International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Improving Document Categorization Models using Explanatory MLP and Batch Normalization: A Novel Methodology Featuring Logistic Regression Weights Transfer

Suresh Reddy Gali[1*], Annaluri Sreenivasa Rao[2], Kranthi Kiran Jeevangar[3], Bhuvana Manchikatla[4], Dhanush Gummadavalli[5], Naga Shivani Karra[6]

Submitted: 29/01/2024 **Revised**: 07/03/2024 **Accepted**: 15/03/2024

Abstract: In this era of increasing textual documents on various platforms, it is important to have a text classification system that can categorize the text documents. We extracted the Reuters8 dataset and reduced its dimensions by using information gain. Later we applied the MLP model on the dataset to classify the text documents. The MLP model is applied by modifying in four different ways, first MLP is applied with assumed weights and to tune the hyperparameters GridSearchCV is used. Then batch normalization was performed where input of each layer is normalized by adjusting the activations. Next explanatory MLP was performed where the weights are taken from linear regression. Finally, linear knowledge was performed where no. of neurons in a hidden layer are taken in a sequence based on the number of categories in the dataset. In the Reuters8 dataset there are 8 classes. Out of all the variations it is found that explanatory MLP has given the best results.

Keywords: *Deep learning, Explanatory MLP, Multi-Layer Perceptron, Linear Knowledge Transfer, Linear Regression.*

1. Introduction

In text categorization, every document will be assigned a class according to the content it has. Text classification systems have become a part of digital life in many ways. Text classification can be used in many aspects like social media monitoring, email categorization, and many more. Text classification can be used in e-commerce to provide a personalized product for customers based on their backgrounds in choices.[1] It can also be helpful in the medical field by automating patient inquiries based on their previous records for diagnosis and therapy history. Text classification has undergone many changes because of deep learning. Deep learning provides advanced ways for analyzing complicated patterns from textual data.[2] For extracting hierarchical features and giving accurate results in classification tasks, deep learning can be used. Deep learning models are very flexible and adaptive. They have high scalability, so generalization is made easier by using deep learning models. These features make deep learning models useful in many ways in real world text categorization. Dimensionality reduction helps to bring down the complexity of textual data. Lowering dimensionality before applying the model gives the best results when text classification is performed. Dimensionality reduction means reducing the number of features, this helps to avoid overfitting and makes calculations easy. It also helps to analyze the decision made by the classification model. Additionally, dimensionality reduction addresses the problem of handling large no of dimensions besides aids in the management of sparse data, leading to text classifiers that are more accurate. classification of such enormous documents into multiple categories, calls for a lot of time and work.[3]

To facilitate this, the Bag of Words technique is employed after removing stop words and stemming words. Visualize representing each document as a list indicating the frequency of each stemmed word, allowing the computer to grasp the essence without concerning itself with the order to enhance the computer's proficiency in this task, it's essential to identify the most crucial words – think of it as highlighting key information. The method employed for this purpose is information gain, assessing how much each word contributes to the computer's effectiveness. Here 90% retained data is used for the further process. This [4] is especially vital for tasks such as categorizing texts. sentiment analysis, long content classification needs additional time and work because of large terminology, more disturbance, and unnecessary information[5].

Now, a diverse array of classification algorithms powers document classification and text similarity tasks. In this

[1 *]*Professor, Department of IT, VNR Vignana Jyothi Institute of Engineering and Technology Hyderabad, India*
[2] *Assistant Professor, Department of IT, VNR Vignana Jyothi Institute of Engineering and Technology Hyderabad, India*
[3] *Department of IT, VNR Vignana Jyothi Institute of Engineering and Technology Hyderabad, India*
[4] *Department of IT, VNR Vignana Jyothi Institute of Engineering and Technology Hyderabad, India*
[5] *Department of IT, VNR Vignana Jyothi Institute of Engineering and Technology Hyderabad, India*
[6] *Department of IT, VNR Vignana Jyothi Institute of Engineering and Technology Hyderabad, India*
** Corresponding Author Email: sreenivasarao_a@vnrvjiet.in*

research Supervised learning algorithm MLP is experimented such as batch normalization, explanatory MLP, traditional MLP and linear Knowledge transfer with labelled data, and they learn from the examples and adapt at sorting documents.

2. Related works

Text classification is the task of labeling text data from a predetermined set of thematic labels**[5]**. There are many current works studied by researchers about text documents classification using machine learning algorithms and multi-layer perceptron. Few are mentioned below.

In this thesis [6], they proposed a unique hybrid text classification model built on deep belief network and softmax regression. To solve the scarce high-dimensional matrix calculation problem of texts data, a deep belief network is introduced[6].

In a study [7] it is explained that MLP uses gradient descent investigate to shrink mean square error between real output and expected output by adjusting weights. The signal error will be maintained back to the network and the weights will be altered to bring the actual output close to the expected output. One of the previous research projects has classified stunting disease using MLP with Grid Search CV.[8] MLP results depend on hyperparameters used and a model will not learn hyperparameters while training. Grid Search CV helps in tuning these hyperparameters and finds the optimal parameters that can give best classification results. In the authors proposed a way to calculate initial weights based on a similarity measure which is based on extended rough set theory and this method is integrated with back propagation learning method to train MLP model. To focus the tasks[9] of high dimensionality using the approaches and technics of the text mining. Where the TF-IDF, weighting method, is the most required methodology to represent the document.

In another study, MLP was used to classify medical records. In this study the authors have worked with 7 different classifiers among which they found MLP gave results with an accuracy of 0.963. For dimensional reduction they used latent semantic analysis.[10] In authors have combined three deep learning models. They combined CNN, MLP and LSTM and proposed a new algorithm which is used to classify documents.[7] While performing preprocessing they obtained two types of vectors, text word vector and text dispersion vector. First the word vector is sent as an input to CNN model to get spatial feature information. This spatial information is sent to LSTM model to get temporal feature information. The other vector is sent to MLP model. The outputs from MLP and LSTM are combined and scaled by sending them to a SoftMax activation function and the final output would be predicted categories. A kind of attributes are tested [11], in sequence with three various neural-network-based routines with increasing complexity.

3. Methodology

3.1 Preprocessing:

R8 dataset is a subset of R21578 dataset. R21578 has 135 classes in the “Topic” category. R8 has the top 8 classes from the “Topic” category. The 8 classes are acq, crude, earn, grain, interest, money-fx, ship, trade. Fig 3.a shows the graph of frequency of documents for each class. ‘earn’ has the highest no. of documents followed by ‘acq’. ‘grain’ class has the least no. of documents i.e. 42. The dataset will contain a significant quantity of unnecessary data and be highly dimensional. It is important to reduce the dimensionality of dataset to lower the complexity of classification process. The first step is handling stop words. Stop words are the most used words everywhere they do not have any weightage in classification, so they are removed. After removing stop words, all words are reduced to their stem form (by removing prefixes, suffixes, or roots. Stemming is performed by using Porter Stemmer. In addition to these, only words with alphabets are chosen removing alphanumeric and numeric words. The resulting documents are represented in matrix form for further processing. Binary matrix and frequency matrix are used to represent the data where binary matrix indicates the presence of words in text files and frequency matrix indicates the count of each feature in a certain document. The binary matrix is obtained from the frequency matrix by replacing non-zero fields with 1. The matrices are converted into data frames where features are columns and text file names are indices. The shape of the data frame is 7126 X 16455.

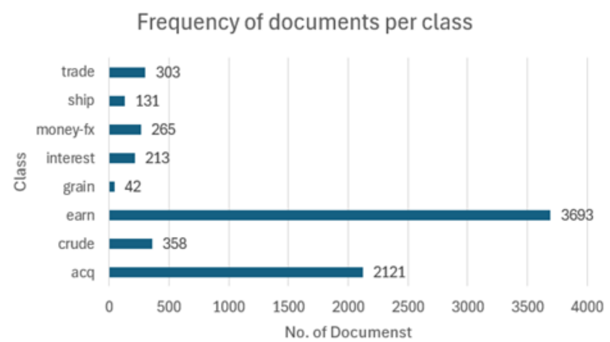

Fig 3.a Frequency of documents per class

Information gain is used for feature selection. The information gain for the dependent variable (‘Label’) of data (R8 dataset) (7126 X 16455) is 3.535548084867161. The information gain of each attribute corresponding to target variable is calculated. Features are sorted in descending order concerning their information gain. The features whose information gain adds up to 90% of the total information gain of all attributes are final features and the remaining features are dropped. The sum of total Information gain of all attributes is 13.9685, 90% of it is 12.5717. 12.5717 is set as threshold and information gain of each attribute is added to the sum until the sum reaches to threshold. The top 5599 attribute's information gain add up to the threshold and only these 5599 features are selected. The shape of the data after feature selection is 7126 X 5599. Only 33.53% of total features are obtained after information gain and they will be used for further process. These 33.53% of attributes have 90% of the data.

3.2 Multi-Layer Perceptron (MLP):

A multi-layer perceptron is a type of Artificial Neural Network that has multiple layers with nodes. Every single neuron is linked to every other neuron in the next layer. The number of neurons in the input layers is equal to the no. of features in the dataset. The hidden layers are present between the input layer and output layer. Neurons in a hidden layer take input from previous layer's neurons and applies a weighted sum adds a bias value, then the result will be passed to the next layers by using an activation function. In Multi-Layer Perceptron, the number of hidden layers and the number of neurons in the network are hyperparameters which means they change according to the complexity of the problem. The number of neurons in the hidden layer are taken as 100, 50, 25, 10. Number of hidden layers are also experimented accordingly. This is the most common approach when building a MLP model.

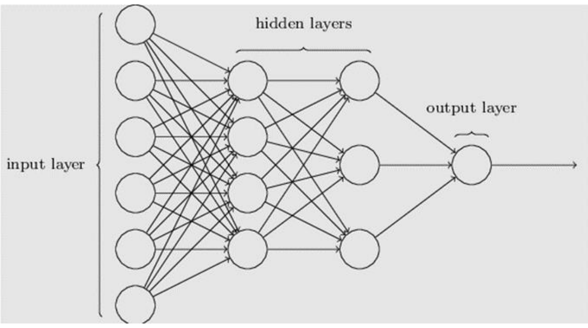

Fig 3.b: MLP architecture

The initialization of weights is an important step which can affect the training process and performance of the model. The weights are assigned randomly to neurons. Some of the algorithms used for weights initialization are:

1. Random initialization
2. Xavier Initialization (Glorot Initialization)
3. He Initialization
4. LeCun initialization

In dense layers (tf.keras.layers.Dense), the default kernel initializer is the "Glorot uniform" initializer. Xavier Initialization (Glorot Initialization) method aims to direct the issues of declining and challenging gradients by scaling the weights built on the no of input and output neurons to each layer. The weights are initialized from a identical or normal distribution with a variance calculated as a function of the number of input and output neurons. This method helps to keep the signal within a reasonable range during training.[12]

While training, the weights of neurons are altered using the optimization algorithm like gradient descent. The aim is to minimize the loss function which calculates the difference between real output and precited output. This process of adjusting the weights is called backpropagation.

3.3 GridSearchCV:

GridSearchCV (Grid Search Cross-Validation) in machine learning is a technique used for tuning hyperparameters. It thoroughly searches in predefined set of hyperparameters. It estimates the model's performance for each hyperparameter using cross validation. The hyperparameters with highest performance are selected. GridSeacrhCV ranks the combination of hyperparameters. The hyperparameters given for the GridSeacrhCV are hidden_layer_sizes [(100,), (100,66)], activation (tanh, relu, logistic), solver(sgd, adam), learning_rate_init(0.01, 0.1). After GridSearchCV the obtained hyperparameters are (100,66) for hidden_layer_sizes, logistic for activation, 0.01 for learning_rate_init and adam for solver. Now, these hyperparameters are used to build a MLP dense model for batch normalization.

3.4 Batch Normalization:

Batch Normalization is a procedure used in neural networks to recover the training speed, strength, and implementation. It normalizes the input of every layer within a mini batch by adjusting the activations. The steps in batch normalization are:

Normalization: For every feature (dimension) in the input, Batch Normalization subtracts the average and divides by the standard deviation of that feature. This transforms the data into unit variance and centres it to zero.[13]

Scaling and Shifting: Once normalization is done, the features are ascended by learnable parameters (gamma) and shifted by another set of learnable parameters (beta). This allows the model to realize the optimum scale and alter each feature, preserving the representation capacity of the network.

Stabilization: Batch Normalization also introduces some noise to each feature, which acts as a shape of regularisation. This helps to prevent overfitting and stabilize the training process, especially in deeper networks.

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Fig 3.c Batch Normalization Formulae

In the above fig 3.c,

m is the no of instances in mini batch

x_i is the input values to next layer.

μ_B is the mean of mini batch.

σ_B is variance of mini batch.

x_i is normalized input.

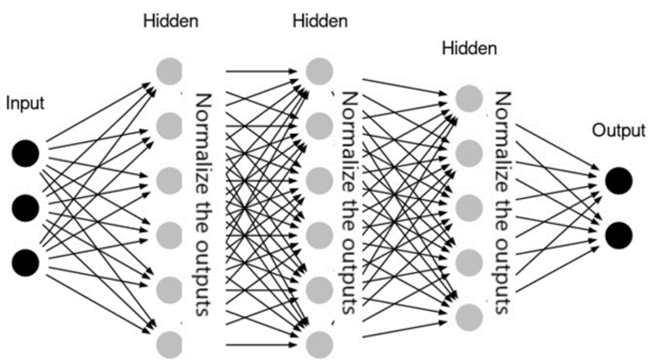

Fig 3.d: Batch Normalization

Here are some benefits of Batch Normalization:

1. Faster convergence
2. Regularization
3. Stabilized training
4. Allows higher learning rates.

3.5 Explanatory MLP:

In Multi-Layer Perceptron, the weights are randomly initialized. They are chosen with 0 mean and variance according to the no. of neurons in the input and output layers, which is the no. of features in the dataset and no. of classes in dependent variable.[14] After initialization using backpropagation, the weights are adjusted accordingly. So, without taking the weights randomly, in our proposed approach the weights are taken from the Linear Regression equation. This is called Explanatory MLP. As the weights are explainable and initialized at the start, it is called Explanatory MLP. The equation has the co-efficients for each feature and intercept similar to weights and bias in MLP. The intercept in the equation is the point on y-axis where all x-coordinates are zero.[15] In Linear Regression, initially the co-efficient are initialized with zeroes (0). Using these co-efficients and values of independent variables(x_i), predicted values are calculated using the equation.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \varepsilon$$

Fig: 3.e Linear regression Equation

X_1, X_2, X_3 are the independent variables.

β_0 is the y-intercept.

β_1, β_2, β_3, ..., are co-efficients of independent variables.

ε is the error term, indicating the disagreement between the predicted and real values of Y.

Now, the residuals are calculated as the difference between y and ŷ. The co-efficients are adjusted iteratively using the minimize the sum of squared residuals.

$$\sum_{i=1}^{N} e_i^{\,2} \text{ Where } e \text{ is the residual.}$$

Optionally, Gradient descent is used to find the exact co-efficients that minimizes the squared sums of residuals. The process goes on iteratively until there is no significant decrease in the value of cost function. Finally, an Equation is generated in the form of *Fig 3.e* with updated co-efficients of independent variables along with an intercept. R8 dataset is of shape (7126, 5599) with 8 output classes. So, the Linear Regression generates 8 equations i.e. each equation for each class. The co-efficient matrix obtained is of shape (8, 5599) along with 8 intercepts. The co-efficeints are used as weights and intercepts are used as bias at MLP. The below *fig 3.f* shows the mean, variance, and standard deviation of the obtained weights.

	Mean	Standard Deviation	Variance
acq	0.001161	0.104668	0.010955
crude	0.002089	0.059489	0.003539
earn	-0.004906	0.096455	0.009304
grain	0.000600	0.037531	0.001409
interest	-0.002889	0.063249	0.004000
money-fx	-0.000128	0.068903	0.004748
ship	0.003995	0.045324	0.002054
trade	0.000078	0.047259	0.002233

Fig: 3.f Mean, Variance and SD of obtained weights

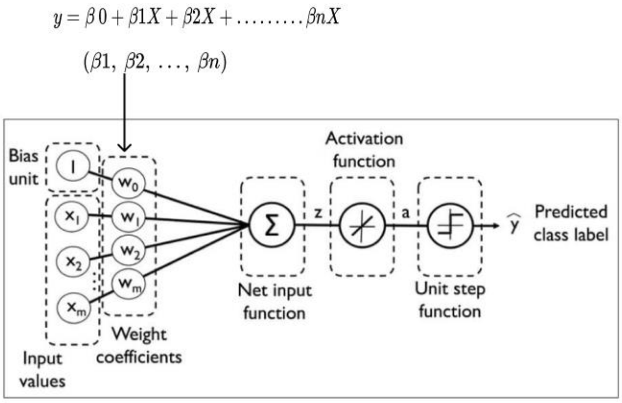

Fig 3.g: Co-efficients of linear regression applied as weights to the neurons.

The mean of obtained co-efficients is close to zero where in MLP, the mean of assumed weights is 0. These weights are initialized to the first hidden layer of MLP. As there are only 8 sets of weights so the no. of neurons in the first layer will be 8. In the succeeding layer the no. of neurons is chosen as 66, 33 and 22. 66 is selected without any reason as it is a hyperparameter. In the succeeding layer the no. of neurons is reduced to half and then reduced by 11. The intercepts are also initialized as the bias for the model. The 8 intercepts are set to 8 neurons in the model.

3.6 Linear Knowledge Transfer:

R8 dataset has 8 classes which means the no. of neurons in the output layer is 8. As, the hyperparameters are no. of hidden layers and no. of neurons, the no. of hidden layers is based on data complexity. If the data is linear no hidden layers are required. If data is less complex 2 to 3 are enough. In this case the hidden layers are selected from 1 to 4. No. of hidden layers are increased and observed that with 2 hidden layers the model gave the best performance. No. of neurons in the subsequent layers should always be decreased. The no. of neurons in the first layer are selected as 64 which is square of 8 and reduced to half in the subsequent layers 32, 16, 8.

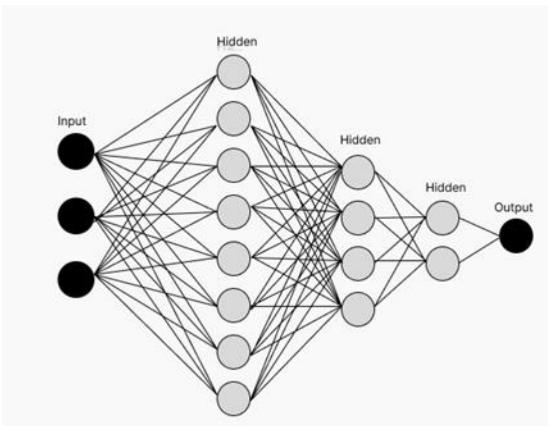

Fig 3.h: Linear Knowledge Transfer

4. Results

One of the metrics used to assess a classification model's performance is Balanced Accuracy. Balanced Accuracy is the average of specificity and sensitivity. The proportion of a model's capacity to identify positive cases is called sensitivity, and the proportion of its capacity to identify negative cases is called specificity. All the performance metrics being considered are their weighted average. The weighted Average multiplies the metrics with ratio of no. of instances to the total instances which is finely suitable for the imbalanced data.

Sensitivity = True Positives / True Positives + False Negatives

Specificity = True Negatives / True Negatives + False Positives

Balanced Accuracy = (Sensitivity + Specificity) / 2

MLP with random weights:

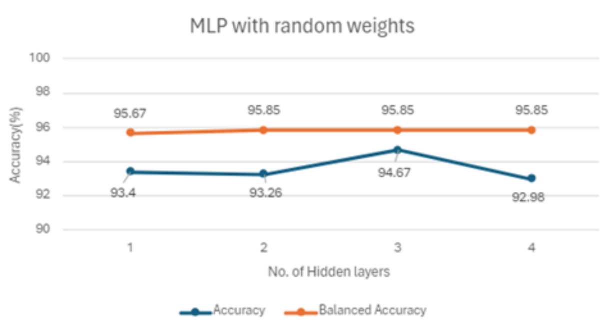

Fig 4.1: Results of MLP with random weights

When MLP is performed with random weights it is found that best results are found when three hidden layers are used. The accuracy and balanced accuracy are found as 94.67 and 95.85 respectively.

MLP with batch normalization:

When MLP is performed with batch normalization best results were found when 2 hidden layers were used. The accuracy and balanced accuracy are observed as 96.35 and 97.66 respectively.

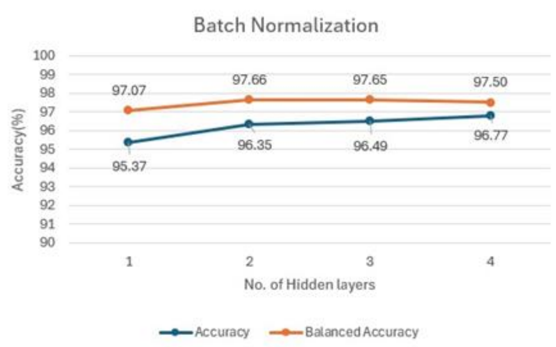

Fig 4.2: Results of MLP with batch normalization

Explanatory MLP:

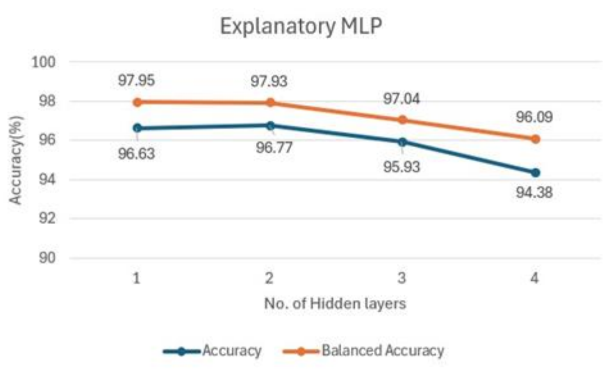

Fig 4.3: Results of Explanatory MLP

Explanatory MLP has given best results out of all the versions of MLP that are used in this study. The best results are observed when 1 hidden layer is used. The accuracy and balanced accuracy are observed as 96.63 and 97.95 respectively.

MLP with linear knowledge transfer:

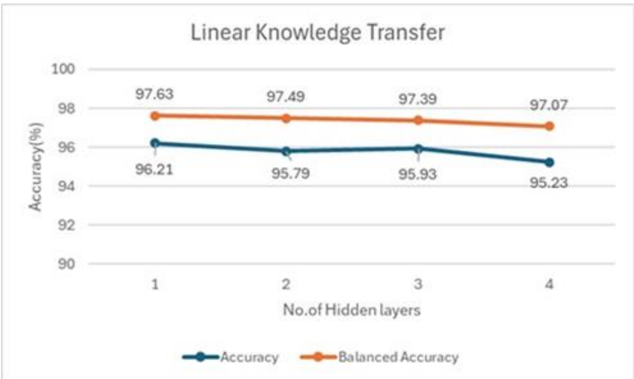

Fig 4.4: Results of MLP with linear knowledge transfer

It is found that when MLP is performed with Linear knowledge transfer, the best results are observed when one hidden layer is used. The accuracy and balanced accuracy are 96.21 and 97.63 respectively.

For Reuters8 dataset the explanatory MLP model has given the most accurate classification results out of the other three models used.

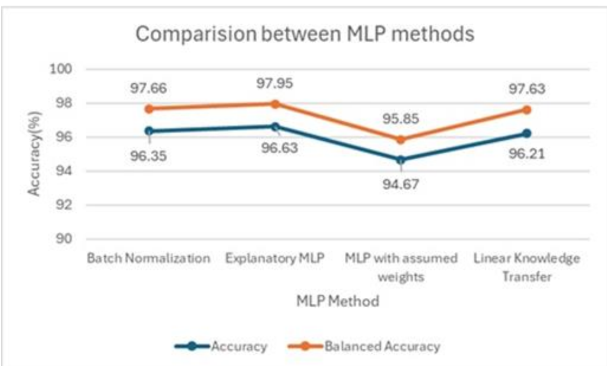

Fig 4.5: Results of all MLP models

The above graph represents the performance of all the MLP models used in this study. Out of all models explanatory MLP has given more accurate results. This is due to the initialization of linear regression weights in the first hidden layer of MLP. Instead of random weights which are unable to explain MLP had the linear regression weights where the data fits into the line minimizing the residual. The weights that are initialized are explainable, so it is called Explanatory MLP. The novel approach of MLP with 2 hidden layers gave the best results than all variations of traditional MLP models. The MLP model of the novel approach has 66 neurons in the first hidden layer and 33 in the second one. Batch Normalization has higher balanced accuracy than linear knowledge transfer. They got almost the same balanced accuracy with a difference of 0.03% and 0.14% difference in accuracy. The traditional and the most common MLP approach with random weights gave the least performance with an accuracy of 94.65 and

balanced accuracy of 95.85%.

R8 dataset is trained and tested with three different algorithms other than MLP. KNN model is developed with two different metrics cosine and Euclidean where cosine gave the best results of 96.81 balanced accuracy and 94.81 accuracy. The frequency matrix is discretized using equal width binning of 10 bins and fed to C4.5 Decision Tree algorithm. Eight models of Logistic Regression are created and trained for each class separately. The dependent variable is transformed as binary for each class respectively. Threshold is chosen as hyperparameter with balanced accuracy as performance metric. One threshold is obtained for assigned to each class and tested. The balanced accuracy observed for Logistic Regression with variable threshold is 98.49% and 97.47% accuracy. Logistic regression tops all the algorithms. Despite initializing the weights using linear regression coefficients, the MLP model came in second out of the four and was unable to outperform Logistic Regression. But Explanatory MLP managed to top among different methods of MLP.

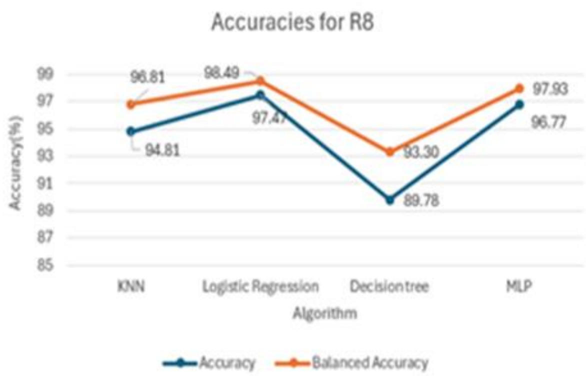

Fig 4.6 Comparison of MLP with other Models

Dataset	Type	Hidden Layers	Accuracy	Balanced Accuracy	f1-score	Precision	Recall	Specificity
R8	Batch Normalization	1	95.37	97.07	0.95	0.95	0.95	0.98
		2	96.35	97.66	0.96	0.96	0.96	0.98
		3	96.49	97.65	0.96	0.97	0.96	0.98
		4	**96.77**	97.50	0.97	0.97	0.97	0.98
	Explanatory MLP	1	96.63	**97.95**	0.97	0.97	0.97	0.99
		2	**96.77**	97.93	0.97	0.97	0.97	0.99
		3	95.93	97.04	0.96	0.96	0.96	0.98
		4	94.38	96.09	0.94	0.94	0.94	0.97
	MLP with random weights	1	93.4	95.67	0.94	0.94	0.93	0.97
		2	93.26	95.85	0.93	0.93	0.93	0.98
		3	94.67	95.85	0.95	0.95	0.95	0.97
		4	92.98	95.85	0.93	0.94	0.93	0.97
	Linear Knowledge Transfer	1	96.21	97.63	0.96	0.96	0.96	0.99
		2	95.79	97.49	0.96	0.96	0.96	0.99
		3	95.93	97.39	0.96	0.96	0.96	0.98
		4	95.23	97.07	0.95	0.95	0.95	0.98

Fig 4.7: Results of all algorithms

5. Conclusion

In this paper we have described how we modified the traditional approaches of MLP and how it improves the accuracy of the model. We performed these methods with Reuters-8 dataset. The algorithms thorough review and validation process is demonstrated by the F1 score, precision, recall, and specificity measures in fig 4.7, which are thoroughly reported alongside accuracy and balanced accuracy. In all cases the specificity (weighted specificity) is observed to be high. Specificity defines the ability to identify true negative cases of each category. In the considered dataset there are 8 classes and in one of the classes the false labels are more than true labels, so the model has always succeeded in predicting the true negative cases. So, the weighted specificity is higher than other metrics. In the increasingly vital fields of machine learning and data mining, classification of data sets is essential. We must be able to collect and understand relevant information from the continually growing amount of data that we produce reliably and efficiently. Text classification has a wide range of application in real life, it can be used in academic papers categorisation, medical records categorisation, sentiment analysis and many more. In future this study can be further extended to work with multiple languages. It can also be used in plagiarism detection.

References

[1] X. Feng, G. Ma, S.-F. Su, C. Huang, M. K. Boswell, and P. Xue, "A multi-layer perceptron approach for accelerated wave forecasting in Lake Michigan," *Ocean Eng.*, vol. 211, p. 107526, Sep. 2020, doi: 10.1016/j.oceaneng.2020.107526.

[2] Nagaraja, U. Boregowda, K. Khatatneh, R. Vangipuram, R. Nuvvusetty, and V. Sravan Kiran, "Similarity Based Feature Transformation for Network Anomaly Detection," *IEEE Access*, vol. 8, pp. 39184–39196, 2020, doi: 10.1109/ACCESS.2020.2975716.

[3] N. Kamath, S. S. Bukhari, and A. Dengel, "Comparative Study between Traditional Machine Learning and Deep Learning Approaches for Text Classification," in *Proceedings of the ACM Symposium on Document Engineering 2018*, Halifax NS Canada: ACM, Aug. 2018, pp. 1–11. doi: 10.1145/3209280.3209526.

[4] M. P. Akhter, Z. Jiangbin, I. R. Naqvi, M. Abdelmajeed, A. Mehmood, and M. T. Sadiq, "Document-Level Text Classification Using Single-Layer Multisize Filters Convolutional Neural Network," *IEEE Access*, vol. 8, pp. 42689–42707, 2020, doi: 10.1109/ACCESS.2020.2976744.

[5] S. Lakhotia and X. Bresson, "An Experimental Comparison of Text Classification Techniques," in *2018 International Conference on Cyberworlds (CW)*, Singapore: IEEE, Oct. 2018, pp. 58–65. doi: 10.1109/CW.2018.00022.

[6] M. Jiang *et al.*, "Text classification based on deep belief network and softmax regression," *Neural Comput. Appl.*, vol. 29, no. 1, pp. 61–70, Jan. 2018, doi: 10.1007/s00521-016-2401-x.

[7] R. Valupadasu and B. R. R. Chunduri, "Automatic Classification of Cardiac Disorders Using MLP Algorithm," in *2019 Prognostics and System Health Management Conference (PHM-Paris)*, Paris, France: IEEE, May 2019, pp. 253–257. doi: 10.1109/PHM-Paris.2019.00050.

I. A. Cahyani, P. I. Ashuri, and C. S. K. Aditya, "Stunting Disease Classification Using Multi-Layer Perceptron Algorithm with GridSearchCV," *Sinkron*, vol. 9, no. 1, pp. 392–401, Jan. 2024, doi: 10.33395/sinkron.v9i1.13245.

[8] M. Bounabi, K. El Moutaouakil, and K. Satori, “A Probabilistic Vector Representation and Neural Network for Text Classification,” in *Big Data, Cloud and Applications*, vol. 872, Y. Tabii, M. Lazaar, M. Al Achhab, and N. Enneya, Eds., Cham: Springer International Publishing, 2018, pp. 343–355. doi: 10.1007/978-3-319-96292-4_27.

[9] M. Zhang, “Applications of Deep Learning in News Text Classification,” *Sci. Program.*, vol. 2021, pp. 1–9, Aug. 2021, doi: 10.1155/2021/6095354.

[10] T. Dönicke, M. Damaschk, and F. Lux, “Multiclass Text Classification on Unbalanced, Sparse and Noisy Data,” in *Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing*, J. Nivre, L. Derczynski, F. Ginter, B. Lindi, S. Oepen, A. Søgaard, and J. Tidemann, Eds., Turku, Finland: Linköping University Electronic Press, Sep. 2019, pp. 58–65. Accessed: Apr. 14, 2024. [Online]. Available: https://aclanthology.org/W19-6207

[11] “Microstrip bandpass filters with ultra-broad rejection band using stepped impedance resonator and high-impedance transformer,” in *IEEE MTT-S International Microwave Symposium Digest, 2005.*, Long Beach, CA, USA: IEEE, 2005, pp. 683–686. doi: 10.1109/MWSYM.2005.1516699.

[12] Y. Han, P. Han, B. Yuan, Z. Zhang, L. Liu, and J. Panneerselvam, “Novel Transformation Deep Learning Model for Electrocardiogram Classification and Arrhythmia Detection using Edge Computing,” *J. Grid Comput.*, vol. 22, no. 1, p. 7, Mar. 2024, doi: 10.1007/s10723-023-09717-3.

[13] K. Sabanci, M. Koklu, and M. F. Unlersen, “Classification of Siirt and Long Type Pistachios (Pistacia vera L.) by Artificial Neural Networks,” *Int. J. Intell. Syst. Appl. Eng.*, vol. 3, no. 2, p. 86, Apr. 2015, doi: 10.18201/ijisae.74573.

[14] Y. Gao, T. Zhu, and X. Xu, “Bone age assessment based on deep convolution neural network incorporated with segmentation,” *Int. J. Comput. Assist. Radiol. Surg.*, vol. 15, no. 12, pp. 1951–1962, Dec. 2020, doi: 10.1007/s11548-020-02266-0.