Supplementary for: A study of prototypical network techniques for cross-subject EEG analysis

Paper #2170

1 Training details

1.1 Network parameters

Table 1. The parameters of IncepMobiNet.

Block	Layer	Filters	size	Output	Operations
Input	-	-	-	(C,1,T)	-
EEG Inception	Spatial Conv	30	1x1	(30,1,T)	-
	Task Normalization	-	-	-	-
	Temporal DWConv	60	1x25	(3*60,1,T)	zero padding = same,grous = 30
		60	1x55		zero padding = same,grous = 30
		60	1x125		zero padding = same,grous = 30
	Task Normalization	-	-	-	-
	Activation	-	-	-	ELU
	Avg pooling	-	1x3	(3*60,1,T//3)	strid = 3
	Dropout	-	-	-	p = 0.5
EEG Mobile	Spatial Conv	3*60	1x1	(3*60,1,T//3)	-
	Task Normalization	-	-	-	-
	Temporal DWConv	3*60	1x25	(3*60,1,T//3)	zero padding = same,grous = 3*60
	Task Normalization	-	-	-	-
	Spatial Conv(skip)	3*60	1x1	(3*60,1,T//3)	-
	Task Normalization(skip)	-	-	-	-
	Add	-	-	(3*60,1,T//3)	
	Activation	-	-	-	ELU
	Avg pooling	-	1x3	(3*60,1,T//9)	strid = 3
	Dropout	-	-	-	p = 0.5
EEG Mobile	Spatial Conv	40	1x1	(40,1,T//9)	-
	Task Normalization	-	-	-	-
	Temporal DWConv	40	1x25	(40,1,T//9)	zero padding = same,grous = 40
	Task Normalization	-	-	-	-
	Spatial Conv(skip)	40	1x1	(40,1,T//9)	-
	Task Normalization(skip)	-	-	-	-
	Add	-	-	(40,1,T//9)	-
	Activation	-	-	-	ELU
	Avg pooling	-	1x3	(40,1,T//27)	strid = 3
	Dropout	-	-	-	p = 0.5
Adaption layer	Fully connected	-	-	(40,64)	-

The parameters of the IncepMobile feature extraction network are shown in Table 1. In this work, we use the IncepMobile network with 1 EEG Inception block and 2 EEG MobileNet blocks.

1.2 Overall Training and Calibration Procedure

The process of network training and calibration is shown in Algorithm 1. The training of CS-EEGPNet is divided into three phases, the first phase is model pre-training, in which we train the network directly using the data from the support set and directly using trainable cofactors as prototypes, which allows the cofactors to quickly learn category-relevant information from multiple subjects. The second stage is the training of the model, in which a few shot normalization and a prototype network based on auxiliary factors are used to reduce the effect of individual differences and allow the model to learn more universal features across multiple subjects. The third stage is the fine-tuning of the model, which can be done to further improve the model performance if the target subjects have sufficient labeled data.

2 Experiments

2.1 Comparison Methods

- Baseline + Fine tuning: A simple transfer learning approach that uses data from existing subjects to train a model and then uses a small amount of data to fine-tune it.
- MUPS-EEG:A simple, computationally efficient meta-updating strategy for solving cross-subject EEG classification problems. It consists of a meta-representation learning phase and a meta-adaptation phase. Meta-representation learning is performed on a known source subject to extract features from different subjects, while meta-adaptation adapts the model to a new subject through few gradient updates without losing the knowledge of the known subject.
- MDMAML:A MAML-based meta-learning approach that aims to enable domain adaptation on models using multiple source domains.
- Prototypical:The vanilla prototypical network, which does not use FSFN layers and learnable auxiliary factors.

3 Supplementary experiments

3.1 Comparison with subject-specific models

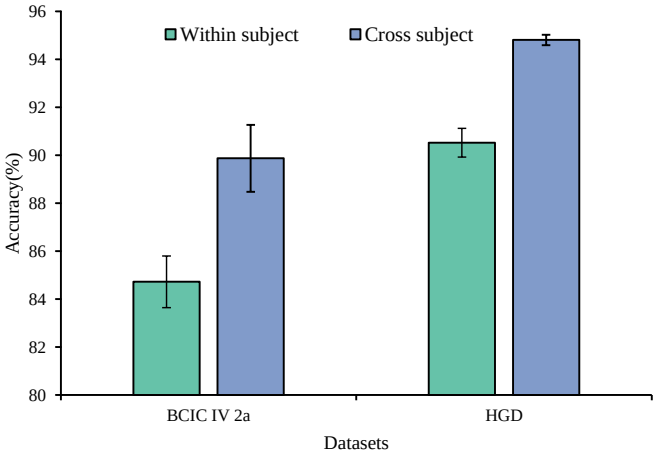

Figure 1. Comparison with subject-specific models.

The CS-EEGPNet proposed in this paper shows very superior performance in cross-subject few shot EEG classification. To further validate the classification performance when the data is sufficient, we

Algorithm 1 training and calibration of CS-EEGPNet.

Input: training tasks $\mathcal{T}_{train} = \{\tilde{\mathcal{T}}_1, \tilde{\mathcal{T}}_2, \cdots, \tilde{\mathcal{T}}_{N_S-1}\}$,small support set $\{(x_{1,1}^{Es}, y_{1,1}^{Es}), \cdots, (x_{N,K}^{Es}, y_{N,K}^{Es})\}$ of test task T_{test}, learning rate η and γ.

Output: Trained parameters Φ, the mean μ and variance σ^2 of all FSFN layers, and prototypes c_n^E.

▷ Pretraining

for for each pretraining epoch **do**:
  for $\mathcal{T}_i$ in $\mathcal{T}_{train}$ **do** :
    Sample $\mathcal{S}_{train} = \{(x_{1,1}^{Ts}, y_{1,1}^{Ts}), \cdots, (x_{N,K}^{Ts}, y_{N,K}^{Ts})\}$ from $\mathcal{T}_i$.
    $e^{Ts}, \mu, \sigma^2 = F(x^{Ts})$
    $c_n^T = a_n$
    Calcuate loss $\mathcal{L} = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\left[d(F(x_{n,k}^{Ts}), c_n^T) + log\left(\sum_{n'=1}^{N} exp(-d(F(x_{n,k}^{Ts}), c_{n'}^T))\right)\right]$
    Updata $\Phi_{t+1} = \Phi_t - \eta\Delta_\Phi\mathcal{L}$.

▷ Training

for for each training epoch **do**:
  for $\mathcal{T}_i$ in $\mathcal{T}_{train}$ **do** :
    Sample $\mathcal{S}_{train} = \{(x_{1,1}^{Ts}, y_{1,1}^{Ts}), \cdots, (x_{N,K}^{Ts}, y_{N,K}^{Ts})\}$ from $\mathcal{T}_i$.
    Sample $\mathcal{Q}_{train} = \{(x_{1,1}^{Tq}, y_{1,1}^{Tq}), \cdots, (x_{N,K}^{Tq}, y_{N,K}^{Tq})\}$ from $\mathcal{T}_i$.
    $e^{Ts}, \mu, \sigma^2 = F(x^{Ts})$
    $c_n^T = \frac{1}{K+1}\left(\sum e_n^{Ts} + a_n\right)$
    Calcuate loss $\mathcal{L} = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\left[d(F(x_{n,k}^{Tq}, \mu, \sigma^2), c_n^T) + log\left(\sum_{n'=1}^{N} exp(-d(F(x_{n,k}^{Tq}), c_{n'}^T))\right)\right]$
    Updata $\Phi_{t+1} = \Phi_t - \eta\Delta_\Phi\mathcal{L}$.

▷ Fune-tuning(optional)

for each fune-tuning epoch **do**:
  $e^{Es}, \mu, \sigma^2 = F(x^{Es})$
  $c_n^E = \frac{1}{K+1}\left(\sum e_n^{Es} + a_n\right)$
  Calcuate loss $\mathcal{L} = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\left[d(F(x_{n,k}^{Es}, \mu, \sigma^2), c_n^E) + log\left(\sum_{n'=1}^{N} exp(-d(F(x_{n,k}^{Es}), c_{n'}^E))\right)\right]$
  Updata $\Phi_{t+1} = \Phi_t - \gamma\Delta_\Phi\mathcal{L}$.

▷ Calibration

$e^{Es}, \mu, \sigma^2 = F(x^{Es})$
$c_n^E = \frac{1}{K+1}\left(\sum e_n^{Es} + a_n\right)$
return Φ, μ, σ^2, and c_n^E

test the average classification accuracies in both cross-subject and within-subject cases with 114 shot samples for the test task on the BCIC IV 2a and HGD datasets, respectively. In the cross-subject case, we use a trained CS-EEGPNet and fine-tune it using the 114 shot data from the test task; and in the within-subject case, the model uses a randomly initialized IncepMobiNet with batch normalization and trains the network using the 114 shot data. As can be seen from the figure, the cross-subject classification accuracies are higher than the within-subject classification accuracies on both datasets, which suggests that CS-EEGPNet effectively extracts useful information from other subjects and avoids the common negative transfer phenomenon in cross-subject classification.