or a distribution to sample from $E \sim p_{\mathcal{E}}$. The goal is to learn a policy, more specifically, parameters or a distribution of parameters of an optimal policy, maximizing the expected sum of rewards across all the environments in $\mathcal{E}$. Let $J_{\mathbb{E}}(\theta)$ denote the sum of expected reward for using policy π with parameters θ in environment E and analogous $J_{\mathcal{E}}(\theta) = \mathbb{E}_{E\sim\mathcal{E}}\, J_E(\theta)$ the expected sum of rewards over tasks of distribution $\mathcal{E}$. The goal of learning is to find: $\max_\theta J_{\mathcal{E}}(\theta) = \max_\theta \mathbb{E}_{E\sim\mathcal{E}}\, J_E(\theta)$.

Denoting by Θ the space of all possible parameters, $\mathcal{T} \sim \Theta$ a distribution over parameters and $\theta \sim \mathcal{T}$ a concrete realisation. We can write $\theta \sim \Theta$ where we identify $\mathcal{T}$ with the Dirac distribution with probability mass at θ. Then we can formulate the learning objective to learn an optimal parameter distribution as: $\max_{\mathcal{T}\in\Theta} J_{\mathcal{E}}(\theta) = \max_{\mathcal{T}\in\Theta} \mathbb{E}_{E\sim\mathcal{E}}\, \mathbb{E}_{\theta\sim\mathcal{T}}\, J_E(\theta) . \mathcal{T}$ can be realized as the distribution of the parameters of a Bayesian network.

3.2. Information Theory

We refer to A.1 for definitions on elementary information theory and A.2 for a definition of fisher information (FI).

Consider a neural network $X \to Z \to Y$ with input X, intermediate encoding Z and output Y. The encoding to the intermediate layer is given by a parametric encoder $f(z \mid x, \theta)$, so $z = f_\theta(x)$. We wish to strive for simplicity, thus we seek an encoding that is maximally informative about the target while satisfying an upper bound (I_c) on the encoded information about the input. This can be formalized, measuring the amount of encoded **information in the activations** as MI $I(Z; X)$, via the information bottleneck(IB) objective (Tishby et al., 2000): $\max_\theta MI(Z;Y;\theta) s.t. MI(X;Z;\theta) \leq I_c$. it is possible to optimize for an embedding being minimally informative about the input, by setting $I_c = 0$.

If we have a DNN and measure the information in the activation's a natural next step is to consider the **information in the weights**. A possible measure of the information in the weights is to consider weights $\theta \sim Q$, where Q is a distribution over weights as in a bayesian NN. Defining a prior P over the weights the encoding length of the weights could be computed as $MI(Q; P)$. (Achille et al., 2019) argue that instead of considering the encoding length of the weights one should rather care about effective information in the weights, as weights, that when perturbed, yield almost the same loss as before don't carry a lot of information.

Recalling the definition of Z, $z = f_w(x)$ as an intermediate layer of a DNN and the motivation of effective information in the weights, that small perturbations in uninformative weights don't change the loss, (Achille et al., 2019)(Def.4.1) define **effective Information in the activation's** informally as the information that are not destroyed by a small perturbation in the weights: $MI_{eff}(x; z) = MI(x; f_{w+n}(x))$.

(Achille et al., 2019) connect in their proposition (4.2) **the information in activation and weights**. They find that, under some technical conditions, i) that if the FI in the weights (FIW) goes to 0, the FI in the activation's (FIA) goes to 0 as well and ii) the effective information between input and activation decreases as the FIW decreases. This suggests, that reducing the information the weights contained about the training set and optimization, for example by increasing the noise in SGD, decreases the mutual information between input and activation's at train and test time.

3.3. SGD as minimizing a energy functional

Let $T = \eta/B$ be the temperature, with step size η and batch size B. Empirically, it has been observed, that SGD prefers to converge to flatter minima as the temperature increases [see remark 3.9 in (Achille et al., 2019) for a discussion]. This suggests that it minimizes a free energy $\mathcal{F}(\theta) = L(\theta) + \frac{T}{2} \log |F(\theta)|$, with loss L. As in RL we seek to maximize the reward instead of minimizing a loss we can define a new free energy as $\mathcal{J}_E(\theta) = J_E(\theta) - \frac{T}{2} \log |F(\theta)|$ that is now maximized.

3.4. Shannon and FI in the weights

Prop. 3.5, remark 3.6 and remark 3.7 in (Achille et al., 2019) implies informally that in the limit of an uninformative prior $(\lambda \to \infty)$ for $P(\theta) = \mathcal{N}(0, \lambda^2 I)$ the only remaining term in the Shannon information in the weights $MI(Q(\theta|W); P(\theta))$ are the log-determinant of the FI and a $\log \lambda$ term.

4. Using information in the weights

IBs in activation's were commonly used in RL. We extend this to a Lagrangian containing the information in the weights. This amounts to an information bottleneck (IB) minimizing the amount of encoded information in the weights subject to the constraint that the policy obtains still a high reward.

4.1. Complexity of a task

The policy might contain weights that are not crucial for the subsequent action, if a weight configuration θ where perturbed to $\theta' = \theta + \Delta\theta$ and $J_E(\theta) \approx J_E(\theta')$ this weights contain basically no useful information about the policy for task E. Thus we do not simply care about the bits of information encoded about the bits but rather about the **effective information about about environment E**. Following (Achille et al., 2019) we define the complexity of an environment E at level β with prior distribution $P(\theta)$ and post-training-distribution $Q(\theta|E)$ as:

$$C_\beta(E, P, Q) = \mathop{\mathbb{E}}_{\theta\sim Q(\theta|E)}[-J_E(\theta)] + \beta KL(Q(\theta|E)\|P(\theta)). \tag{1}$$

For any fixed β there is a Q^* minimizing the task complexity. The complexity involving Q^* is called the **effective information in the weights for environment E at level** β. Note that we minimize the negative return plus the information in the weights which is equivalent to maximizing the return minus information in the weights. The complexity of a task distribution $\mathcal{E}$, $C_\beta(\mathcal{E}, P, Q)$ is defined analogous. Choosing a time dependent centroid prior, i.e. $P(\theta_t) = 1/N \sum_{i=1}^{N} Q(\theta_{t-1}|E_i)$ recovers the elastic averaging SGD objective (Zhang et al., 2014).

equation 1 takes the form of an information Lagrangian between the return of the policy with parameters θ and the Shannon information in the weights with information target zero. Using the result from 3.4 implies in particular to the Lagrangian involving the return and the FIW. This equals in particular the free energy expression for a reinforcement learning problem.

5. Generalisation in Multi Task Learning

We begin by restating the reduction of the control policy learning problem (which equals the MT) to the supervised learning problem by (Majumdar et al., 2018) in table 5. The return is the discounted sum of rewards obtained by following policy π_θ in environment $E \in \mathcal{E}$ with horizon H. Policy parameters θ are updated by any RL algorithm based on rollout-data of environment E in order to maximize the sum of rewards. Single task RL is recovered by choosing $\mathcal{E}$ as singleton E. Given a task distribution $\mathcal{E}$ consisting of train and test tasks we give a PAC-Bayes bound to connect the information in the learned weights and the return on the training tasks to the return on the test tasks.

Supervised Learning	MT-RL
Input Data: x $\in X$	Environment: E $\in \mathcal{E}$
Hypothesis: f$_\theta : X \to Y$	Rollout: $\pi_\theta : \mathcal{E} \to (S \times A)^H$
Loss: l: $\Theta \times X \to \mathbb{R}$	Return: C: $\Theta \times E \to \mathbb{R}$

Theorem 5.1. *((McAllester, 2013)(2), (Achille et al., 2019)(3.2) Under certain technical conditions (see the appendix for a full statement)for fixed $\beta > 1/2$, $\delta > 0$, prior weight distribution $P(\theta)$, learned weight distribution $Q(\theta|\mathcal{E}_{train})$ we have with probability at least $1 - \delta$ over the samples of $\mathcal{E}$ that:*

$$\begin{aligned} J_{test}(Q) \geq & \frac{1}{1-\frac{1}{2\beta}} \left[E_{\theta\sim Q}\left[J_{\mathcal{E}_{train}}(\theta)\right]\right. \\ & -\frac{\beta}{N}\left(KL(Q\|P) + \log\frac{1}{\delta}\right), \end{aligned} \tag{2}$$

where $J_{test}(Q)$ is the average return of policy with weights $\theta \sim Q$ on the test tasks $\mathcal{E}_{test}$.

Real-world tasks do not come from a massive distribution, they follow rather a hierarchical structure. Thus it is more realistic to have a set of different tasks, each one occurring in different variations. An example of this structure are the Metaworld (Yu et al., 2019) benchmark ML10 and ML45. ML10 allows for the decomposition $ML10 = \{ML_i\}_{i=1}^{10}$ into ten separate tasks, where each individual task ML_i represents a distribution of tasks due to varying starting and goal positions. The plain policy learning objective (eq. 3.1) considers only one task distribution we extend, by identifying for example $ML_i = \mathcal{E}_i$, the policy learning problem to hierarchical task distributions $M = \{\mathcal{E}_1, \ldots \mathcal{E}_M\}$ where it becomes: $\max_{\mathcal{T}\in\Theta} J_{\mathcal{E}}(\theta) = \max_{\mathcal{T}\in\Theta} \mathbb{E}_{E\sim\mathcal{E}_i} \mathbb{E}_{\mathcal{E}_i\sim M} \mathbb{E}_{\theta\sim\mathcal{T}} J_E(\theta)$.For this we can again bound the expected test error:

Theorem 5.2. *(McAllester, 2013)(4), (Achille et al., 2019)(3.2) Let M be a distribution over tasks, $\mathcal{E}_i \sim M$ we get the following bound over all possible data sets:*

$$J_{test}(Q) \geq \frac{1}{1-\frac{1}{2\beta}} \left[\mathop{\mathbb{E}}_{\mathcal{E}_i\sim M}\left[\mathop{\mathbb{E}}_{\theta\sim Q}[J_{\mathcal{E}_i}(\theta)]\right] - \frac{\beta}{N} KL(Q\|P)\right], \tag{3}$$

This means that minimizing the task complexity $C_\beta(\mathcal{E}_{train}, P, Q)$ does not only maximize the return on the training set but also a lower bound on the return on the test set of environments. The PAC-Bound optimization-method developed by (Majumdar et al., 2018) could be applied to optimize this objective directly.

6. Max-ent RL and information in weights

Using the property $MI(X;Y) = H(Y) - H(Y|X)$ it becomes apparent that maximum entropy RL $\pi(a|s)$ is up to a constant equivalent to minimizing the mutual information $MI(a|s)$. We have argued, that SGB minimizes a free energy of the form $-J_E(\theta) + \frac{T}{2} log|F(\theta)|$.

We have further seen the connection of information in weights and activation's and the fact that information in the weights going to zero drives the fisher-information and the effective information in the activation's to zero. We have shown that reducing the effective information in the weights helps generalisation.

We analyse this FIW - effective information in activation's max-ent RL chain of reasoning by empirically logging the FIW and the negative log likelihood of the policy during training of a Soft Actor Critic (SAC) (Haarnoja et al., 2018). It is intractable to compute the full FIW of the policy $F = \mathbb{E}_s \left[\nabla_w \log \pi_w(a|s) \nabla_w \log p_w(a|s)^T\right]$. As a tractable approximation, we resort to log-likelihood and FIW over the replay buffer $F = \mathbb{E}_{\boldsymbol{s \sim B}} \left[\nabla_\theta \log \pi_\theta(a|s) \nabla_\theta \log p_\theta(a|s)^T\right]$.

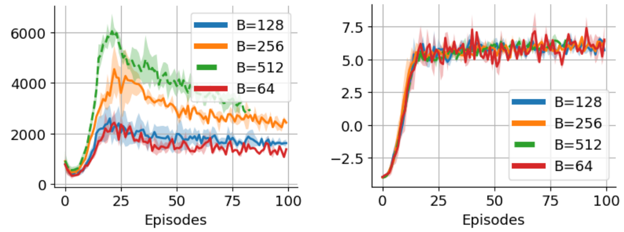

Figure 1. The median of FI (left) and $\log(\pi(a_t|s_t))$ (right) over the different episodes for a SAC trained on HalfCheetah, evaluated over 3 seeds. Different colors denote different batch-sizes.

Figure 1 shows in the left plot the median of FIW and in the right plot the median of $\log(\pi)$ against the episode number over 3 different seeds. Different colored lines correspond to different batch-sizes. Figure 1 conveys two messages. At the beginning of the FIW is expanding rapidly, corresponding to the encoding of the training data and the learning algorithm. As training continues there are fewer and fewer new information to encode and a contraction in the amount of the FI can be observed, corresponding to the optimisation towards flat minima. The expansion timing is comparable to the log probability of the taken actions that increases first and flattens subsequently. This flattening on a certain entropy level corresponds to a continuing compression of the information. Figure 2 in the appendix shows the median average discounted return. A curve of return minus log prob would show a similar dynamics as reward minus FIW. Thus we see empirically, that max-ent RL optimizes simultaneously the max-ent RL and the FIW Lagrangian objective.

Secondly, the choice of the optimization algorithm has a significant impact upon the learning behavior, remember in particular the temperature $T = \eta/B$ in the energy functional. In figure 1 we plot the fisher for multiple different batch sizes and see indeed that a larger batch-size corresponds to a lower temperature, resulting in a larger FIW.

We have shown in that minimizing the information in the weights improves generalisation. Furthermore, we have seen that minimizing the information in the activation via max entropy RL is an effective way to reduce the FIW, which in turn is a bound on the generalisation performance. This suggests that max-ent RL maximizes a lower bound performance over the domain of test environments form the training distribution which lends support to the suggestion of (Eysenbach and Levine, 2019) that max-ent helps by targeting adversarialy chosen reward functions. We conjecture that the empirical success of max-ent RL is due to directly optimizing an objective involving the return and the information in activation's (thus indirectly the information in the weights), rather than normal RL with SGD where the weight information minimization is only implicitly present in the energy minimisation formulation of SGD.

7. Conclusion and Future Work

The connection of information in weights and activation's allows one to draw connections between different concepts. This is exemplified at the example of adding noise for exploration. Adding noise to the action can be seen as a form of reducing the information in activation's while adding noise in parameter space can be seen as a direct way of reducing the FIW, as it enforces the policy to learn weights of low curvature. Future work includes exploring further connections through an information in the weights lens, in particular in meta learning.

While it is known that the minimizing the information in the weights drives also the information in the activation's to 0 there is a lack of a result in the reverse direction. Intuitively it is however clear that if the information in the activation's are minimized the information in the weights, which correspond to all previous data and the learning algorithm, should also be minimized. Future work is necessary to investigate different types of information in activation's and weights.

As the computational cost of the entropy regularisation is lower than of the computation of mutual information in the activation, which is in turn, cheaper than computing the information in the weights it remains to be seen if any of the different information Lagrangian's of the reasoning chain of quantities bounded against each other, allows for more effective optimisation, of the information in the weights governing generalisation, than max-ent RL.

In this work, we have drawn a connection between the information in activation's and weights. We have related the information in the weights and generalisation properties via a PAC-Bayes bound. We hope to inspire the community to watch and exploit the weights of the networks in the learner more intensively.

References

Achille, A., Paolini, G., and Soatto, S. (2019). Where is the information in a deep neural network?

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. (2016). Deep variational information bottleneck.

Bellman, R. (1957). A Markovian decision process. *Journal of Mathematics and Mechanics*, 6(5):679–684.

Cover, T. M. and Thomas, J. A. (1991). Elements of information theory.

Eysenbach, B. and Levine, S. (2019). If maxent rl is the answer, what is the question?

Goyal, A., Islam, R., Strouse, D., Ahmed, Z., Larochelle, H., Botvinick, M. M., Bengio, Y., and Levine, S. (2019). Infobot: Transfer and exploration via the information bottleneck. *ArXiv*, abs/1901.10902.

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic algorithms and applications. *ArXiv*, abs/1812.05905.

Hinton, G. E. and van Camp, D. (1993). Keeping the neural networks simple by minimizing the description length of the weights. In *COLT '93*.

Hochreiter, S. and Schmidhuber, J. (1997). Flat minima. *Neural Computation*, 9:1–42.

Igl, M., Ciosek, K., Li, Y., Tschiatschek, S., Zhang, C., Devlin, S., and Hofmann, K. (2019). Generalization in reinforcement learning with selective noise injection and information bottleneck.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational Bayes. *CoRR*, abs/1312.6114.

Majumdar, A., Farid, A., and Sonar, A. (2018). Pac-bayes control: Learning policies that provably generalize to novel environments.

McAllester, D. A. (2013). A pac-bayesian tutorial with a dropout bound. *ArXiv*, abs/1307.2118.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M. A., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518:529–533.

Rakelly, K., Zhou, A., Quillen, D., Finn, C., and Levine, S. (2019). Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *ICML*.

Tirumala, D., Noh, H., Galashov, A., Hasenclever, L., Ahuja, A., Wayne, G., Pascanu, R., Teh, Y. W., and Heess, N. (2019). Exploiting hierarchy for learning and transfer in kl-regularized rl.

Tishby, N., Pereira, F. C., and Bialek, W. (2000). The information bottleneck method.

Tishby, N. and Zaslavsky, N. (2015). Deep learning and the information bottleneck principle. *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5.

van Hasselt, H., Guez, A., and Silver, D. (2015). Deep reinforcement learning with double q-learning.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A. J., Chung, J., Choi, D. H., Powell, R. W., Ewalds, T., Georgiev, P., Oh, J., Horgan, D., Kroiss, M., Danihelka, I., Huang, A., Sifre, L., Cai, T., Agapiou, J. P., Jaderberg, M., Vezhnevets, A. S., Leblond, R., Pohlen, T., Dalibard, V., Budden, D., Sulsky, Y., Molloy, J., Paine, T. L., Gulcehre, C., Wang, Z., Pfaff, T., Wu, Y., Ring, R., Yogatama, D., Wünsch, D., McKinney, K., Smith, O., Schaul, T., Lillicrap, T. P., Kavukcuoglu, K., Hassabis, D., Apps, C., and Silver, D. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pages 1–5.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. (2019). Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*.

Zhang, S., Choromanska, A., and LeCun, Y. (2014). Deep learning with elastic averaging sgd.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. (2008). Maximum entropy inverse reinforcement learning. In *AAAI*.