Double-GAE

Jacob Hilton

October 15, 2020

We'd like to train autoregressive models using RL. But we have a choice: we can think of either completions or tokens as actions. This gives rise to two different policy gradient estimators. Here we'll analyze these estimators, and put those that use GAE [Schulman et al., 2015] into a unified perspective, using an advantage estimator we'll call Double-GAE.

1 Notation and terminology

Let's call completions *outer actions* and denote them using boldface letters, and let's call tokens *inner actions* and denote them using lightface letters with two indices, a completion index and a token index. Likewise for states, rewards and so on. We ignore the distinction between states and observations, and so states are prompts. We write states and actions next to one another to indicate concatenation. Thus

$$\mathbf{a}_t = a_{t0}a_{t1}\dots a_{t(n_t-1)},$$

where n_t is the length of the tth completion, and

$$s_{tu} = \mathbf{s}_t a_{t0}a_{t1}\dots a_{t(u-1)}$$

for $0 \leq u \leq n_t - 1$. We assume that the outer and inner rewards are related by

$$\mathbf{r}_t = r_{t0} + r_{t1} + \dots + r_{t(n_t-1)},$$

and we refer to r_{tu} for $u < n_t - 1$ as *inner-only rewards*.

2 Double-GAE

Suppose our inner policy is given by an autoregressive model $\pi_\theta(a_{tu} \mid s_{tu})$. Then the outer policy is given by

$$\boldsymbol{\pi}_\theta(\mathbf{a}_t \mid \mathbf{s}_t) = \pi_\theta(a_{t0} \mid s_{t0})\,\pi_\theta(a_{t1} \mid s_{t1})\dots\pi_\theta\left(a_{t(n_t-1)} \mid s_{t(n_t-1)}\right).$$

It follows that the outer policy gradient estimator

$$\hat{\mathbb{E}}_t\left[\nabla_\theta \log \boldsymbol{\pi}_\theta(\mathbf{a}_t \mid \mathbf{s}_t)\,\hat{\mathbf{A}}_t\right] = \hat{\mathbb{E}}_t\left[\sum_{u=0}^{n_t-1} \nabla_\theta \log \pi_\theta(a_{tu} \mid s_{tu})\,\hat{\mathbf{A}}_t\right] = \hat{\mathbb{E}}_t[n_t]\,\hat{\mathbb{E}}_{tu}\left[\nabla_\theta \log \pi_\theta(a_{tu} \mid s_{tu})\,\hat{\mathbf{A}}_t\right],$$

where $\hat{\mathbf{A}}_t$ is the outer advantage estimator. Hence the outer policy gradient estimator can be viewed (up to a global constant) as a special case of the inner policy gradient estimator by using the inner advantage estimator $\hat{A}_{tu} = \hat{\mathbf{A}}_t$ for all u. So there is no loss of generality by considering only the inner policy gradient estimator, as long as we allow non-standard inner advantage estimators.

Two natural choices for the inner advantage estimator are standard GAE, which we will call *inner GAE*, and the estimator obtained by using GAE with the outer policy gradient estimator, which we will call *outer GAE*. We would like to view both of these as special cases of a more general advantage estimator. To do this,

we introduce Double-GAE $(\gamma_{\text{in}}, \gamma_{\text{out}}, \lambda_{\text{in}}, \lambda_{\text{out}})$, which is parameterized by inner and outer discount rates γ_{in} and γ_{out} and bootstrapping parameters λ_{in} and λ_{out}. First we define the one-step backups

$$\delta^V_{tu} := \begin{cases} -V(s_{tu}) + r_{tu} + \gamma_{\text{in}} V\left(s_{t(u+1)}\right), & u < n_t - 1, \\ -V(s_{tu}) + r_{tu} + \gamma_{\text{out}} \gamma_{\text{in}}^{-(n_t-1)} V\left(s_{(t+1)0}\right), & u = n_t - 1. \end{cases}$$

Using these, we define the Double-GAE advantage estimator by

$$\hat{A}_{tu} := \sum_{k=0}^{\infty} \sum_{v=*}^{n_{t+k}-1} (\gamma_{\text{out}}\lambda_{\text{out}})^k (\gamma_{\text{in}}\lambda_{\text{in}})^{v-u} \delta^V_{(t+k)v}, \qquad \text{where} \qquad * := \begin{cases} u, & k = 0 \\ 0, & k > 0. \end{cases}$$

To recover inner GAE, we must assume that $n_t = n$ is constant. If moreover $\gamma_{\text{out}} = \gamma_{\text{in}}^n$ and $\lambda_{\text{out}} = \lambda_{\text{in}}^n$, then δ^V_{tu} reduces to the standard inner one-step backup, and $\hat{A}_{tu}$ reduces to the standard inner GAE estimator. Hence Double-GAE $(\gamma_{\text{in}}, \gamma_{\text{in}}^n, \lambda_{\text{in}}, \lambda_{\text{in}}^n)$ is equivalent to inner GAE $(\gamma_{\text{in}}, \lambda_{\text{in}})$.

To recover outer GAE, we must assume both that there are no inner-only rewards, and that we only have an outer value function $\mathbf{V}$, from which we obtain our inner value function by taking $V(s_{tu}) = \mathbf{V}(\mathbf{s}_t)$. If moreover $\gamma_{\text{in}} = \lambda_{\text{in}} = 1$, then $\delta^V_{tu} = 0$ for $u < n_t - 1$ while $\delta^V_{t(n_t-1)}$ reduces to the standard outer one-step backup, and $\hat{A}_{tu}$ becomes independent of u, reducing to the standard outer GAE estimator for step t. It follows that inner Double-GAE $(1, \gamma_{\text{out}}, 1, \lambda_{\text{out}})$ is equivalent (up to a global constant) to outer GAE $(\gamma_{\text{out}}, \lambda_{\text{out}})$.

We conclude from this that Double-GAE is general enough to describe most inner advantage estimators we are likely to care about, while keeping the number of hyperparameters minimal. Even though in practice n_t may not be constant, and we may consider it unhelpful to use a value function that can only differentiate between outer states, we have hopefully captured the most important features of any useful estimators.

3 Simplifying Double-GAE

We can split the double sum in the definition of the Double-GAE advantage estimator into a single sum and a double sum whose inner terms do not depend on u:

$$\begin{aligned} \hat{A}_{tu} &= \sum_{l=0}^{n_t-1-u} (\gamma_{\text{in}}\lambda_{\text{in}})^l \delta^V_{t(u+l)} &+ (\gamma_{\text{in}}\lambda_{\text{in}})^{-u} \sum_{k=1}^{\infty} (\gamma_{\text{out}}\lambda_{\text{out}})^k \sum_{v=0}^{n_{t+k}-1} (\gamma_{\text{in}}\lambda_{\text{in}})^v \delta^V_{(t+k)v} \\ &= \hat{A}^{\text{comp}}_{tu} &+ (\gamma_{\text{in}}\lambda_{\text{in}})^{-u} \sum_{k=1}^{\infty} (\gamma_{\text{out}}\lambda_{\text{out}})^k \hat{A}^{\text{comp}}_{(t+k)0}, \end{aligned}$$

where $\hat{A}^{\text{comp}}_{tu}$ is defined by

$$\hat{A}^{\text{comp}}_{tu} := \sum_{l=0}^{n_t-1-u} (\gamma_{\text{in}}\lambda_{\text{in}})^l \delta^V_{t(u+l)}.$$

$\hat{A}^{\text{comp}}_{tu}$ is similar to the standard GAE estimator truncated after the end of the current completion, the only difference being the presence of γ_{out} in $\delta^V_{t(n_t-1)}$.

If $\gamma_{\text{in}} = \lambda_{\text{in}} = 1$, then $\hat{A}^{\text{comp}}_{tu}$ simplifies to

$$\hat{A}^{\text{comp}}_{tu} = -V(s_{tu}) + r_{tu} + r_{t(u+1)} + \cdots + r_{t(n_t-1)} + \gamma_{\text{out}} V\left(s_{(t+1)0}\right).$$

In particular, $\hat{A}^{\text{comp}}_{t0}$ is simply the standard outer one-step backup,

$$\hat{A}^{\text{comp}}_{t0} = \boldsymbol{\delta}^V_t := -V(\mathbf{s}_t) + \mathbf{r}_t + \gamma_{\text{out}} V(\mathbf{s}_{t+1}).$$

Hence $\hat{A}_{tu}$ simplifies to

$$\hat{A}_{tu} = -\left[-V(s_{t0}) + r_{t0} + r_{t1} + \cdots + r_{t(u-1)} + V(s_{tu})\right] + \sum_{k=0}^{\infty} (\gamma_{\text{out}}\lambda_{\text{out}})^k \boldsymbol{\delta}^V_{t+k}.$$

This is almost identical to standard outer GAE, differing only by the u-step backup correction term in square brackets.

4 Choice of hyperparameters

It would be perverse for $\gamma_{\text{in}}^{n_t}$ to be much smaller than γ_{out}, since that would involve putting more weight on rewards further into the future. Furthermore, if completions are not particularly long, then taking γ_{in} to be slightly less than 1 will not provide much variance reduction anyway. This suggests that taking $\gamma_{\text{in}} = 1$ is a safe choice.

One could argue similarly for taking $\lambda_{\text{in}} = 1$, which would allow us to use the simplified calculation. However, λ is typically much lower than γ, so this may be a poor choice when completions are longer than a few tens of tokens, say.

It is hard to say much more than this without empirical data, but our overall recommendations are:

- For completions of just a few tokens, take $\gamma_{\text{in}} = \lambda_{\text{in}} = 1$ (enabling the simplified calculation), and use reasonable defaults for γ_{out} and λ_{out} such as $\gamma_{\text{out}} = 0.999$, $\lambda_{\text{out}} = 0.95$ or a little lower.
- For completions much longer than a few tens of tokens, consider significantly lowering λ_{out} and taking $\lambda_{\text{in}} = \lambda_{\text{out}}^{1/n}$, where n is the number of tokens in a typical completion. It may also help to lower γ_{out}.
- For completions of an intermediate length, the first option might work fine, perhaps with lowering λ_{out}, but the second option could be worth trying.

References

J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.