EPILEPTIFORM SPIKE DETECTION VIA CONVOLUTIONAL NEURAL NETWORKS

Alexander Rosenberg Johansen[a,b], *Jing Jin*[b], *Tomasz Maszczyk*[b], *Justin Dauwels*[b], *Sydney S. Cash*[c], *M. Brandon Westover*[c]

[a] Technical University of Denmark, DTU Compute, Lyngby, Denmark

[b] Nanyang Technological University, School of Electrical and Electronic Engineering, Singapore

[c] Massachusetts General Hospital Neurology Department, and Harvard Medical School, USA

ABSTRACT

The EEG of epileptic patients often contains sharp waveforms called "spikes", occurring between seizures. Detecting such spikes is crucial for diagnosing epilepsy. In this paper, we develop a convolutional neural network (CNN) for detecting spikes in EEG of epileptic patients in an automated fashion. The CNN has a convolutional architecture with filters of various sizes applied to the input layer, leaky ReLUs as activation functions, and a sigmoid output layer. Balanced mini-batches were applied to handle the imbalance in the data set. Leave-one-patient-out cross-validation was carried out to test the CNN and benchmark models on EEG data of five epilepsy patients. We achieved 0.947 AUC for the CNN, while the best performing benchmark model, Support Vector Machines with Gaussian kernel, achieved an AUC of 0.912.

***Index Terms*—** Epilepsy, Spike detection, EEG, Deep learning, Convolutional neural network

1. INTRODUCTION

Epilepsy refers to a group of chronic brain disorders characterized by recurrent seizures, affecting approximately 65 million people worldwide [1]. Electroencephalography (EEG) is the primary diagnostic test for epilepsy, which provides a continuous measure of cortical function with excellent temporal resolution. Significant efforts are spent on interpreting EEG data for clinical purposes. In current clinical practice, visual inspection and manual annotation are still the gold standard for interpreting EEG, which is tedious and ultimately subjective. In addition, experienced electroencephalographers are in short supply [2]. As a result, a great need exists for automated systems for EEG interpretation.

The finding of primary importance for the diagnosis for epilepsy is the presence of interictal discharges, also known as "spikes" and "sharp waves", hereafter referred to collectively simply as "spike(s)". Automated spike detection would enable wider availability of EEG diagnostics and more rapid referral to qualified physicians who can provide further medical investigation and interventions. However, spikes are difficult to detect in a consistent manner due to the large variability of spike waveforms between patients among other factors [3]. Great attempts have been made to detect spikes by general classifications such as mimetic, linear predictive, and template based methods [4]. Many recent spike detection algorithms combine multiple methodologies, such as local context [5, 6, 7], morphology [8, 6, 7, 9, 10, 11], field of spike [8, 7, 12, 11, 13], artifact rejection [8, 6, 7], and temporal and spatial contexts [8, 13, 14]. Unfortunately, none of them are universally accepted or tested on a significantly large dataset of patients and spikes. To date, no algorithmic approach has overcome these challenges to yield expert-level detection of spikes [3].

In this study, we analyze the scalp EEG recordings of five patients diagnosed with epilepsy. Suspected interictal epileptiform spikes were cross-annotated by two neurologists. We applied convolutional neural networks (CNNs) to learn the discriminative features of spikes. CNNs are statistical models incorporating prior knowledge about the discriminative features of spikes. CNNs are commonly applied for finding local pixel dependencies [15]. Furthermore, CNNs have been proven successful in surpassing human accuracy in image recognition tasks [16], time-series tasks for text analysis [17] and biological sequences [18]. In addition, CNNs possess much fewer connections compared to a fully-connected neural network, due to the sparsity and the parameter sharing across the filters. Nevertheless, the computational complexity remains a key challenge to implementing CNNs. To address this issue, we utilize Graphical Processing Units (GPUs) and high performance libraries for modeling CNNs.

We benchmarked the CNN approach with several standard classification methods. Leave-one-patient-out cross-validation was conducted to generate the receiver operating characteristic (ROC) curve for each model, with the average area-under-the-curve (AUC) as the benchmark criterion. We achieved 0.947 AUC for the CNN, while the best performing benchmark model, Support Vector Machines with Gaussian kernel, achieved an AUC of 0.912.

This paper is organized as follows. In Section 2, we provide information on the EEG data considered in this study.

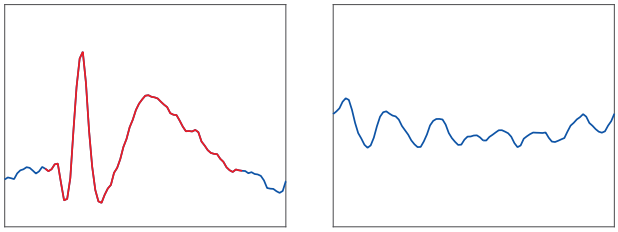
We also elaborate on the design of the CNN model. In Section 3, we provide results for the CNN approach and four standard classifiers. In Section 4, we offer concluding remarks and ideas for future research.

2. METHODS

2.1. Epileptiform EEG

We analyzed the scalp EEG recordings of five patients with known epilepsy. Each EEG data lasts 30 minutes recorded from 19 standard 10-20 scalp electrodes. The sampling rate is 128Hz. A high-pass filter at 1Hz was applied to remove the baseline drifts. A notch filter centered at 60Hz was applied to remove the power line interference. The common average referencing montage was applied to remove the common EKG artifacts.

As shown in Fig. 1a, interictal epileptiform spikes are morphologically defined events with an outstanding sharp peak distinguishable from the background fluctuations (see Fig.1b). Spikes were cross-annotated independently by two neurologists, and extracted with a fixed duration of 0.5 s. In order to reduce the computational load, we randomly sampled 1,500 spikes and 150,000 background waveforms from each subject. Leave-one-patient-out cross-validation was conducted to evaluate the CNN model. Moreover, 10% of the training data was further extracted under stratified sampling for validation, in order to tune the CNN model.



(a) (b)

Fig. 1: Illustration of (a) an interictal epileptiform spike, and (b) a background waveform.

2.2. Convolutional Neural Networks

The problem of spike detection is typically ill-posed, i.e., there are many models which fit the training patients or spikes well but do not generalize well. In other words, there is often not sufficient training data to allow accurate estimation of class probabilities throughout the input space. Convolutional neural networks (CNNs) [19] are suitable for such scenario, since they incorporate constraints and can achieve some degree of shift and deformation invariance. The architecture of a CNN (see Fig. 2) typically contains multiple layers: convolutional layers, dense layers, and an output layer. We applied the backpropagation algorithm to train the CNN. In this way, we can extract both low-level and high-level features from the input layer.

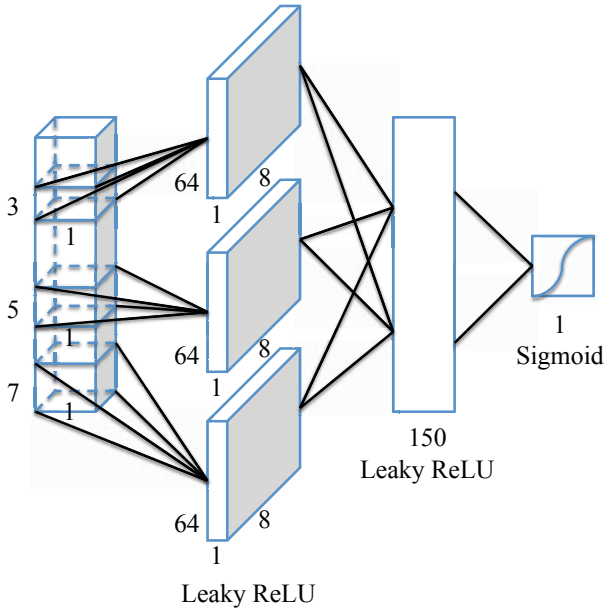


Fig. 2: The architecture of a the CNN model applied.

2.2.1. *Activation function*

In computational networks, the activation function defines the output of a layer given an input or a set of inputs. Nonlinear activation functions are commonly used in neural networks to improve the learning capacity and system robustness for nontrivial problems. Let us denote the output of layer ℓ as h_ℓ (with h_0 the input layer), $\Theta_{\ell+1}$ as the weight matrix of layer $\ell+1$, and $z_{\ell+1}$ as the linear combination of the weighted input to each neuron. The output of layer $\ell+1$ is computed as follows:

$$z_{\ell+1} = h_\ell \Theta_{\ell+1},$$
$$h_{\ell+1} = a(z_{\ell+1}). \quad (1)$$

The activation function is denoted by $a(z)$. The most commonly used nonlinear activation functions in neural networks are the Logistic Sigmoid $a(z) = 1/(1+e^{-z})$ and the Hyperbolic Tangent $a(z) = (e^z - e^{-z})/(e^z + e^{-z})$. Non-saturating activation functions such as the Rectifier Linear Unit (ReLU) with $a(z) = \max(0,\ z)$ have become more popular for their non-vanishing gradients and computational efficiency [15]. However, the ReLU also suffers from "dead" gradients, i.e. from large gradients that may deactivate the neurons, and further disable the network. In order to avoid

such problems, we applied the Leaky ReLU [16] as our activation function in CNN, which is defined as:

$$a(z) = \max(\alpha z,\ z), \tag{2}$$

with $\alpha = 0.1$. "Leaky" refers to the additional slope when $a(z) < 0$.

2.2.2. The convolutional architecture

In 1D Convolutional architectures, the processing steps of (1) are modified as follows:

$$z^{id}_{\ell+1} = x^i_\ell \Theta^d_{\ell+1},$$

$$h^{id}_{\ell+1} = a(z^{id}_{\ell+1}), \tag{3}$$

where the input $x^i_\ell \in \mathbb{R}^{kc\times 1}$ represents a co-located vector of length k in c channels and i refers to the specific co-located vector. The weight matrix $\Theta^d_\ell \in \mathbb{R}^{1\times kc}$ corresponds to a spatial weight filter with kc connections, in the d^{th} output channel, between the input layer and each neuron in the output layer. As a result, convolutional layers are configurable to their filter size k and number of filters d [15], leading to many possible configurations of the CNN architecture.

Multiple convolutional layers can help to reduce the parameter space, and to model non-linear mappings [20]. Moreover, combining different filter sizes can improve the overall performance [18]. Therefore, in this paper, we did not only test a single convolutional layer, but also stacked different convolutional layers on top of each other, and tested convolutional layers with different filter sizes. Based on our results, the implementation of multiple convolutional layers with different filter sizes provided the best validation performance, and thus was used to build the final CNN model.

2.2.3. The final CNN model

As depicted in Fig. 2, the CNN model contains five layers. The first three layers are convolutional layers with different filter sizes. The fourth layer is a fully-connected layer, which feeds into a fifth binary logistic layer (for normalization purpose) to make final decisions on whether an input waveform is a spike or not. During the training process, the CNN network is optimized by minimizing the binomial cross-entropy on the basis of the probability output from the neural network as:

$$L(x, y) = -f(x)\log(y) - (1-y)\log(1-f(x)), \tag{4}$$

where $f(x)$ is the prediction of the neural network given x.

After being activated by inputs, the convolutional layers are merged and passes the inputs to the neurons of the next fully-connected layer. The output of the fully-connected layer is then passed to the logistic output function. The leaky ReLU nonlinear activation function is applied after each convolutional layer as well as the fully-connected layer. The three convolutional layers in the CNN model contains one-dimensional filter of size 3, 5 and 7 respectively, and a stride of 1 with 8 filters each. The fully-connected layer has 150 hidden neurons. In order to avoid overfitting, we are using dropouts [21] with $p = 0.5$ on the fully-connected layer.

The data is extremely skewed (or "imbalanced"), since there are vastly more background waveforms than spikes. Therefore, the CNN may mostly model the background waveforms, instead of the spikes. This problem is addressed by training the CNN by means of balanced mini-batches.

Our CNN model is relatively small in size (220k parameters) compared to conventional CNN models for image processing [15]. Due to the small dataset used in this study, we limited the number of parameters in the CNN model to avoid overfitting while keeping the model simple. For future work, data augmentation [15], such as shifting the spikes by a few data points, may alleviate the problem of overfitting by generating synthetic data.

2.2.4. Details of learning

Our CNN model was trained with stochastic gradient descent using a batch-size of 4096 (2048 spikes and 2048 background waveforms). The update rule for the weight matrix Θ involves the Nesterov momentum [22] as follows:

$$v_{t+1} = \mu v_t - \epsilon \Delta f(\Theta_t + \mu v_t),$$

$$\Theta_{t+1} = \Theta + v_{t+1}, \tag{5}$$

with $\epsilon > 0$ the learning rate, $\mu \in [0, 1]$ the momentum coefficient, and $\Delta f(\Theta_t + \mu v_t)$ the gradient. In this study, the learning rate ϵ was set by grid searching from 10^{-3} to 10^{-5}, where $2\,10^{-4}$ gave the best convergence in the training data. The momentum applied was set to $\mu = 0.9$ and not optimized further on.

The CNN network was trained for a maximum of 50 epochs. The training was stopped when the validation error and training error diverged from one another. Leave-one-patient-out crosstesting was carried out to test the CNN model. The training was performed on a Tesla K40 GPU. We applied the Python built Theano library [23, 24] to compile to CUDA (a GPU interpretable language). We made use of the Lasagne library [25] to build and configure our CNN model.

3. RESULTS

We carried out a benchmark experiment to compare CNN against a variety of classifiers such as Suppport Vector Machines (SVM) with Gaussian kernel [26], Random Forest (RF) [27], k-Nearest Neighbor (KNN) [28], and C4.5

Decision Tree (DT) [29]. The receiver operating characteristic (ROC) curve was computed for each model, and the area under the ROC curve (AUC) was calculated as benchmark criterion. The larger the AUC, the better the performance. The AUC values are listed in Table 1.

Models	AUC
CNN	0.947
SVM	0.912
RF	0.883
KNN	0.835
DT	0.817

Table 1: AUC for the CNN model and standard classifiers.

Our numerical results show that the CNN achieves the largest AUC of 0.947, which outperforms the other four classifiers. As illustrated in Fig. 3, the ROC curves show that the CNN has favorable specificity and sensitivity compared to the other classifiers. However, due to the small dataset of only 5 patients, AUC scores varied a lot among different patients. Investigation with a much larger dataset is necessary to further explore the capabilities of CNNs in detecting epileptiform spikes from EEG signals.

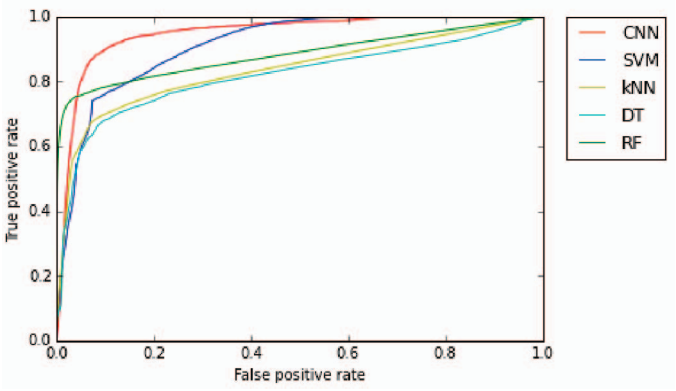


Fig. 3: ROC curves for various statistical models.

4. CONCLUSION

In this paper, we develop a CNN model for detecting spikes in the EEG of epileptic patients. Our numerical results for a small pool of 5 patients show that the CNN performs better than four standard classifiers. The CNN model was relatively small compared to previous contributions in the field of convolutional neural networks, since we limited the number of parameters in order to avoid overfitting. In future work, we will consider datasets of hundreds of epilepsy patients, and will train larger CNN models. Large-scale CNN models may yield even better detection results.

5. REFERENCES

[1] Epilepsy Foundation of America, "About Epilepsy: The Basics," *Epilepsy Foundation, http://www.epilepsy.com/learn/about-epilepsy-basics*, 2014.

[2] Brad A Racette, David M Holtzman, Timothy M Dall, and Oksana Drogan, "Supply and demand analysis of the current and future us neurology workforce," *Neurology*, vol. 82, no. 24, pp. 2254–2255, 2014.

[3] Jing Jin, Justin Dauwels, Sydney Cash, and M Brandon Westover, "Spikegui: Software for rapid interictal discharge annotation via template matching and online machine learning," in *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*. IEEE, 2014, pp. 4435–4438.

[4] Scott B Wilson and Ronald Emerson, "Spike detection: a review and comparison of algorithms," *Clinical Neurophysiology*, vol. 113, no. 12, pp. 1873–1881, 2002.

[5] Jean Gotman and Li-Yan Wang, "State dependent spike detection: validation," *Electroencephalography and clinical neurophysiology*, vol. 83, no. 1, pp. 12–18, 1992.

[6] Pedro Guedes De Oliveira, Carlos Queiroz, and Fernando Lopes Da Silva, "Spike detection based on a pattern recognition approach using a microcomputer," *Electroencephalography and clinical neurophysiology*, vol. 56, no. 1, pp. 97–103, 1983.

[7] Scott B Wilson, Christine A Turner, Ronald G Emerson, and Mark L Scheuer, "Spike detection ii: automatic, perception-based detection and clustering," *Clinical neurophysiology*, vol. 110, no. 3, pp. 404–411, 1999.

[8] J Gotman and LY Wang, "State-dependent spike detection: concepts and preliminary results," *Electroencephalography and clinical neurophysiology*, vol. 79, no. 1, pp. 11–19, 1991.

[9] Claudie Faure, "Attributed strings for recognition of epileptic transients in EEG," *International journal of bio-medical computing*, vol. 16, no. 3, pp. 217–229, 1985.

[10] BLK Davey, WR Fright, GJ Carroll, and RD Jones, "Expert system approach to detection of epileptiform activity in the EEG," *Medical and Biological Engineering and Computing*, vol. 27, no. 4, pp. 365–370, 1989.

[11] WRS Webber, Brian Litt, K Wilson, and RP Lesser, "Practical detection of epileptiform discharges (EDs) in

the EEG using an artificial neural network: a comparison of raw and parameterized EEG data," *Electroencephalography and clinical Neurophysiology*, vol. 91, no. 3, pp. 194–204, 1994.

[12] Andrew J Gabor and Masud Seyal, "Automated interictal EEG spike detection using artificial neural networks," *Electroencephalography and clinical Neurophysiology*, vol. 83, no. 5, pp. 271–280, 1992.

[13] Bhuvana Ramabhadran, James D Frost Jr, John R Glover, and Periklis Y Ktonas, "An automated system for epileptogenic focus localization in the electroencephalogram," *Journal of clinical neurophysiology*, vol. 16, no. 1, pp. 59–68, 1999.

[14] Michael A Black, Richard D Jones, Grant J Carroll, Alison A Dingle, Ivan M Donaldson, and Philip J Parkin, "Real-time detection of epileptiform activity in the EEG: a blinded clinical trial," *Clinical EEG and Neuroscience*, vol. 31, no. 3, pp. 122–130, 2000.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, Eds., pp. 1097–1105. Curran Associates, Inc., 2012.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *CoRR*, vol. abs/1502.01852, 2015.

[17] Xiang Zhang and Yann LeCun, "Text understanding from scratch," *CoRR*, vol. abs/1502.01710, 2015.

[18] Søren Kaae Sønderby, Casper Kaae Sønderby, Henrik Nielsen, and Ole Winther, "Convolutional lstm networks for subcellular localization of proteins," *arXiv preprint arXiv:1503.01919*, 2015.

[19] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back, "Face recognition: A convolutional neural-network approach," *Neural Networks, IEEE Transactions on*, vol. 8, no. 1, pp. 98–113, 1997.

[20] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[21] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[22] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, Sanjoy Dasgupta and David Mcallester, Eds. May 2013, vol. 28, pp. 1139–1147, JMLR Workshop and Conference Proceedings.

[23] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio, "Theano: a CPU and GPU math expression compiler," in *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010, Oral Presentation.

[24] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio, "Theano: new features and speed improvements," Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop, 2012.

[25] Sander Dieleman, Jan Schlter, Colin Raffel, Eben Olson, Sren Kaae Snderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, Jeffrey De Fauw, Michael Heilman, diogo149, Brian McFee, Hendrik Weideman, takacsg84, peterderivaz, Jon, instagibbs, Dr. Kashif Rasul, CongLiu, Britefury, and Jonas Degrave, "Lasagne: First release.," Aug. 2015.

[26] Marti A. Hearst, Susan T Dumais, Edgar Osman, John Platt, and Bernhard Scholkopf, "Support vector machines," *Intelligent Systems and their Applications, IEEE*, vol. 13, no. 4, pp. 18–28, 1998.

[27] Vladimir Svetnik, Andy Liaw, Christopher Tong, J Christopher Culberson, Robert P Sheridan, and Bradley P Feuston, "Random forest: a classification and regression tool for compound classification and qsar modeling," *Journal of chemical information and computer sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.

[28] Leif E Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, no. 2, pp. 1883, 2009.

[29] J. Ross Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.