TUNING RECURRENT NEURAL NETWORKS WITH REINFORCEMENT LEARNING

Natasha Jaques[12], **Shixiang Gu**[134], **Richard E. Turner**[3], **Douglas Eck**[1]
[1]Google Brain, USA
[2]Massachusetts Institute of Technology, USA
[3]University of Cambridge, UK
[4]Max Planck Institute for Intelligent Systems, Germany
jaquesn@mit.edu, sg717@cam.ac.uk, ret26@cam.ac.uk, deck@google.com

作者多么的 solid 啊!!

这篇文章是在音乐上面做的，也就是要产生好听的音乐。

然后就是训练 rnn，但是是用 rl 的技术优化的

首先看看我们的这个 DQN 的定义

π^* is known to satisfy the following Bellman optimality equation,

$$Q(s_t, a_t; \pi^*) = r(s_t, a_t) + \gamma \mathbb{E}_{p(s_{t+1}|s_t,a_t)}[\max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \pi^*)] \quad (1)$$

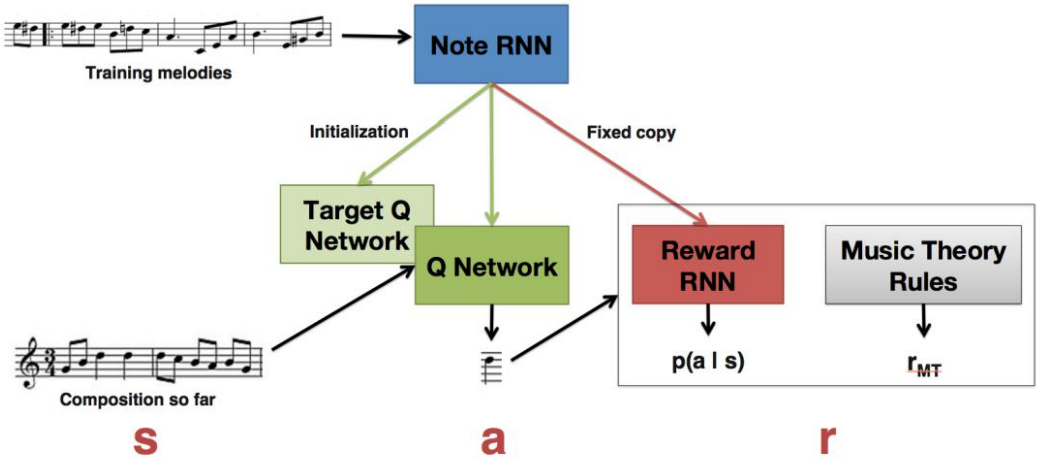
where $Q^\pi(s_t, a_t) = \mathbb{E}_\pi[\sum_{t'=t}^{\infty} \gamma^{t'-t} r(s_{t'}, a_{t'})]$ is the Q function of a policy π. Q-learning techniques (Watkins & Dayan, 1992; Sutton et al., 1999) learn this optimal Q function by iteratively minimizing the Bellman residual. The optimal policy is given by $\pi^*(a|s) = \arg\max_a Q(s, a)$. Deep Q-learning(Mnih et al., 2013) uses a neural network called the *deep Q-network* (DQN) to approximate the Q function $Q(s, a; \theta)$. The network parameters θ are learned by applying stochastic gradient descent (SGD) updates with respect to the following loss function,

$$L(\theta) = \mathbb{E}_\beta[(r(s, a) + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta))^2] \quad (2)$$

where β is the exploration policy, and θ^- is the parameters of the *Target Q-network* (Mnih et al.,

注意最后一个，那个 β 的是 exploration policy,其实最后就是一个值，在 Minh 的 DQN 里面是有一定概率随机选，还有一定概率是通过当前模型产生一个最优值，总之是一个数。

然后作者在这个里面也用了这些
总体模型如下图：



可以看到有一个 note-rnn 就是在以前的音乐模型上面用 lstm 训练一个语言模型类似的东西，然后把这个当成一个初始值发给 RL 里面的几个模型。

作者在这个里面定义了两个回报函数，其中一个是 rMT,也就是 music theory，也即从音乐的角度来看的回报，还有一个是 reward rnn，就是当前的这个动作在语言模型上面的概率。也就是一种数据统计角度的输出，这个东西以后就不变了。

然后 value-function 是用的 DDQN，有两个 model，都是用这个 not-rnn 初始化。
最后的回报函数：

$$r(s,a) = \log p(a|s) + r_{MT}(a,s)/c$$

最后的目标以及在测试的时候的策略为：

$$L(\theta) = \mathbb{E}_\beta[(\log p(a|s) + r_{MT}(a,s)/c + \gamma \max_{a'} Q(s',a';\theta^-) - Q(s,a;\theta))^2]$$
$$\pi_\theta(a|s) = \delta(a = \arg\max_a Q(s,a;\theta)).$$

作者在这个里面还介绍了一个比较牛逼的东西：
3.1 RELATIONSHIP TO STOCHASTIC OPTIMAL CONTROL （SOC）

$$\begin{aligned}\log p(\tau|b=1) &= \log \int p(\tau)p(b|\tau)d\tau \\ &\geq \mathbb{E}_{q(\tau)}[\log p(\tau)p(b|\tau) - \log q(\tau)] \\ &= \mathbb{E}_{q(\tau)}[r(\tau)/c - \mathbf{KL}[q(\tau)||p(\tau)]] = L_v(q)\end{aligned}$$

where $q(\tau)$ is the variational distribution. Rewriting the variational objective $L_v(q)$ in Eq. 6 in terms of policy π_θ, we get the following RL objective with KL-regularization,

$$L_v(\theta) = \mathbb{E}_\pi[\sum_t r(s_t,a_t)/c - KL[\pi_\theta(\cdot|s_t)||p(\cdot|s_t)]]. \quad (9)$$

In contrast, the objective in Section 3 is,

$$L_v(\theta) = \mathbb{E}_\pi[\sum_t r(s_t,a_t)/c + \log p(a_t|s_t)]. \quad (10)$$

可以看出来这个就是核心，也就是施加了一个对于目标 q 值以及我们策略 q 值的一个交叉熵。这样就是让我们的目标 q 值和策略 q 有相同的分布。
上面公式 10 就是本文以前的目标函数，但是这个里面的目标函数是除了这个 p(a|s)还有它对我们当前策略的交叉熵，也就是我们的 π 要是非常好的话那么他就应该完完全全符合当前的目标 q。并且使当前的目标 r(st,at)变得大。
所以对这个交叉熵，作者设计了两个新的目标函数

$$L(\theta) = \mathbb{E}_\beta[(\log p(a|s) + r_{MT}(s,a)/c + \gamma \log \sum_{a'} e^{\Psi(s',a';\theta^-)} - \Psi(s,a;\theta))^2]$$
$$\pi_\theta(a|s) \propto e^{\Psi(s,a;\theta)}$$

$$L(\theta) = \mathbb{E}_\beta[(r_{MT}/c(s,a) + \gamma \log \sum_{a'} e^{\log p(a'|s') + G(s',a';\theta^-)} - G(s,a;\theta))^2]$$
$$\pi_\theta(a|s) \propto p(a|s)e^{G(s,a;\theta)}.$$

这个都是从

Ψ-learning (Peters et al., 2010) and G-learning (Fox et al.)得来的。具体可以看定义

Metric	Note RNN	Q	Ψ	G
Notes excessively repeated	63.3%	**0.0%**	**0.02%**	**0.03%**
Mean autocorrelation - lag 1	-.16	**-.11**	**-.10**	.55
Mean autocorrelation - lag 2	.14	**.03**	**-.01**	.31
Mean autocorrelation - lag 3	-.13	**.03**	**.01**	17
Notes not in key	0.1%	1.00%	0.60%	28.7%
Compositions starting with tonic	0.9%	**28.8%**	**28.7%**	0.0%
Leaps resolved	77.2%	**91.1%**	**90.0%**	52.2%
Compositions with unique max note	64.7%	56.4%	59.4%	37.1%
Compositions with unique min note	49.4%	51.9%	**58.3%**	**56.5%**
Notes in motif	5.9%	**75.7%**	**73.8%**	**69.3%**
Notes in repeated motif	0.007%	**0.11%**	**0.09%**	0.01%

Table 1: Statistics of music theory rule adherence based on 100,000 randomly initialized compositions generated by each model. The top half of the table contains metrics that should be near zero, while the bottom half contains metrics that should increase. Bolded entries represent significant improvements over the *Note RNN* baseline.

最后的效果如图所示，可以看出来右边还是Ψ-learning 比较好。当然差的也不多

在这个文章的附录里面有 OFF-POLICY METHODS DERIVATIONS FOR KL-REGULARIZED REINFORCEMENT LEARNING

介绍那两个交叉熵的算法。