CS230

Brain Maximum Principal Strain (MPS) Evaluation in Head Impact

Mabel Jiang
Mechanical Engineering
Stanford University
jyh1998@stanford.edu

Mingruo Shen
Civil and Environmental Engineering
Stanford University
mingruos@stanford.edu

Shunyao Xu
Electrical Engineering
Stanford University
shunyaox@stanford.edu

Abstract

As a necessary demand of athletes and medical professionals to recognize potential damage region in brain and to diagnose Mild Traumatic Brain Injury (mTBI) in time, a fast and practical mTBI detection and prediction method is in need to be developed. In this project, 8 dimensions of kinematic data are used to train, validate, and test on the 1-dimensional Convolutional Neural Network (1D CNN) models to predict for both Maximum Principal Strain (MPS) and Maximum Principal Strain Rate (MPSR). Models developed from this project have the potential to detect brain deformation from contact sports practically.

1 Introduction

This project is mainly focused on the development of a reliable model to detect Mild Traumatic Brain Injury (mTBI) in head impact. Commonly seen in contact sports games, mTBI usually comes with symptoms such as headache, fatigue, or even temporary loss of consciousness [1]. Therefore, it is of necessity for athletes and medical professionals to understand the potential brain damage it might cause at specific brain regions, to monitor real-time mTBI, and to customize specific suggestions during daily exercises and games. Accordingly, a fast and practical method of mTBI detection and prediction is in need.

Based on the previous CS230 project by Xianghao Zhan and Yiheng Li [2], our project aims to develop a more accurate deep learning model for head impact detection using larger datasets, preprocessed inputs, and different algorithms. The input of our model are a series of kinematic data of head impact. We use a 1-dimensional Convolutional Neural Network (1D CNN) model to output predicted Maximum Principal Strain (MPS) and Maximum Principal Strain Rate (MPSR), which are indicators for evaluating the brain deformation, and evaluate the model and the prediction by Root Mean Squared Error (RMSE).

2 Related work

O'Keeffe, Eoin et al.[3] had implemented KTH finite element(FE) model in estimation of brain tissue deformation. However, a FE model's relatively high computational cost does not meet the requirement of real time monitoring of this project. Zhan, X et al.[4] had proved that deep learning model have much lower computational cost than the traditional FE modeling method. They realized real-time brain deformation monitoring and prediction of brain MPS by applying a five layers DNN pre-computed model. While selecting inputs, they did not include linear acceleration since it was shown that linear acceleration is not related with brain strain. Similarly, for our model, we only included 8 channels of data, angular velocity and angular acceleration out of 12 channels. Liu, L et al.[5] had proposed two robust deep learning algorithms regarding herbal medicine origins classification, one 1D CNN model and one multivariate time pooling model. Two dimensionality reduction methods, PCA(Principal Component Analysis) and LDA(Linear Discriminant Analysis) were exploited in their modeling. For our model, we chose not to reduce the dimension because it potentially leads to some amount of data loss and we decided to keep all the components to achieve better results. Shaoju Wu et al.[6] also implemented a CNN model to predict the brain MPS. Unlike other models using the velocity profiles data as input, the data were instead transformed to a 2D image which then was input into CNN. This creation is inspiring with respect to the monitoring and visualization of brain deformation. Regarding the architecture of 1D CNN model, Ullah et al.[7] had proposed a pyramidal 1D CNN model, where lower level layers have larger number of filters than the higher level layers. The similar 1D CNN architecture was implemented in our model and will be explained in details in the later Experiments/Results/Discussion section.

3 Dataset and Features

The data we used in our deep learning model, collected by Stanford Instrumented Mouthguard, is the kinematic data of head impact generated from real sport games, including College football matches (CF) and Mixed Martial Arts (MMA) [8, 9, 10, 11]. A Head Modeling (HM) dataset, consisting of simulated data, is also exploited to make up for the lack of real data. The kinematic dataset, consisting of angular velocity and angular acceleration data in three dimensions, is measured in 6 degrees of freedom. Together with the magnitudes, kinematic data input are 8 dimensional.

	Head Modeling (HM)	College Football (CF)	Mixed Martial Arts (MMA)
Number of Data	12,780	302	457
Time Frame (ms)	70 → 200	100 → 200	200
Label Size	4,124	4,124	4,124

Table 1: Information and Dimension of the Dataset

The Table 1 above shows the information and dimensionality of the datasets we use. The time frame in each dataset is different and the longest time frame is 200 ms. We decided to standardize all dataset's time scale to 200 ms as the longest time frame by putting the data in the middle and concatenating zeros before and after the available data for datasets with shorter time frames. For HM data, with an initial shape of (12780, 8, 70), the training data was standardized to (12780, 8, 200), and then flattened to (12780, 1600). The Figure 1 below shows an example of input data in the scaled time frame. There are 12780 samples in total. The training, dev, and test sets are split with fraction of 8 : 1 : 1. There are 10224 samples of data in training set, 1278 samples of data in dev set, and 1278 samples of data in test set. To make our training faster, our data is normalized. Figure 1 below shows an example of flattened and normalized data sample from HM.

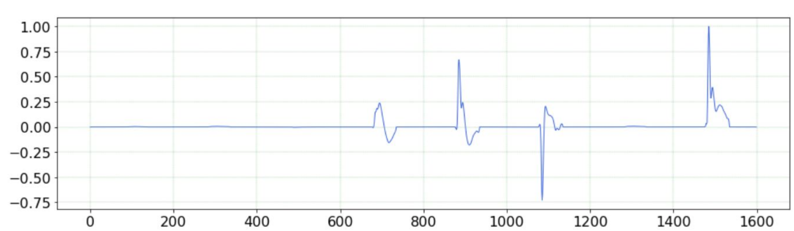

Figure 1: Preprocessed Data in 1600 ms Time Frame

In order to account for the difference between HM and the real datasets, which comes from the time frame difference of the datasets and the heterogeneity of College Football and Mixed Martial Arts, we split both CF and MMA datasets by half, concatenate 50 percent of real sports data with the dev set, and leave 50 percent of real data as test set. Therefore, in total, there are 10224 samples of data in the training set, 1657 samples of data in the dev set, and three test sets consists of 1278 HM data, 151 CF data, and 229 MMA data.

The outputs of our model are the MPS and MPSR, which are indicators for evaluating the brain deformation. With the shape of (4124,1), it represents the 4124 different regions in the brain. We chose not to reduce the dimensionality because it potentially leads to some amount of data loss and we decided to keep all the components to achieve better results.

4 Methods

As the past project used DNN and LSTM models, we would like to approach this problem with a different method. In this project, we focused on the implementation of 1 Dimensional Convolutional Neural Network, developed under Tensorflow and Keras framework. With 1D signal as input, the computational cost would be lower than 2D CNN and would be beneficial for this project as we have over ten thousands data samples to go over hundreds of epochs.

For the baseline model, with an input shape of (1600,1), the first two layers are convolutional layers with 64 filters and 2x2 kernel size. The activation function is ReLU. To downsample the feature map, the third layer is a MaxPooling layer with 2x2 kernel size. Then the feature was flattened, and there are two fully connected layers at the end. The first fully connected layer has 1000 neurons, and the second one has 100 neurons. The output shape is (4124,1). The batch size is 128, the optimizer is Adam with default learning rate and decay, and the epoch is 100.

Some equations associated with the algorithms: Rectified Linear Unit (ReLu) has an equation of $y = max(0, x)$; the mathematical formulas for Adam (simplified):

$$v = \beta_1 v - (1 - \beta_1)\nabla\theta$$

$$s = \beta_2 s - (1 - \beta_2)\nabla\theta^2$$

$$\theta = \theta - \alpha v/sqrts$$

Because we are addressing a regression problem instead of classification, we would like to keep track of the accuracy by error between the real data and prediction. Therefore, the loss function we use in the models is Mean Squared Error, which is defined as the equation $MSE = 1/n * \sum(y - ypred)^2$. The performance metric we use is Root Mean Squared Error, which is better compared to MSE as it penalizes more on larger errors. The mathematical equation of RMSE is:

$$RMSE = sqrt1/n * \sum(y - ypred)^2$$

We will further discuss the architecture and algorithms of the models we use in the next section.

5 Experiments/Results/Discussion

Apart from dimensionality, 1D CNN follows the same methodology as 2D CNN. Figure 2 shows the architecture of our 1D CNN model. The input of the model is first processed by two convolutional layers. Each of these two layers has 64 filters and 2x1 kernel size. Activation function ReLU is used in both of these layers. After these two convolutional layers, the model uses a Maxpooling layer to reduce the number of features and select the dominant ones. The Maxpooling layer also uses a kernel of 2x1. A dropout layer with dropout rate 0.5 is applied to prevent overfitting. After that, the tensor is flattened and passed to two fully connected layers. The first fully connected layer has 400 neurons, and the second one has 100 neurons. A output of size 4124x1 is generated by propagating through all these layers.

Using the architecture described above, the 1D CNN model is constructed. A default learning rate of 0.001 is implemented because we don't expect the model to learn too faster. With a huge input and output size, a relatively lower learning rate can ensure a stable learning process and prevent the model from learning a sub-optimal set of weights too fast. Our primary metrics is the root mean square error. Accuracy is not applicable in this case since this is not a classification problem. We want the output of our model to get as close as possible to the expected outcome, but in many cases it will never be exactly the output we want. By tuning the number of epoch and mini-batch size, we decide to train the model for 100 epoch and use 128 mini-batch size.

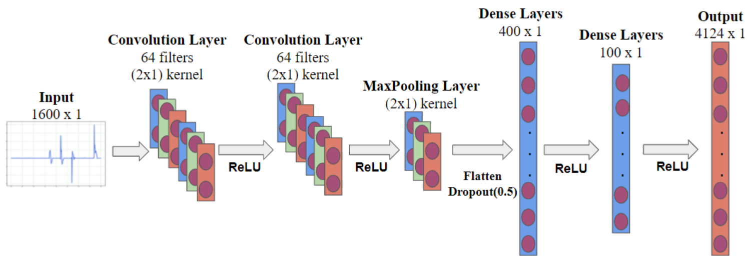

Figure 2: 1D-CNN Architecture

Using Maximum Principle Strain Rate (MPSR) as label, the 1D CNN model was trained. Figure 3 (left) presents the training result as a plot of the root mean square error for each epoch. From the figure, it's clear that the model tends to overfit to our training set because the gap between training set error and dev set error is getting larger and larger.

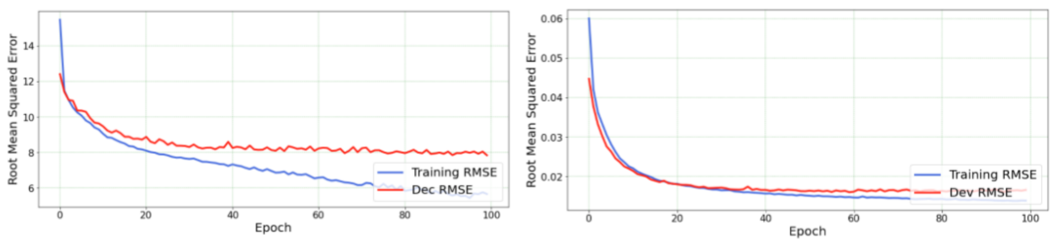

Figure 3: Training Result - Model 1 (MPSR Label)   Training Result - Model 1 (MPS Label)

The same model was trained again Using Maximum Principle Strain (MPS) as label. Figure 3 (right) shows the training outcome. Compared with Figure 3 (left), the overfitting problem is significantly smaller.

As mentioned in the Related Work Section, the architecture of a pyramidal 1D CNN would reduce the numbers of learnable parameters and hence reduce the risk of overfitting [7]. Its architecture is shown in Figure 4. To further mitigate the overfitting, we changed the architecture of our model. Batch normalization was added to each convolutional layer, and the number of filters for each layer is slightly modified to achieve a pyramidal structure. As shown in Figure 5, this model has a relatively better result than the previous model in terms of variance, but the bias is getting a little bit worse.

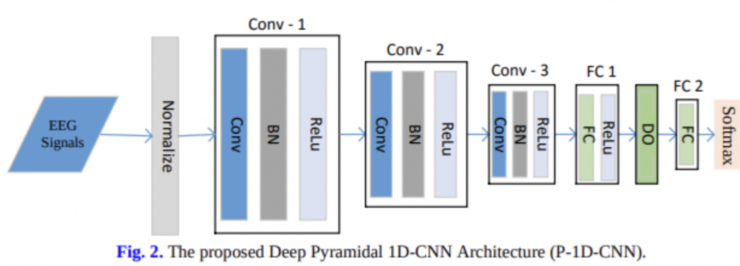

Figure 4: Pyramidal 1D CNN Architecture [7] - Model 2

After the models were trained, the performance of these models were evaluated by predicting on test set. Test set can help us make sure that the model is good enough for unseen data. Since our test set contains data from Head Modeling, College Football (CF), and Mixed Martial Arts (MMA), we use our models to predict on them separately in order to illustrate the performance of our model when predicting on difference types of data. The table below shows the results. As presented in the table, the models generally achieves a low error when predicting on HM data because HM data takes a larger portion of our training set. Moreover, the table shows the inference speed for each model.

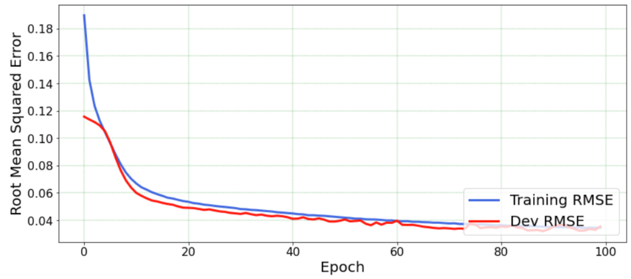

Figure 5: Training Result - Model 2

Inference speed is calculated by predicting on 1000 samples and then divides the time it took by 1000. It turns out to be a very important metrics because we want our model to be deployed in real-time. Based on the table, all of the three models achieves a pretty good inference speed.

	Model 1 (MPS)	Model 2
Root Mean Square Error (HM)	0.013983	0.035186
Root Mean Square Error (CF)	0.072358	0.073391
Root Mean Square Error (MMA)	0.056512	0.076651
Inference Speed	0.032226 s	0.024455 s

Table 2: Prediction on Test Set and Inference Speed

6 Conclusion/Future Work

As discussed in the previous section, the Model 1 (MPSR label) and Model 1 (MPS label) result in great performance on the HM dataset. Accounting for different distributions and heterogeneity of different sports, it is reasonable that the RMSE of CF and MMA are relatively higher than the RMSE of HM. From Table 2, the errors of CF and MMA are also relatively small and show good performance. The reason Model 1 works better than Model 2 is that we purposefully adjusted the algorithm, including parameters and layers, to reach a better performance, so the algorithm would be very suitable in this project, while the Model 2 architecture comes from another paper with the topic of Epilepsy Detection Using EEG Brain Signals [7]. As the pyramidal architecture is also used in brain signal prediction, the performance is also relatively good, especially for CF dataset.

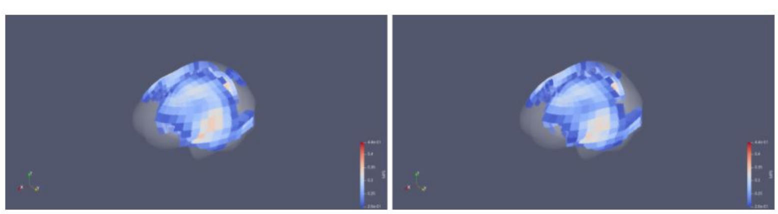

Figure 6: 3D visualization of real brain impact   3D visualization of predicted brain impact

The Figure 6 above shows an example of 3D visualization of real brain impact (left) and predicted brain impact (right) from Model 1, which clearly show that the predicted brain impact is very close to the real brain impact. This indicates this model has the potential to detect brain deformation from contact sports practically.

The future work would attempt to use Domain Regularized Component Analysis (DRCA) to create a subspace projection in order to reduce drifts and achieve more accurate and reliable results from different sports dataset [12].

7 Contributions

Mabel Jiang: I did the literature review, looked into possible model related to brain field, and contributed to the coding of the second 1D CNN model. Additionally, I wrote section 1, section 2, and model 2 part of section 5 of the final report.

Mingruo Shen: I obtained the data, contributed to the coding part of Model 1 (MPS) and (MPSR), and decided the loss and performance metrics. Moreover, I wrote the abstract, section 3, 4, and 6 of the final project.

Shunyao Xu: I was responsible for designing and training the 1D CNN models. I also helped with data processing. Moreover, I assisted the result analysis. For example, I evaluated the performance of the models on test set and generated figures to use in the report. Additionally, I wrote the section 5 of the final report.

8 Code

https://github.com/shunyaoxu/Stanford$_{C}S230_{Project}$

References

[1] McInnes K, Friesen CL, MacKenzie DE, Westwood DA, Boe SG (2017) Mild Traumatic Brain Injury (mTBI) and chronic cognitive impairment: A scoping review. PLoS ONE 12(4): e0174847. https://doi.org/10.1371/journal.pone.0174847

[2] Zhan X., Li Y. (2020). Fast Brain Strain Evaluation in Head Impact.

[3] O'Keeffe, Eoin et al. "Dynamic Blood-Brain Barrier Regulation in Mild Traumatic Brain Injury." Journal of neurotrauma vol. 37,2 (2020): 347-356. doi:10.1089/neu.2019.6483

[4] Zhan, Xianghao et al. "Rapid Estimation of Entire Brain Strain Using Deep Learning Models." IEEE transactions on bio-medical engineering vol. 68,11 (2021): 3424-3434. doi:10.1109/TBME.2021.3073380

[5] Liu, L., Zhan, X., Duan, Z., Wu, Y., Wu, R., Guan, X., Wang, Z., Wang, Y., Li, G. (2021). Classifying herbal medicine origins by temporal and spectral data mining of Electronic Nose. arXiv.org. https://arxiv.org/abs/2104.06640.

[6]Shaoju Wu, Wei Zhao, Kianoosh Ghazi, and Songbai Ji. (2019). Convolutional neural network for efficient estimation of regional brain strains. Scientific Reports. 9. 10.1038/s41598-019-53551-1.

[7] Ullah, Ihsan, et al. (2018). An Automated System for Epilepsy Detection Using EEG Brain Signals Based on Deep Learning Approach. arXiv.org. https://arxiv.org/abs/1801.05412.

[8] Liu, Y., Domel, A. G., Cecchi, N. J., Rice, E., Callan, A. A., Raymond, S. J., Zhou, Z., Zhan, X., Zeineh, M., Grant, G., amp; Camarillo, D. B. (2021). Time window of head impact kinematics measurement for calculation of brain strain and strain rate in American football. arXiv.org. https://arxiv.org/abs/2102.05728.

[9] Camarillo, D. B., Shull, P. B., Mattson, J., Shultz, R.,amp; Garza, D. (2013). An instrumented mouthguard for measuring linear and angular head impact kinematics in American football. Annals of Biomedical Engineering.

[10] Liu, Y., Domel, A. G., Yousefsani, S. A., Kondic, J., Grant, G., Zeineh, M., amp; Camarillo, D. B. (2020). Validation and comparison of instrumented mouthguards for measuring head kinematics and assessing brain deformation in football impacts. arXiv.org. https://arxiv.org/abs/2008.01903.

[11] Hernandez, F., Wu, L. C., Yip, M. C., Laksari, K., Hoffman, A. R., Lopez, J. R., Grant, G. A., Kleiven, S., Camarillo, D. B. (2015). Six Degree-of-Freedom Measurements of Human Mild Traumatic Brain Injury. Annals of biomedical engineering, 43(8), 1918–1934. https://doi.org/10.1007/s10439-014-1212-4

[12] Zhang, L., Liu, Y., He, Z., Liu, J., Deng, P., Zhou, X. (2017). Anti-drift in E-nose: A subspace projection approach with drift reduction. Sensors and Actuators B: Chemical, 253, 407–417. https://doi.org/10.1016/j.snb.2017.06.156

Code and Libraries

Tensorflow: Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey

Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng.(2015) TensorFlow: Large-scale machine learning on heterogeneous systems, Software available from tensorflow.org.

Keras: Chollet, F., others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras