

Wireless Multimedia Networks Lab1

Throughput Prediction written by 313553024 蘇柏叡

1. 本實驗訓練過程與前置作業介紹

由於助教學長已經將程式碼、資料目標欄位、損失函數等皆設計完畢，因此我們僅需要根據自身環境執行程式碼。在執行的過程中，我們首先需要定義我們的資料集，由於5個scenario的資料筆數均相同，且助教已經把多個scenario資料集串接好，因此我們僅需要把想放入的資料填入即可，故針對訓練、測試的比例，我並未做任何的更動(為90%:10%)。此外，由於本人使用的硬體設施並不支援多線程的num_workers，因此若一樣配備無法支援多線程的num_workers者，可以直接將其值設置為0，便可正常運作。此外，由於範例訓練程式碼僅直接使用單一學習率在10個Epoch進行訓練，為追求更精準，我選擇調整Epoch數量，並且加入Early Stopping、LR scheduler技術除了避免使用單一學習率容易造成的過擬合現象，並且還能根據驗證集的性能更彈性地進行訓練。

2. Baseline Model:

- 模型架構介紹

本次實驗使用的Baseline Model係助教提供之模型，其架構為Encoder-Decoder架構。首先在Encoder端，會由輸入層將輸入的維度轉成128通道，並且再透過線性層將128通道降維至32通道，並且準備進入Decoder端。至於在Decoder端，則是會將32通道還原成128通道，並且再接一個線性層把輸出變成一個通道。此外在過程中也會設計Dropout避免模型過度擬合。此外，在每個Layer之間，均會加入Activation(Base Model採用的是ReLU)以解決線性不可分問題並且防止梯度消失之問題。

- 實驗結果圖

本研究先使用Baseline Model並且針對全部資料進行訓練與測試，所得之結果圖如圖1(為使用全資料集，並且繪製測試集前200筆data)，此外和原始檔案較為不同之處在於我們引入了Early Stopping、LR scheduler做為調整，以避免模型因為訓練Epoch過多或過少導致無法有效擬合的問題。因此，我們將依據驗證集的損失值去比對，若是Val_loss連續超過5個Epoch未改善，則會將原先學習率乘上factor(我們設0.5)，至於Early Stopping部分則是連續10個Epoch未改善才會觸發該條件，至於最大Epoch我們設為50。另外，在繪製損失函數的圖2中我們可以發現，雖然原先設定Epoch數量為50，然而隨著Training Loss不斷地下降，但Validation Loss在第5個Epoch後並沒有更

低的損失值出現，經過調降學習率後Validation Loss依然未改善，因此從而觸發Early Stopping的條件。至於結果部分，雖然可以看出大部分的預測與真實值的趨勢有符合，但是在許多筆資料中存在極為嚴重的誤差，因此我們在下一章節將進一步改善此 Baseline Model，試圖提升模型性能。

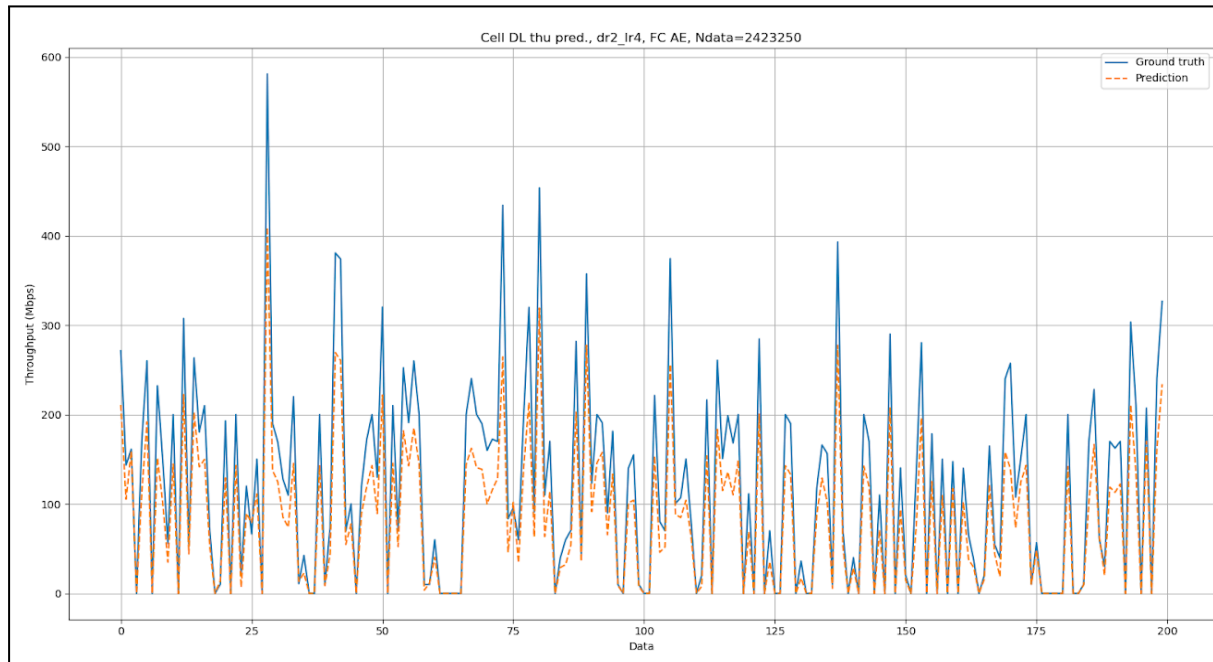


圖1 為測試集前200筆的Throughput Prediction

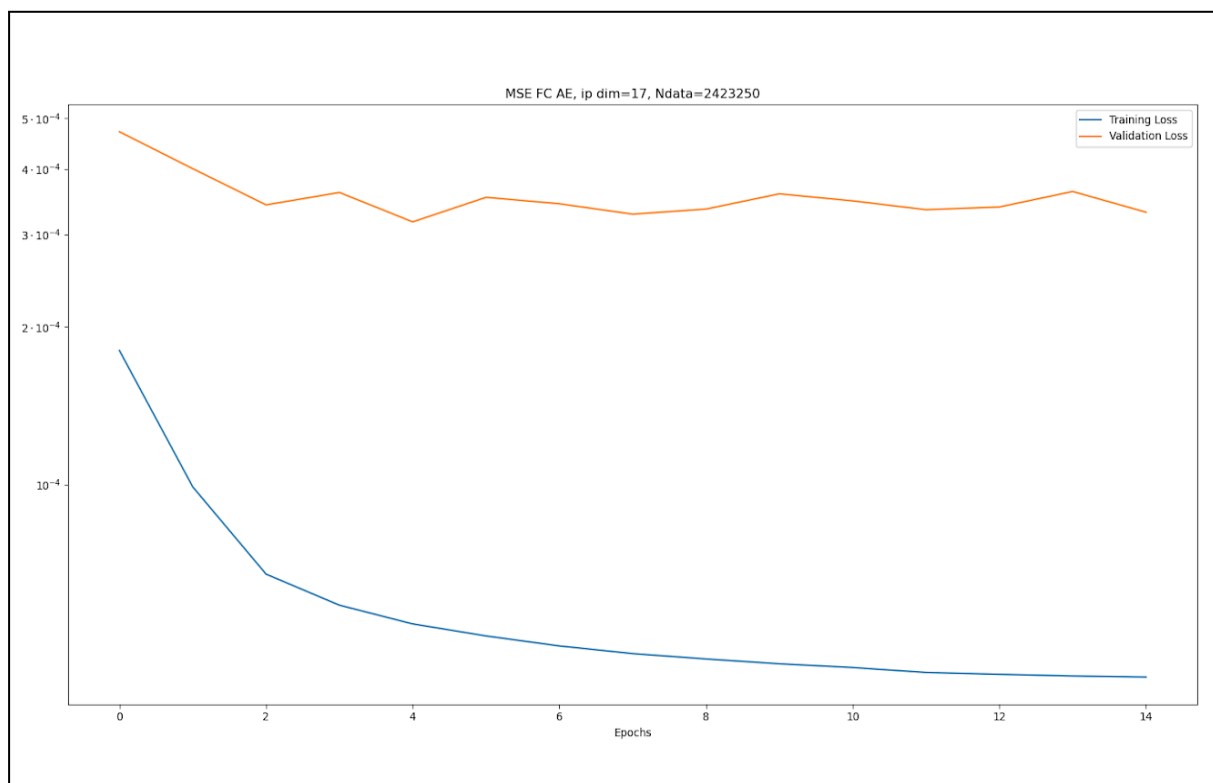


圖2 為Throughput Prediction訓練過程中的訓練、驗證損失函數圖

3. Our proposed Model:

– 模型架構介紹

本研究使用自行建立的Model一樣是採用Encoder-Decoder結構，不過除了在輸入層改成將原始輸入改成256通道之外，在結構中亦加入了各兩個殘差模塊。殘差模塊的功能是為了避免在倒傳遞時因為梯度消失導致訓練不佳，以及在網路過深的情況下可能會造成退化的問題，因此使用殘差模塊可以解決以上兩大問題，進而提升模型準確性。圖四為本研究自行建立之模型架構圖。

首先在編碼器部分，其流程為將輸入維度為17的資料先經過一層線性層將其映射至256通道，接著經過兩個殘差模塊並且均保持256通道，最終進入一個線性層把256通道壓縮至128通道。接著，解碼器部分會將壓縮好的128通道送入兩層殘差模塊，並且保持128通道，最終在送入兩個線性層分別將128通道映射回256通道再壓縮回1維，並且通過ReLU將負值過濾為0後再進行輸出。至於殘差模塊的內部結構，我們會再於圖5、6進行更詳細的呈現與說明。

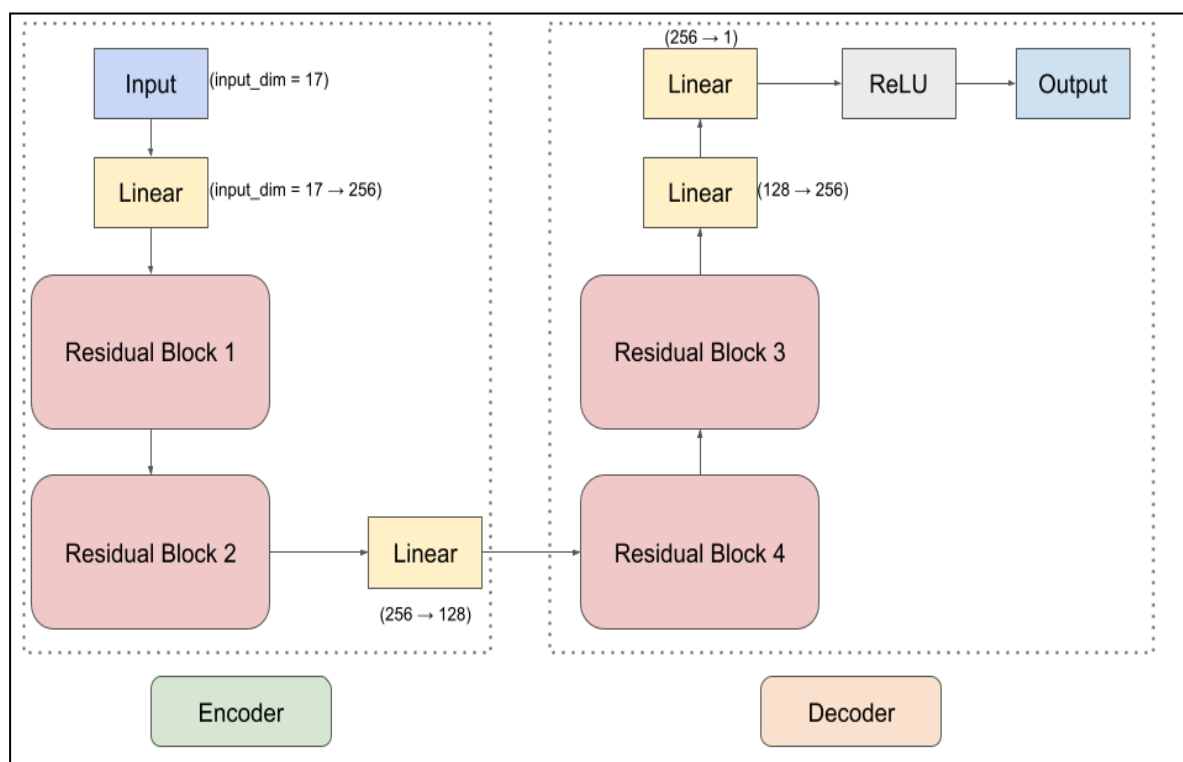


圖4 為自行建立之模型架構圖

在殘差模塊內部結構的部分，我們發現由於編碼器中Residual Block 1和Residual Block 2結構相同，因此我們只使用圖5進行表示。在此結構中分為兩個分支，在input x部分會是由先前線性層映射出的256通道，並且在殘差模塊內部會先經過一個線性層將通道壓縮成原先的一半，並且通過ReLU、Dropout(和原設定一致為0.2)後，再經過一層線性層，把通道映射回256通道，並且和原先Input x進行殘差連接並且輸出。在解碼器中使用的殘差模塊內部結構與先前Residual Block 1和Residual Block 2

結構相同，只是Residual Block 3和Residual Block 4的input x為128通道，並且第一次通過線性層時是將通道砍半為64，再經過ReLU、Dropout後，再經過一次線性層映射回128通道再和input x做殘差連接並輸出。

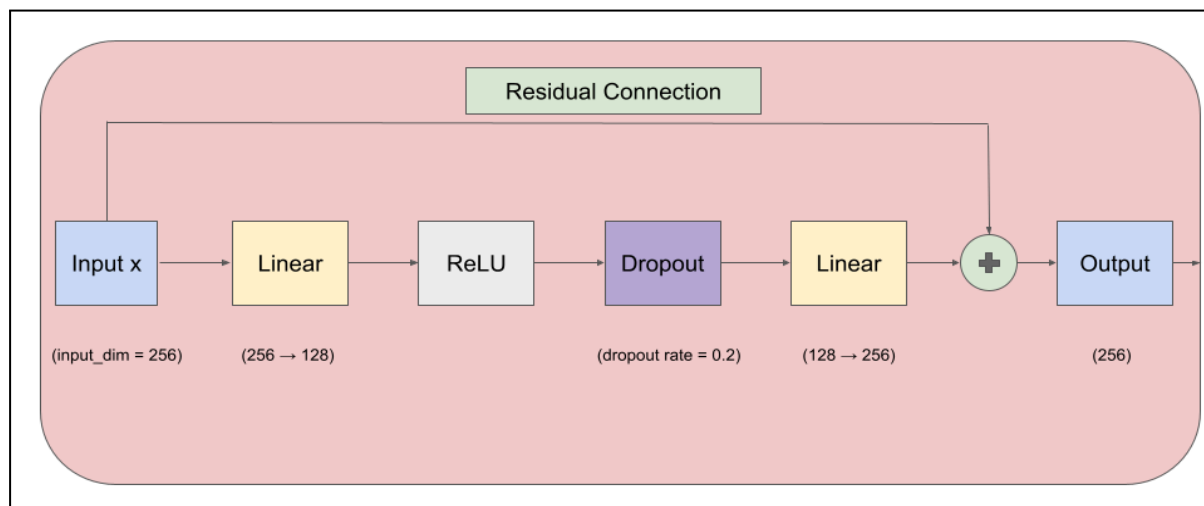


圖5 編碼器中殘差模塊的內部架構圖

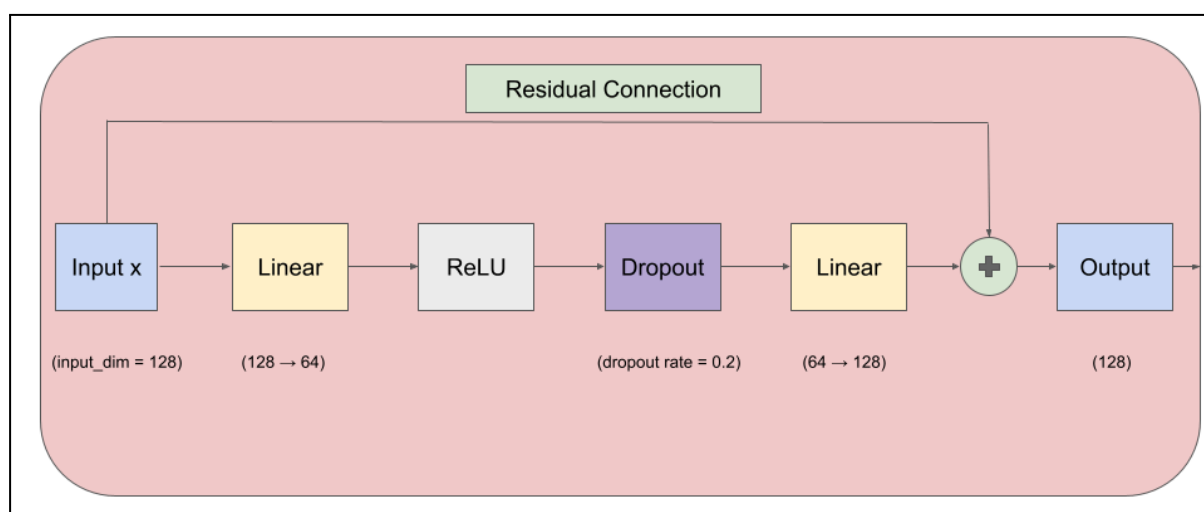


圖6 解碼器中殘差模塊的內部架構圖

－ 實驗結果圖

本研究先使用自行建立的Model並且針對全部資料進行訓練與測試，並且我們仍然會依據驗證集的損失值去比對，若是連續Val_loss超過5個Epoch未改善，則會將原先學習率乘上factor(我們設0.5)，至於Early Stopping部分則是連續10個Epoch未改善才會觸發該條件。訓練後之結果圖如圖7，經與圖1相比對可清楚發現，在我們提出的模型中，預測出的Throughput與Ground truth是更為貼合的。另外，在繪製損失函數的圖8中我們可以發現，不論是訓練損失亦或是驗證集的損失，其Loss Curve均是收斂且並未發生過擬合之問題，縱使驗證損失在過程中存在些許的震盪，但仍然持續降低且在經過50個Epoch訓練後，訓練、驗證損失均為約略低於 2×10^{-5} 是相較於原先Baseline Model於驗證集表現約 3.3×10^{-5} 是提升許多的，因此不論是從Throughput

Prediction的折線圖或者是損失函數的損失曲線圖，均能證明我們提出之方法是較優於Baseline Model的。

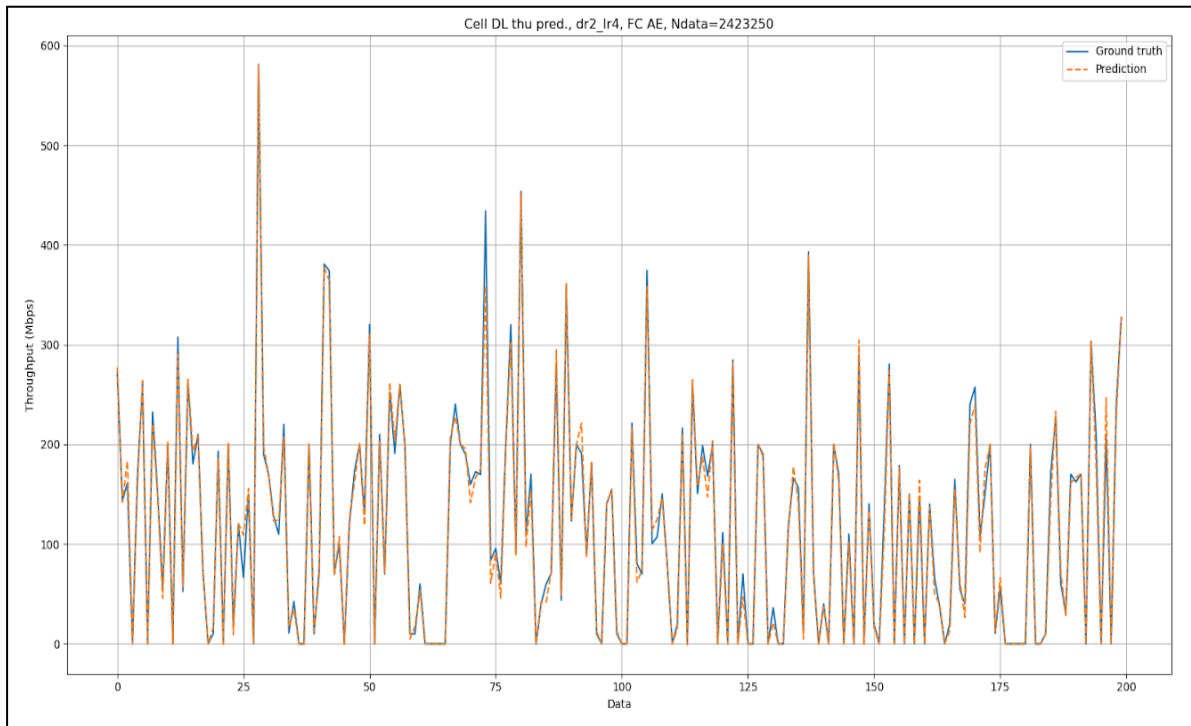


圖7 為測試集前200筆的Throughput Prediction

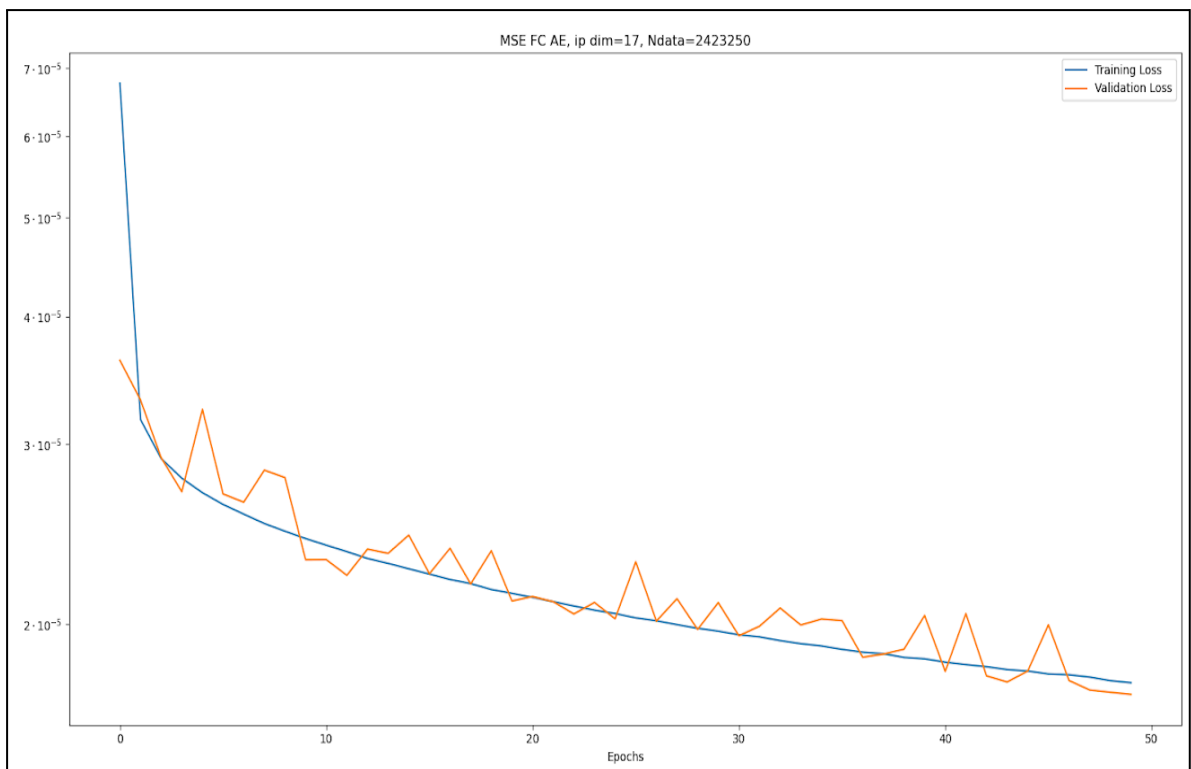


圖8 為使用Our Model在Throughput Prediction訓練過程中的訓練、驗證損失函數圖