Epilepsy-Seizure Detection using Convolutional Neural Networks

Fragoulis Logothetis and Vissarion Berthold Konidaris

Abstract— **Recent research suggests that electrophysiological changes develop minutes to hours before the actual clinical onset in focal epileptic seizures. Seizure prediction is a major field of neurological research, enabled by statistical analysis methods applied to features derived from intracranial Electroencephalographic (EEG) recordings of brain activity. However, no reliable seizure prediction method is ready for clinical applications. In this study, we use modern machine learning techniques to predict seizures from a set of spectrograms. Convolutional Neural Network is implemented on 6 patients suffering from medically intractable focal epilepsy. For each patient, the model predicts 100% of the seizures with no false alarm.**

Keywords– EEG, Seizure Detection, Epilepsy, Machine Vision, Deep Learning, CNN

I. INTRODUCTION

For individuals with drug-resistant epilepsy, responsive neurostimulation systems hold promise for augmenting current therapies and transforming epilepsy care. Of the more than two million Americans who suffer from recurrent, spontaneous epileptic seizures, 500,000 continue to experience seizures despite multiple attempts to control the seizures with medication. For these patients responsive neurostimulation represents a possible therapy capable of aborting seizures before they affect a patients normal activities.

In order for a responsive neurostimulation device to successfully stop seizures, a seizure must be detected and electrical stimulation applied as early as possible. A seizure that builds and generalizes beyond its area of origin will be very difficult to abort via neurostimulation. Current seizure detection algorithms in commercial responsive neurostimulation devices are tuned to be hypersensitive, and their high false positive rate results in unnecessary stimulation. In addition, physicians and researchers working in epilepsy must often review large quantities of continuous EEG data to identify seizures, which in some patients may be quite subtle. Automated algorithms to detect seizures in large EEG datasets with low false positive and false negative rates would greatly assist clinical care and basic research.

In this project we will work on datasets from patients with epilepsy undergoing intracranial EEG monitoring to identify a region of brain that can be resected to prevent future seizures are included in the contest. These datasets have varying numbers of electrodes and are sampled at 5000 Hz, with recorded voltages referenced to an electrode outside the brain. Since the number of electrodes and the recording conditions differ across patients, we have trained a different classifier for each patient. Other models and machine learning algorithms like Decision Trees, Random-Forest Classifiers and other well-known methods could be employed but on this letter we focus on CNN architectures and Neural Network as flatten layers.

II. INFORMATION ABOUT INPUT DATA

A. Electroencephalography (EEG)

Electroencephalography, or more commonly known as EEG, is an electrophysiological monitoring method to record the electrical activity of the brain. It is typically noninvasive, with the electrodes placed along the scalp, although invasive electrodes are sometimes used such as in electrocorticography. EEG measures voltage fluctuations resulting from ionic current within the neurons of the brain. In clinical contexts, EEG refers to the recording of the brain's spontaneous electrical activity over a period, as recorded from multiple electrodes . Diagnostic applications generally focus either on event-related potentials or on the spectral content of EEG. The former investigates potential fluctuations time locked to an event like stimulus onset or button press. The latter analyses the type of neural oscillations (popularly called "brain waves") that can be observed in EEG signals in the frequency domain.

B. From EEG to Spectrograms

It could be easily observed that a serious issue on these project is to extract the best features to recognize on-time a possible seizure. Plenty of features could be described and analyzed on such a work. Some interesting examples are fractal dimension, mobility, complexity, skewness, kurtosis, variance and frequency energy at following bands: delta (0.5-4Hz), theta (4-7Hz), alpha(7-14Hz), beta(14-30Hz), gamma(30-100H00z). Those features even though if they provide informative details for the EEG of a specific patient, they are not suitable as input in a CNN network.

Working on this issue, we have determined that the best input of a CNN network are Spectrograms. The last is a simple FFT of EEG signals and are three dimensional images. The first dimension (X-axes) is the duration of the EEG signal, the second dimension (Y-axes) is the Frequency domain and the third dimension of is the amplitude of its frequency in a specific second (pixel color intensity). The color of spectrogram gives special information of the frequency amplitude. In our dataset we have a great amount of those spectrograms originated from different pairs of EEG electrodes called channels. The size (pixels) of its

image is 128×128, so the spectrogram cube (Figure 2) is $128 \times 128 \times number\_of\_channels$. The frequency range is from 0 to 312.5 Hz and the time range is from 0 to 1 sec. Each spectrogram is tagged with labels, where the label called Interictal is a non-seizure image and label called Ictal is a seizure image.

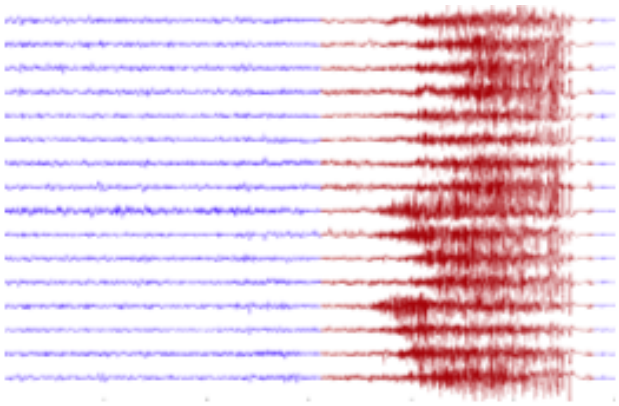

Fig. 1. EEG (μV)

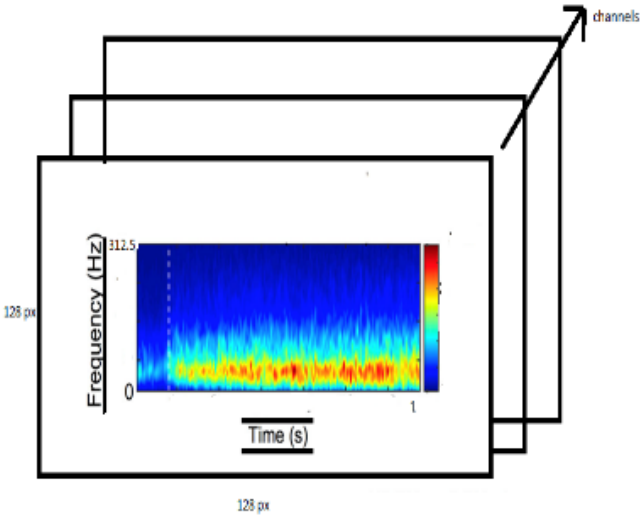

Fig. 2. Spectrograms

III. CNN NETWORK ARCHITECTURE

A. Convolutional Layers

- All convolutions use 3×3 filters with a specific padding value, so to keep intact the size of spectrogram.
- The input image has plenty of channels, depended on the number of electrodes used on the examination of a patient. The convolution layers progressively increases the number of output channels up to the value of 256.
- The max pooling layers have a size of 2×2 and stride value equal to 2. These layers will reduce both the height and the width of the layers input by 2.
- ELU non-linear activation function was employed on each convolutional layer.

B. Fully Connected Layers

- Flatten layer has 2 hidden layers with 256 and 512 neurons respectively.
- ELU non-linear activation function was employed on each layer except from the last one.
- The outputs of the topology are directly fed on a SoftMax Function to convert them to probabilities.
- Output is coded using one hot encoding. [0 1] vertex for Interictal situation and [1 0] for Ictal situation.

Finally, we used the batch normalization technique on all our layer, both on convolutional and fully connected ones, to reduce the training time.

IV. USED TECHIQUES FOR IMPLEMENTATION

A. Weighted SoftMax

The classic SoftMax model is the following :

$$Y = \frac{e^{x^T w}}{\sum_{k=1}^{K} e^{x^T w_k}} \qquad (1)$$

Where $x^w T$ is the output of a neuron after passing through the Elu activation function of the previous layer. This model works perfectly if the number of Interictal spectrograms are almost equal to Ictal spectrograms. But the dataset we worked on has a serious disadvantage. The number of Interictal spectrograms are nine point five times more, than the Ictal ones. In other words, 95% of our dataset accounts for Interictal spectrograms. Given that we are aware about that prior probabilities of the two classes, it could be easily extracted that a spectrogram is Interictal with prior probability 0.95 and is Ictal with prior probability 0.05. Using the above information a weighted SoftMax function was devised.

$$Y_{est} = \frac{e^{prior\_probability \cdot x^T w}}{\sum_{k=1}^{K} e^{prior\_probability \cdot x^T w_k}} \qquad (2)$$

B. Weighted Cross Entropy

The Cross Entropy(CE) loss is defined as:

$$Loss = ylog(Y_{est}) - (1-y)log(1-Y_{est}) \qquad (3)$$

where y is the target we want to reach and Yest is defined from Equation (2). Given that the weighted SoftMax function is weighted up by prior probabilities, the loss function and specifically the first member of Eq.3 needs to be scaled in order to minimize its really high value, as Yest value is really low on an ictal situation. To overcome this issue, the first member of Eq.3 needs to multiplied by a scaling factor.

$$Loss = -scale_{factor} \cdot ylog(Y_{est}) - (1-y)log(1-Y_{est}) \qquad (4)$$

In our study, let this scale factor be equal to 0.8. Fig.3 illustrates how loss function differs for Interictal and Ictal data for this specific scale factor.

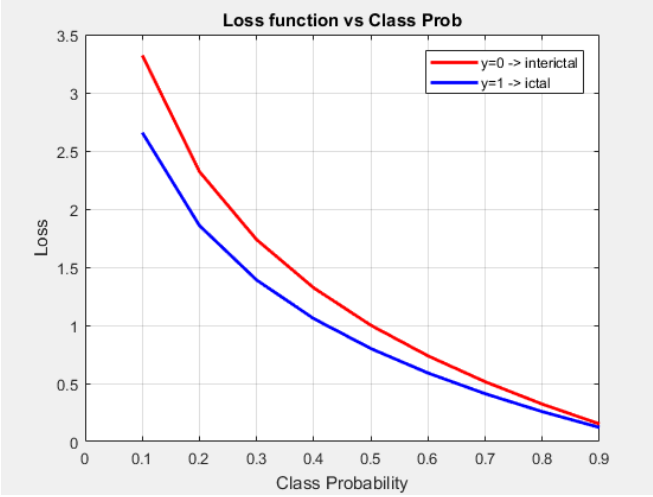

Fig. 3. Weighted Loss Function

C. Momentum and Adam Optimizer

Adam, is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. The method is straightforward to implement, is computationally efficient, has little memory requirements, is invariant to diagonal rescaling of the gradients and is well suited for problems that are large in terms of data and/or parameters. The method is also appropriate for non-stationary objectives and problems with very noisy and/or sparse gradients. Momentum (or SGD with momentum) is a method which helps accelerate gradients vectors on the right directions, thus leading to faster convergence. It is one of the most popular optimization algorithms and many state-of-the-art models are trained using it. In our project, we used both Adam and Momentum Optimizer and we compared their effectiveness.

D. Dropout Regularization

Dropout is a technique where randomly selected neurons or CNN filters are ignored during training. They are dropped-out randomly. This means that their contribution to the activation is temporally removed on the forward pass and any weight updates are not applied on the backward pass. In our work we have adopted the above technique both on Convolutional Layers and on Flatten Layer. Dropout could easily destabilize the training procedure if some of the input filters in the first convolutional layer are temporally removed. For this reason, the first Convolutional Layer includes all the main information about the source images, because by dropping out some of this filters could provide bad information through the next layers.

E. Image Pre-Processing

Since the data are from intra-cranial recordings there is no need for data cleaning and data preprocessing. Nevertheless, we passed the spectrograms though a logarithmic function before feeding them to the CNN in order to increase the differences of pixel intensities.

V. F1-SCORE AND TRAINING STOP

Except of the well known metrics that are used to measure the accuracy of a model, we additionally used the F1-Score. In statistical analysis of binary classification, the F1 score (also F-score or F-measure) is a measure of a tests accuracy. It considers both the precision p and the recall r of the test to compute the score: p is the number of correct positive results divided by the number of all positive results returned by the classifier, and r is the number of correct positive results divided by the number of all relevant samples (all samples that should have been identified as positive). The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

$$F_\beta = \frac{(1+\beta^2)\cdot \text{true positive}}{(1+\beta^2)\cdot \text{true positive} + \beta^2 \cdot \text{false negative} + \text{false positive}}.$$

Fig. 4. F1 Score

In our study, is equal to 1. After a great amount of experiments we noticed that it is more than important, to stop the training procedure at the correct moment. It is more than difficult to define the correct moment of stopping and it differs from problem to problem. In order for our network to achieve the best possible accuracy, we formulated our own stopping metric as:

$$Metric = w \cdot Loss\_On\_Validation\_Set$$
$$+(1-w)\cdot(1 - F1\_Score\_On\_Validation\_Set) \qquad (5)$$

where w takes values between 0 and 1. When this metric gets a minimum value and after N epochs this minimum remains unchanged, the training stops and the trained model

of N epochs before the stopping is selected as selected as the best one. The above procedure has a heuristic nature, but after a plethora of experiments it turned out to be the best solution to achieve the best accuracy on the test set. In the following experiments, Fig 11 illustrates the fluctuation of the above metric as the epochs pass and is depicted with a red dot at the moment-epoch that training phase was stopped.

VI. EXPERIMENTS

We implemented the above architecture on the Tensorflow Tool, which is developed by Google and it is currently one of the best tools for machine learning purposes. We now proceed by representing a series of experiments that we have conducted in order to reach to a conclusion for the best CNN architecture. All the experiments presented below are for Patient number 7. We initially split the data into two sets. The first set, comprising the 80% percent all the data, was used as a training set. The second set was divided again into two sets, a validation and a test set. All splits were done using the Stratified Splitting technique.

A. Pre-Processing of images

On this experiment we present the learning process for pre-processed and unprocessed input data. In our best model we feed the spectrograms to our CNN after passing them though a logarithmic function of base 10 (pre-processing). We then evaluate the learning against the same model but without any preprocessing on the input data. Figure 5 depicts the F1_Score that is achieved on the test set throughout all of the training epochs. By observing this plot, it becomes evident that the learning process is much more stable for unprocessed inputs, but the model with processed inputs achieves a significantly higher F1 score.

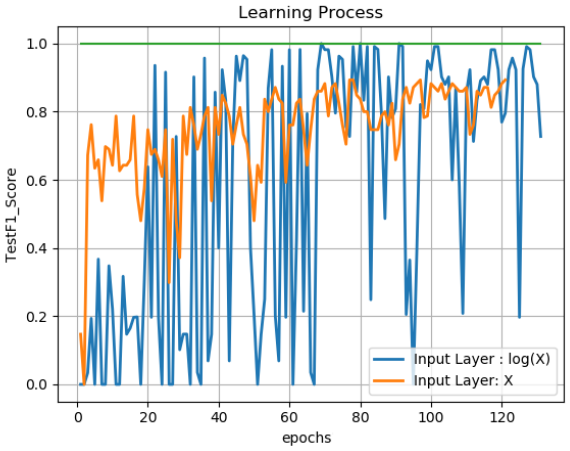

Fig. 5. F1-Score of the first experiment

B. Weighted Cross Entropy Loss vs Simple Cross Entropy

After our first experiment we proceed by examining the robustness of our modified cross entropy loss function against the original cross entropy loss. On Figures 6 and 7 we can see the decay of the training loss throughout training, as well as the increase of the F1 Score of the two models on the test dataset. It turned out that the learning was much faster with our weighted cross entropy loss, compared to the original one, as the training loss drops dramatically from the very first epochs. It takes almost 100 epochs for the model with the classic cross entropy cost function to drop the training loss to comparable levels with our technique, and even then it stacks on a plateau for another 100 epochs, without being able to achieve anything better. It can also be seen on the Figure 7 that the modified cross entropy loss, although unstable, reaches a higher F1 Score on unseen data. We believe that this is due to the fact that our modified CE is adapted specifically to this problem, as it takes into account the prior probabilities of the two classes.

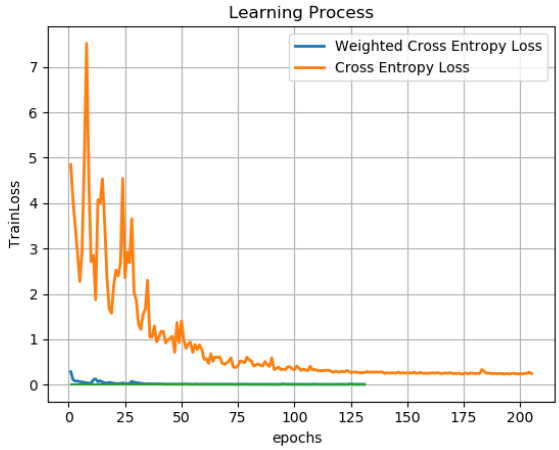

Fig. 6. Training Loss on the training set for the second experiment

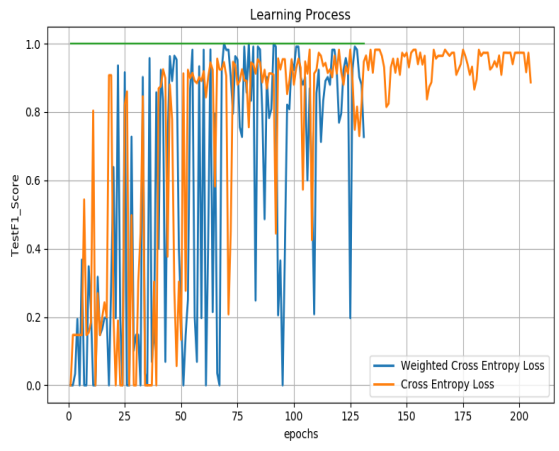

Fig. 7. F1-Score of the second experiment

C. With and Without Dropout

Another technique that seems to have made a difference in our model is the use of dropout in all of our convolutional

and original layers. The dropout technique, being a kind of an ensemble method, is widely adopted by the learning community for its regularization effects. This can be also seen in our experiments presented on the next two figures (Figures 8 and 9), as the model that is trained without dropout achieves lower loss on the training data, but performs worse on the test data than the model with dropout, proving that dropout makes our architecture to generalize better and avoid the dangerous pitfall of overfitting the training set.

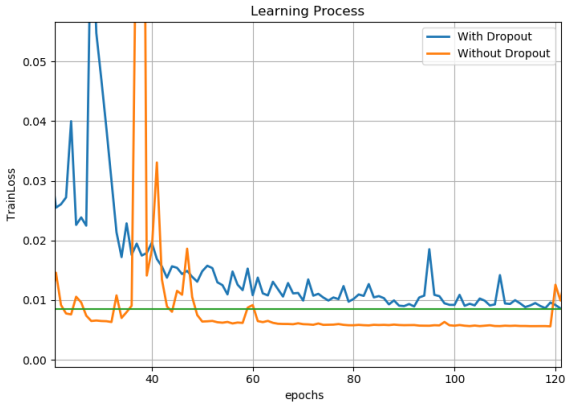

Fig. 8. Training Loss on the training set for the third experiment

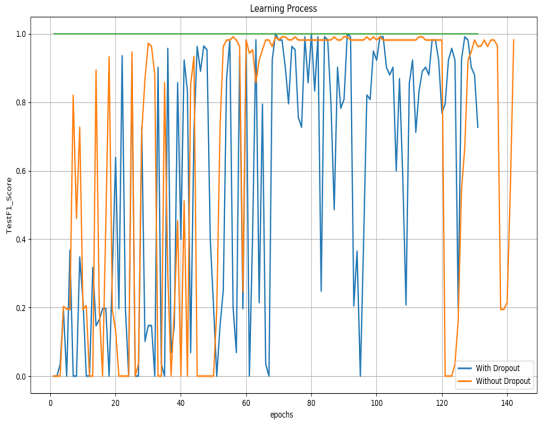

Fig. 9. F1-Score of the third experiment

D. Adam versus Momentum Optimizer

Our last experiment was based on comparing the optimization procedure of the Adam and Momentum optimizer on our architecture. For our best model we used the Momentum optimizer with exponential learning rate decay. The exponential decay step is performed each time we fit a batch to the model. On the other hand, the Adam optimizer has the property of adapting the learning rate for each individual learnable parameter on its own, a feat that makes it applicable in a wide range of problems. In Figure 10 we can observe the F1-Score on the test set during the training process for our model using the two optimizers. It turned out that the Momentum reduced the training time by more than fifty percent, while achieving much better F1 score than the Adam. This confirms the claim that simpler optimizers can perform better than more complex ones if they are properly adapted on the problem at hand.

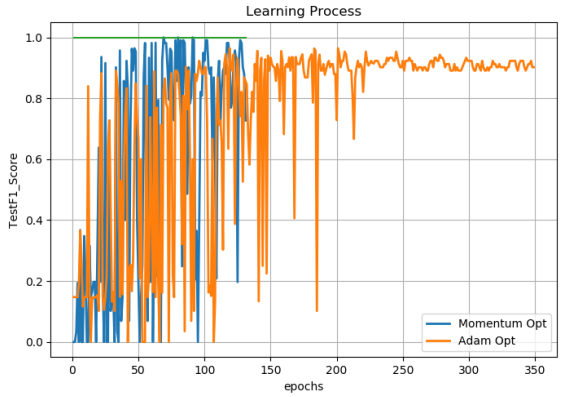

Fig. 10. Training Loss on the training set for the forth experiment

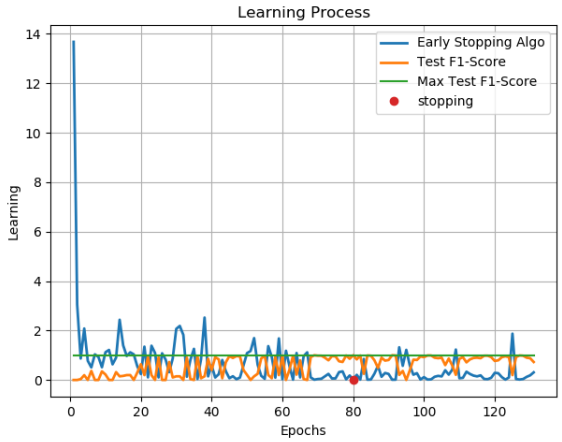

Fig. 11. Early stopping in training

E. Early Stopping in Training

Knowing when to stop the learning procedure is a very crucial on the field of deep learning, as it not only reduces the training time but also prevents the topology from overfitting the data. Our stopping technique utilized both the loss and the F1-Score of the validation set to ensure that at our model is close to optimal at stopping time. The situation depicted on the Figure 11 above is a good indication of this claim. It can be seen that when our stopping formula is minimized, which happens on the red dot (i.e. maximized F1-Score and

minimized loss on the validation set) the score on the test set is optimal.

VII. ACCURACY OF METHOD

The results of our method on the seven patients proved to be quite surprising. The quality of the intra-cranial recordings, combined with the power of CNN models and the techniques that were used, have gotten us some very promising results. All of the seven CNN models that where trained on the data of each patient managed to classify all the brain activities correctly, achieving maximum F1 Score on all of them.

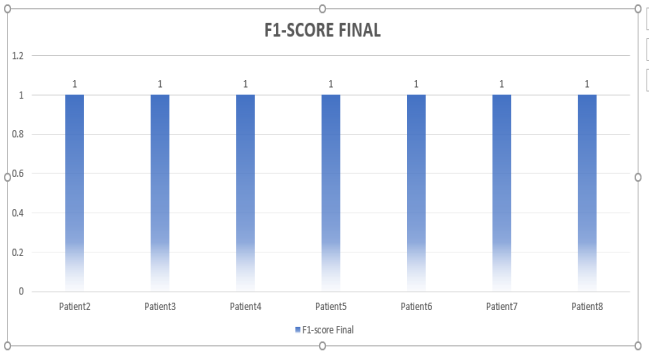

Fig. 12. F1-Scores of the CNNs

VIII. CONCLUSION

In conclusion, Convolutional Neural Networks seem to be not only a novel but also a very promising tool in bio-medical problems. It is interesting to see how the potential of these tools can be harvested to better understand all kinds of such severe diseases.

IX. FUTURE WORK

In a future work we hope to experiment more on this problem by combining appropriately the models of all patients using some interesting deep learning techniques like transfer learning.

REFERENCES

[1] A robust method for detecting interdependences: application to intracranially recorded EEG J. Arnhold ,, P. Grassberger , K. Lehnertz , C.E. Elger

[2] Practical method for determining the minimum embedding dimension of a scalar time series ,Department of Mathematics, University of Western Australia.

[3] Abdel-Hamid O, Mohamed A. r, Jiang H, Deng L, Penn G, Yu D (2014): Convolutional neural networks for speech recognition. IEEE/ACM Trans Audio Speech Lang Process 22:15331545.

[4] Ang KK, Chin ZY, Zhang H, Guan C (2008): Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface. In IEEE International Joint Conference on Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence), pp 23902397.

[5] Chatrian GE, Petersen MC, Lazarte JA (1959): The blocking of the rolandic wicket rhythm and some central changes related to movement. Electroencephalogr Clin Neurophysiol 11:497510.

[6] Chin ZY, Ang KK, Wang C, Guan C, Zhang H (2009): Multi-class lter bank common spatial pattern for four-class motor imagery BCI. In Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2009. EMBC 2009, pp 571574.

[7] Clevert D-A, Unterthiner T, Hochreiter S (2016): Fast and accurate deep network learning by exponential linear units (ELUs). ArXiv e-Prints volume 1511:page arXiv:1511.07289. son, Nonlinear resonant circuit devices (Patent style), U.S. Patent 3 624 12, July 16, 1990.

[8] Collinger JL, Wodlinger B, Downey JE, Wang W, Tyler-Kabara EC, Weber DJ, McMorland AJ, Velliste M, Boninger ML, Schwartz AB (2013): High-performance neuroprosthetic control by an individual with tetraplegia. Lancet 381:557564.

[9] Daly JJ, Wolpaw JR (2008): Braincomputer interfaces in neurological rehabilitation. Lancet Neurol 7:10321043.

[10] Das K, Giesbrecht B, Eckstein MP (2010): Predicting variations of perceptual performance across individuals from neural activity using pattern classiers. NeuroImage 51:14251437.

[11] Domhan T, Springenberg JT, Hutter F (2015): Speeding Up Automatic Hyperparameter Optimization of Deep Neural Networks by Extrapolation of Learning Curves. In Twenty-Fourth International Joint Conference on Articial Intelligence.

[12] Dosovitskiy A, Brox T (2016): Inverting Visual Representations with Convolutional Networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). arXiv:1506.02753.

[13] Duchi J, Hazan E, Singer Y (2011): Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res 12:(Jul):21212159.

[14] Gadhoumi K, Lina J-M, Mormann F, Gotman J (2016): Seizure prediction for therapeutic devices: A review. J Neurosci Methods 260:270282.

[15] Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V (2016): Domain-adversarial training of neural networks. J Mach Learn Res 17:135.

[16] Antoniades A, Spyrou L, Took CC, Sanei S (2016): Deep learning for epileptic intracranial EEG data. In 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP), pp 16.

[17] Bach S, Binder A, Montavon G, Klauschen F, Muller K-R, Samek W (2015): On pixel-wise explanations for non-linear classier decisions by layer-wise relevance propagation. PLoS ONE 10:e0130140.

[18] Ball T, Demandt E, Mutschler I, Neitzel E, Mehring C, Vogt K, Aertsen A, Schulze-Bonhage A (2008): Movement related activity in the high gamma range of the human EEG. NeuroImage 41:302310.

[19] Bashivan P, Rish I, Yeasin M, Codella N (2016): Learning Representations from EEG with Deep Recurrent-Convolutional Neural Networks. In arXiv:1511.06448 [cs]. arXiv: 1511.06448.

[20] Benjamini Y, Hochberg Y (1995): Controlling the false discovery rate: A practical and powerful approach to multiple testing. J R Stat Soc Ser B Methodol 57:289300.

[21] Blankertz B, Tomioka R, Lemm S, Kawanabe M, Muller K-R (2008): Optimizing spatial lters for robust EEG single-trial analysis. IEEE Signal Process Magaz 25:4156

[22] Brunner C, Leeb R, Muller-Putz G, Schlogl A, Pfurtscheller G (2008): BCI Competition 2008Graz Data Set A. Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology. pp 136142

[23] Canolty RT, Edwards E, Dalal SS, Soltani M, Nagarajan SS, Kirsch HE, Berger MS, Barbaro NM, Knight RT (2006): High gamma power is phase-locked to theta oscillations in human neocortex. Science 313:16261628.

[24] Cecotti H, Graser A (2011): Convolutional neural networks for P300 detection with application to brain-computer interfaces. IEEE Trans Pattern Anal Mach Intell 33:433445.

[25] Georgopoulos AP, Schwartz AB, Kettner RE (1986): Neuronal population coding of movement direction. Science 233:14161419.

[26] Giusti A, Ciresan DC, Masci J, Gambardella LM, Schmidhuber J (2013): Fast image scanning with deep max-pooling convolutional neural networks. In 2013 IEEE International Conference on Image Processing, pp 40344038.

[27] Goodfellow I, Bengio Y, Courville A (2016): Deep Learning. MIT Press. http://www.deeplearningbook.org

[28] Gratton G (1998): Dealing with artifacts: The EOG contamination of the event-related brain potential. Behav Res Methods Instrum Comput 30:4453.

[29] Hajinoroozi M, Mao Z, Jung T-P, Lin C-T, Huang Y (2016): EEG-

based prediction of drivers cognitive performance by deep convolutional neural network. Signal Process Image Commun 47:549555.

[30] Hammer J, Fischer J, Ruescher J, Schulze-Bonhage A, Aertsen A, Ball T (2013): The role of ECoG magnitude and phase in decoding position, velocity, and acceleration during continuous motor behavior. Front Neurosci 7:200.

[31] Haufe S, Meinecke F, Gorgen K, Dahne S, Haynes J-D, Blankertz B, Biemann F (2014): On the interpretation of weight vectors of linear models in multivariate neuroimaging. NeuroImage 87:96110.

[32] He K, Zhang X, Ren S, Sun J (2015): Deep residual learning for image recognition. arXiv 1512:03385 [cs] arXiv:1512.03385.

[33] Hertel L, Barth E, Kaster T, Martinetz T (2015): Deep convolutional neural networks as generic feature extractors. In 2015 International Joint Conference on Neural Networks (IJCNN), pp 14.

[34] Comparing SVM and Convolutional Networks for Epileptic Seizure Prediction from Intracranial EEG Piotr W. Mirowski, Member, IEEE, Yann LeCun, Deepak Madhavan, and Ruben Kuzniecky.M. Young, The Technical Writers Handbook. Mill Valley, CA: University Science, 1989.