

## Final Project

組員：611415102 林彥佑 407420041 劉季恆、408410002 林靖軒、  
408410098 蔡嘉祥、408410102 楊力行

方法：

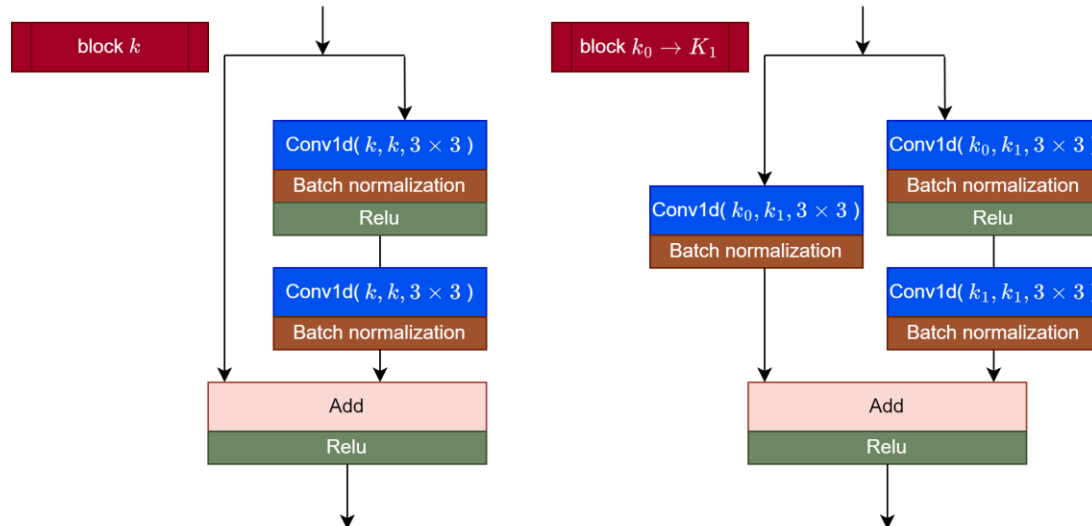
使用 ResNet 架構來做為模型的訓練

### Feature Extraction : 1D Convolution

之所以使用 convolution 的原因是因為資料為一個 1 dimension 3 channel 的時序訊號 (window size 沿用原始設置 32)，而 1d convolution 可以拿來做訊號的處理，所以我們想用 1D convolution layers 來取代 Linear layers 當作我們的 Feature Extraction operator。

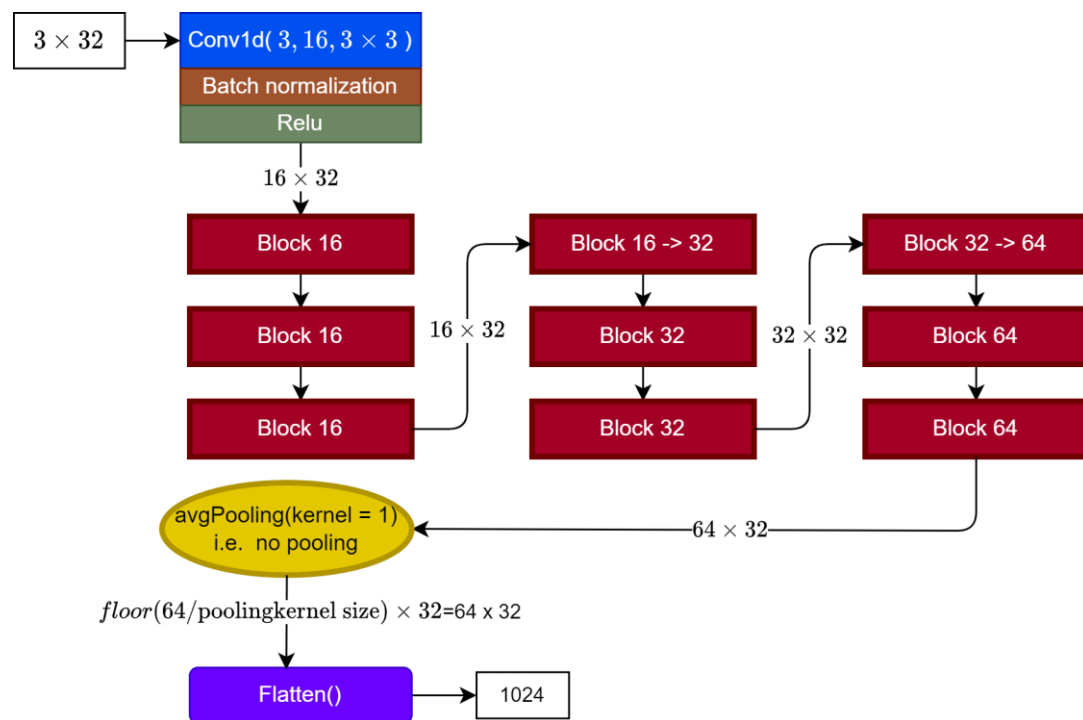
\*\*由於 convolution layers 會使參數增多 (每個 kernel 都要學)，所以我們打算使用 Residual Network 的架構。

最終，我們參考 ResNet20 架構，把其中 2D convolution 改成 1D convolution



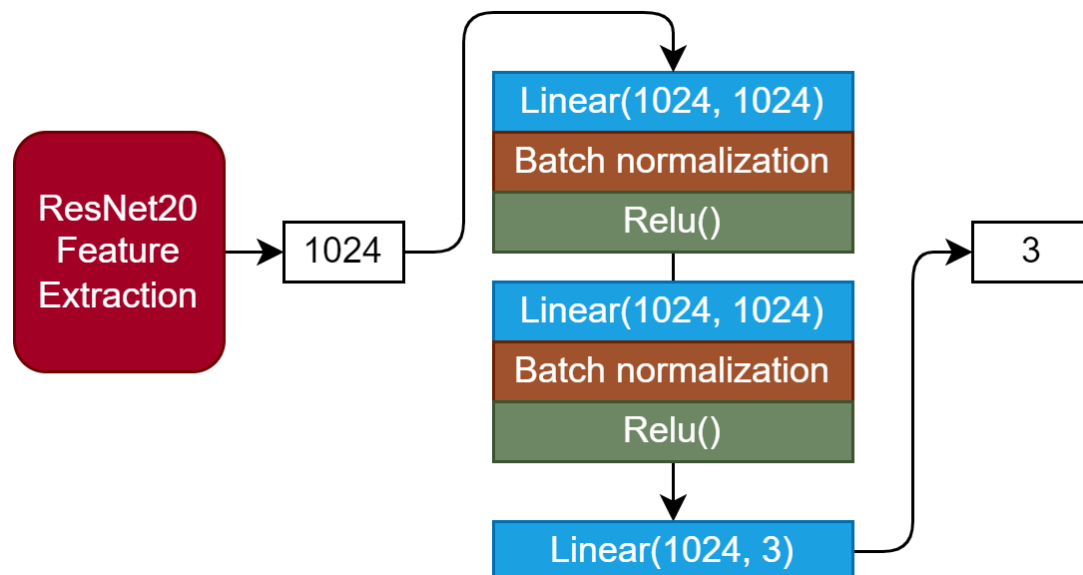
\*\* 原本的架構當 out channel 數量提升之時，旁邊的 shortcut kernel size 是 1x1；我們這邊不小心誤用成 3x3 了

## Feature Extraction model

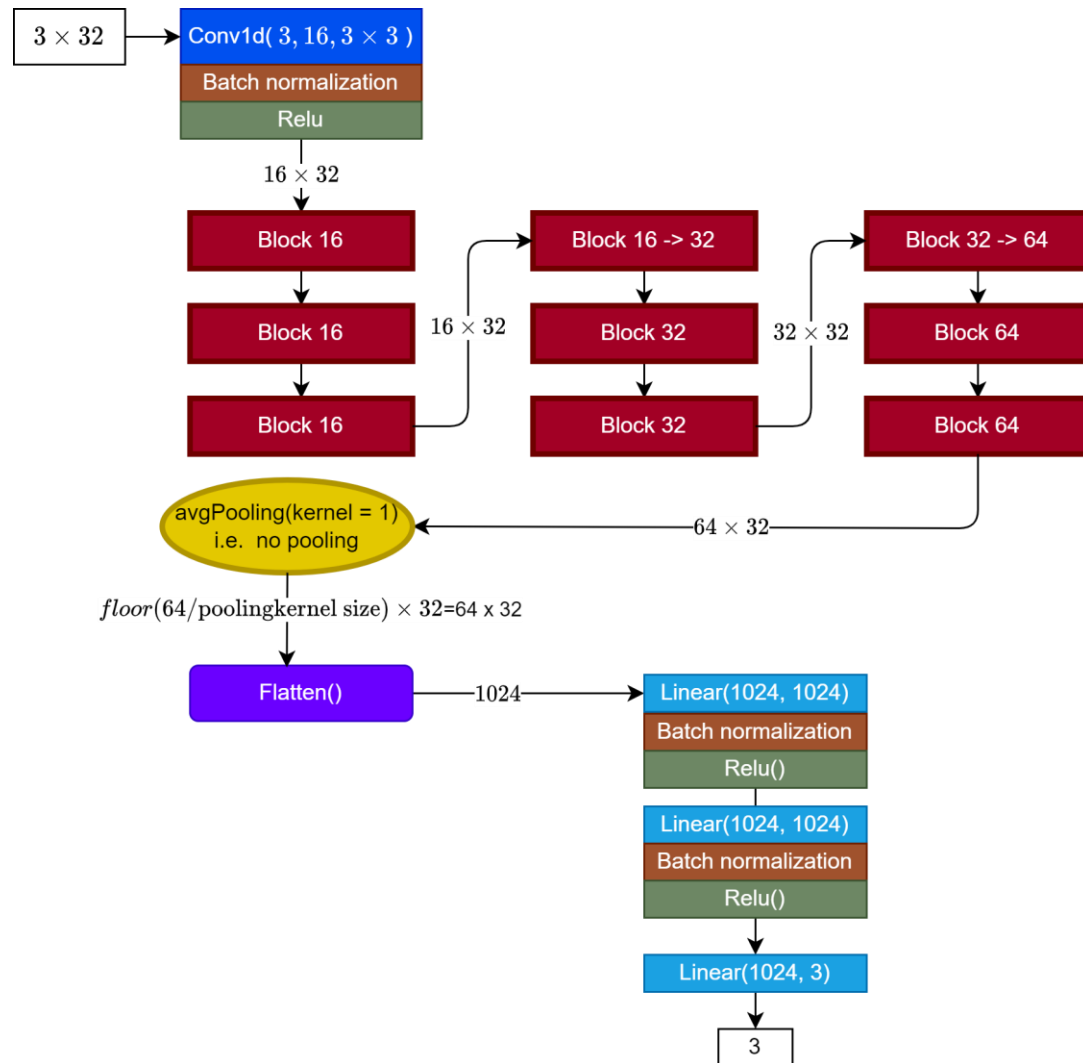


## Classifier : Vanilla Linear Layers with activation function

把使用 kernel 抓取特徵完的 features 送進k層的 Linear Layers 來作為 classifier。這邊我們選用了2層 Linear layers with activation function 在加最後 output layer 當作我們的分類器。



## Entire Model:



## 超參數設定:

batch\_size = 1024

window\_size = 32(每次 Input 的總時序資料長度)

window\_future = 8(取目標時間點後多少為 Input)

window\_past = window\_size - window\_future(取目標時間點前多少為 Input)

wx = 8(資料 padding 長度的參數)

num\_epoch=50

classifier\_layers=[1024, 1024]

分數：

	public	private
我們這組 score	0.300666	0.284011
Sample code score	0.273919	0.263645
改良率	11.95%	7.724%

\*(public 跟 private 的排名是不同 code，private 那次的 code 在 public 的 score 為 0.280748)

Public 排名(317)：

317	ML_34		0.300666	20	6d
-----	-------	--	----------	----	----

Private 排名(132)：

132	185 ML_34			0.284011	20	6d
-----	-----------	--	--	----------	----	----

Sample code score：



MAYUKH BHATTACHARYYA · COPIED FROM PRIVATE NOTEBOOK +334,-269 · 3MO AGO · 9,865 VIEWS

## PyTorch FOG End-to-End Baseline [LB 0.254]

Python · [Parkinson's Freezing of Gait Prediction](#), [Copy Train Metadata](#)

Notebook Input Output Logs Comments (24)



Competition Notebook

[Parkinson's Freezing of Gait Prediction](#)

Run

2377.0s - GPU P100

Private Score

0.273919

Public Score

0.263645



交叉驗證：

Epoch: 47

100%|██████████| 4883/4883 [06:28<00:00, 12.56it/s]

Train Loss: 0.0079

100%|██████████| 4868/4868 [02:43<00:00, 29.80it/s]

Validation Loss: 0.4688, Validation Score: 0.246, ClassWise: 0.027,0.651,0.059

=====

Epoch: 48

100%|██████████| 4883/4883 [06:28<00:00, 12.57it/s]

Train Loss: 0.0077

100%|██████████| 4868/4868 [02:39<00:00, 30.48it/s]

Validation Loss: 0.4707, Validation Score: 0.243, ClassWise: 0.034,0.648,0.049

=====

Epoch: 49

100%|██████████| 4883/4883 [06:28<00:00, 12.57it/s]

Train Loss: 0.0076

100%|██████████| 4868/4868 [02:39<00:00, 30.61it/s]

Validation Loss: 0.4810, Validation Score: 0.242, ClassWise: 0.032,0.643,0.052

=====

### 結果討論:

一開始試著新增層數來增加模型的深度，但效果有它的上限，後來經過觀察後發現資料屬於時序性資料，所以將架構由 DNN 改為 LSTM，但進步的幅度仍然有限，之後又想到因為 1D convolution 可以用來做訊號處理，所以將 1D convolution layers 來取代 Linear layers 當作 Feature Extraction operator，最後參考 ResNet20 的架構把 2D convolution 修改成 1D convolution，抓取完特徵後，再用 2 層 Linear layers with activation function 加上 output layer 當作我們的分類器。