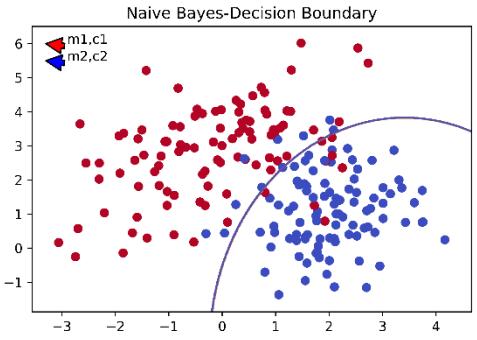
University of Southampton

Comp6229 – 2017/18 Machine Learning Assignment

Junming Zhang **29299527** jz1g17@ecs.soton.ac.uk

1. Neural Network Approximation



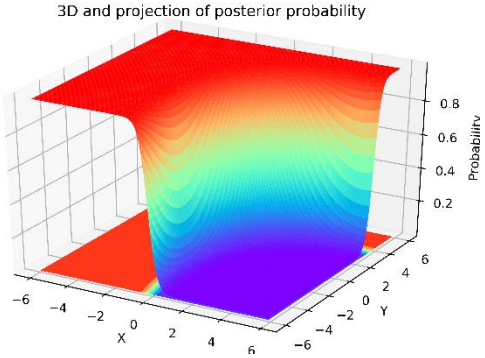


Figure 1. Decision boundary of 2 classes *Figures 2. Posterior probability distribution in 3D graph*

The first step generates 100 samples with two features which are $m_1 = \begin{pmatrix}0\\3\end{pmatrix}, C_1 = \begin{pmatrix}2 & 1\\1 & 2\end{pmatrix}$ and $m_2 = \begin{pmatrix}2\\1\end{pmatrix}, C_2 = \begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$. At the same time, label output of different features 0 and 1.

Posterior probability $p(\omega_i|x)$ is proportional to $p(x|\omega_i)p(\omega_i)$. To find the decision boundary, apply discriminant function in $g_i(x) = p(x|\omega_i)p(\omega_i)$. Samples are generated by Gaussian distribution, therefore apply natural logarithm to solve the equation which is $g_i(x) = \ln p(x|\omega_i) + \ln p(\omega_i)$. Therefore, the discriminant function can be represented as $g_i(x) = -\frac{1}{2}(x - \mu_i)^T * \Sigma_i^{-1}(x - \mu_i) - \ln 2\pi - \frac{1}{2}\ln|\Sigma_i| + \ln p(\omega_i)$ and the decision boundary would be the solution of $g1(x) = g2(x)$. In this case,

$$g1 = -0.5 * (x - m_1')' * inv(C_1) * (x - m_1) - log(2 * pi) - 0.5 * log(det(C_1))$$

$$g2 = -0.5 * (x - m_2')' * inv(C_2) * (x - m_2) - log(2 * pi) - 0.5 * log(det(C_2))$$

Therefore, the decision boundary of two class can be plot as Figure 1.

Bayes classifier in two class problem is $P[\omega_1|x] = \frac{p(x|\omega_1)P[\omega_1]}{p(x|\omega_1)P[\omega_1] + p(x|\omega_2)P[\omega_2]}$. A simple expression can be obtained by eliminate molecule which is $P[\omega_1|x] = \frac{1}{1 + \frac{p(x|\omega_2)P[\omega_2]}{p(x|\omega_1)P[\omega_1]}}$ and the probability density function $p(x|\omega_i)$ is $\frac{1}{\sqrt{2\pi^i detC_i}}\exp(-\frac{1}{2}(x - m_i)^T * C_i^{-1}(x - m_i))$. Therefore, the posterior probability of each sample can be computed and plotted as Figure 2. It could find the changing tendency is corresponding to the distribution of those samples.

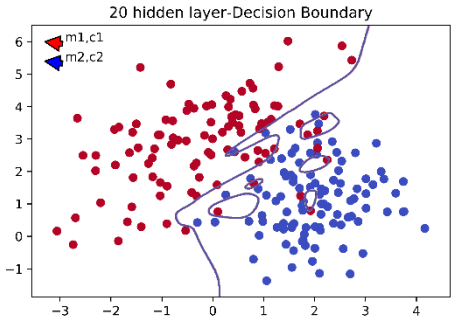


Figure 3. neural network with 20 hidden layer

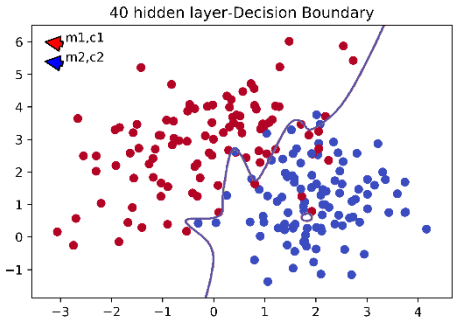


Figure 4. neural network with 40 hidden layer

The activation function of the hidden layer is $\tanh$ because the derivate function be computed by the original function value which is $1-\tanh^2 x$. The porpose of the neural network is update parameters by finding those parameters minimize the error.

The prediction is based on forward propagation:

$z_1 = xW_1 + b_1,\ a_1 = \tanh(z_1),\ z_2 = a_1W_2 + b_2,\ a_2 = e^{z_2} = \hat{y}.$

$\hat{y}$ is the predicted output which defines as negative log likelihood.

The loss for prediction is defined as $L(y,\hat{y}) = -\frac{1}{2}\sum_{n\in\text{example number}}\sum_{i\in 2} y_{n,i} log\hat{y}_{n,i}.$

The derivate function calculated by back propagation:

$\text{delta3} = \hat{y} - y, \text{dW}_2 = a_1^T delta3,\ \text{db}_2 = a_1^T delta3,\ \text{dW}_1 = x^T delta2,\ \text{db}_{12} = delta2.$

Then the parameters updated by decreasing with the learning rate for gradient descent.

Finally, the learned parameters will use to predict the class and drew the decision boundary which is shown in Figure 3 and 4.

Decision boundary by	Bayes's optimal	Neural network 20 hidden layers	40 hidden layers
Accuracy	0.91	0.965	0.93

Table 1. accuracy of Bayes's and neural network

It could find the decision boundary of Bayes's optimal is a smooth curve, the calculated speed is faster than the neural network, however, the accuracy of it is lower than neural network. The decision boundary of the neural network is an irregular curve. The calculated speed is increasing with higher hidden layers because more parameters are needed to calculate. However the accuracy of it is not linear, it related to the choice of training data, hidden layers, learning rate and regularization strength. In this case, 20 hidden layers have higher accuracy than 40 hidden layers.

2. Mackey-Glass Time Series

Mackey-Glass time series refers to delayed differential equation $\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x(t-\tau)^{10}} - bx(t)$ and the data used in this part is the solve of 4th order Runge-Kutta at discrete, equally spaced time steps. 2000 samples are generated and the Parameters choice is shown in Table 2.

a	b	tau	x0	deltat	sample_n	Interval
0.2	0.1	0.965	1.2	1	2000	1

Table 2. parameters to generate Mackey-Glass Time Series

The first 1500 samples will be used as training data and the remain 500 samples will be used as test data. Using 20 samples to predict next sample, the design matrix of training data will be:

$$X = \begin{pmatrix} Y_1 = [sample_1 & sample_2 & \dots & sample_{19} & sample_{20}] \\ Y_2 = [sample_2 & sample_3 & \dots & sample_{20} & sample_{21}] \\ & & \vdots & & \\ Y_{1480} = [sample_{1480} & sample_{1481} & \dots & sample_{1498} & sample_{1499}] \\ Y_{1481} = [sample_{1481} & sample_{1482} & \dots & sample_{1499} & sample_{1500}] \end{pmatrix}$$

2.1 Linear Prediction of Mackey-Glass Time Series

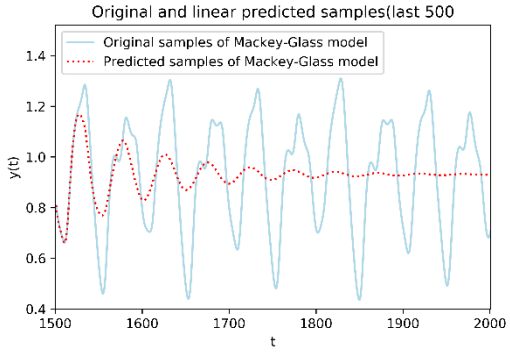


Figure 5. Mackey-Glass data with linear prediction

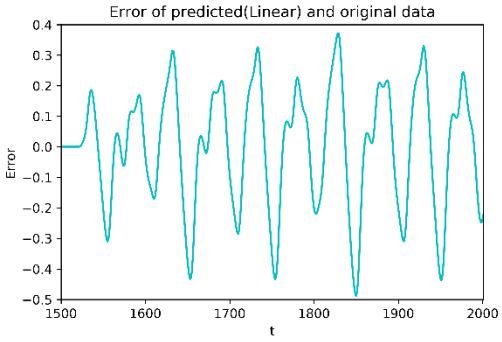


Figure 6. Error of linear prediction

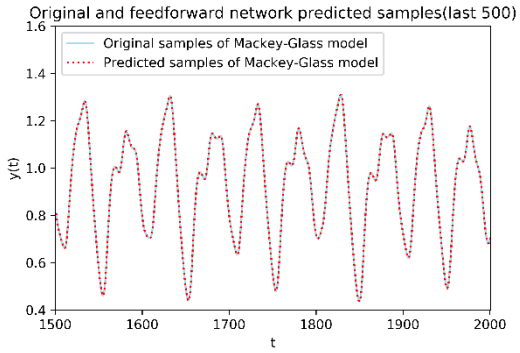
The blue line in Figure 5 is the line chart of last 500 generated samples. For linear prediction, the relationship between output y and design matrix X is $y = X\beta + \Sigma$. Therefore, the optimal coefficients β can be calculated by least squares method which is: $\beta = (X^T X)^{-1} X^T y$.

Because $y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{1480} \\ y_{1481} \end{pmatrix}$ and $X = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{1480} \\ Y_{1481} \end{pmatrix}$, therefore, $\beta = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{19} \\ \beta_{20} \end{pmatrix}$. Then the time series can be predicted by using 20 samples multiply the coefficients β. The predicted result is shown as

the red dashed line in the Figure 5. It could find this model could predict short-term time series with low error which is shown in Figure 6. However, the predicted data start convergence after prediction of 150 data. The mean-square error is used to observe the fitting level:

$$\text{MSE(Linear prediction)} = \frac{1}{n}\sum_{i=1}^{n}(\overline{X_i} - X_i)^2 = 0.0384$$

2.2 Feedforward Neural Network Prediction of Mackey-Glass Time Series



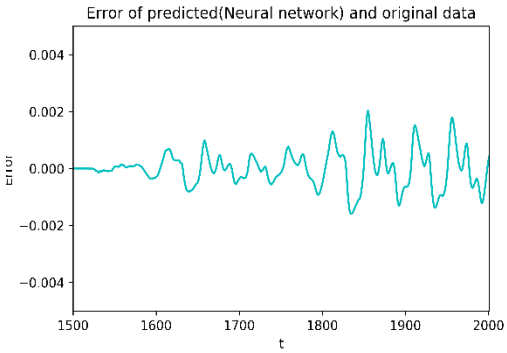


Figure 7. Mackey-Glass data with neural network

Figure 8. Error of neural network prediction

For a neural network, using feedforward network to train design matrix X and Mackey-Glass time series data. Set the hidden layer to 20 and start training until the minimum gradient reached. To predict the time series: $\mathrm{p}(1501) = \mathrm{net}(\mathrm{input}(1481{:}\,1500)$, and with the new predicted $\mathrm{p}(1501)$, the next point can be predicted by $\mathrm{p}(1502) = \mathrm{net}(\mathrm{input}(1482{:}\,1501)$. Figure 7 is the line chart of predicted time series in 1500 to 2000. It could find the predicted data is fitting the original data and the error is smaller than linear prediction which is shown in Figure 8. The mean-square error is $\mathrm{MSE(Neural\ network)} = \frac{1}{n}\sum_{i=1}^{n}(\overline{X_i} - X_i)^2 = 4.459\times10^{-5}$.

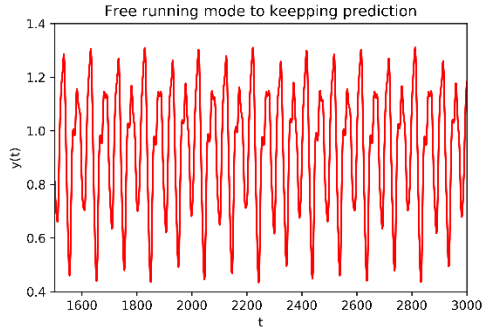


Figure 9. network prediction in next 1500 time series

It could find the predicted time series is still oscillation. Because the Mackey-Glass time series is sinusoidal like and the interval of $\mathrm{y(t)}$ is small, therefore it is hard to convergence.

3. Financial Time Series

Using Python Pandas library to process download FTSE100 data. First, extract the data in close and vol columns, then delete the first row. For data in vol, it uses M and B represent numbers, thus, M and B needed to change with 10^6 and 10^9.

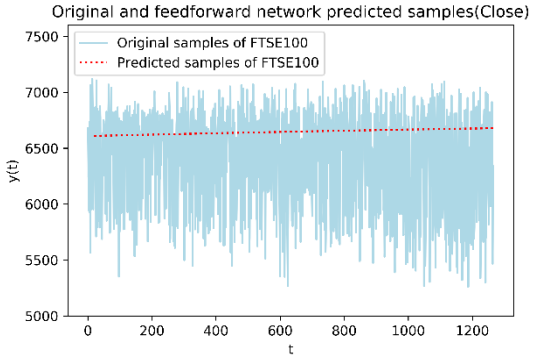


Figure 10. FTSE data(close) with neural network

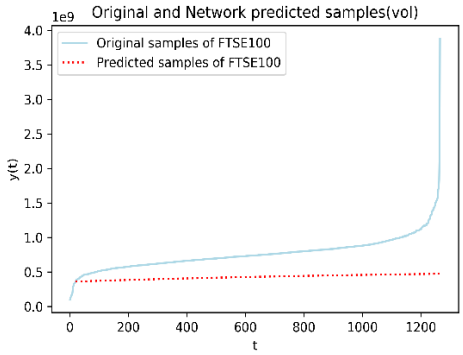


Figure 11. FTSE data(vol) with neural network

For predicting FTSE index value of next day, the neural network construction is similar to Part 2.2. Calculate design matrix first:

$$X(Close) = \begin{pmatrix} Y_1 = [sample_1 & sample_2 & \dots & sample_{19} & sample_{20}] \\ Y_2 = [sample_2 & sample_3 & \dots & sample_{20} & sample_{21}] \\ & & \vdots & & \\ Y_{1245} = [sample_{1245} & sample_{1246} & \dots & sample_{1263} & sample_{1264}] \\ Y_{1246} = [sample_{1246} & sample_{1247} & \dots & sample_{1264} & sample_{1265}] \end{pmatrix}$$

$$X(Vol) = \begin{pmatrix} Y_1 = [sample_1 & sample_2 & \dots & sample_{19} & sample_{20}] \\ Y_2 = [sample_2 & sample_3 & \dots & sample_{20} & sample_{21}] \\ & & \vdots & & \\ Y_{1245} = [sample_{1245} & sample_{1246} & \dots & sample_{1263} & sample_{1264}] \\ Y_{1246} = [sample_{1246} & sample_{1247} & \dots & sample_{1264} & sample_{1265}] \end{pmatrix}$$

And then define the output data: $\mathrm{y(Close)} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{1245} \\ y_{1246} \end{pmatrix}$ and $\mathrm{y(Vol)} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{1245} \\ y_{1246} \end{pmatrix}$.

Next is using feedforward network with 20 hidden layers and 'trainbr' function. Then the predicted data can be obtained by forwarding 20 samples and itself could use as the input samples to predict next value.

The predicted value of close price and volume are shown in Figure 10 and 11. It could find the predicted data is convergent to a small area. The reason for it could be some mistakes in network construction, the value of a trained dataset is huge, therefore it may need normalization first.

It is hard to use this model to make money because the stock changing with lots of factors, 1 feature cannot predict the accurate value. Moreover, the market discipline may change suddenly, the tendency of history data is hard to predict the sudden change. However, short-term changing tendency could be predicted with a higher accuracy. People could invest those stock with stable history data and increasing tendency.

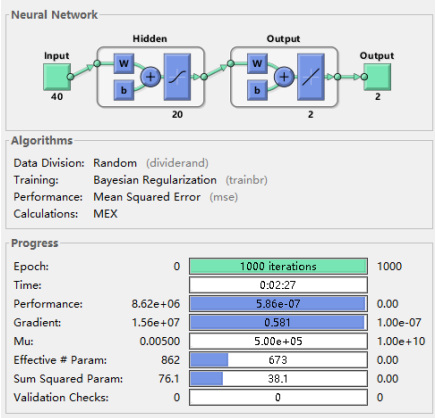


Figure 12. Neural network for FTSE data(close and vol)

With additional input volume traded, combine close price and volume traded first and following the steps, design matrix and network construction. In this case, 40 inputs will be used to calculate the output which is shown in Figure 12. However, the training result is not well.