The better solution: A comparison of Linear Regression and Multi-Layer Perceptron based on RMSE

*

Omik Save
School of Energy Matter and Transport Engineering
Arizona State University
Tempe, United States
osave@asu.edu

Gautam Sharma
CIDSE
Arizona State University
Tempe, United States
gsharm17@asu.edu

***Abstract*—This reports compares effectiveness of Multi-Layer Perceptron (MLP) model and linear regression model with gradient descent to predict happiness score based on unweighted parameters. The goal of this study is to determine optimum parameters of each model and then compare the best solution of each model. The models are evaluated on the basis of accuracy and root mean squared error (RMSE). A brief comparison of various model tuning parameters for each model is described. MLP surpassed regression model with an accuracy of 87.44% and RMSE of 0.45. Finally, the paper describes effect of data provided to each model and its effect on the accuracy of prediction.**

***Index Terms*—MLP, linear regression, multi-variable regression, neuron, SGD, ReLU, ADAM solver.**

I. Introduction

Started in 2012, World Happiness Report evaluates the satisfaction of populace in the countries they reside. Today, governments of countries around the world consider this report as a standard evaluation of their policies, leadership and trust amongst the people. A standard called dystopia or an imaginary worst country to live in, is used to compare the mentioned 155 countries on the basis of economy, health, corruption in government, freedom and generosity of people.

The challenge in this problem statement lies in unreliability of contributing factors mentioned above to calculate happiness score. Summation of individual factors and the residuals result in a highly deviate happiness score. Hence it is important to model the problem precisely to achieve an accurate fit.

This paper highlights key differences between simple linear regression and multi-layer perceptron (MLP) [4] with one hidden layer to fit data. Effects of factors like number of parameters, neurons of hidden layer, threshold function on loss or errors of both linear regression and MLP is discussed.

II. Problem Formulation

A. Linear regression

A simple linear regression [6] finds constants for a given set of input parameters in order to match the instantaneous output. It uses the mathematical form of slope-intercept to produce predictions given by,

$$y = mx + c \tag{1}$$

where y is the instantaneous output, x is the instantaneous input and c is a constant to measure offset from origin. A modified equation based on simple regression allows fitting dissimilar number of outputs and inputs given by

$$y = w_n^T \times x_n \tag{2}$$

where w represents weights of features and n denotes number of features. The weights of 2 are generated using training data and later the model is implemented on test data to find accuracy of prediction. An error is defined as the difference between predicted and the loss value.

B. Multi-layer perceptron

MLP is an multi-linked algorithm whose indivisible part is a single neuron. A neuron can be assumed as a junction that sends and receives data. For an artificial neural network with one hidden layer is given by,

$$y = \phi(b^{(2)} + W^{(2)}(\psi(b^{(1)} + W^{(1)}))) \tag{3}$$

where b is network bias and W is weights between input to hidden layer and hidden to output layer respectively. ϕ and ψ are activation functions chosen based on application. The properties of MLP model is determined by ReLU [7] activation functions and ADAM solver [3]. A ReLU is a rectifier function that outputs 0 for $x \leq 0$ and x for $x > 0$. Similarly, the

derivative of ReLU is 0 for $x \leq 0$ and 1 for $x > 0$. The ADAM solver features an adaptive learning rate as compared to constant learning rate for a stochastic gradient descent. Thus, ADAM updates learning rate per-parameter basis and achieves faster convergence even with noisy sparse data.

III. Algorithm Implementation

The dataset [2] obtained promotes features like health, trust in government (corruption), generosity, freedom, economy and dystopia residual (a combination of unexplained features). These features are used to correlate to an output score called happiness score. The training data consists of 80% of the dataset and test data occupies the rest 20%. Both linear regression model and MLP model are trained using the training data and their performance is evaluated in test data. In practice,

Algorithm 1: Linear Regression with gradient descent

initialize $W_0 = 0$ where $w \in W$ is weight matrix;
for $t = 0$ *to* T **do**
  fetch input $x_t \in \mathbb{R}$;
  hypothesis $h_0(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 ...$ so on
  Cost function=
  $J(\theta_0, \theta 1...) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^i) - y^i)^2$;
  update $\theta_j = \theta_j - \eta(y_i - \theta \cdot X_i) x_{ij}$
end

Algorithm 2: MLP algorithm

initialize $W_0 = 0$ where $w \in W$ is weight matrix;
for $t = 0$ *to* T **do**
  fetch input $x_t \in \mathbb{R}$;
  predict $\hat{y_j} = w_{ji} \cdot X$;
  calculate error $e_j = (d_j - y_j)$;
  update $w_{ji} = w_{ji} - \eta * e_j \phi_j'(v_j) y_i$
end

the MSE is updated [5] every step during the learning. If the MSE gradient is positve, the error will keep increasing if the weights are not changed. Hence, the decision to reduce weights is taken by the algorithm. Similarly, if the gradient calculated is negative, the weights would need to be increased to reach convergence.

IV. Result

Results were obtained by first implementing linear regression on the given data. Initially 6 variables are chosen as the input variables i.e Economy, Family, Health, Freedom, Generosity, Dystopia Residual. These 6 were specifically chosen from the lot as they were common in all data entries across each year. The following figure1 shows the sample output of plotting Family vs Happiness score.

The linear regression was done with the help of scikit-learn [1] in Python. The first set of training was done using all 6 features which gave us an accuracy of 0.84 while testing. The linear regression model found it hard to achieve higher

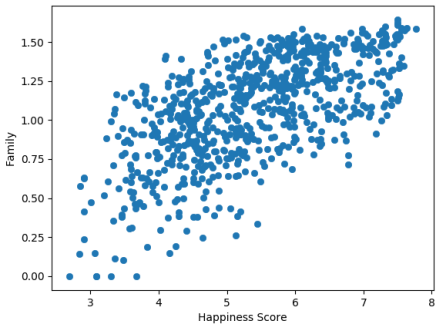

Fig. 1. Correlation of Family vs Happiness score

accuracy due to variations in the input feature vector. Next, we tried linear regression using only 1 feature. The linearly fitted model of happiness score with economy as an input can be seen in Figure 2

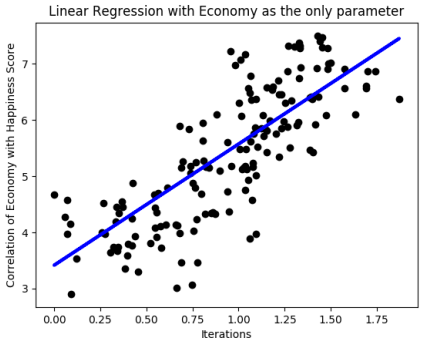

Fig. 2. Linear Regression with only one parameter

Furthermore, each parameter was considered in succession to evaluate the variation in error as a function of parameters. It can be observed in Figure3 that the lowest error occurs with 6 parameters. Taking only economy as our parameter

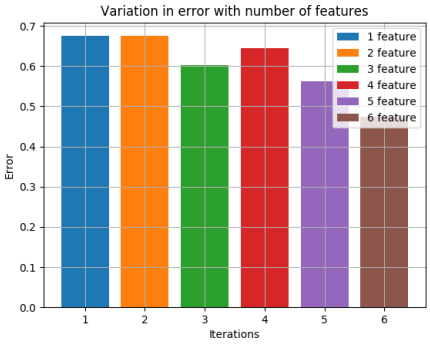

Fig. 3. Variation in error with parameters

we get a RMSE of 0.66 whereas taking only Family as the parameter we get RMSE of 0.834. Figure4 shows the error $y_{actual} - y_{predicted}$. In the next phase of training we apply a

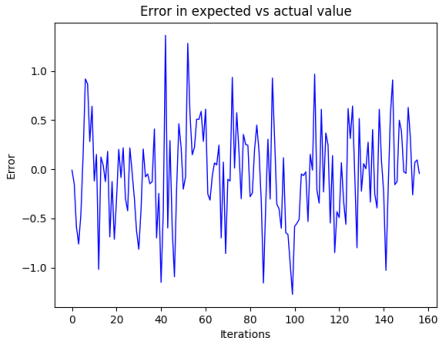

Fig. 4. Error during testing phase using regression

multi layered perceptron to the given problem. We have used scikit-learn to solve MLP convergence. We start by adding 10 neurons to our hidden layer, then 50 and then 100.The maximum recorded accuracy (0.8744) and RMSE of 0.45 was achieved by using ADAM solver with 1 hidden layer of 100 neurons. Figure 5 shows convergence of loss function with number of neurons. Upon experimenting it was also found

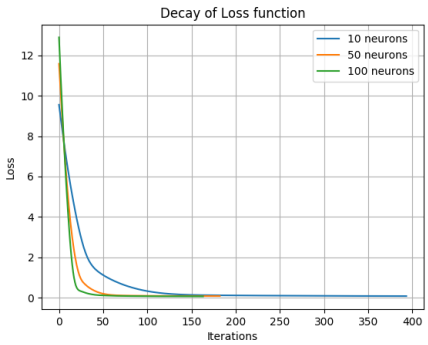

Fig. 5. Loss function with number of neurons

that ReLU performs much better than $tanh$ during back-propagation as $tanh$ leads to vanishing gradient. Figure 6 summarizes the result. Athough the result look minimalist they tend to increase with the complexity of the neural network.

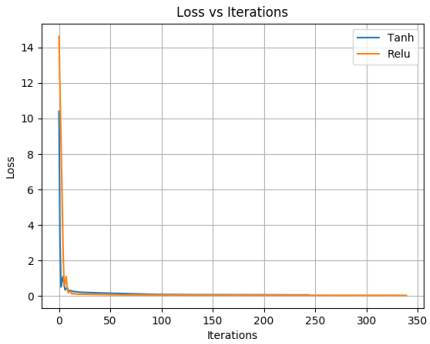

Fig. 6. $tanh$ vs ReLU

Similar to the aforementioned comparison of number of features with error we also compute the same for our MLP model. Figure 7 summarizes the result. We also tried to tune α to

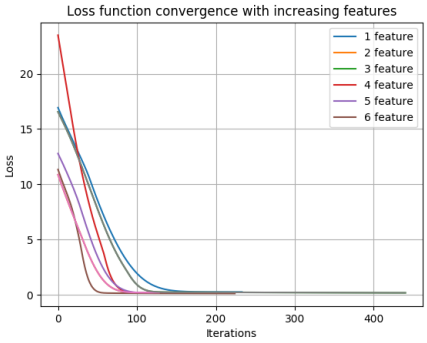

Fig. 7. Loss function convergence with increasing features

get the optimal learning rate. We found α = 0.01 to be the optimal learning rate for this given problem. Figure 8 shows our findings.

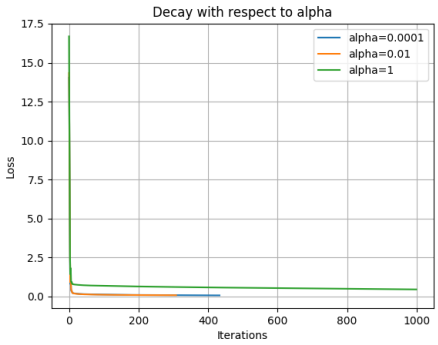

Fig. 8. Decay with varying alpha

In conclusion, the predictions of MLP model and linear regression with respect to actual observations can be visualized in Figure 9.

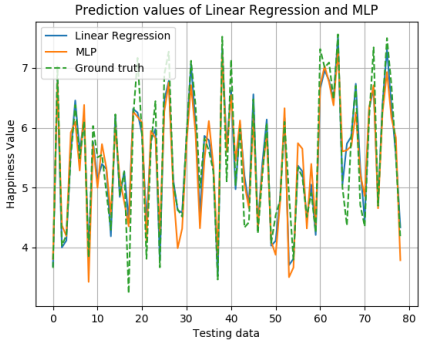

Fig. 9. Comparison of predictions by MLP and regression w.r.t ground truth

V. Conclusion and Future Scope

Although, both models were able to predict data accurately, MLP outperformed regression model. Firstly, the accuracy of MLP was 87.44% while regression model lagged at 84%. Secondly, the RMSE of MLP was 0.45 whereas the RMSE of regression was significantly greater at 0.66.

An important observation between the two models was that with lesser features, the error of regression model increased significantly. However, even with lesser features MLP errors decreased at slower rates but with sufficient epochs matched the accuracy of MLP with all features. Thus, MLP can perform efficiently with missing data or data with high variance.

Thus, a complex problem like happiness score, where contribution of individual factors towards output is not known, MLP can be preferred over regression fit to predict results.

References

[1] scikit-Learn Community. *scikit User Guide*. URL: https://scikit-learn.org/stable/user_guide.html.

[2] Kaggle and United Nations. *World Happiness Report*. 2017. URL: https://www.kaggle.com/unsdsn/world-happiness#2017.csv.

[3] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. arXiv: 1412.6980 [cs.LG].

[4] Fionn Murtagh. "Multilayer perceptrons for classification and regression". In: *Neurocomputing* 2.5 (1991), pp. 183–197. ISSN: 0925-2312. DOI: https://doi.org/10.1016/0925-2312(91)90023-5. URL: http://www.sciencedirect.com/science/article/pii/0925231291900235.

[5] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: *CoRR* abs/1609.04747 (2016). arXiv: 1609.04747. URL: http://arxiv.org/abs/1609.04747.

[6] Yale University. *Linear Regression*. 1997. URL: http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm.

[7] Bing Xu et al. *Empirical Evaluation of Rectified Activations in Convolutional Network*. 2015. arXiv: 1505.00853 [cs.LG].