LEARNING DISCRIMINATIVE FEATURES FROM ELECTROENCEPHALOGRAPHY RECORDINGS BY ENCODING SIMILARITY CONSTRAINTS

Sebastian Stober

Research Focus Cognitive Sciences, University of Potsdam, Germany
sstober@uni-potsdam.de

ABSTRACT

This paper introduces a pre-training technique for learning discriminative features from electroencephalography (EEG) recordings using deep neural networks. EEG data are generally only available in small quantities, they are high-dimensional with a poor signal-to-noise ratio, and there is considerable variability between individual subjects and recording sessions. Similarity-constraint encoders as introduced in this paper specifically address these challenges for feature learning. They learn features that allow to distinguish between classes by demanding that encodings of two trials from the same class are more similar to each other than to encoded trials from other classes. This tuple-based training approach is especially suitable for small datasets. The proposed technique is evaluated using the publicly available OpenMIIR dataset of EEG recordings taken while participants listened to and imagined music. For this dataset, a simple convolutional filter can be learned that significantly improves the signal-to-noise ratio while aggregating the 64 EEG channels into a single waveform.

Index Terms— EEG, Music Perception, Feature Learning

1. INTRODUCTION

Over the last decade, deep learning techniques have become very popular in various application domains such as computer vision, automatic speech recognition, natural language processing, and bioinformatics where they produce state-of-the-art results on various tasks. At the same time, there has been very little progress investigating the application of deep learning in cognitive neuroscience research, where these techniques could be used to analyze signals recorded with electroencephalography (EEG) – a non-invasive brain imaging technique that relies on electrodes placed on the scalp to measure the electrical activity of the brain. EEG is especially popular for the development of brain-computer interfaces (BCIs), which work by identifying different brain states from the EEG signal.

Working with EEG data poses several challenges. Brain waves recorded in the EEG have a very low signal-to-noise ratio and the noise can come from a variety of sources like electrical surroundings, muscle activity, eye movements, or blinks. Usually, only certain brain activity is of interest, and this signal needs to be separated from background processes. EEG lacks spatial resolution on the scalp with additional spatial smearing caused by the skull but it has a good (millisecond) time resolution to record both, slowly and rapidly changing dynamics of brain activity. Hence, in order to identify the relevant portion of the signal, sophisticated analysis techniques are required that should also take into account temporal information.

Furthermore, no matter how much effort one puts into controlling the experimental conditions during EEG recordings, there will always be some individual differences between subjects and between recording sessions. This can make it hard to combine recordings from different subjects to identify general patterns in the EEG signals. A common way to address this issue is to average over many very short trials such that differences cancel out each other. When this is not feasible because of the trial length or a limited number of trials, an alternative strategy is to derive signal components from the raw EEG data hoping that these will be stable and representative across subjects.

This is where deep learning techniques could help. For these techniques, training usually involves the usage of large corpora. As EEG data are high-dimensional[1] and complex, this also calls for large datasets to train deep networks for EEG analysis and classification. Unfortunately, there is no such abundance of EEG data. Unlike photos or texts extracted from the Internet, EEG data are costly to collect and generally unavailable in the public domain. It requires special equipment and a substantial effort to obtain high quality data. Consequently, EEG datasets are only rarely shared beyond the boundaries of individual labs and institutes. This makes it hard for deep learning researchers to develop more sophisticated analysis techniques tailored to this kind of data.

This paper introduces a pre-training technique called *similarity-constraint encoding* that addresses the common challenges of working with EEG data described above. It is able to learn discriminative features from a small dataset with high data dimensionality and a bad signal-to-noise ratio. After a brief review of related work in Section 2, the proposed pre-training technique is introduced in Section 3. Further, Section 4 describes an experiment that demonstrates a successful application of similarity-constraint encoding using the publicly availble OpenMIIR dataset. Results are discussed in Section 5. Section 6 concludes the paper.

[1] A single trial comprising ten seconds of EEG with 64 channels sampled at 100 Hz has already 64000 dimensions and the number of channels and the sampling rate of EEG recordings can be much higher than this.

2. RELATED WORK

The potential of deep learning techniques for neuroimaging has been demonstrated very recently for functional and structural magnetic resonance imaging (MRI) data [1]. Prior applications of deep learning techniques on EEG data comprise epileptic seizure prediction using convolutional neural networks (CNNs) [2], detecting anomalies related to epilepsy using deep belief nets (DBNs) that process individual "channel-seconds", i.e., one-second chunks from a single EEG channel [3], and classifying different sleep stages using DBNs combined with hidden Markov models (HMMs) [4]. Furthermore, there have been some applications of CNNs for BCIs such as classifying steady-state visual evoked potentials (SSVEPs), i.e., brain oscillation induced by visual stimuli, by integrating the Fourier transform between convolutional layers [5] and detecting P300 waves (a well established waveform in EEG research) [6]. There has also been early work on emotion recognition from EEG using deep neural networks [7, 8]. In our earlier work, we applied stacked denoising auto-encoders (SDAs) and CNNs to classify EEG recordings of rhythm perception and identify their ethnic origin – East African or Western – [9] as well as to distinguish individual rhythms [10].

Our pre-training technique proposed here is further related to siamese networks [11] and triplet networks [12]. Both methods process multiple input instances in parallel using identical pipelines and then try to optimize distances in the embedding space such that instances belonging to the same class are close to each other – in contrast to instances from other classes. Siamese networks consider the *absolute* (Manhattan) distance between input pairs using the L1-norm as distance measure and target values of 0 and 1 for pairs belonging to the same or other classes respectively. Triplet networks instead consider input triplets comprising a reference instance as well as a more similar instance (same class) and a less similar instance (different class). Here, the L2-norm (Euclidean distance) is applied in the embedding space. Our proposed approach differs in that we do not aim to learn a distance metric or an embedding into Euclidean space. We are looking for distinctive time series using the dot product for comparison. The rationale behind this is explained in the following section. These differences are crucial for the success of our approach as the experiment in Section 4 shows.

3. SIMILARITY-CONSTRAINT ENCODING

The idea of similarity-constraint encoding (SCE) is derived from auto-encoder pre-training [13]. Usually, EEG trials recorded during an experiment belong to different conditions, which are often used as class labels to train a classifier. Demanding that trials belonging to the same condition are encoded similarly facilitates learning features representing brain activity that is stable across trials. Features to be used in classification tasks should furthermore allow to distinguish between the respective classes. This can be achieved by a training objective that also considers how trials from other classes are encoded.

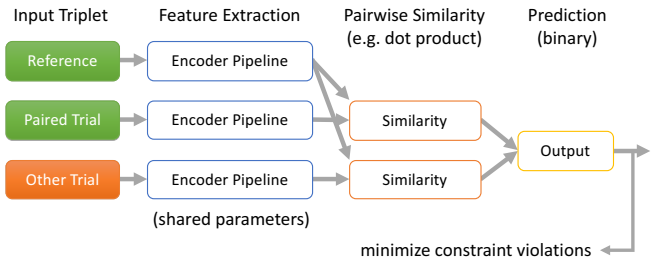

Fig. 1. Processing scheme of a similarity-constraint encoder.

In the most basic form, the encoded representations of two trials belonging to the same class are compared with an encoded trial from a different class. The desired outcome of this comparison can be expressed as a *relative similarity constraint* as introduced in [14]. A relative similarity constraint (a,b,c) describes a relative comparison of the trials a, b and c in the form "a is more similar to b than a is to c." Here, a is the *reference trial* for the comparison. Based on this formalization, we define a cost function for learning a feature encoding by combining all pairs of trials (a,b) from the same class with all trials c belonging to different classes and demanding that a and b are more similar. The resulting set of trial triplets is then used to train a similarity-constraint encoder network as illustrated in Figure 1.

All trials within a triplet that constitutes a similarity constraint are processed using the same encoder pipeline. This results in three internal feature representations. Based on these, the reference trial is compared with the paired trial and the trial from the other class resulting in two similarity scores. We propose to use the dot product as similarity measure because this matches the way patterns are compared in a neural network classifier and it is also suitable to compare time series. The output layer of the similarity constraint encoder finally predicts the trial with the highest similarity score without further applying any additional affine transformations. The whole network can be trained like a common binary classifier, minimizing the error of predicting the wrong trial as belonging to the same class as the reference. The only trainable part is the shared encoder pipeline. This pipeline can be arbitrarily complex – e.g., also include recurrent connections within the pipeline.

Optionally, the triplets can be extended to tuples of higher order by adding more trials from other classes. This results in a gradually harder learning task because there are now more other trials to compare with. At the same time, each single training example comprises multiple similarity constraints, which might speed up learning. In the context of this paper, we focus only on triplets.

4. EXPERIMENT

4.1. Dataset and Pre-Processing

The OpenMIIR dataset [15] is a public domain dataset of EEG recordings taken during music perception and imagination.[2] Data was collected from 10 subjects who listened to and imagined 12 short music fragments – each 7–16 s long. The 12 music stimuli

[2] Available at https://github.com/sstober/openmiir

were taken from 8 well-known pieces and comprised 4 songs recorded each with and without lyrics and 4 purely instrumental pieces as listed in Figure 2. These stimuli systematically span several musical dimensions such as meter, tempo and the presence of lyrics. All stimuli were normalized in volume and kept as similar in length as possible with care taken to ensure that they all contained complete musical phrases starting from the beginning of the piece. The pairs of recordings for the same song with and without lyrics were tempo-matched. The stimuli were presented to the participants in several conditions while EEG was recorded. This paper focuses on the perception condition where participants were asked to just listen to the stimuli. The presentation was divided into 5 blocks that each comprised all 12 stimuli in randomized order. In total, 60 perception trials were recorded per subject.

The EEG data were preprocessed as described in [15] using the MNE-python toolbox [16] to remove unwanted artifacts. We kept the original sampling rate of 512 Hz and normalized all trial channels to zero mean and range $[-1,1]$. Data of one participant were excluded because of a considerable number of trials with movement artifacts due to coughing. All trials needed to be cut off at 6.9 s, the length of the shortest stimulus, as zero-padding would have easily revealed the classes. This resulted in an equal input size of 3518 samples times 64 channels for 540 trials from 9 subjects.

4.2. Encoder Pipeline and Classifiers

The primary aim of this experiment was to demonstrate the usefulness of the proposed pre-training techniques. Thus, a very simple encoder pipeline was chosen and the number of hyper parameters was kept as low as possible to minimize their impact. The encoder pipeline consisted of a single convolutional layer with just a single filter and without a bias term. This filter aggregated the 64 raw EEG channels into a single waveform processing one sample (over all channels) at a time. I.e. it had the shape 64x1 (channels x samples). The hyperbolic tangent (tanh) was used as activation function because its output range matched the value range of the network inputs ([-1,1]). No pooling was applied.

As an estimate of the unknown signal-to-noise ratio within the data, a linear support vector machine classifier (SVC) was trained using Liblinear [17] on

- baseline (1): the raw EEG data,
- baseline (2): the averaged EEG data (mean over all channels as a naïve filter), and
- the output of the pre-trained encoder pipeline

interpreting an increase in the stimulus classification accuracy as a reduction of the signal-to-noise ratio. As additional classifier, a simple neural network (NN) was trained on the encoder pipeline output. This network consisted of a single fully-connected layer with a Softmax non-linearity. No bias term was used. This resulted in one temporal pattern learned for each of the 12 stimuli, which could then be analyzed. For further comparison, we also trained and end-to-end neural network that had the same structure as the encoder pipeline combined with the neural network classifier but was initialized randomly instead of pre-training. Furthermore, we also used a siamese network and a triplet network as alternative method for pre-training. All tested methods are listed in Table 1.

4.3. Training and Evaluation Scheme

A nested cross-validation scheme was chosen that allowed to use each trial for testing once. The outer 9-fold cross-validation was performed across subjects, training on 8 and testing on the 9th subject. The inner 5-fold cross-validation was used for model selection based on 1 of the 5 trial blocks. Training was divided into two phases.

In the first phase, the encoder pipeline was trained using the proposed similarity-constraint encoding technique with the hinge loss as cost function. Stochastic gradient descent (SGD) with a batch size of 1000 and the Adam [18] step rule was used. Training was stopped after 10 epochs and the model with the lowest binary classification error on the validation triplets was selected. Triplets were constructed such that all trials within a triplet belonged to the same subject as the simple encoder pipeline likely could not easily compensate inter-subject differences. The validation triplets consisted of a reference trial from the validation trials and the other two trials drawn from the combined training and validation set of the inner cross-validation. This way, a reasonable number of validation triplets could be generated without sacrificing too many trials for validation.[3] The final encoder filter weights were computed as mean of the 5 fold models. The output of this filter was used to compute the features for the second training phase.

In the second phase, the two classifiers were trained. For the SVC, the optimal value for the parameter C that controls the trade-off between the model complexity and the proportion of non-separable training instances was determined through a grid search during the inner cross-validation. For the neural network classifier, 5 fold models were trained for 100 epochs using SGD with batch size 120, the Adam step rule, and the hinge loss as cost function. The best models were selected based on the 12-class stimulus classification performance on the validation trials and then averaged to obtain the final classifier.

The experiment was implemented in Python using the frameworks Theano [19] as well as Blocks and Fuel [20]. The full code to run the experiment and generate the plots shown in this paper is available as supplement.[4] As the OpenMIIR dataset is public domain, this assures full reproducibility of the results presented here.

5. RESULTS

Table 1 lists the classification accuracy for the tested approaches. Remarkably, all values were significantly above chance. Even for baseline 2, the value of 12.41% was significant at p=0.001. This significance value was determined by using the cumulative binomial distribution to estimate the likelihood of observing a given classification rate by chance.

[3] At least 2 of the 5 trials per class and subject are required to construct within-subject triplets.

[4] https://dx.doi.org/10.6084/m9.figshare.4530797

Table 1. Classifier accuracies for the 12-class stimulus identification task and significance values for the comparison against our proposed method using McNemar's tests (n=540).

Classifier & Input	Accuracy	McNemar's test (mid-p) vs. *
SVC, raw EEG	18.52%	0.0002
SVC, raw EEG channel mean	12.41%	<0.0001
End-to-end NN, raw EEG	18.15%	0.0001
SVC, siamese network features	12.96%	<0.0001
SVC, triplet network features	25.56%	0.29
NN, SCE features	27.22%	0.82
*SVC, SCE features	27.59%	

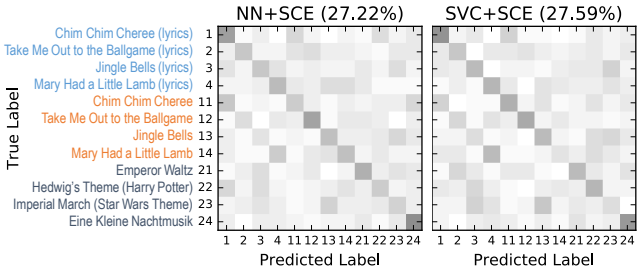

Fig. 2. 12-class confusion matrices for the music stimuli (listed on the left) for the classifiers trained on the SCE features. Middle: SVC. Right: Neural network classifier. Results were aggregated from the 9 outer cross-validation folds (n=540).

To evaluate whether the differences in the classification accuracies produced by the different methods are statistically significant, McNemar's tests using the "mid-p" variant suggested in [21] were applied. The obtained p-values are also shown in Table 1. Using our proposed similarity-constraint encoding method resulted in the best classification accuracy. As the very similar confusion matrices in Figure 2 show, the choice of the classifier for the SCE output had almost no impact on the classification outcome. The very significant improvement of the classification accuracy over the two baselines is a strong indicator for a reduction of the signal-to-noise ratio. Notably, the pre-trained filter is very superior to the naïve filter of baseline 2 that was actually harmful judging from the drop in accuracy. The SCE approach also outperformed the two related pre-training techniques. However, the difference of 2% compared to the triplet network was not statistically significant. Investigating the difference between these two encoder models, we found that their correct predictions only overlap by 70% which might turn out beneficial if they were combined in an ensemble. The encoder weights were much more stable across folds for SCE. Consequently, the temporal patterns learned by the neural network classifier turned out more crisply defined and sparse. We attribute this difference to our choice of the dot product as similarity measure.

Apart from the main diagonal in the confusion matrices, two parallel diagonals can be seen that indicate confusion between stimuli 1–4 and their corresponding stimuli 11–14, which are tempo-matched recordings of songs 1–4 without lyrics. Analyzing the averaged neural network parameters visualized in Figure 3

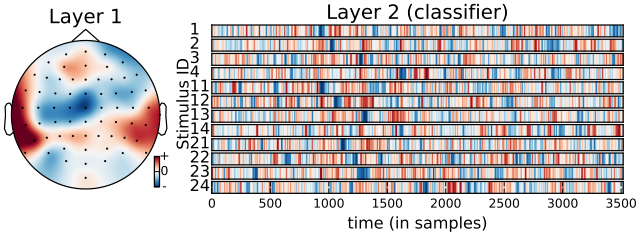

Fig. 3. Visualization of the average neural network parameters (from the 9 outer cross-validation folds). Layer 1: mean of convolutional layers from the pre-trained encoders (SCE). The filter weights only differed in small details across folds. Layer 2: mean of classifier layers trained in the supervised phase.

shows similar temporal patterns for these stimuli pairs.[5] A detailed analysis of the network layer activations revealed noticeable peaks in the encoder output and matching weights with high magnitude in the classifier layer that often coincide with downbeats – i.e., the first beat within each measure, usually with special musical emphasis. These peaks are not visible in the channel-averaged EEG (baseline 2). Thus, it can be concluded that the encoder filter has successfully extracted a component from the EEG signal that contains musically meaningful information.

6. CONCLUSIONS

Trying to determine which music piece somebody listened to based on the EEG is a challenging problem. Attempting to do this across subjects and with a small training set (less than 55 minutes in total, or less than 7 minutes per subject), makes the task even harder. Specifically, in the experiments described above, classifiers were trained for a 12-class problem with an input dimensionality of 225,280 given only 5 examples per class and subject, training on 8 subjects and testing on the 9th.

Thanks to the *similarity-constraint encoding* (SCE) pre-training technique introduced in this paper, a simple convolutional filter was learned that reduced the data dimensionality by factor 64 and at the same time significantly improved the signal-to-noise ratio. Using the resulting feature representation, the classification accuracy substantially increased. The trained neural network classifier is simple enough to allow for interpretation of the learned parameters by domain experts and facilitate findings about the cognitive processes. For learning such simple models, the pre-training is essential as it would be almost impossible to obtain the result shown in Figure 3 by basic supervised training. As next step, more complex models – as pre-trained encoders as well as classifiers – will be tested that are likely to further improve the classification accuracy.

Acknowledgments: This research was supported by the donation of a Geforce GTX Titan X graphics card from the NVIDIA Corporation.

[5]The average model is only for illustration and analysis. For testing, the respective outer cross-validation fold model was used for each trial.

7. REFERENCES

[1] S.M. Plis, D.R. Hjelm, R. Salakhutdinov, E.A. Allen, H.J. Bockholt, J.D. Long, H.J. Johnson, J.S. Paulsen, J.A. Turner, and V.D. Calhoun, "Deep learning for neuroimaging: a validation study," *Frontiers in Neuroscience*, vol. 8, 2014.

[2] P. Mirowski, D. Madhavan, Y. LeCun, and R. Kuzniecky, "Classification of patterns of EEG synchronization for seizure prediction," *Clinical Neurophysiology*, vol. 120, no. 11, pp. 1927–1940, 2009.

[3] D.F. Wulsin, J.R. Gupta, R. Mani, J.A. Blanco, and B. Litt, "Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement," *Journal of Neural Engineering*, vol. 8, no. 3, 2011.

[4] M. Längkvist, L. Karlsson, and M. Loutfi, "Sleep stage classification using unsupervised feature learning," *Advances in Artificial Neural Systems*, 2012.

[5] H. Cecotti and A. Gräser, "Convolutional Neural Network with embedded Fourier Transform for EEG classification," in *19th International Conference on Pattern Recognition (ICPR)*, 2008, pp. 1–4.

[6] H. Cecotti and A. Gräser, "Convolutional Neural Networks for P300 Detection with Application to Brain-Computer Interfaces," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 433–445, 2011.

[7] S. Jirayucharoensak, S. Pan-Ngum, and P. Israsena, "EEG-Based Emotion Recognition Using Deep Learning Network with Principal Component Based Covariate Shift Adaptation," *The Scientific World Journal*, 2014.

[8] W.-L. Zheng, J.-Y. Zhu, Y. Peng, and B.-L. Lu, "EEG-based emotion classification using deep belief networks," in *2014 IEEE International Conference on Multimedia and Expo (ICME)*, 2014, pp. 1–6.

[9] S. Stober, D.J. Cameron, and J.A. Grahn, "Classifying EEG recordings of rhythm perception," in *15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 649–654.

[10] S. Stober, D. J. Cameron, and J. A. Grahn, "Using convolutional neural networks to recognize rhythm stimuli from electroencephalography recordings," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 1449–1457.

[11] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a Similarity Metric Discriminatively, with Application to Face Verification," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, Washington, DC, USA, 2005, CVPR '05, pp. 539–546, IEEE Computer Society.

[12] E. Hoffer and N. Ailon, "Deep metric learning using Triplet network," *arXiv:1412.6622 [cs, stat]*, Dec. 2014, arXiv: 1412.6622.

[13] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, and others, "Greedy layer-wise training of deep networks," *Advances in neural information processing systems (NIPS)*, vol. 19, pp. 153, 2007.

[14] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons," *Advances in neural information processing systems (NIPS)*, pp. 41–48, 2004.

[15] S. Stober, A Sternin, A.M. Owen, and J.A. Grahn, "Towards music imagery information retrieval: Introducing the OpenMIIR dataset of EEG recordings from music perception and imagination," in *16th International Society for Music Information Retrieval Conference (ISMIR)*, 2015, pp. 763–769.

[16] A. Gramfort, M. Luessi, E. Larson, D.A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. Hämäläinen, "MEG and EEG data analysis with MNE-Python," *Frontiers in Neuroscience*, vol. 7, 2013.

[17] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[18] D.P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[19] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proc. of the Python for Scientific Computing Conference (SciPy)*, 2010.

[20] B. van Merriënboer, D. Bahdanau, V. Dumoulin, D. Serdyuk, D. Warde-Farley, J. Chorowski, and Y. Bengio, "Blocks and fuel: Frameworks for deep learning," *arXiv:1506.00619*, 2015.

[21] M.W. Fagerland, S. Lydersen, and P. Laake, "The mcnemar test for binary matched-pairs data: mid-p and asymptotic are better than exact conditional," *BMC medical research methodology*, vol. 13, no. 1, pp. 1, 2013.