深度學習

HW13

學號 : B103012002

姓名 : 林凡皓

VAEs和GANs在訓練過程中對超參數非常敏感，而且通常需要很多訓練epochs。為了讓這次練習更加容易完成，我們使用MNIST資料及來做練習。MNIST資料即包含60000張訓練圖片與10000張測試圖片，每一張圖片都包含一個位於黑色背景中心位置上的白色數字。

1. FC-VAE Encoder

Fully-connected VAE network encoder會接收一張圖片並將他flatten，然後送進三層的Linear + ReLU layers。我們會使用hidden dimension representation和兩個Linear layer來預測posterior和posterior log variance。架構大致如下

一張含有 行, 圖表, 圓形, 螢幕擷取畫面 的圖片

自動產生的描述

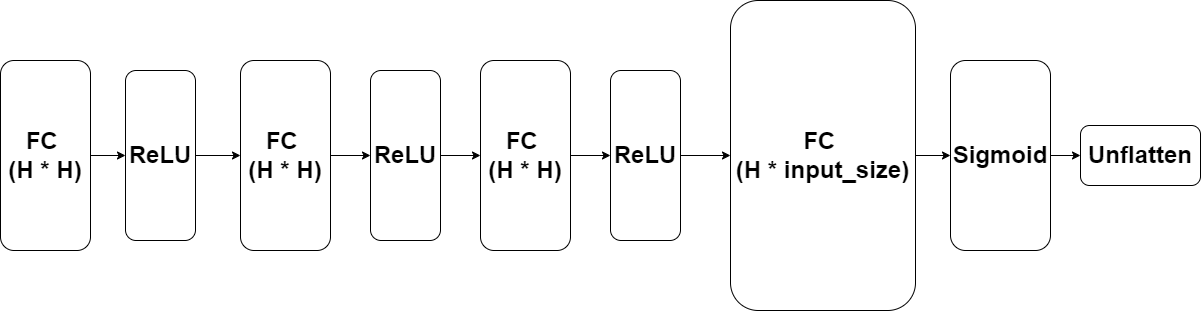
其中中間藍色的Fully connected layer會有三個。

* 實現方式 :

利用nn.Sequential來將Flatten、Linear、ReLU串接起來，經過此sequential model後的shape = (N, hidden\_dimention)。在此Sequential model後方接上兩個Linear layer，分別用來將hidden features映射到預測posterior mean和posterior log variance，mean和log variance的形狀皆為(N, Z)。

1. FC-VAE Decoder

Decoder會用latent space representation來還原圖片，其架構如下



* 實現方法 :

有點類似於encoder的相反。利用nn.Sequential來實現decoder架構。該架構會接收shape = (N, Z)的輸入，並輸出與encoder輸入相同形狀。

1. Reparametrization

在forward path中，我們會需要根據mean和variance來估計posterior z。一種最簡單的實現方式為根據我們的mean和variance產生normal distribution，但是我們會需要對這個分布做backpropagation，但是這個分布為不可微分的。因此我們改為從一個固定的分布中聲稱成初始隨機數據，並計算z作為(, variance, mean)的函數，也就是

z = mean + variance\*。

* 實現方法 :

透過對log variance取指數後開根號得到variance，並根據z = mean + variance\* 計算z。

* 執行結果 :



1. FC-VAE Forward

* 實現方法 :

將輸入送進encoder後得到encoder output，並根據此輸出去計算mean跟log variance。將mean和log variance用來估計posterior z (raparametrization)，最後再將z送進decoder得到還原後的圖片。

1. Loss Function

VAE的loss如下



其中包含reconstruction loss和KL divergence的部分。

Reconstruction loss用來計算輸入圖片像素與解碼後圖片像素的binary cross entropy，KL divergence是強制潛在空間分布接近先驗分布，這部分需要實現向量化並應用minibatch。

* 實現方法 :

Reconstruction loss的部分即為binary cross entropy，可以直接透過nn.functional.binary\_corss\_entropy來實現。KL diverence可以根據其公式實現，公式為



* 驗證結果 :



1. Train a model and Visualize result

利用剛才建立好的VAE模型與loss function來訓練模型，經過10個epochs的訓練後，結果如下

一張含有 文字, 字型, 螢幕擷取畫面, 印刷術 的圖片

自動產生的描述

接著我們利用訓練好的VAE來生成新的圖片。我們可以初始化某個分布作為latent space z，並將latent space送進decoder中來產生新的圖片，產生的結果如下



可以看到大部分結果算是清楚，但是9、7、6、0在某些地方會有不容易判別的現象。

1. Latent Space Interpolation

在latent space中，我們可以在兩個latent vector之間進行內插，生成新的latent vector，這些內插後的向量可以用來產生照片，從而觀察模型平滑過度的效果。結果如下

一張含有 樣式, Rectangle, 正方形, 對稱 的圖片

自動產生的描述

1. Conditional FC-VAE

Conditional VAE為VAE的變形，他可以利用標籤訊息來進行條件式生成，這樣我們就可以選擇生成特並的類別數據。這邊實作的CVAE架構與前面實作的VAE架構相同，差別只在於CVAE在輸入與latent space中加入one-hot 標籤向量。

* 實現方法 :

使用VAE的架構作調整，需要調整個地方如下

1. Encoder first layer : 除了接收flatten後的圖像之外，還要去接收一個one-hot標籤向量
2. Decoder first layer : decoder first layer現在變成是將latent space + one-hot向量映射到hidden dimension。
3. Forward path : 在forward path部分，送進encoder之前需要將flatten後的圖像與one-hot向量合併，進到decoder之前需要將latent space與one-hot向量合併。
4. Train model

用與訓練VAE時相同的訓練方式來訓練CVAE，經過10個epochs後，訓練結果如下

一張含有 文字, 字型, 螢幕擷取畫面, 印刷術 的圖片

自動產生的描述

訓練好模型後，我們可以根據我們想要的類別來產生資料，這邊依序產生0~9的數字，結果如下



1. 額外嘗試

* 修改與重新訓練VAE :

嘗試修改VAE架構以及訓練參數，看看訓練結果會有甚麼影響。

以下幾種嘗試都是訓練10個epochs。

1. 將Linear + ReLU的層數降到2 :

調整encoder和decoder的架構，將Linear + ReLU層的數量調整至2，並訓練10個epochs結果如下

一張含有 文字, 字型, 螢幕擷取畫面, 黑與白 的圖片

自動產生的描述

最終loss為104，利用此模型來生成一些圖片，結果如下



可以看到圖片跟三層Linear + ReLU的模型比起來，更加模糊且更難去判斷數字為多少。

1. 將Linear + ReLU的層數升到4 :

調整encoder和decoder的架構，將Linear + ReLU層的數量調整至4，並訓練10個epochs結果如下

一張含有 文字, 字型, 螢幕擷取畫面, 印刷術 的圖片

自動產生的描述

最終loss為124，比兩層和三層Linear + ReLU的模型都還要高。

利用此模型生成圖片，結果如下



由結果可以看到這是目前生成出來最清楚的圖片，每一張都可以很簡單的辨認出數字為多少。

由調整架構的嘗試可以發現到，提高Linear + ReLU的層數會造成模型的loss提高，但是在生成圖片的表現上，較多Linear + ReLU可以生成出更加清楚以及容易辨別的圖片，我猜想這是因為更多層的Linear + ReLU可以萃取出更多的特徵，讓還原圖片時有更高的準確度。

1. 提高latent\_size :

架構採用三層的Linear + ReLU。訓練方式與先前都相同，差別只有將latent\_size提升到30。訓練10個epochs之後結果如下

一張含有 文字, 字型, 螢幕擷取畫面, 印刷術 的圖片

自動產生的描述

利用此模型來生成圖片，結果如下



可以看到生成的圖片很多都無法辨別數字，比latent\_size = 15時的表現還要糟糕。

1. 減少latent\_size :

架構採用三層的Linear + ReLU。訓練方式與先前都相同，差別只有將latent\_size降低到10。訓練10個epochs之後結果如下

一張含有 文字, 字型, 螢幕擷取畫面, 黑色 的圖片

自動產生的描述

利用此模型來生成圖片，結果如下



可以看到表現也沒有latent\_size = 15時來的好。

由這次嘗試可以發現到latent size並不是越大越好或是越小越好，而是根據使用場景會有合適的大小。要怎麼去決定latent size，這邊有一些建議 :

* 嘗試不同latent size，並根據輸出結果選擇合適的latent size。
* 對訓練數據進行SVD奇異值分解，非零的奇異值可以提供一個大概的latent size選擇。
* 設定比較大的latent size但是在loss function中加入regularization，讓encoder被迫只能使用所需要的維度，以減少overfitting。
* 修改與重新訓練CVAE

嘗試修改CVAE架構以及訓練參數，看看訓練結果會有甚麼影響。

以下幾種嘗試都是訓練10個epochs。

1. 將Linear + ReLU的層數降到2 :

調整encoder和decoder的架構，將Linear + ReLU層的數量調整至2，並訓練10個epochs結果如下

一張含有 文字, 字型, 螢幕擷取畫面, 印刷術 的圖片

自動產生的描述

最終loss為104，但是在過程中loss可以降到100以下。利用此模型來生成一些圖片，結果如下



可以看到圖片跟三層Linear + ReLU的模型比起來，更加模糊且更難去判斷數字為多少。

1. 將Linear + ReLU的層數升到4 :

調整encoder和decoder的架構，將Linear + ReLU層的數量調整至4，並訓練10個epochs結果如下

一張含有 文字, 字型, 螢幕擷取畫面, 印刷術 的圖片

自動產生的描述

最終loss為112，比兩層和三層Linear + ReLU的模型都還要高。

利用此模型生成圖片，結果如下



由結果可以看到每一張都可以很簡單的辨認出數字為多少，與先前嘗試過的CVAE比較起來也清楚許多。

由調整架構的嘗試可以發現到，CVAE的結果基本上與VAE的結果一樣，這其實是預料之內的，畢竟兩者的架構與運作原理基本上一致。

1. 提高latent\_size :

架構採用三層的Linear + ReLU。訓練方式與先前都相同，差別只有將latent\_size提升到30。訓練10個epochs之後結果如下

一張含有 文字, 字型, 螢幕擷取畫面, 印刷術 的圖片

自動產生的描述

利用此模型來生成圖片，結果如下



可以看到生成的圖片其實算是清楚，跟latent\_size = 15時的表現相差不大。

1. 減少latent\_size :

架構採用三層的Linear + ReLU。訓練方式與先前都相同，差別只有將latent\_size降低到10。訓練10個epochs之後結果如下

一張含有 文字, 字型, 螢幕擷取畫面, 印刷術 的圖片

自動產生的描述

利用此模型來生成圖片，結果如下



可以看到表現跟之前嘗試的相差不大。

由這次嘗試可以發現到比起VAE，CVAE似乎對latent size的敏感度沒有那麼高，原因我猜想是因為在訓練CVAE時會加入標籤的資訊，模型可以根據標籤去抓取到該數字特徵，有點類似於分類器的概念，因此在還原或是生成圖片時，模型可以更好的去根據使用者給定的標籤然後根據該標籤之特徵來生成圖片。

1. Reference

[1] BC\_Wang “變分自編碼器(VAE)的代碼理解” <https://blog.csdn.net/B_C_Wang/article/details/74908408>

[2] StackExchange “ What is an appropriate size for a latent space of (variational) autoencoders and how it varies with the features of the images ?” <https://ai.stackexchange.com/questions/37272/what-is-an-appropriate-size-for-a-latent-space-of-variational-autoencoders-and>

[5] OpenAI. (2023). ChatGPT (Mar 14 version) [Large language model]. <https://chat.openai.com/>