E-ISSN 2581 – 7957
P-ISSN 2277 – 3916
CVR Journal of Science and Technology, Volume 18, June 2020
DOI: 10.32377/cvrjst1820

Reducing Overfitting Problem in Machine Learning using L1/4 Activation Function

Sathya Prakash Racharla[1], Mohammad Umar[2]and V. D. S. Krishna[3]
[1]Asst.Professor, CVR College of Engineering/CSE Department, Hyderabad, India
Email: prakashscits@gmail.com
[2]Asst.Professor, CVR College of Engineering/CSE Department, Hyderabad, India
Email: umarmohd25@gmail.com
[3]Asst.Professor, CVR College of Engineering/CSE Department, Hyderabad, India
Email: varanasi.kris@gmail.com

***Abstract:*Machine learning has various applications. Machine learning model has two problems.Overfittingand Underfitting.Underfitting is a statistical model or a machine learning algorithm,it cannot capture the underlying trend of the data.A statistical model is said to be overfitted, when it is trained with a lot of data.When model has trained on fewer features, the machine will be too biased, and then the model gets underfitting problem. So, it is needed to train the model on more features and there is one more problem occurs.Overfitting problem can be reduced by using regularization functions and data augmentation. In the previous research on activation functions, Hock Hung Chieng Proposed an activation function called Flatten-T Swish: a thresholdReLU, which is a multiplication of Relu and sigmoid function.**

Index Terms: Flatten-T swish, Machine learning, activation function

I. INTRODUCTION

To design a machine learning model, our model has to be trained on dataset. Our model has to be trained based on some features which are given. Sometimes our model accuracy is low on new data. There may be two types of problems occur. Overfitting and Underfitting.

A. Underfitting:

If few features are given to train the model, sometimes it can wrongly identified unknown data as shown in figure 1. To reduce this problem feature selection should be incresed. In figure1, there is an expanation about under fitting, perfect fitting and Overfitting. A statistical model[1] or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trendof the data. In the figure line cannot beclustering perfectly.

B. Overfitting:

If huge features are given to our train model also, will get problem to our statistical problem, then our model will get Overfitting problem as shown in figure 1. When modelhas trained on fewer features, the machine will be too biased, and then the model getsunderfitting problem. So, it is needed to train the model on more features and there is one more problem occurs. If model has trained on more and more features, more variances comeinto the picture, and then our model identification efficiency is very less [2]. It leads our model to becomeworse. This problem is known as Overfitting problem. A statistical model is said to be Overfitted, when model has trained with a lot of data. When itgets trained with so much of data, it starts learning from the noise and inaccurate data entries in our data set.

Bias: it calculates our model predictions. It shows the correct difference when it is compared to the actual value.Variance: It denotes the predictions given by our model for a given point which varies between different realizations.

How to overcome underfitting?

- Find more features
- Try high variance machine learning models (Decision tree, K-NN, SVM)

When the test error is much heavier than training error, it can be concluded that there is Overfitting problem.

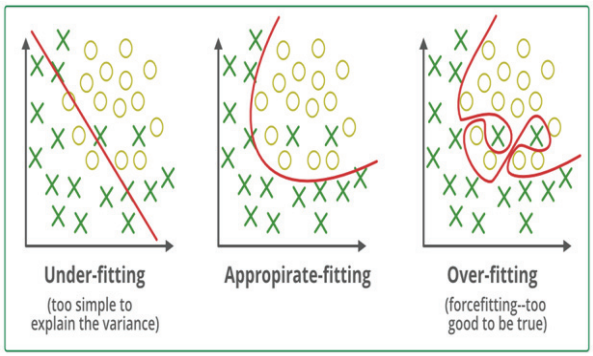

Figure 1. Underfitting and Overfitting cases in ML

E-ISSN 2581 – 7957
P-ISSN 2277 – 3916

DOI: 10.32377/cvrjst1820

II. Various Activation Functions in Neural Networks

A. Linear Function

General equation for a straight line, **y=mx**, this equationis used for linear function.

- Apply this equation for input layer whereinactivation functions to last layers.
- Linear function range is (-∞,+∞)
- There are some issues in this linear function. If this function is differentiated, it is independent of x, i.e input.The best example for this type of function is house rent analysis. This is regression problem, using this linear function, linear function for output layers can be added.

B. Sigmoid Function

This activation function plotted as' S' shape. General equation for a straight line, $Z = 1/(1 + e^{-x})$, this equation is used for linear function[3][10].

- Apply this equation for input layer whereasactivation functions to last layers.
- Linear function range is (0,1)
- There are some issues in this sigmoid function, when differentiated this function dependents on 'x' input.
- The best example for this type of function is house rent analysis. This is regression problem[6], using this linear function, linear function for output layers can be added.
- Result can be determined to be ***1*** if value is greater than **0.5** and ***0*** otherwise.

C. Tanh Function

This activation function also known as **Tangent Hyperbolic function[4].**.This isfar better than sigmoid and linear activation functions.This function can be derived from sigmoid activation function by using shifting principle.

$g(x) = \tanh(x) = 2/(1 + e^{-2x}) - 1$
or
$\tanh(x) = 2 * \text{sigmoid}(2x) - 1$

- Apply this equation for input layer,thenactivation functions to last layers.
- Linear function range is (-1,+1)
- There are some issues in this linear function. If this function is differenciated, it is independent of x, i.e input.
- Due to the range (-1,+1), the mean becomes 0. Then the data will be centered.
- Due to this, the next layer learning becomes much easier.

D. Rectified linear unit (RELU)

Mostly this type of activation function is used in *hidden layers* of convolutional Neural network. It is a nonlinear function[5].

General equation for a straight line, ***R(x) = max(0,x)***, this equation is used for linear function.

- Apply this equation for input layer, whereasactivation functions to last layers.
- Activation function range is (0,+∞)
- There are some issues in this linear function. If this function is differenciated, it is independent of x, i.e input.
- It gives an output x if x is positive and 0 otherwise.
- RELU activation function is faster than all other activation functions.

E. Softmax Function

This activation function is also similar to sigmoid function, which is useful for all classification problems.

This activation function is a nonlinear function [7]. The activation function would squeeze the outputs for each module between 0 and 1 and also divide by the sum of these outputs.

To choose activation function, RELU can be used, which is suitable for all applications.
For binary classification applications,sigmoid activation function can be used.

K-folds cross validation:
In K-folds cross validation, it is needed to divide our data into k different subsets or folds. K-1 subsets are used to train our data and leave the last subset or last folder as test data. Average the model against each of the folds and then finalize our model, after that it has to betested against the test set.
Final accuracy = average (round1, round2, round3 …)
How to overcome Overfitting?

- During training only, our model has to be tested against validation data.
- If validation accuracy is lower, regularization has to be done.
- Repeat this process of regularization, until no Overfitting result.

In order to create less complex (parsimonious) model, when there are large number of features in the dataset, some of the Regularization techniques used to address Overfitting and feature selections are:
1. L1 Regularization
2. L2 Regularization
A regression model that uses L1 regularization technique is called *Lasso Regression* and model which uses L2 is called *Ridge Regression*.
The key difference between these two is the penalty term.
Ridge regression adds "squared magnitude" of coefficient as penalty term to the loss function[2]. Here the *highlighted* part represents L2 regularization element.

$$\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\,\beta_j\right)^2 + \lambda \sum_{j=1}^{p} \beta_j{}^2 \quad (1)$$

Here, if *lambda* is zero then you can imagine to get back to OLS. However, if *lambda* is very large then it will add too much weight and it will lead to underfitting. Having said that

CVR Journal of Science and Technology, Volume 18, June 2020
DOI: 10.32377/cvrjst1820

it's important how *lambda* is chosen. This technique works very well to avoid Overfitting issue.
Lasso Regression (Least Absolute Shrinkage and Selection Operator) adds "*absolute value of magnitude*" of coefficient as penalty term to the loss function.

$$\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij}\,\beta_j\right)^2 + \lambda \sum_{j=1}^{p}\left|\beta_j\right| (2)$$

Again, if *lambda* is zero then it is good to get back to OLS whereas very large value will make coefficients zero hence it will underfit.
The key difference between these techniques is that Lasso shrinks the less important feature's coefficient to zero thus, removing some feature altogether. So, this works well for feature selection in case huge number of features are there.
Traditional methods like cross-validation, stepwise regression to handle Overfitting and perform feature selection work well with a small set of features but these techniques are a great alternative when dealing with a large set of features.

III. Proposed Algorithm

To improve the performance in the existed algorithms, I want to develop a new algorithm which combines the convolutional neural network with proposed Activation function called S-inclined Activation function.
To reduce the Overfitting problem,regularization functions and data augmentation are used. In the previous research on activation functions, *Hock Hung Chieng*proposedan activation function [8] called Flatten-T Swish: a thresholdReLU, which is a multiplication of Relu and sigmoid function. In my research, I want to convolute GRelu(Gaussian) and Sigmoid functions for a better results.

$$\text{GReLU(x)}=\begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases}$$

$$\text{Sigmoid(x)}=\frac{1}{1+ex}$$

Absolute value of Convolution of these functions

$$|\text{GReLU(x)}*\text{Sigmoid(x)}| = \sum_i (x_i - \hat{x}_i)^2 + \lambda \sum_i \sqrt[4]{w_i} (4)$$

5. Development of proposed algorithm L1/4 Regularization
As compared to L1 regularization and L1/2 regularization, our proposed regularization method L1/4- regularization will give for better performance. L1/4- regularization pushes all the weights absolute fourth order value.

Proposed Algorithm: L1/4 Regularization:
As compared to L1 regularization, L1/4- regularization is best opted methodology for better performance. L1/4- regularization pushes all the weights absolute fourth order value. According to XU ZongBen, ZHANG Hai [6] if the order of weight is decreased more accurate values in fitting problems can be acheived, so

$$\text{Minimize absolute } \left(\sum_i (y_i - \hat{y}_i)^2 + \lambda \sum_i \sqrt[4]{w_i}\right) (3)$$

Here also, if λ is zero then get back to OLS. However, if *lambda* is very large then it will add too much weight and it will lead to underfitting. Having said that it's important how *lambda* is chosen. This technique works very well to avoid Overfitting issue.
Lasso Regression (Least Absolute Shrinkage and Selection Operator) adds "*absolute value of magnitude*" of coefficient as penalty term to the loss function.

layer 1(Hidden layer) :-
X(1) = Y(1)W + z(1)
z(1) = a(1)
Here,

- *W(1) be the vectorized weights assigned to neurons of hidden layer i.e. w1, w2, w3 and w4*
- *z(1) is the vectorized form of any linear function.*
- *W be the vectorized input features i.e. i1 and i2*
- *z is the vectorized bias assigned to neurons in hidden layer i.e. a1 and a2*
- *a(1) is the vectorized output of layer 1*

*(**Note:** don't consider activation function here)*

Calculation at Output layer:
Layer 2 (output layer) :-

// 2 is output from layer 1
a(2) = Z(2)a(1) + c(2)
k(2) = c(2)

// Replace the value of c(1) here
c(2) = (Z(2) * [Z(1)X + z(1)]) +a(2)
z(2) = [Z(2) * Z(1)] * X + [Z(2)*a(1) + a(2)]

Let,
[Z(2) * Z(1)] = Z
[Z(2)*a(1) + a(2)] = a

Final output : a(2) = Z*X + a
Which is again a linear function

```
Train on 34300 samples, validate on 14700 samples
Epoch 1/10
34300/34300 [==============================] - 14s - loss: 0.4231 - acc: 0.8652 - val_loss: 0.1756 - val_acc: 0.9461
Epoch 2/10
34300/34300 [==============================] - 12s - loss: 0.1771 - acc: 0.9487 - val_loss: 0.1526 - val_acc: 0.9573
Epoch 3/10
34300/34300 [==============================] - 13s - loss: 0.1296 - acc: 0.9625 - val_loss: 0.1111 - val_acc: 0.9683
Epoch 4/10
34300/34300 [==============================] - 13s - loss: 0.1055 - acc: 0.9691 - val_loss: 0.1137 - val_acc: 0.9688
Epoch 5/10
34300/34300 [==============================] - 12s - loss: 0.0924 - acc: 0.9739 - val_loss: 0.1184 - val_acc: 0.9677
Epoch 6/10
34300/34300 [==============================] - 12s - loss: 0.0821 - acc: 0.9755 - val_loss: 0.1059 - val_acc: 0.9712
Epoch 7/10
34300/34300 [==============================] - 13s - loss: 0.0714 - acc: 0.9794 - val_loss: 0.1103 - val_acc: 0.9699
Epoch 8/10
34300/34300 [==============================] - 13s - loss: 0.0635 - acc: 0.9812 - val_loss: 0.0965 - val_acc: 0.9766
Epoch 9/10
34300/34300 [==============================] - 13s - loss: 0.0588 - acc: 0.9825 - val_loss: 0.1133 - val_acc: 0.9718
Epoch 10/10
34300/34300 [==============================] - 15s - loss: 0.0572 - acc: 0.9826 - val_loss: 0.1085 - val_acc: 0.9729
```

Figure 2. Results of L1/4 activation function for 34,000 samples

IV. CONCLUSIONS

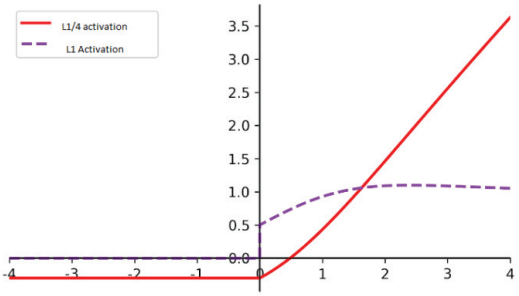

Figure 3. Efficiency of ML model

TABLEI.
COMPARISION OF ACTIVATION FUNCTIONS

Activation function	accuracy	Learning rate
sigmoid	90.58	1.35
ReLU	89.9	0.56
L1	91.58	0.78
L2	96.5	2.72
L1/4	96.9	3.5

L1 regularization has the tendency to produce sparse coefficients. Which means Lasso shrinks the less important feature's coefficient to zero thus removing some feature altogether. L2 regularization on the other hand does not remove most of the features.

L1 regularization formula does not have an analytical solution but L2 regularization does. So, it is computationally more efficient to do L2 regularization.

REFERENCES

[1] I. Bilbao and J. Bilbao, "Overfitting problem and the over-training in the era of data: Particularly for Artificial Neural Networks," 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, 2017, pp. 173-177.

[2] M. Molinier and J. Kilpi, "Avoiding Overfitting When Applying Spectral-Spatial Deep Learning Methods on Hyperspectral Images with Limited Labels," IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 2019, pp. 5049-5052.

[3] A Flexible Sigmoid Function of Determinate Growth xinyou yin, jangoudriaan, egbert a. lantinga, janvos, huub j. spiertz Annals of Botany, Volume 91, Issue 3, February 2003, Pages 361–371,https://doi.org/10.1093/aob/mcg029
Published: 01 February 2003.

[4] On Overfitting in Model Selection and Subsequent Selection Bias in Performance Evaluation Gavin C. Cawley, Nicola L. C. Talbot; 11(70):2079−2107, 2010.

[5] D. Stursa and P. Dolezel, "Comparison of ReLU and linear saturated activation functions in neural network for universal approximation," 2019 22nd International Conference on Process Control (PC19), StrbskePleso, Slovakia, 2019, pp. 146-151.

[6] R. Murugadoss and M. Ramakrishnan, "Universal approximation using probabilistic neural networks with sigmoid activation functions," 2014 International Conference on Advances in Engineering & Technology Research (ICAETR - 2014), Unnao, 2014, pp. 1-4.

[7] Journal of Data Science 12(2014),563-574 Softmax Model as Generalization upon Logistic Discrimination Suffers from Overfitting F. Mohammadi Basatini1 and Rahim Chinipardaz2∗ 1 Deartment of Statistics, Shoushtar Branch, Islamic Azad University. 2 Deartment of Statistics, ShahidChamranUniversityn

[8] Flatten-T Swish: a thresholdedReLU-Swish-like activation function for deep learning Hock Hung Chieng, Noorhaniza Wahid, Pauline Ong, Sai Raj Kishore Perla Neural and Evolutionary Computing (cs.NE); Machine Learning (cs.LG); Machine Learning (stat.ML) journal reference:International Journal of Advances in Intelligent Informatics, 4(2), 76-86 DOI:10.26555/ijain.v4i2.249

[9] B. Ding, H. Qian and J. Zhou, "Activation functions and their characteristics in deep neural networks," 2018 Chinese Control And Decision Conference (CCDC), Shenyang, 2018, pp. 1836-1841.

[10] K. Hara and K. Nakayamma, "Comparison of activation functions in multilayer neural network for pattern classification," Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94), Orlando, FL, USA, 1994, pp. 2997-3002 vol.5.

[11] Khoshgoftaar, T.M., Allen, E.B. Controlling Overfitting in Classification-Tree Models of Software Quality. *Empirical Software Engineering* **6**, 59–79 (2001). https://doi.org/10.1023/A:1009803004576.