UNIVERSITY OSNABRÜCK

Inner Speech - Project Report

Fabienne Kock, Lucas Liess-Duquesne, Sascha Mühlinghaus

April 1, 2022

1 INTRODUCTION

To decode inner speech from EEG-data is an exciting idea but at the same time a little unsettling. The basis for our project stems from a study that was published in 1985 by Tsai et al. (2009). They have proven that mere manipulation of neural firing is sufficient to alter the behavior of mice. Hence, neural activity actually corresponds to behavioral traits and the same applies for mental states. Another prerequisite for our project is the EEG (Electroencephalography) brain imaging technique that allows to extract, in a non-invasive manner temporally very precise information about neural activity. However, the interpretation of such data poses a challenge to scientists as the measurements do not have a good spatial resolution and are additionally vulnerable to instrumental and subject dependent noise.

2 PROJECT DESCRIPTION

The aim of this project is to classify EEG-data that represents the neural activity of a human subject. In the experiment the subjects were presented with a "directional cue" and asked to engage mentally with it. As we do not have the capacities nor abilities to record such data, we based our project proposal on a freely accessible dataset Nieto, Peterson, Rufiner, Kamienkowski, and Spies (2022) about "Inner Speech". A follow up study on this dataset already provides evidence that there are distinguishable features of brain activity among the inner speech categories in respect to the cue presented in the experiment Nieto, Rufiner, and Spies (2021). Additionally, a study which analyzed the dataset with the "convolutional neural network" (CNN) architecture "EEGNet" Lawhern et al. (2018) was able to classify the

data with an average accuracy of 29.7% which is slightly above a random classification of 25% van den Berg, van Donkelaar, and Alimardani (2021). Together this shows that the EEG recordings principally allow a feature extraction for distinguishing the respective cue and at least to a minor extent a classification with "deep learning" (DL) approaches is possible.
In this project, we want to test whether it is possible to successfully train only with the data from the action interval(3.2). From our point of view, the data measured before and after inner speech might contain patterns that are related to the cue. Additionally, we thought that there was some room left for improvement in terms of data preparation. A successful automatized classification of certain mental states would improve scientific analysis as well as the medical treatment of paralyzed patients.

The task at hand is fine-grained, as the differentiation between states is mainly provided by the semantic difference of the word cues (difference between "up" and "left"), which are supposedly decoded into neural activity. On the other hand, binary classification between sleeping and wake states have already been accomplished by DL approaches (Abbasi et al., 2020) because the activity patterns are more easily distinguishable. Our dataset poses a greater challenge due to its nuanced differences between classes, which implies a larger data demand for DL. In general, a major difficulty is that the data acquisition for such a paradigm requires human subjects and sufficient instrumental equipment to record data. Compared to other data acquisition procedures in which the data is already present and can be simply collected, in our case the data needs to be recorded. Nevertheless, we had some ideas in mind on how to decrease the learning effort of the network and thereby increase the accuracy of our classifier. CNN architectures are the standard approach for analyzing EEG-data, that is because EEG channels placed on the head surface measure electric impulses which depict a spatial coherence, meaning channels that are close to each other are more likely to depict similar activity patterns. As CNNs successfully extract features of both conventional and brain images, we chose that model architecture.

Our enthusiasm for the project is due to the idea of using DL to differentiate between the brain activity that corresponds to semantic representation of mental states. Given the rather little provided data such classification is very challenging, however as this situation applies for many other brain computer interface (BCI) related learning tasks, it would be interesting to find ways of tackling this difficulty. In the following, our data and the functionality of EEG will be briefly described followed by the different approaches we chose to increase performance of the classifier. Lastly, we will present our results and give a short outlook on what further work could be done.

3 The Data

3.1 Electroencephalography

Electroencephalography (EEG) is a method that measures and records electric charges through electrodes which are usually placed on the scalp of a subject. These electrodes are able to detect very subtle changes in a spatially close environment with a temporal resolution of milliseconds, the used EEG has a sampling rate of 1024 Hz. The activity is mainly induced by the postsynaptic potential of a neuron and not the action potential, besides the electrode is only able to detect the activity of larger groups of neurons which are active at the same time, these electric signals are summed up and therefore stand up from a baseline activity. This implies that certain activity cannot be pinned down to small neuron populations, instead the precision lies around a cm scale, which is one of the major drawbacks of the EEG. Today, usually the raw measurements are processed and filtered to make them clearer. Often the data is transformed by a Fourier-Transformation to retrieve different frequency oscillations. In this case the raw measurements were not altered but directly saved. Later for preprocessing purposes the data was digitally filtered and reduced to a resolution of only 254 Hz.

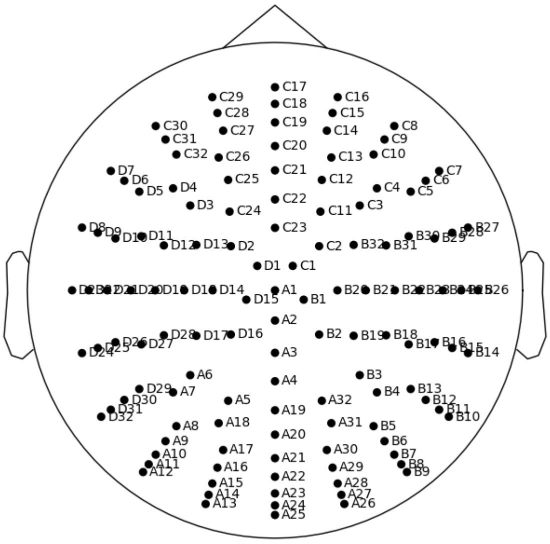


Figure 1: The figure visualizes the location of electrodes on the BioSemi cap with 128 electrodes. The eight electrodes not shown on this figure where external electrodes used to capture noise. (Nieto et al., 2022, p. X)

3.2 Dataset

As mentioned above we are using a dataset created by Nieto et al. in 2009. In total 10 subjects where examined in three separate sessions each. The participants sat in front of a screen which in certain intervals displayed an arrow which either indicated "left", "right", "upwards" or "downwards", following this visual cue, respectively one of three conditions was tested. These conditions are "inner speech", "pronounced speech" and "imagined speech". In inner speech the participant is asked to imagine the displayed "directional word", without any motor action involved. In pronounced speech the "direction word" should be spoken out loud and in the visualized condition a dot displayed at the center of the screen should be imagined to be moved in the displayed direction (Nieto et al., 2021). already cleaned the channel data and extracted the 128 channels via ICA. As such each trial contains the recordings of 128 channels for 4.5 seconds of cleaned EEG data. The cued action takes place between 1 and 3.5 seconds. As such we filtered the data points for each channel in each trial to only contain the data points starting from 1 second to 3.5 seconds (Figure 2). Eventually, we retrieve a dataset of shape trials*channels*data points with the size (trials, 128, 640) for each of the three conditions. To visualize the place of recording of each channel on the scalp he placement of the electrodes can be seen in Figure 1.

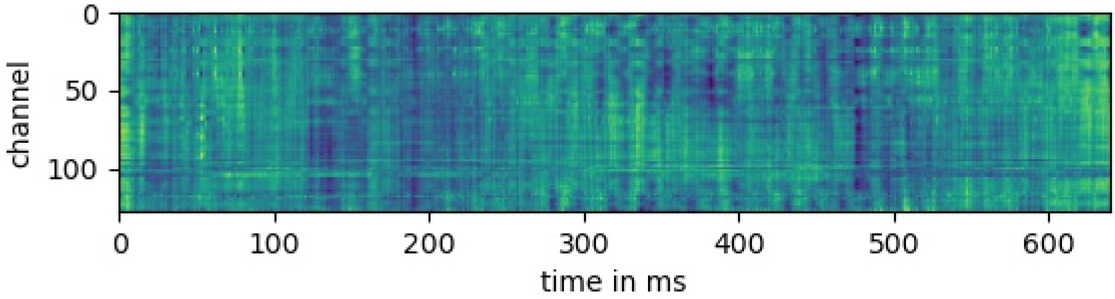


Figure 2: examplary data of the interval 1 - 3.5 seconds. In this specific example the channels are zscore normalized.

In order to filter and simplify the data, we took two approaches into consideration, one was Principal Component Analysis (PCA), which projects high dimensional data into lower dimensional space, and the other an autoencoder that reduces complexity by creating a latent space which than would be fed into the classifier. Here we only apply and test the first approach and compare it with working with the raw data.

4 Background

4.1 Pretraining/Transfer learning

We came up with the idea of pretraining our models rather late during the course of our project. We trained the weights of our classifier network with more conditions than just the inner speech data in order to familiarize the model with EEG data. Again besides, the inner speech condition, there was also the condition silent speech in which the subject was required to reproduce the motor action of normal speech without actually producing sound. The third condition was imagined speech during which subjects imagined themselves moving the cue. Hence, the conditions are very similar but differ to a degree. Generally, pretraining or transfer learning(Bengio, 2012) refers to training artificial neural networks (ANNs) with data that is not directly related to the target function but may contribute to the learning of general concepts . As an instance it is possible to load weights for a certain kind of image classification for specific domains like food recognition (Singla, Yuan, & Ebrahimi, 2016) but also more sophisticated tasks as the classification of x-ray images (Alebiosu & Muhammad, 2019) is possible with transfer learning. This approach enables a weight training that prevents overfitting behavior of the classifier due to very little data points and a complex network architecture. Overfitting refers to the phenomenon in which DL networks fit very tightly to the training data and are not able to generalize the key concepts to unseen data. This is usually indicated by a high training and a low test accuracy.

4.2 Principal component analysis

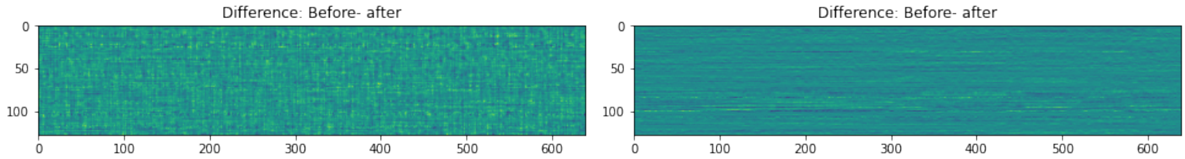
The goal of principal component analysis (PCA) is to project high dimensional data into fewer dimensions without losing key-information. It projects the data along new axes, the so called components, that explain the most variance in the data. Overall, this should make the differences in the data more easily observable. We conducted two different PCA approaches on the data: 1) reshaping the channels and time dimension into one so the EEG data went from shape trials*channels*time to trials*reshaped(channels*time) and reducing the reshaped dimension for each trial and 2) reducing the channel dimension for each trial. For each of the approaches we used the PCA method from sklearn.decomposition. To account for differences between subjects and trials we scale the data using the RobustScaler of the sklearn.preprocessing functions (for documentation see: RobustScalar), before we apply PCA. As the method only allowed two input dimensions we fitted the data for the channel reduction on the mean of all data samples in the training set, before applying the PCA on each sample individually.

To make sure the number of principal components we reduce the data to represents most of the variance of the extracted inner speech data we ran the PCA several times on a training split of the data (see Table 1). To make sure we have a high amount of variance explained in the test and validation data as well, we choose to use the number of components needed to achieve 98% on the training data. This would allow us to account for differences in information in the train, test and validation data bit still hopefully achieve an explained variance of above 90%.

Furthermore, to visualize the data reduction, we plotted the PCA transformed data, the original and the reduction reconstructed data (see Figure 4) and the difference between the original and reconstructed data (see Figure 3) for the first sample using 46 components. Taking at a look at the reconstructed data (see Figure 3) we can see that the application of the PCA does not seem to erase the structure of the data but seems to smooths it out. This can also be seen in the difference images (see Figure 3. PCA does seem to act like a filter. We assume that the PCA transformed data is more easy to learn for the network since we distribute the data on axes that explain most variance.

Variance explained	**Principal Components**	
	reshaped	*channel*
0.8	443	7
0.85	573	9
0.9	756	14
0.95	1038	26
0.98	1319	46
original	81920	128

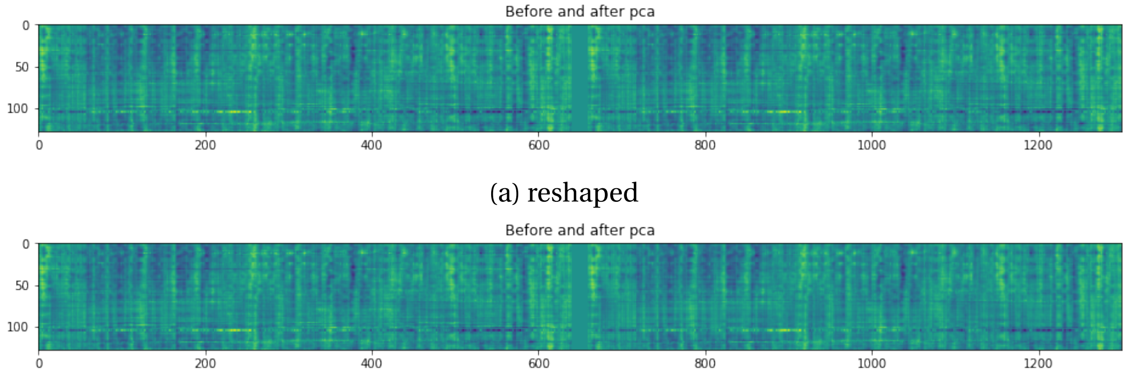
Table 1: Principal components explaining a set percentage of variance for PCA fit on a training dataset of the inner-speech condition data in the time frame 1s to 3.5s



(a) reshaped

(b) channel

Figure 3: Visualization of the difference in the data from sample 1 before PCA and reconstructed from PCA for the (a) reshaped condition, (b) channel condition



(a) reshaped

(b) channel

Figure 4: Visualization of the data from sample 1 before PCA and reconstructed from PCA for the (a) reshaped condition, (b) channel condition

4.3 K-Fold cross validation

The k-fold cross validation is a resampling procedure that is used to evaluate the performance of machine learning (ML) models on a very limited dataset. The parameter k defines how many times the data is split. Each of the splits will be used once as the validation split for the model which has been trained on the rest of the splits. With k-1 being the number of training splits and one remaining split is used to test the model after training. This is repeated n times, every time a randomly chosen split serves as testing data. The mean accuracy measure of the repeated training and test runs provides a metric that is more robust than the result of a single run, however it is important to report the standard deviation as the accuracies for every run may vary strongly (Refaeilzadeh, Tang, & Liu, 2009). This method was also used in the paper by van der Berg et al. (van den Berg et al., 2021).

5 Analysis

5.1 Fitting to Raw Data

As a baseline we tried several different ways to train the model on the original data without dimension reduction or data augmentation. This includes subject dependent models, subject independent models which are either pretrained or not. For all our experiments we decided to use EEGNet (Lawhern et al., 2018) as it is regarded as a model which performs good over a variety of EEG-related tasks(van den Berg et al., 2021). It should be noted, that our raw-data models where executed on a server using two Nvidia GTX-Geforce 980 (each 4GB memory). We implemented tensorflows Mirrored Strategy to utilize both available GPU's. To evaluate the raw models performance we decided to repeat a 4-fold cross validation 20 times, shuffling the dataset for every new 4-fold. However, we pretrain again before every 4-fold to average our results as best as possible.

EEGNet: The goal in subject dependent models is to build a model that only classifies the data of a single subject. It is important to note that the specific measurements vary a lot between subjects. By only focusing on a single target, we hope, the model can concentrate on the specifics of this one subject's brain activity. To compare the effect of using the complete interval versus only using the "action interval" we always trained the models on a non-filtered and a filtered dataset.

5.1.1 No Pretraining

This model is exclusively trained on a single subjects inner speech condition data. Here EEGNet is trained with a batch size of 20 and trained over 20 epochs. We used 64 as a kernel length and 40% dropout to prevent overfitting. Results in Table 2

5.1.2 Pretrain on participant's other two conditions

This model is pretrained on the "pronounced speech" and "visualized condition" data of the participant in question. Here EEGNet is trained with a batch size of 10, a kernel length of 64 and 40% dropout to prevent overfitting. The model is pretrained for 35 epochs, followed by 10 epochs with the subject "inner speech" data. Here we decided to freeze the first layer hoping to prevent overfitting. Results in Table 3.

5.1.3 Results

It immediately becomes clear that the interval used to train the models has an significant effect on the validation accuracy. Using the whole interval always results in a higher validation accuracy than only using the action interval that is to say, only giving the network access to the "inner speech" data and not the EEG-data related to the other intervals.

Without pretraining EEGNet achieves a validation accuracy of 28.3% when trained on the complete interval and a validation accuracy of 25.6% when trained exclusively on the action interval. The validation accuracy obtained by applying EEGNet to the whole interval is very similar to the validation accuracy obtained by van den Berg et al. (2021). This is to be expected but the validation accuracy of the model which was only trained on the action interval is close to chance. This indicates that the contextual data provided by the other intervals is the actual reason van den Berg et al. (2021) achieved a higher than chance accuracy.

The use of pretraining showed to be a valid way to classify the action interval of inner speech data. Again, training with all intervals performs better than only training on the action interval which performs above chance. Training on all intervals resulted in a mean validation accuracy of 29% and using only the action interval resulted in 26.7%. This indicates strongly that pretraining is a method that enables classification of the interval which contains inner speech EEG-data. Interestingly, the mean validation accuracy decreases after pretraining, both for the model trained on all intervals and the model trained on the action interval (see Table 3). This can only be explained by overfitting of the model after training on the inner speech data.

part.	**no pretraining with context**	**no pretraining without context**
1	0.293 (± 0.051)	0.28 (± 0.059)
2	0.277 (± 0.053)	0.264 (± 0.056)
3	0.269 (± 0.059)	0.252 (± 0.065)
4	0.271 (± 0.055)	0.226 (± 0.047)
5	0.272 (± 0.06)	0.258 (± 0.052)
6	0.283 (± 0.051)	0.238 (± 0.062)
7	0.29 (± 0.053)	0.266 (± 0.056)
8	0.306 (± 0.061)	0.267 (± 0.062)
9	0.306 (± 0.047)	0.256 (± 0.049)
10	0.264 (± 0.053)	0.249 (± 0.047)
mean	0.283 (± 0.056)	0.256 (± 0.058)

Table 2: Cross validation for no subject dependent pretraining model only trained on condition "inner speech"

5.2 PCA on Reshaped data

The first step in examining the viability of PCA reduced data for EEG-data classification was to train a model on the reshaped PCA data. The PCA on the reshaped data was implemented

part.	**pretraining with context**		**pretrain without context**	
	pretrain	pretrain + train	pretrain	pretrain + train
1	0.31 (± 0.055)	0.294 (± 0.053)	0.276 (± 0.053)	0.27 (± 0.061)
2	0.326 (± 0.035)	0.25 (± 0.043)	0.248 (± 0.058)	0.248 (± 0.053)
3	0.298 (± 0.039)	0.263 (± 0.069)	0.302 (± 0.051)	0.265 (± 0.064)
4	0.286 (± 0.061)	0.283 (± 0.06)	0.224 (± 0.052)	0.236 (± 0.054)
5	0.304 (± 0.039)	0.306 (± 0.054)	0.264 (± 0.046)	0.276 (± 0.048)
6	0.364 (± 0.034)	0.251 (± 0.056)	0.358 (± 0.026)	0.261 (± 0.04)
7	0.274 (± 0.05)	0.319 (± 0.051)	0.312 (± 0.039)	0.259 (± 0.048)
8	0.328 (± 0.046)	0.351 (± 0.062)	0.262 (± 0.052)	0.282 (± 0.056)
9	0.382 (± 0.062)	0.295 (± 0.053)	0.336 (± 0.042)	0.308 (± 0.051)
10	0.342 (± 0.06)	0.292 (± 0.057)	0.292 (± 0.034)	0.263 (± 0.051)
mean	0.321 (± 0.06)	0.29 (± 0.064)	0.287 (± 0.06)	0.267 (± 0.056)

Table 3: Cross validation accuracies across participants for dependent models, pretrainend on the other two conditions of the subject. Training progress in Figure 9.

as part of the training itself. PCA was fit and applied to the training and test data before they were transformed into tensors and further preprocessed. The event data was transformed into tensors. During preprocessing we transformed the PCA data into float32 and added a new dimension and the event data was one hot encoded before the dataset was shuffled, batched and prefetched.

5.2.1 Pretraining

Since the amount of data is quite small, we decided to make use of transfer-learning and pretrained our model on the pronounced speech and visualized conditions, using the same time window between 1 and 3.5 seconds and the same amount of channels for the PCA as we do in the training. We did not cross-check, how much information was lost by applying the PCA with the components optimized for the inner speech condition on the pretraining data since we simply wanted to familiarize the model with PCA transformed EEG data. To accommodate the components size of 1319 we split the pretraining dataset into 50-50. The pretraining batch size was 15. Training happened for 50 Epochs.

5.2.2 Training

The training data consist only of the samples of the inner speech condition for the time window between 1 and 3.5. Since we need at least 1319 samples to apply PCA on the reshaped data we do not perform cross validation. Instead we use the use a validation split where we validate the data once on random 10 Percent of the training data. Training lasted 30 Epochs with a batchsize of 10 since the amount of data was quite low. Since the sample size of the individual subjects was to small to apply the PCA, no subject-dependent training happened.

5.2.3 Feed Forward Net

Model: For the reshaped data we trained a simple Feed-Forward Network(FFN) comprised of two dense layers followed by a drop out layer each and another two dense layers. The activation function for all dense layers, but the last was ReLu. The last layer, the output layer, had a softmax activation function for classification. The number of units in the input layer was 128, the next dense layer had 64 units and the last layer 16. The output layer had a unit size of 4. The first dropout layer has a droupout-rate of 0.5, the second a rate of 0.3.Since we have a classification task with 4 categories one hot encoded, we chose categorical crossentropy as loss function. The optimizer was Adam.

Results: Pretraining the model resulted in a training accuracy above 0.4 though the validation accuracy stayed close to chance. Looking at the loss we can see that the model strongly overfitted onto the training data (see Figure 5) even though we had a 50/50 train/test split. This leads to the assumption that the model most likely learned the training data by heart rather than the information necessary for classification.

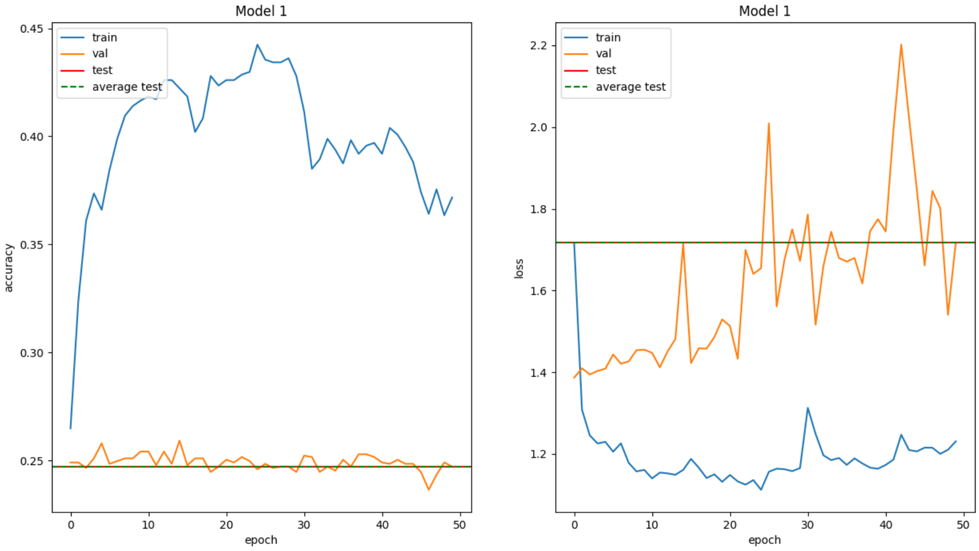


Figure 5: Pretraining of the Feed Forward model for 50 Epochs with subplots for accuracy (left) and loss (right)

Training and classifying the training data on the pretrained model led to no results. Both training and validation accuracy danced around chance (see Figure 6).

Though pretraining itself was successful, it seems to have no later effect. Training the model directly on the training data led to similar results as training the model on the pretrained model (see Table 4. The training accuracy danced around 0.25 and the loss stayed near constant.

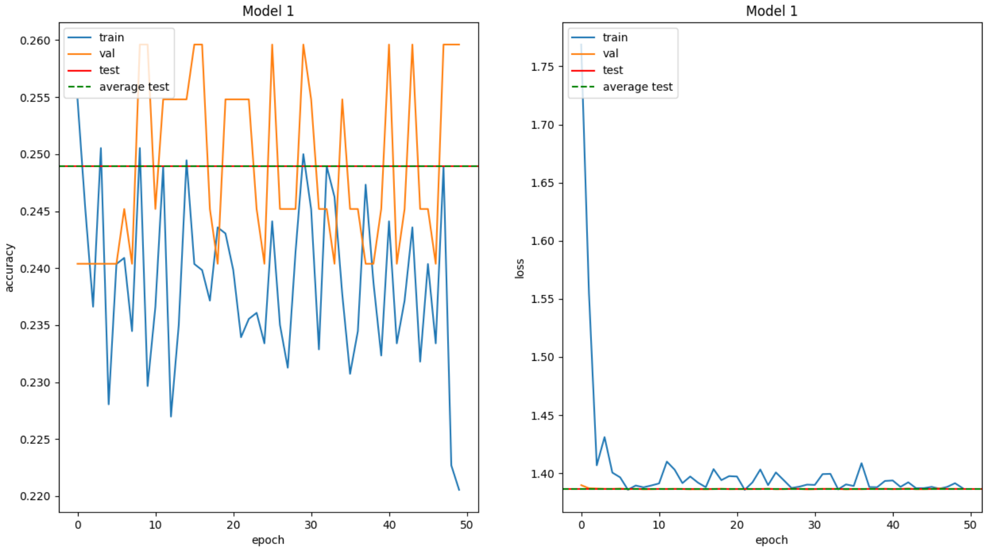


Figure 6: Training on the pretraining of the Feed Forward model for 50 Epochs with subplots for accuracy (left) and loss (right)

participant	**no pretraining**	**50 epochs pretraining**
all	0.248	0.249

Table 4: Cross validation accuracy with standard deviation across 10 trials across all subjects for training with the Feed Forward Net

As such we assumed that the model is inadequate to classify our amount of the PCA transformed reshaped inner speech EEG-data and stopped trying to improve the model. The failure to learn may be caused by the small size of the inner speech dataset. During successful pretraining we had more than double the amount of data.

5.3 PCA ON CHANNEL DATA

Reshaping the data seems to make it loose information which is important for proper classification. As such we applied in a second approach PCA directly onto the channel*time data for each trial.

5.3.1 PRETRAINING

Adopting the approach for the PCA on reshaped data, we tried again to pretrain our model on the pronounced speech and visualized conditions. The time window and the amount of channels for the PCA are the same as the one used during training. Since the number of channels is constant we could split the data into a training (0.75) and a test (0.25) set of our choice. To make sure all event types were equally distributed we used the sklearn method stratifiedShuffleSplit to gain the training and test set. Since the amount of data for the pretraining was roughly double the amount of the inner speech data we started by using a slightly larger batchsize of 15 and trained for 30 epochs.

5.3.2 Training

We used k-fold cross validation to examine our model using the pca transformed inner speech data. To account for variance in the data we used 10 folds, leaving us with 10 training and test splits of size 0.9 and 0.1. To make sure that the results are not due to the order of the participants and the event categories are equally distributed among folds the sklearn method StratifiedKFold with shuffle enabled to create the 10 splits. Enabling shuffle makes sure that the participants data is shuffeld before the split and the StratifiedKFold compared to KFold makes sure that the events are equally distributed among the splits. We furthermore set no random_state value to make sure that the splits are not uniform each time we run the training method for all data. While this makes it hard to directly compare the results of two trainings it makes sure that no random allotment of splits makes the cross validation especially good. The batchsize used for training was 10 and the model was trained for 30 epochs. In a second step we examined each participant individually, loading only the inner speech data of one participant respectively, before training the model using k-fold cross validation the same way as on the whole data.

5.3.3 EEGNet

Since we train the raw data on the EEGNet we want to try to train the PCA reduced data on the EEGNet as well. We assume we wont achieve great increases in accuracy, since the model seems to work well for EEG data and the PCA transformation is likely to decrease the EEG-characteristics of the data that might enhance the work of the model. Still, it is interesting to see if some of the EEG-characteristics remain and aid in training an EEG data focused model like the EEGNet.

Model: To accommodate that we do no longer have clean EEG data but compressed EEG data we only use 64 kernels instead of the advised 128 kernels. As a dropout-factor we use 0.3 and no spatial dropout. As a loss function we used categorical crossentropy, a standard loss for non-binary classification problems. To be able to update the learning rate while training we chose to use Adam as an optimizer and not standard gradient descent. Furthermore, we increased the learning rate from the 0.001 standard for Adam to 0.005 after several short trial runs to gentle the loss slope.

Results: Pretraining the EEGNet did not yield significant results. The validation accuracy was with 0.259 close to chance while the actual training accuracy rose to around 0.3. Taken together with the rising loss function it can be said that the model seems to have a problem with overfitting (see Figure 7). This might be a problem due to the similarity of the EEG-data across trial types and events. Reducing the dimensions with PCA might actually exacerbate this problem. We still used the pretrained model to train the inner speech data set, since we only wanted the model to be familiarized with EEG data.

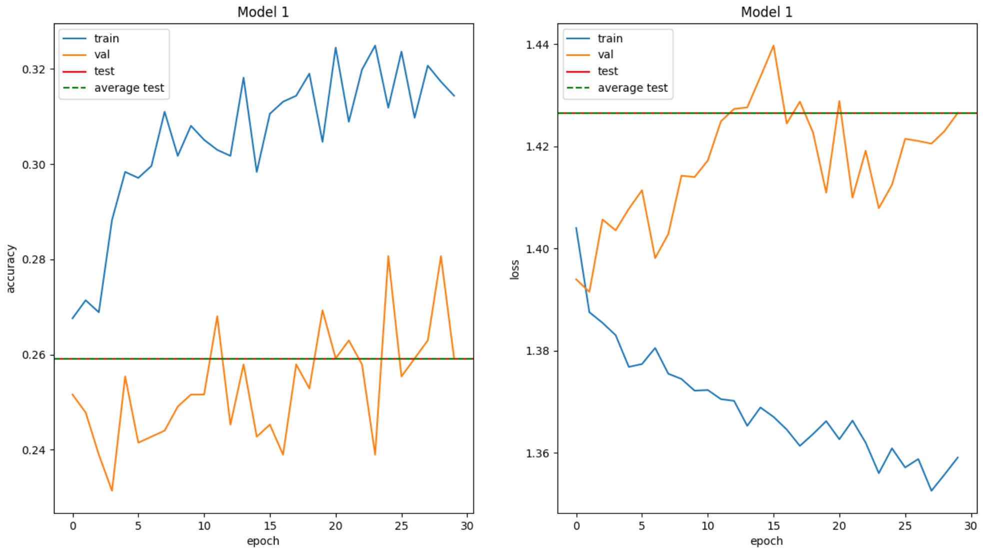


Figure 7: Pretraining of the EEGNet model for 30 Epochs with subplots for accuracy (left) and loss (right)

Using the whole training data, the cross validation accuracy for training and classification on the pretrained model shows no increase in accuracy compared to the validation accuracy of the pretraining. Overall, it seems to actually be slightly worse (see Table 5). Taking a look at the 10 folds training (see Appendix: Figure 10) we again see a slight trend towards overfitting, since the validation loss increases from around 1.4 to around 1.5 while the training loss decreases from above 1.4 to around 1.3. The training accuracy seems to reach around 0.4 while the validation accuracy stays around chance. A reason for this might be that the model is to big and does not really learn to differentiate the images but learns the images itself. A second problem might be the similarity of the data samples. As mentioned before, all four categories show similar activation patterns which makes it hard for the model to learn the differences.

Taking a look at the training and classification per subject the pretrained model performs even worse. The cross validation accuracy is below or around chance (see Table 5) while the training accuracy comes close to 1 most of the time. Taking a look at the loss we can see a validation loss usually above 2 and slightly increasing, while the training loss converges to close to 0 which indicates that the model is overfitting. The standard deviation increases showing that the range of validation accuracy across folds increases compared to the range of validation accuracy across all training data. This might actually be due to the small sample size of training and testing data for each subject. The average number of samples is only 200 leading to a training and testing dataset of 190 to 10 for each fold. Interestingly, we saw an increase in validation accuracy of more than 0.2 in participant 8 and 10. Participant 8 showed the second largest cross validation accuracy and smallest standard deviation across all subjects. This is partially in line with previous findings of van den Berg et al. (2021) that mention training the data on participant 8 led to the best accuracy and lowest standard deviation.

participant	no pretraining	30 epochs pretraining
1	0.25 (± 0.123)	0.25 (± 0.081)
2	0.258 (± 0.081)	0.225 (± 0.099)
3	0.289 (± 0.147)	0.244 (± 0.122)
4	0.254 (± 0.068)	0.196 (± 0.077)
5	0.288 (± 0.092)	0.25 (± 0.053)
6	0.296 (± 0.106)	0.254 (± 0.087)
7	0.229 (± 0.086)	0.25 (± 0.081)
8	0.245 (± 0.082)	0.27 (± 0.046)
9	0.263 (± 0.078)	0.244 (± 0.106)
10	0.225 (± 0.113)	0.288 (± 0.08)
mean	0.26	0.247
all	0.243 (± 0.03))	0.245 (± 0.02)

Table 5: Cross validation accuracy with standard deviation across 10 folds for each viable subject and across all subjects for training with the EEGNet

Finally, we also examined the cross validation accuracy for each subject and overall for the training data on the untrained model and found similar effects to the pretrained model (see Table 5). The validation accuracy, with which the untrained model predicts the overall accuracy for the training data, is around chance. The training accuracy ranges from close to 0.3 to near 0.4 for the 10-folds (see Appendix: Figure 11). The model seems to slightly overfit as the validation loss increases from around 1.4 to 1.5 while the training loss decreases from 1.4 to 1.3 (see Appendix: Figure 11).

Differences to the pretrained model appear in the subject dependent training. It shows mostly an increase in cross validation accuracy leading to validation accuracies slightly above chance compared both to subject-dependent training on the pretrained model and the general training on the untrained base model (see Table 5). Interestingly subjects that performed quite well on the pretrained model performed worse on the not pretrained model and vice versa. The training accuracy ranges most of the time between 0.75 and 1 and is on average slightly lower than for the subject-dependent pretrained model. The validation loss stays mostly constant above or around 2 while the training loss decreases to close to 0. This indicates similar to the pretrained model a slight overfitting.

If we compare the mean of the subject-dependent to the subject independent training we see no significant difference in validation accuracy for the pretrained model (see Table 5). For the model without pretraining we see a minimal increase in validation accuracy for mean of the subject-dependent validation accuracies compared to the validation accuracy of training the model with all participants. This effect is quite small, but might mean that the model finds it hard to distinguish between the events for all subjects together, due to the inter-subject differences in the EEG-data. Training on only one subject makes sure that subject dependent noise stays similar across all trials, so the model can ignore it more easily.

In general we can say that our assumption that the EEGNet is not suitable to train the PCA transformed EEG data on seems to hold. Only subject-dependent training leads to validation accuracies slightly above chance. Furthermore, there seemed to be no clear effect of pretraining on the validation accuracies. It only seems to have a minimal effect on the standard deviation leading to a more narrow curve of validation accuracies across the folds. Though this effect does not hold for all subjects it is more pronounced in the subject-dependent testing.

5.3.4 Convolutional Neural Network

The original EEG-data could easily be represented in images of size channel*time. Reducing the channels left us with a smaller image. As such we chose to train a small CNN, which has proven to be successful at classifying our channel data. We hoped to achieve at least a cross validation accuracy of 29%, similar to the results achieved by van den Berg et al. (2021) with the EEGNet.

Model The network consists of two blocks made up of a convolutional layer followed by a batch normalization layer, a max pooling layer and a dropout layer. The convolutional layers have a kernel size of 3 and same-padding and differ in the filter size. The first convolutional layer has 64 filters and the second 16. The first dropout layer had adropout rate of 0.6 which was halved for the second dropout layer. The first dropout layer has double the amount of droput compared to the second. Before the output of those blocks is classified in a Dense layer with four units using a softmax activation function it runs through a leaky ReLu and is flattened. In this case leaky ReLu is used due to a slight increase in validation performance compared to a sigmoid activation and for its property to counter vanishing gradients. As a loss function we used categorical crossentropy, a standard loss for non-binary classification problems. To be able to update the learning rate while training we chose to use Adam as an optimizer and not standard gradient descent. Hyperparameters were first chosen arbitrarily and then subsequently adapted in several trial runs, where we changed parameters and trained a subset of two to five participants for 15 epochs to check if the changes lead to training improvement.

Results The pretraining for 50 Epochs did not yield significant results with a final validation accuracy of 0.238. While the training accuracy seems to rise above chance around 0.28, the validation accuracy only shows a small increase to around 0.26 (see Figure 8). The training loss indicates a small but steady decrease while the validation loss peaks at around 10 Epochs before leveling around 1.39. We overall liked the network trend towards bot rising training and validation accuracy and decreasing training loss, though the level validation loss might indicate a problem with overfitting and the accuracy was actually quite low for 50 Epochs compared to the initial testing. We assumed that the low accuracy was due to the larger amount of data used compared to pretraining but initially did not see any problems since the accuracy seemed to increase.

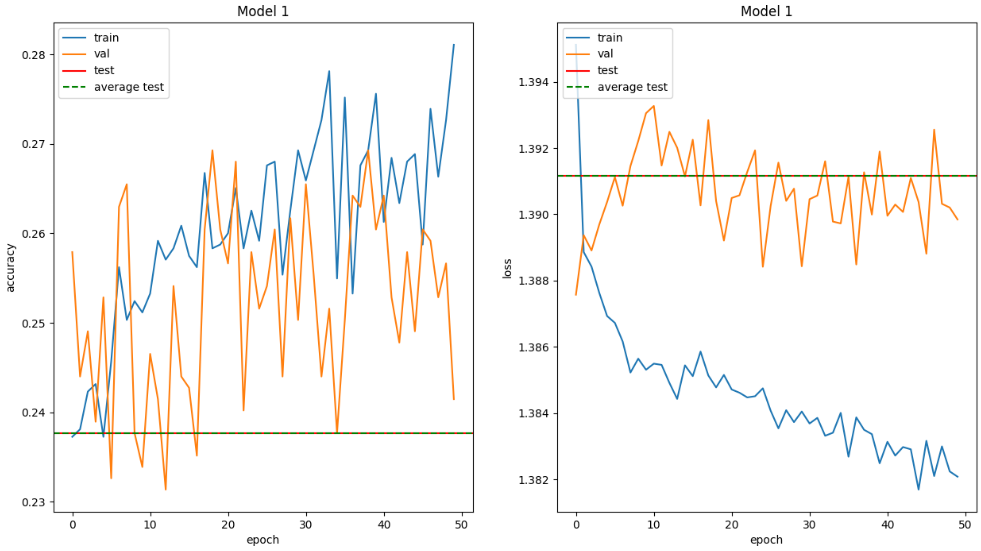


Figure 8: Pretraining of the CNNet model for 30 Epochs with subplots for accuracy (left) and loss (right)

Using the pretrained model to fit the inner speech data lead to no training success at all. Learning accuracy and loss stayed more or less constant around chance and 12 respectively across the epochs for all 10 folds (see Appendix: Figure 12). The model seemed incapable of adapting to the training data after the pretraining. Subject-dependent training on the pretrained model showed no significant results either (see Table 6). Validation accuracy ranged from below chance with 0.2 for subject 8 to slightly above chance with 0.278 for subject 6. The standard deviation ranged from 0.05 to nearly 0.11 showing a rather narrow range of validation accuracy.

Interestingly, though we had initial success during the model testing, training the model directly led to no significant results with the validation accuracy staying close to chance across the k-fold training (see Appendix: Figure 13). The training accuracy increases minimal but stays around chance. The training loss decreases mostly slightly from below 1.39 to below 1.385 while the validation loss shows no clear trend toward increasing or decreasing but stays between 1.38 and 1.4. Together this indicates that the learning rate chosen might have been to small. Comparing the cross validation accuracy to the cross-validation accuracy for training on the pretrained model shows a minimal decrease in training accuracy. This indicates that pretraining had no real effect on the subsequent training.

Subject-dependent training does not increase the accuracy significantly. There seems to be a general increase in cross validation accuracy compared to the pretrained model and standard deviation, but this does not hold for all subjects (see Table 6). Overall, the training accuracy ranges between 0.23 and 0.3. Taking a look at the training accuracy we can see a range from 0.2 to 0.4. Together with the decrease of training loss from around 1.4 to around 1.35 this is indicating that the model learns something. This only sight increase of accuracy and the minimal downwards slope of the loss during training with the model directly might indicate the choice of a learning rate that is to low. The validation loss shows no clear trend so that it is not possible to say that the model is overfitted.

participant	**no Pretraining**	**50 Epochs Pretraining**
1	0.3 (± 0.087)	0.23 (± 0.064)
2	0.246 (± 0.051)	0.267 (± 0.077)
3	0.261 (± 0.119)	0.272 (± 0.0631)
4	0.246(± 0.096)	0.25 (± 0.078)
5	0.296 (± 0.078)	0.26 (± 0.095)
6	0.242 (± 0.111)	0.278 (± 0.051)
7	0.267 (± 0.073)	0.217 (± 0.076)
8	0.23 (± 0.078)	0.2 (± 0.05)
9	0.256 (± 0.095)	0.269 (± 0.108)
10	0.288 (± 0.075)	0.23 (± 0.105)
mean	0.2632	0.247
all	0.242 (± 0.028)	0.249(± 0.023)

Table 6: Cross validation accuracy with standard deviation across 10 folds for each viable subject and across all subjects for training with the simple CNN

Comparing he mean cross validation accuracy of the participants to the overall cross validation loss of the whole data we can see for the model without pretraining that averaging subject-dependent training leads to the better cross validation accuracy, while for the pretrained model training across all subjects at once seemed to have worked better (see Table 6). Moreover, the difference is larger for the model without pretraining than for the model with pretraining.

Overall, it can be said that most likely the model was unable to properly learn the training data, though it does seem to work for a small subset of the training data and shows a preliminary success for the pretraining. As such the bad results are most likely the choice of inadequate hyperparameters together with a maybe too shallow network structure for the given data.

5.4 PCA - Model Comparison

Directly comparing the three models used to classify two types of PCA reduced EEG data leads to the conclusion that there were no significant effects of the PCA reduction type, pretraining or the network on the classification success. All models showed performance around chance level (see Figure 7). All models pretrained reasonably well with the EEGNet achieving the highest validation accuracy of 0.259. Looking at the training with and without pretraining the FFNet on the reshaped data performed the best with a validation accuracy of 0.248 and 0.249 respectively.

model	Pretraining	Pretraining + Training	only Training
FFNet	0.247	0.248	0.249
EEGNet	0.259	0.245 (± 0.02)	0.243 (± 0.03)
SimpleConv	0.242 (± 0.028)	0.246 (± 0.03)	0.242 (± 0.028)

Table 7: Cross validation accuracy of the 3 trained networks across 10-fold training for the whole training data after pretraining, pretraining with training and after only training

6 Discussion

6.1 Raw Data

Training EEGNet with raw EEG-data leads to a validation accuracy higher than chance. Interestingly, training our models results in higher accuracy when training on all intervals instead of only training the model on the action interval. For this effect there are two explanations that both make sense. For one it could be that the cue interval contains some meaningful patterns related to the arrow that was displayed on the screen. The other explanation is that the other intervals provide a kind of baseline of the subjects brain activity which helps the model to filter out meaningful patterns. Generally, we believe that there is room for improvement in validation accuracy by exploring hyperparameters. We initially based our hyperparamters on processor capacities and visual data exploration. During testing of our model we only adapted them minimally, due to the fact that a more profound search of hyperparameters was not compatible with time constraints.

6.2 PCA

Applying PCA does not seem to improve the overall classification performance compared to classifying the raw data, even though we project the data into lower dimensions that better show the variance. This might be because applying PCA changes the original data to a degree that makes it harder for the model to distinguish the small alternations that differentiate the event categories. Moreover, the distinguishing features may be closely distributed and as such not picked up in our number of components. The same applies for inter subject differences. General training on the data of only one person leads to slightly better results. This may be because the number of components are not bloated by interpersonal differences in the EEG data. Even though the data is cleaned and standardized these differences might actually be larger than the actual differences encoding the inner speech and as such explain more variance. This leads to the problem that the network actually learns the differences between the participants while trying to label them according to the event types. It may explain, why the accuracy mostly stagnated around chance as there are 10 participants and only event labels. An idea for the future would be to change the units in the output layers to 10 and try to classify the subjects to examine whether the hypothesis holds. Another general problem we kept running into is the problem of overfitting. This is likely due to the small amount of data and the high similarity of the EEG-Data.

Finally, we had problems developing an adequate model ourselves that can predict the PCA transformed data. Besides problems with the data this might be because we did not conduct rigorous exploration of network size and hyperparameters. We rather arbitrarily choose a small network since the we did not have a large amount of data and preliminary visual analysis of the data itself showed no complicated patterns. This might make it hard for the model to learn the underlying information to categorize the data. Furthermore, we reduced the size of paramteres continuously during trying to optimize the hyperparameter settings before training the settings on only a small sub-sample of the data for few epochs. Good preliminary results may actually have been the result of a good setting of the randomly initialized weights and a fitting subset of participants. As subject dependent training has shown the models generalizes better to some participants than others we might have had a good combination during short hyperparameter testing but as the results of the CNN on the PCA with channel data shows that must not indicate that it generalizes well to the entire data.

7 Outlook

Our approach to explore possibilities to raise the classification accuracy on the inner speech data of Nieto et al. (2021) has been inconclusive, but there are still many possibilities to exhaust in the future. As we performed little hyperparameter-search this would be the most obvious way to improve the accuracy of our models. Moreover, since overfitting is an inherent problem when working with little high-dimensional data, it might be useful to investigate methods that counter this problem. Some promising methods include early stopping, data augmentation and autoencoders. Early stopping prevents overfitting by stopping the training process when the training or test metrics fulfill some condition. Data augmentation is another technique which prevents overfitting and potentially leads the model to learn a better representation of the data. Here the difficulty is to choose the right augmentations. One final approach that could replace PCA as a way to reduce dimensionality is the use of an autoencoder to encode the dataset into it's latent space. This tackles the curse of dimensionality by compressing the data into a meaningful compact form. Although these methods looked promising time contraints prevented us investigating these methods in detail and providing sufficient comparable results. When testing data augmentation techniques we decided to implement a mix of up-down, left-right data flipping. Furthermore we experimented with heavy noise addition, for instance perlin noise to introduce structured noise into the data.

8 Conclusion

The goal of this project was to achieve a more than random classification accuracy on the "inner speech" condition of the Nieto et al. (2021) "Inner Speech Dataset". A previous paper by van den Berg et al. (2021) working with the same dataset achieved a mean of 29.7% over all subjects in an participant dependent approach. As van den Berg et al. (2021) used the whole time interval of 4.5 seconds it is not clear if this higher than random accuracy is due to the participants inner speech or other parts of the data. Our experiments suggest that the higher than chance accuracy achieved by van den Berg et al. (2021) might be largely attributable to the influence of context information. Using a model that is pretrained on the other EEG-data of the other conditions we where able to achieve an accuracy of 26.7%. This shows that it is theoretically possible to predict the data only using the interval containing inner speech. Sadly, trying to reduce the dimensionality using PCA before training did not lead to any conclusive results.

REFERENCES

Abbasi, S. F., Ahmad, J., Tahir, A., Awais, M., Chen, C., Irfan, M., … others (2020). EEG-based neonatal sleep-wake classification using multilayer perceptron neural network. *IEEE Access*, *8*, 183025–183034.

Alebiosu, D. O., & Muhammad, F. P. (2019). Medical Image Classification: A Comparison of Deep Pre-trained Neural Networks. In *2019 ieee student conference on research and development (scored)* (pp. 306–310).

Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. In *Proceedings of icml workshop on unsupervised and transfer learning* (pp. 17–36).

Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2018). EEGNet: a compact convolutional neural network for eeg-based brain–computer interfaces. *Journal of neural engineering*, *15*(5), 056013.

Nieto, N., Peterson, V., Rufiner, H. L., Kamienkowski, J. E., & Spies, R. (2022). Thinking out loud, an open-access EEG-based BCI dataset for inner speech recognition. *Scientific Data*, *9*(1), 1–17.

Nieto, N., Rufiner, H. L., & Spies, R. (2021). Preliminary feasibility analysis of inner speech as a control paradigm for brain-computer interfaces.

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. *Encyclopedia of database systems*, *5*, 532–538.

Singla, A., Yuan, L., & Ebrahimi, T. (2016). Food/non-food image classification and food categorization using pre-trained googlenet model. In *Proceedings of the 2nd international workshop on multimedia assisted dietary management* (pp. 3–11).

Tsai, H.-C., Zhang, F., Adamantidis, A., Stuber, G. D., Bonci, A., De Lecea, L., & Deisseroth, K. (2009). Phasic firing in dopaminergic neurons is sufficient for behavioral conditioning. *Science*, *324*(5930), 1080–1084.

van den Berg, B., van Donkelaar, S., & Alimardani, M. (2021). Inner Speech Classification using EEG Signals: A Deep Learning Approach. In *2021 ieee 2nd international conference on human-machine systems (ichms)* (pp. 1–4).

9 Appendix

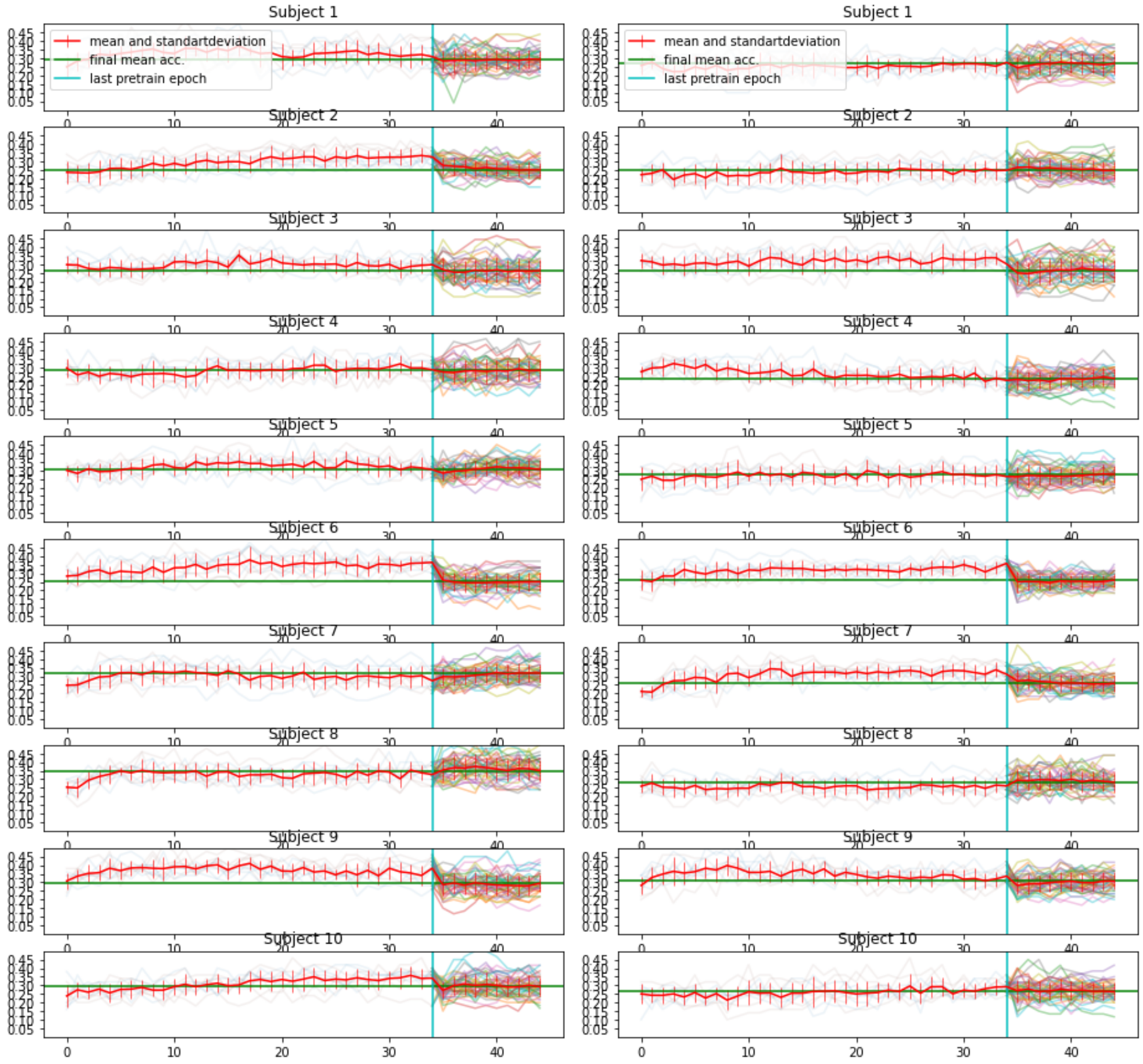


Figure 9: Subject Dependent pretraining on two other conditions of the subject. Training on complete interval (left) and training only on action interval (right)

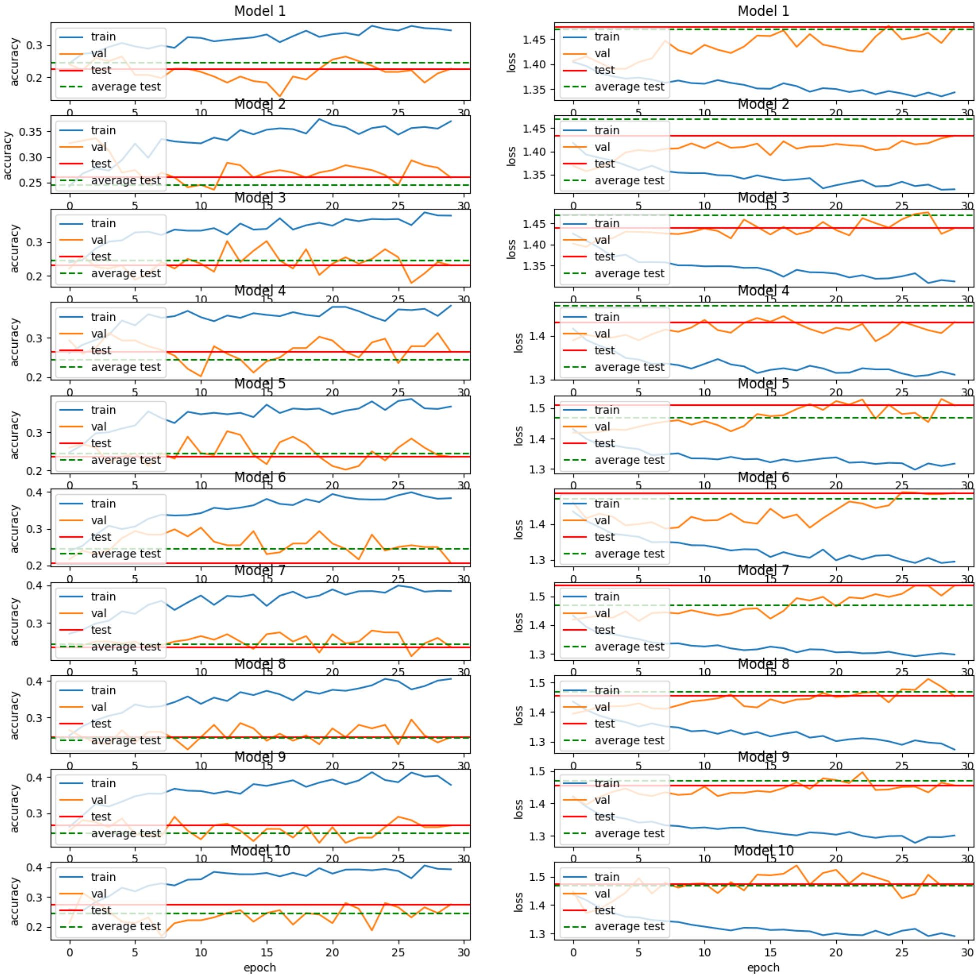


Figure 10: Loss (right) and accuracy(left) for training and 10-fold cross validation of channel reduced inner speech data on the pretrained EEGNet

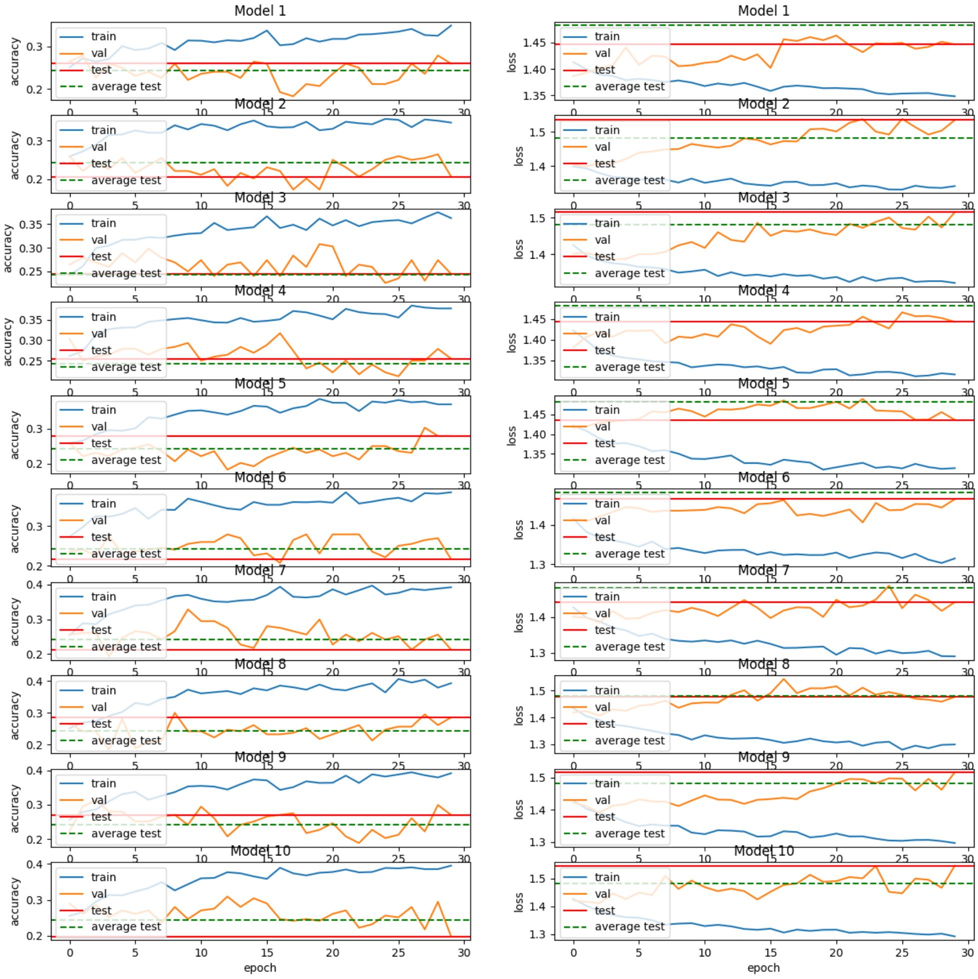


Figure 11: Loss (right) and accuracy(left) for training and 10-fold cross validation of channel reduced inner speech data on the EEGNet

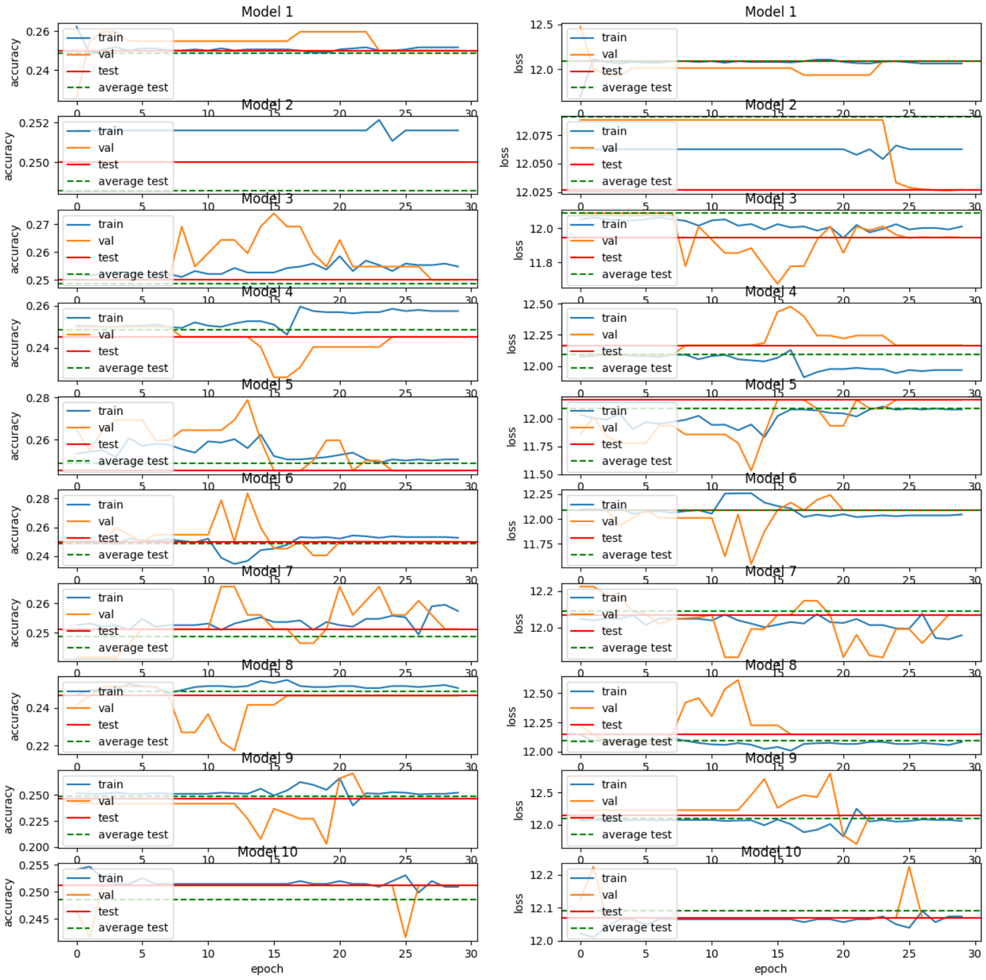


Figure 12: Loss (right) and accuracy(left) for training and 10-fold cross validation of channel reduced inner speech data on the pretrained CNNet

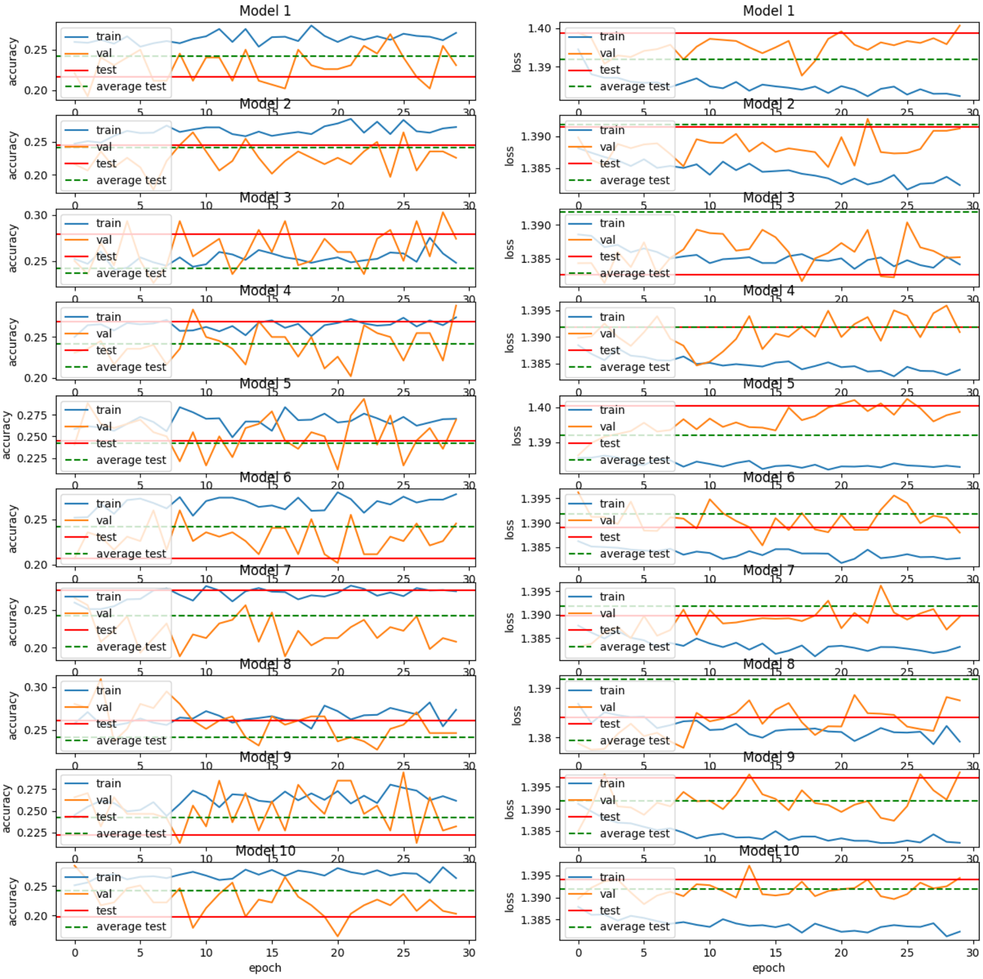


Figure 13: Loss (right) and accuracy(left) for training and 10-fold cross validation of channel reduced inner speech data on the CNNet