**Lab6:** **Let’s Play DDPM**

**系級:智能系統 學號:312581006 姓名:張宸瑋**

1. Report
2. Introdunction

|  |
| --- |
| 這次的lab主要是要實現DDPM，根據多標籤條件，來生成符合的圖片。我們必須透過ivlevr dataset訓練一個diffusion model，並且透過test.json以及new\_test.json來生成指定條件的圖片。 |
|  |
| Conditional DDPM示意圖 |
| 而DDPM的主要概念是，將輸入圖像經過多次的添加噪音，並且透過我們的網路，學習添加到的噪音，然後透過逐漸去噪的方式，來生成我們最後要生成的目標圖像。以下為示意圖。 |
|  |
| DDPM正向傳播示意圖 |
|  |
| DDPM逆向擴散示意圖 |
|  |
| 演算法架構 |

1. Implementation details

|  |
| --- |
| Design of UNet |
|  |
| Type of DDPM and Noise Schedule |
|  |

|  |
| --- |
| Sampling |
|  |
| Loss function |
|  |
|  |
| 在loss function的選擇上我使用Mean Square Error，來計算預測出的噪音，以及添加到輸入的噪音的loss。 |
| Specify the hyperparameters (learning rate, epochs, etc.) |
| * 1. Learning rate : 0.001   2. Epochs : 150   3. Optimizer : AdamW   4. lr\_scheduler = get\_cosine\_schedule\_with\_warmup(   optimizer=opt,  num\_warmup\_steps=500,  num\_training\_steps=(len(train\_loader)\* n\_epochs),  )   * 1. Timestep : 1000 |

1. Result and discussion

* Show your accuracy screenshot based on the testing data

|  |
| --- |
|  |
| Testing Results |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | - Show your synthetic image grids and a progressive generation image. | |  | |  | | Test.json | |  | |  | |
| New\_test.json |

|  |  |
| --- | --- |
|  |  |
| Timestep = 0 | Timestep = 50 |
|  |  |
| Timestep = 150 | Timestep = 200 |
|  |  |
| Timestep = 250 | Timestep = 300 |
|  |  |
| Timestep = 350 | Timestep = 400 |
|  |  |
| Timestep = 450 | Timestep = 500 |
|  |  |
| Timestep = 550 | Timestep = 600 |
|  |  |
| Timestep = 650 | Timestep = 700 |
|  |  |
| Timestep = 750 | Timestep = 800 |
|  |  |
| Timestep = 850 | Timestep = 900 |
|  |  |
| Timestep = 950 | Timestep = 999 |
| Progressive generation image | |

* Discuss the results of different model architectures or methods.

|  |
| --- |
| 介紹: |
| 在這次的實作中，我嘗試使用DDIM來取代DDPM，對於DDPM來說，它需要較長的擴散步數，才能得到好的結果，假設我的擴散步數為1000步的話，那我再去噪的時候，就必須推理1000次，才能夠得到原圖，這樣是非常耗時的，因此透過DDIM的方式，它主要的目的是，透過更小的採樣步數，來加速我們的生成過程，它跟DDPM的訓練目的是一樣的，只是它不再限制擴散過程必須是一個馬爾科夫鏈。 |
| DDIM實現 |
|  |
| 實驗結果比較 |
|  |
| DDIM |
|  |
| DDPM |
| 在這裡我們可以看到，兩者在推理的速度上的差異，假設我的DDIM的num\_inference\_step = 50，其推理速度相較於DDPM是快20倍的。 |
|  |
| DDIM推理結果 |
|  |
| DDPM推理結果 |
| 這裡我們也可以看到，即使DDIM相較於DDPM，其採樣步數較小，也沒有因為這樣而失去圖片生成的品質，反而還優於DDPM。 |

1. Experimental results

|  |
| --- |
|  |
|  |
| test.json |
|  |
|  |
| new\_test.json |

程式碼執行注意事項:

1. 檔案目錄結構必須如下

