Learning to Optimize with Gradient Based Exploration Strategies

I. Objective

Many machine learning techniques utilize gradient-based optimization methods to select model parameters. Although gradient-based optimization methods have been successful, algorithms such as Stochastic Gradient Descent and ADAM can be sometimes fail to escape anomalous regions of the optimization space.

Prior work has proposed formulating the gradient-based optimization problem as a learning problem in and of itself, as in Andrychowicz *et al.* [1] and Wichrowska *et al.* [15]. However, we note that these approaches still are susceptible to local minima in non-convex optimization problems, such as those presented by neural networks.

We propose the design of a learned optimizer that prioritizes exploration, which enables the learned optimizer to consistently find near optimal parameter settings, even in the presence of pathological loss surfaces. We formulate gradient-based optimization as a reinforcement learning problem where we extend the DDPG algorithm by Lillicrap *et al.* [12] with an ensemble of exploration policies. These exploration policies enable us to augment the learned optimizer with samples from diverse regions of the state space.

II. Background and Related Work

Work, such as that of Kingma *et al.* [8], attempt to improve to stochastic gradient descent based optimization. However, Glorot *et al.* [5] and Mishkin *et al.* [13] demonstrate the importance of initialization on overall model performance. Prior work has formulated optimization as a learning problem in and of itself. Andrychowicz *et al.* [1] utilize a recurrent neural network to learns an optimization strategy based on prior domain specific experience. Wichrowska *et al.* [15] study a hierarchical RNN that is meta-trained by a standard optimizer. Finally, Li *et al.* [11] use guided policy search to learn an optimization policy using the previous gradients and realized objective value in addition to the current parameters.

Several papers examine augmenting reward with an auxiliary objective to maximize entropy [6, 7] or visit states at where a learned dynamics model is weak [2]. The techniques used in Fu *et al.* [3] are similar to those in [2], where instead of determining the difficulty in dynamics modeling, states are explored based on how difficult they are to distinguish from states that have been visited in the past. However, all of these approaches focus mainly on novelty, while the exploration policies we create ensure that promising, yet diverse regions of the state-space are explored.

Another class of methods implicitly encourage exploration by training several local policies, constrained to a global policy, such as guided policy search algorithms[10][9][4]. Osband *et al.* [14] approximates a distribution over Q-values via bootstrapping. These techniques each increase exploration but fail to explore far beyond the greedy policy.

III. Formulation

We aim to utilize techniques in deep reinforcement learning to learn robust optimizers for a variety of function classes, with a particular focus on neural networks. The goal of optimization is to determine some parameter vector θ such that $f(\theta)$ is minimized. Thus, deterministic optimization algorithms can essentially be described as a policy $\pi(f,(\theta^0,...\theta^{i-1}))$ that outputs a parameter update $\Delta\theta$ [11].

Typically some gradient based approach is used for π, but in this work we aim to learn a more robust parameter update policy using DDPG, a state-of-the-art off-policy deep reinforcement learning algorithm [12], with the reward function simply being the function value $f(\theta)$. For our state representation, we use a fixed length history of $(f(\theta),\theta,\Delta\theta)$ tuples, while the action is simply the parameter update $\Delta\theta$.

To ensure robustness to parameter initialization, we roll-out an ensemble of exploration policies in parallel during training that are encouraged to collect experience from promising, yet diverse regions of parameter space. This experience can then be used to augment the experience of the trained policy π to make it aware of a variety of high-reward regions in parameter space. Consider a set of N exploration policies. For a parameter vector of dimension M, each of the $N << M$ exploration policies is assigned a unique coordinate $i \in \{1,...M\}$ to explore along. Then, at any given timestep, each exploration policy takes a step along its assigned coordinate with probability p_{coor}, while with probability $1-p_{coor}$ it takes a step in the direction of its local gradient. The value of p_{coor} is decayed over time. Furthermore, we also decay the step-size of the exploration policies over time. The motivation behind this exploration strategy is that it encourages exploration policies to spread out and explore diverse regions of the state space, but eventually start drawing experience from locally optimal areas. We also plan include ways to reset exploration policies that spend too long in low-reward regions and to prune poorly performing coordinate directions.

To establish the method's efficacy, we make a number of comparisons to establish this work's contributions:

1) Compare against other "learning to learn" methods on image classification tasks like CIFAR-100.
2) Benchmark against existing gradient descent methods using accuracy, sample complexity, wall clock time etc. to give a holistic picture of the tradeoffs being made, to exchange the algorithm's complexity with accuracy.
3) Compare approaches that naively explore (n runs of ε-greedy SGD) with explorers that directly optimize for state-space coverage.

References

[1] M. Andrychowicz, M. Denil, S. G. Colmenarejo, M. W. Hoffman, D. Pfau, T. Schaul, and N. de Freitas, “Learning to learn by gradient descent by gradient descent”, *CoRR*, vol. abs/1606.04474, 2016.

[2] P. A. Bradly C. Stadie Sergey Levine, “Incentivizing exploration in reinforcement learning with deep predictive models”, in *Int. Conference on Learning Representations (ICLR)*, 2015.

[3] J. Fu, J. Co-Reyes, and S. Levine, “Ex2: Exploration with exemplar models for deep reinforcement learning”, in *Advances in Neural Information Processing Systems 30*, 2017, pp. 2577–2587.

[4] D. Ghosh, A. Singh, A. Rajeswaran, V. Kumar, and S. Levine, “Divide-and-conquer reinforcement learning”, *CoRR*, vol. abs/1711.09874, 2017.

[5] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks”, in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[6] T. Haarnoja, H. Tang, P. Abbeel, and S. Levine, “Reinforcement learning with deep energy-based policies”, in *Proc. Int. Conf. on Machine Learning*, 2017, pp. 1352–1361.

[7] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor”, *CoRR*, vol. abs/1801.01290, 2018.

[8] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization”, *arXiv preprint arXiv:1412.6980*, 2014.

[9] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies”, *CoRR*, vol. abs/1504.00702, 2015. arXiv: `1504.00702`.

[10] S. Levine and V. Koltun, “Guided policy search”, in *Proceedings of the 30th International Conference on Machine Learning*, S. Dasgupta and D. McAllester, Eds., ser. Proceedings of Machine Learning Research, vol. 28, Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, pp. 1–9.

[11] K. Li and J. Malik, “Learning to optimize”, *CoRR*, vol. abs/1606.01885, 2016.

[12] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning”, *CoRR*, vol. abs/1509.02971, 2015. arXiv: `1509.02971`.

[13] D. Mishkin and J. Matas, “All you need is a good init”, *arXiv preprint arXiv:1511.06422*, 2015.

[14] I. Osband, C. Blundell, A. Pritzel, and B. Van Roy, “Deep exploration via bootstrapped dqn”, in *Proc. Advances in Neural Information Processing Systems*, 2016, pp. 4026–4034.

[15] O. Wichrowska, N. Maheswaranathan, M. W. Hoffman, S. G. Colmenarejo, M. Denil, N. de Freitas, and J. Sohl-Dickstein, “Learned optimizers that scale and generalize”, *CoRR*, vol. abs/1703.04813, 2017.