EEE 511: Prediction of happiness index through linear regression model and multilayer perceptron

Jian Meng
ECEE Department
Arizona State University
jmeng15@asu.edu

Manvitha Pannala
ECEE Department
Arizona State University
mpannal1@asu.edu

Siyu Xiang
ECEE Department
Arizona State University
sxiang8@asu.edu

***Abstract*—In this report, we first collected the happiness index of each country and the scores of various variables related to the happiness index (including family, health, government, generosity, ...). Obviously, people's happiness index is directly related to the score of the variable, so we try to construct a linear regression model and a multilayer perceptron to predict the happiness index. The RMSE corresponding to the two models is the same value.**

I. Introduction

The happiness of citizens in various countries is directly related to the following indicators according to statistics: Economy (GDP per Capita), Family, Health (Life Expectancy), Freedom Trust (Government Corruption), Generosity, Dystopia Residual. If these factors are quantified, we can score the happiness of citizens in different countries. We found the statistical results of happiness scores in various countries in the World Happiness Report dataset from 2015 to 2019. [1] In this report, we explore how to train linear regression models and multilayer perceptrons based on existing data.

A. Multivariate linear regression

Univariate linear regression uses a major influencing factor as the independent variable to explain the change of the dependent variable. In realistic problems, the change of the dependent variable is often affected by several important factors, so it is necessary to use two or more factors as independent variables to explain the changes in the dependent variable. This is called multiple regression or multiple regression. When there is a linear relationship between multiple independent variables and dependent variables, the regression analysis performed is multiple regression. Let y be the dependent variables, X1, X2 ... Xk are independent variables, then the multiple linear regression model is:

$$Y = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + ... + \theta_k X_k + e$$

When establishing a multivariate linear regression model, in order to ensure that the regression model has excellent interpretation ability and predictive effect, we should first pay attention to the selection of independent variables. The guidelines are: (1) The independent variable must have a significant impact on the dependent variable and have a linear relationship; (2) The linear correlation between the independent and dependent variables must be true, not formal; (3) There should be some mutual exclusion between independent variables, that is, the correlation between independent variables should not be higher than the correlation between independent variables and dependent variables; (4) The independent variable should have complete statistical data, and its predicted value is easy to determine.

After the relationship between the independent variables and the dependent variable is determined, the parameters of the model need to be determined through the test group data. First we need to build a cost function:

$$J(\theta_0, \theta_1, ..., \theta_n) = \frac{\sum_{t=1}^{m} (h_\theta(X_i) - Y_i)^2}{2m}$$

where $h_\theta(X_i)$ is the hypothesis, Y_i is the given outputs corresponding to training data set. Update parameters of this model using gradient descent until convergence.

$$\theta_j := \theta_j - \eta \frac{\partial J(\theta_0, \theta_1, ..., \theta_n)}{\theta_J}$$

j=0,1,2,...,n, and η is a positive number, which is called the learning rate. After we getting the model parameters, we can use it to predict the happiness scores.

B. Multilayer perceptron

After the establishment of the linear regression model, we obtained the corresponding RMSE values. In the next part, we trained a multi-layer perceptron model (1 hidden layer) that can perform comparably well as the linear regression model measured by the RMSE.

An important feature of multilayer perceptrons is multiple layers. We call the first layer the input layer, the last layer the output layer, and the middle layer the hidden layer. MLP does not specify the number of hidden layers, so you can choose the appropriate number of hidden layers according to your needs. And there is no limit to the number of neurons in the output layer.

After the basic model is set up, we need to update the weights in the model through training. Due to the multi-layer network structure, it is impossible to use the loss to directly update the parameters of the hidden layer in the middle, but the back propagation of the loss from the top to the bottom can be used to estimate the parameters.

II. Experiments

Because we used data from 2015 to 2019, in order to make the training more accurate, we need to use this data comprehensively. So before sending the dataset into the model for training / test, the dataset was constructed by combining all separate csv files from 2015 to 2019. With the objective of fully predict the happiness score based on all the related features, for each file, the extracted features are listed as follow:

1) GDP
2) Family
3) Health
4) Freedom
5) Generosity
6) Corruption

The target output of the supervised learning is the happiness scores. The algorithms are implemented in PyTorch via Python, the code are available in GitHub: https://github.com/mengjian0502/eee511_team3_assignment/tree/master/assignment02. For details about the code execution, please read the README file under the assignment02 directory.

A. Linear regression model

In our report, We use 6 independent variables, so we need to get the values θ_0 to θ_6 through training, where θ_0 is the bias and θ_1 to θ_6 are the parameters corresponding to the independent variables listed above. More specifically, the linear regression model can be considered as a 1 linear combination layer with linear activation function. The weights are the multi-variate coefficients that are trainable through gradient descent.

In our experiments, we used a total of 782 samples, of which 80% of the samples were used for training and 20% of the samples were validation sets.The learning rate we chose is 0.1, and use gradient descent method to iterate the parameters above. The following figure illustrate the performance of training validation:

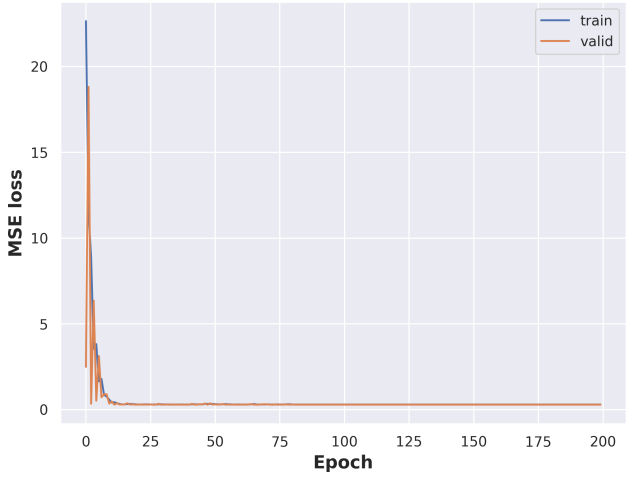

Fig. 1. Training & Validation of the linear regression model

In this model, we used the value of MSE to measure the rationality of the parameters of the linear model, the MSE formula is as follow:

$$MSE(\theta_0, \theta_1..\theta n) = \frac{\sum_{i=1}^{m}(J_i)}{m}$$

Where J_i is the cost of each prediction, in model training, we want to find a set of model parameters to minimize the average loss of the training samples. Final parameters (θ_0 to θ_5): [1.1216, 0.6970, 1.0923, 1.3819, 0.3625, 0.9328], c = 2.1702, and finally we got the result MSE = 0.2908.

B. Multi-layer perceptron model

1) Structure of the MLP: As we discussed before, the objective of the MLP is to predict the relationship between the input features and the target scores. Therefore, the dimensionality of the input should be 6 and the output is 1. Based on the empirical results, the number of neurons in the hidden layer has been set to 5.

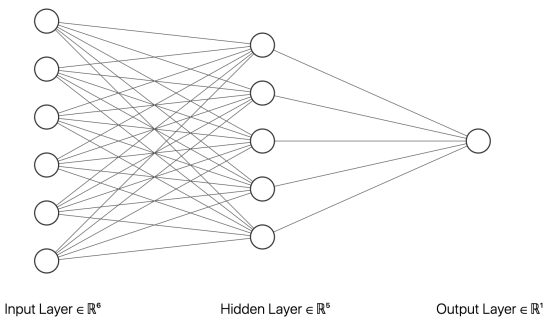

Fig. 2. Structure of the multi-layer perceptron

Since the basic scenario of the assignment can be classified as a regression problem, therefore we choose the sigmoid function as the activation function of the neural network except the output neuron.

2) Training & Validation: The total number of samples in the combined dataset is 782. We split the entire dataset into training set (80%) and the validation set (20%). We used SGD with momentum of 0.9 and learning rate starting from 0.1 and scaled by 0.1 at epoch 80, 120 to avoid the overshoot during the gradient descent. The mini-batch size of 128 is used, and the maximum number of epochs is 200. The best trained model has been saved. More specifically, the model has been switched to *evaluation* mode during the validation phase, therefore the model can be tested directly. The following figure illustrate the performance of training & validation:

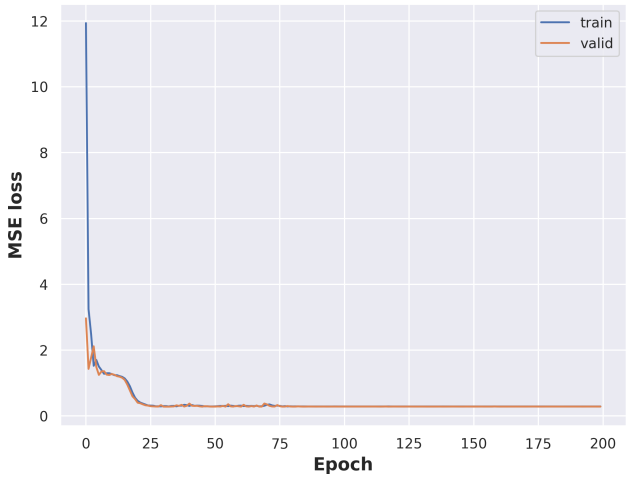

Fig. 3. Training & Validation of the multi-layer perceptron

Since the model has 6 input features, therefore it is hard to visualize the input/output relationship via a 6-D plot. In this report, we choose to use the MSE error as the measurement of the performance.

To evaluate the effect of the number of hidden neurons, we sweep the number of the hidden neurons from 1 to 10, the results has been summarized as follows:

TABLE I
MLP MODEL WITH DIFFERENT NUMBER OF HIDDEN NEURONS

Number of hidden neurons	MSE error (Test)
1	1.267
2	1.279
3	1.267
4	1.269
5	0.278
6	0.286
7	0.286
8	0.287
9	0.279
10	0.282

Note that, for each number of hidden neuron, we run the script for 5 times then collect the best validation loss. Theoretically, hidden neuron between 5 and 10 has the best and relatively stable performance. Compare with the linear regression model, the MLP has a comparable or even better performance. The requirements of the assignment has been satisfied. Moreover, if we set the number of hidden neurons to 100, the mse error will become 1.317, too many hidden neurons will introduce the overfitting problem to the model, as the result the error will increase.

TABLE II
COMPARABLE PERFORMANCE BETWEEN MLP AND LINEAR REGRESSION

Methods	MSE error (Test)
MLP	**0.279**
LR	0.291

By comparing the training & validation processes of two different algorithms, both of the methods has a relatively fast descent, with only 25 to 50 training epochs, the loss start to fully converged. However, the MLP has less oscillations, by tuning the learning rate of the regression model to a smaller value (0.01), the oscillation become much more less.

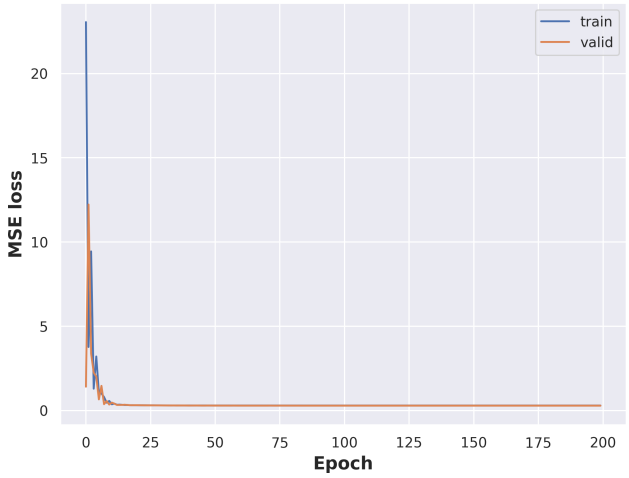

Fig. 4. Training & Validation of the linear regression model with lr = 0.01

III. CONCLUSION

In this report, we first collected the happiness scores of the people in various countries and related scores of various indicators, and integrated data from 2015 to 2019. We use this dataset to train multivariate linear regression models and multi-layer perceptrons.

For a linear regression model, we first need to establish the cost function, then we can measure the performance of this model through MSE. In a multilayer perceptron, since we have six indicators related to happiness scores, it is hard to visualize the input / output relationship via a 6-D plot. In this report, we also choose to use the MSE error as the measurement of the performance. This involves a choice of the number of hidden nodes in the hidden layer. If the number of choices is too small, the result will be biased. If the choice is too large, this will introduce the overfitting problem to the model

REFERENCES

[1] Kutner, Michael H., et al. Applied linear statistical models. Vol. 5. New York: McGraw-Hill Irwin, 2005.
[2] Haykin, Simon. Neural networks: a comprehensive foundation. Prentice Hall PTR, 1994.