(۷) روش های Normalizing Flow یک شیوه دیگری از مدل های generative هستند

که برای حل مشکلات مدل های Autoregressive و VAE پیشنهاد شده است:

در اینجا برای بدست آوردن توزیع حاشیه ای p(x) داریم:

$$P_x(x;\theta) = P_z(f_\theta^{-1}(x)) \left| \det\left[\frac{\partial f_\theta^{-1}(x)}{\partial x}\right] \right| \quad *$$

نرمال در واقع از این می آید که تغییر متغیر یک پیاپی نرمال بسازیم تبدیل invertible

می دهد و flow بدین معناست که تبدیلات invertible می توانند با یکدیگر ترکیب شوند

که تبدیلات invertible (معکوس پذیر) پیچیده تری تشکیل شود حال این روش ها با استفاده از تبدیلات

transformation های پیاپی روی یک تابع توزیع کار می کنند) بر اساس قاعده * ژاکوبیان توابع توزیع ادامه

بیانی حساب می کنند و بدین ترتیب با استفاده از یک تابع ساده توابع توزیع پیچیده تر را

می سازند. تبدیل توزیع ها در این شبکه به روش بالا حساب می شود

ادامه * $\log P_i(z_i) = \log P_{i-1}(z_{i-1}) - \log\left(\det\left|\frac{\partial f_i}{\partial z_{i-1}}\right|\right)$

حال اگر * توابع اولیه خوب انتخاب شوند و اصطلاحا خوش فرم باشند می توانیم محاسبات

را به سادگی انجام دهیم. تابع Loss نیز در اینجا به صورت زیر تعریف می شود:

$$Loss = \frac{-1}{|S|} \sum_{x \in S} \log P(x)$$

بنابراین با توجه به این روابط و تابع Loss می توانیم شبکه را آموزش دهیم، backprop انجام

دهیم و پارامترهای مدل را بدست آوریم. ~~همچنین برای تولید نمونه از توزیع اولیه~~

همچنین برای تولید نمونه از توزیع اولیه که معمولا نرمال است یک نمونه‌گیری انجام می‌دهیم

سپس با احتمالی برابر تبدیل‌های $f_1, f_2, \ldots, f_k$ یک نمونه از توزیع مقصد بدست می‌آوریم.

~~این روش برای این که کاری را انجام بدهیم~~ (۸)

(۸) این روش این کار را انجام می‌دهد که به جای این که Q-learning را روی جفت state/action انجام دهیم میتیم کل وضعیت [state, action, reward, next_state] ها را در یک جدول بزرگ نگهداری می‌کند. در واقع فاز یادگیری از کسب تجربه جدا است و بر مبنای این است که از جدول نمونه برداری انجام دهیم.

~~در~~ در واقع این روش در الگوریتم‌های Deep Reinforcement استفاده می‌شود که بتوانیم از ارزش کل بدست آمده در تمامی حالات استفاده کنیم و مدل را به خوبی آموزش دهیم. البته این روش سعی می‌کند trade off بین تجربه کردن و استفاده از تجربه (exploration and exploitation) را با الگوریتم ε-greedy رعایت کند و نهایتاً به خروجی مطلوب تری نسبت به بقیه روش دست پیدا کند (این الگوریتم در DQN ها هم استفاده می‌شود)

فواید استفاده از این روش (۱) استفاده مؤثر از تجربیات قبلی یا یادگیری چندباره

(۲) converge کردن بهتر وقتی که نیاز به یک تخمین کردن کارایی داریم

معایب استفاده از روش‌های چند مرحله‌ای یادگیری مانند Q(λ) سخت‌تر کارایی می‌شود.