摘要

前言（資料來源）

文獻探討

研究方法

結論

參考資料

Appendix

Encoder-Decoder Attention模型

現存的諸多模型皆存在著Encoder-Decoder的架構，例如：Seq2Seq、語音辨識、機器翻譯等領域，都以Encoder-Decoder模型，進行相應的深度學習領域研究[]。Encoder可以視為利用時間數列資料的每一個時間點t-1進行預測，最後輸出一個t的結果，可表示成Encoder = (e0, e1, …, et-1)，而其輸出結果為(ê1, ê2, …, êt)，其中Encoder的輸入稱為Input，通常是欲預測的資料。而Decoder亦是利用時間數列資料的時間點T-1去預測T，可以表示成Decoder = (d0, d1, …, dt-1)，其相對應輸出為(d̂1, d̂2, …, d̂t)，模型上期望訓練完成後的Encoder最終輸出êt可做為Decoder的d0輸入放入Decoder模型當中，以期得出Decoder的輸入在每一個時間點的輸出。Encoder的輸入層經過MFCC轉換過後的一秒英文數字檔案，為99個時間點的39維MFCC特徵，而Decoder的輸入為該音檔的音素（Phoneme）。以每個時間點的輸出作為特徵，透過全連接層（Fully-connected）辨識該時間點是否有音素，以及該音素為何。Attention[]可視作嫁接Encoder與Decoder的橋樑，由於目前的RNN皆存在梯度消失以及梯度爆炸的問題，而Attention為一個十分簡單的全連接層架構，會對每一個時間點的RNN輸出進行特徵提取，並得出該輸出的注意力分配。使用RNN + Attention模型能夠過濾、提取出，對於輸出來說重要且具有時間相關性的特徵。

Encoder

Encoder分為三大部分，全連接層、ResNet層[]，以及LSTM層。全連接層先對每個時點的39維特徵進行線性投射至高維度，突顯該時點的特徵。經由三個ResNet層將鄰近的時間也納入考量，ResNet將模型輸入經由三層Convolution層進行特徵擷取，且將輸入與第三層Convolution的結果相加，再透過ReLU輸出，可以有效避免模型的梯度消失。將第一與第二層的ResNet每三個時間點的特徵進行連接，使時間步長t壓縮成t/3，也就是新的每個時間步，包含3個時間步長的資訊，避免LSTM層在過長的Recurrent之後，出現梯度消失或者梯度爆炸，導致輸出結果收斂至錯誤的輸出。而最後一層的ResNet輸出，會作為Encoder的LSTM層的每一個時間點的輸入(e0, e1, …, et-1)得到(ê1, ê2, …, êt)。

Attention

Attention可以簡單分為兩個全連接層，與一個輸出，稱為Context Vector（Ct）。將Encoder的輸出(ê1, ê2, …, êt-1)稱作query，而êt稱為value。

Decoder

Decoder主要為一層LSTM，其輸入為音素序列。音素序列首先會先通過一個Embedding層，將每個音素投射至指定維度的空間，使得音素不會因為音素的編號數字影響輸出結果，音素序列前後會加入序列開始（Start of sequence）與序列結束（End of sequence）的信號，亦會同時透過Embedding層進行轉換，稱經過Embedding的音素序列為音素向量。接著將音素向量與Decoder的開始信號與Attention模型所輸出的Ct連接起來，作為LSTM層的輸入值，對開始信號之後的每一個時間步進行預測，直到預測的結果出現序列結束信號，又或者達到設定的預測個數上限時，才會停止。最後，將該預測的結果通過全連接層進行分類，進而得到每個時間點所預測的音素。

模型訓練