ANALYSIS AND MODELING FOR PHYSIOLOGICAL TIME-SERIES DATA

Samuel Cowin[⋆†]*, A59004897*

[⋆]University of California San Diego, La Jolla, CA 92093-0238,
[†]Department of Electrical and Computer Engineering

ABSTRACT

As analyzing physiological time-series data becomes more important, mechanisms for enabling research without being medical domain experts is essential. Commonly, physiological data requires a different curated approach to extract out relevant features depending on the problem, the subjects, and the amount of data collected. Statistical approaches that are used in other machine learning problems to automate out some of the feature extraction difficulty do not always translate well to physiological data. This project aims to shed light on advancements in statistical deep learning that capture some of the autonomy back to allow exploration on datasets where the investigator is not a subject matter expert.

1. INTRODUCTION

Investigation of time-series physiological data is becoming more of a prevalent domain as sensor arrays that can be placed external to the body achieve higher fidelity, become cheaper, and computational power becomes more ubiquitous. Additionally, high profile companies such as Neuralink through their brain-machine interface or Apple through the Apple Watch demonstrate that there are current opportunities for commercial success when tackling problems in this domain.

Typically, how these problems are handled is that there are medical experts (i.e. neuroscientist for EEG data) or specialized physiological data experts within the team that can guide the engineers on what features to utilize, how the data needs to be handled, and general processing tips. Some common approaches for feature selection are multifractal analysis [1], power spectral density, entropy, or wavelet transform [2] among others. The common theme across these methods is that they are highly problem specific and require specialized knowledge. Variations and bias in feature selection from one problem to the next can drastically change classification performance.

Along these lines, the data analysis and modeling experts that are often deployed to investigate these problems are not medical experts, and thus lack the knowledge to select curated features from the physiological data. For this reason, this project will utilize statistical approaches that allow for meaningful inference without needing the domain specific background.

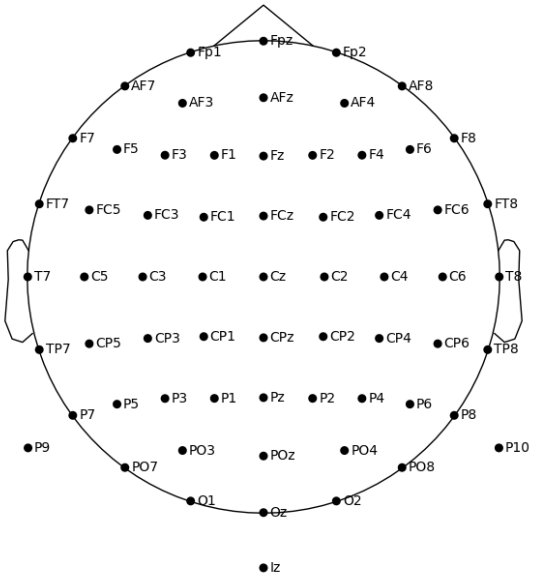

Fig. 1. 64-channel EEG cap visualization

2. DATASET

Given the scope of the project and the lack of medical expertise, a sample EEG dataset from MNE [3] is used to focus the project on statistical methods and not dataset construction. The portion of the dataset used in this project was the 60-channel EEG cap from the Neuromag Vectorview system to acquire data at MGH/HMS/MIT Athinoula A. Martinos Center for Biomedical Imaging [3]. To get a sense of what the data captured was, an example of how a sensor array sits on the head is seen in Figure-1 which was output from the MNE API for a similar 64-channel cap.

From an engineering perspective, this means there are 60 sensor channels (features) that have time-series data for each feature, and this is for each event (stimulus) that is tracked. There were 376 events that were tracked, and the

ECE 225A: Probability and Statistics for Data Science, Fall 2021

Statistic	Sensor1	Sensor 2	Sensor 3
Mean	0.00018	0.00066	0.00081
Median	0.00015	0.00046	0.00057
Std	0.085	0.084	0.085
Min	-0.550	-0.577	-0.519
Max	0.8759	0.8609	0.8538

Table 1. Sample statistics for first three sensors

full dataset size is 376x41700, or when broken into channels, 376x60x151. In addition to raw EEG data, the response that was asked of the participants was regarding stimuli that was either visually sent to the left or right of their perceptual field or auditory sent to their left or right ear. These event tags can thus be used as labels for a classification task of predicting the event from the EEG data and have a related shape of 376x1.

3. DATA EXPLORATION

As mentioned prior, processing physiological data without being an expert can be a challenge, but there are some initial processing steps that are more ubiquitous and follow fundamental engineering principles. First, is that the data across channels is highly variant in terms of the range of the values. The complexity in this analysis can be simplified by getting the z-scores of the values in each feature column to standardize the data:

$$Z = \frac{X - \mu}{\sigma}$$

Additionally, EEG data tends to be very small in magnitude, and thus this can confuse various analysis systems. For this reason, scaling the data by 1000 or similar can help models not confuse the signaling with noise. Finally, for the preliminary investigation below into sensor importance, the events were combined together so that there was a dataset of 60 channels with a time-series of 43488 samples each. Sample statistics are included in Table-1 for the first few sensors.

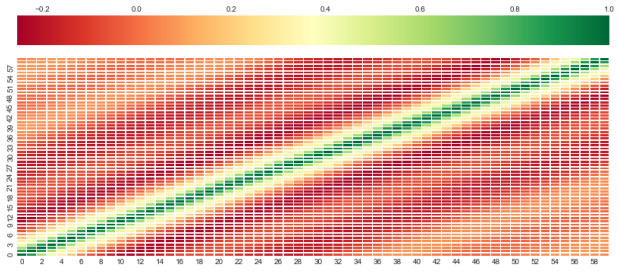

Fig. 2. Correlation Matrix for 60-channel EEG Sensors

Now that the raw data is in a form that is usable, some initial exploration into the importance of all the sensors was done. The normalized correlation matrix, Figure-2, was computed to get an idea if there were any redundancies in the data per the equation below:

$$\sigma_{XY} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

While understandably difficult to read, the point of the correlation was to see if any of the sensors were related as to be expected. This turns out to be the case, since there is very strong correlation among sensors that are next to each other and very weak correlation otherwise. This makes sense as the brain activity in nearby sensors should be related as they seemingly will pickup similar signals. This is seen by the green values in the correlation matrix indicating strong correlation where groups of sensors next to each other all have green indication.

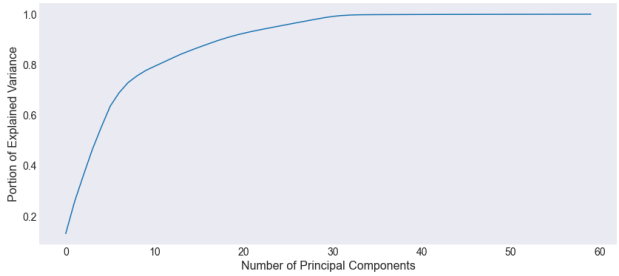

Fig. 3. Principal Components Analysis of 60-Channel EEG Sensors

Ultimately, this does not help significantly in understanding how the sensors play into the total dynamic of the recordings and if any should potentially be left out. Another common method that is used in both physiological data exploration and engineering in general is Principal Components Analysis. This takes the covariance matrix that was just computed before normalization and uses it as a mechanism to extract out the features that contribute the most to the overall dataset. This is done by extraction of the maximum eigenvalues. From this analysis, it seems that dropping a significant portion of the sensors would still allow the data to explain the majority of the variance per Figure-3. This is the approach that would be taken by someone unfamiliar with the data itself, and the reduction of the dimensionality of the data in this manner will be shown to be problematic later when we discuss modeling.

4. REPRESENTATION LEARNING

Given that the statistical approaches mentioned thus far in Section-3 are limited in how they can be used to understand the data, a more powerful deep learning based statistical method will be explored next. This comes in the form of a Variational Autoencoder for representation learning of the data. At a high level, what this method aims to do is reconstruct the provided data from a lower dimensional space that is learned through neural networks and statistical modeling.

The focus of this project was on the statistical modeling portion of the discussion, but a brief description of a simple autoencoder is necessary to understand the enhancements. The main architecture can be broken into a few layers - fully-connected dense layers and activation functions. The flow is as follows:

- Initial input of the data into the network
- Fully-connected layer that takes the input data and through additions and multiplications reduces the dimension to a lower degree
- Pass through an activation function that introduces nonlinearity
- This lower dimensional space is the latent space
- Pass the latent space through a fully-connected layer that increases the dimensionality back to the original
- Introduce another nonlinearity in order to get the output into a state that is compatible with a loss function to optimize the network
- Compute the loss function and optimize weights throughout network to improve performance

This model works well for simplistic problems, but the latent space is still relatively uninformative which was the problem with the previous statistical methods we showed. In fact, if the activation functions are linear, this architecture performs similarly to the PCA model shown prior. For this reason, this project improves on the latent space representation learning method in a traditional autoencoder by introducing the more sophisticated statistical modeling of the variational autoencoder.

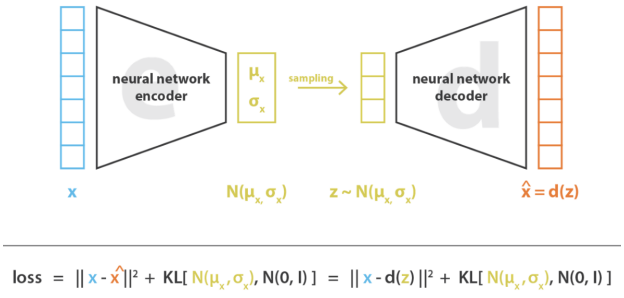

Fig. 4. Architecture of VAE with resampling trick [4]

The main differences in the variational autoencoder, shown in Figure-4, boils down to how the latent space is constructed and the loss function that is used to optimize the network. The encoder and decoder (fully-connected layers) for the linear autoencoder were deterministic, but we will now assume these structures are probabilistic. Calling the input data x and the latent space z, we are after the distributions $P(Z|X)$ for the encoder, and $P(X|Z)$ for the decoder. Conventionally, and given the normalization done on this data, a Gaussian distribution is chosen to represent the data for this problem. Now, an assumption needs to be made on the prior distribution that will represent the latent space which is typically chosen as the standard Gaussian. Given these assumptions and relating these entities with Bayes' Rule, we get the following [5]:

$$P(Z|X) = \frac{P(X|Z)P(Z)}{P(X)} = \frac{P(X|Z)P(Z)}{\int P(X|V)P(V)dV}$$

Thus, the dense layers from the traditional encoder to get the latent space are replaced with dense layers that estimate the parameters of the mean and variance of the distribution of the latent space for the variational autoencoder. Due to the intractability of the integral in general that was just introduced, optimization methods are used in order to solve for the encoder and decoder distributions iteratively. This would normally not be possible, since during back-propagation of the loss gradient through the network there would be no way to bridge the gap between the Z used to decode the latent space and the parameters that are used to represent this latent space. To bypass this, the Z that is sampled for the decoder as shown in Figure-4 is taken by a reparameterized Gaussian distribution reconstructed from the standard Gaussian augmented by the estimated parameters. This creates a linear function that does not disrupt the gradient flow in the network and is computed as follows to sample Z [5]:

$$Z \sim \sigma_X N(0,1) + \mu_X$$

Finally, to optimize for this new representation, two loss functions are utilized - reconstruction error and Kullback–Leibler Divergence. The reconstruction error is straight forward and is typically used as the mean-squared error [5]:

$$MSE = \frac{1}{n}\sum(Y - \hat{Y})^2$$

This is desired as it builds in a trade-off between bias and variance in the estimates that help the model to fit them closely. Next, the KL divergence is computed [5]:

$$D_{KL}[Q(Z|X)||P(Z)] = -\frac{1}{2}[\sum(\ln\sigma^2 + 1 + \sigma^2 + \mu^2)]$$

Where we are trying to decipher the similarity in the encoding distribution and the prior assumption of the hidden variable. This is desired as we want the reconstruction to look

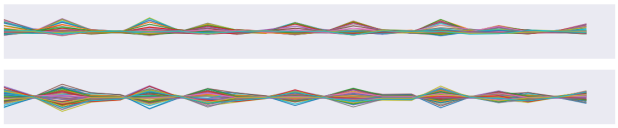

Fig. 5. Reconstruction of EEG Signals with VAE

reasonably close to the input data and not just mathematically close. In order to assist with the trade-off in the losses, hyperparameters were introduced to balance their effects. The reconstruction of the input data can be seen in Figure-5 along with the input data that is being modeled. This was computed using a latent space dimension of 30, which effectively is halving the dimensionality of the original data. The reconstruction appears to be quite strong, indicating that even with this heavily reduced dimensional latent space, the encoding captures the essence of the data much in the way PCA was attempting to.

A latent space of 2 dimensions was plotted in Figure-6 along with the event labels as colors that were extracted at the beginning of processing. Clearly, even with the severely reduced dimensionality just to make visualization easier, the latent space is able to naturally create differences in the event markers. The cyan dots correspond to the K-Means clustered centers of the events, which will not be detailed here as this is not the focus of the project. The point of showing these centers was to indicate that learning is even able to occur with severely reduced data given the more sophisticated representation of the latent space.

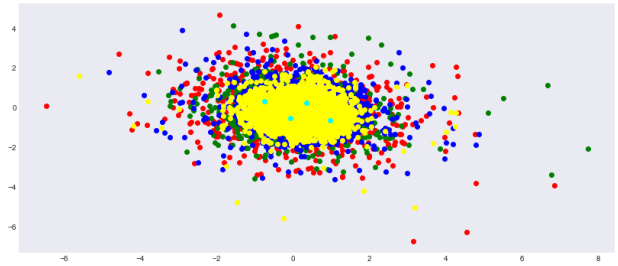

Fig. 6. Latent Space Representation of Encoded EEG Data

5. CLASSIFICATION

Due to the complexity in the data in that the labels are event based, many traditional machine learning methods aren't applicable for classification here. This is because the 60 channel EEG data and resulting time-series information is tied to an event, and thus the 3D data needs to be passed in this manner or else you lose information. Thus, for classification on this problem, a neural network architecture is used in EEGNet [6] which was designed specifically for EEG data classification in mind. This project is mainly focused on statistical methods for representation learning, but some background on EEGNet is useful before leading to results.

The fully-connected layers that were introduced for the autoencoder and VAE are abandoned here for convolutional layers. These layers create filter maps that help the network decipher important features within the data while using the same loss gradient back-propagation method for optimization as mentioned before. Specifically, to reduce the complexity of the model, depth-wise and separable convolutions are used that effectively create the same feature map, but reduce the parameters in the network. What this allows, is less data is needed in order to train the network. For most physiological data applications, data is limited due to it being taken from human subjects, thus this fits perfectly for this application. Another important addition for this network compared to the ones mentioned prior is Batch Normalization. This takes the same Z score normalization mentioned in Section-3 and applies it to the data prior to the activation layers. This helps the network in training among other benefits. Finally, since instead of reconstruction this is a classification problem, the final output activation is the softmax included below to output class probabilities to pair with the categorical cross-entropy loss to train the network:

$$S(\mathbf{z}) = \frac{e^{z_c}}{\sum_{j=1}^{C} e^{z_j}}, \; CE = -log(S(\mathbf{z}))$$

The results contained in Table-2 show the classification accuracy score when using the original data, the data after being reduced through the VAE, and the data after being reduced through PCA. The results are striking in that not only did the VAE even outperform using the original full dataset, but PCA performed significantly worse. It is worth noting that the latent space dimensionality for the VAE data was 30 for this test and the kept components after PCA were 30 components. This is not exactly a direct comparison, but does showcase that PCA may fall short in being able to adequately reduce the complexity in the physiological data while training the VAE seems to fit the application well. The MNE dataset this was tested on is somewhat of a toy dataset in that it had little noise, and was small in the total data it included. For that reason, these results may not translate readily to similar applications on other datasets.

Data	Original	VAE	PCA
Accuracy	0.79	0.85	0.45

Table 2. Results of passing EEG data through EEGNet for classification

6. CONCLUSION

Overall, physiological data is complicated to work with and many statistical methods that are used do not translate into this domain. Additionally, feature selection and engineering normally requires domain specific knowledge that provides a barrier to entry for those interested in the field from engineering. This project illuminated some deep learning methods that help bridge the gap in being able to use statistical inference directly on the data without having this expertise. Overall, the VAE representation learning demonstrated itself as a superior dimensionality reduction technique while also indicating that the EEG data does benefit from utilizing fewer features.

7. REFERENCES

[1] Todd Zorick, Katy Gaines, Gholam Berenji, Mark Mandelkern, and Jason Smith. Information transfer and multifractal analysis of eeg in mild blast-induced tbi. *Computational and Mathematical Methods in Medicine*, 2021:1–8, 04 2021.

[2] Jiang Wang and Mei Wang. Review of the emotional feature extraction and classification using eeg signals. *Cognitive Robotics*, 1:29–40, 2021.

[3] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti Hämäläinen. Meg and eeg data analysis with mne-python. *Frontiers in Neuroscience*, 7:267, 2013.

[4] Rocca Joseph. Understanding variational autoencoders (vaes), Mar 2021. https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73.

[5] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014.

[6] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of Neural Engineering*, 15(5):056013, Jul 2018.

[7] 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, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, 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. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.