**Lab3: Leukemia Classification**

**系級:智能系統 學號:312581006 姓名:張宸瑋**

1. Introduction

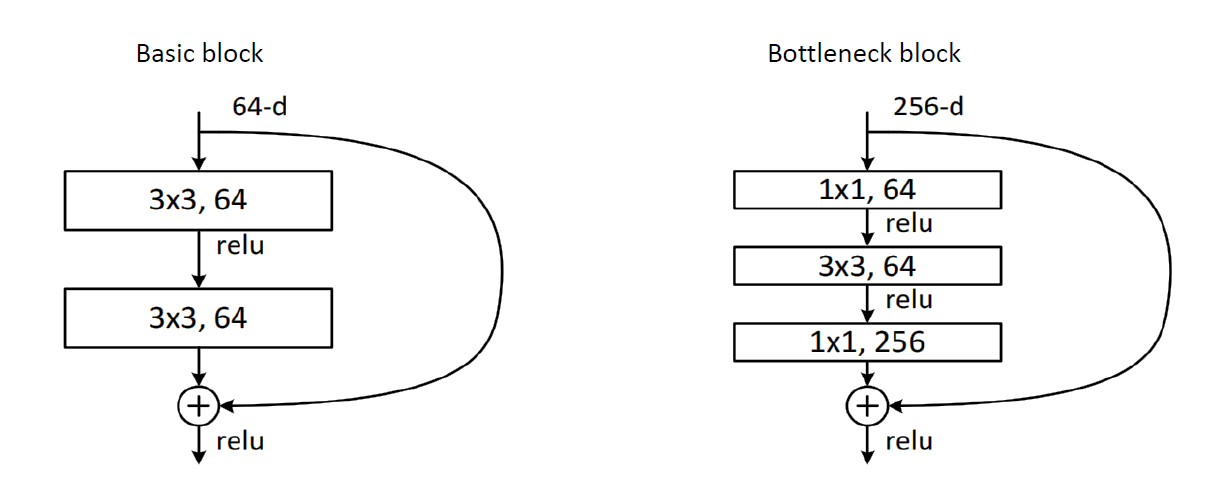
這次lab要實作出ResNet18、ResNet50、ResNet152三種不同的架構，並且自行撰寫dataloader程式，設計自己的數據預處理方法，並且透過ResNet，來對急性淋巴细胞白血病（acute lymphoblastic leukemia,ALL）來做分類，判斷輸入的圖片是正常細胞，還是白血病細胞。

|  |
| --- |
| 分類目的 |

而這次的模型，主要會根據下面這張架構圖來去坐實作。

|  |
| --- |
| https://pic1.xuehuaimg.com/proxy/csdn/https:/img-blog.csdnimg.cn/20190808211534608.png |
| ResNet模型架構  而ResNet最大的特色就是引入了殘差模塊(Residual Block)，透過殘差連接將每個層的輸入映射到輸出，有效的解決了梯度消失與梯度爆炸的問題，也因為這樣子，透過搭建更深的網路，能夠讓模型學到更多的特徵，提升模型的表達能力。 |

以下是兩個不同的殘差模塊的架構差異，在這次實驗中，ResNet18是採用Basic block架構，而ResNet50以及ResNet152是採用Bottleneck block架構。



1. Experiment setups
2. The detail of your model

|  |
| --- |
|  |
| BasicBlock架構實現 |
|  |
| Bottleneck架構實現   |  | | --- | |  | | ResNet18架構 | |  | | ResNet50架構 | |  | | ResNet152架構 | |

1. The details of your Dataloader

|  |
| --- |
|  |
| 資料集目錄 |
|  |
| 透過getData搭配參數mode，來決定要讀哪個csv的檔案，因為這次的實驗我是在kaggle notebook上進行的，所以csv檔以及訓練用圖片都放在kaggle/input這個路徑下。 |
|  |

|  |
| --- |
| 1.\_\_init\_\_(self, root, mode, transform=None)：  - root：數據集的根目錄。  - mode：數據加載模式，用於指定是訓練集、驗證集還是測試集。  - transform：數據預處理和轉換的函數，用於對圖像進行預處理。  2.\_\_len\_\_(self)：  - 返回數據集的樣本數量。  3.\_\_getitem\_\_(self, idx)：  - 根據給定的索引idx，獲取數據集中的一個樣本。  - img\_name：根據索引獲取圖像的文件名。  - image：從文件名加載圖像，並將其轉換為 RGB 格式。  - label：根據索引獲取對應圖像的標籤。  - 如果 transform函數存在，則將圖像應用預處理和轉換。 |

1. Describing your evaluation through the confusion matrix

|  |
| --- |
|  |
| ResNet18 confusion matrix |
|  |
| ResNet50 confusion matrix |
|  |
| ResNet152 confusion matrix |

討論:

從三個模型的confusion martix我們可以看到，在細胞為正常的情況下模型預測是白血球細胞的情況比較高，因此這代表我們可能需要增加正常細胞的訓練樣本，或者是平衡正常細胞跟白血球細胞的樣本數量，來優化我們的模型。

1. Data Preprocessing
   * 1. How you preprocessed your data?

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | |  | | 在這次的預處理中，我先將整份dataset透過上述程式碼先做了預處理，並且保存在資料夾中，供我之後訓練做使用，而這段程式碼最主要的功能是對每張圖像做二值化，並且根據二值化的遮罩裁切我們感興趣的區域(細胞本身)，最後調整成我們所需要的大小，以下為預處理過後的示意圖。 | |  | | 透過上述程式碼所得到的結果 | |  | | 將整份資料集做上述操作後，會得到這份檔案，將這份檔案下載下來，透過這份檔案的資料集來做訓練 | |  | | 上述程式碼是用來對訓練資料進行預處理的pytorch轉換，裡面包括了隨機水平翻轉影像、隨機旋轉影像、隨機應用高斯模糊，以及隨機改變影像亮度、對比度、飽和度、色調，最後再將影像轉換成pytorch張量，並且對影像進行標準化。 | |  | | 而因為評估模型的過程中，我們希望使用原始的訓練集跟驗證集，以及固定的預處理方法來做評估，這樣才可以確保評估的一致性，以及公平性，因此我們僅將資料做pytorch張量轉換，以及標準化。 | |  | | 然而，因為我們需要知道資料的平均值以及標準差，才能夠做標準化，因此我們透過上述的程式碼，來計算出訓練樣本的平均值以及標準差，並套用在我們的標準化轉換參數中。 | |

* + 1. What makes your method special?

在一開始訓練時，我都是先透過CenterCrop，先從圖片的中心剪裁圖片，接著在resize成指定大小，但是這樣子訓練到最後很容易過擬合。並且不管用什麼樣的方法都沒辦法解決，因此最後再查看資料集後，發現有些細胞呈現較不規則的形狀，像下面這兩張細胞圖，如果我採用中心剪裁的方式，很有可能會把這些比較不規則狀的細胞圖特徵給剪裁掉，可能會導致重要的特徵沒辦法包含在剪裁後的區域，導致模型沒辦法訓練到重要特徵，也會導致模型的泛化能力下降，因此最後我採用透過二值化遮罩的方式，裁切出我們感興趣的區域，也就是細胞本身，這樣我們不但能夠使模型訓練到重要特徵，也能提高模型的泛化能力，我認為這個預處理的操作，是我模型準確率提升非常大的一個關鍵，而透過pytorch轉換，包括隨機水平翻轉，隨機選轉影像等等，也能夠改善我的模型的泛化能力。

|  |  |
| --- | --- |
|  |  |
| 不規則狀細胞圖 | |

1. Experimental results
   * 1. The highest testing accuracy

|  |
| --- |
|  |
| The highest testing accuracy |

* + 1. Comparison figures

|  |
| --- |
|  |
| ResNet18 accuracy growth curve |
|  |
| ResNet50 accuracy growth curve |
|  |
| ResNet152 accuracy growth curve |
|  |
| Compare and visualize the accuracy trend between the 3 model architectures |

討論:

在這裡我們可以分別看到三個不同架構的準確率成長圖，都在某一個epoch後，training accuracy呈現下降的趨勢，那是因為我發現模型已經開始呈現過擬合的現象，因此我透過正則化的技術，以及pytorch transforms模組來增加多樣性，來提升模組的泛化能力，因此我們可以看到，在增加這些技術後，模型看起來有更好的擬合，並且提升了它的泛化能力，而我們也可以看到三個模型的準確率比較圖中，可以看到模型在評估方面的準確率表現較不穩定，這裡我在想可能是因為batch size設定32比較小的關係，導致模型梯度的更新較不穩定，但是如果將batch size調大，可能會導致內存不夠，因此這是我們在訓練時要權衡的。

1. Discussion

透過這次的實驗，讓我了解到ResNet的網路結構，以及它的設計精隨，也因為這次要透過三個模型來對我們的資料集做訓練，因為每個網路的複雜度不一樣，因此在初始化超參數的時候，是一個非常重要的關鍵，如果超參數的配置適當，我們在訓練過程中會更為順利，而我在這次的訓練中，我主要的策略是，先透過一組適當的超參數做訓練，如果練到最後模型的表現呈現過擬合的狀況，就會取消訓練，並且記錄最後訓練以及在這輪訓練中模型表現最好的參數配置，並且透過更改像是weight decay的值，或者透過pytorch的transforms增加數據多樣性，來提升模型的泛化能力，透過這次的實驗，除了讓我對ResNet有了更多的認識，並且體會到它的奧妙，也讓我對模型訓練的整個過程更加熟悉，並且能透過不同的方式來改善模型的效能。