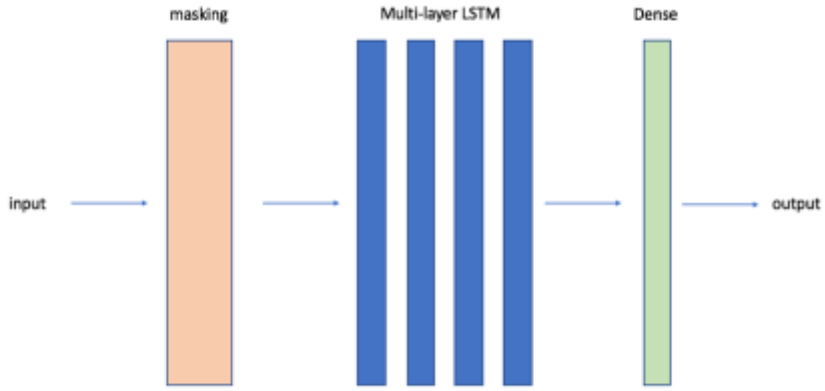
ADL hw1

R06922057 梁智泓

- **Model description**
  - **Input data:**

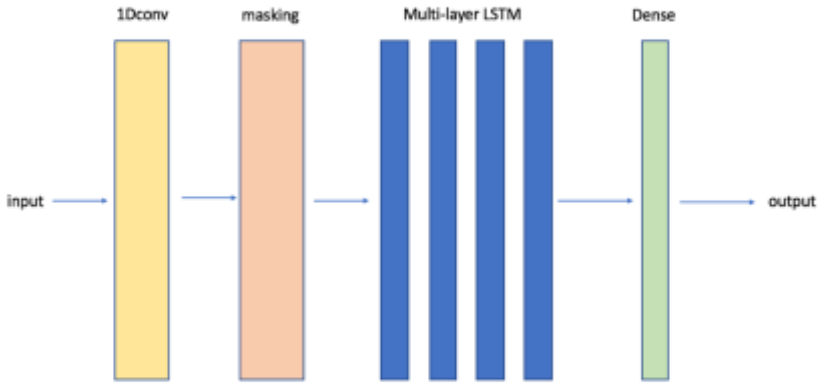
    將全部 data 的 0.1 作為 validation set，0.9 作為 training set
  - **RNN:**

    如下圖所示，我的 RNN model 主要是以 LSTM 為主體，將每句話中每個 frame 作為 input，而為了要使每句話 frame 的數目一樣，我先對 input data 做 padding，將每句話 frame 數目拉到 800，少的部分以 feature 全為 0 的 frame 來補，疊了多層的 LSTM 後，以 48 個 phone 做分類。



  - **RNN + CNN**

    如下圖所示，我的 RNN + CNN model 同樣會先做 padding，做完 padding 之後，將每句話中的 frames 與前後時間的 frame 的做一層的 1dConv，重新得到新的 frame feature 之後，再餵進多層的 LSTM 中，以 48 個 phone 做分類。



- **How to improve performance**

  一開始，我使用了最簡單的 1 層 CNN 加上 1 層 RNN，然而跑出來的結果非常的不好，為了優化 model，我發現了幾個問題，並使用了下列方法解決，使 model 的準確度更高

  1. **Masking**：因為 padding 的關係，導致我增加了太多的無意義的 frame，所以 model 在算 loss 跟 acc 時，會連同這些無意義 frame 一起算進去，導致 model 太快收斂，無法正確的 train 到真正重要的 frame data，為了解決這樣的問題，我在 model 的一開始，做了 masking，讓 model 能在一開始便忽略那些無意義的 frame，從真正重要的 frame data 下手。

  2. **New label**：因為 padding 的關係，會多出很多空的 frame feature，一開始我是將其 label 成 sil，然而這些與原本 data 中的 frame 一點關係都沒有，會影響到真正 sil 的 phone model，所以我決定多出一個 label，也就是第 49 個 null phone，然後將所有補上去的 frame 都使用 null phone，避免影響 model。

  3. **Bi-direction LSTM**：傳統 RNN 中的 LSTM 只會不斷透過時間 t 之前的 frame 來判斷目前 frame 的 label，然而在 phone 的 domain 中，每個音節都會受到下一個要說的音節的影響，進而有類似「變調」的效果，例如中文的『可口』，為了讓 model 學習這樣的 heuristic，我將傳統的 LSTM 換成了 Bi-direction LSTM，使 model 可以同時參考前面與後面的 frame 做 training。

  4. **Multi-layer LSTM**：最一開始，我只使用了一層 output_size=1024 的 LSTM 做 training，雖然出來的結果不算太差，但後來使用了多層 output_size 較少的 LSTM 後，結果明顯的變好，原因是因為較後面的 layer，就能學到更加 high level 的 feature，讓分類能夠更加精準。

  5. **CNN larger kernel**：類似 Bi-direction LSTM，CNN 的作用就是透過 1Dconv，將前後 frame 的 feature 也考慮進去，而為了讓 CNN 能夠考慮更多鄰近的 frame，我將 1Dconv 的 kernel 拉大，讓他可以得到更多鄰近 frame 的資訊。

- **Experiment**

RNN：

	Test score	Valid score	OutSize	epochs
LSTM1	17.5	0.70	1024	20
LSTM2	16.8	0.71	1024	20
BiDir LSTM1	15.0	0.75	1024	20
BiDir LSTM2	15.6	0.76	1024	30
BiDir LSTM3	11.22	0.7813	400	20
BiDir LSTM5	9.89	0.7816	400	15
BiDir LSTM6	9.22	0.803	400	20
BiDir LSTM7	10.23	0.7883	400	15

由上表可以看出，將 LSTM 換成 Bi-direction LSTM 之後，準確度有明顯的上升。除此之外，也可以觀察到，當每層 layer 的 output size 縮小並增加 layer 數，也會使 predict 的結果更加精準，因為他學到了更多 high level 的 feature。然而不斷的增加 layer 也會有問題，當 data 的複雜度沒有那麼高時，layer 越多反而會影響 training 的結果。

CNN：

	test	valid	filter	kernel	LSTM out size	epoch
Cnn1+LSTM1	26.2	0.66	64	3	128	30
Cnn1+LSTM4	17.56	0.701	64	3	256	50
Cnn1+BiDir LSTM1	15.2	0.7521	64	3	1024	20
Cnn1+BiDir LSTM3	Unknown	0.7723	64	5	400	20
Cnn1+BiDir LSTM5	Unknown	0.7533	64	5	200	25
Cnn1+BiDir LSTM6	10.045	0.7799	64	5	400	15
Cnn1+BiDir LSTM6	Unknown	0.7819	128	5	400	20

因為上傳次數有所限制，所以無法知道部分 test 的結果，但可以直接從 validation 的結果觀察。以整體來說，加入 CNN 後的 net 會比單純的 Bi-Direction LSTM 來的差，可能是因為我在設計 net 時，會先將 data 過 1Dconv 之後才做 masking，可能會因此少掉部分的 feature。而單純比較 CNN+RNN 的 net 的話，可以觀察到層數越少，預測的結果越不好。除此之外，每層 LSTM 的 output size 不能太小，否則會學不到精細的 feature。而 CNN filter 的數量與 kernel 的 size 越大，準確度也會越高。