AN ACTOR-CRITIC ALGORITHM FOR SEQUENCE PREDICTION

Dzmitry Bahdanau **Philemon Brakel**
Kelvin Xu **Anirudh Goyal**
Université de Montréal

Ryan Lowe **Joelle Pineau**[*]
McGill University

Aaron Courville[†]
Université de Montréal

Yoshua Bengio[*]
Université de Montréal

Motivation

$$
\begin{aligned}
y_t &\sim g(s_{t-1}, c_{t-1}) \\
s_t &= f(s_{t-1}, c_{t-1}, e(y_t)) \\
\alpha_t &= \beta(s_t, (h_1, \ldots, h_L)) \\
c_t &= \sum_{j=1}^{L} \alpha_{t,j} h_j
\end{aligned}
$$

- teacher forcing
- discrepancy between training and testing conditions
- directly improve the test time metrics (Reward)

$$R\left(\hat{Y}, Y\right) = \sum_{t=1}^{T} r_t \left(\hat{y}; \hat{Y}_{1 \ldots t-1}, Y\right)$$

Reward shaping

$$\left(R\left(\hat{Y}_{1\ldots1}\right), R\left(\hat{Y}_{1\ldots2}\right), \ldots, R\left(\hat{Y}_{1\ldots T}\right)\right)$$

$$r_t\left(\hat{y}_t; \hat{Y}_{1\ldots t-1}\right) = R\left(\hat{Y}_{1\ldots t}\right) - R\left(\hat{Y}_{1\ldots t-1}\right)$$

Value Functions

We define the value of an unfinished prediction $\hat{Y}_{1\ldots t}$ as follows:

$$V(\hat{Y}_{1\ldots t}; X, Y) = \underset{\hat{Y}_{t+1\ldots T} \sim p(.|\hat{Y}_{1\ldots t}, X)}{\mathbb{E}} \sum_{\tau=t+1}^{T} r_\tau(\hat{y}_\tau; \hat{Y}_{1\ldots\tau-1}, Y).$$

We define the value of a candidate next token a for an unfinished prediction $\hat{Y}_{1\ldots t-1}$ as the expected future return after generating token a:

$$Q(a; \hat{Y}_{1\ldots t-1}, X, Y) = \underset{\hat{Y}_{t+1\ldots T} \sim p(.|\hat{Y}_{1\ldots t-1} a, X)}{\mathbb{E}} \left(r_t(a; \hat{Y}_{1\ldots t-1}, Y) + \sum_{\tau=t+1}^{T} r_\tau(\hat{y}_\tau; \hat{Y}_{1\ldots t-1} a \hat{Y}_{t+1\ldots\tau}, Y) \right)$$

$$
\begin{aligned}
\frac{dV}{d\theta} = \frac{d}{d\theta} \mathop{\mathbb{E}}_{\hat{Y} \sim p(\hat{Y})} R(\hat{Y}) = \sum_{\hat{Y}} \frac{d}{d\theta} \left[p(\hat{y}_1) p(\hat{y}_2|\hat{y}_1) \ldots p(\hat{y}_T|\hat{y}_1 \ldots \hat{y}_{T-1}) \right] R(\hat{Y}) = \\
\sum_{t=1}^{T} \sum_{\hat{Y}} p(\hat{Y}_{1\ldots t-1}) \frac{dp(\hat{y}_t|\hat{Y}_{1\ldots t-1})}{d\theta} p(\hat{Y}_{t+1..T}|\hat{Y}_{1\ldots t}) R(\hat{Y}) = \\
\sum_{t=1}^{T} \sum_{\hat{Y}_{1\ldots t}} p(\hat{Y}_{1\ldots t-1}) \frac{dp(\hat{y}_t|\hat{Y}_{1\ldots t-1})}{d\theta} \sum_{\hat{Y}_{t+1\ldots T}} p(\hat{Y}_{t+1..T}|\hat{Y}_{1\ldots t}) \sum_{\tau=1}^{T} r_\tau(\hat{y}_\tau; \hat{Y}_{1\ldots \tau-1}) = \\
\sum_{t=1}^{T} \sum_{\hat{Y}_{1\ldots t}} p(\hat{Y}_{1\ldots t-1}) \frac{dp(\hat{y}_t|\hat{Y}_{1\ldots t-1})}{d\theta} \\
\left[r_t(\hat{y}_t; \hat{Y}_{1\ldots t-1}) + \sum_{\hat{Y}_{t+1\ldots T}} p(\hat{Y}_{t+1..T}|\hat{Y}_{1\ldots t}) \sum_{\tau=t+1}^{T} r_\tau(\hat{y}_\tau; \hat{Y}_{1\ldots \tau-1}) \right] = \\
\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\hat{Y}_{1\ldots t-1} \sim p(\hat{Y}_{1\ldots t-1})} \sum_{a \in A} \frac{dp(a|\hat{Y}_{1\ldots t-1})}{d\theta} Q(a; \hat{Y}_{1\ldots t-1}) = \\
\mathop{\mathbb{E}}_{\hat{Y} \sim p(\hat{Y})} \sum_{t=1}^{T} \sum_{a \in \mathcal{A}} \frac{dp(a|\hat{Y}_{1\ldots t-1})}{d\theta} Q(a; \hat{Y}_{1\ldots t-1})
\end{aligned}
$$

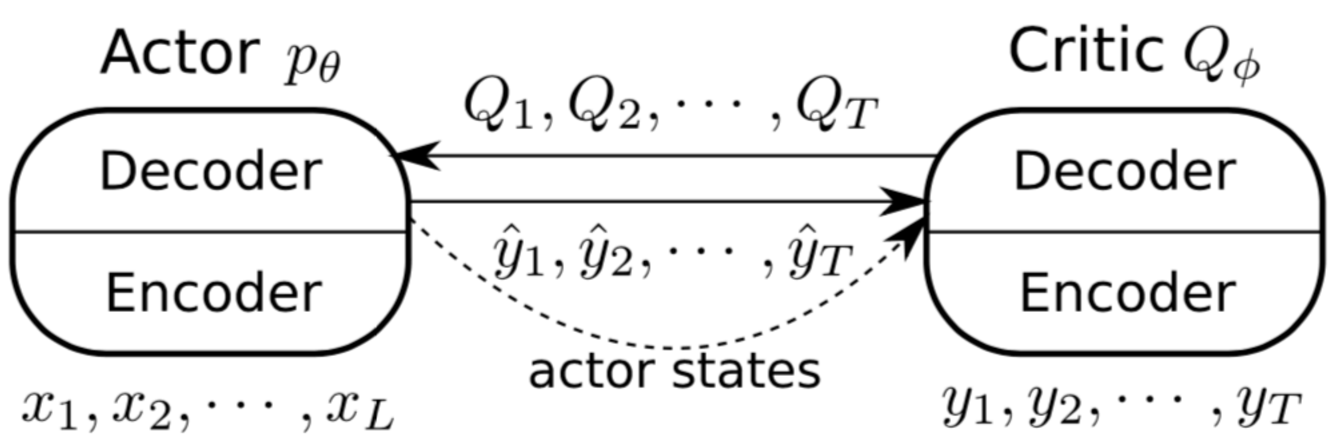
ACTOR-CRITIC FOR SEQUENCE PREDICTION

$$\frac{dV}{d\theta} = \mathop{\mathbb{E}}_{\hat{Y} \sim p(\hat{Y}|X)} \sum_{t=1}^{T} \sum_{a \in \mathcal{A}} \frac{dp(a|\hat{Y}_{1\ldots t-1})}{d\theta} Q(a; \hat{Y}_{1\ldots t-1}).$$

$$\widehat{\frac{dV}{d\theta}} = \sum_{k=1}^{M} \sum_{t=1}^{T} \sum_{a \in \mathcal{A}} \frac{dp(a|\hat{Y}^k_{1\ldots t-1})}{d\theta} Q(a; \hat{Y}^k_{1\ldots t-1})$$

$$\widehat{\frac{dV}{d\theta}} = \sum_{k=1}^{M} \sum_{t=1}^{T} \frac{d \log p(\hat{y}^k_t|\hat{Y}^k_{1\ldots t-1})}{d\theta} \left[\sum_{\tau=t}^{T} r_\tau(\hat{y}^k_\tau; \hat{Y}^k_{1\ldots \tau-1}) - b_t(X) \right]$$

ACTOR-CRITIC FOR SEQUENCE PREDICTION



Temporal-difference learning

$$\hat{Q}\left(\hat{y}_t; \hat{Y}_{1\ldots t-1}\right)$$

naïve Monte-Carlo	$\sum_{\tau=t}^{T} r_\tau \left(\hat{y}_\tau; \hat{Y}_{1,\ldots,\tau-1}\right)$
temporal difference (TD)	$q_t = r_t\left(\hat{y}_t; \hat{Y}_{1\ldots t-1}\right) + \sum_{a\in A} p\left(a \mid \hat{Y}_{1\ldots t}\right) \hat{Q}\left(a; \hat{Y}_{1\ldots t}\right)$

Applying deep RL techniques

- When *critic* $\hat{Q}$ is non linear, the TD policy evaluation might diverge.
- Using a *target network* $\hat{Q}'$ to compute q_t, which is updated more slowly than $\hat{Q}$.
- Sample from a delayed *actor*, whose weights are slowly updated to follow the actor that is actually trained.

Dealing with large action spaces

$$C_t = \sum_a \left(\hat{Q}(a; \hat{Y}_{1\ldots t-1}) - \frac{1}{|\mathcal{A}|} \sum_b \hat{Q}(b; \hat{Y}_{1\ldots t-1}) \right)^2$$

Algorithm 1 Actor-Critic Training for Sequence Prediction

Require: A critic $\hat{Q}(a; \hat{Y}_{1 \dots t}, Y)$ and an actor $p(a|\hat{Y}_{1 \dots t}, X)$ with weights ϕ and θ respectively.

Initialize delayed actor $p^{'}$ and target critic $\hat{Q}'$ with same weights: $\theta' = \theta$, $\phi' = \phi$.

while Not Converged **do**

Receive a random example (X, Y).

Generate a sequence of actions $\hat{Y}$ from $p^{'}$.

Compute targets for the critic

$$q_t = r_t(\hat{y}_t; \hat{Y}_{1 \dots t-1}, Y) + \sum_{a \in \mathcal{A}} p^{'}(a|\hat{Y}_{1 \dots t}, X) \hat{Q}'(a; \hat{Y}_{1 \dots t}, Y)$$

Update the critic weights ϕ using the gradient

$$\frac{d}{d\phi}\left(\sum_{t=1}^{T}\left(\hat{Q}(\hat{y}_t; \hat{Y}_{1 \dots t-1}, Y) - q_t\right)^2 + \lambda_C C_t\right)$$

$$\text{where } C_t = \sum_{a}\left(\hat{Q}(a; \hat{Y}_{1 \dots t-1}) - \frac{1}{|\mathcal{A}|}\sum_{b} \hat{Q}(b; \hat{Y}_{1 \dots t-1})\right)^2$$

Update actor weights θ using the following gradient estimate

$$\frac{d\widehat{V(X, Y)}}{d\theta} = \sum_{t=1}^{T} \sum_{a \in \mathcal{A}} \frac{dp(a|\hat{Y}_{1 \dots t-1}, X)}{d\theta} \hat{Q}(a; \hat{Y}_{1 \dots t-1}, Y) + \lambda_{LL} \sum_{t=1}^{T} \frac{dp(y_t|Y_{1 \dots t-1}, X)}{d\theta}$$

Update delayed actor and target critic, with constants $\gamma_\theta \ll 1$, $\gamma_\phi \ll 1$

$$\theta' = \gamma_\theta \theta + (1 - \gamma_\theta)\theta', \; \phi' = \gamma_\phi \phi + (1 - \gamma_\phi)\phi'$$

end while

Algorithm 2 Complete Actor-Critic Algorithm for Sequence Prediction

1: Initialize critic $\hat{Q}(a; \hat{Y}_{1...t}, Y)$ and actor $p(a|\hat{Y}_{1...t}, X)$ with random weights ϕ and θ respectively.

2: Pre-train the actor to predict y_{t+1} given $Y_{1...t}$ by maximizing $\log p(y_{t+1}|Y_{1...t}, X)$.
3: Pre-train the critic to estimate Q by running Algorithm 1 with fixed actor.
4: Run Algorithm 1.

SPELLING CORRECTION

Table 1: Character error rate of different methods on the spelling correction task. In the table L is the length of input strings, η is the probability of replacing a character with a random one. LL stands for the log-likelihood training, AC and RF-C and for the actor-critic and the REINFORCE-critic respectively, AC+LL and RF-C+LL for the combinations of AC and RF-C with LL.

Setup	Character Error Rate				
	LL	AC	RF-C	AC+LL	RF-C+LL
$L = 10, \eta = 0.3$	17.81	17.24	17.82	**16.65**	16.97
$L = 30, \eta = 0.3$	18.4	17.31	18.16	**17.1**	17.47
$L = 10, \eta = 0.5$	38.12	35.89	35.84	**34.6**	35
$L = 30, \eta = 0.5$	40.87	37.0	37.6	**36.36**	36.6

Table 2: Our IWSLT 2014 machine translation results with a convolutional encoder compared to the previous work by Ranzato et al. Please see 1 for an explanation of abbreviations. The asterisk identifies results from (Ranzato et al., 2015). The numbers reported with $\leq$ were approximately read from Figure 6 of (Ranzato et al., 2015)

Decoding method	Model					
	LL*	MIXER*	LL	RF	RF-C	AC
greedy search	17.74	20.73	19.33	20.92	**22.24**	21.66
beam search	≤ 20.3	≤ 21.9	21.46	21.35	**22.58**	22.45

MACHINE TRANSLATION