Mental Workload Classification

Saurav Thakur [1]

[1]Department of CSEE, University of Essex
st23588@essex.ac.uk

Abstract

Mental Workload is the amount of mental capacity needed to complete any given task. Stress, worry, and other mental tiredness can be lessened with the aid of mental workload measurement. In the domains of education, driving, air traffic control, and healthcare, Mental Workload is a key player. Several methods are investigated in this research to categorize the workload. Depending on how well various topics complete the mental task, the workload is divided into low and medium mental workload categories. Convolutional Neural Networks (CNN) are employed for workload classification. CNN yields an accuracy rate of about 91%.

***Index Terms* - Mental Workload, Convolutional Neural Networks (CNN), Principal Component Analysis (PCA), Fast Fourier Transform (FFT)**

Introduction

In order to identify and address any mental health problems such depression, anxiety, emotional instability, psychological stress, and exhaustion, workload evaluation is vital. These illnesses may have far-reaching effects on people's lives as well as the general health of society. These problems can be successfully addressed or controlled by taking an accurate mental workload measurement and offering the necessary subject matter expertise. Creating reliable models to measure patients' workloads can be beneficial for enhancing the quality of healthcare as a whole.

An extensive body of research has shown a robust relationship between mental workload and electroencephalogram (EEG) signals. These brain signals provide a window into an individual's mental state and may be obtained non-invasively. This makes them essential in a variety of fields that require prolonged cognitive involvement, such as air traffic control, driving, and medicine.

For example, in the aviation sector, using EEG signals to monitor a pilot's mental state can help adapt automated systems for maximum performance and safety. Similarly, by continuously obtaining brain signals and modifying the automated driving system accordingly, real-time monitoring of a driver's mental burden in the automotive industry can greatly lower the probability of traffic accidents. Additionally, the use of electric cars, which frequently just have one paddle for starting, accelerating, and stopping, might reduce mental strain even more and make driving safer and more effective.

The University of Essex provided an EEG dataset that the researchers used in this study to record patients' brain signals while they performed mental activities with low to medium mental works. Fourier Transform was used to extract 62 characteristics from the data, and Convolutional Neural Networks (CNN) were then used to classify the mental workload.

This research is important because it may lead to the development of accurate and dependable models for measuring mental workload, which could have broad effects in a number of fields. Understanding and measuring the cognitive demands that people face allows us to create interventions, modify systems, and put policies into place that support safety, well-being, and optimal performance—benefiting people in the long run as well as society at large.

Related Works

Pušica and Kartali's [3] explores transformer-based models, adversarial inference for learning invariant representations from EEG data, and convolutional neural networks for EEG decoding and visualization. Along with a systematic review on deep learning-based electroencephalography analysis, their extensive study on deep learning for EEG data analytics offers essential insights into the possibilities of these state-of-the-art techniques in deciphering the complex mechanisms of brain activity.

M. Saadati, J. Nelson, and H. Ayaz [4], In their study, Convolutional Neural Networks (CNNs) are applied to the classification of mental workload using functional Near-Infrared Spectroscopy (fNIRS) data. In this arena, our study's original and inventive technology performs noticeably better than conventional techniques like Support Vector Machines (SVMs) and Deep Neural Networks (DNNs).

Based on EEG signals captured during multitasking activities, U. Singh and M. K. Ahirwal [1] categorize mental workload levels (low or high). To achieve accurate classification, the study uses SVM with various kernels to calculate 12 statistical features from the EEG signal dataset. SVM applying a linear kernel over the standard deviation feature produced the best accuracy of 88.88% and an F1-score of 88.31%.

Method

Preprocessing dataset

If we use all of the features, the model will perform poorly because each sample has too many features and also we have a comparatively small number of samples. Therefore, to extract the relevant information from the dataset, we must preprocess the data. Two methods have been used for preprocessing the data.

- Fast Fourier Transform (FFT)
- Principal Component Analysis (PCA)

The dimensionality of the dataset is decreased by FFT and PCA while retaining the important information. As a feature extraction technique, FFT outperformed PCA by a slight amount in terms of generalization. For every second of data, the 62 features in FFT have been retrieved. Additionally, 62 features in PCA were retrieved from the entire two seconds of data. 62 features was choosen because 62 important feature produced the best results.

Experimentation

During the experimentation phase, two different models were evaluated. The perceptron model exhibited overfitting issues, which likely due to the "curse of dimensionality" phenomenon. When the number of features (input dimensions) is excessively high compared to the number of training samples, the feature space becomes sparse. This means that most training samples are scattered across various regions of the high-dimensional feature space, making it challenging for the perceptron to identify an effective hyperplane for class separation. Additionally, with a large number of features, the perceptron can more easily fit the training data, including noise and outliers, leading to overfitting and poor generalization to unseen data. For perceptron based model the learning rate was 0.01. This learning rate was choosen after many experiments done and was performing better than using any other learning rate. Models like Simple RNN and LSTM also tried but since the data had only 2 seconds of feature which means the data did not had long time dependency, so in this case also the model did not generalized well.
In contrast, the convolutional neural network (CNN) model performed well, successfully capturing the essential features from the data and producing accurate results. The dataset consisted of 360 samples, with each sample containing data from 62 electrodes, spanning 2 seconds and comprising 256 data points per electrode. To extract meaningful features from the 512 data points, principal component analysis (PCA) and fast Fourier transform (FFT) techniques were employed. PCA was used to extract 62 essential features from the 512 data points, resulting in a final preprocessed data shape of (360, 1, 62, 62). Additionally, FFT was applied, leading to a data shape of (360, 2, 62, 62).

Because of important features were retrieved form each second, the Fast Fourier Transform (FFT) outperformed Principal Component Analysis (PCA) by a little margin. The two approaches' feature extraction procedures varied in the following ways: The input data was split into two halves, each with 256 features . The top 62 most significant features were taken out of the first half (256 features). Another 62 of the most significant traits were retrieved from the second half (256 features). As a result, FFT was used to extract a total of 124 features—62 from the first half and 62 from the second.
But in PCA, Entire feature in the 512 features of the input data was regarded as a single block. PCA was used to extract the top 62 most significant features from this single block of 512 features.
Compared to PCA, which collected features from the full dataset as a single block, the FFT approach kept more useful information by separating the input data into two halves and extracting key features separately from each half. Because of this, the FFT method was able to capture and maintain significant features more successfully, which resulted in marginally improved generalization performance.

Model Architecture

Now let's explore our final model architecture in brief:
Convolutional Layer 1 (conv1):
A sequential block containing the following layers:

- A 2D convolutional layer with input_shape input channels, 10 output channels, a kernel size of 3x3, a stride of 1x1, and padding of 1.
- A batch normalization layer with 10 features.
- A ReLU activation function.
- A Dropout layer with probability 0.3.
- Another 2D convolutional layer with 10 input channels, 40 output channels, a kernel size of 3x3, a stride of 1x1, and padding of 1.
- A batch normalization layer with 40 features.
- A ReLU activation function.
- A 2D max pooling layer with a kernel size of 2x2 and a stride of 2x2.
- A Dropout layer with probability 0.3.

Convolutional Layer 2 (conv2):
A sequential block containing the following layers:

- A 2D convolutional layer with 40 input channels, 20 output channels, a kernel size of 3x3, a stride of 1x1, and padding of 1.
- A batch normalization layer with 20 features.
- A ReLU activation function.
- A Dropout layer with probability 0.3.
- Another 2D convolutional layer with 20 input channels, 10 output channels, a kernel size of 3x3, a stride of 1x1, and padding of 1.
- A batch normalization layer with 10 features.
- A ReLU activation function.
- A 2D max pooling layer with a kernel size of 2x2 and a stride of 2x2.
- A Dropout layer with probability 0.3.

Classifier (classifier):
A sequential block containing the following layers:

- A flattening layer that flattens the input tensor.
- A fully connected linear layer with 2 output features.

By testing with different hyperparameters and choosing the settings that produced the maximum accuracy, the model architecture was carefully created. Given that this was a binary classification task, the following settings and hyperparameters were selected:

- **Loss Function:** Because the Binary Cross-Entropy loss function works well for binary classification problems, it was employed.
- **Optimizer:** A learning rate of 0.01 was used with the Adam optimizer. Due to its flexible learning rate and proficiency with sparse gradients, Adam is a well-liked option.
- **Dropout:** To help against overfitting, a dropout layer with a probability of 0.3 was added. A regularization technique called dropout forces the surviving neurons to carry the representational weight during training by randomly dropping out (setting to zero) a portion of the neurons.
- **Layers and Kernel Sizes:** A great deal of experimentation was conducted with regard to the quantity of hidden layers, their arrangements, and kernel sizes. The combination that yielded the highest accuracy on the validation set was used to determine the final architecture.

Trying various values and combinations for the number of layers, dropout probability, kernel sizes, and other pertinent hyperparameters was part of the hyperparameter tuning procedure. The goal of this iterative process was to identify the ideal configuration that avoided overfitting while still attaining high accuracy.

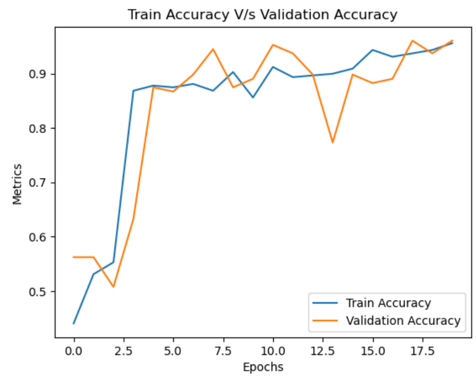
Results and Conclusion

The results achieved are shown in the table below. The results shown below are approximate values which are round off to it's nearest value.

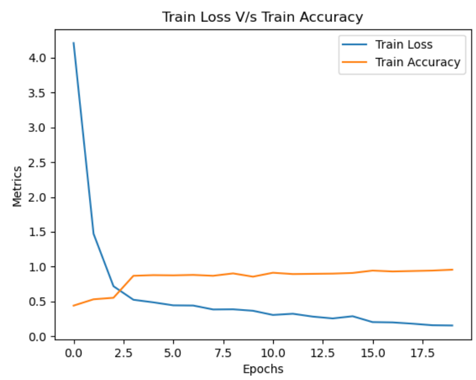
Model	Train Accuracy	Test Accuracy
Perceptron Model	98%	51%
CNN Model using PCA	90%	89%
CNN Model using FFT	95%	91%

Table 1: Performance Results

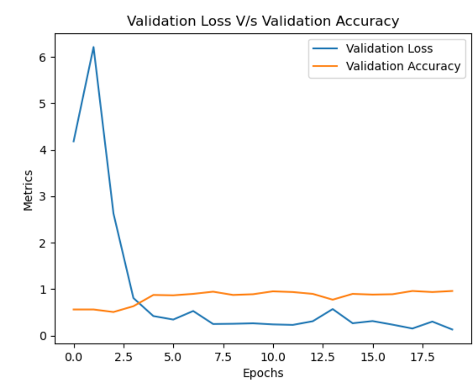
The plots which are shown below are for CNN model using FFT as feature extraction technique, because FFT performed better than using PCA as a feature extraction method.



(a) Train Accuracy vs Validation Accuracy



(b) Train Loss vs Train Accuracy



(c) Validation Loss vs Validation Accuracy

Figure 1: Performance Plots

Future scope

Future scope can be to investigate more on sophisticated deep learning architectures, such as transformers, graph neural networks, or hybrid models, to better classify mental stress from EEG and neuropsychological data. Since we don't have as much data, we can also employ strategies like data augmentation and transfer learning.

References

[1] U. Singh and M. K. Ahirwal, "Mental Workload Classification for Multitasking Test using Electroencephalogram Signal," in Proc. IEEE Int. Conf. Technol. Res. Innov. Betterment Soc. (TRIBES), Raipur, India, 2021, pp. 1-6. doi: 10.1109/TRIBES52498.2021.9751676.

[2] D. T. C. Dolmans, M. Poel, J.-W. J. R. van 't Klooster, and B. P. Veldkamp, "Perceived Mental Workload Classification Using Intermediate Fusion Multimodal Deep Learning," Front. Hum. Neurosci., vol. 14, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnhum.2020.609096. [Accessed: 27-Mar-2024]. doi: 10.3389/fnhum.2020.609096.

[3] M. Pušica, A. Kartali, L. Bojović, I. Gligorijević, J. Jovanović, M. C. Leva, and B. Mijović, "Mental Workload Classification and Tasks Detection in Multitasking: Deep Learning Insights from EEG Study," Brain Sci., vol. 14, no. 2, p. 149, 2024. doi: 10.3390/brainsci14020149.

[4] M. Saadati, J. Nelson, and H. Ayaz, "Mental Workload Classification From Spatial Representation of FNIRS Recordings Using Convolutional Neural Networks," in Proc. IEEE Int. Workshop Mach. Learn. Signal Process. (MLSP), Pittsburgh, PA, USA, 2019, pp. 1-6. doi: 10.1109/MLSP.2019.8918861.