

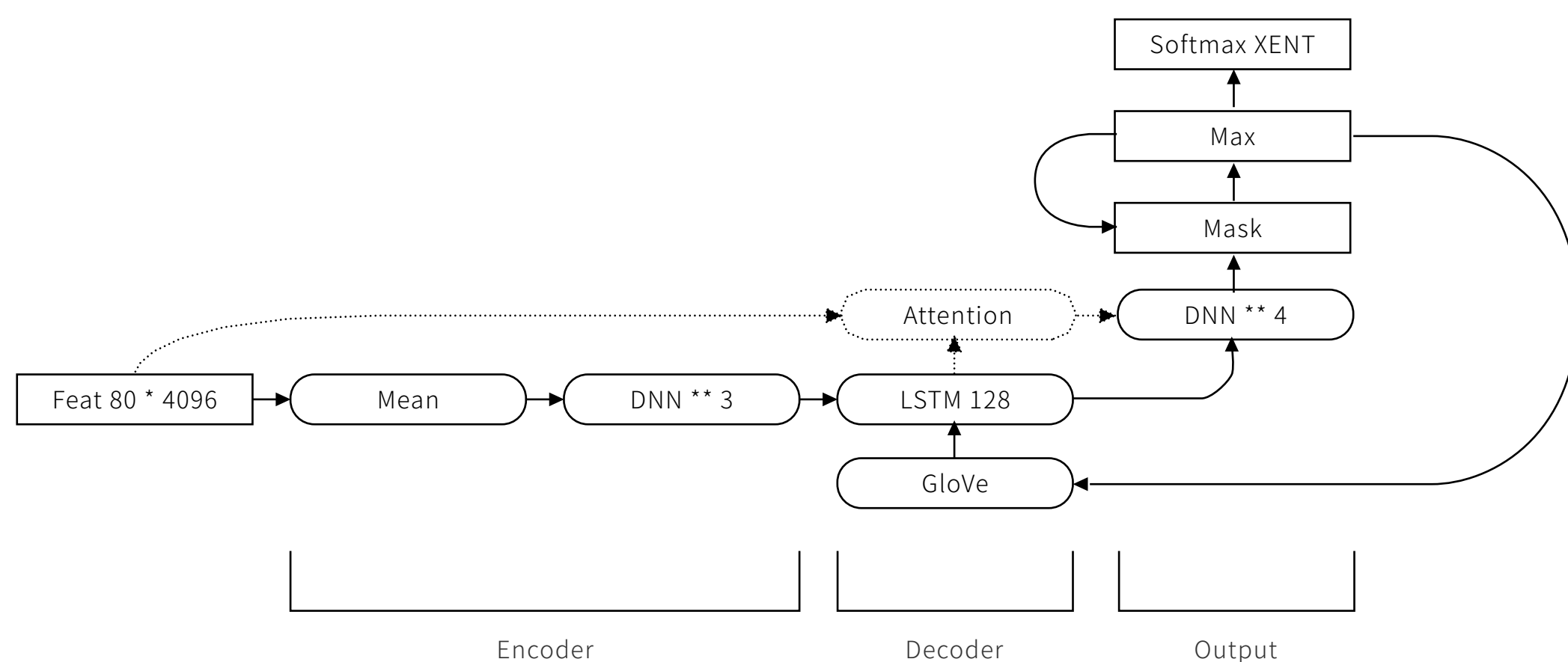
ADL Homework #2 Report

b04902013 鄧逸軒

1. Model Description

Encoder 是把 80 個 frame 的特徵做平均之後, 用 3 層的 DNN 轉成 decoder 的初始狀態. Word embedding 使用 GloVe (840B 300D). Decoder 由一層 LSTM 構成, 輸出對 80 個 frame 做 attention 之後與原始輸出接起來, 用 4 層 DNN 轉換成每個字的機率. 每層都有 dropout 0.2. 使用 RMSProp 做優化.

Network Structure



2. Attention Mechanism

RNN 的輸出會和每個 frame 算 attention, 之後把 feature 做加權平均之後和 RNN 的輸出連接起來.

$$a_{ik} = \text{softmax}(W_{att} \cdot \tanh(W_{enc}u_i + W_{dec}v_k))$$

$$s_k = \sum_{i=0}^{80} a_{ik}u_i$$

$$c_k = [s_k; v_k]$$

Attention 在 RNN 更深的時候會影響效果非常多, 不過在架構較淺的時候沒有什麼效果. 因為最終的模型我是採用 1 層 RNN 作為 decoder, 所以沒有加入 attention.

P.S. 作業要實做的 attention 我放在另外一個資料夾內.

3. How to improve your performance

在 decoder 輸出的時候, 我把前面預測過的字用個 mask 去掉, 也就是說他不會預測出重複的字, 儘管這沒有讓 BLEU 變高, 但是預測出來的句子看起來比較自然. 冠詞剛好因為句首的要是大寫, 所以主詞和受詞都可以有一個冠詞, 而不會被去掉.

3. Experimental Results and Settings

3.1 Schedule Sampling

使用 Schedule sampling 之後 BLEU 會增進非常多, 但是句子變得殘缺不全, 輸出很多都變得像是 ‘A man is a the.’ 只剩下文法.

試著調高使用 true label 的比率之後雖然有改善, 但看起來還是不太自然. 考慮到最後是用 peer review, 我決定不使用 schedule sampling.

3.2 Language Modeling

原本Decoder 輸入的是 embedding, 想說能不能改成輸入一些更高層次的特徵, 所以我訓練了一個 Language Model, 使用 LM 中 RNN 的 output 作為 decoder 的 input. Language Model 的準確率大概可以到 90%

因為這次的作業允許使用外部 Data, 我去找了 LM1B 的資料來做訓練, 不過最後都沒獲得比較好的結果, 所以沒有使用在最終的模型裡面.

3.3 Parameter Tuning

Encoder 使用 RNN 或是取平均效果都差不多, 抱著越簡單的模型越好的想法, 我用平均取代掉 Encoder 的 RNN.

Test Set BLEU Score							
Net	Dec * 1	Dec * 1 Attention	Dec * 2	Dec * 2 Attention	Enc RNN	Dec * 1 LM + Att	Dec * 1 LM1B+Att
BLEU	0.65	0.62	0.50	0.64	0.63	0.64	0.64