Stress Detection with Convolutional Neural Networks

Technical Report

Coleton Annett
Department of Computer Science
University of Victoria
Victoria, BC V8P 5C2
cannett@uvic.ca

Ty Ellison
Department of Software Engineering
University of Victoria
Victoria, BC V8P 5C2
tyellison@uvic.ca

Abstract

With a growing consensus on the detrimental effects of stress on both physical and psychological well being, having the ability to detect levels of stress in an individual is vital to inform medical treatment and pave the way for more robust early-detection systems. Over recent years, work has been made to progress this field through the use of various deep neural network architectures for both binary stress detection and 3-class emotion classification. This report goes over an attempt to replicate both binary, and 3-class classification techniques utilising CNNs for stress/emotion detection put forward by Li et. al.'s paper "Stress detection using deep neural networks" (Li and Liu [2020]). We demonstrate the training and evaluation of models that agree within 5% of the accuracies presented in the paper, with approximately 94% accuracy in binary classification and 96% accuracy in 3-class emotion detection. Additionally, effort was made to add a forecasting component which would allow the model to pre-emptively classify when a subject is going to feel stressed.

Github repository: github.com/Locrian24/csc421-project-stress-classification

1 Introduction

Li et. al's seminal paper on stress detection utilising deep neural networks highlights the efficacy of such architectures to model the highly non-linear relationships in medical sensor data while also allowing for highly accurate analysis of this data. In particular, they provide methodology for the training and evaluation of two specific types of DNNs: a convolutional neural network

architecture, and a multilayer perceptron architecture. Both are shown to extrapolate off of the widely known WESAD dataset with high accuracy in mood detection and classification. This report focuses specifically on the implementation of the CNN provided by Li et. al. This CNN was trained both in binary stress detection and 3-class emotion classification (between baseline, stress, and amused states). In either case, our model provided comparable accuracies to the original paper within at most 6% difference with the binary classification task.

Dataset generation, model architecture, and certain hyperparameters were drawn from the paper to ensure comparable experiments. This allowed us to not only have the best probability for replication, but also to simultaneously test the results of the original paper. In addition to attempting to replicate the original paper, efforts were made to utilise methods in time-series forecasting to extrapolate the CNN model obtained into a method for stress forecasting.

2 Dataset Generation

Luckily, the dataset used in the original paper is the widely accessible WESAD (WEarable Stress and Affect Dataset) dataset. This allowed for the quick retrieval and augmentation of the data to fit the needs of the model. The dataset consists of sensor data over a period of time in which each of the 15 participants were given various tasks that would elicit different emotional responses.

Two devices: a chest-worn device, and a wrist-worn device were used to record the raw sensor data. Sensor data includes body temperature, respiration, and accelerometer measurements. Specifically for the training of the CNN, as per experiments from the paper, only sensor data reported by the chest-worn device was used to train and evaluate the CNN. This totalled to 8 different sensor data vectors, each representing a different sensor's measurements. These measurements were synchronized to a sampling rate of 700hz, which over the time of the experiments led to a very large number of sample data. To feed this data into our desired model, each participant's sensor data was segmented into 5 second blocks containing each of the 8 sensor vectors.

The enormity of the dataset required the use of the python library WebDataset to store each 3500x8 sample matrix as an individual file with a corresponding label. This allowed for the distribution of the prepared dataset between team members before model evaluation while still maintaining efficiency in training performance compared to the alternative of dealing with the original dataset's format. Consequently, this format of 3500x8 matrix samples necessarily included the sensor data over a time interval, and so the labels of each sample were chosen to be the most frequent label of the sensor data within each block. The shuffled dataset of samples was 70/30 split for training/evaluation respectively.

3 CNN Architecture

The CNN built was based completely off of the model presented in the original paper. Although the architecture presented in the original paper is detailed, instances of details such as filter size of individual convolution blocks differed between figures and so the source of truth was deemed to be what was given in Li and Liu [2020](Table 1) and is what was implemented in the final model. The main architecture involved the passing of each of the 8 sensor data vectors into their own 1D convolutional block, where the outputs of each block is then concatenated into a single vector and passed as input into 3 fully-connected hidden layers with ReLU activation functions between them. Finally, for binary classification, the final layer was a single output node with the sigmoid activation function. This was changed for the 3-class emotion classification task, where 3 output nodes were used in conjunction with the softmax activation function to output the approximated probabilities of each state.

For practical implementation details, PyTorch Lightning was used in training to allow for features such as efficient learning rate approximation and easy GPU acceleration within a Google Colab environment. As mentioned before, WebDataset was used to store datasets generated from the raw available data as POSIX tar archives which allowed for large performance advantages during training due to its efficient streaming data access when coupled with PyTorch's DataLoader object.

Most of the hyperparameters needed for metric tuning for each model were provided by the original paper. This includes specifying a batch size of 40, epoch size of 100. However, some hyperparameters such as learning rate were not included and had to be determined manually. When training the model, we took advantage of PyTorch Lightning's implementation of efficiently approximating good learning rates based on the method proposed by Smith [2015], which helped with fast prototyping of a model that aligns with Li et. al.'s paper in terms of accuracy. Note, as is expected with binary/multi-class classification tasks, binary and categorical cross entropy losses were chosen for each model respectively.

4 CNN Model Evaluation

Keeping with the evaluaton methodology presented in the original paper, the trained models were evaluated against both binary and multiclass classification problems, with each problem having models that were trained with and without accelerometer data included in the training data. This was to provide a one-to-one comparison of their results to a previous paper (Schmidt et al. [2018]), and so this evaluation was implemented to also provide comparisons to Li et. al.'s results. These results are summarised in Table 1, and Table 2.

The trained model with accelerometer data obtained approximately 94.49% on the test dataset in the case of binary classification for the purpose of stress detection, and the model trained without

accelerometer data obtained 93.4% accuracy on the same dataset. In comparison, the original paper obtained 99.8% accuracy on the model with ACC sensor data, and 99.14% without. Although lower than the results of the original paper, our model still performed better than the traditional machine learning approaches that were compared against in the paper, those of Schmidt et. al..

Table 1: Binary-classification task evaluation

	ACC data included	ACC data removed
Our model	94.49%	93.4%
Original model	99.8%	99.14%

In the case of 3-class emotion classification, differentiating between the emotion classes: baseline, stress, and amused, the CNN model (with ACC) achieved 96.6% accuracy on the test set when trained on ACC sensor data, whereas without the ACC sensor data, it only obtained approximately 73.83% accuracy in the same task. In comparison to the results from the original paper, their model achieved 99.55% accuracy for emotion classification with the ACC data, and 93.64% accuracy without. This discrepancy in results without the accelerometer data could be due to differences in learning rates, and other hyperparameters that were not specified in the original paper, or other contributing factors.

Table 2: Multiclass-classification task evaluation

	ACC data included	ACC data removed
Our model	96.6%	73.83%
Original model	99.55%	93.64%

In both cases, our model obtained results within 6% difference in accuracy against the original models when trained with the ACC sensor data. Again, although smaller than the difference in results when trained without the ACC data, this smaller discrepancy is most likely be explained by a difference in chosen hyperparameters. Even still, this performance displays the superiority of deep neural network models in this problem space compared to more traditional machine learning techniques, like those presented in Schmidt et al. [2018].

5 Forecasting Methodology and Results

Time series forecasting is the process of using sequences of historical data to predict future values of a function. Multivariate time series forecasting extends this process from one dimension to many dimensions, which allows for extrapolation to be performed on tensors. In the context of the WESAD dataset, multivariate time series forecasting can be performed on the data to predict future biomarker values. Prior to forecasting, the data must be structured as a matrix such that

each column of the matrix represents a snapshot in time, and each row represents an input variable. However, since the WESAD dataset is so large, it would be impractical to write an entire matrix into memory at one time. Therefore, the data must be reshaped into a tensor such that the tensor slices the matrix into chunks which can fit into RAM.

Choosing the model architecture to perform forecasting is non-trivial. Since the goal is to extrapolate function values into the future, the selected model must be capable of learning temporal patterns in the input data. However, network architectures like those listed above (MLP and CNN) which perform classification fail to robustly capture temporal patterns. An intuitive explanation of this phenomena would be that both MLPs and CNNs learn relationships between the input variables at the current time step, that is to say that there is no way of factoring in previous values of input data to inform the model of historical patterns. Thus, models which contain some sort of cyclical relationship between internal layers, such as Recurrent Neural Networks, are favourable for time series forecasting. The specific type of RNN that was implemented in this project was an Long Short Term Memory or LSTM network - although it should be noted that LSTMs are no longer considered state-of-the-art. Transformer networks now outperform many RNNs in predefined benchmark tests/competitions. However, due to the lack of support for recent transformer networks in the deep learning frameworks used for this project, the pytorch LSTM was selected to perform time series forecasting. Unfortunately, the LSTM could not be trained in time to provide results and is still currently being built and tuned. At this point, the time series forecasting component will be left for future work.

6 Conclusion

Li et. al.'s paper "Stress detection using deep neural networks" (Li and Liu [2020]) presents various architectures for DNN approaches for the analysis of the WESAD dataset for the purpose of emotion classification and detection. By attempting to replicate the paper based on the data and information provided within, we managed to obtain a comparable model when trained on the same dataset. Due to the nature of the data needed to adequately evaluate the model in real-life conditions, there is still more that can be done in regard to robust forecasting and emotion modeling, however these methods present the supremacy of deep neural networks within this problem space. More specifically, the success of DNNs may be attributed to two factors, the first being the temporal resolution of the WESAD dataset. Since the sensors used to collect the WESAD data sample anywhere from 4 to 700 times per second, a lot of data can be generated and fed into a network - allowing for higher ceilings in terms of model accuracy. Secondly, the functions which describe the biomarkers being measured are periodic and asymptotic, indicating that a less complex model should be capable of capturing the dynamics of this system. This also implies that long term forecasting should be more feasible since the functions are constrained to fall within a set range, and that violations of periodicity and/or smoothness may indicate serious health problems.

References

Russell Li and Zhandong Liu. Stress detection using deep neural networks. *BMC Medical Informatics and Decision Making*, 20, 12 2020. doi:10.1186/s12911-020-01299-4.

Leslie N. Smith. No more pesky learning rate guessing games. *CoRR*, abs/1506.01186, 2015. URL `http://arxiv.org/abs/1506.01186`.

Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, and Kristof Van Laerhoven. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, ICMI '18, page 400–408, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356923. doi:10.1145/3242969.3242985. URL `https://doi.org/10.1145/3242969.3242985`.