# Deep Learning Lab3 Report

110511118 陳孟頡

**Task 1.**

## I. Overview

### 1. Parameter size

總共使用參數量為 **22.127 M**

```
config = {
    # model architecture
    "embedded_dim": 256,
    "head_num": 1,
    "ff_hidden_dim": 512,
    "encoder_num": 2,
    "decoder_num": 1,

    # training configuration
    "device": torch.device("cuda" if torch.cuda.is_available() else "cpu"),
    "batch_size": 32,
    "learning_rate": 3e-4,
    "eta_min": 5e-5,
    "epoch_num": 200,
    "regularization": 5e-5,
    "dropout_rate": 0.3,
    "smoothing_rate": 0.1,
    "distillation_rate":0.0,
    "augmentation": True,
}
```

```
EMB_SIZE            = config["embedded_dim"]
NHEAD               = config["head_num"]
FFN_HID_DIM         = config["ff_hidden_dim"]
NUM_ENCODER_LAYERS  = config["encoder_num"]
NUM_DECODER_LAYERS  = config["decoder_num"]
DROPOUT_RATE        = config["dropout_rate"]
SRC_VOCAB_SIZE      = tokenizer_cn.vocab_size
TGT_VOCAB_SIZE      = tokenizer_en.vocab_size
DEVICE              = config["device"]

transformer = Seq2SeqNetwork(NUM_ENCODER_LAYERS, NUM_DECODER_LAYERS, EMB_SIZE,
                             NHEAD, SRC_VOCAB_SIZE, TGT_VOCAB_SIZE, FFN_HID_DIM, DROPOUT_RATE)

for p in transformer.parameters():
    if p.dim() > 1:
        nn.init.xavier_uniform_(p)

transformer = transformer.to(DEVICE)
param_transformer = sum(p.numel() for p in transformer.parameters())
print (f"The parameter size of transformer is {param_transformer/1000000} M")
#   The parameter size of model should be less than 100M (100,000k) !!!
#   The parameter size of model should be less than 100M (100,000k) !!!
#   The parameter size of model should be less than 100M (100,000k) !!!
```

```
The parameter size of transformer is 22.126916 M
```

### 2. Accuracy (BLEU score)

最高 validation accuracy(BLEU score)為 **0.2651**

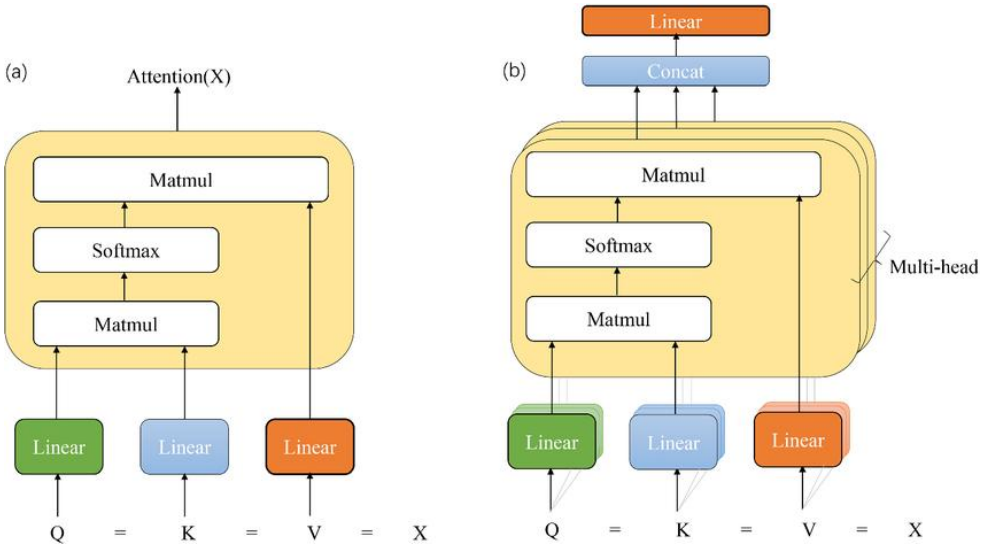
```
Epoch: 185, Train loss: 4.8026, Val loss: 5.5191, Val Acc: 0.2641, LR = 0.000080, Epoch time = 21.87s
(model saved)

Epoch: 186, Train loss: 4.8017, Val loss: 5.5114, Val Acc: 0.2651, LR = 0.000076, Epoch time = 21.86s
(model saved)
```

## II. Transformer Structure

### 1. Multi-head attention

Multi-head attention 在 transformer 有著萃取不同知識的重要角色，例如對於整個句子、單一文字、甚至是文法的不同知識理解。然而，此次作業中的 training data 較少，較難發揮 multi-head 的優點，反而可能使模型容易產生 overfitting 現象，因此此次作業中採用 head number = 1，也就是 single head 來實現。




### 2. Encoder

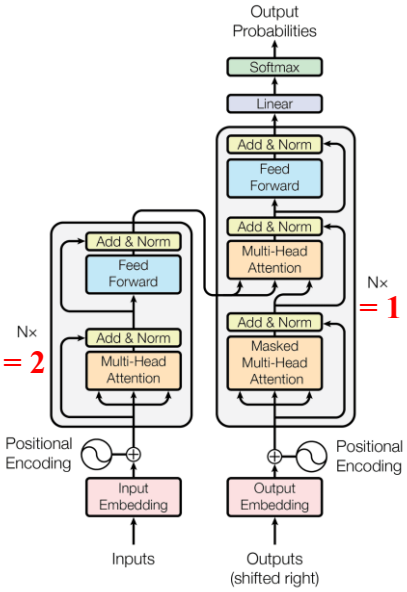
Encoder 在 transformer 中，負責將輸入 encode 至高維度中，使 transformer 能更好提取資料的相關性，達到更高的準確度。

為避免模型過於複雜產生 overfitting 現象，在加上多層 dropout 後，採用 2 層 encoder layer 能達到最佳準確度。

### 3. Decoder

在 encoder 將輸入 map 到高維後，decoder 會依照所學到的相關性將資料 decode 出來，形成輸出翻譯句子。在 decode 時，除了 encoder 提供的輸入資料外，也會將前一次的輸出進行 cross attention，來取得 sequence 相關性。

為避免發生 overfitting，使用 1 層 decoder layer。

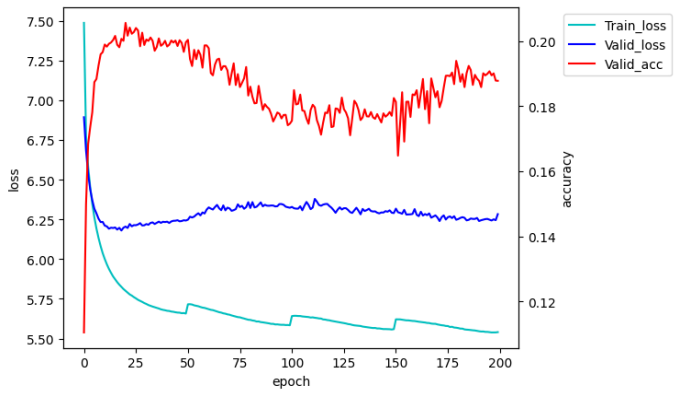


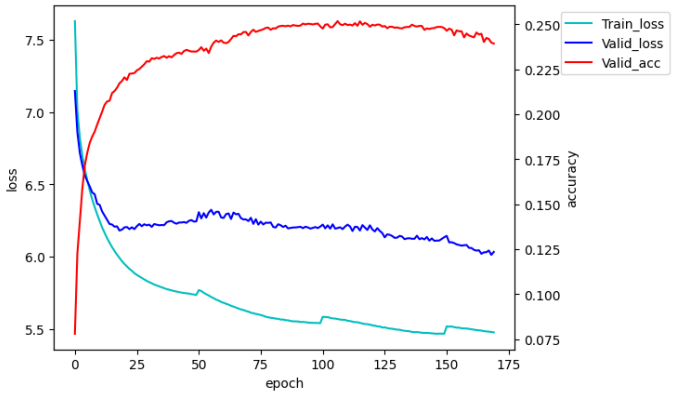
4. **Positional encoding**

由於 transformer 輸入本身並不具有前後順序資訊(self-attention 特性)，但在翻譯任務中，字詞先後會造成文意不同，因此需在輸入資料加上位置資訊，是為 positional encoding。

本次作業中嘗試兩種不同 encoding 方式，分別為 sinusoidal positional encoding 以及 learnable positional encoding。前者有著較快的 inference 速度，以及不易產生 overfitting 的優點，而後者則是有能達到更高 accuracy 的能力。

經過多次測試後，最後採用 sinusoidal positional encodings，原因為此次訓練資料較少，若使用過多參數，容易產生 overfitting，sinusoidal positional encoding 相較 learnable positional encoding 較不易發生此現象，可從兩者的 loss 曲線看出。




左圖: learnable positional encoding
右圖: sinusoidal positional encoding

5. **Token embedding**

訓練 transformer 模型時，所使用的資料為數值數據，無法處理一般文字，token embedding 即作為將離散符號（例如單字或字元）轉換為 continuous vector 的方法。所 embed 出的 vector，將語義訊息包含在其中，讓 transformer attention 機制能萃取出其相關性。

6. **Create mask**

用來產生輸入 sequence 以及 target sequence mask 的函式。Mask 能使 model 專注於提取有效 tokens 間的資訊，而不會受 future token 或其他無效 token 影響。

## III. Training Strategy

### 1. Overview

| | |
|---|---|
| Optimizer | Lookahead with Adam |
| Learning rate scheduler | CosineAnnealingWarmRestarts |
| Loss function | CrossEntropyLoss with label smoothing |
| Data augmentation | MarianMT soft label |
| Others | Knowledge distillation |
| | Sharpness-aware minimization |
| | RMS normalization |
| | GeLU activation function |

### 2. Optimizer

經過測試，使用 Lookahead optimizer 進行訓練，其 base optimizer 為 Adam，加上 weight decay。以下分別介紹 Adam 以及 Lookahead:

```
base_optimizer = torch.optim.Adam(transformer.parameters(),
                                  lr=config["learning_rate"],
                                  betas=(0.9, 0.98), eps=1e-9,
                                  weight_decay=config["regularization"])

optimizer = Lookahead(base_optimizer, k=5, alpha=0.5)
```

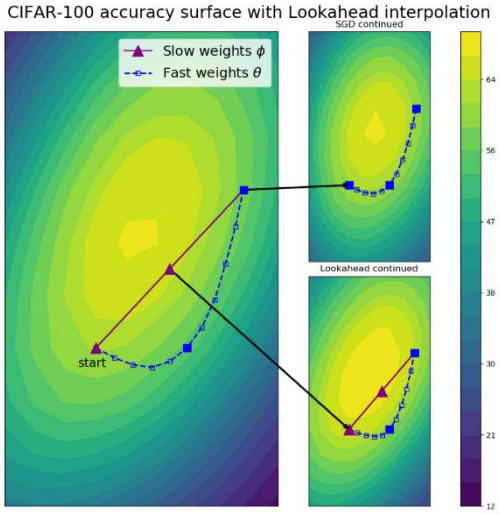
(1) Adam:

結合了 momentum、RMSprop 以及 adaptive learning rate，使超參數的設定變得相對容易，也能達到一定的 performance。因為是用小 dataset，因此加上 regularization，來避免模型過快發生 overfitting，導致 validation accuracy 無法進一步提高。

(2) Lookahead:

Lookahead optimizer [1]做為讓訓練更加穩定，且對於超參數較不敏感的 optimizer，在此次 translation 任務中達到不錯的訓練效果。

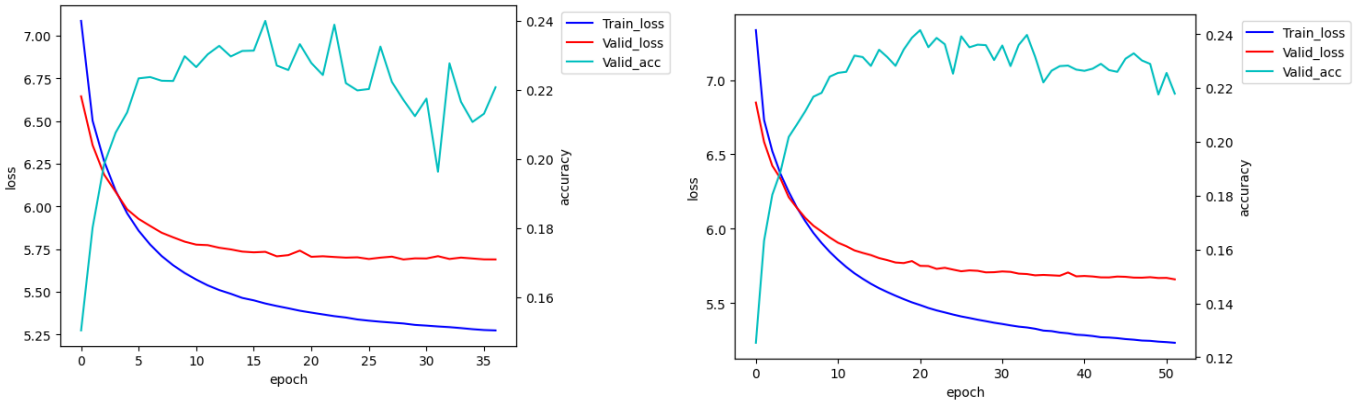
其運作原理如右圖所示，首先經由 base optimizer 先更新 5 次"fast weight"，再讓 Lookahead 取平均，並更新"slow weight"，達到平穩更新的效果。



**Algorithm 1** Lookahead Optimizer:

**Require:** Initial parameters $\phi_0$, objective function $L$
**Require:** Synchronization period $k$, slow weights step size $\alpha$, optimizer $A$
**for** $t = 1, 2, \ldots$ **do**
  Synchronize parameters $\theta_{t,0} \leftarrow \phi_{t-1}$
  **for** $i = 1, 2, \ldots, k$ **do**
    sample minibatch of data $d \sim \mathcal{D}$
    $\theta_{t,i} \leftarrow \theta_{t,i-1} + A(L, \theta_{t,i-1}, d)$
  **end for**
  Perform outer update $\phi_t \leftarrow \phi_{t-1} + \alpha(\theta_{t,k} - \phi_{t-1})$
**end for**
**return** parameters $\phi$

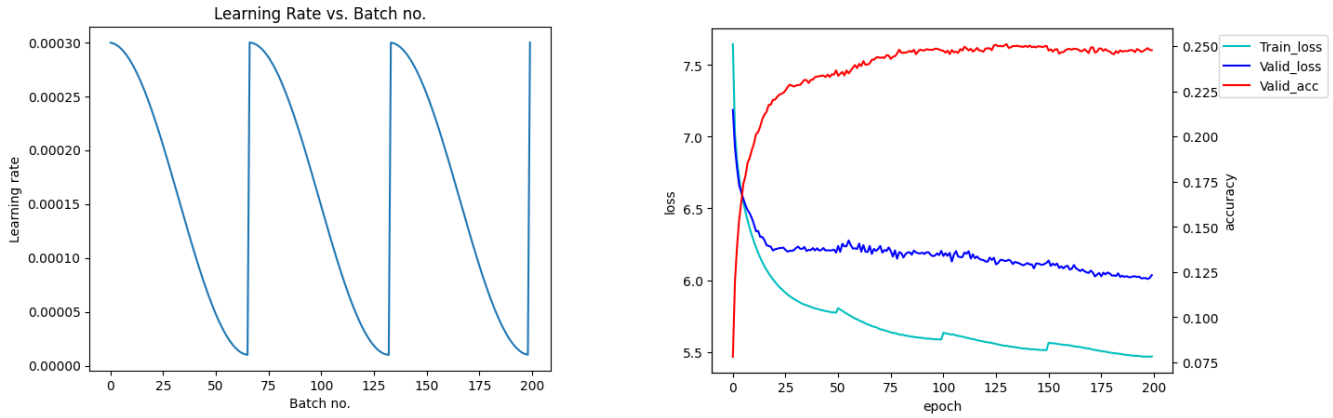
從 loss 以及 accuracy 變化圖可看出，使用 Lookahead optimize 能使更新更為穩定(accuracy 變化較為明顯)。




左圖: 使用 Adam optimizer
右圖: 使用 Lookahead optimizer (Adam 作為 base optimizer)

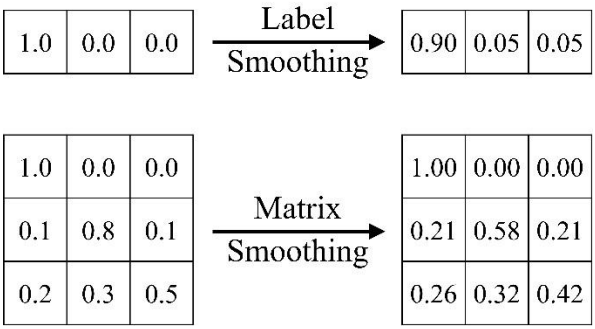
### 3. Learning rate scheduler

觀察訓練時 loss 下降情形，推測當 learning rate 下降到一定程度時，會進入 error surface 的 local minima，而為了跳出 local minima，使用 CosineAnnealingWarmRestarts，每過一定數量 epoch 便進行 learning rate 重置，達到更高的準確度。



## 4. Loss function (Label smoothing)

Label smoothing 為 regularization 的一種，透過轉換 one-hot target label，讓其他 class 也有為小機率，來避免模型對訓練資料 overconfidence。此方法對小的 dataset，以及非一一對應的翻譯任務有為小的準確準確度提升。




## 5. Data augmentation

由於此次翻譯任務僅有少量的資料，因此模型很容易發生 overfitting 現象，如何減少模型發生 overfitting 成爲增加準確度關鍵。使用 data augmentation 增加資料變化性即是方法之一。

使用已訓練好之模型 MarianMT，將英文翻譯成簡體中文，並在訓練模型時以 30% 之機率使用 MarianMT 所翻譯之句子來計算 loss，便能減少模型對原先資料過度 overconfidence，使準確度進一步升高。

Original Data

| | english | chinese |
|---|---|---|
| 0 | Slowly and not without struggle, America began... | 美国缓慢地开始倾听，但并非没有艰难曲折。 |
| 1 | Dithering is a technique that blends your colo... | 抖动是关于颜色混合的技术，使你的作品看起来更圆滑，或者只是创作有趣的材质。 |
| 2 | This paper discusses the petrologic characteri... | 本文以珲春早第三纪含煤盆地的地质构造背景为依据，分析了煤系地层的岩石学特征。 |
| 3 | The second encounter relates to my grandfather... | 第二次事件跟我爷爷的宝贝匣子有关。 |
| 4 | One way to address these challenges would be t... | 解决这些挑战的途径包括依照麻瓜在南非的经验设立真相与和解委员会。 |
| ... | ... | ... |
| 49995 | You were too obtuse to take the hint. | 你太迟钝了，没有理解这种暗示。 |
| 49996 | Therefore, in the event the mortgagee of ship ... | 因此，在这种情况下船舶抵押权人放弃了债务人提供的担保就会影响其他担保人的利益，导致抵押权人的... |
| 49997 | Fourth, puncture administrative bloat. | 第四，削弱行政膨胀。 |
| 49998 | Massimo Oddo says he won't be thinking about h... | 马西莫·奥多声明他不会在世界杯决赛圈比赛结束之前考虑未来的俱乐部。 |
| 49999 | The observing mortals' statue The uncompleted ... | 《冷眼观俗尘》 尚未完工的佛祖雕像，超凡脱俗的神情，似乎在禅悟街巷里匆忙行人的百态人生。 |

50000 rows × 2 columns

Augmented Data

| | english | chinese |
|---|---|---|
| 0 | Slowly and not without struggle, America began... | 美国开始倾听。 |
| 1 | Dithering is a technique that blends your colo... | 抖动是一种把颜色混为一谈的技巧 使颜色看起来更平滑 或只是创造有趣的纹理 |
| 2 | This paper discusses the petrologic characteri... | 本文件讨论湖川第三级煤炭盆地地质结构背景下含煤层的含煤层的油气特征。 |
| 3 | The second encounter relates to my grandfather... | 第二次碰面涉及我祖父的宝箱。 |
| 4 | One way to address these challenges would be t... | 应对这些挑战的一个办法是建立一个以南非麻瓜经验为模式的真相与和解委员会。 |
| ... | ... | ... |
| 49995 | You were too obtuse to take the hint. | 你太迟钝了 不敢接受暗示 |
| 49996 | Therefore, in the event the mortgagee of ship ... | 因此,如果船舶抵押权人放弃债务人提供的保修权,影响到其他保修人的利益,则放弃应无效或部分无效。 |
| 49997 | Fourth, puncture administrative bloat. | 第四,穿刺行政浮肿。 |
| 49998 | Massimo Oddo says he won't be thinking about h... | 马西莫·奥多说他不会考虑他的俱乐部未来 直到世界杯决赛结束 |
| 49999 | The observing mortals' statue The uncompleted ... | 旁观者雕像 未完成的佛像 超越了表达的一步 站在街上 似乎在观察人 |

50000 rows × 2 columns

```python
class TextTranslationDataset(Dataset):
    def __init__(self, src, dst, aug_src=None, aug_dst=None, dis_dst=None, augmentation=False):
        self.src_list = src
        self.dst_list = dst
        self.aug_src_list = aug_src
        self.aug_dst_list = aug_dst
        self.dis_dst_list = dis_dst
        self.augmentation = augmentation

    def __len__(self):
        return len(self.src_list)

    def __getitem__(self, idx):
        if (self.augmentation == True):
            random_integer = random.randint(1, 10)

            if(random_integer < 4):
                return self.aug_src_list[idx], self.aug_dst_list[idx], self.dis_dst_list[idx]

            else:
                return self.src_list[idx], self.dst_list[idx], self.dis_dst_list[idx]

        else:
            return self.src_list[idx], self.dst_list[idx], self.dis_dst_list[idx]
```

## 6. Knowledge distillation

另一個與 data augmentation 相像，增加模型準確度的方法，為準備一較大模型，並使用大模型萃取出之特徵來訓練小模型，是為 knowledge distillation。

一樣使用 MarianMT，將中文 ground truth 簡體中文語句，翻譯回英文，將此 back translated 英文句子與原先英文句子一同送入模型，並計算 loss，再進行更新。

Distilled Data

| | chinese | english |
|---|---|---|
| 0 | 美国缓慢地开始倾听，但并非没有艰难曲折。 | The United States has slowly begun to listen, ... |
| 1 | 抖动是关于颜色混合的技术，使你的作品看起来更圆滑，或者只是创作有趣的材质。 | It's about color blending, making your work lo... |
| 2 | 本文以珲春早第三纪含煤盆地的地质构造背景为依据，分析了煤系地层的岩石学特征。 | This paper analyses the rocky characteristics ... |
| 3 | 第二次事件跟我爷爷的宝贝匣子有关。 | The second incident was related to my grandfat... |
| 4 | 解决这些挑战的途径包括依照麻瓜在南非的经验设立真相与和解委员会。 | These challenges can be addressed through the ... |
| ... | ... | ... |
| 49995 | 你太迟钝了，没有理解这种暗示。 | You're too slow, you don't understand that hint. |
| 49996 | 因此，在这种情况下船舶抵押权人放弃了债务人提供的担保就会影响其他担保人的利益，导致抵押权人的... | Thus, in such a case, the abandonment of the s... |
| 49997 | 第四，削弱行政膨胀。 | Fourthly, it weakens the expansion of the admi... |
| 49998 | 马西莫·奥多声明他不会在世界杯决赛圈比赛结束之前考虑未来的俱乐部。 | Massimo. Odo says he won't think about future ... |
| 49999 | 《冷眼观俗尘》 尚未完工的佛祖雕像，超凡脱俗的神情，似乎在禅悟街巷里匆忙行人的百态人生。 | The unfinished statue of the Buddha, the prepo... |

50000 rows × 2 columns

## 7. Sharpness-aware minimization

在大型語言模型訓練中，常會遇到 error surface 起伏而難以進行 gradient descent 找到最小值，而 SAM(sharpness-aware minimization)[2]，透過兩階段的更新，來進行參數最佳化。與一般 optimization 不同，SAM 並非尋找能使 loss 最小的參數，而是試著將函數更新至 loss 平緩區，由此增加模型泛化能力。

然而經過實驗，此次的模型使用 SAM 並不會使模型有更高的準確度，推測原因為 error surface 較為平緩，並沒有陡峭的 local minima，因此使用 restart scheduler 即可跳脫 local minima，達到高準確度。且使用 SAM 會將模型訓練時間增加接近兩倍，引此本次最後並未使用 SAM 進行訓練。

**Input:** Training set $\mathcal{S} \triangleq \cup_{i=1}^{n}\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}$, Loss function $l : \mathcal{W} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$, Batch size $b$, Step size $\eta > 0$, Neighborhood size $\rho > 0$.
**Output:** Model trained with SAM
Initialize weights $\boldsymbol{w}_0$, $t = 0$;
**while** *not converged* **do**
  Sample batch $\mathcal{B} = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), ...(\boldsymbol{x}_b, \boldsymbol{y}_b)\}$;
  Compute gradient $\nabla_{\boldsymbol{w}} L_{\mathcal{B}}(\boldsymbol{w})$ of the batch's training loss;
  Compute $\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})$ per equation 2;
  Compute gradient approximation for the SAM objective (equation 3): $\boldsymbol{g} = \nabla_{\boldsymbol{w}} L_{\mathcal{B}}(\boldsymbol{w})|_{\boldsymbol{w}+\hat{\boldsymbol{\epsilon}}(\boldsymbol{w})}$;
  Update weights: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \boldsymbol{g}$;
  $t = t + 1$;
**end**
**return** $\boldsymbol{w}_t$

**Algorithm 1:** SAM algorithm

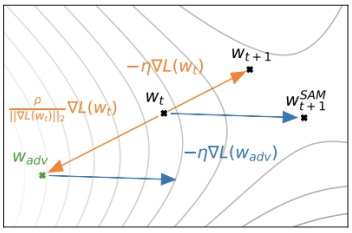



Figure 2: Schematic of the SAM parameter update.

## 8. RMS normalization

為了更進一步增加模型訓練速度，本次 transformer 架構中將原先的 layer normalization 改為 Root Mean Square Layer Normalization [3]。RMSnorm 特點在於，他不考慮 LayerNorm 中的 batch statistics 以及 mean subtraction，直接將 activation scale 為固定的方均根，由此來穩定訓練過程，並提升訓練效率。

| | Weight matrix re-scaling | Weight matrix re-centering | Weight vector re-scaling | Dataset re-scaling | Dataset re-centering | Single training case re-scaling |
|---|---|---|---|---|---|---|
| BatchNorm | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |
| WeightNorm | ✓ | ✗ | ✓ | ✗ | ✗ | ✗ |
| LayerNorm | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ |
| RMSNorm | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ |
| *p*RMSNorm | ✓ | ✗ | ✗ | ✓ | ✗ | ✓ |

## 9. Activation function

本次測試之 activation function 如下:

(1) ReLU:

最簡易之 activation，但較不適合 transformer。

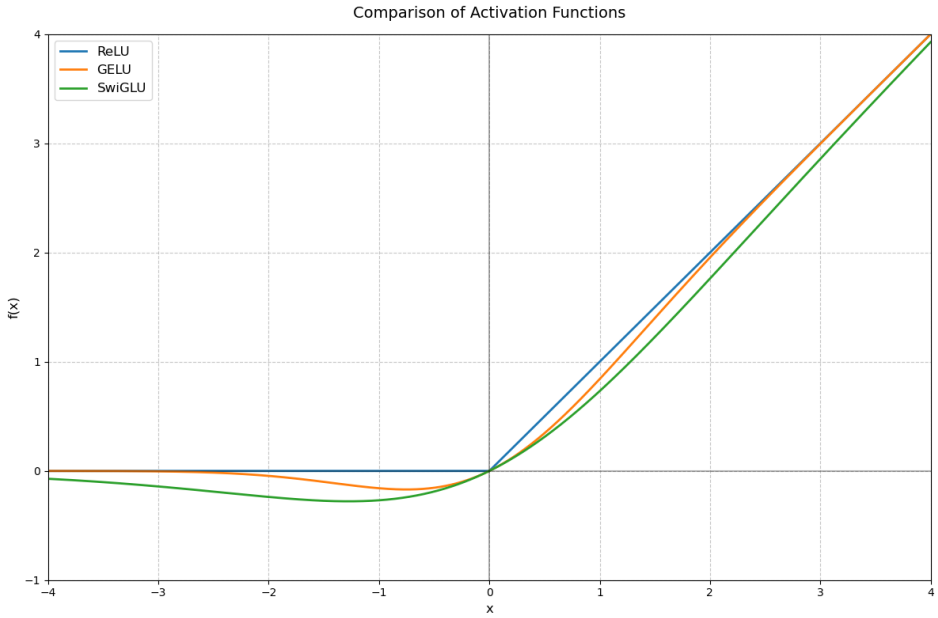
(2) GeLU:

有著比 ReLU 更為平滑的變換曲線，因此對於資料的微小變化能萃取的更加完整；負數不為 0 的特性使 GeLU 的梯度流更加順暢，易於訓練；種種原因使 GeLU 成為 transformer activation function 不二選擇。

(3) SwiGLU(SwiGated Linear Units):

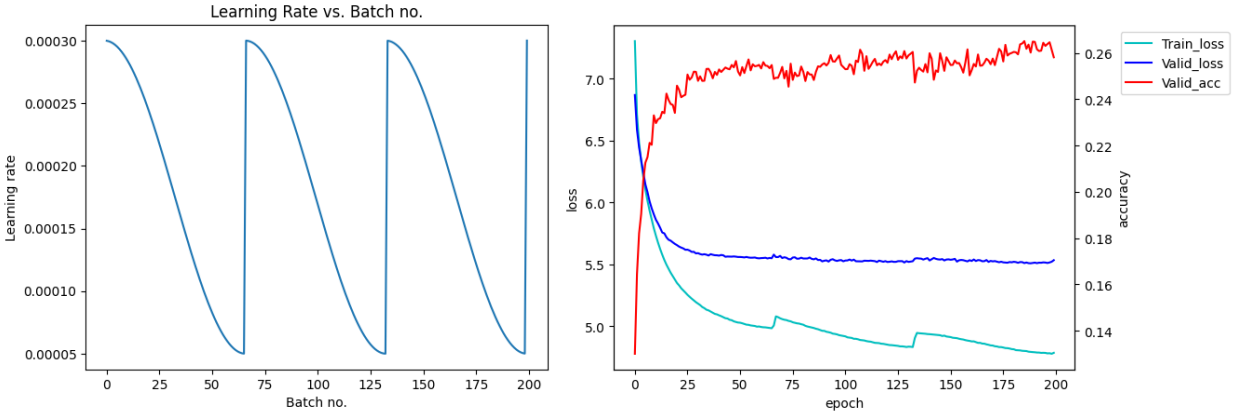
SwiGLU [4]結合了 Swish 和 GLU 的優點，有更平滑的 activation 以及更強的 expressiveness，使其非常適合大規模模型和多樣化資料，包括 NLP 任務。

然而其 gate 特性，使 SwiGLU 使用容易使模型 overfitting，因此本次作業最後採用 GeLU 作為 activation function。




## IV. Result

1. Training loss




2. Validation accuracy

```
Best model was saved at epoch 186, with validation accuracy: 0.2651
```

3. Inference

```
Input:          : 你好，欢迎来到中国
Prediction      : You'll welcome to China.
Ground truth    : Hello, welcome to China.
Bleu Score (1gram):  0.75
Bleu Score (2gram):  0.7071068286895752
Bleu Score (3gram):  0.6299605369567871
Bleu Score (4gram):  0.0
```

```
Input:          : 早上好，很高兴见到你
Prediction      : Good morning, very happy, very happy to see you.
Ground truth    : Good morning, nice to meet you.
Bleu Score (1gram):  0.4444444477558136
Bleu Score (2gram):  0.235702246427536
Bleu Score (3gram):  0.0
Bleu Score (4gram):  0.0
```

```
Input:          : 祝您有个美好的一天
Prediction      : I wish you have a good day.
Ground truth    : Have a nice day.
Bleu Score (1gram):  0.4285714328289032
Bleu Score (2gram):  0.26726123690605164
Bleu Score (3gram):  0.0
Bleu Score (4gram):  0.0
```

## V. Reference


[1] Michael R. Zhang, James Lucas, Geoffrey Hinton, Jimmy Ba, "Lookahead Optimizer: k steps forward, 1 step back"

[2] Pierre Foret, Ariel Kleiner, Hossein Mobahi, Behnam Neyshabur, "Sharpness-Aware Minimization for Efficiently Improving Generalization"

[3] Biao Zhang, Rico Sennrich, "Root Mean Square Layer Normalization"

[4] Noam Shazeer, "GLU Variants Improve Transformer"