Supplementary Material

January 14, 2020

1 Evolution: from Reinforce Algorithm to Advantage Actor Critic Algorithm

Recent years, many works have applied the Reinforce Algorithm to improve a Seq2Seq model. [3] applied the policy gradient to improve their Seq2Seq models and updated the model by gradient:

$$\sum_{n=1}^{N} \frac{1}{\Pr(\hat{Y}^n|X^n)} \frac{d\ \Pr(\hat{Y}^n|X^n)}{d\theta} R(\hat{Y}^n) \tag{1.1}$$

$$= \sum_{n=1}^{N} \frac{d\ \log \Pr(\hat{Y}^n|X^n)}{d\theta} R(\hat{Y}^n) \tag{1.2}$$

Where $\Pr(\hat{Y}^n|X^n)$ is the probability of generating the n^{th} sequence and N is the number of samples in current batch (for the mini-batch gradient descent). $\frac{1}{\Pr(\hat{Y}^n|X^n)}$ is introduced to correct the sampling bias for $\hat{Y}^n$ (named 'log-derivative").

The sequence-level updating generates lots of variance during training. We can define a word level Reinforce algorithm. [2] introduced the reinforce learning rule for words by:

$$\sum_{n=1}^{N}\sum_{t=1}^{T} \frac{d\ \log \Pr(\hat{y}_t^n|\hat{Y}_{1\dots t-1}^n, X^n)}{d\theta} \Big[\sum_{\gamma=t}^{T} r(\hat{y}_\gamma^n; \hat{Y}_{1\dots\gamma-1}^n) - b_t(X)\Big] \tag{1.3}$$

Where $b_t(X)$ is the baseline function and $r(\hat{y}_\gamma^n; \hat{Y}_{1\dots\gamma-1}^n)$ is the reward for the predicted word $\hat{y}_\gamma^n$. However, this method evaluates only the predicted words instead of all candidate words. To provide a more comprehensive evaluation, [1] replaced the current 1-sample estimate to $\mathcal{A}$-sample estimate ($\mathcal{A}$ is an action space containing all candidate words). The gradient function becomes:

$$\sum_{n=1}^{N}\sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \frac{d\pi_A(a|\hat{Y}_{1\dots t-1}, X)}{d\theta} Q(a, \hat{Y}_{1\dots t-1}^n, Y^n) \tag{1.4}$$

Where Q function computes the Expected Cumulative Reward ($\mathbb{E}(\sum_{t=\gamma}^{T} r[\hat{y}_\gamma^n; \hat{Y}_{1\dots\gamma-1}^n)]$) after selecting word $\hat{y}_\gamma^n$ at state $(\hat{Y}_{1\dots\gamma-1}^n, X^n)$. In our paper, We apply an Critic model to estimate the Q function by minimizing the Temporal Difference (TD) error. Furthermore, in order to provide more accurate evaluation, instead of directly including Q function into gradient computation, we apply an advantage function which subtracts an expected state value from current the Q function, which significantly reduces training variance.

2 Proof of Claim 1

We prove the claim 1 by firstly proving that the gradient estimate is unbiased (step 1) and then demonstrating that advantage reduces the training variance (step 2):

Step 1: we prove that the gradient estimate computed by our Actor-Critic algorithm is unbiased.

Note that the Seq2Seq model applies the discrete action space $\mathcal{A}$ (vocabulary set) and a finite-time zone T (length of the predicted sequence). The gradient estimate of our Actor is:

$$\mathbb{E}\Big\{\sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \pi_A(a|\hat{Y}_{1\dots t-1}, X) A^{\pi_A}(a; \hat{Y}_{1\dots t-1}^n, Y^n)\Big\}$$

$$= \mathbb{E}\Big\{\nabla_\theta \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \pi_A(a|\hat{Y}_{1\dots t-1}, X)\Big[\hat{Q}_C(a; \hat{Y}_{1\dots t-1}^n, Y) - \hat{V}_C(\hat{Y}_{1\dots t-1}^n, Y$$

$$= \mathbb{E}\Big[\nabla_\theta \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \pi_A(a|\hat{Y}_{1\dots t-1}, X)\hat{Q}_C(a; \hat{Y}_{1\dots t-1}^n, Y)\Big]$$

$$- \mathbb{E}\Big[\nabla_\theta \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \pi_A(a|\hat{Y}_{1\dots t-1}, X)\hat{V}_C(\hat{Y}_{1\dots t-1}^n, Y)\Big]$$

For any predicting step t, the gradient of $\sum_{a\in\mathcal{A}} \pi_A(a|\hat{Y}_{1\dots t-1}, X)$ multiplied with $\hat{V}_C(\hat{Y}_{1\dots t-1}^n, Y)$ is

zero. This is because:

$$\mathbb{E}\Big[\nabla_\theta \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \pi_A(a|\hat{Y}_{1\ldots t-1}, X)\hat{V}_C(\hat{Y}^n_{1\ldots t-1}, Y)\Big]$$
$$= \mathbb{E}\Big\{\sum_{t=1}^{T}\nabla_\theta\Big[\sum_{a\in\mathcal{A}}\pi_A(a|\hat{Y}_{1\ldots t-1}, X)\Big]\hat{V}_C(\hat{Y}^n_{1\ldots t-1}, Y)\Big\}$$
$$= \mathbb{E}\Big[\sum_{t=1}^{T}\nabla_\theta \cdot 1 \cdot \hat{V}_C(\hat{Y}^n_{1\ldots t-1}, Y)\Big]$$
$$= \mathbb{E}\Big[\sum_{t=1}^{T} 0 \cdot \hat{V}_C(\hat{Y}^n_{1\ldots t-1}, Y)\Big]$$

Subtracting $\hat{V}_C(\hat{Y}^n_{1\ldots t-1}, Y)$ from $\hat{Q}_C(a; \hat{Y}^n_{1\ldots t-1}, Y)$ will not influence the estimate of gradient during training and thus the gradient estimate of our Actor is unbiased.

Step 2: we prove that applying the advantage (Reducing the state value $\hat{V}_C$ from $\hat{Q}_C$) will decrease the variance of the gradient estimate during the training.

Proof: our model applies discrete action space and finite-time zone. The training variance of the gradient estimate is:

$$Var\Big[\nabla_\theta \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \pi_A(a|\hat{Y}_{1\ldots t-1}, X)A^{\pi_A}(a; \hat{Y}^n_{1\ldots t-1}, Y^n)\Big]$$
$$\overset{(i)}{=} \mathbb{E}\Big\{\Big[\nabla_\theta \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \pi_A(a|\hat{Y}_{1\ldots t-1}, X)A^{\pi_A}(a; \hat{Y}^n_{1\ldots t-1}, Y^n)\Big]^2\Big\}$$
$$- \mathbb{E}\Big\{\Big[\nabla_\theta \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \pi_A(a|\hat{Y}_{1\ldots t-1}, X)A^{\pi_A}(a; \hat{Y}^n_{1\ldots t-1}, Y^n)\Big]\Big\}^2$$

Equal (i) uses the definition of the variance $Var(X) = E(X^2) - [E(X)]^2$. Claim 1 shows the advantage causes no bias to the second expectation. We consider only the first expectation:

$$\mathbb{E}\Big[\nabla_\theta \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \pi_A(a|\hat{Y}_{1\ldots t-1}, X)A^{\pi_A}(a; \hat{Y}^n_{1\ldots t-1}, Y^n)\Big]^2$$
$$\overset{(ii)}{\approx} \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \mathbb{E}\Big[\nabla_\theta \pi_A(a|\hat{Y}_{1\ldots t-1}, X)A^{\pi_A}(a; \hat{Y}^n_{1\ldots t-1}, Y^n)\Big]^2$$
$$\overset{(iii)}{\approx} \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \mathbb{E}\Big[\nabla_\theta \pi_A(a|\hat{Y}_{1\ldots t-1}, X)\Big]^2 \cdot \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \mathbb{E}\Big[A^{\pi_A}(a; \hat{Y}^n_{1\ldots t-1}, Y^n)\Big]^2$$
$$= \sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \mathbb{E}\Big[\nabla_\theta \sum_{a\in\mathcal{A}} \pi_A(a|\hat{Y}_{1\ldots t-1}, X)\Big]^2 \cdot$$
$$\sum_{t=1}^{T}\sum_{a\in\mathcal{A}} \mathbb{E}\Big[\hat{Q}_C(a; \hat{Y}^n_{1\ldots t-1}, Y) - \hat{V}_C(\hat{Y}^n_{1\ldots t-1}, Y)\Big]^2$$

Approximation (ii) is where we approximate the variance of the sums by the sums of the variances. Approximation (iii) holds as we assume independence among the values involved in the expectation.

To reduce the training variance, we must minimize the term $\mathbb{E}\{[\hat{Q}_C(a; \hat{Y}^n_{1\ldots t-1}, Y) - \hat{V}_C(\hat{Y}^n_{1\ldots t-1}, Y)]^2\}$. It is similar to solve a least square problem. We find the state value applied by the advantage $\hat{V}_C(\hat{Y}^n_{1\ldots t-1}, Y) = \mathbb{E}_{a\sim\mathcal{A}}(\hat{Q}_C[a; \hat{Y}^n_{1\ldots t-1}, Y)]$ is an optimal solution to this problem. So, by reducing $\hat{V}_C(\hat{Y}^n_{1\ldots t-1}, Y)$ from $\hat{Q}_C(a; \hat{Y}^n_{1\ldots t-1}, Y)$, we reduce the variance of gradient estimate.

3 Significant Test for Recall, Precision and F1-Score

To assess whether the quality of the sentences predicted by our CE+AC model are significantly different from that from the comparison methods, we conduct a paired t-test for the precision, recall and f1-score (applied to evaluate our Actor). The results are reported in Table 1, Table 2 and Table 3. We find when compared to the Logician, AC and ME+AC, the Precision, the Recall and the F1-Score computed with our CE+AC model are significantly different for both beam search and greedy search and for all exploration widths. But when compared to R-Logician, the predictions from CE+AC model have the similar Recall. Because for the Facts that have not appeared in the predictions of R-Logician, our model has limited ability to generate them.

Table 1: P-value of significant test between our CE+AC(W2) model and the comparison methods for both beam search and greedy search.**Exploration width=2**

	Beam Search			Greedy Search		
Model	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Logician	8.52E-6	3.30E-2	9.76E-4	1.07E-9	1.38E-4	5.25E-7
R-Logician	8.76E-3	8.21E-1	1.70E-1	1.15E-3	1.23E-1	1.60E-2
AC	1.03E-5	2.16E-2	8.14E-4	1.49E-8	1.65E-3	8.31E-6
ME+AC	2.95E-5	7.07E-2	3.02E-3	1.33E-8	3.13E-4	2.55E-6

Table 2: P-value of significant test between our CE+AC(W4) model and the comparison methods for both beam search and greedy search.**Exploration width=4**

	Beam Search			Greedy Search		
Model	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Logician	1.52E-3	4.73E-2	9.05E-3	1.47E-13	1.60E-10	2.31E-12
R-Logician	1.44E-1	8.77E-1	4.32E-1	1.52E-5	6.47E-5	2.37E-5
AC	1.48E-3	2.89E-2	6.83E-3	3.61E-12	1.57E-8	1.44E-10
ME+AC	2.66E-3	9.16E-2	1.84E-2	6.63E-13	3.06E-10	7.34E-12

Table 3: P-value of significant test between our CE+AC(W2) model and the comparison methods for both beam search and greedy search.**Exploration width=6**

	Beam Search			Greedy Search		
Model	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Logician	1.60E-3	4.57E-2	9.04E-3	5.95E-12	5.33E-10	3.54E-11
R-Logician	2.02E-1	9.75E-1	5.41E-1	9.62E-5	1.04E-4	9.07E-5
AC	1.51E-3	2.75E-2	6.69E-3	1.18E-10	4.39E-8	1.75E-9
ME+AC	2.71E-3	8.75E-2	1.80E-2	2.34E-11	9.77E-10	1.03E-10

References

[1] Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, et al. An Actor-Critic Algorithm for Sequence Prediction. *CoRR*, abs/1607.07086, 2016.

[2] Marc'Aurelio Ranzato, Sumit Chopra, et al. Sequence Level Training with Recurrent Neural Networks. *CoRR*, abs/1511.06732, 2015.

[3] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *AAAI*, pages 2852–2858, 2017.