ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 11, November 2016

Survey on objective functions used in Deep Neural Network based Sentiment Analysis Models

M.Poongothai,
Department of Information Technology,
Institute of Road and Transport Technology,
Erode, India.
poongothaiirtt@gmail.com

M.Sathyakala,
Department of Information Technology,
Institute of Road and Transport Technology,
Erode, India.
msathyakalairtt@gmail.com

Abstract

The aim of improving the efficiency of a Deep Neural Network based Learning model is to reduce the optimization function that could be split based on the problems of classification and regression. The concept of objective functions is crucial in Deep Learning as it needs to be optimized in order to get better prediction or a more efficient model. Some of the objective functions used in Neural Network based Learning are discussed in this paper.

Keywords:
Deep Neural Network, Learning Model, Objective Function

1. Introduction

Deep learning is one of Data Science's surging fields that relies heavily on more knowledge. The choice of the activation function is Neural Network is an important step. In Binary classification problem, the sigmoid activation function is sufficient whereas in other problems, the Rectified Linear Unit activation function could be used. Some of the other important parameters in Deep Learning are Weights, Bias and hyper parameters such as the Learning rate, number of hidden layers, and so on.

In Deep Learning, the idea of objective functions is important because it needs to be optimized in order to achieve a better prediction or a more effective model. The function used to minimize or maximize is called the objective function or criterion. When it is used for minimization, it is called as a cost function, loss function, or error function. The cost function reduces all the various good and bad aspects of a possibly complex system down to a single number, a scalar value, which allows candidate solutions to be ranked and compared [1]. In calculating the error of the model during the optimization process, a loss function must be chosen. This can be a challenging problem as the function must capture the properties of the problem and be motivated by concerns that are important to

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 11, November 2016

the project and stakeholders. It is important, therefore, that the function faithfully represent the design goals. If a poor error function is chosen, creates the unsatisfactory results.

2. Objective Functions

2.1 Maximum-Likelihood Estimation:

Maximum-likelihood estimation is used as the model for parameter estimation. The Bayesian Decision theory is about designing a classifier that minimizes total expected risk, especially, when the costs (the loss function) associated with different decisions are equal, the classifier is minimizing the error over the whole distribution. Thus, the Bayes Decision Rule is stated as "decide w_1 if P(w1|x)>P(w2|x); otherwise w_2", where w_1,w_2 are predictions of different classes.

From a perspective of minimizing error, it can also be stated as,

$w=\arg\min(w) \int_{-\infty}^{\infty} P(\text{error} \mid x)\, P(x)\, dx$,

where P(error|x)=P(w1|x) if we decide w_2 and P(error|x)=P(w_2|x) if we decide w_1.

By applying Bayes' theorem :

$$P(w_i|x) = \frac{P(x|w_i)\, P(w_i)}{P(x)} \quad (1)$$

and if we further assume the zero/one loss function, which is a same loss for all errors, the Bayes Decision rule can be reformulated as:

$$h_{bayes} = \arg\max_{w} P(x|w)\, P(w) \quad (2)$$

where h_{bayes} is the prediction and P(w) is the priori probability.

2.1 Regression Loss Functions

2.1.1 Mean Absolute Error

In Regression problems, the intuition is to reduce the difference between the actual data points and the predicted regression line. Mean absolute error is one such function to do so which takes the mean of the absolute value of the difference between the actual and the predicted value for all the examples in the data set. The magnitude of errors are measured without the directions. Though it is a simple objective function but there is a lack of robustness and stability in this function. Also known as the L1 loss, its value ranges from 0 to infinity.

Prediction Error → Actual Value - Predicted Value

This prediction error is taking for each record after which we convert all error to positive. This is achieved by taking Absolute value for each error as below;

Absolute Error → |Prediction Error|

Finally to calculate the mean for all recorded absolute errors (Average sum of all absolute errors).

MAE = Average of All absolute errors

$$mae = \frac{\sum_{i=1}^{n} abs\,(y_i - \lambda(x_i))}{n} \quad (3)$$

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 11, November 2016

2.1.2 Mean Squared Error

Similar to the mean absolute error, it squares the difference between the real and the expected data points instead of taking the absolute value. To highlight those points that are further away from the regression line, the squaring is performed. Mean Squared Error is often referred to as the cost function in regression issues and the objective is to minimize the cost function to its optimum global level in order to get the best fit line to the data.

This decrease in loss or gradient descent is a gradual process where it first a value is initialized and then the parameters are modified to the global optimum at each descent. The descent speed depends on the learning rate that needs to be modified as a very small value will lead to a gradient descent of a slow phase, whereas a greater value could not converge at all. Mean Squared Errors, however are sensitive to outliers. The range of values is always between 0 and infinity.

The prediction error is the difference between the true value and the predicted value for an instance. In [9], an image is compared (i.e.,), the pixel loss is essentially a measure of how far is the target image's pixels are from the predicted/generated image's pixels.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \quad (4)$$

m,n are the image matrix rows and columns, I is the predicted image, K is the actual image.

2.1.3. Huber

The penalty incurred by the procedure for estimating f is defined by the Huber loss function. The Huber function is linear for large values, while it is quadratic in nature for small values. The degree to which the value needs to be small depends entirely on the hyper parameter delta to make it quadratic. It was indeed possible to tune this hyper parameter. Also referred to as the Smooth Mean Absolute Error, relative to the other functions, the sensitivity of Huber loss to outliers is less. The loss of Huber at zero is distinguishable. When the hyperparameter delta approaches 0, the Huber loss approaches the Mean Absolute Error, and when the delta approaches infinity, it approaches the Mean Squared Error. The value of the delta will decide the amount of outlier you are willing to consider. The residuals greater than delta are minimized by L1 while the residuals smaller than delta are minimized by L2.

$$L_\delta = \begin{cases} \frac{1}{2}(y - f(x))^2, & if\ |y - f(x)| \le \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, & otherwise \end{cases} \quad (5)$$

Quadratic

Linear

2.1.4. Log-Cosh loss

A function for regression optimization, which is smoother than L2. The hyperbolic cosine logarithm of the prediction error is known as the function of log-cash loss. It is equal to half of its square for a small value, and for a large value, it is equal to the

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 11, November 2016

difference between its absolute logarithm value of 2. By occasional incorrect predictions, log-cosh is not affected too much and almost operates close tc the mean squared error. Unlike Huber, it can be distinguished twice. Log-cosh, however, ofter suffers from the gradient problem. In [8] Robust estimation methods, can be found in the literature for training artificial neural networks. For instance, a smoother function than L2, using a logcosh loss, has been proposed.

2.1.5. Cosine Proximity

The cosine proximity is determined between the expected and the actual value by this loss function which minimizes the dot product between them. In this case, there is a maximum similarity between the unit vectors if they are parallel, defined by 0. In the case of orthogonality, however, it is expressed differently by +1. In [7],the cross-patent comparisons using NLP vector outputs are much more internally consistent than using vectorizations of patent class selections. For two patents, *i* and *j*, the cosine similarity between them is:

$$sim(i,j) = \frac{PV_i \cdot PV_j}{\| PV_i \| \| PV_j \|} \qquad (6)$$

Where PV_i is the patent vector representation of *i*. This is preferred to Euclidean distance as it is factors in the "size" of the vector; a Euclidean distance measure would assign positive distance to two vectors that contained the exact same words, but of different quantities. Cosine similarity normalises all measures to be in the range [−1, 1].

2.1.6. Poisson

The deviation of the observed distribution from the expected distribution is calculated by the function of Poisson loss, which is the variant of a Poisson distribution. The distribution is limited to a binomial for a normal approximation, as the likelihood becomes null and trials become infinite. The Exponential Log Likelihood is comparable to the Poisson in Deep Learning. In[7], the author proposed a modified Poisson regression approach(i.e., Poisson regression with a robust error variance) to estimate this effect measures directly. A simple 2-by-2 table is used to justify the validity of this approach. The subject *i* has an underlying risk that is a function of x_i, say $\pi(x_i)$. Because $\pi(x_i)$ must be positive, the logarithm link function is a natural choice for modeling $\pi(x_i)$, giving

$$\log[\pi(x_i)] = \alpha + \beta x_i. \qquad (7)$$

The relative risk (RR) is then given by exp(β). If a Poisson distribution is assumed for y_i, the log-likelihood is given by

$$l(\alpha, \beta) = C \cdot \sum_{i=1}^{n} [y_i(\alpha + \beta x_i) - \exp(\alpha + \beta x_i)], \qquad (8)$$

2.1.7. Hinge

The loss function used is known as the Hinge loss for training classifiers, which meets the maximum-

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 11, November 2016

margin criterion. The output of the predicted function should be raw in this case. The sign and the expected output of the real output data point will be the same. If the expected output is greater than 1, the loss will be equal to zero. With the actual results obtained, the loss increases linearly when the sign is not equal. It's mainly used in Support Vector Machines. In[5], proposed the optimization of a certain convex loss function φ, analogous to the hinge loss used in support vector machines (SVMs). Its convexity ensures that the sample average of this surrogate loss can be efficiently minimized.

The convex surrogate loss

$$\phi_d(z) = \begin{cases} 1 - az & \text{if } z < 0, \\ 1 - z & \text{if } 0 = z < 1, \\ 0 & \text{otherwise.} \end{cases} \quad (9)$$

where a = (1−d)/d ≥ 1. The next result states that the minimizer of the expectation of the discrete loss.

$l_{d,\delta}(z)$and the convex loss $\varphi_d(z)$ remains the same.

2.1.8. Cross Entropy

The Cross Entropy loss function is used in the binary classification problem, where the labels are either 0 or 1. However, in the case of a multi-classification problem, multi-class cross entropy is used. The divergence is evaluated by the cross entropy function between two probability functions. The difference between the two distributions will be large if the cross entropy is large, but if the difference is small, they are the same. Cross entropy does not suffer from the issue of slow divergence, as seen in the mean squared error function due to the Sigmoid activation function. The learning speed is fast when the difference is large and slow when the difference is small. Chances of reaching the global optimum is more in case of the cross entropy loss function because of its fast convergence. In [4], the ANN is implemented using the cross entropy error function in the training stage. The cross entropy function is proven to accelerate the backpropagation algorithm and to provide good overall network performance with relatively short stagnation periods.

Cross-entropy can be calculated using the probabilities of the events from P and Q, as follows:

H(P, Q) = – sum x in X P(x) * log(Q(x)) (10)

Where P(x) is the probability of the event x in P, Q(x) is the probability of event x in Q and log is the base-2 logarithm, meaning that the results are in bits.

2.1.9. Kullback-Leibler

Kullback-Leibler divergence, also known as entropy, information divergence, measures the diversion of one probability distribution from a second predicted probability distribution. As it is a distribution-wise asymmetric measure, it is not considered a statistical measure of spread. Similarity is assumed when the value of the loss function for Kullback-Leibler is 0, whereas 1 implies that distributions behave differently.

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 11, November 2016

In[3], KullbackLeibler (KL) divergence is used to exploit a safety mode on opponent modeling. the Kullback– Leibler divergence of Q from P, denoted DKL(P||Q), is a measure of the information lost when Q is used to approximate P. The definition of D_{KL} is ,

$$D_{KL}=(P||Q)=\sum \ln\left[\frac{P(i)}{Q(i)}\right]P(i) \qquad (11)$$

In words, it is the expectation of the logarithmic difference between the probabilities P and Q, where the expectation is taken using the probabilities P. [6] discussed about the combination of Graph cut liver segmentation and Fuzzy with MPSO tumor segmentation algorithms. The system determines the elapsed time for the segmentation process. The accuracy of the proposed system is higher than the existing system. The algorithm has been successfully tested in multiple images where it has performed very well, resulting in good segmentation. It has taken high computation time for the graph cut processing algorithm. In future work, we can reduce the computation time and improves segmentation accuracy.

2.1.10. Negative Logarithm Likelihood

The accuracy of a classifier is calculated by the negative logarithm likelihood function, used commonly in neural networks. This function, which gives out the likelihood of each class, follows the principle of probabilistic confidence. The negative log likelihood of Py_i

$$L_i=-\log(p_{y_i}) \qquad (12)$$

where $Py_k= e^{f_k} / \sum_j e^{f_j}$

f_k is an element for a certain class k in all j classes.

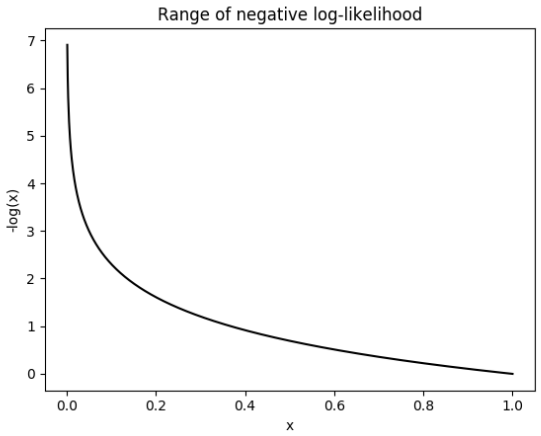

Figure: *The loss function reaches infinity when input is 0, and reaches 0 when input is 1.*

In [2], the most prominent features in model selection criteria that have been used so far in iterated density estimation evolutionary algorithms (IDEAs, EDAs, PMBGAs) were analyzed.The negative log–likelihood is a basis of the inference features when the Kullback–Leibler divergence is used.

3. Conclusions:

The efficiency of a Deep Neural Network based Learning model is depends on the reduction of the optimization function that could be split based on the problems of classification and regression. The concept of objective functions is crucial in Deep

ISSN (ONLINE): 2454-9762
ISSN (PRINT): 2454-9762

International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)
Vol. 2, Issue 11, November 2016

Learning as it needs to be optimized in order to get better prediction or a more efficient model. Some of the objective functions used in Neural Network based Learning are discussed. Each objective function has its unique functionality, applying the this function for the apt situation is again an issue.

References:

[1] Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks, 1999.

[2] Peter A.N. Bosman, Dirk Thierens Negative Log–Likelihood And Statistical Hypothesis Testing As The Basis Of Model Selection In IDEAs,2000.

[3] Using Kullback-Leibler Divergence to Model Opponents in Poker Jiajia Zhang1 , Xuan Wang2 , Lin Yao3 , Jingpeng Li 1 , Xuedong Shen 1, Computer Poker and Imperfect Information: Papers from the AAAI-14 Workshop,2014.

[4] "G.E.Nasr,E.A.Eadr," Cross Entropy Error Function in Neural Networks: Forecasting Gasoline Demand" Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference, May 14-16, 2002, Pensacola Beach, Florida, USA.

[5] Peter L. Bartlett, Marten H. Wegkamp,"Classification with a Reject Option using a Hinge Loss", Journal of Machine Learning Research 9 (2008) 1823-1840.

[6] Christo Ananth, D.R.Denslin Brabin, "ENHANCING SEGMENTATION APPROACHES FROM FUZZY K-C-MEANS TO FUZZY-MPSO BASED LIVER TUMOR SEGMENTATION", Agrociencia, Volume 54, No. 2, 2020,(72-84).3.

[7] Yihua Chen, Eric K. Garcia, Maya R. Gupta," Similarity-based Classification: Concepts and Algorithms", Journal of Machine Learning Research 10 (2009) 747-776

[8] R. Neuneier and H. G. Zimmermann. How to train neural networks. In Neural Networks: Tricks of the Trade. 1998.

[9] Dr. Rajesh Mehra, "Estimation of the Image Quality under Different Distortions"International Journal of Advanced Trends in Computer Science and Engineering,2016.