8th International IEEE EMBS Conference on Neural Engineering
Shanghai, China, May 25 - 28 , 2017

Convolutional Neural Network-based Transfer Learning and Knowledge Distillation using Multi-Subject Data in Motor Imagery BCI

Siavash Sakhavi[1] and Cuntai Guan[2]

Abstract— **In Brain Computer Interfaces (BCIs), with multiple recordings from different subjects in hand, a question arises regarding whether the knowledge of previously recorded subjects can be transferred to a new subject. In this study, we explore the possibility of transferring knowledge by using a convolutional network model trained on multiple subjects and fine-tuning the model on a small amount of data from a new subject, thus, reducing the calibration time by reducing the time needed to record data and train a model. Our results show a significant increase in 4-class classification accuracy on the BCI IV-2a competition data, even when a small subset of the data is provided for training.**

I. INTRODUCTION

Transferring knowledge from subject to subject is a challenging problem in Brain Computer Interface (BCI) systems due to anatomical differences between the subjects [1] or statistical variations in the data [2]. The main goal in most transfer learning solutions proposed in literature is to reduce the calibration time needed to gather new information regarding a new subject. Recording data is time consuming, meaning it is mentally exhausting for the subject which in turn affects the quality of the data recorded. Therefore, it is highly favourable if the information from other subjects can be used to reduce the calibration time (i.e. use a lower of number samples from the new subject and/or using a pre-trained model).

As discussed in a survey by Jayaram et. al [2], there are mainly two approaches in transfer learning: domain adaptation (DA) or rule adaptation (RA). In DA, the solution proposed for transferring knowledge between subjects is bringing their data into a common space. These algorithms have mainly focused on finding spatial filters that are common amongst subjects [3], [4] and have been the to-go method for transfer learning. In RA, the classifier is changed based on a distribution of classifiers between the subjects[5].

In this study, we are taking a RA-based approach to the problem of classifying motor imagery EEG signals: rather than bringing the data to a common space, we train a neural network model that captures information from multiple subjects and stores the information as the parameters of the network. This network has been trained to classify these data correctly to their corresponding classes. The choice of a neural networks as the classification model is due to their high capacity and flexibility in design [6]. Such a high capacity network cannot be trained in the presence of low number of samples and is sure to over-fit. To our knowledge, this is one of the first attempts to use neural network based method for transfer learning in BCI.

After training, the network is then transferred to a new subject and fine-tuned to match the subject's classification needs. These needs can either be using less data to reduce the calibration time in a single session or using the fine-tuned model to transfer to a new session of the same subject (session-to-session transfer).

Based on the above, in this paper, we propose a pipeline for EEG which:

- Extracts EEG representations from multiple subjects *independently* (III-A)
- Uses a deep convolutional neural network to train a model on the multi-subject data (IV-B)
- Transfers the model parameters to train/fine-tune on the new subject's data (III-B)
- Utilizes the labels estimated by the transferred model to regularize the training/fine-tuning process (III-B)

Using this pipeline, we have achieved significant accuracy increase in same session transfer when a low number of samples is used to train the model. A schematic of the pipeline can be seen is figure 1.

II. DATA

We are using the BCI competition IV-2a dataset [7] which is 9-subject 4-class motor-imagery dataset with 72 samples per class per session. Each trial contains a cue in which after the subject performs 4 seconds of motor imagery for each of the classes (right hand, left hand, tongue and feet). The data recorded contains two sessions with the first session being the train data and second session being the test data in the original competition.

III. METHODS

A. EEG Representation

We have used the FBCSP[8], [9] algorithm to extract the representation similar to our previous paper [10]. Using the FBCSP algorithm, we first find the spatial filters and frequency bands that are contributing to the discriminance between the classes based on the log-energy features. The selected spatial filters are then applied to the original time-series data and the envelope is extracted using absolute value of the analytic signal. The envelope power is considered as the feature.

It should be noted that, similar to the FBCSP algorithm, the energy of each channel is divided by the energy of the

[1]Siavash Sakhavi is with Faculty of Electrical and Computer Engineering at the National University of Singapore and the Institute for Infocomm Research (I^{2}R) at A*STAR, Singapore. sakhavi@u.nus.edu
[2] Cuntai Guan is a Professor at School of Computer Science and Engineering, Nanyang Technological University ctguan@ntu.edu.sg

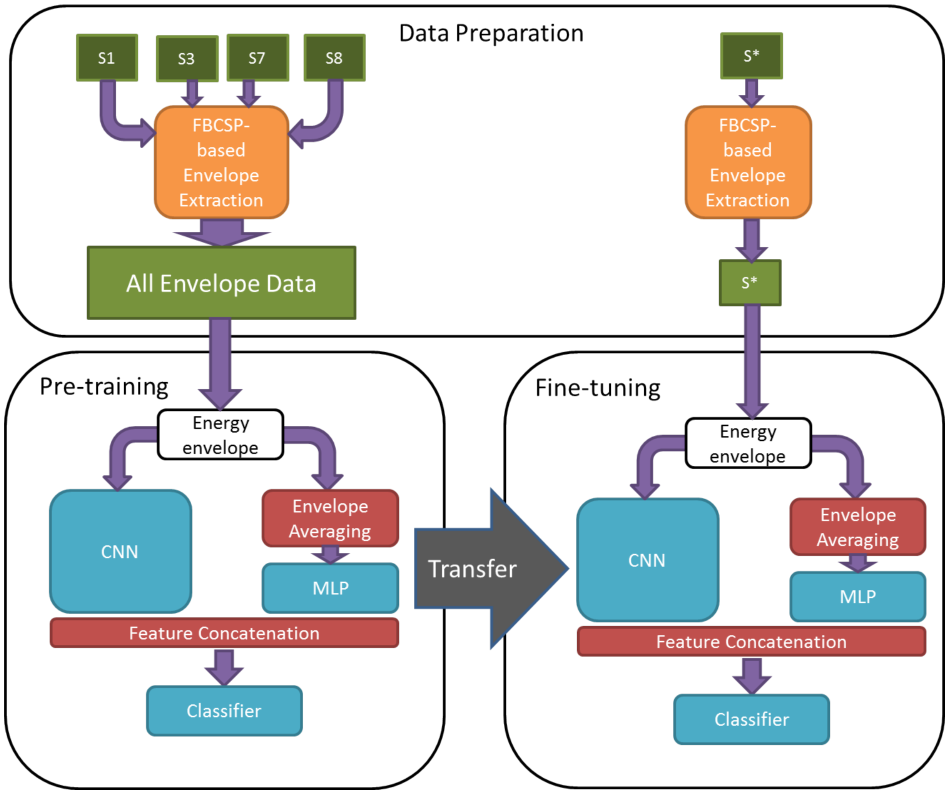

Fig. 1. Schematic of the algorithm proposed in this paper.

paired channels from the CSP algorithm. This means the end representation of the data is relative energy, not absolute energy. If absolute energy is used instead of relative energy, there is no guarantee that the energy content of one subject is similar to another subject. But when relative energy is used, the data is scaled into a space in which the data is more similar. This can be a simple step towards adapting the domain of the data without considering the data of other subjects.

The EEG representation extraction procedure can either be applied to each session individually or all sessions can be viewed as one. Using multiple individual sessions and/or union of multiple sessions can be used to increase the number of data in the multi-subject setting when the session of the other subjects are not important (e.g. when training a model for the new subject). For example, if using data from subject x to transfer to subject y, it is not important which session of subject x is used.

After extracting the representation, the average channel means are deducted and a scale of one over the median maximum of all trials is applied on the data.

B. *Transfer Learning & Knowledge Distillation*

By not using data from the new subject, we have a metric of what the model is learning about the new subject data by looking at the classification accuracy of the new subject's training set and by looking at the distribution of the labels for each of the trials. In other words, we will have two sets of labels for the training data of the new subject: one set which are the hard labels that have been provided based on the task (*hard labels*) while the other set of labels have been estimated by the pre-trained model on the subjects other than the new subject (*soft labels*). The soft labels contain useful information regarding the distribution of the data: if the hard and soft labels are not similar for a certain trial, this shows that there is a possibility that the subject has not performed well in that specific trial. In order to utilize the soft labels, we have turned to the knowledge distillation technique.

The knowledge distillation technique by Hinton [11], [12] introduces an algorithm to train networks based on the hard labels and also soft predictions that have been made about the data using a previously trained network. Given that the hard labels (task-related) of trial x_i is y_i and the soft label is s_i, the loss function for training the classifier f can be defined as :

$$f^* = \arg\min_f \frac{1}{N}\sum_{i=1}^{N}[\lambda(\mathcal{L}(y_i, f(X_i))) \\ + (1-\lambda)(\mathcal{L}(s_i, f(X_i)))] \quad (1)$$

In equation 1, $\mathcal{L}$ is the loss function. The last layer of function f is a softmax function ($\sigma(x)$) and s_i is the output of another network (f_t) which has been passed through an additional softmax with a temperature ($s_i = \sigma(f_t(X_i)/T)$). In our proposed algorithm, f_t is the network that has been trained on all the other subjects' data. λ is a constant value that balances the weight of the algorithm between the soft label and hard label. The temperature, T, is used to decrease or increase the similarity between the classes. If T is high, the probability of all classes become even. If T is low, one class becomes more probable than other classes and becomes similar to hard labeling. We have chosen a $T = 0.25$ for the results of this paper.

IV. RESULTS

A. Data Preparation

Regarding the FBCSP method, we have used 9 frequency bands, 2 pairs of CSP features and a one-versus-rest strategy to calculate the spatial features. This leads to 32 spatially filtered channels. The 4 second motor imagery period has been used for calculating the CSP spatial filters and also been used for the envelope features. With a sampling rate of 250 Hz, the number of samples is 1000. Because the envelope feature is baseband, the number of samples can be decreased to 40, yielding a 32×40 dimension for the input feature.

During the pre-training stage of the classifier, the FBCSP algorithm is performed on each session individually and also the union of sessions for subjects other than the current subject, resulting in ~ 2000 samples of training data per class. Although we can use the data from all subjects, in this study we have decided to use the data from subjects that have high session-to-session performance. Therefore, the number of samples used for training the networks is ~ 1000 per class. The subjects used to train the network are subjects $1, 3, 7$, and 8. Naturally, when the network is fine-tuned on the above subjects, the subject's data will be excluded from pre-training.

B. CNN Architecture

Using multi-subject data increases the number of samples used for training. This gives the opportunity to utilize classifiers that have high learning capacity such as Convolutional Neural Networks (CNN)[13]. We have utilized the model shown in figure 1. The convolutional neural network configuration is brought in table I.

TABLE I
DETAILS OF THE CNN ARCHITECTURE

Layer #	Layer Type	Patch size / Stride	Hidden Unit
1	Convolution	$4 \times 1/2 \times 1$	32
2	Convolution	$3 \times 1/2 \times 1$	32
3	Convolution	$3 \times 1/2 \times 1$	32
4	Convolution	$1 \times 32/1 \times 1$	32
5	Linear	-	128

Assuming the input contains two dimensions of channel and time ($X \in \mathbf{R}^{C \times T}$), the CNN contains two types of convolution: temporal (Layers 1,2,3) and channel-Wise (Layer 4). Temporal convolution applies kernel independent of which channel it belongs to and channel-wise convolution mixes the channels in the channel dimension. Convolution and linear units are without bias. After each linear unit, there are batch normalization[14] and ReLU [15] activation units. After each convolution layer there is a ReLU layer only. The main reason for this design choice is that batch normalization, in images, tends to normalize the activation layers in each individual pixel. In signals, this is not an desired operation.

The MLP contains a single linear layer with the hidden unit number of 128. The input to the MLP unit is the average of the energy envelope of each of the channels over time. Therefore, the dimension of the input to the MLP layer is 32 and the output is 128.

After concatenation of the features from the MLP and CNN models, a vector of 256 features is created and fed into a linear classifier. The model is then trained given a KL-Divergence loss function and using the Adam optimization algorithm[16]. It should be emphasized that the network is optimized jointly; This means that the error is back-propagated from the loss function to the MLP and CNN networks.

To increase the classification accuracy and decrease the initialization dependency, an ensemble of 5 networks with different initialization is trained. The Torch7 [17] deep learning software has been used to design and train the networks. To compare the classification performance, we have set the baseline to be a SVM classifier operating on the FBCSP energy features, extracted from the time interval of 0.5 to 2.5 s after the cue.

C. Same Session Subset Results

For the results of this section, we use the second session data provided for all subjects. We extract 5, 10, and 20 samples per class from the session and use the rest of the samples are used for evaluation. The results for each of the sample sizes can be seen in Table II. Note that these results are not cross-validation results but rather a small subset of the data selected randomly. As shown in the table, for $\lambda = 1$, in all sample sizes, using transfer learning boosts the average accuracy over subjects. This increase of accuracy can be also seen in each subject individually in almost all subjects. At the same time, for smaller samples, using a smaller value for λ shows better results. This can be contributed to the fact that for smaller sample sizes, such as 5 and 10 samples per class, are not sufficient for fine-tuning and a regularization is needed to reduce over-fitting.

Based on the Wilcoxon signed-rank test, the increase in the accuracy for 5 and 20 samples is significant with a bound of $p < 0.05$, but for 10 samples the Wilcoxon score is not low enough for the significance bound of $p < 0.05$. This can be solved by increasing the capacity of the network or increasing the number in the ensemble which will be explored in future studies. Furthermore, the difference between the best λ and $\lambda = 1$, is not significant based on the Wilcoxon score. A significant difference relative to changes in λ may be seen in cross-validation results.

An interesting observation in table II is related to some subjects that have relatively high performance in the SVM, subjects $3, 8$, and 9. There performance increase is in the order of 9 to 16 percent increase in the 5 sample case. This is an indication that their data is very similar to other good subjects. As a result, the accuracy given by fine-tuning the

TABLE II

RESULTS FOR SMALL SUBSET SELECTION FROM THE SECOND SESSION OF THE BCI COMPETITION DATA

	5 Samples			10 Samples			20 Samples		
	SVM	$\lambda = 1$	$\lambda = 0.75$	SVM	$\lambda = 1$	$\lambda = 0.5$	SVM	$\lambda = 1$	$\lambda = 0.75$
Subject 1	54.48	53.36	**54.85**	60.89	**62.10**	59.27	69.71	**72.12**	70.67
Subject 2	29.10	29.85	**29.85**	36.69	**39.52**	38.71	47.12	**49.04**	**49.04**
Subject 3	50.75	**76.87**	**76.87**	71.37	77.42	**78.23**	80.77	**84.13**	83.65
Subject 4	39.55	**41.79**	41.04	33.47	**51.61**	48.79	54.33	**63.46**	59.13
Subject 5	27.99	35.07	**35.45**	**44.76**	43.55	44.35	**58.65**	**58.65**	56.73
Subject 6	29.10	30.22	**30.22**	29.84	28.23	**30.24**	35.10	**38.46**	**38.46**
Subject 7	52.61	54.48	**54.85**	59.27	**84.27**	83.87	75.48	**87.50**	86.54
Subject 8	46.64	56.34	**56.72**	63.31	**66.13**	**66.13**	77.40	**83.65**	**83.65**
Subject 9	58.21	**69.03**	**69.03**	72.18	73.39	**75.81**	82.21	90.38	**91.35**
Average	43.16	49.67	**49.88**	52.42	58.47	**58.38**	64.53	**69.71**	68.80

model on a small sample can be an indication of whether the subject is able to perform motor-imagery well.

V. CONCLUSIONS

This paper proposes a new pipeline for classifying 4-class motor-imagery data using transfer learning. The results show a significant increase in both small subset calibration in the three cases shown. Although Training neural networks is time consuming but when using for deployment, each forward pass of a sample is in the order of milliseconds. Therefore, are pipeline proposes a accurate *and* fast deep neural network model that can deployed after a few samples have been obtained.

There are a few parameters that have been pre-selected such as the temperature value for soft labels or the number of hidden nodes in the MLP and/or CNN. Cross-validation must be done to effectively select the parameters. Amongst the parameters, the parameter λ is not selected and only the best values of lambda have been reported and due to being useful for small sample size, it is a priority to find a way to optimize the value.

REFERENCES

[1] M. Wronkiewicz, E. Larson, and A. K. C. Lee, “Leveraging anatomical information to improve transfer learning in braincomputer interfaces,” *Journal of Neural Engineering*, vol. 12, no. 4, p. 046027, aug 2015.

[2] V. Jayaram, M. Alamgir, Y. Altun, B. Scholkopf, and M. Grosse-Wentrup, “Transfer Learning in Brain-Computer Interfaces,” *IEEE Computational Intelligence Magazine*, vol. 11, no. 1, pp. 20–31, feb 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7379089/

[3] M. Krauledat, M. Tangermann, B. Blankertz, and K. R. Müller, “Towards zero training for brain-computer interfacing,” *PLoS ONE*, vol. 3, no. 8, p. e2967, aug 2008. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0002967

[4] H. Kang and S. Choi, “Bayesian common spatial patterns for multi-subject EEG classification,” *Neural Networks*, vol. 57, pp. 39–50, 2014.

[5] M. Alamgir, M. Grosse-wentrup, and Y. Altun, “Multitask learning for brain-computer interfaces,” in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 17–24.

[6] I. Goodfellow, Y. Bengio, and A. Courville, “Deep learning,” 2016, book in preparation for MIT Press. [Online]. Available: http://www.deeplearningbook.org

[7] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. J. Miller, G. R. Müller-Putz, G. Nolte, G. Pfurtscheller, H. Preissl, G. Schalk, A. Schlögl, C. Vidaurre, S. Waldert, and B. Blankertz, “Review of the BCI Competition IV,” *Frontiers in Neuroscience*, vol. 6, p. 55, 2012. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2012.00055/abstract

[8] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, “Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b,” *Front. in neurosci.*, vol. 6, 2012.

[9] K. K. Kai Keng Ang, Z. Y. Zhang Yang Chin, H. Haihong Zhang, and C. Cuntai Guan, “Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface,” in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, jun 2008, pp. 2390–2397.

[10] S. Sakhavi, C. Guan, and S. Yan, “Parallel convolutional-linear neural network for motor imagery classification,” in *2015 23rd European Signal Processing Conference (EUSIPCO)*. IEEE, aug 2015, pp. 2736–2740.

[11] G. Hinton, O. Vinyals, and J. Dean, “Distilling the Knowledge in a Neural Network,” mar 2015. [Online]. Available: http://arxiv.org/abs/1503.02531

[12] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik, “Unifying distillation and privileged information,” *arXiv*, pp. 1–10, nov 2015. [Online]. Available: http://arxiv.org/abs/1511.03643

[13] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Handwritten digit recognition with a back-propagation network,” in *Advances in Neural Information Processing Systems 2 (NIPS*89)*, D. Touretzky, Ed. Denver, CO: Morgan Kaufman, 1990.

[14] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *Proceedings of The 32nd International Conference on Machine Learning*, 2015, pp. 448–456. [Online]. Available: http://jmlr.org/proceedings/papers/v37/ioffe15.html

[15] X. Glorot, A. Bordes, and Y. Bengio, “Deep Sparse Rectifier Networks,” in *AISTATS*, vol. 15, 2011, pp. 315–323. [Online]. Available: http://eprints.pascal-network.org/archive/00008596/

[16] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” dec 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[17] R. Collobert, K. Kavukcuoglu, and C. Farabet, “Torch7: A matlab-like environment for machine learning,” in *BigLearn, NIPS Workshop*, no. EPFL-CONF-192376, 2011.