Low and medium workload classification with contrasting deep learning algorithms

Warren G. Martin

***Abstract*—This report investigates how feature engineering impacts the performance of Convolution Neural Networks (CNN), Long Short-Term Memory (LSTM), and Logistical Regression models for processing EEG data. Each of the models were put through two sets of training sessions; one with raw EEG data and the other using time-domain features (mean, standard deviation, skewness, and kurtosis) and connectivity features(mean upper triangle correlation). Each of the sessions shed light on the significants of feature engineering with Logistic Regression, CNN, and LSTM models, resulting in a lower validation and training loss, higher precision accuracy, and stopping earlier in the training process due to quicker learning. While feature selection did not help the Logistic model preform better much better, the feature selection and implementation process suggests that feature engineering can significantly improve the performance of all models.**

I. Introduction

ELECTROENCEPHALOGRAPHY (EEG) measures activity of the brain; vital for studying cognitive states. To understand the activity of the brain deep learning models are utilized[1]. Convolutional Neural Networks (CNNs), Long Short-Term Memory Networks, and Logistic Regression models, are known for finding patterns in complex data structures. [2],[3][4]

This study implemented Logistic Regression, CNN, and an LSTM model for analyzing EEG data and evaluated the impact of feature engineering on the multiple models. The feature extraction engineering process pulls important information from the raw data and condenses it down into significant features, enhancing the models ability to classify correctly against the data.

The feature engineering process extracted time-domain features, mean, standard deviation, skewness, kurtosis and mean upper triangle correlation. The features were chosen for their ability to capture important signals in the EEG data, EEG$_{r}esearch$

Two experiments were conducted for each model: one with feature-engineered and one with just the raw EEG. The models were tested on a dataset of 62 channels, 512 time points, and 360 samples, with the goal of classifying low and medium workload.

II. Linear Regression from scratch

The first model used to test the raw data was linear Regression, constructed from scratch, which in this case is not utilizing any python library besides Numpy. The model failed to produce any significant results in loss and accuracy,

.

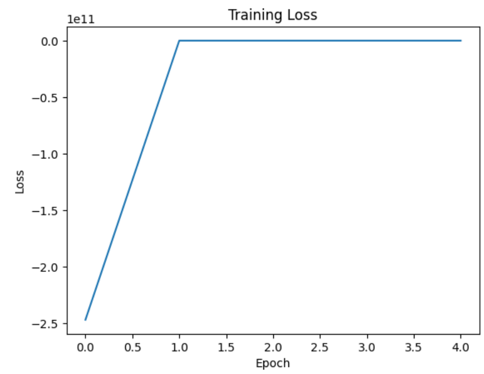

Fig. 1. Logistic model graph showing loss over epochs

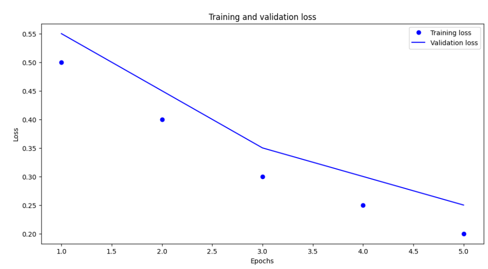

Fig. 2. Logistic model graph showing loss over epochs

leaving the loss around .50 to .40, no matter the changes to the learning rate, batch size, or number of epochs. See fig. 1.

Average Accuracy: 0.5000

After utilizing feature extraction and many iterations in the python code, the logistic model seemed to make some improvements, but upon further sessions, the model refused to generate anything consistent, alternating between .50 and .20. This was either due to the quality of the code or the lack of domain knowledge with deep learning; or both. See Fig. 2,

III. Convolution Neural Network

The CNN was the second model which was tested on the raw data, without feature engineering. See graph In Fig. 3. The Convolution Neural Network started out with a great accuracy right out of the gate, .86, no matter how I changed the parameters, learning rate, epochs, and early stopping learning mechanics.

The CNN with feature engineering made a vast improvement, producing a .94 validation accuracy. while final settling a 78 epochs.

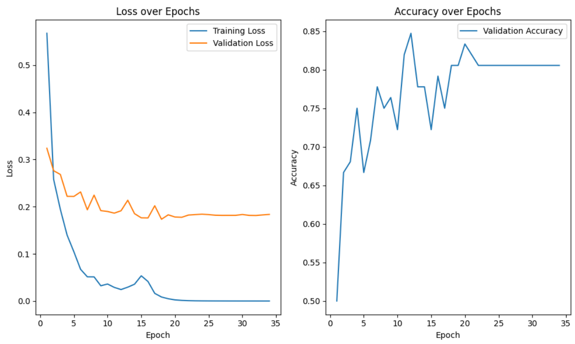

Fig. 3. CNN without Feature engineering, Early stopping was epoch 26 - Final Validation Accuracy: 0.8611

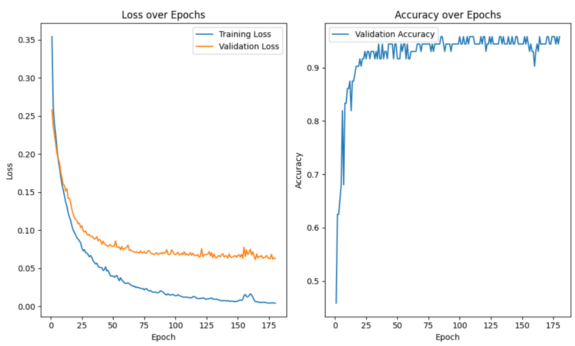

Fig. 4. CNN with feature engineering, Early stopping at epoch 78 CNN with feature Engineering, Final Validation Accuracy: 0.9444

IV. Long Short-Term Memory

The last model tested with raw data was Long Short-Term Memory, a type of RNN. The models accuracy was around .72, see In Fig. 5, a sin

In Fig. 6, a sin

V. Conclusion

The experiments demonstrated the improvements two models can make with feature engineering for CNN and LSTM, and the lack of change with the Logistic Regression Model.

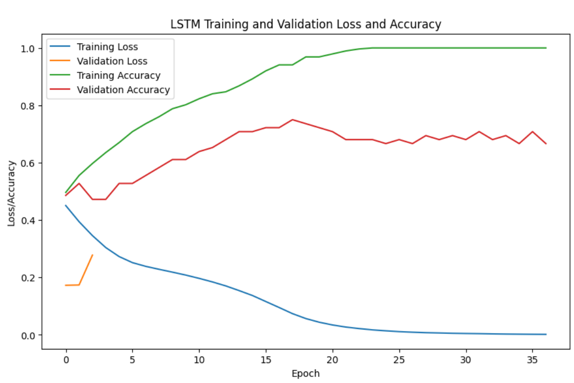

Fig. 5. LSTM with no Feature Engineering

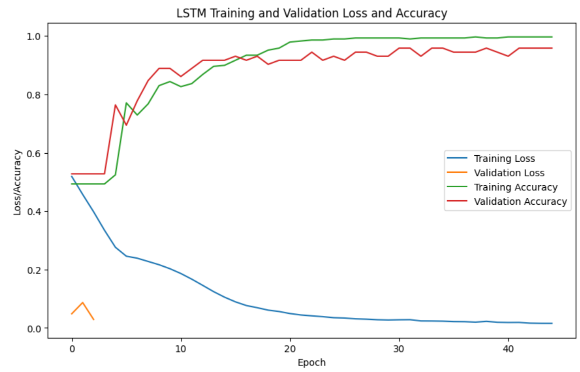

Fig. 6. LSTM with feature engineering utilized

CNN and LSTM models produced lower training and validation losses, with higher validation accuracy, and stopped training way before the last epoch; resulting in an overall better model.

The testing sessions highlight feature engineering as an enhancing solution to performance of deep learning models for EEG data analysis,

References

[1] J. Britton, L. Frey, J. Hopp *et al.*, *Electroencephalography (EEG): An Introductory Text and Atlas of Normal and Abnormal Findings in Adults, Children, and Infants*. Chicago: American Epilepsy Society, 2016. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK390346/

[2] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2016.

[3] A. Subasi and E. Erçelebi, "Classification of eeg signals using neural network and logistic regression," *Computer Methods and Programs in Biomedicine*, vol. 78, no. 2, pp. 87–99, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169260705000246

[4] B. Langmead, C. Trapnell, M. Pop, S. Read, and S. L. Salzberg, "Rna-seq analysis using RNA-Seq data RNA-Seq and EST data," *Genome Biology*, vol. 2, no. 10, p. RESEARCH0002, 2000. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10691701/