Machine Learning - Homework 9

資工四 B05902023 李澤諺

May 21, 2020

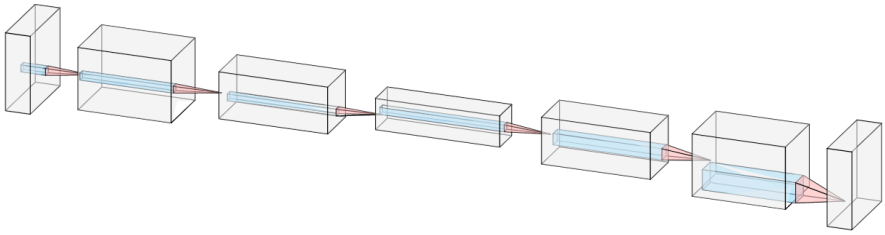
1. (3%) 請至少使用兩種方法 (autoencoder 架構、optimizer、data preprocessing、後續降維方法、clustering 算法等等) 來改進 baseline code 的 accuracy。

(1) 記錄改進前、後的 accuracy 分別爲多少。

(2) 使用改進前、後的方法，分別將 val data 的降維結果 (embedding) 與他們對應的 label 畫出來。

(3) 盡量詳細說明你做了哪些改進。

以下爲 baseline 的作法，首先，其會將所有圖片中的 pixel 數值皆 normalize 到 -1 和 1 之間，以此進行 data preprocessing，接著，以下爲 baseline 所使用的 Autoencoder 架構：



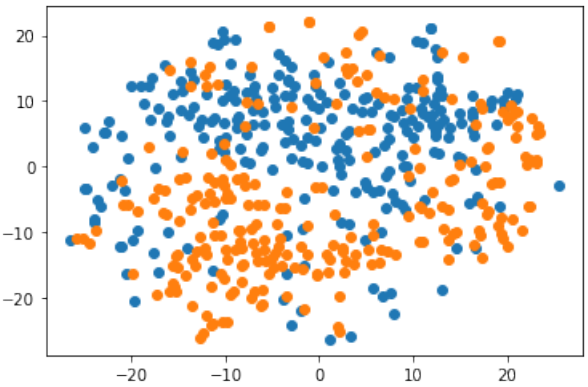
encoder	Conv2d(3 , 64 , kernel_size = 3 , stride = 1 , padding = 1)
	ReLU()
	MaxPool2d(2)
	Conv2d(64 , 128 , kernel_size = 3 , stride = 1 , padding = 1)
	ReLU()
	MaxPool2d(2)
	Conv2d(128 , 256 , kernel_size = 3 , stride = 1 , padding = 1)
	ReLU()
	MaxPool2d(2)

decoder	ConvTranspose2d(256 , 128 , kernel_size = 5 , stride = 1)
	ReLU()
	ConvTranspose2d(128 , 64 , kernel_size = 9 , stride = 1)
	ReLU()
	ConvTranspose2d(64 , 3 , kernel_size = 17 , stride = 1)
	Tanh()

接著，baseline 使用了 Adam 作爲 optimizer，並使用 mean square error 作爲 loss function，以此訓練 Autoencoder，其中 batch size 爲 64，learning rate 爲 0.00001，weight decay 爲 0.00001，訓練了 100 個 epoch，以此得到最後的 model，最後，baseline 會使用訓練好的 Autoencoder 對圖片進行第一次的 dimension reduction，以此將其降到 4096 維，再使用 kernel PCA 進行第二次的 dimension reduction，其中 kernel 爲 RBF，以此將其降到 200 維，再使用 t-SNE 進行第三次的 dimension reduction，以此將其降到 2 維，最後使用 mini-batch K-means 進行 clustering，此外，在以上的過程中，所有的 random seed 皆設爲 0，以下爲 baseline 所得到的 accuracy：

Validation	Test	
	Public	Private
0.64200	0.67836	0.67859

下圖爲 baseline 所得到的 dimension reduction 結果：



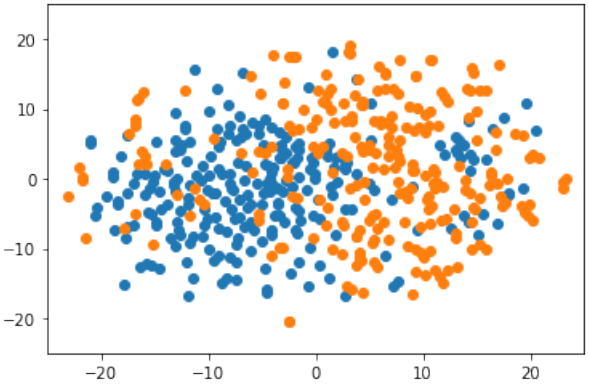
接著，我對 baseline 的方法進行了兩個修改，第一個修改爲我對圖片進行了 data augmentation，我所使用的 data augmentation 如下表所示：

RandomAffine(15 , translate = (0.1 , 0.1) , scale = (0.9 , 1.1))
RandomHorizontalFlip()
ColorJitter(brightness = 0.1 , contrast = 0.1 , saturation = 0.1 , hue = 0.1)

第二個修改爲我調整了一些 hyper-parameter，我將 batch size 改爲 16，並將 learning rate 改爲 0.0001，以下爲在做完這兩個修改之後所得到的 accuracy：

Validation	Test	
	Public	Private
0.76600	0.84141	0.84188

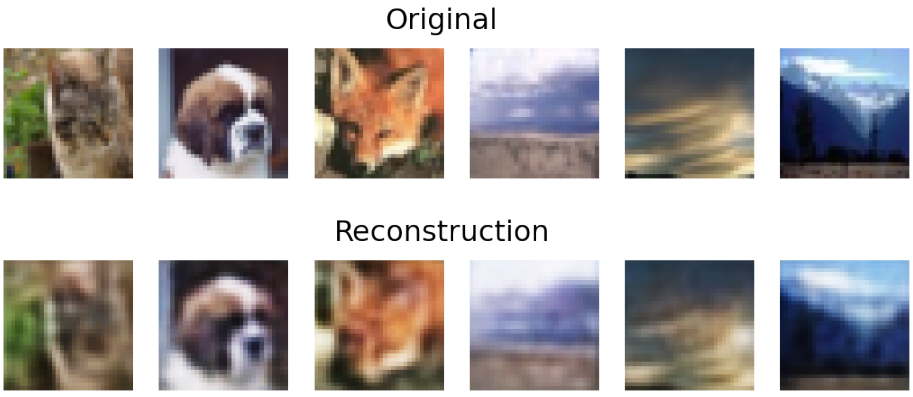
下圖爲改進後的方法所得到的 dimension reduction 結果：



(事實上，我有試過在不改變 baseline 的作法下，隨機給定 random seed，結果發現 baseline 所得到的 accuracy 變化非常大，甚至可以非常接近 strong baseline 的 accuracy，並且由第 3 題可知，Autoencoder 的 loss 越低，並不會使得最後的 accuracy 隨之變高，由此可知在本次作業中要達到很高的 accuracy 也需要一定的運氣成分。)

2. (1%) 使用你 accuracy 最高的 autoencoder，從 trainX 中，取出 index 1、2、3、6、7、9 這 6 張圖片。

(1) 畫出他們的原圖以及 reconstruct 之後的圖片。

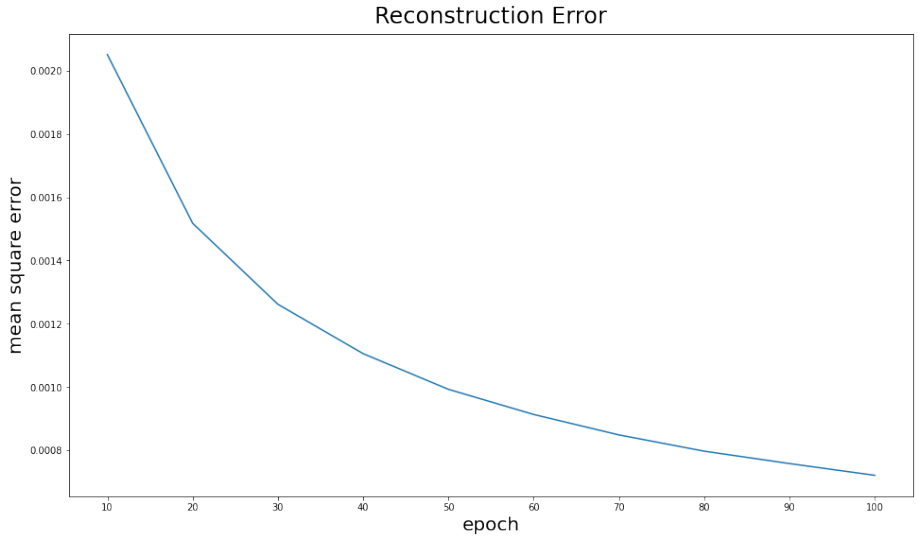


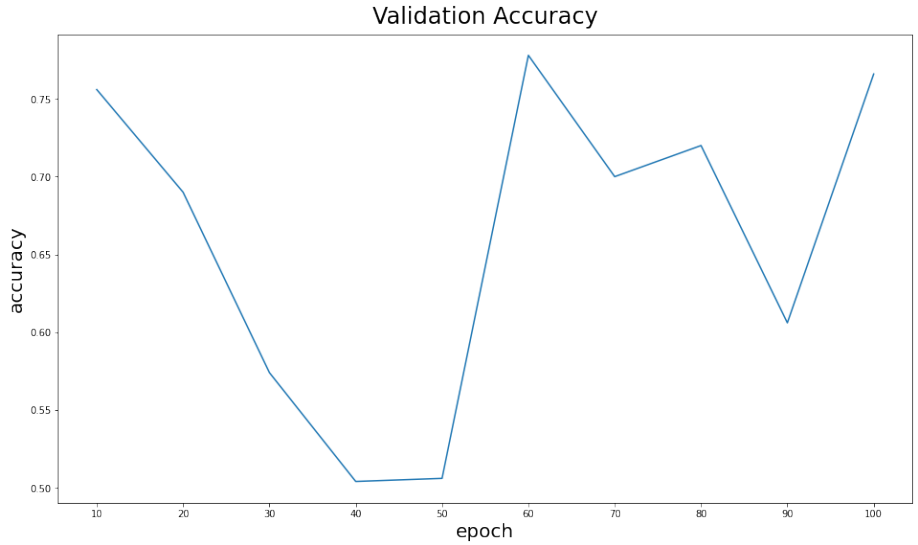
3. (2%) 在 autoencoder 的訓練過程中，至少挑選 10 個 checkpoints。

(1) 請用 model 的 reconstruction error (用所有的 trainX 計算 MSE) 和 val accuracy 對那些 checkpoints 作圖。

(2) 簡單說明你觀察到的現象。

我在改進的方法之中，在訓練 Autoencoder 時每 10 個 epoch 就計算目前的 reconstruction error 和 validation accuracy，結果如下圖所示：





由此可以看出，Autoencoder 的 reconstruction error 越低，並不會使得 validation accuracy 隨之變高，其可能是因爲 Autoencoder 爲了降低 reconstruction

error，有時可能會找出極端而不具代表性的 feature，進而使得難以 clustering 所致。