Machine Learning (HW02)

Course： NCTU-ECM5094-ML

***ID：**309505002 ***Name：**鄭紹文

Q1：Sequential Bayesian Learning.

1.

```
Data size N = 5
※mean vector Mn :
[[ 15.3918259 ]
 [-27.24893207]
 [ 13.76463907]]
※covariance matrix Sn :
[[ 269.04934203 -577.62671303  361.07317844]
 [-577.62671303 1261.47181152 -803.66739994]
 [ 361.07317844 -803.66739994  523.15065575]]

Data size N = 10
※mean vector Mn :
[[ 15.3974152 ]
 [-27.40629378]
 [ 14.00195039]]
※covariance matrix Sn :
[[ 168.04688541 -341.15540081  202.52975073]
 [-341.15540081  706.27405822 -429.99999167]
 [ 202.52975073 -429.99999167  270.34416277]]

Data size N = 30
※mean vector Mn :
[[ 19.22763557]
 [-35.78002777]
 [ 19.39941303]]
※covariance matrix Sn :
[[ 24.12006368 -54.9357445   37.11498943]
 [-54.9357445  127.7793514  -88.33383235]
 [ 37.11498943 -88.33383235  62.70588466]]

Data size N = 80
※mean vector Mn :
[[ 18.85466897]
 [-34.50677874]
 [ 18.21950653]]
※covariance matrix Sn :
[[  6.59105561 -14.94010939  10.02307411]
 [-14.94010939  34.88289614 -24.16334859]
 [ 10.02307411 -24.16334859  17.34855927]]
```

fig. 1

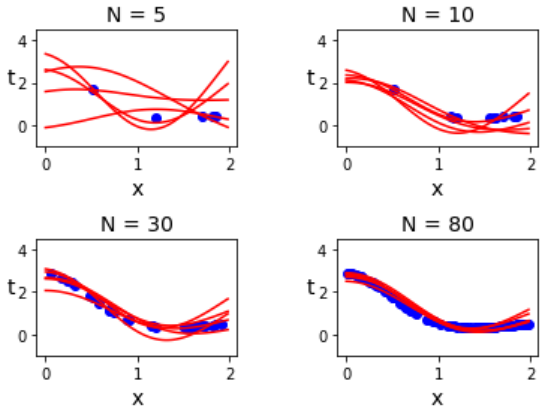


fig. 2

2.

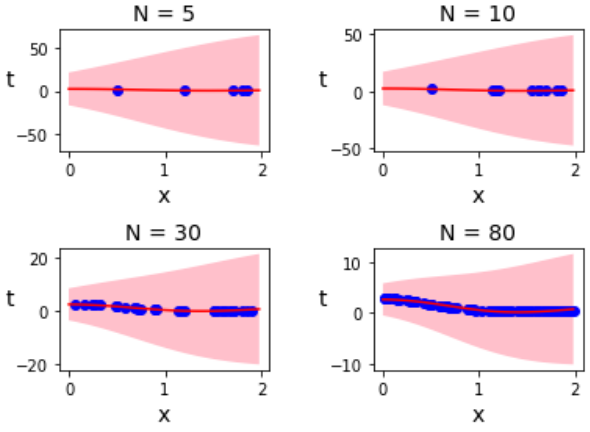


fig. 3

3.

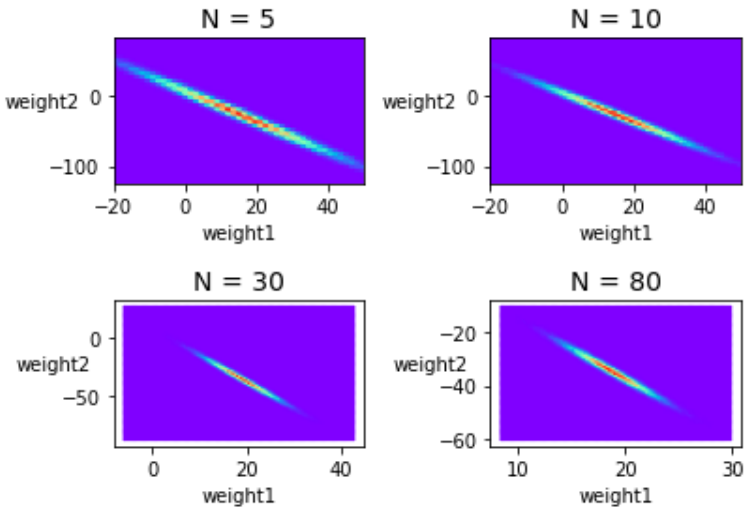


fig. 4

4.

Generate five curves samples from the parameter posterior distribution.

$$p(w|\mathbf{t})=N(w|mN, SN) \quad \text{(func-1.1)}$$

$$mN= SN(SN^{-1}-1mN^{-1}+\beta\Phi T\boldsymbol{t}) \quad \text{(func-1.2)}$$

$$SN^{-1}=SN^{-1}-1+\beta\Phi T\Phi \quad \text{(func-1.3)}$$

首先將 data set 分成 data size N(分別為 5、10、30、80)，並使用 Sigmoidal function $\Phi_j(x) = \sigma(\frac{x-\mu_j}{s})$ 為 basis function，再由 func-1.2、func-1.3 得到 posterior distribution 中的 mean vector(m_N)、covariance matrix(S_N)。

第一小題：fig.2 為由所得之 mean vector、covariance matrix 所形成的高斯分布，利用 np.random.multivariate_normal 語法隨機產生 w，隨機抽樣 5 個 w，並利用 5 個不一樣的 w 對 dataset 中的抽樣點算出對應的 t，並畫出相應的 5 條 curve；同時觀察利用一次增加一筆新的資料的方式且透過 func-1.2、func-1.3 更新 w 的 mean、covariance 可知，每當加入新的資料與資料量增加 w 的 covariance 會逐漸變小。由 fig.1 與 fig.2 可知，當資料數較少時(N=5)，會導致模型的 covariance 較大，然隨著資料數(N)增加，curve 可以 fit 的越好，也越平滑。

第二小題：當加入新的資料點時，此模型的 predictive distribution of target value t 會經過新資料點的附近，使得在此新資料點的 covariance 會變較小；fig.3 為根據課本上公式求出每個Φ的 variance：$\sigma_N^2(x) = \frac{1}{\beta} + \phi(x)^T S_N \phi(x)$ (func-1.4)，由於 covariance 必為正，無法看出隨資料數增加，covariance 有逐漸減小的趨勢，所以將式 func-1.4 中的 $\frac{1}{\beta}$ 去掉後所得之圖，如 fig.3 所示，隨資料數增加，covariance 會有逐漸減小的趨勢。

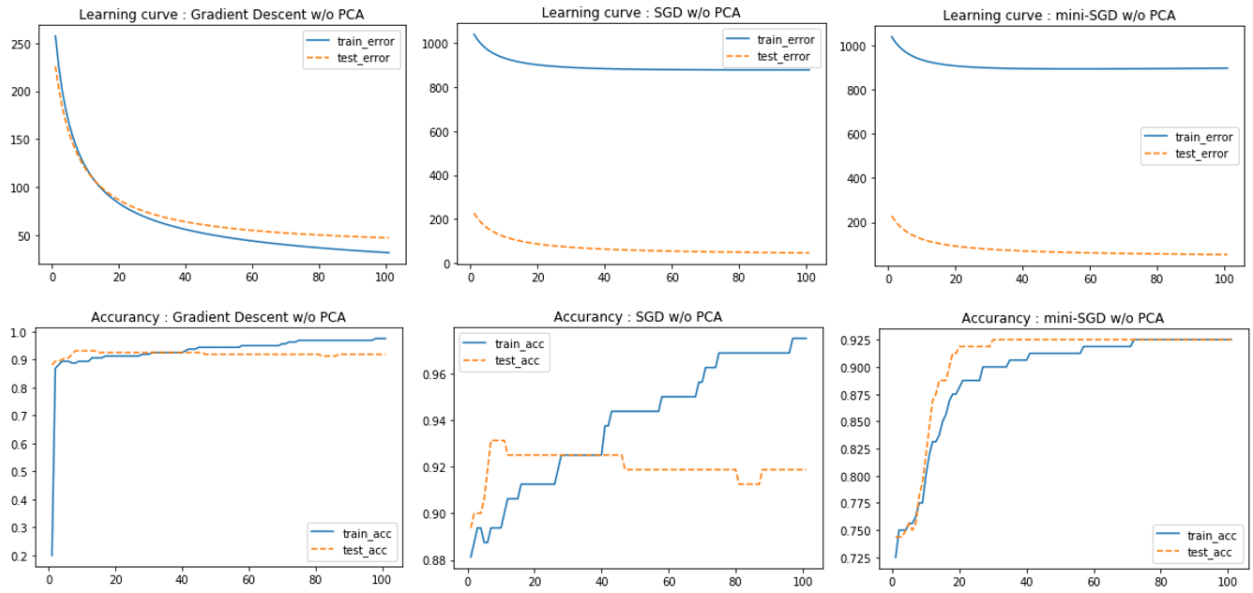
求 Target Value 公式：$p(t|x, t, \alpha, \beta) = N(t|m_N^T\phi(x), \sigma_N^2(x))$

第三小題:fig.4 為透過 np.random.multinomial_normal 隨機產生 100000 組不同的 w，可看出當迭代次數多且隨著資料點增加，w 的 covariance 會根據每一次迭代逐漸變小，w 的離散程度會越來越小(covariance↓)。

Q2：Logistic Regression.

1.

(a).



(b).

```
classification result
TYPE : Gradient Descent
w/o PCA
Training accurancy : 0.975
Test accurancy     : 0.91875
```

```
classification result
TYPE : SGD
w/o PCA
Training accurancy : 0.975
Test accurancy     : 0.9125
```

```
classification result
TYPE : mini-SGD
w/o PCA
Training accurancy : 0.9
Test accurancy     : 0.875
```

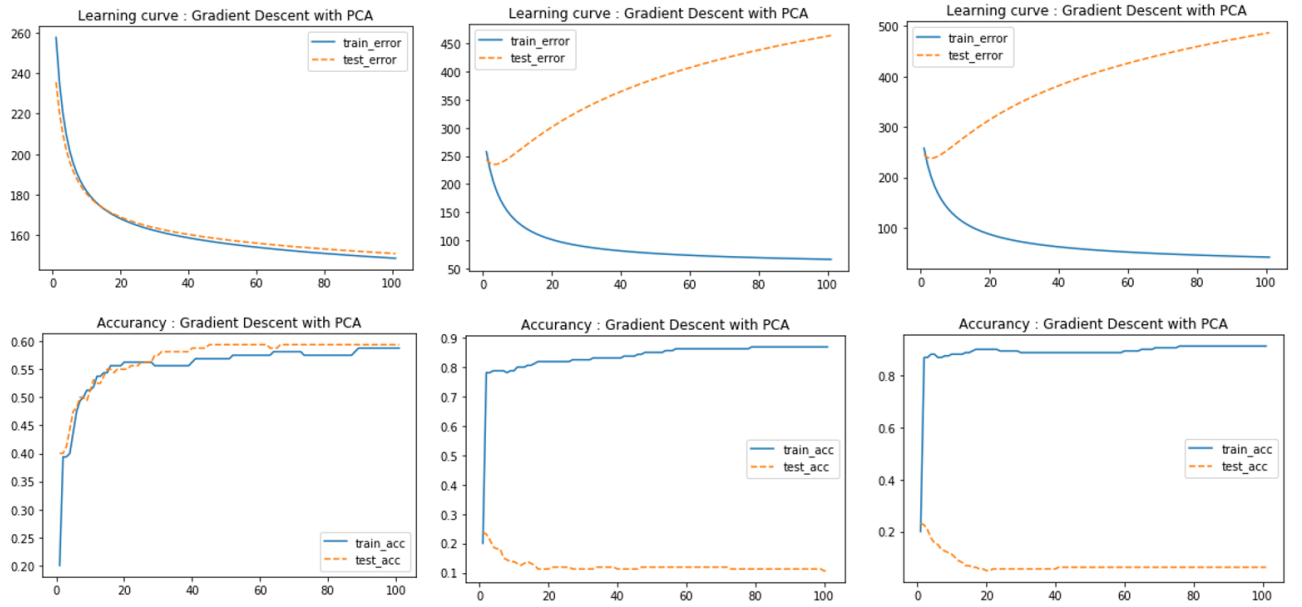
2.

(a). GD with PCA :

```
2-2.GD with PCA 2
classification result
TYPE : Gradient Descent
with PCA
Training accurancy : 0.5875
Test accurancy     : 0.59375
```

```
2-2.GD with PCA 5
classification result
TYPE : Gradient Descent
with PCA
Training accurancy : 0.86875
Test accurancy     : 0.10625
```

```
2-2.GD with PCA 10
classification result
TYPE : Gradient Descent
with PCA
Training accurancy : 0.9125
Test accurancy     : 0.0625
```

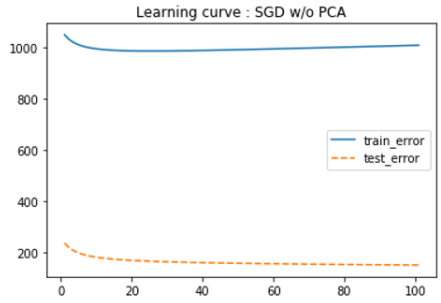


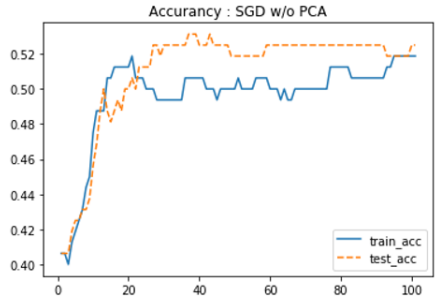
SGD with PCA:

2-2.SGD with PCA 2

classification result

```
TYPE : SGD
with PCA
Training accurancy : 0.51875
Test accurancy     : 0.525
```

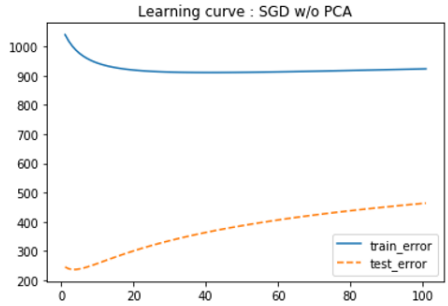


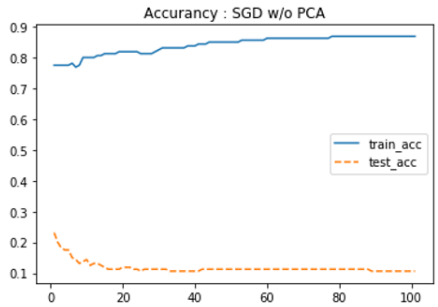


2-2.SGD with PCA 5

classification result

```
TYPE : SGD
with PCA
Training accurancy : 0.86875
Test accurancy     : 0.10625
```

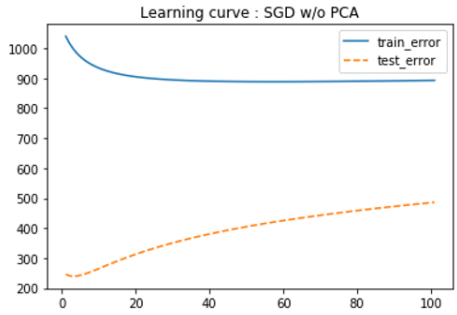


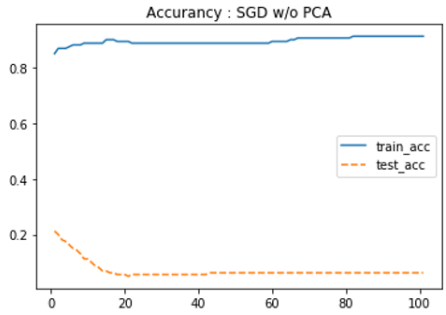


2-2.SGD with PCA 10

classification result

```
TYPE : SGD
with PCA
Training accurancy : 0.9125
Test accurancy     : 0.0625
```



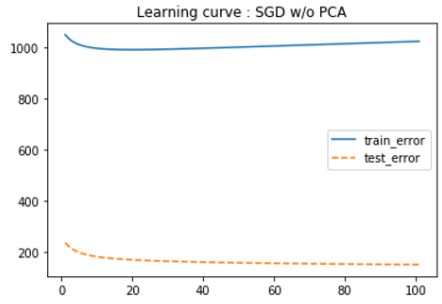


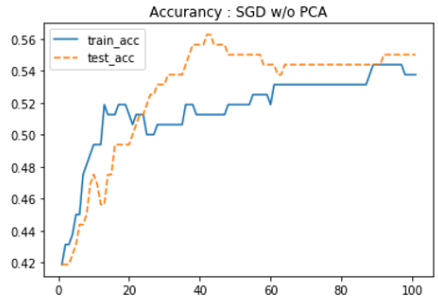
Mini SGD with PCA:

2-2.miniSGD with PCA 2

classification result

```
TYPE : SGD
with PCA
Training accurancy : 0.5375
Test accurancy     : 0.55
```

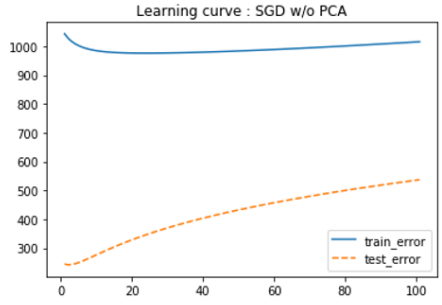


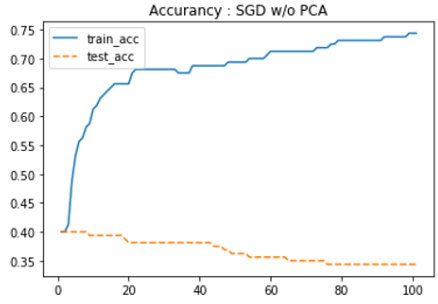


2-2.miniSGD with PCA 5

classification result

```
TYPE : SGD
with PCA
Training accurancy : 0.74375
Test accurancy     : 0.34375
```

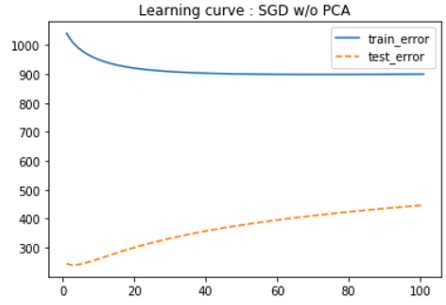


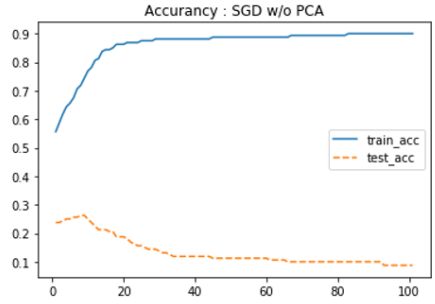


2-2.miniSGD with PCA 10

classification result

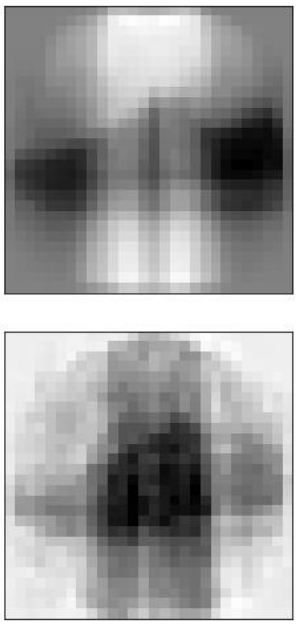
```
TYPE : SGD
with PCA
Training accurancy : 0.9
Test accurancy     : 0.0875
```



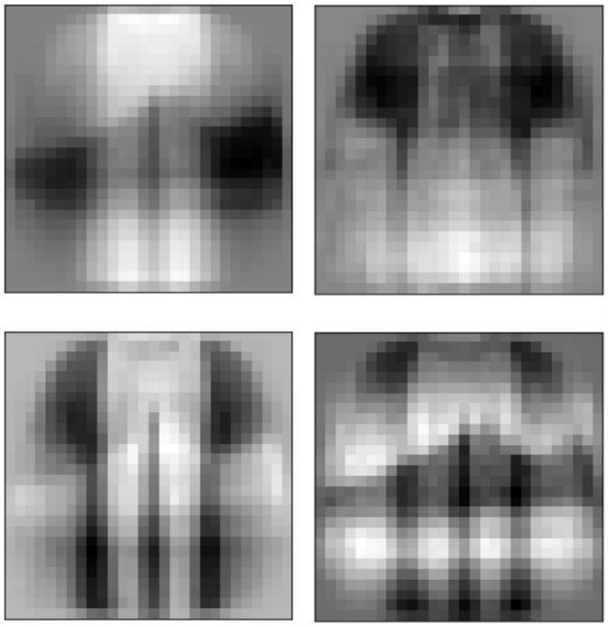


(b).

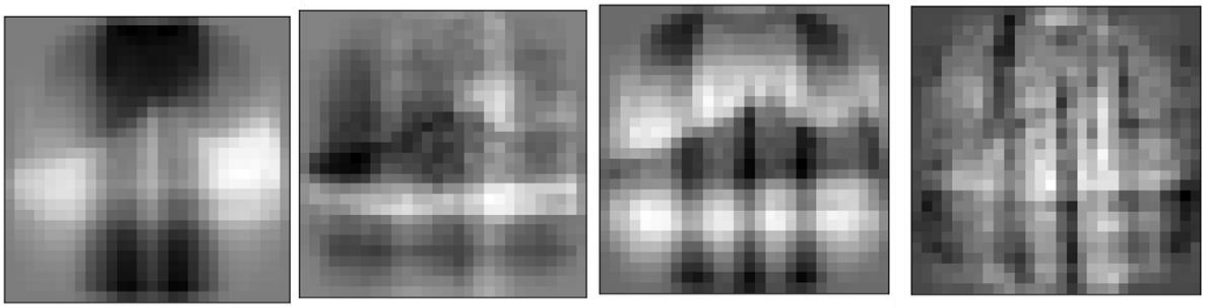
dimension = 2

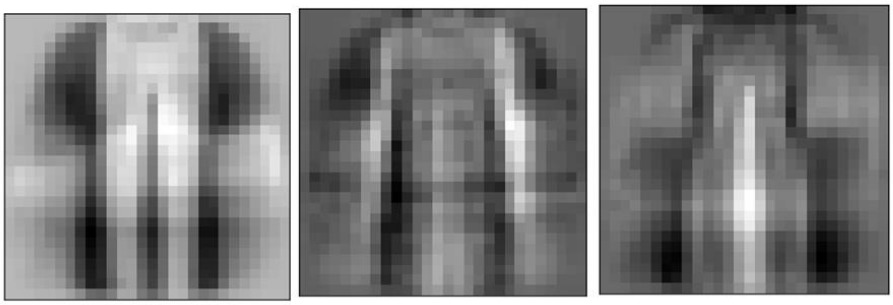


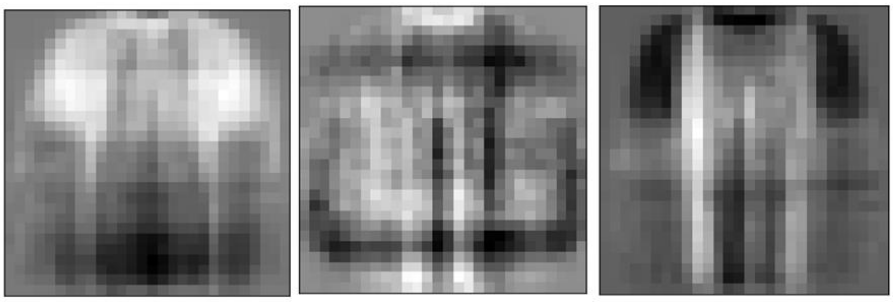
dimension = 5



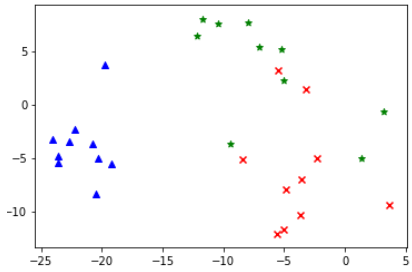
dimension = 10





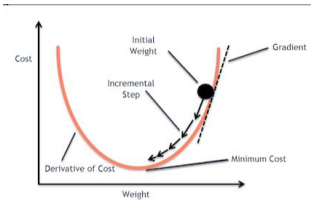


3.



4.

gradient descent(梯度下降法)主要是透過對 loss function 偏微分等方法，找出最佳的參數，找到它的最小值常用的方法，作法即是將在一個 epoch 中將所有 data 掃過一遍，更新 weight 後，算一次 loss function，在進行下一次 epoch，其有兩大缺點即：(1)當 dataset 很大時，要花很久的時間才能掃完且更新一次，需要較多時間才能收斂；(2)若不小心進入鞍點，或 global min，會無法繼續更新。



故引生出 Stochastic Gradient Descent（SGD），每一次 iteration 計算 mini-batch 的梯度，然後對參數進行更新，也就是每次會從訓練集中隨機取得 mini batch 張影像進行訓練，藉此不僅可避免記憶體的不足，更重要的是可以藉由這種隨機性跳脫局部最小值(local minimum)，這種隨機 mini batch 的 GD 被稱作 Stochastic Gradient Descent (SGD) ，缺點在於選擇合適的 learning rate 比較困難，對所有的參數更新使用同樣的 learning rate。對於稀疏數據或者特徵，有時對於不經常出現的特徵可能想更新快一些，對於常出現的特徵更新慢一些，這時候 SGD 就不太能滿足要求。

由第一題結果圖中可發現，GD 準確率高，SGD 跟 mini-SGD(batch size 32)比起來，mini-SGD(batch size 32)在 accurancy 較 SGD 來的快達到準確，原因我猜想是隨機丟取 32 進 training 並更新，一次量不大又不會太少，達到更新的目的，也不會較耗時，最後的準確率也跟一次丟全部(GD)做訓練雷同，符合預期，雖然時間這部分並未顯示出來，然在 DATASET 較大的時候我相信會反映出現。

比較有疑問的是在 SGD 以及 mini-SGD(batch size 32)的 learning curve 不是很合理，雖然曲線變化合理，然後數值很有疑慮，這部分我試過很多方式都是這樣，不知道是否為程式哪裡寫錯。這部份想請教，不知道是否能公布正確版本做參考，這樣日後做機器學習等訓練、project 可以較上手…，或是做個步驟教學之類的影片…QQQ

第二題加入 PCA (Principal Component Analysis, PCA)，希望達到 dimension reduction，通常預測/分類能力通常是隨著維度數(變數)增加而上升，然當模型樣本數沒有繼續增加的情況下，預測/分類能力增加到一定程度之後，預測/分類能力會隨著維度的繼續增加而減小。PCA 的核心在於：將原始數據拆解成更具代表性的主成分，並以其作為新的基準，重新描述數據。

由第二題(a)結果圖中可觀察到，PCA 維度越高，其 training accurancy 越高，雖然結果中 testing 走向不是很合理，一樣試過很多方式都未能達到預期，如上希望能得到解答…

然透過(b)確實可發現 PCA 維度越高的圖片還原率較高，反映出與預期更加貼近。