Neurocomputing. 2020; 410: 220-228

EEG signal processing with separable convolutional neural network for automatic scoring of sleeping stage

Enrique Fernandez-Blanco[a], Daniel Rivero[a], Alejandro Pazos[a,b]

[a] *Faculty of Computer Science, CITIC, University of A Coruna, A Coruña 15071, Spain*
[b] *INIBIC, Complexo Hospitalario Universitario de A Coruña, A Coruña 15006, Spain*

Abstract

Nowadays, among the Deep Learning works, there is a tendency to develop networks with millions of trainable parameters. However, this tendency has two main drawbacks: overfitting and resource consumption due to the low-quality features extracted by those networks. This paper presents a study focused on the scoring of sleeping EEG signals to measure if the increase of the pressure on the features due to a reduction of the number though different techniques results in a benefit. The work also studies the convenience of increasing the number of input signals in order to allow the network to extract better features. Additionally, it might be highlighted that the presented model achieves comparable results to the state-of-the-art with 1000 times less trainable and the presented model uses the whole dataset instead of the simplified versions in the published literature.

Keywords

Convolutional neural networks; Deep learning; EEG; Signal processing; Sleep scoring

1. Introduction

According to one the reports from the World Health Organisation (WHO) [1], sleeping problems are one of the major issues for many people around the world independently of the country and social role. Problems to obtain a good quality sleep have been related to other major diseases and disorders like depression, stress, heart problems, diabetes or early Alzheimer [2]. Consequently, in recent times, the resources dedicated to study sleeping problems have been increased and the sleeping units in most of the hospitals have multiplied their activity [3]. Although this is a good new, the increment in the activity has driven the specialist to a difficult situation with a huge amount of data to process.

Any specialist from an sleeping unit has as main working tool the polysomnography (PSG), which is a recording of different signals registered through the night. Those records include different signals such as Electrocardiograms, Electroencephalograms (EEGs), breath or movement records, registered for a patient during a period between 6 and 24 h. After that registering process, the specialists have to manually label the different stages of the sleep and the result is a high time-consuming processes which is quite prone to mistakes due to fatigue or monotony [4].

Many researchers have focused their attention on automatising this process. To do that, a common approach is to use the EEGs in the PSG to label the sleeping stage according to one of two main widely accepted guidelines known as Rechtschaffen & Kales (R&K) [5] and the American Academy of Sleep Medicine (AASM) [6], respectively. The latter one is the current standard defining a set of 5 stages that a patient can go through during his sleep, however, some studies have point out to 81% as the agreement ratio among experts when labelling the same PSG [7]. The main reason for that disagreement is the nonexistence of a common set of features identified and used by all of them. This is one of the mayor issues when automatising through machine learning has been tried, the unreliability on the background truth.

In order to deal with this issue, some works have focused on using automatic feature extraction techniques, such as, Genetic Programming [8], although recently Convolutional Neural Networks have gained great popularity among the scientific community due to their remarkable achievements in different tasks like sound processing or detection of different elements in pictures [9], [10]. However, most of the previous applications present oversized models with to many parameters to be adjusted. This has two main collateral effects: first, the models needs a tremendous amount of resources for its training and execution and second, more parameters induce the models to memorise and consequently to overfit the solution.

This paper uses the whole dataset known as SHHS-1 [11] in order to test on a real-world problem how the increment on the pressure on the convolutional layers could drive to a reduction in the requirements and a better generalisation. Additionally, a first approach to solve the aforementioned dataset is provided without removing the outlayers or reducing the number of patients. The aim is to test the architecture and the different approaches on a real-life problem.
The paper presents on Section 2 a review of related works with special attention to the most modern ones which have used part of the dataset. Section 3 presents the proposed architecture, a summary of the key elements of convolutional neural networks and a description of the dataset used on the tests which are contained in 4. Finally, Section 5 and Section 6 present the conclusions and the future lines to work, respectively.

2. State of the art

Over the years, researchers have done uncountable efforts to improve the extraction of features from different types of signals. One that has attracted particular interest has been Electroencephalograms (EEGs), trying to improve the solution of problems such as the detection of epileptic seizures [12], [13], evaluation of the state of the subject [14], [15] or the development of Brain-Controlled Interfaces [16], [17]. Among the problems related to EEG processing, one which has particularly focused a lot of attention is the scoring of the sleeping stages.

Focusing on the latter mentioned problem, many researchers have attempted to use different approaches to improve the automation of this task. However, if the spotlight is put over the automatic extraction of features, the amount of works drops down significantly. The first work worth to be mentioned is [18], where the authors use of Fuzzy Logic in combination with an iterative classification method in order to perform the identification of the different sleeping stages. This work uses EEGs keeping the timeline and tries to extract the features from it. Going along the same line of keeping the timeline, [19] proposed to apply a decision tree to classify the features extracted from the raw signal. Later that year, the same researchers propose the use of a multiscale entropy combined with a simple Linear Discriminant Analysis (LDA), which overcomes their previous results [20]. A different approach was presented in [21], where a Hidden Markov Model was proposed to perform the classification.

On the other hand, other works have preferred not to keep the timeline and look for features in alternative search spaces, like [22], which performs a wavelet transform before extracting different features and use them with different machine learning techniques to classify the signals. Another example of alternative representation used is [23] where the authors use a combination of the energies from different frequency bands to perform the classification with an Artificial Neural Network (ANN).

Partially related, it is worth to mention those works which do not use the signal as it is while they focus their attention on some high-level features such as statistical features [24], power spectral density [25], graph theory features [26] or moment features [27].

Finally, in recent years, some works have focused their attention on the uprising techniques framed under the name of Deep Learning by using Random Belief Networks [28] or Convolutional Neural Networks [29]. All of those techniques are based on the same principle, each new layer of the neural networks extracts higher-level features from the information on the input. Applying this principle to EEGs, [30] proposed the use of Autoencoders to solve the labeling problem on a 20-patient dataset [31]. In [32], the authors proposed a dual pipeline network called DeepSleep which overcome the results in [30] on the same dataset and, additionally, the paper also presents a second set of results on a different dataset [33]. Going along with those developments, Fernandez-Blanco et al. in [34] studies on the same dataset the advantages of using multiple signals with a convolutional neural network pointing out to an advantage of processing several EEG signals simultaneously.

However, deep learning works previously mentioned use 2 relatively small datasets, more related to the work presented in this paper, the most relevant results due to the size of the dataset used can be found in [35]. That work uses the dataset known as SHHS-1 [11], which contains more than 5800 records from patients. However, the authors of that work have removed the outlayered data and trimmed the number of the wake periods, simplifying this problem while results get far from reality. Even though, the proposal is worth mentioned because using a set of convolutional layers is capable of extracting features and use them as the input of a Multilayer Perceptron for classification.

The main problem of the previously mentioned works is the use of convolutional layers which extract hundreds of low-quality feature, those low-quality features result in the requirement of more resources to train and use the models. This work explores if there is a possibility to obtain similar or better results but reducing the hardware requirements by increasing the pressure on the quality of the features extracted in each layer through reducing the number of filters to learn and the use of dropout layers. This reduction of the number of features also comes along with improved control of the overfitting because each feature has to represent more general knowledge in order to cover as much of the search space as possible.

3. Materials and methods

3.1. Convolutional neural networks

Convolution in Neural Networks was proposed for the first time by Fukushima in 1982 [36]. Although in 1998 Yann LeCunn [29] proposed its use in the recognition of documents, until 2012 [37] with the modifications proposed by Hinton et al. in the calculation of the gradients the use and construction of CNN was restricted to very small laboratory problems. However, since those advances, Convolutional Neural Networks (CNN) and Deep Learning, in general, have meant an important step forward in many knowledge areas by becoming the state-of-the-art for many problems.

CNNs establish a hierarchy of layers where each neuron receives as inputs a spatial-close related piece of the input information. Each neuron on a convolutional layer receives a different input data like a sliding window over the signal or image, and the weights are the same for all neurons instead of being different, as in classical neural networks. Consequently, the result is the convolution of an input feature map $X(l-1)$ with a set of learnable filters $W(l)$ and adds biases $b(l)$ to, finally, apply some kind of transfer function g like in Eq. (1)

$$X^{(l)}=g^{l}(X^{(l-1)}*W^{(l)}+b^{(l)}) \quad (1)$$

By stacking several layers that apply the Eq. (1), the result is a network where each layer extracts more general information from the information on the previous layer but conditioned by the spatial relationship [38]. Consequently, CNNs are composed by a number of convolutional layers which extracts the features of the signals or images. After the extraction, the resulting features are used with some sort of machine learning technique such as Support Vector machines or fully-connected perceptrons for the classification problem, while a Softmax layer could be used for regression. This architecture has been successfully used in many applications, although most of these works are mainly related to image processing, such as face recognition [39] or image classification [40], while signal processing is quite limited to natural language processing [41] or human voice recognition [42].

3.1.1. Depthwise separable convolution

Although CNNs have been a successful change on the tack for many problems in recent years, one of its major issues is how expensive in time and resources this operation is. To deal with it, one of the major contributions has been the Pointwise Separable Convolution layers [43], which reduce drastically the computational cost.

$$\#Operations_{Conv}=Ch*K_{sz}*(S-K_{sz})*F \quad (2)$$

$$\#Operations_{SepConv}=Ch*K_{sz}*(S-K_{sz})+Ch*(S-K_{sz})*F \quad (3)$$

This type of convolution is the result of combining two simple convolutions, first, a spatial convolution which is performed on each input channel to extract a small number of features. The second convolution is performed as a point-wise convolution, i.e 1x1 convolution, to projects the results of the combination of the channels in a new space as it can be seen in Fig. 1. The benefits in time and memory costs of the application of this kind of convolution are evident if we examine the equations Eqs. (2), (3).

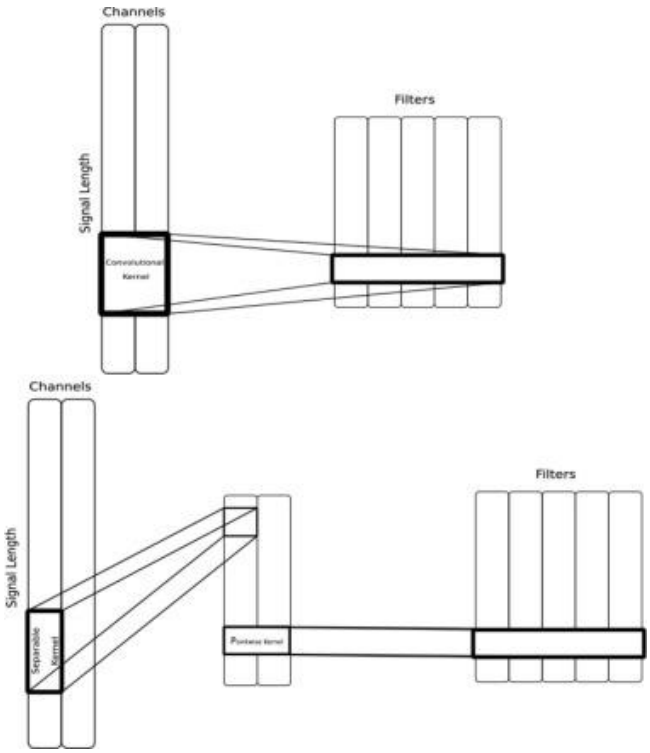

Fig. 1. Conventional Convolution and its equivalent Deapthwise Separable Convolution.

The number of operation in the traditional convolution (#OperationsConv) is represented in Eq. (2), where Ch represents the number of channels, Ksz is the size of the kernel to be applied, F the number of filters and S is the length of the signal. Comparing that amount to the number of operations in a separable convolution (#OperationsSepConv) which is represented in Eq. (3), the results is a lower the ratio between the number of operations for a normal convolution and a separated one (Eq. (4)). According to this relationship, the reduction is inversely proportional to the size of the Kernel (Ksz) and the number of filters (F). Therefore, Deepthwise Separable Convolutional is cheaper while its results are equivalent to a normal convolution.

$$\frac{\#Operations_{SepConv}}{\#Operations_{Conv}} = \frac{1}{K_{sz}} + \frac{1}{F} \tag{4}$$

For example, in a single convolutional layer for a single dimension like the ones used in this paper, suppose that we use a kernel of size 7×2 and we extract 10 filters with each convolution. Therefore the result of the relation in Eq. (4) for that kernel will be 114+110=0.1714, that is an 82.85% of fewer gradients to be calculated and updated in a single layer. Additionally, fewer weights also come with better control of the overtraining because the networks have less room to memorize instead of learning.

3.1.2. Overfitting

One of the biggest issues in the application of machine learning in general, and deep learning in particular, is the natural tendency of the training techniques to overfit the models [44], [45], [46]. When a model, instead of extracting the knowledge behind the presented patterns, is mainly memorizing them, it is what is called an overfitted model. Over the years, many approaches have tried to tackle these questions, like the use of validation sets or the inclusion of different penalization terms in the optimization techniques.

This problem is even bigger in the Deep Learning models because of the increment in the number of weights drives to a higher capability of memorization by the network. Several proposals can be found in the literature, for example, introducing a special layer between the other known as the Dropout layer, which will drop some outputs from a previous layer forcing the following to relay on other connections. In this way, the training process tries to spread knowledge over the whole bunch of connections instead of specific ones. However, the problem is still rooted because the number of parameters in the networks tends to increase along with the capabilities of the systems. That is the reason why, the only effective way to control the overfit, is minimizing the number of weights and biases in a network.

3.1.3. Evaluation criteria

In order to contribute to control the overfitting of the model, the dataset was split into three datasets, training, validation and tests, according to 0.7, 0.1 and 0.2 ratios, respectively. The model with the lowest validation value was kept to proceed with the test step. Additionally to the accuracy and the confusion matrix, measures for Cohen's Kappa and F1 score are also provided to measure the performance of the model. Cohen's Kappa provides an estimation of the agreement between the algorithm and the technicians according to Eq. (5), excluding the chances of random agreement. To do that, the formula uses the observed agreement (p0) and the probabilities of a chance agreement (p_e).

$$\kappa = \frac{p_0 - p_e}{1 - p_e} \tag{5}$$

On its own, F1 score combines precision (or positive predictive value, PPV) and recall (or true positive rate, TPR) in a single measure according to the Eq. (6), where PPV represents the ratio between the true positives (TP) among all the cases labeled as positive by the model, while TPR represents The positive cases identified among the total number of positives in the ground truth

$$F_1 = \frac{PPV * TPR}{PPV + TPR} \tag{6}$$

3.1.4. Electroencephalograms

One of the very few and non-invasive techniques to study the brain is electroencephalography. The records produced by this test, called electroencephalograms, register the alterations produced by the brain with its electrical activity over time. Widely used in different clinical problems such as diagnosis of epilepsy [47], depth of anesthesia [48], or sleep disorders [35]. These signals are captured by using a set of electrodes placed on the patient's scalp following the standard 10/20 [49] and calculating the difference between the potential of two electrodes. Those electrodes are also the source of the names of the different recorded channels, for example, in this work channels, C4-A1 and C3-A2 are going to be used because those 4 nodes were the ones used when the dataset was recorded. One of the main issues of these records is the amount of noise in the recorded signals. Potential measured in the scalp has an amplitude between 10 V and 100 V, this potential has to be preamplified d by a factor of 1000 to 10,000 times in order to be registered by the instruments which result in signals with many artifact and false positives. This high-level noise with the non-stationary nature of these signals results in a particular complex analysis as can be seen in Fig. 2.

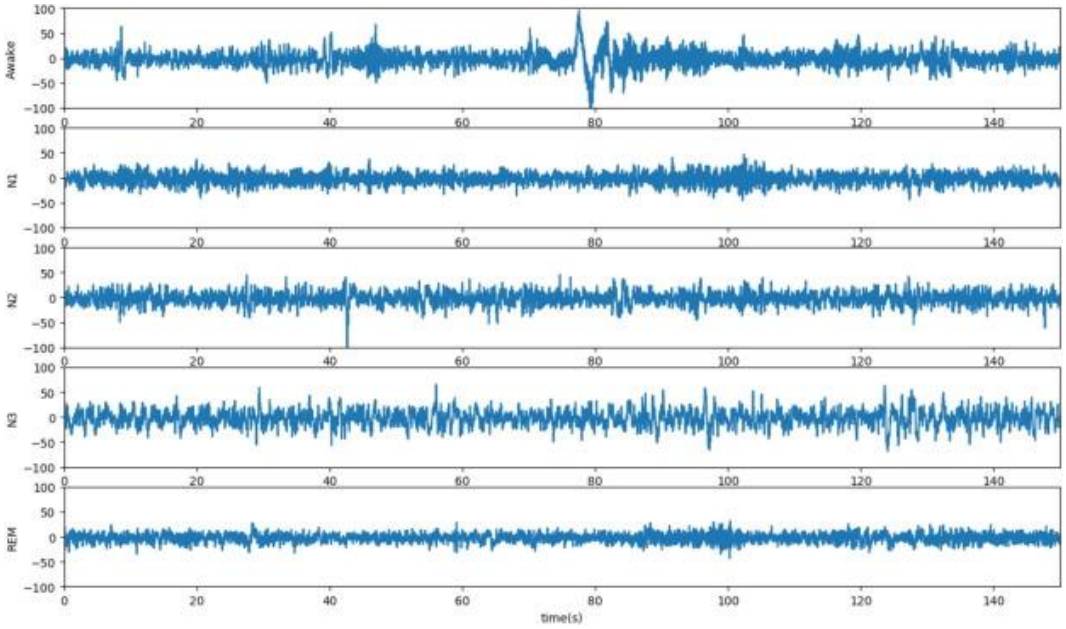

Fig. 2. Samples for each labeled stage of a EEG from SHHS-1 dataset.

3.1.5. Dataset

The multi-center cohort study known as Sleep Heart Health Study (SHHS) was carried out from 1995 until 2010. It was an initiative from the American National Heart Lung and Blood Institute whose objective was to determine the relationship between sleep-disorders and high-risk cardiovascular issues.

SHHS polysomnographic records contain two EEG channels (C4-A1 and C3-A2), two EOG channels, one EMG channel, one ECG channel, two inductance plethysmography channels (thoracic and abdominal), a position sensor, a light sensor, a pulse oxymeter, and an airflow sensor.

Additionally, SHHS [11] was divided into two different datasets: SHHS-1 and SHHS-2, corresponding to two different visits of the same patient cohort. However, in this work, which is focused on using the EEG channels, only the SHHS-1 was used due to the homogeneity of the records. In SHHS-1, every EEG channel was recorded at 125 Hz and manually scored by a single technician according to R&K scoring rules [5], while in SHHS-2 EEGs were recorded between 125 Hz and 128 Hz. Therefore, the dataset used contains records from 5804 patients which were labeled on 30s epochs in several sleep stages: Awake, S1, S2, S3, S4, REM and Unknown. The results are over 49Gbytes of data between the signals and the corresponding labels. It must be highlighted that the labeling was performed before AASM standard [6], consequently, an adaptation can be performed following the rules cited on the guidelines. The result was a label between Awake, N1, N2, N3, and REM for each epoch as it is shown in Table 1.

Table 1. Conversion between R&K and AASM guidelines.

Guideline	Sleeping Stages						
R&K	Awake	S1	S2	S3	S4	REM	Unknown
AASM	Awake	N1	N2	N3		REM	–

3.2. Proposed architecture

To solve the problem of sleep scoring described in Section 3.1.5, this work proposes an architecture mainly base on Depthwise Separable Convolutional layers. In this particular case, the resulting schema after several tests is shown in Fig. 3. As any Deep Learning model, the architecture presents two parts: first, the feature extractor and, second, the classifier.

As input, the system receives 1 or 2 EEG signals from the SHHS-1 corresponding to the C3-A2 and C4-A1 channels. Those signals, which were recorded at 125 Hz each, were labeled in intervals of 30 s, consequently, a label is available for every 3750 samples in the input signal, and they are generically called sections. The number of sections to be used in the input has been one of the components that this work has put under the spotlight performing many tests. More specifically, for each label, additionally, to the segment, tests were performed by including from 0 to 3 previous sections and from 0 to 2 latter ones.

With respect to the feature extraction, it is made by a succession of depthwise separable convolutional layers followed by *MaxPooling*. This particular combination of those 2 convolutional layers and the maxpooling layer is represented on Fig. 3 by blocks named *SeparatedConvolutionMP*, which are defined by the size of the kernel(K), the number of features(F) and the maxpooling size (M).

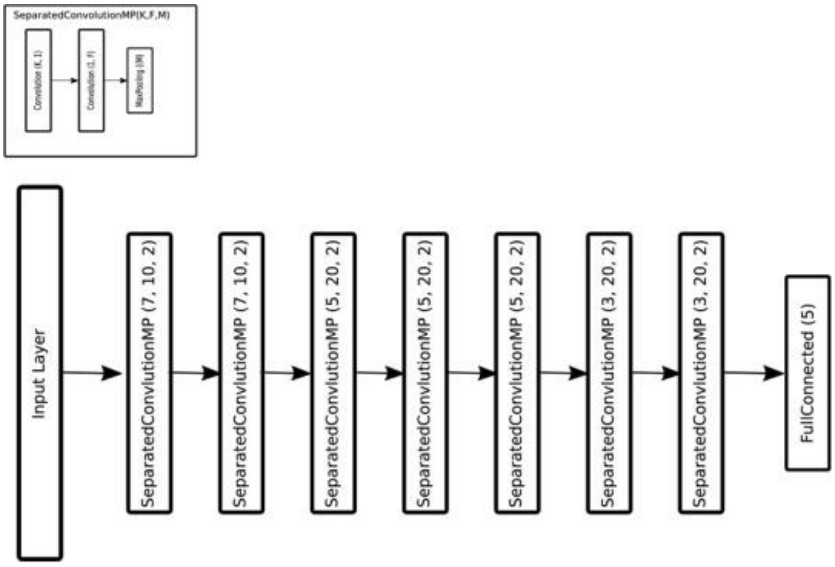

Fig. 3. Proposed architecture.

This particular combination of layers increases the pressure on the extracted features for three main reasons. First, each couple of layer which perform a separable convolution uses fewer parameters, as was already explained in Section 3.1.1, this makes that parameters more important to represent the solution space. Second, in this case, a reduction in the number of filters was also made, in the proposed architecture the separable convolutional layers extract between 10 to 20 filters oppositely to other works who extracted over 100. Finally, the last step to increase the pressure over the extracted features is the use of the *MaxPooling* layers reducing the number of features by half in each layer. Comparing the result of the last convolutional layer which is a set of 1700 inputs used as inputs for the classifier with the 37,500 data points in the input makes an idea of that pressure increment.

It is also worth to mention that Kernels sizes run from 7 to 3. The size of the kernels starts from the achievements of [50], which points to 7 to the best size to extract information from the brain, while the reduction and number of layers was the result of a set of exploratory tests, some of which are shown in Section 4.

4. Tests

Different architectures and configurations have been tested on the dataset described in Section 3.1.5. The main objective which has driven the tests has been to understand the influence of the number of parameters, weights and bias, and the depth on the network in the solution of the problem. Going along with this objective, the influence of the Deep-wise Separable Convolution to solve signal processing problems was also put under the spotlight. Dataset was split into 3 subsets for training, validation, and test. That split process was done according to the number of patients instead of the patterns in order to keep the 3 different datasets as much separated as possible and closer to a real scenario. The percent for the division were set as 70%, 10% and 20% for each one of the three subsets: training, validation, and test respectively. Once the data was defined, tests were carried out by following always the same pipeline. Once the architecture is defined, a training process is performed with mini-batches of 50 input patterns. After going through the whole training dataset, the validation dataset is evaluated. The training process will

stop after 100 iterations over the whole training set or after 10 iterations while the loss of the validation dataset has not improved. The result is the network with the lowest error on the validation test, which is evaluated with the test dataset. A summary of the sections contained on the 1161 patient used for testing is on Table 2. That table also contains the percentage of a particular subset with respect to the total.

Table 2. Sleeping Stages on test.

Awake	N1	N2	N3	REM	TOTAL
335,798	42,919	459,378	151,697	162,047	1,151,830
(29%)	(3%)	(40%)	(13%)	(15%)	

Finally, the classifier starts with a Dropout layer which is used during training with a probability of 0.5 in order to improve the generalization of the network. The rest of the classifier network is composed by a single fully-connected layer, that applies a Softmax function over the output. This output has been changed to a one-hot encoded in order to apply a categorical cross-entropy as the loss function. Finally, it may be appointed that the optimization is done with a gradient optimization known as ADAM [51].

Although, it is not the main objective of the paper, it may be highlighted that this is the first model which uses the whole SHHS-1 dataset without trimming data or removing any patient.

The results of these tests can be seen in Fig. 4. That figure represents the accuracy obtained by different architectures on respect to the number of parameters used on those architectures. Therefore, in this figure, each circle in the scatter plot represents a test with a different architecture, where the x position represents the number of parameters and the position on the y-axis represents the accuracy. Additionally, the size of the circles is proportional to the number of layers, while the colour represents the kind of convolutional process undertaken. Two kinds of convolutions are represented, with darker points the conventional convolution and lighter ones for the architectures with separable convolution. It should be highlighted the number of parameters has been represented in a logarithmic scale for a more convenient representation due to the differences between the networks, which run from 3,939 parameters to 884,915. Analyzing the figure, it is obvious that increasing the depth of the network is more profitable than increasing the number of parameters. The figure shows networks with 20 times fewer parameters return equivalent or better results. As it can be seen on the top left corner, the architectures with a better performance with less or equal number of parameters to train have always been the ones with the Depth-wise Separable Convolutions, which increased the number of layers while keeping the number of parameters under control as it can be seen by the size of the lighter dots.

Fig. 5 allows to take a closer look by only considering networks with less than 50,000 parameters. This figure shows the same information as Fig. 4, but changing the scale of the number of parameters to normal and removing any test with more than 50,000 parameters.

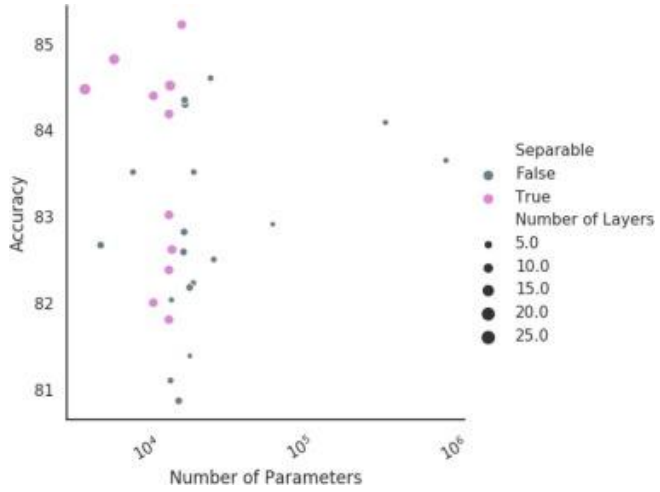

Fig. 4. Tests performance according to number of trainable parameters, depth and type of convolution.

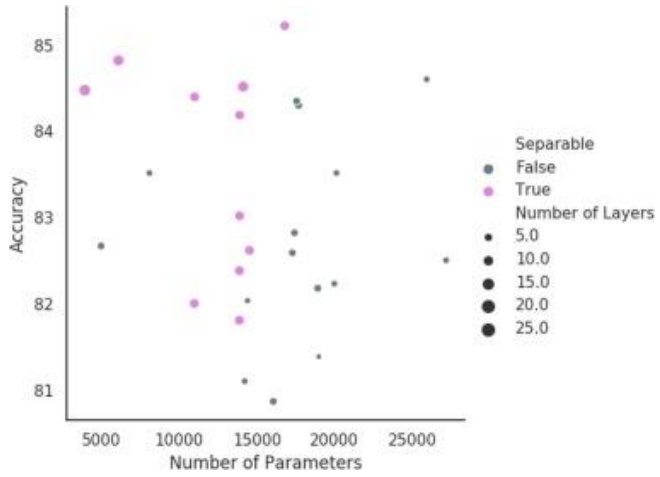

Fig. 5. Tests performance according to number of trainable parameters, depth and type of convolution for less than 50,000 parameters.

It is clear that with the same number of parameters the networks perform better with depth-wise separable convolution as every light dot is on the top-left corner of the figure. When the network has the same number of parameters it always performs better increasing the depth rather than increasing the width of the networks. As can be seen in the figure, if we fix the attention on a number of parameters and take a look from down to up, the size of the dots is constantly increasing. However, if we take a look from left to right with the exception of two points, to obtain a similar performance the number of parameters has to be drastically increased. This fact is associated with the increment on the pressure of the weights mentioned in Section 3.1.1. Oppositely to other works that have used this dataset, keeping a very small bunch of weights points out to make them more valuable and representative improving the results with a lower resource cost.

Additionally, if the number of signals is on the spotlight the results of the test are represented in Fig. 6. It shows the accuracy of different architectures according to the size of the input. The input is represented in the x-axis by tuples which are the number of previous and latter sections used. In those tuples, the first number is the number of previous 30s sections considered before the one to be classified as additional information. On the other hand, the second number represents the number of sections taken from after the section to be classified. Finally, the colour of the dots represents if there has been used one of the signals or both of them as input.

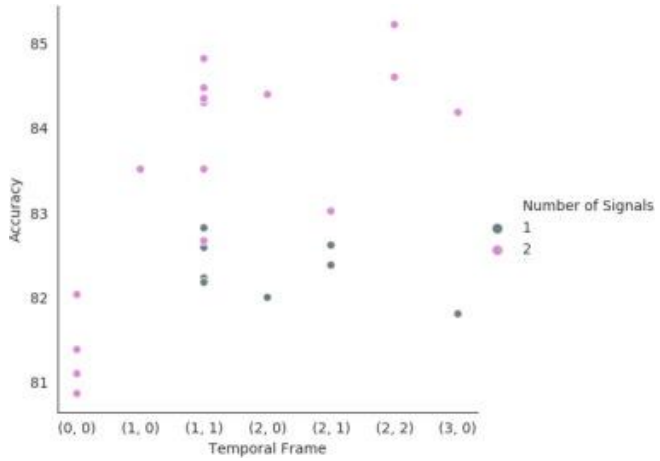

Fig. 6. Test according to the accuracy obtained on respect to temporal frame use.

In Fig. 6, the tests show a clear benefit of using both EEG signals at a time instead of only one of them. This fact comes to confirm something that was pointed out in [34] on a smaller dataset. Moreover, when the attention is focused on the temporal window uses as input, it is also clear that additional information, independently of the number of signals used, has a significant benefit in the performance. However, in the figure, it is also clear that there is a point where additional information has no effect, for example, tests performed with 3 previous sections and none of the latter show worse performance than test performed only with 2 previous sections and none of the later ones. Specific cases are shown in Table 3, there is a comparison of the proposed architecture but changing the input configuration of sections used before and after and the number of signals. Another point to highlight is the small difference in terms of the number of trainable parameters between the architectures using one or two signals.

All in all, the best results were obtained by using a network with a Depth-wise Separable convolution together with the two EEGs recorded in the PSG. The classification also showed an important improvement when the input contains additional information to the section to be classified. More specifically, the use of 2 previous sections and 2 latter ones showed the best general performance. Table 4 shows the confusion matrix of the test, which represent the results obtained by the network against the ground truth. A closer analysis of that matrix shows that the network has problems mainly on identifying the class N1being the only category under the 80% of precision and recall. This category has as mayor problem the underrepresentation in any dataset, for example, SHHS1 has only 3% of the total amount of sections labeled as N1 as can be seen in Table 1. In fact, many works join the states N1 and N2 in a single state labeled as ”swallow sleep” due to the fact that even the expert technicians have problems to separate one from the other. Even though this category bias the general performance measurements of the network, when the behaviour of the remaining classes is analyzed, the precision and recall are always between

82% and 92%. These measures are considered is acceptable due to the fact that among human experts a general coincidence of 84% has been observed [52].

Table 3. Comparison of different input configuration with the proposed architecture.

Sections Before	Sections After	Number of Signals	Accuracy	κ	F1 (macro)	Trainable Parameters
2	0	1	82.0%	0.74	0.72	10,982
2	0	2	84.4%	0.78	0.74	10,982
2	1	1	82.4%	0.75	0.72	13,882
2	1	2	83.0%	0.76	0.74	13,899
3	0	1	81.8%	0.74	0.71	13,882
3	0	2	84.2%	0.78	0.75	13,899

Table 4. Confusion matrix of the best network in test.

		Ground truth					
		Awake	N1	N2	N3	REM	Precision
Network	Awake	306,336	9,517	10,806	216	7,207	91.69%
	N1	3,520	9,476	2,951	2	1,598	54.00%
	N2	16,663	14,842	420,743	26,891	18,958	84.47%
	N3	1,269	13	23,097	124,530	61	83.59%
	REM	8,010	9,071	1,781	58	134,223	87.64%
	Recall	91.23%	22.08%	91.59%	82.09%	82.83%	

Table 5. Performance of the different works on SHHS-1 and their complexity

.	Number of Patients	Accuracy	*k*	F_1 (macro)	Trainable Parameters
[35]	5,728	87.00%	0.81	0.86	199,068,478
This work	5,804	85.22%	0.79	0.76	16,799

Overall, the general performance of the network can be seen in Table 5 together with another work that has used the same dataset. It should be highlighted that the presented work, oppositely to the one presented by [35], do not perform any additional filtering or clipping of the signals.

While the results are comparable with the already published architectures, it might be highlighted that the complexity of the one presented in this paper is definitely lower if the number of parameters to modify is taken into account. With 16,799 trainable parameters, the presented model could be trained in almost any desktop computer. More specifically, the tests shown in this paper have been executed in an Intel i7 2.6 GHz with 16Gbytes of RAM and the support of an

NVidia Titan X graphical card. Each training and test has taken been 2 and 6 days on the described computer. Something that may be highlighted is the fact that results from [35] have been impossible to reproduce because the removed patients and the patients used on the dataset were not published. Additionally, the previously published network requires an infrastructure that the authors can not afford.

5. Conclusions

As the main conclusion of this work, it can be stated that similar results to already published works can be obtained by improving the quality of the features extracted by the convolutional networks. Instead of networks with lots of low-quality weights, increasing the pressure on those weights to extract more meaningful features is an advantage. One of the benefits from the shrink down of the number of parameters is overfitting control. Fewer parameters mean that features can not memorize the patterns instead of learning the features of a group, as oppositely of what can happen in other works like [30], which has more than 500 million parameters for a fraction of those patterns. Another side effect of this shrink is an increase in the overall training process, due to fewer parameters mean less gradient modification to calculate and, therefore, less time to perform the training. Finally, going along with the aforementioned benefits, the contention of the number of parameters while increasing the depth allows to extract higher-level features with less resources. For example, in this work, Deep-wise separable convolution and pooling layers have been used to increase the pressure on the extracted features by the convolutional layers. By applying those two easy mechanisms to control the number of parameters, this work has shown similar results to the ones achieved by previously published ones although with a reduction of 1000 times less trainable parameters.

Another point that has confirmed some recently published works on small data sets like [47], [53], [34], is the fact that using several signals recorded simultaneously improves the results on the classification. As tests shown in Table 3, it is better to use more information in the input allowing the network to extract as much information as possible.

Finally, the last fact to be mention is the improvement in the results when not only the section to be classified is used but also temporally related information. As it is also shown in Fig. 6, the use of previous and latter sections as inputs of the network does not suppose a significative impact on the number of parameters but it really makes a difference in the performance of the network.

6. Futures works

This work opens several possibilities to continue its development, first, it is clear that the dataset has a problem with the class N1 which is clearly unrepresented compared with the remaining ones. To explore alternatives to deal with this issue should be on top of the list, for example, by weighting with higher values the samples of the lower representative classes during the training.

Second, the inclusion of other signals contained in the PSG like electromyograms, electrooculograms or electrocardiograms could improve the labeling by integrating information from different sources. Different alternatives are open at this point, especially due those signals have been sampled at different rates and its integration is not straightforward. Although some recent works such as [54] have shown its advantages on small datasets.

Third, it would be also interesting to check the behaviour of this architecture with other datasets. Usually, each dataset records different EEG channels. It would be interesting to check the behaviour of this architecture with those small datasets. Furthermore, this could be a good

opportunity to check if the transfer learning observed principles, observed in recently published works [55], [56], is applicable in the signal processing.

Finally, as it was aforementioned, the use of more than one signal simultaneously is a clear improvement for the architecture, however not always all the signals are available, for example with a stick-off pad. A set of tests in order to check the failure tolerance of the systems when one of the signals is misleading or simply does not exist.

Funding

This work has been partially funded by the Carlos III Health Institute and the European Regional Development Funds (FEDER) [PI17/01826]. It was also partially supported by different grants and projects from the Xunta de Galicia [ED431D 2017/23; ED431D 2017/16; ED431G/01; ED431C 2018/49].

CRediT authorship contribution statement

Enrique Fernandez-Blanco: Conceptualization, Methodology, Software, Validation, Formal analysis, Resources, Writing - original draft, Writing - review & editing, Visualization, Investigation. Daniel Rivero: Conceptualization, Formal analysis, Resources, Writing - review & editing, Project administration. Alejandro Pazos: Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors want to acknowledge the support from Nvidia, who has donated the GPU used in the experiments of this publication, and Centro de Supercomputación de Galicia (CESGA), who allows to conduct the first exploratory tests on their installations.

References

[1] World Health Organization, et al., Global recommendations on physical activity for health. 2010 (2015).

[2] S. Stranges, W. Tigbe, F.X. Gómez-Olivé, M. Thorogood, N.-B. Kandala, Sleep problems: an emerging global epidemic? findings from the indepth who-sage study among more than 40,000 older adults from 8 countries across africa and asia, Sleep 35 (8) (2012) 1173–1181.

[3] E.S. Ford, A.G. Wheaton, T.J. Cunningham, W.H. Giles, D.P. Chapman, J.B. Croft, Trends in outpatient visits for insomnia, sleep apnea, and prescriptions for sleep medications among us adults: findings from the national ambulatory medical care survey 1999–2010, Sleep 37 (8) (2014) 1283–1293.

[4] B. Boashash, S. Ouelha, Automatic signal abnormality detection using timefrequency features and machine learning: a newborn eeg seizure case study, Knowl.-Based Syst. 106 (2016) 38–50.

[5] A. Rechtschaffen, A. Kales, A manual of standardized terminology and scoring system for sleep stages of human subjects. Los Angeles: Brain information service, Brain Research Institute, University of California at Los Angeles..

[6] R.B. Berry, R. Budhiraja, D.J. Gottlieb, D. Gozal, C. Iber, V.K. Kapur, C.L. Marcus, R. Mehra, S. Parthasarathy, S.F. Quan, et al., Rules for scoring respiratory events in sleep: update of the 2007 aasm manual for the scoring of sleep and associated events, J. Clin. Sleep Med. 8 (05) (2012) 597–619.

[7] C.W. Whitney, D.J. Gottlieb, S. Redline, R.G. Norman, R.R. Dodge, E. Shahar, S. Surovec, F.J. Nieto, Reliability of scoring respiratory disturbance indices and sleep staging, Sleep 21 (7) (1998) 749–757.

[8] E. Fernandez-Blanco, D. Rivero, M. Gestal, C. Fernández-Lozano, N. Ezquerra, C. Robert Munteanu, J. Dorado, A hybrid evolutionary system for automated artificial neural networks generation and simplification in biomedical applications, Curr. Bioinform. 10 (5) (2015) 672–691.

[9] D. Wang, J. Chen, Supervised speech separation based on deep learning: an overview, IEEE/ACM Trans. Audio Speech Lang. Process. 26 (10) (2018) 1702–1726.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2018) 834–848.

[11] S.F. Quan, B.V. Howard, C. Iber, J.P. Kiley, F.J. Nieto, G.T. O'connor, D.M. Rapoport, S. Redline, J. Robbins, J.M. Samet, et al., The sleep heart health study: design, rationale, and methods, Sleep 20 (12) (1997) 1077–1085.

[12] A.T. Tzallas, M.G. Tsipouras, D.I. Fotiadis, A time-frequency based method for the detection of epileptic seizures in eeg recordings, in: Computer-Based Medical Systems, 2007. CBMS'07. Twentieth IEEE International Symposium on, IEEE, 2007, pp. 135–140..

[13] A.R. Hassan, A. Subasi, Automatic identification of epileptic seizures from eeg signals using linear programming boosting, Comput. Methods Programs Biomed. 136 (2016) 65–77.

[14] Z. Gao, X. Wang, Y. Yang, C. Mu, Q. Cai, W. Dang, S. Zuo, Eeg-based spatio–temporal convolutional neural network for driver fatigue evaluation, IEEE Trans. Neural Netw. Learn. Syst. 30 (9) (2019) 2755–2763.

[15] U.R. Acharya, S.L. Oh, Y. Hagiwara, J.H. Tan, H. Adeli, D.P. Subha, Automated eeg-based screening of depression using deep convolutional neural network, Comput. Methods Programs Biomed. 161 (2018) 103–113.

[16] X. Zhao, H. Zhang, G. Zhu, F. You, S. Kuang, L. Sun, A multi-branch 3d convolutional neural network for eeg-based motor imagery classification, IEEE Trans. Neural Syst. Rehabil. Eng. 27 (10) (2019) 2164–2177.

[17] V.J. Lawhern, A.J. Solon, N.R. Waytowich, S.M. Gordon, C.P. Hung, B.J. Lance, Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces, J. Neural Eng. 15(5)..

[18] C. Berthomier, X. Drouot, M. Herman-Stoïca, P. Berthomier, J. Prado, D. Bokar-Thire, O. Benoit, J. Mattout, M.-P. d'Ortho, Automatic analysis of single-channel sleep eeg: validation in healthy individuals, Sleep 30 (11) (2007) 1587–1595.

[19] S.-F. Liang, C.-E. Kuo, Y.-H. Hu, Y.-S. Cheng, A rule-based automatic sleep staging method, J. Neurosci. Methods 205 (1) (2012) 169–176.
[20] S.-F. Liang, C.-E. Kuo, Y.-H. Hu, Y.-H. Pan, Y.-H. Wang, et al., Automatic stage scoring of single-channel sleep eeg by using multiscale entropy and autoregressive models, IEEE Trans. Instrum. Meas. 61 (6) (2012) 1649–1657.
[21] L. Doroshenkov, V. Konyshev, S. Selishchev, Classification of human sleep stages based on eeg processing using hidden markov models, Biomed. Eng. 41 (1) (2007) 25–28.
[22] A.R. Hassan, M.I.H. Bhuiyan, Automated identification of sleep states from eeg signals by means of ensemble empirical mode decomposition and random under sampling boosting, Comput. Methods Programs Biomed. 140 (2017) 201–210.
[23] Y.-L. Hsu, Y.-T. Yang, J.-S. Wang, C.-Y. Hsu, Automatic sleep stage recurrent neural classifier using energy features of eeg signals, Neurocomputing 104 (2013) 105–114.
[24] A.R. Hassan, S.K. Bashar, M.I.H. Bhuiyan, On the classification of sleep states by means of statistical and spectral features from single channel electroencephalogram, in: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2015, pp. 2238–2243.
[25] M. Ronzhina, O. Janoušek, J. Kolárˇová, M. Nováková, P. Honzík, I. Provazník, Sleep scoring using artificial neural networks, Sleep Med. Rev. 16 (3) (2012) 251–263.
[26] G. Zhu, Y. Li, P.P. Wen, Analysis and classification of sleep stages based on difference visibility graphs from a single-channel eeg signal, REM 806 (2014) 803.
[27] A.R. Hassan, S.K. Bashar, M.I.H. Bhuiyan, Automatic classification of sleep stages from single-channel electroencephalogram, in: India Conference (INDICON), 2015 Annual IEEE, IEEE, 2015, pp. 1–6.
[28] G.E. Hinton, Deep belief networks, Scholarpedia 4 (5) (2009) 5947.
[29] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.
[30] O. Tsinalis, P.M. Matthews, Y. Guo, Automatic sleep stage scoring using timefrequency analysis and stacked sparse autoencoders, Ann. Biomed. Eng. 44 (5) (2016) 1587–1597.
[31] B. Kemp, A.H. Zwinderman, B. Tuk, H.A. Kamphuisen, J.J. Oberye, Analysis of a sleep-dependent neuronal feedback loop: the slow-wave microcontinuity of the eeg, IEEE Trans. Biomed. Eng. 47 (9) (2000) 1185–1194.
[32] A. Supratak, H. Dong, C. Wu, Y. Guo, Deepsleepnet: a model for automatic sleep stage scoring based on raw single-channel eeg, IEEE Trans. Neural Syst. Rehabil. Eng. 25 (11) (2017) 1998–2008.
[33] C. O'reilly, N. Gosselin, J. Carrier, T. Nielsen, Montreal archive of sleep studies: an open access resource for instrument benchmarking and exploratory research, J. Sleep Res. 23(6) (2014) 628–635..
[34] E. Fernandez-Blanco, D. Rivero, A. Pazos, Convolutional neural networks for sleep stage scoring on a two-channel eeg signal, Soft. Comput. 24 (2020) 4067–4079.
[35] A. Sors, S. Bonnet, S. Mirek, L. Vercueil, J.-F. Payen, A convolutional neural network for sleep stage scoring from raw single-channel eeg, Biomed. Signal Process. Control 42 (2018) 107–114.
[36] K. Fukushima, S. Miyake, Neocognitron: a self-organizing neural network model for a mechanism of visual pattern recognition, in: Competition and Cooperation in Neural Nets, Springer, 1982, pp. 267–285..
[37] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. (2012) 1097–1105.
[38] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521(7553) (2015) 436..
[39] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface closing the gap to humanlevel performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1701–1708.
[40] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, Int. J. Comput. Vision 115 (3) (2015) 211–252.
[41] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.N. Sainath, et al., Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[42] W. Chan, N. Jaitly, Q. Le, O. Vinyals, Listen, attend and spell: a neural network for large vocabulary conversational speech recognition, in: Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, IEEE, 2016, pp. 4960–4964..

[43] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2017, pp. 1800–1807..
[44] L. Vanneschi, M. Castelli, S. Silva, Measuring bloat, overfitting and functional complexity in genetic programming, in: Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, ACM, 2010, pp. 877–884.
[45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[46] M. Cogswell, F. Ahmed, R. Girshick, L. Zitnick, D. Batra, Reducing overfitting in deep networks by decorrelating representations, arXiv preprint arXiv:1511.06068..
[47] E. Fernández-Blanco, D. Rivero, M. Gestal, J. Dorado, Classification of signals by means of genetic programming, Soft. Comput. 17 (10) (2013) 1929–1937.
[48] M. Esmaeilpour, A.R.A. Mohammadi, Analyzing the eeg signals in order to estimate the depth of anesthesia using wavelet and fuzzy neural networks, IJIMAI 4 (2) (2016) 12–15.
[49] R. Oostenveld, P. Praamstra, The five percent electrode system for highresolution eeg and erp measurements, Clin. Neurophysiol. 112 (4) (2001) 713–719.
[50] S. Sakhavi, C. Guan, S. Yan, Learning temporal information for brain-computer interface using convolutional neural networks, IEEE Trans. Neural Netw. Learn. Syst. 29 (2018) 5619–5629.
[51] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, CoRR abs/1412.6980..
[52] R.G. Norman, I. Pal, C. Stewart, J.A. Walsleben, D.M. Rapoport, Interobserver agreement among sleep scorers from different centers in a large dataset, Sleep 23 (7) (2000) 901–908.
[53] S. Chambon, V. Thorey, P.J. Arnal, E. Mignot, A. Gramfort, A deep learning architecture to detect events in eeg signals during sleep, in: 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), IEEE, 2018, pp. 1–6..
[54] F. Andreotti, H. Phan, N. Cooray, C. Lo, M.T. Hu, M. De Vos, Multichannel sleep stage classification and transfer learning using convolutional neural networks, in: 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), IEEE, 2018, pp. 171–174..
[55] H.-C. Shin, H.R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, R.M. Summers, Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning, IEEE Trans. Med. Imag. 35 (5) (2016) 1285–1298.
[56] S.-A. Rebuffi, H. Bilen, A. Vedaldi, Efficient parametrization of multi-domain deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8119–8127.