Brain Computer Interface, classification task with different techniques

Emanuele Rucci[1]

[1]Emanuele Rucci 2053183, Sapienza, University of Rome

Abstract
In this study developed for the Elective in Ai, several techniques to classify EEG brain signaling have been addressed with the primary goal of using an auto encoder to obtain the latent representation of a signal. In a nutshell this study proposes three architectures: a collection of models specialized to classify signal for each subset of class belonging to a certain task from the dataset, a more generalist model which is able to deal with all the tasks together and as last a meta learning approach based on a well-known architecture, namely Prototypical Network with an attempt of enhance the latent representation by the user of contrastive learning.

Keywords
EEG signal classification, Meta-Learning, Few-Shot EEG signal classification

1. Hightlights

- The auto encoder variants are worse than the classic auto encoder
- Proposed meta learning approach
- Specialized model work better
- Meta learning approach might be improved the document;
- Performance could be improved by the use of feature extraction.

2. Introduction

Brain-Computer Interface (BCI) technology has garnered significant attention in recent years due to its potential to revolutionize human-computer interaction by enabling direct communication between the brain and external devices. One of the fundamental tasks in BCI systems is the classification of Electroencephalogram (EEG) brain signals, which involves interpreting neural activity patterns to infer the user's intentions or mental states. Accurate and efficient EEG signal classification is crucial for the development of reliable BCI applications, ranging from assistive technologies for individuals with motor disabilities to advanced neurofeedback systems for cognitive enhancement.

This study aims to explore different techniques for EEG signal classification, in order to have a model able to decode in real time the intention of the user with respect to a robot.

Primary focus of this study consists in leveraging autoencoders and sparse autoencoder to obtain latent representations of EEG data. Specifically, the study proposes:

Specialized models tailored to classify signals for each subset of class belonging to a specific task from the dataset. These models aim to capture task-specific features and optimize classification performance for individual tasks.

A more generalist model designed to handle multiple tasks simultaneously. By incorporating a broader range of features and leveraging shared information across tasks, this model offers flexibility and scalability in BCI applications.

A meta-learning approach based on the Prototypical Network architecture.

Contrastive learning to train an auto encoder used to observe the behavior of the generalist model and the meta learning approach.

Through rigorous experimentation and evaluation, we seek to compare the performance of these architectures and demonstrate their efficacy in EEG signal classification tasks.

3. Related Works

Prior research in EEG signal classification has explored various techniques aimed at interpreting neural activity patterns for BCI applications. Autoencoders have emerged as a significant tool in signal processing, including EEG signal analysis [2][3].

Existing approaches in BCI classification have tackled the challenge of handling classification tasks using different strategies. In recent years, meta-learning techniques have gained attention for their potential to address the adaptability and scalability requirements of real-time EEG classification systems. Meta-learning frameworks,

Woodstock'21: Symposium on the irreproducible science, June 07–11, 2021, Woodstock, NY

such as Prototypical Networks [4][5], have been proposed to enable systems to accommodate new mental states over time without the need for extensive retraining. These approaches aim to facilitate the rapid adaptation of BCI systems to evolving user needs and application requirements.

The incorporation of contrastive learning techniques in EEG signal classification [6] represents a novel approach to address the challenges posed by the similarity of neural activity patterns across different mental states. By leveraging contrastive learning frameworks, researchers aim to enhance the separation of classes in the latent space representation of EEG signals, thereby improving classification performance.

Overall, prior work in EEG signal classification has laid the groundwork for exploring diverse techniques and methodologies to address the complexities of interpreting neural activity patterns for BCI applications. The present study builds upon this body of research by proposing and evaluating various architectures, including specialized models, generalist models, contrastive learning frameworks, and meta-learning approaches.

4. Metodology

4.1. Dataset

The Healthy dataset comprises over 1500 EEG recordings obtained from 109 volunteers, each performing various motor and imagery tasks while EEG signals were recorded using the BCI2000 system [1]. Each sample is composed of 64 float values plus a label (see event codes below).

Experimental setup: each participant engaged in 14 experimental runs, consisting of different tasks and baseline measurements.

Two one-minute baseline runs were conducted:

- Baseline with eyes open.
- Baseline with eyes closed.

Task Runs:

Each participant performed three two-minute runs for each of the following tasks:

- Task 1: Real motor action - Opening and closing left or right fist in response to a target.
- Task 2: Imagined motor action - Imagining opening and closing left or right fist in response to a target.
- Task 3: Real motor action - Opening and closing both fists or both feet in response to a target.
- Task 4: Imagined motor action - Imagining opening and closing both fists or both feet in response to a target.

Event Codes:

For each action, a set of labels describe the possible event which describe the mental state of the user during the experiment. Each event code consists of an event type indicator (T0, T1, or T2) concatenated with the Task number it corresponds to.

For instance, "TASK1T2" indicates the onset of real motion in the right fist, while "TASK3T2" indicates the onset of real motion in both feet [1].

This dataset provides a diverse set of EEG recordings encompassing various motor and imagery tasks, making it valuable for studying EEG signal classification in BCI applications.

Features Dataset:

We extracted Power Spectral Density (PSD) features from the frequency domain using Welch's method. Specifically, we chose to calculate the PSD for three frequency bands: 'Delta': (0.5, 4hz),'Theta': (4, 8hz), 'Alpha': (8, 13hz), 'Beta': (13, 30hz), 'Gamma': (30, 80hz), because they provide insights into various cognitive and neurological states. Calculated PSDs were normalized. They were converted to decibels and averaged across frequencies for each frequency band to generate one feature per band per channel. These features were then horizontally stacked to form a feature vector for each epoch. These features have been extracted for each channel of 14 selected channels (sensitive to the motor cortex), generating a feature matrix suitable for data analysis and machine learning tasks.

4.2. One model per task

In this setting has been trained a model for each task. This is to study to which extend a deep learning model could be able to deal with the raw EEG dataset. More specifically, this study wants also to understand if one of the regularizations technique from the literature [7][8] [9] would enhance the model. Generally speaking, the process in this phase is composed of two steps executed for each task:

1. Autoencoder pretraining minimizing the reconstruction loss (mean squared error) as default, in combination with other terms in certain experiments

$$MSE = \frac{1}{n}\sum_{i=1}^{n}[g(f(x_i)) - x_i]^2$$

Here f(x) represents the output of the encoder, while g(x) represents the output of the decoder, while x is the input and n is the number of samples.

2. Train a classifier (a feed forward network) using the Cross Entropy Loss.

The hidden size of all the auto encoder has been fixed to 128 to further reduce the hyper parameters search space. Figure 1 shows the architecture schema for this type of model.

4.2.1. Sparse AutoEncoder

First let's introduce a sparse autoencoder. The term "sparse" refers to the idea that the autoencoder is encouraged to learn representations of the data that are sparsely activated, meaning only a small subset of neurons are activated at a time. Sparse Constraint: In a sparse autoencoder, a regularization term is added to the objective function to encourage sparsity in the activations of the neurons in the encoding layer. This means that only a few neurons should be active at any given time, effectively capturing the most salient features of the input data. The objective function jointly minimizes the MSE and this new regularization term, ending up in the following Loss function:

$$L(x, f(x)) = \frac{1}{n}\sum_{i=1}^{n} L(x_i, g(f(x_i))) + \lambda \sum_{j=1}^{k} KL(\rho \parallel \hat{\rho}_j)$$

4.2.2. Weight Decay

This type of autoencoder has another regularization term in the objective such that overfitting should be avoided. Weight decay is a common form of regularization that penalizes large weights in the network, impose penalty on the magnitude of the weights in the network.

$$L(x, f(x)) = \frac{1}{n}\sum_{i=1}^{n} L(x_i, g(f(x_i))) + \lambda \sum_{j=1}^{k} \|W_i\|^2$$

4.2.3. Non negative constraints

In this settings, the model's weights are constraint to be non-negative. $\theta^{t+1} = \text{clamp}(0, \theta^t)$

4.2.4. The classifier

The classifier is applied on the hidden representation of the encoder. The training uses the cross-entropy loss, the prediction is obtained as follows:

$y_k = \arg\max_k C_k(f(x))$ Where C_k is the classifier for class k which has j labels, is composed of 3 layers: a linear layer, a RELU activation function and another linear that maps into the j labels space, $f(x)$ is the encoder and y_k is the prediction for the class k.

4.3. One general model

This architecture has been devised in order to assess the capacity of an Autoencoder to extract meaningful information from the EEG considering all the tasks available in the dataset. The loss of such model again can incorporate all the previous discussed enhancement (sparsity, weight decay… More details in the experiments section).

Also in this case, the hidden representation extracted from the encoder are then used to train a classifier. More specifically, have been tried a total of four classification heads;

4.4. Contrastive Learning

The concept behind this architecture stems from both its structural design and the experimental setup used to generate the dataset. Essentially, the hypothesis posits that merely conceptualizing a movement does not necessarily yield significantly distinct EEG signals compared to the actual execution of that movement. In other words, signals from different tasks may exhibit similarities.

To tackle this potential challenge, a contrastive learning framework has been employed. This framework aims to derive a latent representation of each sample, ensuring that samples belonging to the same class are positioned closely to each other in a latent hidden space, according to a defined distance metric, while samples from different classes are positioned farther apart.

The objective process that leads to the creation of the contrastive autoencoder is the InfoNCE [10]. This loss requires $X = x_1, \ldots, x_N$ which is the set of N random samples containing one positive sample (same class) and N-1 negative samples. This led to the optimization of the following function:

$$L_N = -\mathbb{E}_X\left[\log\left(\frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}\right)\right]$$

This architecture has been trained with the idea that could improve an autoencoder already trained from paragraph 2.3, but have been studied also the behavior of an auto encoder trained from scratch.

After the pretraining phase has been conducted, a fine-tuning and a classification head training has been performed. Figure 2 shows the overall classification architecture.

4.5. Meta-Learning

In tasks like real-time EEG classification for mental states, a crucial requirement for such systems is the ability to adapt and accommodate new mental states over time. Let's illustrate this with an example: imagine an operator in a company employing such a system to teleoperate a robot in real time. As time progresses, the company may enhance the robot's capabilities by adding new actions it can perform, each associated with a distinct mental state. In the existing frameworks described earlier, this requirement poses a significant challenge to the long-term usability of the system. Adapting the framework to handle new mental states typically involves retraining the entire system to account for changes in labels. Moreover, this process can lead to a decrease in performance as classification becomes more complex. To address this challenge, we turn to meta-learning techniques introduced in [4]. Specifically, the Prototypical Network has been employed in various experiments to tackle this issue.

4.5.1. Prototypical Networks

In few-shot classification scenarios, we are provided with a limited support set consisting of N labeled examples, denoted as $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$. Here, each $x_i \in \mathbb{R}^D$ represents the D-dimensional feature vector of an example, and $y_i \in \{1, \ldots, K\}$ represents its corresponding label, where K denotes the total number of classes. S_k signifies the subset of examples labeled with class k. The idea of this type of network is to classify a sample by comparing it with a prototype with shape $\mathbb{R}^M$. The prototypes are computed by selecting samples from the set of samples of a given class and calculating the mean. Here, we compute the mean of the z representation of the support set (the one extracted from the encoder $f_\phi(x_i)$).

$$c_k = \frac{1}{|S_k|} \sum_{(x_i, y_i) \in S_k} f_\phi(x_i)$$

Considering the value of a distance function $d : \mathbb{R}^M \to [0, +\infty)$ from a sample x_i to the computed prototypes, the minimization process makes use of the Cross Entropy Loss in order to maximize the likelihood $p_\phi(y = k \mid x)$.

In the experiments proposed both Euclidean and cosine distances have been employed.

The training episode includes the computation of a set of fallback prototypes that are used in the case in which in certain batch there is a missing label for which it is impossible to compute the prototype; then includes the computation of $f_\phi(x_i)$, the selection of the support set from the current batch, the generation of the prototypes, the computation of the distance and finally the loss.

This framework also includes the usage of a contrastive pretrained encoder. Figure 3 shows the overall architecture for the meta learning framework.

5. Experiments & Result

This section presents the experiments conducted using both raw EEG data and feature-extracted data. We explore three different modeling approaches: individual task-specific models, a general model, and contrastive learning, culminating in a meta-learning framework.

5.1. One model per task

In this setting, the raw EEG dataset is used to train four autoencoders and four classifiers, one for each task. Each classifier is equipped with a projection head selected from a set of candidates based on performance. The training pipeline is as follows:

- Pre-train the autoencoder on the reconstruction task.
- Train the classifier's projection head, fine-tuning the autoencoder.

A Bayesian hyperparameter search (HPO) with early stopping was conducted during the pretraining of the autoencoder to prevent overfitting. Ten trials were performed, focusing on minimizing the mean squared error (MSE) loss.

The table below summarizes the classifiers' performance on the test set:

Task	Test Accuracy	Test Loss height1
0.76	0.85 2	0.92
0.21 3	0.70	0.97 4
0.92	0.27 height	

Figures 4 and 5 illustrate the confusion matrix and training plots for Task 1.

5.2. One general model

This approach uses the same pipeline as the task-specific models but focuses on a general model. The pretraining results for the autoencoder are shown below:

Model Name	Test Loss
1	0.014
2	0.0035
3	0.0018
4	0.0018
5	0.0073
6	0.0114
7	0.016

The hyperparameters for each model configuration include:

- Model 1: non negative constraint, weight decay loss
- Model 2: weight decay loss
- Model 3: only MSE
- Model 4: sparsity loss, weight decay loss
- Model 5: weight decay loss enabled
- Model 6: sparsity loss, weight decay loss
- Model 7: sparsity loss enabled

Head	Accuracy	Loss
1	0.47	2.080
2	0.47	2.043
3	0.56	1.63
4	0.55	1.82

Table 1
Accuracy and loss for each head

The sparsity factor is 0.05, with a loss coefficient of 0.001 for both sparsity and weight decay losses.

The classifier heads' performance is summarized below:

Figures 6 and 7 show the confusion matrix and training plots for Head 3.

Regarding the basic task of distinguishing between tasks (using the features dataset), the experiment that has been conducted employs the head type 3 given the success obtained in the set of experiments for the raw EEG dataset.

Head 3's performance (chosen because the one with higher performance from set of above experiments) with the features dataset, are reported in the following table.

Head	Accuracy	Loss
3	0.66	0.90

Table 2
Performance for head 3 with the features dataset

5.3. Contrastive Learning

This set of experiments includes the training of a general autoencoder from scratch, and the finetuning of the general pretrained autoencoder's encoder from the previous approach.

The strategy used to select negative samples, useful for the computation of the loss, consists in the selection of 60% of the samples in the considered batch with a class different from actual observed sample.

The inference loss result in the following table reports over the whole set of labels from all the tasks, as expected, the finetuning leads to lower value of the loss, hence, a better separation of the classes in the latent space representation of the raw EEG signal. This shows the effectiveness of the idea of iterative enhance a pretrained encoder able to encode mental states.

Encoder	Test Loss
From scratch	3.98
Finetuning	3.24

Table 3
Test loss for different encoders

Given the hypothetical enhanced separation of the classes in the latent space, the finetuned contrastive encoder has been then used to train a classifier and a prototypical network in the meta learning framework. Again, the classifier directly employed for the training is the classification head 3 and the weights of the encoder itself are still trainable, in order to further enhance again the representation extracted for the classification task.

The meta learning framework employs the Euclidean distance and as well as the classifier, is based on the finetuned contrastive encoder to extract the latent vector. Also, these last two discussed experiments have been equipped with an early stopping callback over the validation accuracy, with a quite big patience value (300) and a min delta of 0.001.

Approach	Test Loss	Test acc.
Classifier	1.83	0.51
Meta learning	2.73	0.48

Table 4
Results for the raw EEG dataset

Approach	Test Loss	Test acc.
Classifier	1.12	0.64
Meta learning	1.13	0.57

Table 5
Results for the features extracted dataset

Figure 8, 10 contains the confusion matrix for the classifier model respectively for raw EEG and features dataset, figure 9 contains the plots for training of the classifier.

5.4. Meta-Learning

Here, four experiments are proposed obtained by the combination of the hyperparameters distance function (Euclidean or cosine similarity) and encoder type (best general model or from scratch); the dataset used is the EEG signal dataset.

The choice of hyperparameters is based on prior trials. The encoder configuration matches exactly with the one described in Section 5.2, which yielded the best performance. For the classification head, we used the one (head 3) that achieved the highest performance in previous experiments.

The table summarizes the results:

For the features extracted dataset, the experiments as well as the performance are reported in the following table:

Figure 11,12 shows the generalizations capabilities of the different models during training, using as metric the validation accuracy (raw EEG dataset, features dataset).

The performance of the classifier equipped with a pretrained contrastive encoder proves to be ineffective, with

Encoder	Test Loss	Test acc.	Distance fn.
Best pretrained	2.56	0.34	Euclidean
Best pretrained	2.55	0.29	Cosine Sim.
From scratch	2.61	0.34	Euclidean
From scratch	2.55	0.31	Cosine Sim.

Table 6
Results for the EEG signal dataset

Encoder	Test Loss	Test acc.	Distance fn.
Best pretrained	1.53	0.39	Euclidean
Best pretrained	1.37	0.35	Cosine Sim.
From scratch	1.36	0.39	Euclidean
From scratch	1.36	0.42	Cosine Sim.

Table 7
Results for the features extracted dataset

a performance metric of 0.51, still below that achieved by fine-tuning a non-contrastive autoencoder (0.56). This discrepancy is also reflected in the test loss, which remains low in the latter case. This is also reflected in the experiments with the features dataset: the pretrained contrastive encoder leads to lower performance (0.64) than the finetuning of non-contrastive autoencoder (0.66).

In the context of the meta-learning approach, the incorporation of a pretrained encoder offers a slight advantage to the model in terms of raw EEG, but this is different in the features dataset for which the best performance is reached with the autoencoder trained from scratch and the distance function cosine similarity. This approach deserves a preliminary baseline and suggests the need for more sophisticated methodologies capable of accommodating an expanded number of labels. Notably, the utilization of a pretrained contrastive encoder significantly improves the meta-learning approach, elevating accuracy from 0.34 to 0.48 for the raw EEG and from 0.42 to 0.57 for features dataset, as expected.

5.5. Discussion

Figure 13, 14 reports an overview of the preformances for both the dataset. From the results of the initial approach, it becomes evident that specialized models exhibit a remarkable ability to handle EEG signals and achieve high classification performance. Further refinement of the encoding-decoding architecture, such as incorporating dropout layers into the autoencoder, may enhance classification precision. Notably, the training plot does not indicate any signs of overfitting thanks to the implementation of early stopping techniques. In contrast, the general model, as anticipated, falls short of achieving high performance levels. This underscores the necessity for more sophisticated data preprocessing methods to potentially mitigate noise and enhance signal quality. This lead to observe that the head 3 for the classification model performs decently on features dataset, suggesting that further improvements in the architecture and/or in the feature extraction phase might be a good future direction.

A plausible explanation for the observed errors in the raw EEG dataset, lies in the challenge of the model learning the requisite latent space separation for accurate sample classification, moreover the subtask labels dedicated to the trial poses a challenge for the classification itself. Analysis of the confusion matrix reveals that the model struggles most with the initial class of the first task. Similarly, examination of the training procedure reveals no instances of overfitting.

5.6. Future work

- Contrastive autoencoder: Pretrained contrastive encoder trained with a combination of MSE loss and contrastive loss to pretrain an autoencoder (both encoder and decoder) to jointly optimize the reconstruction and the latent space separation.
- Given that specialized models are quite effective, a modular architecture would arise; A mixture of expert network in which the router sends to the right expert the signal sample to be classified
- A more robust preprocessing of the data that remove noise from the raw dataset.

5.7. Limitations

Despite the promising results, this study has several limitations that must be acknowledged. Firstly, the reliance on specialized models for each task can be impractical for real-world applications where the mapping between mental states and tasks may change over time. This necessitates frequent retraining, which is both time-consuming and resource-intensive. Secondly, while the contrastive learning approach showed potential, its effectiveness was not consistently superior across all tasks and datasets, indicating that further refinement of this method is needed. Additionally, the generalist model's performance was suboptimal, suggesting that more advanced preprocessing techniques and noise reduction methods could be crucial for improving its efficacy. Another limitation is the dependency on a specific dataset, which may not fully represent the diversity of EEG patterns encountered in different contexts or populations. Future studies should consider incorporating more varied datasets to enhance the generalizability of the findings. Lastly, the meta-learning approach, although showing improvements, still requires substantial development to handle a larger number of classes effectively. Addressing these limitations in future work will be essential for advancing the practical application of EEG signal classification in brain-computer interface systems.

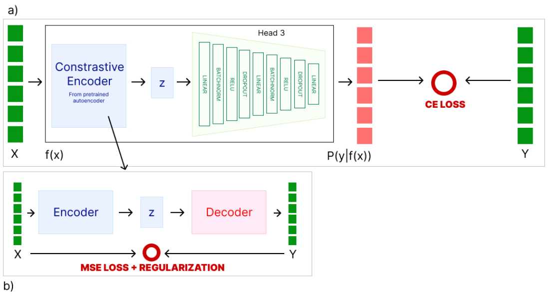

Figure 1: Architecture of models valid both for approach 1 and 2. A) is the whole classifier architecture: encoder, projection head B) the auto encoder with the Mse loss and the regularization methods explained in paragraph 2.2

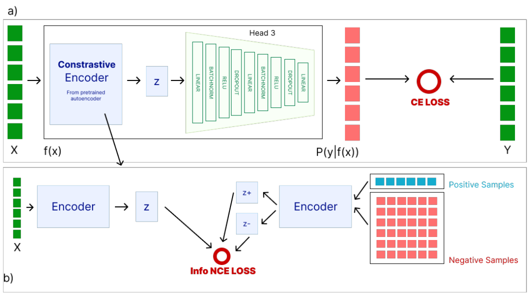

Figure 2: A) Classifier with the contrastive pretrained encoder B) Contrastive Encoder procedure

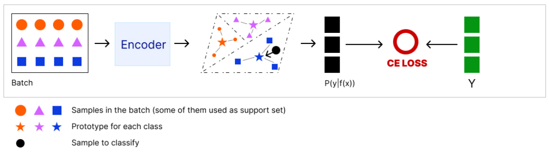

Figure 3: Prototypical Network schema: Creation of prototypes from the batch, assignment of the sample to the most closest centroid, computation of the loss.

Figure 4: Confusion matrix of each classifier on the correspondent test set of the class.

6. Conclusions

This study introduces various approaches to address EEG signal classification utilizing autoencoders. The results highlight that employing a specialized model for each task yields optimal outcomes. However, this strategy is contingent upon the stability of the mapping between mental states and robot actions over time. Should this mapping evolve, the need for retraining specialized models arises, presenting a potential drawback. Conversely, the meta-learning approach, while serving as a preliminary baseline, reveals insights into the efficacy of a contrastive approach in augmenting the capabilities of meta-learning frameworks. These findings underscore the potential for further exploration and refinement in future research endeavors.

7. Acknowledgments

Project conducted for one of the submodules, held by Professor Napoli, of the Elective in AI course, Artificial Intelligence and Robotics 2023/2024 Sapienza, University of Rome.

References

[1] N. C. for Adaptive Neurotechnologies (NCAN), Bci2000 system. (n.d.), ???? URL: http://www.bci2000.org.

[2] S. Phadikar, N. Sinha, R. Ghosh, Unsupervised feature extraction with autoencoders for eeg based multiclass motor imagery bci, Expert Systems with Applications 213 (2023) 118901. URL: https://www.sciencedirect.com/science/article/pii/S0957417422019194. doi:https://doi.org/10.1016/j.eswa.2022.118901.

[3] Q. Li, Y. Liu, Y. Shang, Q. Zhang, F. Yan, Deep sparse autoencoder and recursive neural network for eeg emotion recognition, Entropy 24 (2022).

Figure 5: Training plot of the classifier model for compactness is reported only for the task 1.

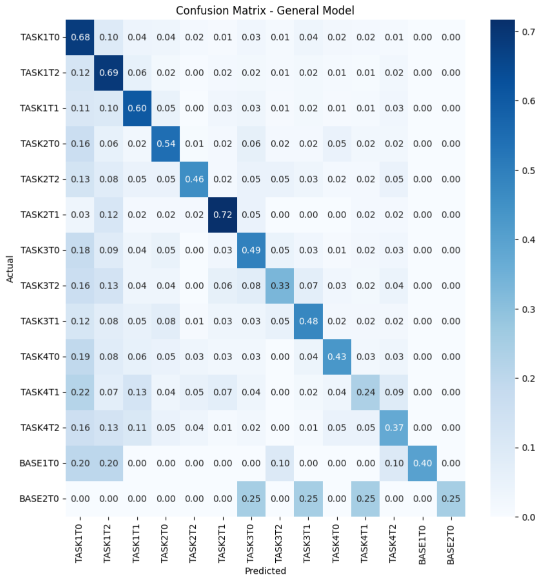

Figure 6: Confusion matrix the best classifier for the general model with head 3

URL: https://www.mdpi.com/1099-4300/24/9/1187. doi:10.3390/e24091187.

[4] J. Li, C. Wu, J. Pan, F. Wang, Few-shot eeg sleep staging based on transductive prototype optimization network, Frontiers in Neuroinformatics 17 (2023). URL: https://www.frontiersin.org/articles/10.3389/fninf.2023.1297874. doi:10.3389/fninf.2023.1297874.

[5] J. Snell, K. Swersky, R. S. Zemel, Prototypical networks for few-shot learning, 2017. arXiv:1703.05175.

[6] X. Jiang, J. Zhao, B. Du, Z. Yuan, Self-supervised contrastive learning for eeg-based sleep staging, 2021. arXiv:2109.07839.

[7] A. NG, Sparse autoencoder, ????

[8] M. Andriushchenko, F. D'Angelo, A. Varre, N. Flammarion, Why do we need weight decay in modern deep learning?, 2023. arXiv:2310.04415.

[9] B. O. Ayinde, J. M. Zurada, Deep learning of constrained autoencoders for enhanced understanding of data, IEEE Transactions on Neural Networks and Learning Systems 29 (2018) 3969–3979. URL: http://dx.doi.org/10.1109/TNNLS.2017.2747861. doi:10.1109/tnnls.2017.2747861.

[10] A. van den Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, 2019. arXiv:1807.03748.

[1] [2] [3] [4] [5] [6] [7] [8] [9] [10]

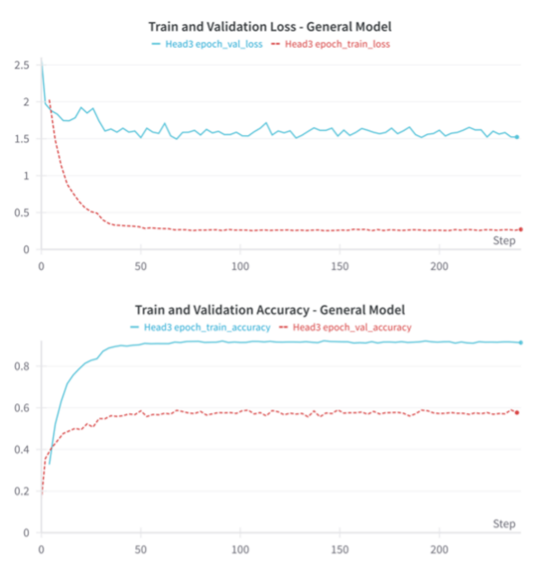

Figure 7: Training plot of the best classifier in the general model settings using the head 3.

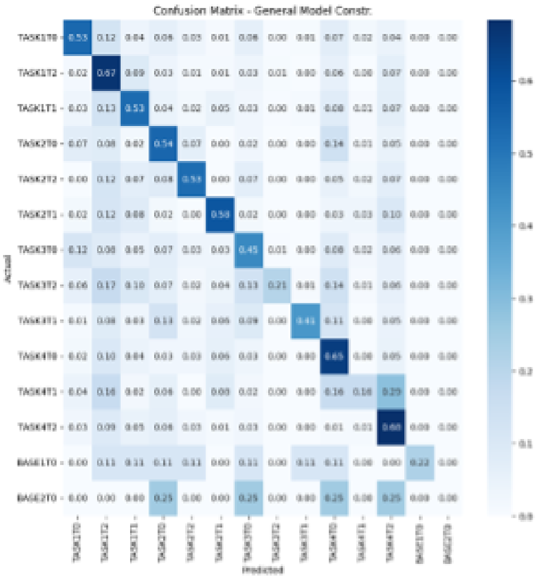

Figure 8: Confusion matrix for the general model with head 3 and the pretrained contrastive encoder.

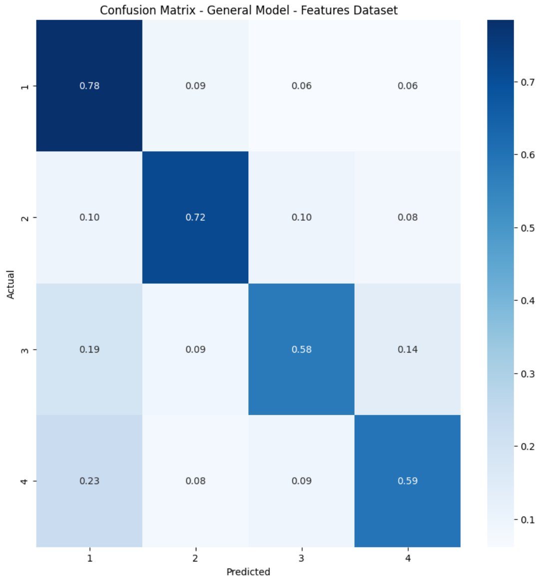

Figure 10: Confusion matrix for the general model with head 3 using the features dataset.

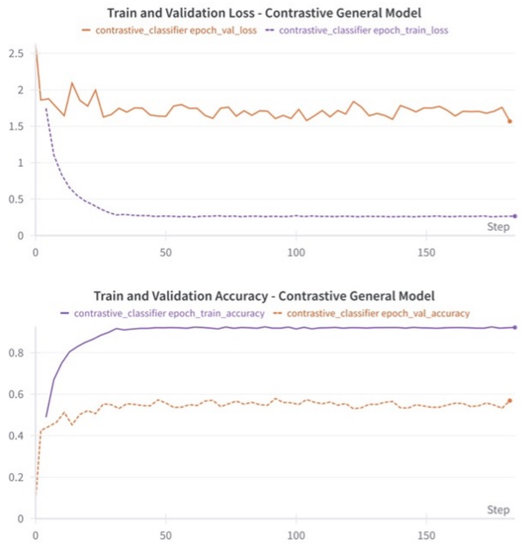

Figure 9: Training plot of the best classifier in the general model settings using the head 3.

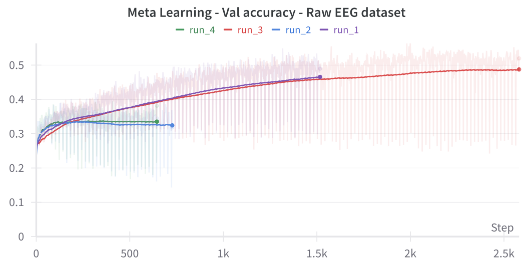

Figure 11: Validation accuracy for the meta learning experiments on the Raw EEG dataset; run 1: pretrained encoder and euclidean distance, run 2: pretrained with cosine sim., run 3 and run 4 respectively with encoder trained from scratch and euclidean and cosine sim.

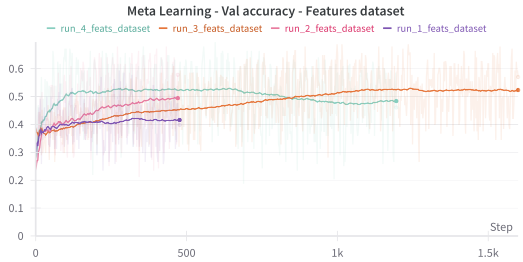

Figure 12: Validation accuracy for the meta learning experiments on the features dataset; run 1: pretrained encoder and euclidean distance, run 2: pretrained with cosine sim., run 3 and run 4 respectively with encoder trained from scratch and euclidean and cosine sim.

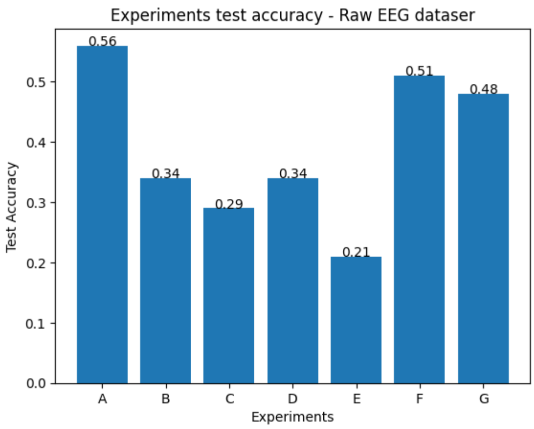

Figure 13: Test accuracy for the whole set of experiments performed. A: general model, B,C,D,E: Meta Learning respectively with pretrained (B,C) and From scratch (D,E) encoder and euclidean dist /cosine sim, E: Pretrained contrastive autoencoder, F: Metalearning approach with pretrained contrastive autoencoer. Dataset used: Raw EEG dataset.

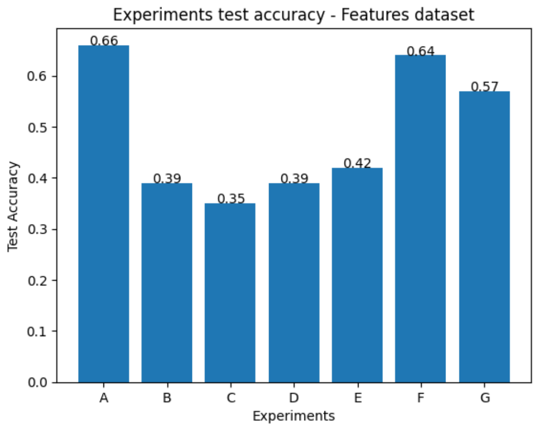

Figure 14: Test accuracy for the whole set of experiments performed. A: general model, B,C,D,E: Meta Learning respectively with pretrained (B,C) and From scratch (D,E) encoder and euclidean dist /cosine sim, E: Pretrained contrastive autoencoder, F: Metalearning approach with pretrained contrastive autoencoer. Dataset used: Features dataset.