Notes on NEURAL ARCHITECTURE SEARCH WITH REINFORCEMENT LEARNING
Barret Zoph, Quoc V. Le

David Meyer

dmm@{1-4-5.net,uoregon.edu,brocade.com,...}

November 28, 2016

Notes

- $\pi_\theta(s,a) = P[A_t = a | S_t = s, \theta] = P_{(a_1:T;\theta_c)}$ (policy parameterized by θ)
- $J(\theta_c) = \mathbb{E}_{P(a_1:T;\theta_c)}[R]$ (hook the RNN loss function to the RL reward)
- $J_1(\theta) = V^{\pi_\theta}(s_1) = \mathbb{E}_{\pi_\theta}[v_1] = \mathbb{E}_{P(a_1:T;\theta_c)}[R]$ (episodic environments)
- $\nabla_{\theta_c} J(\theta_c) = \sum\limits_{t=1}^{T} E_{P(a_1:T;\theta_c)} \big[\nabla_{\theta_c} \log P(a_t | a_{(t-1):1}; \theta_c) R\big]$ (REINFORCE policy gradient)
- a_t is the predicted action (a) and $a_{(t-1):1}$ is the state s up to step $t-1$ encoded in the RNN

1 Computing the gradient analytically

First, we assume that the policy π_θ is differentiable wherever it is non-zero (this is a softer requirement than requiring π_θ be differentiable *everywhere*). In addition, we know the gradient: $\nabla_\theta J(\theta)$. In this case, let $p(\mathbf{x};\theta)$ be the likelihood parametrized by θ and let $\log p(\mathbf{x};\theta)$ be the *log likelihood*. Then

$$
\begin{aligned}
y &= p(\mathbf{x};\theta) && \text{\# definition; see above} && (1)\\
z &= \log y = \log p(\mathbf{x};\theta) && \text{\# definition; z is the log likelihood} && (2)\\
\frac{dz}{d\theta} &= \frac{dz}{dy}\cdot\frac{dy}{d\theta} && \text{\# chain rule definition} && (3)\\
\frac{dz}{dy} &= \frac{1}{p(\mathbf{x};\theta)} && \#\ \frac{\log(X)}{dX} \approx \frac{1}{X} && (4)\\
\frac{dy}{d\theta} &= \frac{d\ p(\mathbf{x};\theta)}{d\theta} = \nabla_\theta p(\mathbf{x};\theta) && \text{\# definition (chain rule, again)} && (5)\\
\frac{dz}{d\theta} &= \frac{dz}{dy}\cdot\frac{dy}{d\theta} = \frac{\nabla_\theta\ p(\mathbf{x};\theta)}{p(\mathbf{x};\theta)} && \text{\# chain rule} && (6)\\
&= \nabla_\theta \log p(\mathbf{x};\theta) && \text{\# using the identity } \nabla_\theta \log(w) = \frac{1}{w}\nabla_\theta w && (7)
\end{aligned}
$$

and setting $w = p(\mathbf{x};\theta)$. Here $\nabla_\theta \log p(\mathbf{x};\theta)$ is known as the score or sometimes the *Fischer* information. So the *log derivative trick* (sometimes *likelihood ratio*) is

$$
\nabla_\theta \log p(\mathbf{x};\theta) = \frac{\nabla_\theta\ p(\mathbf{x};\theta)}{p(\mathbf{x};\theta)}
$$

Setting $\pi_\theta(s,a) = p(\mathbf{x};\theta)$ we see that

$$
\begin{aligned}
\nabla_\theta \pi_\theta(s,a) &= \pi_\theta(s,a)\frac{\nabla_\theta\ \pi_\theta(s,a)}{\pi_\theta(s,a)} && && (8)\\
&= \pi_\theta(s,a)\nabla_\theta \log \pi_\theta(s,a) && \text{\# log derivative trick} && (9)
\end{aligned}
$$

and the score function is $\nabla_\theta \log \pi_\theta(s,a)$.

Now, since here $\pi_\theta(s,a) = P(a_{1:T};\theta_c)$, we have

$$
\begin{aligned}
\nabla_{\theta_c} J(\theta_c) &= \sum_{s\in S} d(s) \sum_{a\in A} \nabla_{\theta_c}\pi_{\theta_c}(s,a)\mathcal{R}_{s,a} && \text{\# defn policy gradient} && (10)\\
&= \sum_{s\in S} d(s) \sum_{a\in A} \pi_{\theta_c}(s,a)\nabla \log \pi_{\theta_c}(s,a) R_{s,a} && \text{\# log derivative trick (Eqn 9)} && (11)\\
&= \mathbb{E}_{\pi_{\theta_c}}\big[\nabla_{\theta_c} \log \pi_{\theta_c}(s,a)R\big] && \text{\# defn expectation} && (12)\\
&= \mathbb{E}_{a_i\sim P}\big[\nabla_{\theta_c} \log P(a_t|a_{(t-1):1};\theta_c)R\big] && \#\ \pi_\theta(s,a) = P(a_{1:T};\theta_c) && (13)\\
&= \sum_{t=1}^{T} P_{(a_1:T;\theta_c)}\big[\nabla_{\theta_c} \log P(a_t|a_{(t-1):1};\theta_c)R\big] && \text{\# REINFORCE pg} && (14)
\end{aligned}
$$