Postprint of: Jesús S. García-Salinas, Alejandro A. Torres-García, Carlos A. Reyes-Garćia, Luis Villaseñor-Pineda, Intra-subject class-incremental deep learning approach for EEG-based imagined speech recognition, Biomedical Signal Processing and Control, Vol. 81 (2023), 104433, DOI: 10.1016/j.bspc.2022.104433

Intra-subject class-incremental deep learning approach for EEG-based imagined speech recognition

Jesus S. Garcia-Salinas[a,b], Alejandro A. Torres-García[a], Carlos A. Reyes-Garcia[a], Luis Villaseñor-Pineda[a]

[a]*Biosignals Processing and Medical Computing Laboratory, Instituto Nacional de AstrofísicaÓptica y Electrónica, Luis Enrique Erro #1, Santa María Tonantzintla, Puebla, México*

[b]*Brain and Mind electrophysiology laboratory, Multimedia Systems Department, faculty of electronics, telecommunications and informatics, Gdansk University of technology, Gdansk, Poland*

Abstract

Brain-computer interfaces (BCIs) aim to decode brain signals and transform them into commands for device operation. The present study aimed to decode the brain activity during imagined speech. The BCI must identify imagined words within a given vocabulary and thus perform the requested action. A possible scenario when using this approach is the gradual addition of new words to the vocabulary using incremental learning methods. An issue with incremental learning methods is degradation of the decoding capacity of the original model when new classes are added. In this study, a class-incremental neural network method is proposed to increase the vocabulary of imagined speech. The results indicate a stable model that did not degenerate when a new word was integrated. The proposed method allows for the inclusion of newly imagined words without a significant loss of total accuracy for the two datasets.

Keywords: EEG, BCI, Imagined speech, Neural networks, Incremental learning

2010 MSC: 00-01, 99-00

Email address: jss.garcia@inaoep.mx (Jesus S. Garcia-Salinas)

1. Introduction

Brain-Computer Interfaces (BCIs) are systems that can transform brain signals into commands to control a device. Different devices can be used to acquire brain signals, and this study focused on electroencephalography (EEG) because of its simplicity, low cost, and non-invasive nature.

Former EEG-based BCIs used external stimuli in which the brain activity related to such stimuli is known [1]. A common approach to BCIs is motor imagery, which involves imagining limb movement. The present study used an internal stimulus related to language known as imagined speech, which is the action of imagining the diction of a word without emitting any sound or articulating any movement [2]. The use of imagined speech may provide a new communication channel and open up the possibility of increasing the vocabulary of imagined words.

Currently, with a lack of consensus, slight changes in machine learning algorithms receive different names. Incremental, class incremental, lifelong, online, never-ending, and evolutionary terms were used for specific cases of the same problem, *transfer learning*. For example, according to the definitions in [3], lifelong learning is a continuous model adaptation method based on constantly arriving data streams. Online learning is applied when training examples are provided only once in the model instead of iterating across training sessions. Incremental learning is a machine learning paradigm in which the learning process occurs whenever new examples emerge and previous learning is adjusted [4]. This study follows the definition of [5] for incremental learning, which is an algorithm that fulfils the following tasks:

- Ability to learn additional information from new data.
- Have no access to the original data used to train the classifier.
- Preserve previously acquired knowledge.
- Ability to incorporate new classes.

Other studies, such as [6, 4], add other criteria to the incremental learning

definition, such as end-to-end architecture and limited processing and memory resources. However, a formal definition has not been established yet.

BCI incremental learning commonly focuses on inter-subject variability, that is, extending a generated model to new subjects [7, 8, 9, 10]. Specifically, for motor imagery BCIs, the idea of increasing the number of available commands has not been widely explored, because few limbs can be used to control a device. Moreover, relating body movements to specific commands can be confusing as more commands are included; this does not occur using imagined speech in which the words are directly related to the command.

One contribution of this study is the possibility of adding new words to a previously generated imagined speech discrimination model. When a BCI is trained for a specific task, its extension requires retraining by adding the information from a new task [11]. A common issue in incremental learning is the degradation of a model when new classes are added. When during the process of learning a new set of patterns, it suddenly and completely erases the knowledge already learned by a neural network [12], it is called *Catastrophic Forgetting*.

This study proposes a neural network architecture capable of extending an existing imagined speech model to recognize a new imagined word while avoiding catastrophic forgetting. This can be considered an intra-subject transfer learning task.

The main contributions of this study are:

1. A model based on neural networks for imagined speech discrimination.
2. An intra-subject incremental learning approach of imagined speech BCIs.

2. Related work

The following section is divided into two main areas: incremental learning methods and incremental learning applications for BCIs. Incremental learning has many application areas, and proposals for different approaches are analyzed within the scope of this study. Moreover, some of these approaches have been

developed for BCI applications, and specialization in this area has also been analyzed to develop an adequate transfer learning method.

2.1. Incremental learning

Multiple approaches have been developed for incremental neural networks, such as fine-tuning, feature extraction, joint training, architecture adaptation, knowledge distillation, and ensembles. The relevant approaches for this study are presented below.

2.1.1. Feature extraction

This approach uses the outputs of a trained neural network as feature vectors for other machine-learning methods to improve performance.

In [13], there was interest in implementing a function that reduced the intra-class distance and increased the inter-class distance of the network outputs. Using these improved distances, an SVM-based classifier was applied to the old and new classes. Following the previous idea, the outputs of any neural network can be used as features for other machine learning methods. In [14], an ensemble of SVM classifiers was proposed, based on a previous study.

The use of instances or information from old classes has also been considered in some studies, and prototype-based incremental learning was presented in [15], which used *exemplar images* as well as class prototypes. Nevertheless, it sets the basis for further studies, such as [16], in which the last layer of a neural network is transformed into a Nearest Class Mean (NCM) classifier, which is a special case of k-Nearest Neighbors. This layer can add a new class by averaging the instances belonging to that class. Finally, classification was performed using a probability softmax function that assigned the input vector to the closest mean.

Owing to the variability in brain signals across different subjects or sessions for the same subject [17], incremental learning provides an area of opportunity for BCIs. Motor imagery takes advantage of the spatial behavior of signals; thus, these BCIs are based on common spatial patterns (CSPs), assuming that a set of invariant spatial filters exists across sessions or subjects [18, 19, 20, 21, 22].

Spatial information is also relevant for imagined speech analysis; [23, 24, 25] showed that widespread brain regions are involved. However, [26, 27, 28, 29] have suggested that focused brain areas are related to imagined speech. In contrast to previous studies, [2, 30, 31, 32, 33, 34] employed frequency analysis for imagined speech to identify the features of signals related to the frequency bands of neural activity.

2.1.2. Architecture adaptation

Neural networks are flexible models that allow for the development of a wide variety of architectures. Incremental learning takes advantage of this property by modifying the neural network architectures to allow the inclusion of new classes. An intuitive idea is to grow the network as a tree as new classes are added.

In [35] a hierarchical model combined with neural networks was proposed. The main idea is to group classes into super-classes, which can be split when new data are fed. A specific classifier was trained for each super-class, and the main drawback was defining a similarity measure to merge or split the super-classes. In [36], super-class networks were grown as trees when new classes were fed. The super-class network evaluates the inputs and determines which sub-class of the network it corresponds to. Subsequently, the branch network creates a sharp classification. In [37], a tree approach was proposed; in this case, the branches correspond to old and new data, and share a base network. The new branches were trained independently and added to the old branches to update the network.

2.1.3. Knowledge distillation

Finally, the use of pre-trained neural network models for new tasks results in a term called *knowledge distillation*, in which the information of a cumbersome network is transferred to a lighter network. In [38] the distillation for incremental learning was improved. However, this proposal requires auxiliary data in the training step, also known as exemplar data, from the old classes.

In [39], the information obtained from old classes was retained by a teacher network, and distilled into a student network that learns new incremental classes using only information from these new classes. In addition, the use of a prototype-based classifier was proposed to retain the information of old classes.

In some cases, only a small number of instances of the new class are available, [40] named this approach "Few shot Incremental Learning," and established that knowledge distillation presents some issues as class imbalance and performance trade-off across new and old classes.

2.2. Incremental learning for BCIs

An extensive review of BCI transfer/incremental learning was presented in [41], the BCI approaches considered motor imagery, event-related potentials, steady-state visual-evoked potentials, affective BCIs, regression problems, and adversarial attacks; imagined speech was not considered. Additionally, possible transfer scenarios were defined as cross-subject, cross-session, cross-device, and cross-task transfer learning. The approach considered in this study is cross-task incremental learning, which refers to an increase in the number of classes for the same subject.a The previous mentioned review indicated that [42] was the only study that applied cross-task transfer learning, in such study it is defined a scenario in which the source subjects and the target subject perform different motor imagery tasks, this is both a cross-subject and a cross-task transferring.

2.3. Summary

Motor imagery is the most common BCI approach, and transfer learning is commonly focused on intersubject variability because it is not possible to add many motor imagery commands. A generated imagined speech model can be extended to new imagined words, which can be considered an intra-subject transfer learning task. Imagined speech provides a scenario in which the same subject can include new words in their vocabulary, thereby expanding the BCI command set.

The previously reviewed concepts were considered in the development of a transfer learning model. In this study, a neural network architecture is proposed to serve as a feature extractor for imagined speech classification. Nevertheless, as seen in previous studies, data preprocessing was also considered.

Moreover, for incremental learning, architectural adaptation was developed to allow the inclusion of a new class. The proposal in this study involves the addition of multiple parallel networks that share outputs to train new classes. Previous studies have proposed architectural adaptations to improve incremental learning and mitigate catastrophic forgetting.

To achieve this, a principle similar to knowledge distillation is followed in which the original architecture shares outputs with the architecture for the new class. Considering the knowledge represented in the original network to achieve a better representation of the new classes improves the transfer learning approach. The main difference between the proposed solutions is that the original network is the same size as the new network.

3. Method

3.1. Datasets

Each dataset used in subsequent experiments had different features and acquisition protocols, as described in this section. Thus, the proposed method was tested under different conditions. Each collection was labeled with a short tag for reference.

5C dataset: The first dataset is obtained from [2]. The EEG of twenty-seven native Spanish speaking subjects was recorded from 14 channels at a 128 Hz sample rate using an Emotiv Epoc device, with a bandwidth of 0.2 to 45 Hz and a notch filter at 50 Hz and 60 Hz. The data consists of five imagined speech Spanish words “Arriba”, “Abajo”, “Izquierda”, “Derecha”, “Seleccionar” (translated in English as “Up”, “Down”, “Left”, “Right”, “Select”), repeated thirty three times each one, with a rest period between repetitions. The recordings were performed in a controlled environment without any sound or visual

noise. However, in the acquisition protocol, words were presented sequentially.

3C dataset: The second dataset is presented in [34] two sets are presented, three short words ("In", "Out, "Up") and two long words ("Cooperate", "Independent"). For each task, six subjects were recorded with one-hundred trials per word. These signals were recorded using a BrainProducts ActiCHamp with 64 channels at 1000 Hz. The data were preprocessed using a band-pass filter between 8 and 70 Hz, a notch filter at 60 Hz, and an electro-oculogram artifact removal algorithm.

All of the mentioned processing was performed by the dataset owners, and no further preprocessing was performed on the datasets, which reduced the time and complexity of future real-time implementations. Testing the proposed method using different databases processed in different ways may allow the robustness of the model to be analyzed.

3.2. Network architecture

In the following experiments, the datasets were down-sampled to 128 Hz to reduce the data, and the Power Spectrum Density (PSD) was computed with the *pwelch* MATLAB function [43] for each channel and further concatenated. Subsequently, a simple convolutional network was trained, as shown in Fig. 1.

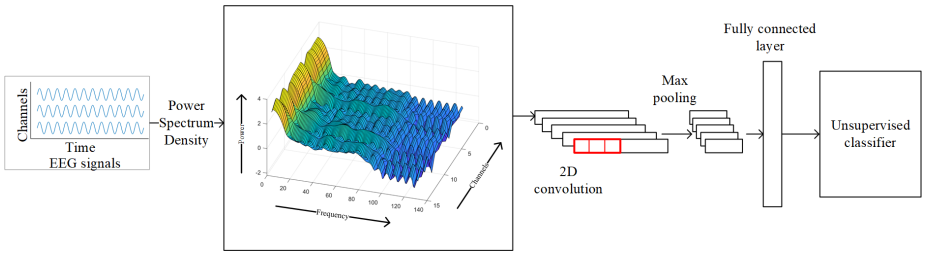


Figure 1: Convolutional network design

This convolutional network was configured as shown in Table 1, and the input size for the network corresponded to the number of channels, which were 14 and 62, respectively, depending on the dataset and frequency values of the PSD, which were 129. The proposed incremental approach considers a neural

network as the feature extractor. The clustering step is then applied to the outputs of the fully connected layer, each instance will have a different output regarding the weights of the neurons. Essentially, the network maps instances to a new space in which clustering is performed.

Table 1: Network parameters

Dataset	**5C**	**3C**
Input	(14, 1, 129)	(62, 1, 129)
Convolution	Kernel size: (1,5), Stride: 1, Filters: 100	
MaxPool	Kernel size: (1,2)	
Fully Connected Layer	1000	

200 First, the network was trained for the original classes. The outputs are used as feature vectors to generate centroids using k-means clustering. Thus, the number of clusters per class can be greater than one (see Fig. 2).

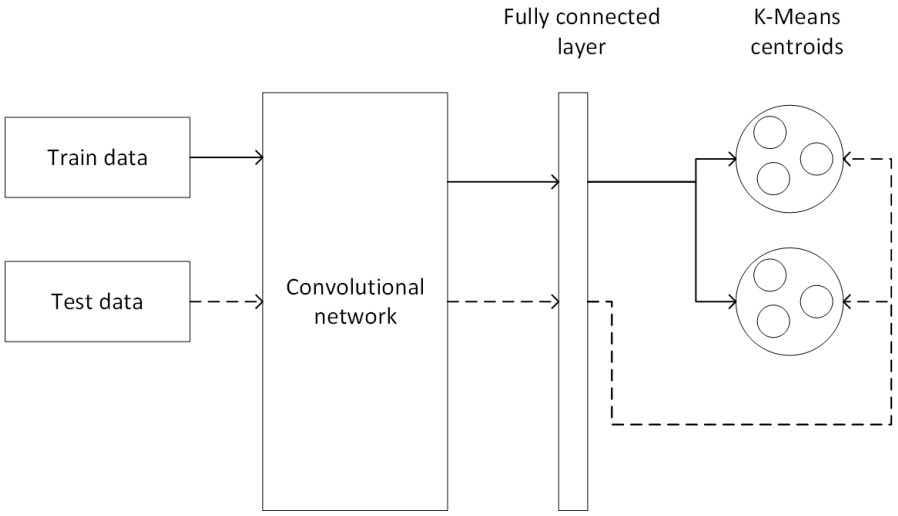


Figure 2: Non-incremental training and test step

The loss function is based on a function for clustering methods that aims to increase the distance between centroids of different classes and to reduce the

distance between centroids of the same class. An adaptation was implemented to consider that multiple centroids could be calculated for each class. The training instances were fed through the network to generate the centroids. Subsequently, for each instance, distance function d computes the closest centroid C to assign a class, as shown in Eq. 1,

$$L = -\sum_{i=1}^{n} log \frac{-e^{argmin(d(v_i, c_p))}}{\frac{1}{K}\sum_{m=1}^{K} -e^{d(v_i, c_m)}} \tag{1}$$

where d is the distance function, n are instances, K is the number of classes, C are the class centroids, C_p is the centroid of the class corresponding to actual instance i, and V are the feature vectors of the instances.

To test the performance of the network, the test data followed the same approach; they were fed through the network and labeled according to the closest centroid.

Once the network is trained with the *original* classes, a *new* class can be added. To achieve this, a new network was created and trained without disturbing or retraining the old network, as shown in Fig. 3.

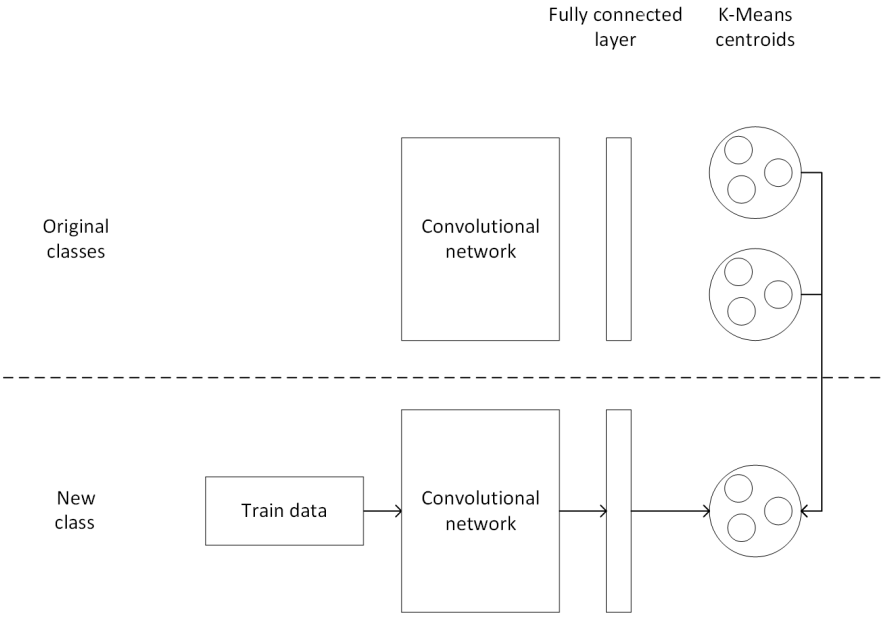


Figure 3: Incremental training step

The centroids of the old network were considered in the distance function to compute the loss function of the new network. This is similar to *knowledge distillation*. However, in the proposed approach, the original network is similar in size to the new network and there are no exemplar instances that retrain the original network.

Finally, to test the performance of the model with the new class, the test data (instances of both the original and new classes) were fed through both networks and the distances to the centroids were saved into two distance matrices. Thus, for the classification step, both matrices are compared, and the lower values are preserved in a final distance matrix that labels the test instances (Fig. 4).

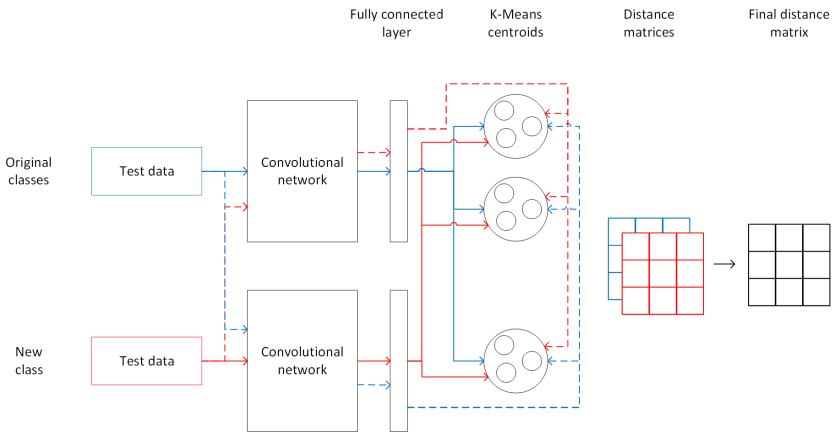


Figure 4: Incremental test step

The non-incremental step of the method is presented in Algorithm 1, where $O\_TrainData$ and $O\_TestData$ refer to the non-incremental dataset. $I\_TrainData$ and $I\_TestData$ are the incremental datasets. O_Labels and I_Labels denote corresponding labels. C_O and C_I are non-incremental and incremental classes, respectively.

Algorithm 1 Non-incremental network

Input: $O\_TrainData$, $O\_TestData$, $I\_TrainData$, $I\_TestData$, O_Labels, I_Labels, C_O, C_I

Output: y

$Net_1 \leftarrow Initialize$ ▷ Netwwork Training
for $Epoch = 1$ **To** 100 **do**
    $X \leftarrow$ Feed Net_1 with $O\_TrainData$
    for all $Class$ **In** C_O **do**
        $K \leftarrow$ kMeans($X[Class]$)
    end for
    for $Batch = 1$ **To** 40 **do**
        $X \leftarrow$ Feed Net_1 with $O\_TrainData[Batch]$
        $D \leftarrow distance(K, X)$ ▷ See Algorithm 3
        $Loss \leftarrow \ell(D)$ ▷ See Eq. 1
        $Net_1 \leftarrow Backpropagation(Net_1, Loss)$
    end for
end for
$X \leftarrow$ feed with $O\_TrainData$ ▷ Network testing
$D \leftarrow distance(K, X)$ ▷ See Algorithm 3
$y \leftarrow evaluate(D, O\_Labels)$ ▷ See Algorithm 4

Algorithm 2 presents the incremental step of the method, i.e., when a new class is added.

Algorithm 2 Incremental network

$Net_2 \leftarrow Initialize$ ▷ Incremental Network Training
for $Epoch = 1$ **To** 100 **do**
    $X \leftarrow$ feed Net_2 with $I\_TrainData$
    for all $Class$ **In** C_I **do**
        $K \leftarrow$ kMeans($X[Class]$) ▷ K contains previous centroids and the new
    end for
    for $Batch = 1$ **To** 40 **do**
        $X \leftarrow$ feed Net_2 with $I\_TrainData[Batch]$
        $D \leftarrow distance(K, X)$ ▷ See Algorithm 3
        $Loss \leftarrow \ell(D)$ ▷ See Eq. 1
        $Net_2 \leftarrow Backpropagation(Net_2, Loss)$
    end for
end for
$X_1 \leftarrow$ feed Net_1 with $O\_TestData$ ▷ Incremental Network Testing
$X_2 \leftarrow$ feed Net_1 with $I\_TestData$
$D_1 \leftarrow distance(K, [X_1, X_2])$ ▷ See Algorithm 3
$X_1 \leftarrow$ feed Net_2 with $O\_TestData$
$X_2 \leftarrow$ feed Net_2 with $I\_TestData$
$D_2 \leftarrow distance(K, [X_1, X_2])$ ▷ See Algorithm 3
for all $Column$ **In** (D_1 **Or** D_2) **do**
    for $Row = 1$ **To** $lenght(K)$ **do**
        $D_f \leftarrow min(D_1[Column, Row], D_2[Column, Row])$
    end for
end for
$y \leftarrow evaluate(D_f, [O\_Labels, I\_Labels])$ ▷ See Algorithm 4

Algorithm 3 presents the distance function used to generate distance matrices, where X is the network output, and K is the set of k-means centroids of all classes. The cosine distance was chosen because of the dimensionality of the

data.

Algorithm 3 function distance(K,X)

Input: K, X

Output: D

for all $Element$ **In** X **do**
  for all $Centroid$ **In** K **do**
    $D[Element, Centroid] \leftarrow cosine\_distance(Element, Centroid)$
  end for
end for

Algorithm 4 was developed to evaluate the calculated distances, where D is a distance matrix in which the columns contain the instances of the test data, the rows contain the centroids, and $Labels$ denote the true class of the instances.

Algorithm 4 function evaluate(D,Labels)

Input: D, $Labels$

Output: y

$y \leftarrow 0$
for all $Column$ **In** D **do**
  $X \leftarrow min(D[Column, :])$
  if $X = Label[Column]$ **then**
    $y \leftarrow y + 1$
  end if
end for
$y \leftarrow y/length(Labels)$

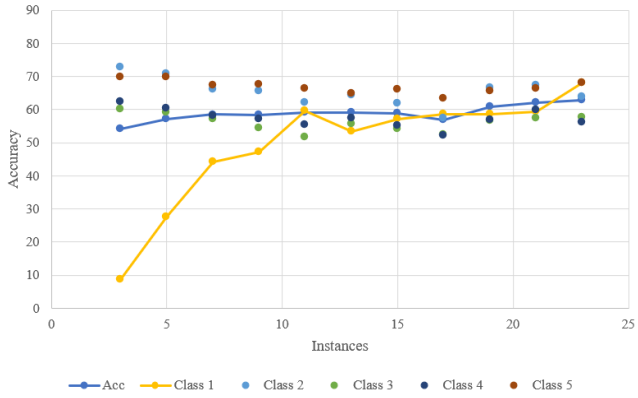
4. Results

Processing was carried out on a dedicated server with two Intel Xeon-Gold 6248 (2.5 Ghz, 84 cores) and eight NVIDIA Tesla V100 32 GB graphic cards (40,960 CUDA cores) using Matlab 2021 and Anaconda 4.8.2, with Python 3.7.11.

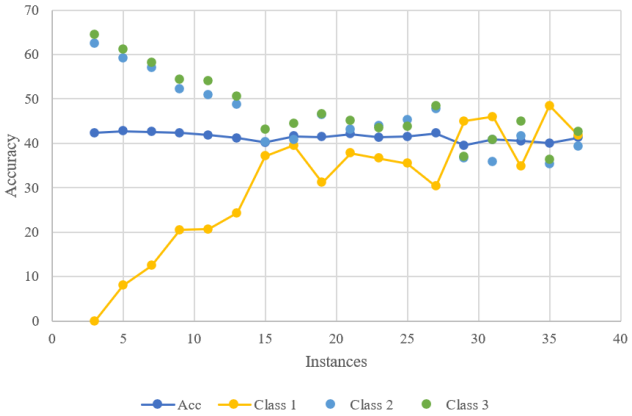
For all experiments, the following considerations were taken: the number of subjects was adjusted to six for all databases; the results were given as the average of the six subjects; the data were split into 80% for training and 20% for testing purposes; each experiment was repeated five times per subject due to the stochastic behavior of the employed methods; the new class was added to the model using a different number of training instances that started with one and increased by two until the complete training instances were used; the complete training set was used for original classes; the incremental class was taken from the dataset and this class was removed for the original classes and added for the incremental training; the incremental class was obtained from each dataset separately, this class was then removed from the original training classes and added to the incremental training; for all datasets, class 1 was used as the incremental class.

The number of centroids was defined empirically by experimentation on a few samples, and because of the performance decrease using a high number of centroids, it was decided to use as few centroids as possible, that is, 3 and 1. Moreover, the number of centroids was the same for all classes, including the new class.

In Fig. 5 the results for the three datasets are presented. As mentioned previously, the results for each dataset are the average performance of the subjects. The figures show the behavior of all classes; the incremental class is highlighted in continuous yellow and the total accuracy in continuous blue. Instances of the new class were progressively added to the model to observe the trade-off between the number of instances and the final accuracy.



(a) Results for **5C** dataset



(b) Results for **3C** dataset

Figure 5: Results using 3 centroids. The blue line represents the total accuracy, and the yellow line represents the incremental class accuracy

As shown in Table 2, the **5C** dataset obtained an accuracy of 59.02 ± 2.7 using one centroid and 58.89 ± 2.4 using three centroids; there was no statistical difference in this case. The accuracies obtained in the **3C** dataset were 41.43 ± 0.9 and 36.91 ± 0.59 for three and one centroids, respectively, indicating better performance with three centroids. The incremental learning accuracies of the three datasets did not show statistical differences; however, some cases showed a faster increase when using fewer instances. The incremental network showed a high total accuracy and a faster increase in incremental class accuracy. The

increase in the incremental class accuracy using a few instances is advantageous for BCIs because it can reduce the time required for training.

Table 2: Incremental network results. A t-test for total accuracy obtained: $p = 0.5221$ for **5C** dataset and $p = 0.12103e^{-18}$ for **3C** dataset. And, respectively, the t-test for incremental class accuracy obtained: $p = 12.09$ and $p = 0.6041$

Dataset	**1 centroid**		**3 centroids**	
	Total	Incremental	Total	Incremental
5C	59.02 ± 2.78	55.38 ± 20.49	58.89 ± 2.4	49.29 ± 17.21
3C	36.91 ± 0.59	26.91 ± 12.52	41.43 ± 0.9	30.57 ± 13.63

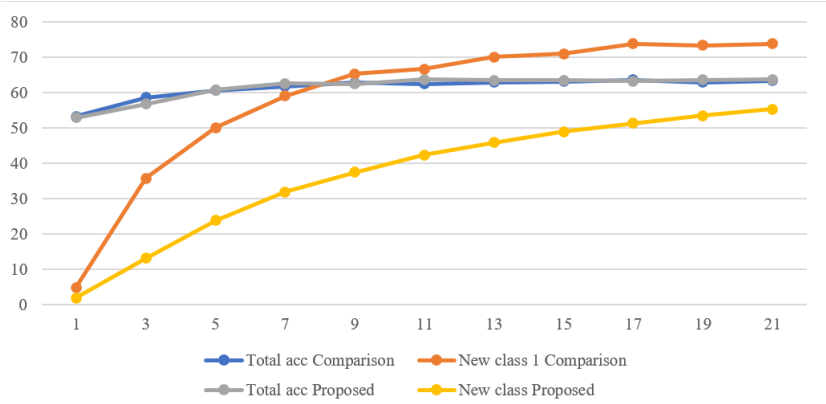
4.1. Additional validation of the method

The proposed method was compared to a method similar to that applied to a different task. This provides experimental evidence of the competitive performance of incremental learning when applied to small dataset scenarios.

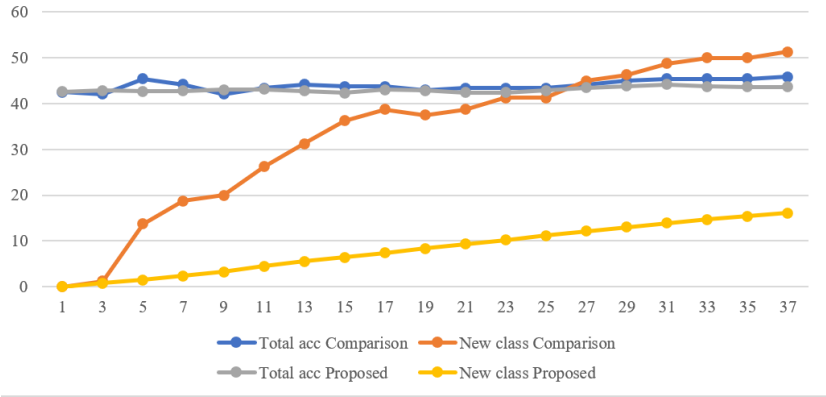
The results obtained were compared with those of [16], who presented an incremental approach to odor classification. In this study, the outputs of the neural network were replaced by means of the classes, that is, a mean centroid, and there was one centroid per class. Subsequently, when a new class was added, new instances were fed through the same network, generating a new centroid.

Despite the use of a different task, the method was oriented to a scenario similar to that proposed in this study, in which a few classes were added to the neural network approach.

Comparative results are shown in Fig. 6.



(a) Results for **5C** dataset



(b) Results for **3C** dataset

Figure 6: The results of the proposed method are in color blue for the total accuracy and the incremental class accuracy in color red. For the [16] method, the total accuracy is presented in color gray and the incremental class accuracy in color yellow.

295 The comparative results with the [16] approach show that both methods have a similar stable total accuracy performance for all datasets. Nevertheless, the incremental accuracy results showed a notable difference, see Table 3. In all cases, the incremental accuracy was higher when the proposed method was used and began to increase using fewer instances.

Table 3: Comparative accuracy results. A t-test for total accuracy obtained: $p = 0.9530$ for **5C** dataset and $p = 0.0043$ for **3C** dataset. Respectively, t-test for incremental class accuracy showed: $p = 0.0047$ and $p = 1.2958e^{-07}$.

Dataset	**Proposed method**		**Baseline method**	
	Total	Incremental	Total	Incremental
5C	61.7 ± 3.16	58.84 ± 20.91	61.45 ± 3.51	36.8 ± 17.5
3C	43.96 ± 1.18	33.48 ± 15.97	43.06 ± 0.53	8.23 ± 5.23

Both approaches maintain a total accuracy that does not decay for any number of instances of the new class, i.e., there is no catastrophic forgetting. Nevertheless, the comparison showed two improvements in the proposed incremental approach: the incremented class achieved stable accuracy in a few instances. In addition, a higher incremental accuracy was achieved. These improvements provide the method with a faster adaptation of the new class using fewer instances while maintaining total accuracy.

5. Conclusions

The results of the incremental network showed stable total accuracy and no drop that resembled *catastrophic forgetting*. For the dataset **5C**, there was a good relationship between *stability* and *plasticity* for each subject. For the **3C** dataset, some subjects exhibited a decrease in accuracy for the old classes, which tended to recover when more instances of the new class were added.

A possible explanation for the variations in the behavior of the datasets is that they differ in the number of channels. A small number of channels may allow a better fit of the network with a few parameters, which can be explored in future experiments by reducing the number of channels of some datasets or increasing the parameters of the network. Another consideration is the number of classes: the **5C** dataset includes five classes, whereas the **3C** dataset includes only three classes. A higher number of old classes may allow robustness of the model that is not perturbed by the inclusion of a new class. Furthermore,

the acquisition protocol was different for each dataset: **5C** allowed subjects to manually determine when they had finished imagining the words, and **3C** fixed a period in which the subjects could repeat the imagination of the word several times.

Finally, the proposed method achieved good performance in contrast to other incremental class learning tasks, particularly incremental learning for images. In [40, 44], a review of various incremental learning methods was presented, most of which were focused on mitigating catastrophic forgetting. These studies showed that the total accuracy decays as more instances of a new class are added, which does not occur in the proposed method. It is important to emphasize that these are different tasks with significant differences. These studies used large image datasets, such as CIFAR100, which contains a large number of classes and instances.

The incremental network performed better for incremental learning than the method proposed in [16]. The results indicated a faster increase in accuracy when fewer instances were used. This behavior is desirable for BCI because of the reduction in data acquisition time.

6. Acknowledgments

The present work was partially supported by CONACyT (scholarship 401887). The authors also thank CONACyT for the computer resources provided through the INAOE Supercomputing Laboratory's Deep Learning Platform for Language Technologies. The second author also thanks CONACyT for supporting this research with a postdoctoral fellowship.

References

[1] L. A. Farwell, E. Donchin, Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials, Electroencephalography and clinical Neurophysiology 70 (6) (1988) 510–523.

[2] A. A. Torres-García, C. A. Reyes-García, L. Villaseñor-Pineda, G. García-Aguilar, Implementing a fuzzy inference system in a multi-objective (EEG) channel selection model for imagined speech classification , Expert Systems with Applications 59 (2016) 1 – 12. doi:http://dx.doi.org/10.1016/j.eswa.2016.04.011.

[3] A. Gepperth, B. Hammer, Incremental learning algorithms and applications, in: European symposium on artificial neural networks (ESANN), 2016, pp. 2–3.

[4] R. R. Ade, P. R. Deshmukh, Methods for Incremental Learning: A Survey, International Journal of Data Mining & Knowledge Management Process (IJDKP) 3 (4) (2013) 119–125. doi:10.5121/ijdkp.2013.3408.

[5] R. Polikar, L. Upda, S. S. Upda, V. Honavar, Learn++: An incremental learning algorithm for supervised neural networks, IEEE transactions on systems, man, and cybernetics, part C (applications and reviews) 31 (4) (2001) 497–508.

[6] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, K. Alahari, End-to-end incremental learning, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 233–248.

[7] N. R. Waytowich, V. J. Lawhern, A. W. Bohannon, K. R. Ball, B. J. Lance, Spectral transfer learning using information geometry for a user-independent brain-computer interface, Frontiers in Neuroscience 10 (SEP). doi:10.3389/fnins.2016.00430.

[8] G. Panagopoulos, Multi-Task Learning for Commercial Brain Computer Interfaces, 17th International Conference on Bioinformatics and Bioengineering (2017) 86–93doi:10.1109/BIBE.2017.00022.

[9] C.-S. Wei, Y.-P. Lin, Y.-T. Wang, C.-T. Lin, T.-P. Jung, A subject-transfer framework for obviating inter- and intra-subject variability in eeg-based drowsiness detection, NeuroImage 174 (2018) 407 – 419.

doi:https://doi.org/10.1016/j.neuroimage.2018.03.032.
URL http://www.sciencedirect.com/science/article/pii/S1053811918302428

[10] H. He, D. Wu, Transfer learning enhanced common spatial pattern filtering for brain computer interfaces (bcis): Overview and a new approach, in: International Conference on Neural Information Processing, Springer, 2017, pp. 811–821.

[11] F. Lotte, C. Guan, Learning from other subjects helps reducing brain-computer interface calibration time, in: 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, 2010, pp. 614–617. doi:10.1109/ICASSP.2010.5495183.

[12] R. M. French, Catastrophic forgetting in connectionist networks, Trends in Cognitive Sciences 3 (4) (1999) 128–135. doi:https://doi.org/10.1016/S1364-6613(99)01294-2.
URL https://www.sciencedirect.com/science/article/pii/S1364661399012942

[13] X. Ye, Q. Zhu, Class-Incremental Learning Based on Feature Extraction of CNN with Optimized Softmax and One-Class Classifiers, IEEE Access 7 (c) (2019) 42024–42031. doi:10.1109/ACCESS.2019.2904614.

[14] M. Hasan, A. K. Roy-Chowdhury, Incremental activity modeling and recognition in streaming videos, in: 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 796–803. doi:10.1109/CVPR.2014.107.

[15] S. A. Rebuffi, A. Kolesnikov, G. Sperl, C. H. Lampert, iCaRL: Incremental classifier and representation learning, Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 2017-January (2017) 5533–5542. arXiv:1611.07725, doi:10.1109/CVPR.2017.587.

[16] Y. Cheng, K. Wong, K. Hung, W. Li, Z. Li, J. Zhang, Deep nearest class mean model for incremental odor classification, IEEE Transactions on Instrumentation and Measurement 68 (4) (2019) 952–962.

[17] H. Morioka, A. Kanemura, J. ichiro Hirayama, M. Shikauchi, T. Ogawa, S. Ikeda, M. Kawanabe, S. Ishii, Learning a common dictionary for subject-transfer decoding with resting calibration, NeuroImage 111 (2015) 167–178. doi:10.1016/j.neuroimage.2015.02.015.
URL https://ac.els-cdn.com/S1053811915001160/1-s2.0-S1053811915001160-main.pdf?{_}tid=15b71e78-051c-11e8-9113-00000aacb360{&}acdnat=1517248070{_}d436f5eca44af35b0542e307f55c3afa

[18] V. Jayaram, M. Alamgir, Y. Altun, B. Scholkopf, M. Grosse-Wentrup, Transfer Learning in Brain-Computer Interfaces, IEEE Computational Intelligence Magazine 11 (1) (2016) 20–31. arXiv:1512.00296, doi:10.1109/MCI.2015.2501545.

[19] H. Kang, Y. Nam, S. Choi, Composite common spatial pattern for subject-to-subject transfer, IEEE Signal Processing Letters 16 (8) (2009) 683–686. doi:10.1109/LSP.2009.2022557.

[20] M. Wronkiewicz, E. Larson, A. K. C. Lee, Leveraging anatomical information to improve transfer learning in brain computer interfaces, Journal of Neural Engineering 12 (4) (2015) 046027.
URL http://stacks.iop.org/1741-2552/12/i=4/a=046027

[21] W. Tu, S. Sun, A subject transfer framework for EEG classification, Neurocomputing 82 (2012) 109–116. doi:10.1016/j.neucom.2011.10.024.
URL http://dx.doi.org/10.1016/j.neucom.2011.10.024

[22] M. Dai, D. Zheng, S. Liu, P. Zhang, Transfer kernel common spatial patterns for motor imagery brain-computer interface classification, Computational and mathematical methods in medicine 2018.

[23] H. Kober, M. Mö, C. Nimsky, J. Rgen Vieth, R. Fahlbusch, O. Ganslandt, New Approach to Localize Speech Relevant Brain Areas and Hemispheric Dominance Using Spatially Filtered Magnetoencephalography, Hum. Brain Mapping 14 (2001) 236–250. doi:10.1002/hbm.XXXX.
URL http://www.neuropsychiatrie.med.uni-erlangen.de/expneuro/pdf/kober-speech-meg.pdf

[24] D. Perani, S. Dehaene, F. Grassi, L. Cohen, S. F. Cappa, E. Dupoux, F. Fazio, J. Mehler, Brain processing of native and foreign languages, NeuroReport-International Journal for Rapid Communications of Research in Neuroscience 7 (15) (1996) 2439–2444.

[25] S. S. Shergill, M. J. Brammer, R. Fukuda, S. C. Williams, R. M. Murray, P. K. McGuire, Engagement of brain areas implicated in processing inner speech in people with auditory hallucinations, British Journal of Psychiatry 182 (JUNE) (2003) 525–531. doi:10.1192/bjp.182.6.525.
URL http://bjp.rcpsych.org/content/bjprcpsych/182/6/525.full.pdf

[26] A. Aleman, E. Formisano, H. Koppenhagen, P. Hagoort, E. H. F. De Haan, R. S. Kahn, The functional neuroanatomy of metrical stress evaluation of perceived and imagined spoken words, Cerebral Cortex 15 (2) (2005) 221–228. doi:10.1093/cercor/bhh124.
URL https://watermark.silverchair.com/bhh124.pdf?token=AQECAHi208BE49Ooan9kkhW{_}Ercy7Dm3ZL{_}9Cf3qfKAc485ysgAAAakwggGlBgkqhkiG9w0BBwagggGWMIIB

[27] S. Deng, R. Srinivasan, M. D'Zmura, Cortical signatures of heard and imagined speech envelopes, Tech. rep., CALIFORNIA UNIV IRVINE DEPT OF COGNITIVE SCIENCES (2013).

[28] N. F. Wymbs, R. J. Ingham, J. C. Ingham, K. E. Paolini, S. T. Grafton, Individual differences in neural regions functionally related to real and imagined stuttering, Brain and Language 124 (2) (2013) 153–164. doi:10.1016/j.bandl.2012.11.013.

URL https://ac.els-cdn.com/S0093934X12002179/1-s2.0-S0093934X12002179-main.pdf?{_}tid=spdf-3aca694e-d084-4844-b5a2-c0203a63d705{&}acdnat=1519765182{_}61eeb886112285f0277f36ed1c822f76

[29] P. K. McGuire, D. A. Silbersweig, R. M. Murray, A. S. David, R. S. J. Frackowiak, C. D. Frith, Functional anatomy of inner speech and auditory verbal imagery, Psychological Medicine 26 (01) (1996) 29. doi:10.1017/S0033291700033699.
URL http://www.journals.cambridge.org/abstract{_}S0033291700033699

[30] P. Suppes, Z.-L. Lu, B. Han, Brain wave recognition of words, Psychology 94 (1997) 14965–14969.
URL http://www.pnas.org/content/pnas/94/26/14965.full.pdf

[31] M. Salama, H. Lashin, T. Gamal, Recognition of unspoken words using electrode electroencephalograhic signals, COGNITIVE 2014 : The Sixth International Conference on Advanced Cognitive Technologies and Applications (2014) 51–55.

[32] J. S. García-Salinas, L. Villaseñor-Pineda, C. A. Reyes-García, A. A. Torres-García, Transfer learning in imagined speech eeg-based bcis, Biomedical Signal Processing and Control 50 (2019) 151–157.

[33] G. A. Pressel Coretto, I. E. Gareis, H. L. Rufiner, Open access database of eeg signals recorded during imagined speech, Proc. SPIE 10160. doi:10.1117/12.2255697.

[34] C. H. Nguyen, G. Karavas, P. Artemiadis, Inferring imagined speech using EEG signals: a new approach using Riemannian Manifold features, Journal of Neural Engineeringdoi:10.1088/1741-2552/aa8235.
URL http://iopscience.iop.org/10.1088/1741-2552/aa8235

[35] T. Xiao, J. Zhang, K. Yang, Y. Peng, Z. Zhang, Error-driven incremental learning in deep convolutional neural network for large-scale image classification, in: Proceedings of the 22Nd ACM International Conference on Multimedia, MM '14, ACM, New York, NY, USA, 2014, pp. 177–186. doi:10.1145/2647868.2654926.
URL http://doi.acm.org/10.1145/2647868.2654926

[36] D. Roy, P. Panda, K. Roy, Tree-CNN: A hierarchical Deep Convolutional Neural Network for incremental learning, Neural Networks 121 (2020) 148–160. arXiv:1802.05800, doi:10.1016/j.neunet.2019.09.010.
URL https://doi.org/10.1016/j.neunet.2019.09.010

[37] S. S. Sarwar, A. Ankit, K. Roy, Incremental Learning in Deep Convolutional Neural Networks Using Partial Network Sharing, IEEE Access 8 (2020) 4615–4628. arXiv:1712.02719, doi:10.1109/ACCESS.2019.2963056.

[38] J. Zhang, J. Zhang, S. Ghosh, D. Li, S. Tasci, L. Heck, H. Zhang, C. C. J. Kuo, Class-incremental learning via deep model consolidation (2020). arXiv:1903.07864.

[39] Y. Hao, Y. Fu, Y. G. Jiang, Q. Tian, An end-to-end architecture for class-incremental object detection with knowledge distillation, Proceedings - IEEE International Conference on Multimedia and Expo 2019-July (2019) 1–6. doi:10.1109/ICME.2019.00009.

[40] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, Y. Gong, Few-shot class-incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12183–12192.

[41] D. Wu, Y. Xu, B.-L. Lu, Transfer learning for eeg-based brain-computer interfaces: a review of progress made since 2016, IEEE Transactions on Cognitive and Developmental Systems.

[42] H. He, D. Wu, Different set domain adaptation for brain-computer inter-

faces: a label alignment approach, IEEE Transactions on Neural Systems and Rehabilitation Engineering 28 (5) (2020) 1091–1108.

[43] P. Stoica, R. Moses, Spectral Analysis of Signals, Pearson Prentice Hall, 2005.
URL https://books.google.com.ar/books?id=h78ZAQAAIAAJ

[44] M. Masana, X. Liu, B. Twardowski, M. Menta, A. D. Bagdanov, J. van de Weijer, Class-incremental learning: survey and performance evaluation, arXiv preprint arXiv:2010.15277.