Hw4 report – ChatBot with Reinforcement learning

R05943135 杯屎賴-

R05943135 江承恩 R05943011 沈恩禾
R05921016 傅鈞笙 R05942072 吳昭霆

環境

CPU	GPU	Memory	OS	Libraries
Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz	GTX 980	HYNIX HMT42GR7BFR4C-RD MEMORY 16GB * 8	Ubuntu 15.04 Mint 17.1 Rebecca	Tensorflow 1.0

Model 描述

我們使用 **SeqGAN**[1] 訓練事先訓練的 S2S model。Discriminator 之輸出 Logit 作為 generator 之 Reward function。在 Sample reward 的時候也使用 Monte Carlo Search 評估整句的 reward 分數

Fig. Generator

- Generator 為二普通單層 LSTM S2S model，分成 encoder，decoder 兩個部分。
- Input 與 Hidden state 皆為 256 維。
- 處理句子字數最長為 20，結尾以<pad> token 填補。
- 運作時先將問題句子輸入 encoder model，將最後產生的 state 輸入給 decoder 產生回答。

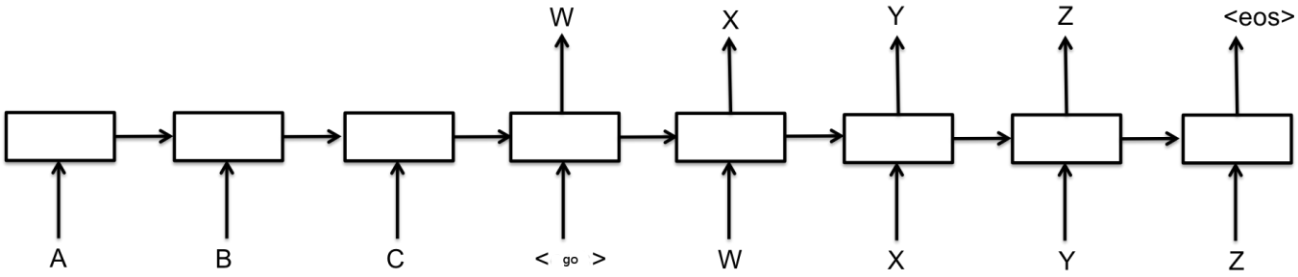


Fig. Generator

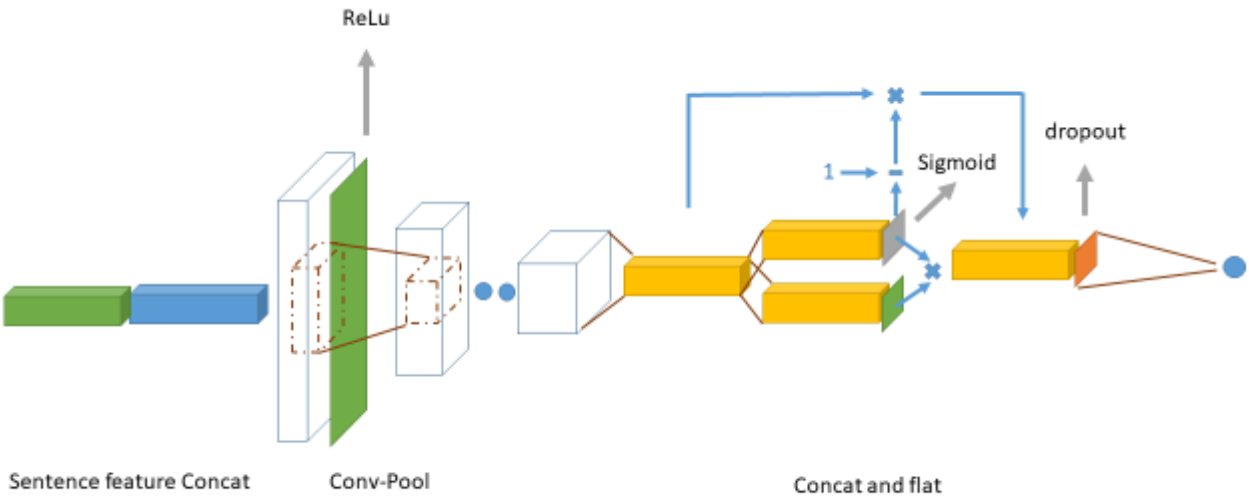


Fig. Discriminator

- Discriminator 為一 deep convolution binary classifier。
- 將問題與回答之 feature 相連，回答可為 dataset 中之真實對答，與 generator 產生之假回答。
- 通過 6 組 convolution + max pooling，filter shape 由 1 至 6，channel 數為 [100,200,200,200,200,100]。
- 使用 highway network，將結果調整後通過兩組 fully-connected layer，分別再通過 ReLu、Sigmoid，輸出$(1 - \mathrm{sigm}(\mathrm{h})) * \mathrm{h} + \mathrm{sigm}(\mathrm{h}) * \mathrm{relu}(\mathrm{h})$。
- 通過 dropout 後通過 fully-connected layer 產生最後 output logit。

Training Data 處理

Training data 僅擷取 Cornell Movie dataset 之對話的部分，共有 8 萬句。

- Vocabulary 字數嘗試 8000,10000,12000,20000。
- 擷取字數為 20 之內的句子，不足者填補<pad> token。

Training 參數

使用 Adam optimizer。

- **batch size**:64
- **learning rate:**0.02
- **Discriminator dropout rate**:0.75

前訓練

Pre-train 即普通 S2S model 之訓練與 convolution classifier 之訓練。

- 將 generator 在 Cornell movie dataset 上將問答傳入訓練 20 個 epoch。
- 將 movie dataset 之問題傳入 generator 產生回答，並將原回答與假回答傳入 discriminator 中訓練 3 個 epoch，接著重洗問答。重複這個過程 10 次。

Roll-out (Monte Carlo Search) GAN reward training

在 sample reward 的時候，拿取 16 組 batch，平均 Discriminator logit 之輸出作為 reward。而每一組 reward 須將句子每個 step 對於下一個 step 每個可能的 output 之機率平均作為此 step 之 reward，最後將 20 個 step 之 reward 平均。

$$\begin{aligned} x^A &= \text{I am John} & D(h^i, x^A) &= 1.0 \\ x^B &= \text{I am happy} & D(h^i, x^B) &= 0.1 \\ x^C &= \text{I don't know} & D(h^i, x^C) &= 0.1 \\ x^D &= \text{I am superman} & D(h^i, x^D) &= 0.8 \end{aligned} \quad \Big\} \; Q(h^i, \text{"I"}) = 0.5 \quad \text{avg}$$

$$generatorLoss: \frac{1}{16}\sum_{i=1}^{16}\sum_{t=1}^{20}\left(Q(h^i, x^i_{1:t})\right) * \nabla LogP_g(x^i_t | h^i, x^i_{1:t-1})$$

- 訓練時 Rollout 參數更新作法為$0.8 * \mathrm{w}_{rollout} + 0.2 * \mathrm{w}_{generator}$
- 訓練時，對於每個 training data set 中的問題，先訓練 Discriminator 5 個 epoch，再訓練 Generator.
- 訓練 Discriminator 時，使用 Generator 根據當前問題產生(3* batch size) 組取樣回答，將 training data 中的(問題, 回答) 當作 positive set，(問題, Generator 產生的回答) 當作 negative set 進行訓練。
- 訓練 Generator 時，使用 rollout 與 Discriminator，根據 **SeqGAN**[1] 論文中的公式更新 Generator 參數。

實驗

問題	回答(S2S)	回答(RL)
then screw you ill rot	the children are back	dont you do this
who is mr williams	just try the dressing room	who knows
everything okay	just a minute but your thoughts tune it	it doesnt matter

比較 S2S 與 RL (SeqGan) 方法產生的回答：
在訓練時，先訓練 Discriminator 再訓練 Generator。
在生成回答時，Generator 中 LSTM cell 的下一個 input 取法，測試使用 tf.argmax() 或是

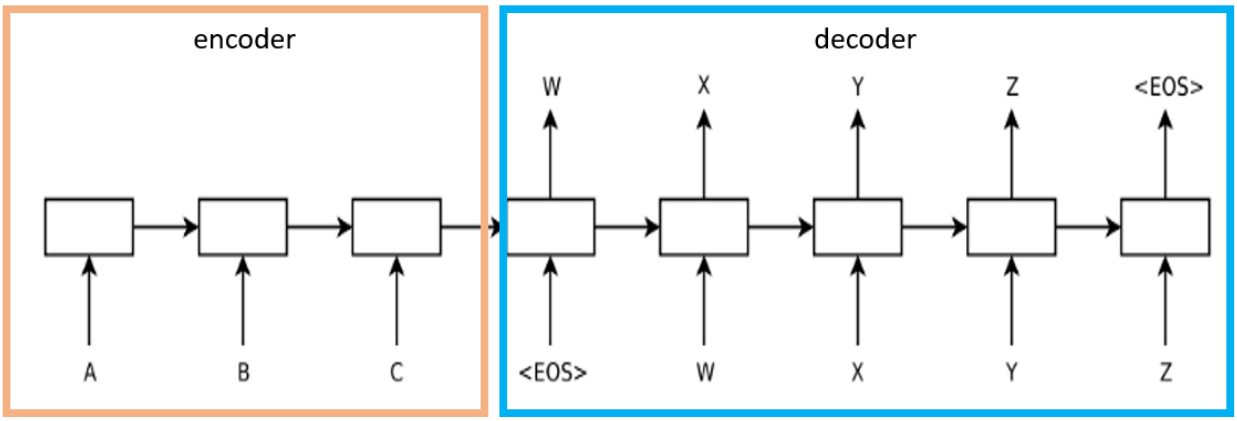
tf.multinomial()。
比較 tf.multinomial() 與 tf.argmax() 方法產生的回答：

問題	回答(tf.multinomial())	回答(tf.argmax())
then screw you ill rot	dont you do this	i dont know
who is mr williams	who knows	i dont know
everything okay	it doesnt matter	i dont know

- 先訓練 Discriminator 直覺上的意義是，對於同一個問題，先訓練 Discriminator 的分辨能力，再讓 Generator 去產生 Discriminator 無法分辨的回答。
- 使用 tf.argmax() 與 tf.multinomial() 的結果會在分析中說明。

分析

- 觀察發現回答可以產生普通合理之文法，但文不對題。判斷應該使用兩組 Lstm cell 參數，encoder、decoder 分別處理問答才能區分兩種句型。



- GAN 更新比例難以拿捏，也許 D 過度訓練造成結果收斂到特定句型，如單字回答或者 I don’t know 等。
- Cornell movie dataset 包含了許多困難與非日常的對話，訓練出來的回答都使用許多口語和複雜的句子，反而沒法在簡單的對話中達到好的效果。應該選用不同的 dataset 做不同的實驗。
- 生成時使用 tf.argmax() ，產生的回答幾乎都是”i don’t know ”或是”i don't think so ”，能是因為這些是 Discriminator 最無法分辨的回答，應該在 reward 的地方對這類的回答進行 punishment。

分工

沈恩禾	江承恩	傅鈞笙	吳昭霆
參數調整	training	編寫架構	資料前處理

References

[1] Lantao Yu at el.“SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient”
[2] Jiwei Li at el. “Deep Reinforcement Learning for Dialogue Generation”