Adversarial Learning for Neural Dialogue Generation

Jiwei Li, WillMonroe, Tianlin Shi, S´ebastien Jean, Alan Ritter and Dan Jurafsky

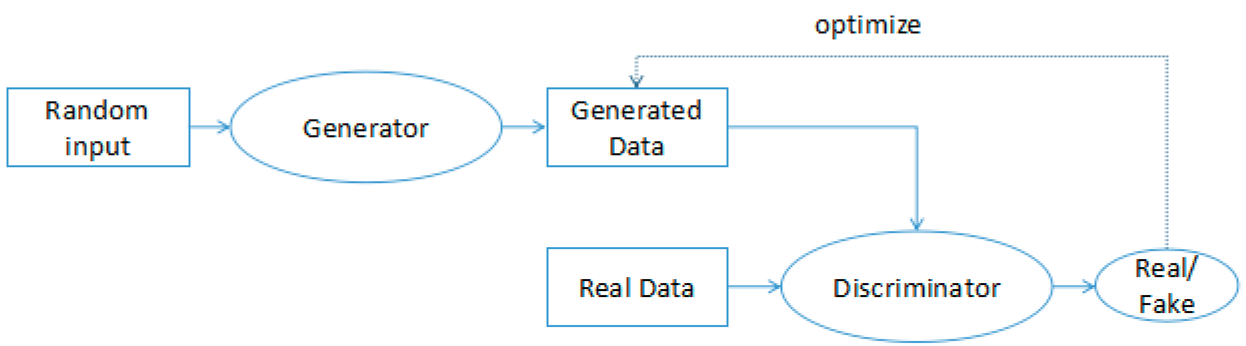
Motivation

GAN+RL to make the mechine generate sequence indistinguishable from human-generated dialogue, and can relief the training-hard of discrete text data problem.

Idea

A good dialogue model should generate uterances indistinguishable from human dialogues. Training two models, a generateor that defines the probability of generating a dialogue sequence, and a discriminator that labels dialogues as human-generated or machine-generated. The discriminator is analogous to the evaluator in the Turing test.

Adversarial Networks



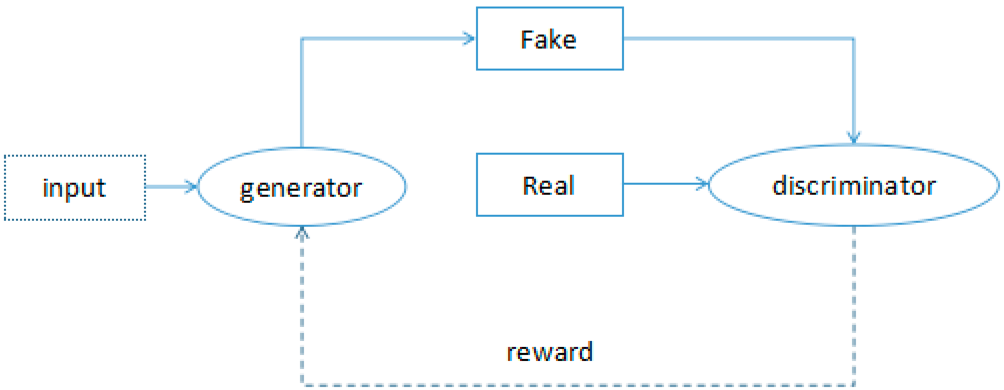
$$V(D,G) = E_{x \sim p_{data}}[logD(x)] + E_{x \sim p_g}[log(1 - D(G(x)))]$$

The optimization progress:

$$minmaxV(D,G)$$

For the discrete text data, the optimization for generator is difficult for it is not end-to-end.

Seq GAN (+RL)



The generator optimization objective becomes maximization the reward.Namely,

max

$$R = \frac{1}{N}\sum_{i=1}^{N} D(x^i) log P(x^i)$$

Adversarial Reinforce

Generative model: Seq2Seq model

Discriminative model: A binary classifier that outputs a label indicating wheter the input is generated by humans or machines.

Policy Gradient Training

According to the input x, the scores of generate the human-generated data is $Q_+(\{x,y\}|\theta)$
The bot generates a dialogue utterance y by sampling from the policy.

Baseline trick

因为这里的reward只有正值，没有负值，也就是只有奖励，没有惩罚。所以，我们对于所有的采样结果都是进行奖励优化。在采样不完全的情况下，这种操作很有可能是结果越来越偏。所以，要减去baseline，这样，我们可以对坏的结果进行惩罚，使得模型可以更快的找到好的方向。

Baseline Trck

$$\nabla J(\theta) \approx [Q_+(\{x, y\}) - b(\{x, y\})] \\ \nabla \log \pi(y|x) \\ = [Q_+(\{x, y\}) - b(\{x, y\})] \\ \nabla \sum_t \log p(y_t|x, y_{1:t-1})$$

这里，π代表生成根据输入x生成response的概率。baseline是一个预先训练好的discriminator模型，并不再进行更新。

Reward for Every Generation Step(REGS)---一人做事一人当

ex.
human-generated response：[I am John]
machine-generated response: [I don't know]
RL model 会对 "I don't know"三个词进行同样的惩罚。但是，"I"词也出现在了human-generated种，我们应该对这三个词区别对待。所以我们在计算reward的时候，应该在生成每个词都计算一个相应的reward，而不是用整体的reward表示生成的一句话。

解决策略

1.生成每个词的时候，进行采用，采取N个样本，把这个N个词与前面已经形成的子序列扔给discriminator进行打分。将打分结果进行能N均值化处理，得到该步的reward.
2.训练一个对子序列打分的discriminator。

REGS

这样我们可以得到:

$$\nabla J(\theta) \approx \sum_t (Q_+(x, Y_t) - b(x, Y_t)) \\ \nabla \log p(y_t | x, Y_{1:t-1})$$

可以看出，在每一步的reward值都是不一样的。

Teacher-Forcing

在早期generator训练时，generator很弱，生成的结果很烂，以至于discriminator产生的得分也会很低。这样generator无法得到good example的知道，很难知道什么才是好的结果。我们可以将Groud-Truth加入到generator中。

方法1：强行将Ground-Truth结果的reward置为1，这相当于最大化Ground-Truth的生成概率，即加入了MLE目标。

方法2：将Ground-Truth结果放入到discriminator中进行打分，根据打分结果将其作为reward。前提是要有个靠谱的discriminator。

Teacher-Forcing algrithom

For number of training iterations **do**
. **For** i=1,D-steps **do**
. Sample (X,Y) from real data
. Sample $\hat{Y} \sim G(\cdot|X)$
. Update D using (X, Y) as positive examples and $(X, \hat{Y})$ as negative examples.
. **End**
.
. **For** i=1,G-steps **do**
. Sample (X,Y) from real data
. Sample $\hat{Y} \sim G(\cdot|X)$
. Compute Reward r for $(X, \hat{Y})$ using D.
. Update G on $(X, \hat{Y})$ using reward r
. Teacher-Forcing: Update G on (X, Y)
. **End**
End

Adversarial Evaluation

训练另外一个Adversarial网络来对生成的response进行评估。如果discriminator的准确率很低，说明response生成的很类人。