

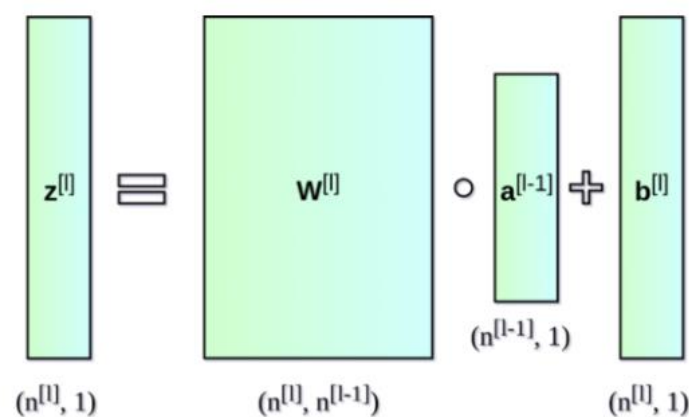
Deep Learning HW1

309512010 陳紀愷

1. Regression

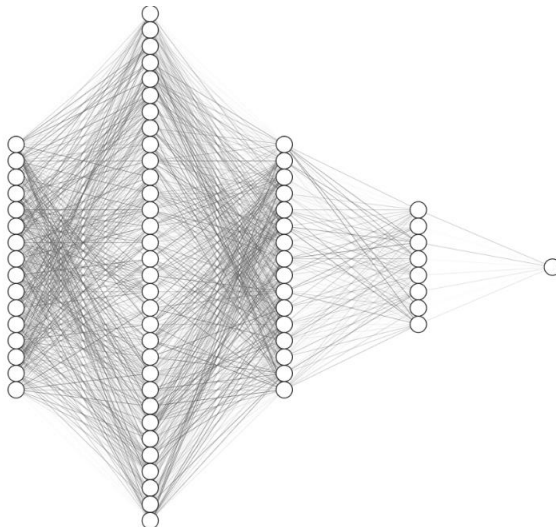
(a) (b) Please try to predict the heating load of buildings by minimizing the sum-of-squares error function.

$$\mathbf{z}^{[l]} = \mathbf{W}^{[l]} \cdot \mathbf{a}^{[l-1]} + \mathbf{b}^{[l]} \quad \mathbf{a}^{[l]} = g^{[l]}(\mathbf{z}^{[l]})$$



除了最基本的正向傳播，我們還要加上 Backpropagation 和 SGD 來搭建我們的神經網路，因此我們需要知道不同的 activation Function 和 Loss Function 以及他們的微分，像是 Relu 的微分是 Step Function，MSE 的微分是 $2 * (\hat{Y} - Y)$ 等等，有了這些之後我們便可以透過不斷的計算出 Gradient 來更新網路架構。最後的成果和資料如下：

(1) network architecture (number of hidden layers and neurons)



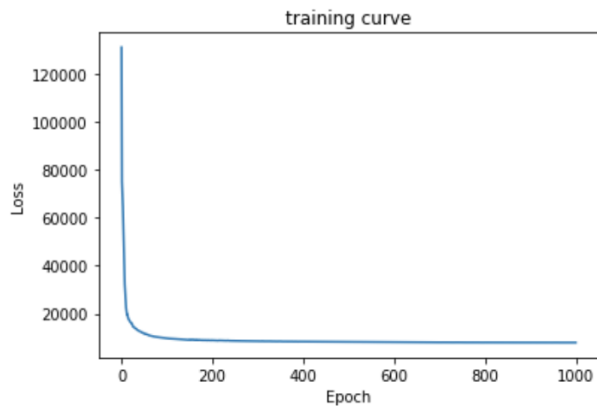
主要分為3層hidden layers
最左邊是Input最右邊是Output
Neuron數從左到右是
(包含最左最右)

[16 – 32 – 16 – 8 – 1]

中間的Activation Function則是
[Relu , Relu , Relu , linear]

p.s. 由於是回歸問題所以最後一層要”linear”

(2) learning curve



$$\text{MSE} = 8138.652276476238$$

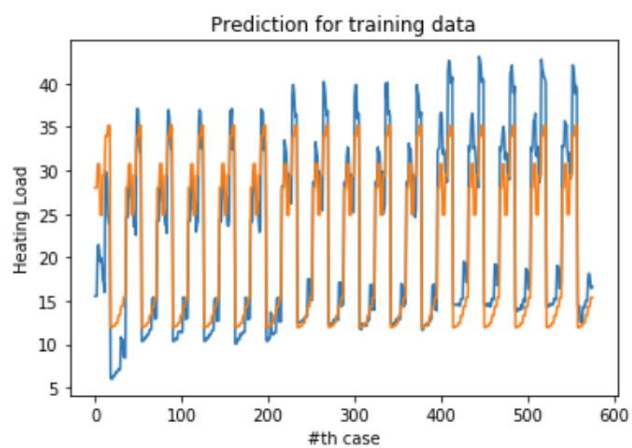
(3) training RMS error

Training RMS error = 3.758

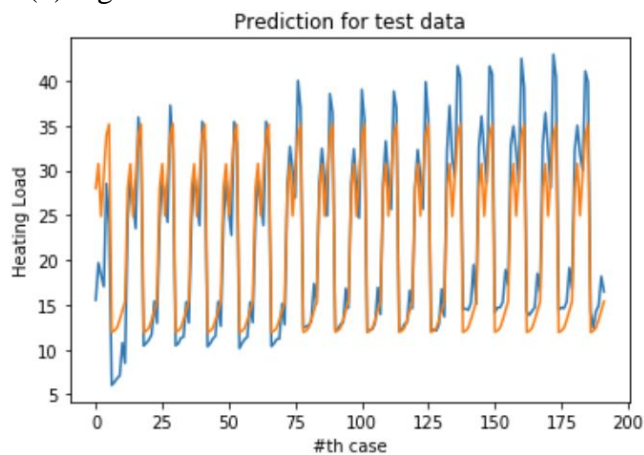
(4) test RMS error

Test RMS error = 3.782

(5) regression result with training labels



(6) regression result with test labels

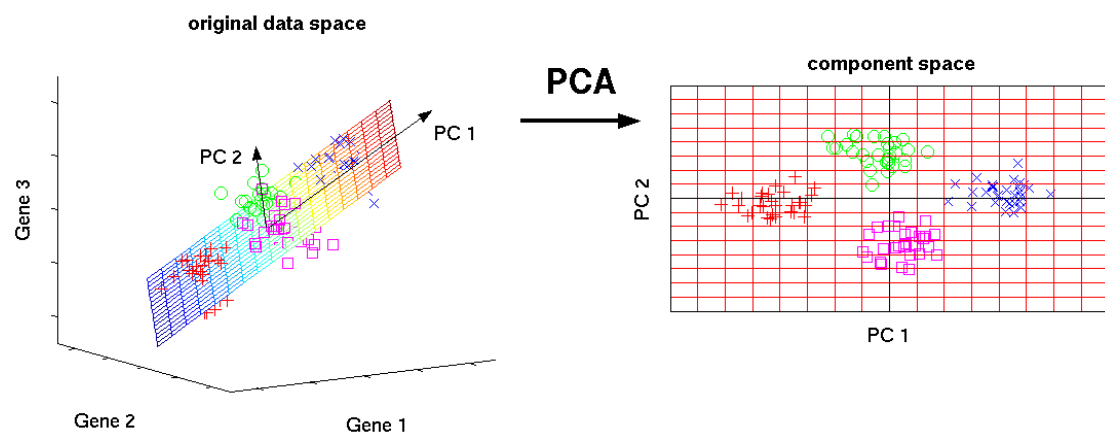


(c) Design a feature selection procedure to find out which input features

influence the energy load significantly and explain why it works. You may compare the performance of choosing different features.

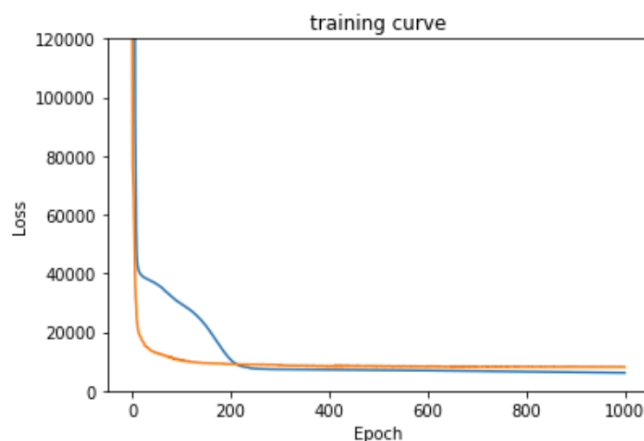
feature selection procedure 我刻了一個 PCA(Principal Component Analysis)的程式來提取特徵，PCA 是可以透過矩陣運算把高維度的資料降成(投影)成低維度，並讓投影後的資料變異最大，也就是可以用低維度來顯示出高維度資料的主成分(提取特徵)。

我接下來分別把原本 16 維度的資料用 PCA 降成 12 維和 8 維，並比較他們的 learning curve和RMS error



PCA , n = 12 , Training RMS error = 3.2619

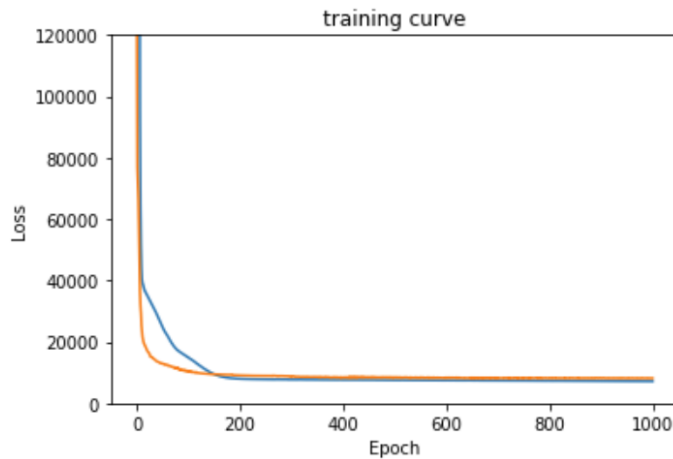
Test RMS error = 3.237



左圖中橘色的線是沒有經過特徵提取的學習曲線，藍色的是 PCA(n=12)後的結果，可以發現雖然中間(0~200 Epoch)時雖然提取過的表現比較差，但在之後的Loss都明顯的比原來的還要好。而RMS也比原本的低大約0.5左右。

PCA , n = 8 , Training RMS error = 3. 527

Test RMS error = 3. 5165



左圖中橘色的線是沒有經過特徵提取的學習曲線，藍色的是PCA(n=8)後的結果，可以發現雖然中間(0~200 Epoch)時雖然提取過的表現比較差，但在之後的Loss都明顯的比原來的還要好。而RMS也比原本的低大約0.3左右。

和 n = 12 相比可以發現這個的效果雖然比較差，但還是比原本沒有經過壓縮特徵提取的訓練結果還要好，可以發現特徵提取對於訓練來說是一個不錯的預處理方式。

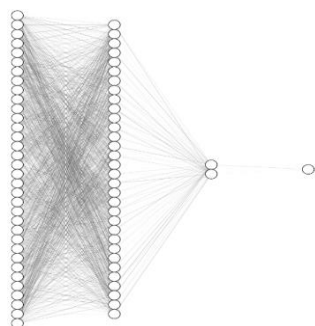
2. Classification

(a) (b) Please try to classify the Ionosphere data by minimizing the cross-entropy error function.

$$E(w) = - \sum_{n=1}^N \sum_{k=1}^K t_{nk} \log y_k(X_n, w)$$

這題和第一題不一樣的地方除了損失函數不同以外，還有在於它是分類問題，所以我們必須把最後一層的activation Function改成sigmoid，使他的輸出最後會落在0~1之間，並以0.5作為分隔，代表預測後的種類結果。

(1) network architecture



主要有2層hidden layers

最左邊是Input Layer最右邊是Output Layer

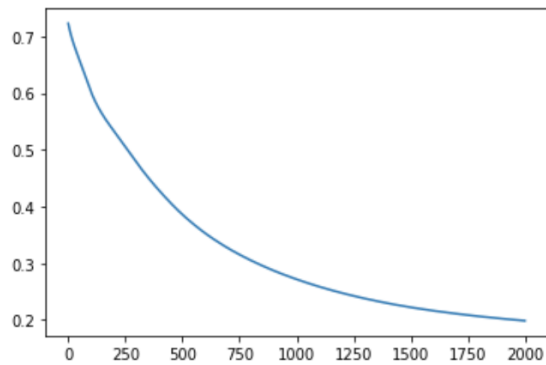
Neuron數從左到右是(包含最左最右)

[34 - 32 - 2 - 1]

中間的Activation Function則是

[Relu , Relu , sigmoid]

(2) learning curve



(3) training error rate = 0.0301 \doteq 3%

(4) test error rate = 0.0986 \doteq 9.8%

(c) Compare the results of choosing different numbers of nodes in the layer before the output layer by plotting the distribution of latent features at different training stage.

