Assignment A

by Ruining Zhang

The report of assignment A

Ruining zhang

University of Sheffield

Registration No:150212283

Abstract: **Competitive learning is one of the important part of unsupervised learning in artificially neural network, which has been widely used in many fields. In this report, two main components will be discussed, there are principal component analysis for digital numbers clusters and competitive learning for digital handwriting recognition. In addition, questions in the assignment sheet will be answered as well.**

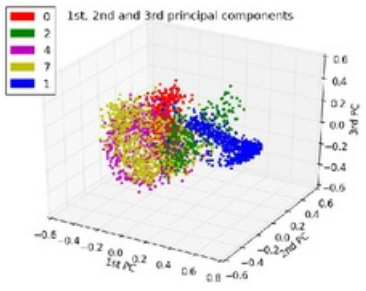
1.PCA

1.1 Brief introduction of the algorithm

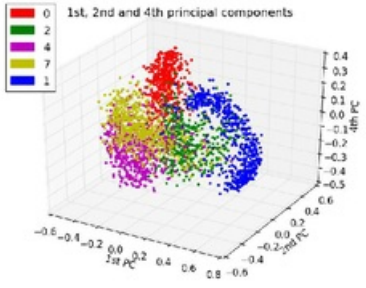
- Step 1: Reading the train data and their labels via the code that shows in the line 14 -16.
- Step 2: Data pre-processing contains Normalization and Mean subtraction. In general, each principal component is dominated by a single variable if data was not normalized. Moreover, in order to maximize the covariance of variables, data should subtract its mean along each dimension.The corresponding code is from line 18 to 31 .
- Step 3 : After the pre-processing proceed, building a covariance matrix only need one sentence by Python. Just like in the line 31.
- Step 4: Extracting eigenvalues and eigenvectors from the covariance matrix and sorting them in descending order of their value, which refers to the code line 36-44.
- Step 5: With respect to requirements, vectors will be applied to create new data set within the top sixth ranks. The range of related code is from line 46 to 50.
- Step 6: Finding indexes of data for each digit. Different colours as labels represent different clusters, then plot the significance components according to demands.

1.2 Plot the outcome

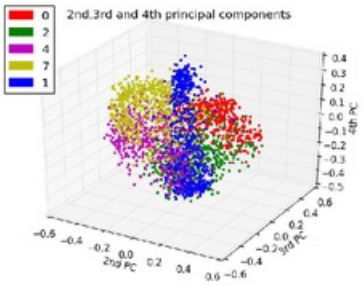
It is convenient to identify the clusters with labels, but will be a little hard without labels. The figure 1 (a) (b) (c) illustrate the first condition and figure 2 (a) (b) (c) indicates the later.



(a) 1 st, 2 nd and 3 rd PC

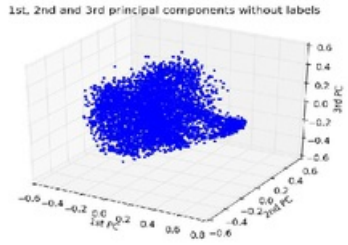


(b) 1 st, 2 nd and 4 th PC

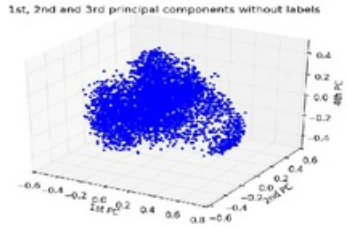


(c) 2 nd, 3 rd and 4 th PC

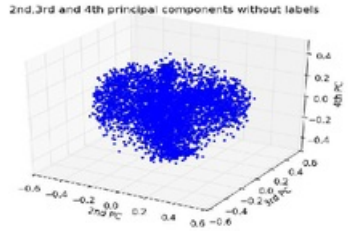
Figure 1: (a) (b) (c) show distributions of data in different principal components



(a) 1 st, 2 nd and 3 rd PC without labels



(b) 1 st, 2 nd and 4 th PC without labels



(c) 2 nd, 3 rd and 4 th PC without labels

Figure 2: (a) (b) (c) show distributions of data in different principal components without labels

Based on figures without labels, only three clusters could be distinguished. However, there are five in each figure. That is the one of the main reason why labels are necessary.

2.competitive learning

2.1 Brief introduction of the algorithm

- Step 1 : Loading train data and initializing serial constants, such as weights, learning rate, running time and numbers of digits and so on. Corresponding code from line 9-21.
- Step 2: Normalizing data and weights, it shows from line 23-34.
- Step 3: Competitive learning rule are implemented by python code(line 44-65). One layer neuron network is illustrated in figure 4.

$$W_i = (w_{i1}, \ldots w_{id})^T \quad (1)$$

$$X^N = (x_{n1}, \ldots, x_{nd})^T \quad (2)$$

Equation (1),(2) are represent for the vector of weight and input data respectively.

The k th weight will be the winner when the equation(3) is satisfied for all i.

$$\vec{W}_k \; \vec{X}^u \geq \vec{W}_i^T \; \vec{X}^u \quad (3)$$

In addition, winning weights would be updated after each iteration by the statement (4) (5).

$$\Delta w_k = \eta(x^u - w_k) \quad \text{on line rule (4)}$$

$$w_k^{new} = w_k^{old} + \Delta w_k \quad (5)$$

- Step 4: Adding noise and bias. There are few units with weights

vector far from any input vector may never win. These units are called “dead units” (shows in figure 5). In order to solve this issue, noise and negative bias are add in the standard competitive learning.The corresponding code of bias and noise are at 65 and 51-53 respectively.

- Step 5:Plotting a series of figures,for example, input images and prototype images, winning times of neurons and change of weights, to display states of winning process. The range of related code is from 68-117.

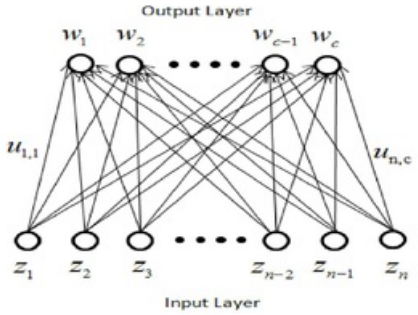


Figure3: The model of one layer neuron network.(Xuebin Qin et.al 2014)

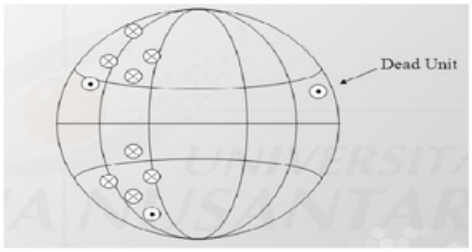


Figure 4: The locations of dead units, which comparing with input vectors. The cross cycle represents input vector and the cycle with black solid points are weights,one of them is dead unit.(http://slideplayer.info/slide/5015922/)

2.1 Significance of normalization of weights and data

Normalization of data and weights are very crucial. On the one hand, each data point may contains enormous components (features) which have different units. For instance, the height and the wight of a man. We just want to focus on their values instead of units. Thus, normalizing data is essential. On the other hand, more features could be involved in the PCA though the normalization. Without this process, it may not be avoided that the outcome of trials are dominated by only one or few components and influence the result of PCA.

Similarly, normalizing weights could give equal chance for every weights to win. Properly one weight will win forever if them are not normalized.

2.3 Optimization of network

According to Li’s research in 2006, dead units will never win and result in the different winning opportunity for all nodes. Many solutions for dead units were proposed in time, for example: Initialization for weights to data samples, updating loser with small learning rate, adding noise or negative bias to firing frequency and applying a conscience mechanism, and so forth(Li et al 2006).

The method of bias and noise are combined together to applied in this case. The computation of output of nodes will be modified during competition, which aims to make the competition easier for frequent losers and more difficult for frequent winners.(Wang 1994) A bias equation is shown below:

$$b_k = \gamma (1/n - x^u) \quad (6)$$

Where γ is constant, which equal to 0.001 in the program. n refers to total number of inputs and x^u is the

frequency of winning for input N_k. With reference to equation(4) and (5), the overall output is set to be:

$$h_k = \vec{w}_k \cdot \vec{x}^u + b_k + x_i \quad (7)$$

x_i is the random noise in here.

With referring to the statistic data collected by Gang in 2016, which attaching in the appendix. Dead units are considered as neurons that under 31 firing times.

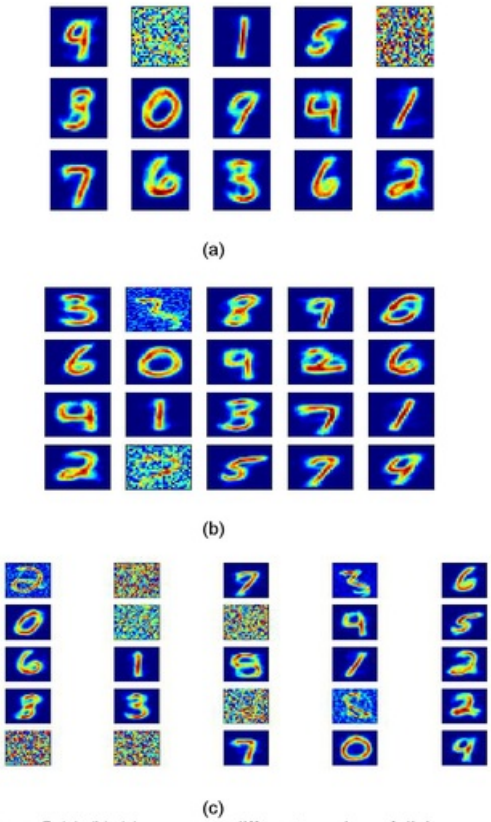
The algorithm was implemented to test the improvement of dead units, the digit data consist of three groups: 15, 20 and 25. Trials of each group repeated 5 times under two conditions: standard program; improved program. Then, outputting the number of dead units within each iteration. The outcome is illustrated in table 1(a) (b) and figure 6 (a) (b) (c) shows images of prototypes below, which be plotted under classical competitive learning rule.

TABLE 1
(a) The number of dead units in improved competitive learning condition (b) The number of dead units in standard competitive learning condition.

Dead units / Times / Digits	1	2	4	5	Average
15	0	0	1	0	0.6
20	3	2	2	2	2.2
25	5	5	2	5	3.8

(a)

Dead units / Times / Digits	1	2	3	4	5	Average
15	2	1	1	3	2	1.8
20	6	5	7	5	5	5.6
25	9	8	10	10	9	9.2



(a) (b) (c)

Figure 5 (a) (b) (c) represent different number of digits, 15, 20 and 25 respectively. Some figures above can not be distinguished are identified dead units, i.e, the second and firth small figures in the first row in the figure (a).

2.4 Average weight change

Figure 6 indicates a curve of the average change of weight on semi - log x axes.

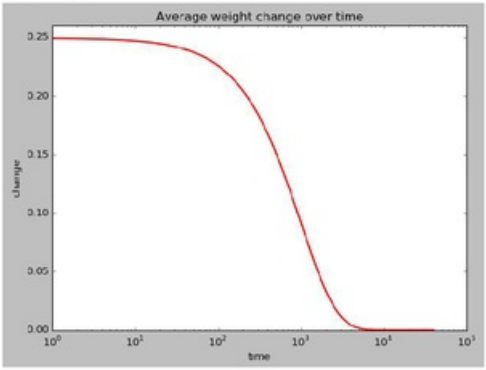


Figure 6 The average weight change over time

From the above figure, the outcome can be conclude: the network has not been

sufficiently learned until approximate 400 times.

2.5 Figure of prototypes

In this case, fifteen trained digits will be plotted(shown in figure 7), each small figure suggests the digit that corresponding to the winner.

Figure 7: Figures of trained prototype

Ten prototypes were found from the network and by comparing to the original labels, there are all correct.

2.6 Correlation matrix

A correlation matrix is used to investigate the dependence between multiple variables(matrix) at the same time. The outcome is a table concluding the correlation coefficients between each variable(matrix) and the others. Here, variables are matrix of prototypes and we are going to explore the dependence between them, which will result in reduction of clusters. The equation of correlation matrix is in below.

$$\rho_{X,Y} = corr(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \quad (8)$$

Where $\rho_{X,Y}$ is the population of correlation coefficient, and $\mathrm{cov}(XY)$ is the covariance matrix between variable X and Y . σ_X, σ_Y mean standard deviations. Moreover, the figure of correlation (figure 8) is symmetrical along the diagonal axis since $corr(X,Y) = \mathrm{cov}(Y,X)$.

As mentioned, correlation matrix are applied to categorize digits, and the relation between two variables can be divided into these parts based on its coefficients.

$0 < \rho \le 1$ positive linear relationship

$\rho = 0$ weak linear relationship

$-1 \le \rho < 0$ negative linear relationship

The larger ρ means higher probability for these two variables in one cluster

The top of colour bar represents strongest relation between two prototype and its magnitude decreases along the vertical axes, therefore blue area refers to weakest relationship.

With respect to this theory, if the coefficient is greater than 0.7, the two variables could be combined as one cluster. For instance, the first prototype and the second one. In addition, the result in 2.5 is verified, and they are in same group, digit sixth.

3. Appendix

3.1 Reference

Li, Z. and Ronald Eastman, J., 2006. The Nature and Classification of Unlabelled Neurons in the Use of Kohonen's Self-Organizing Map for Supervised Classification. *Transactions in GIS*, 10(4), pp.599-613.

Qin, X., Wang, M., Lin, J.S. and Li, X., 2014. Power cable fault recognition based on an annealed chaotic competitive learning network. *Algorithms*, 7(4), pp.492-509.

Rumelhart, D.E., McClelland, J.L. and PDP Research Group, 1988. *Parallel distributed processing* (Vol. 1, pp. 443-453). IEEE.

Wang, R., 1996. A network model for the optic flow computation of the MST neurons. *Neural Networks,* 9(3), pp.411-426.

Assignment A

GRADEMARK REPORT

FINAL GRADE

68/0

GENERAL COMMENTS

Instructor

PCA description should be comprehensive.
i.e. provide information about the implications of each step, rather than just listing them.
e.g. what is an eigenvalue/principle component intuitively.

PCA discussion should include automatic methods for clustering (e.g. k-means clustering)

Good description of normalising weights and data, Normalisation should also include method.

Optimisation good, though it would be improved by optimising other critical parameters such as learning rate which are present in both methods, imporved and not-improved, and may have a significant effect if modified.
Also with several runs it would be sufficient to show the mean and standard deviation (and even a wisker plot if there are outliers)

Missing reference to Gang (2016) in references, without this it is not clear as to why neurons firing under 31 times are considered dead. It is more typical to set a relative threshold as some low percentage of firings, as even dead neurons can fire occasionally given a sufficiently noisey inputs.

Correlation matrix is well described, but figure 8 appears to be missing.

Presentation is good.

Extras: bias and noise, and additional useful plots and explanations

Appendix exists but is empty, code snippets are missing.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7