**COLLEGE OF ENGINEERING**

Preliminary Exam Report

**Deep Learning Approaches to Automate Seizure Detection**

Vinit Shah

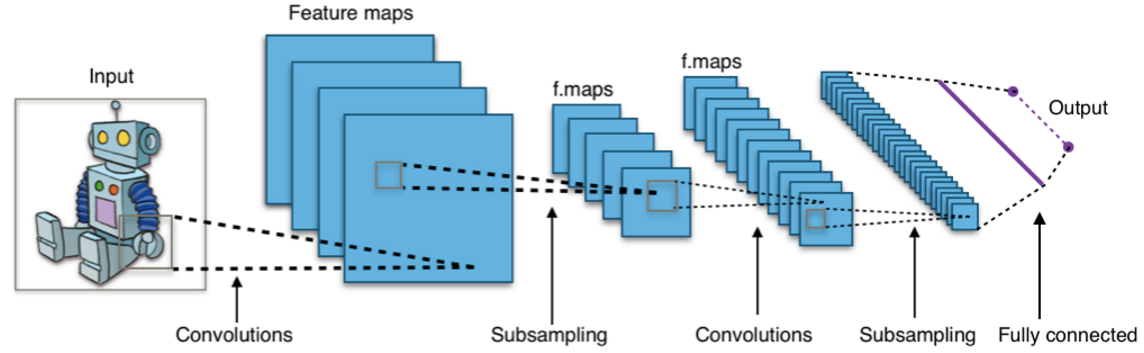
[vinitshah@temple.edu](mailto:vinitshah@temple.edu)

Institute for Signal and Information Processing,

Department of Electrical Engineering,

Temple University

April 2017



**Executive Summary**

Identifying seizures from electroencephalography (EEG) signals is a very time consuming task for neurologists. In addition, this task heavily relies in the judgement of the observer, in many cases yielding a low interrater agreement. The advent of technology has enabled hospitals to collect more Long-Term Monitoring (LTM) EEGs, which are widely used studies for the diagnosis and localization of epilepsy. To speed up the LTM interpretation process, QEEG tools have been developed. QEEG tools save time by presenting a gist of an EEG record on small sliding windows. QEEG tools are discussed in brief in this report, providing information from one of the assigned papers “Sensitivity of quantitative EEG for seizure identification in the intensive care unit”. The identification of seizures using QEEG tools is prone to missing seizures too (Haider et al, 2016). A brief description about an automatic seizure detection algorithm’s performance of one of the leading technologies in the field (Persyst Inc.) is provided, and the performance is analyzed. This picture presents a motivation for the development of new machine learning algorithms that perform seizure detection.

One of the papers discussed in this report called “Multi-task seizure detection: addressing intra-patient variation in seizure morphologies” discusses direct attempts to develop an automatic seizure detection system using Support Vector Machines (SVMs) by training and evaluating their system on the publicly available seizure dataset, CHB-MIT. The system is trained and evaluated on 23 subjects to measure False Positive Rates (FPR) when sensitivity is 100% in the Area Under the Curve. The generated results are compared with a standard SVM approach. Here, 15 out of 23 cases were showing an improvement in FPR greater than 10%. 6 of the remaining cases are showing a worsening in FPR of more than 10%. The median overall improvement in FPR is 27% with the proposed approach. Certain heuristic approaches such as no-trigger zone, variation in latency, etc. are taken in to consideration during these experiments.

During this decade, the deep learning field has become very popular. One of the most remarkable success cases in the area of deep learning is given by the task of image recognition, which has been greatly improved through the implementation of Convolutional Neural Networks (CNNs). One of the variants of CNNs, called Doubly Convolutional Neural Networks (DCNNs) is discussed in the third selected paper. These networks differ from regular CNNs in that an operation that checks the correlations of the adjacent meta-filters is added for every layer. The experiments were performed on three image classification benchmarks: CIFAR-10, CIFAR-100 and ImageNet. On all these datasets, with relatively smaller number of parameters DCNN outperforms CNN and CNN-variants by reducing error by 0.98% up to 3.15% without data augmentation and with data augmentation it is reducing error by 2.35% up to 6.51%. Also, the doubly convolutional layer is able to consistently improve the performance over the standard CNN regardless of the depth where it is plugged in.

The papers emphasize the need of seizure detection algorithms in ICU/EMU environments and some efficient ways to achieve the required goals. The second selected paper presents a model that achieves nearly perfect performance, but since it was trained and evaluated with a very homogeneous data source, it is very unlikely that the good performance will translate to clinical environments, where the EEG records present a much broader plurality of conditions. The preprocessing steps presented in the paper, which involves the adaptive way of collection of samples, represent a useful approach. DCNN capability of evaluating correlation between adjacent filters could additionally be a very good approach for the seizure detection task because the majority of False Positives occur due to artifacts. Evaluation of correlation along multiple channels can efficiently help to decrease the False Alarms while maintaining the sensitivity.

Table of Contents

[1 Introduction 1](#_Toc480465156)

[2 Importance of QEEG tools in Neurology 1](#_Toc480465157)

[2.1 QEEG Tools 1](#_Toc480465158)

[2.2 Study conducted on ICU EEGs for Seizure Identification 2](#_Toc480465159)

[2.3 Performance of Seizure detection algorithm (Persyst Inc. software) 3](#_Toc480465160)

[3 Seizure detection using SVM 3](#_Toc480465161)

[3.1 CHB-MIT dataset 3](#_Toc480465162)

[3.2 Preprocessing data before applying it to SVM 3](#_Toc480465163)

[3.3 Algorithm execution and Results 4](#_Toc480465164)

[4 Convolutional neural networks 7](#_Toc480465165)

[4.1 What is CNN and how CNN works? 7](#_Toc480465166)

[4.2 Variants of Basic CNN 8](#_Toc480465167)

[4.3 The Neuro-Scientific Basis for Convolutional Networks 10](#_Toc480465168)

[5 Doubly convolutional neural networks 10](#_Toc480465169)

[5.1 DCNN implementation and Algorithm development 10](#_Toc480465170)

[5.2 Results of DCNN 12](#_Toc480465171)

[6 Conclusion 13](#_Toc480465172)

[7 References 14](#_Toc480465173)

# Introduction

Epilepsy which is the main primary cause of having epileptic seizures affects approximately 1% of the world’s population (Annegers, 1997). To detect seizures, many methods and related technologies have been emerged with time. Most cost-effective and convenient method developed so far is Scalp EEGs which is a non-invasive way to record electrical activity generated by brain.

To detect seizures automatically, there has not been much work done in software development side to ease lives of EEG experts; mainly due to lack of information about patient’s history, clinical correlation and other heuristic approaches that are being considered during identification. One approach using SVM was proposed by a research group from University of Michigan where seizure versus non-seizure EEG detection rate was reported correctly on 83% of patients with reduced False positives on 70% of patients.

Deep learning which is recently revolutionizing the machine learning field (especially in speech & image recognition) has not been efficiently applied to solve such a difficult problem to our best knowledge. Evolved version of CNN called “Doubly Convolutional Neural Network” could become a good approach to solve seizure detection problem. DCNN uses sub-filters known as k-translation correlation filters to quantify convolved information obtained in a more formal fashion. Since, EEG channels highly rely on each other for detecting seizures, artifacts and other relevant events. Taking a correlation among the channels could potentially increase the specificity of the system and hence reduce number of False Positives.

# Importance of QEEG tools in Neurology

## QEEG Tools

With the advent of technology, various tools and techniques such as FMRI, EEG have been developed to detect brain damage/lobe-isolation, epileptiform activities and epileptic seizures. Recently, hospitals generate long hours of EEGs (LTM, CEEG) for individual patients being admitted, and neurologists/technologists need to review all epochs to find out events of interest. This is a very tedious, time consuming task and susceptible to missing events. To ease the process, various transformation techniques of EEG signals using DSP is developed to effectively find out events of interests (e.g. Seizures). Such tools are called QEEG (Quantitative EEG) tools which interprets EEG waveform in much effective and abstract form. QEEG tools can have multiple display methods to interpret electrical activities related to brain. Some examples of QEEG displays are aEEG, Asymmetry Index, CDSA (Color Density Spectral Array), etc (Haider et al, 2016) ()

The gray boxes in shows the possible seizures occurred. The QEEG windows shown here are abstract form of 6-hour long EEGs. Detecting seizures from these windows makes training of nurses/technologists much easier and eventually identifying seizures much faster. QEEG tools allows neurologists to diagnose more patients per unit time and help them take necessary actions sooner. Today, such (QEEG) tools are the only reliable, faster and convenient way of detecting seizure events for neurologists.

EMU (Epilepsy Monitoring Unit) and ICU (Intensive Care Unit) can have different patterns of EEGs due to effects of medication. A study was conducted on 15 ICU patients to check reliability of QEEG tools on CEEG files collected from ICU environment.

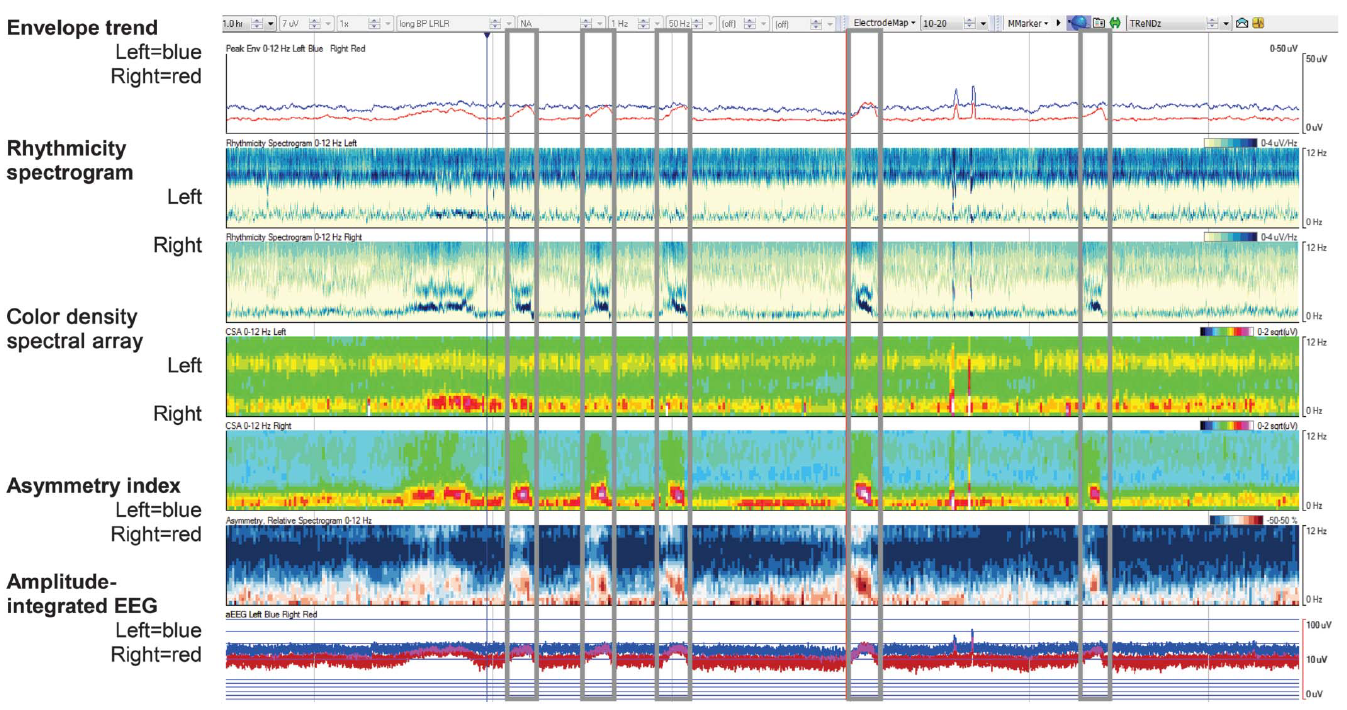


Figure . Example of a 1-hour quantitative EEG (QEEG) panel without automated seizure detection (SzD) as viewed by the QEEG and QEEG + raw reviewers

## Study conducted on ICU EEGs for Seizure Identification

EMU (Epilepsy Monitoring Unit) and ICU (Intensive Care Unit) can have different EEGs due to effects of medication on patients. A study from Emory University was conducted on 15 ICU patients to check the reliability of QEEG tools and seizure detection algorithm (Persyst Inc.) on ICU EEGs (Haider et al, 2016). 18 expert neurophysiologists contributed in this study where 9 of the neurologists prepared a gold standard database as a reference. Other 9 neurologists examined the EEGs using (1) only QEEGs and (2) raw EEGs + QEEG slides.

From 126 total seizure events, there were 32% generalized, 36% hemispheric, 28% focal and remaining 4% indeterminate seizures. Using QEEG review only, neurologists could detect 67% (mean) and with QEEG + raw EEG, 68% of total seizures were identified. shows a good comparison of neurologist’s and automatic seizure detection algorithm’s capabilities to identify seizures using QEEG slides only and QEEG + raw EEGs. 1 min. and 2.5 min. variation were allowed for identifying seizures. From , We observe that, using QEEG tools temporal accuracy for identification of seizures is less accurate. SzD (Automatic Seizure detection algorithm) has very bad sensitivity for identifying seizures too. (with significantly low FP)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Allowed margins for Seizure identification | Q | | QR | | SzD | |
| Sensitivity | FA | Sensitivity | FA | Sensitivity | FA |
| 1 min. Onset variation | 51% | 1/hour | 63% | 0.5/hour | 25% | 0.07/hour |
| 2.5 min. Onset variation | 67% | 1/hour | 68% | 0.5/hour | 27% | 0.07/hour |

Table . Sensitivity and FA of Expert NeuroPhysiologists for identification of Seizures using Q (QEEG only), QR (QEEG + raw EEG) and SzD (Automatic Seizure detection algorithm)

## Performance of Seizure Detection Algorithm (Persyst Inc. software)

The paper provides very little information about the performance of seizure detection algorithm. Recently, the most popular seizure detection system which also has QEEG tools built-in is Persyst. Persyst’s sensitivity on discussed ICU EEG database is 26.5% with FPR of 0.07/hour (Haider et al, 2016). The threshold for seizure detection was set in a way that it shows minimal False Positives (FP). As we can see that sensitivity of seizure detection algorithm is extremely low which encourages us to develop a more efficient algorithm to detect seizures using state-of-art deep learning techniques.

# Seizure detection using SVM

## 3.1 CHB-MIT dataset

CHB-MIT is one of the open source database which contains a small subset of seizure data. In this database, there are 22 subject cases have been recorded (22 Males and 17 Females in ages ranging from 1.5 years to 22 years). Because this data is available as open-source with annotations, it has been popular among the research groups who are working on seizure detection algorithm development (Esbroeck et al, 2016, p. 309) (Shoeb et al, 2011). A group from University of Michigan has developed an algorithm based on Support Vector Machine with unique adaptation based preprocessing of data (Esbroeck et al, 2016, p. 309). They claim to have better performance on seizure detection by preparing data certain way and applying SVM on it.

The research conducted by Alex Van Esbroeck’s research team focused on intra-patient variation of seizure morphologies. With the fact that there is a limited amount of data to train classifiers specific to each patient’s seizure types and to avoid Overfitting for similar repeating seizure patterns, this group proposes seizure detection with multi-task learning framework. By leveraging a formulation of multitask learning that couples the parameters of individual tasks, the proposed approach bootstraps shared knowledge between seizures of different morphologies to identify common structure present in all types of seizure observed.

## Preprocessing data before applying it to SVM

For data preparation, they use adaptive segmentation approach that places boundaries where the energy of the signal is changing sharply. The signals used for P individual channels can be denoted as = [ [n]….. [n]]. The segmentation used here is discrete form of the nonlinear energy operator [NLEO] to identify the points where signal energy is changing. The NLEO for a channel can be defined as

|  |  |  |
| --- | --- | --- |
|  | [n] = [n – 1][n – 2] - [n] [n – 3] | (1) |

Segment boundaries in each channel are identified by using the NLEO with a sliding window. [n] then measures the sum of the absolute difference in frequency-weighted energy within 2N length window centered at sample n over all P channels:

|  |  |  |
| --- | --- | --- |
|  | [n] = | (2) |

The value of [n] suggests the change in energy. The threshold value for define boundaries is defined as:

|  |  |  |
| --- | --- | --- |
|  | T[n] = | (3) |

The final segmentation boundaries are detected by finding the local maxima of the threshold function.

The collecting appropriate features is an essential part of any classification task related to machine learning. The features, according to this paper, are collection of spectral energy levels for all concatenated channels which are pre-filtered signals in range 0.5-25 Hz. At the final level stage, further concatenation of feature-data from previous two windows are used to stack them and yield final feature vector.

## Algorithm execution and Results

Patients with epilepsy can exhibit various types of seizure patterns. To increase the performance of an algorithm not only for inter-patient variability but also intra-patient variability, multitask learning approach is used so that one can exploit the shared structure between features. In the proposed paper, shared structure during seizure is being exploited as a means of bootstrapping shared knowledge between seizures of different morphologies. The reason for doing that is lack of data and an effort to generalize the training for each seizure type related to individual patient.

In standard two class SVM classification, to define decision function with maximum margin between data points and the boundary, one should find solution to following equation:

|  |  |  |
| --- | --- | --- |
|  |  | (4) |

The approach they are using here learns solutions for T tasks using a classification function for each task t, . The task specific separating hyperplane is defined as

|  |  |  |
| --- | --- | --- |
|  |  | (5) |

Where is shared across all the tasks and is specific to each task t. To separate hyperplanes, we optimize optimization function from calculating cost function as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (6) |

and are positive regularization parameters and are slack variables which measures error made by each final mode . The regularization coefficient λ in Eq. (4) corresponds to /T. Once solved, the multi-task and can be obtained from w of the standard SVM.

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

This shows an unsupervised approach to separate the individual types of seizures per patient. This results in a collection of detectors which are specific to the specific to the seizure, corresponding to , the hyperplanes for each task/seizure. These hyperplanes are a combination of discriminative component shared across all tasks and the task-specific components .

Just to obtain seizure/no-seizure classification, we can remove component (task/seizure-type specific component) and use only shared component . Then the classification function becomes

|  |  |  |
| --- | --- | --- |
|  | f(x) = | (8) |

Specific to seizure-type and seizure/no-seizure classification can be understood pictorially from Figure 2.

Here, , shows two dimensional approach showing the examples of windows from 3 seizures, s1, s2, s3 represented by ‘ + ‘ and non-seizure windows represented by ‘ – ‘. (a) shows common discriminant direction shared among all seizure-specific directions v1, v2, v3 learned by multi task SVM. Figure 2(b) shows the resultant hyperplane when only the shared direction is used for classification, and contrasts it with the hyperplane resulting from the standard SVM discriminant direction w. As one can see that shared direction form multi-task learning allows decision boundary to contribute equally on each seizure-types instead of overfitting towards more likely seizures-types.

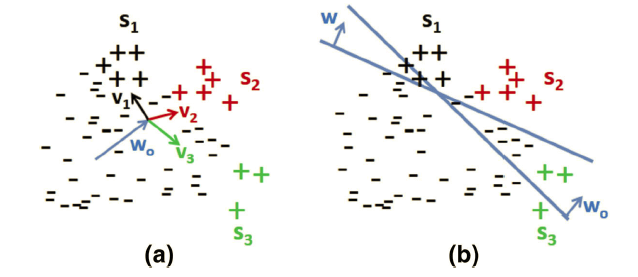


Figure . (a) shows a common component and seizure-specific components (b) sketch of the resultant hyperplane corresponding to standard SVM decision w and the resultant hyperplane corresponding to our approach

Results of seizure detection are measured based on AUC operating point, Latency, False Positive Rate(FPR) and difference in percentage between classic SVM versus proposed approach. These numbers are collected when there was 100% seizure detection. Table 2 shows the results for this proposed approach and here, FPR decreases by 27.17% (median) for multi-task classification.

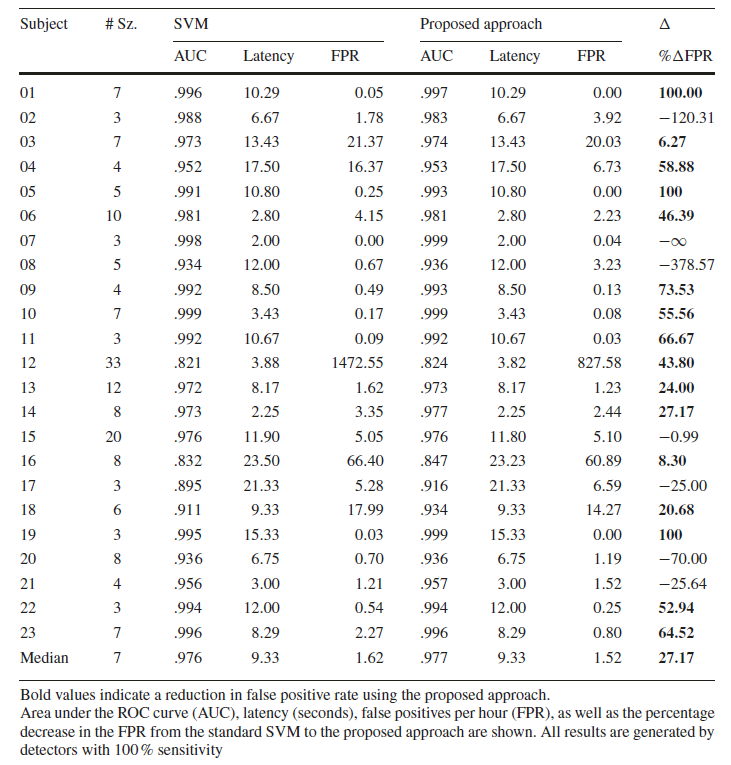


Table . Performance of the standard SVM and the proposed method on each subject in the CHB-MIT dataset, as well as number of seizures per subject

# Convolutional neural networks

## What is CNN and how CNN works?

CNNs are a special kind of neural network for processing data that has a grid like topology. Examples include time-series data or Images. In time-series data, one dimension would be regular time intervals. Convolutional Networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

In discrete world, the convolution operation can be defined as

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

Here, in convolutional network terminology, the first argument (the function of x) to the convolution is often referred to as the input and the second argument (the function of w) as the kernel. The output is sometimes referred to as the feature map. In machine learning applications, these arguments are usually multidimensional arrays and referred as “tensors”.

Unlike other neural networks where each node is assigned a weight defined, CNN can be configured in a way that it takes very less parameters and provide efficient results. CNNs typically have sparse weights instead of tied weights and each member of kernel is used at every position of the input (except for some of the boundary pixels, depends on the design decisions regarding the boundary) (Goodfellow, Bengio & Courville, 2017). These usage of shared parameters makes CNN very efficient. Further advancements can be made for efficiency depending on the application.

A typical layer of convolutional network consists of three stages (). In the first stage, the layer performs several convolutions in parallel to produce a set of linear activations. In the second stage, each linear activation is run through a non-linear activation function, such as rectified linear activation function (detector stage). In the third stage, we use a pooling function to modify the output of the layer further.

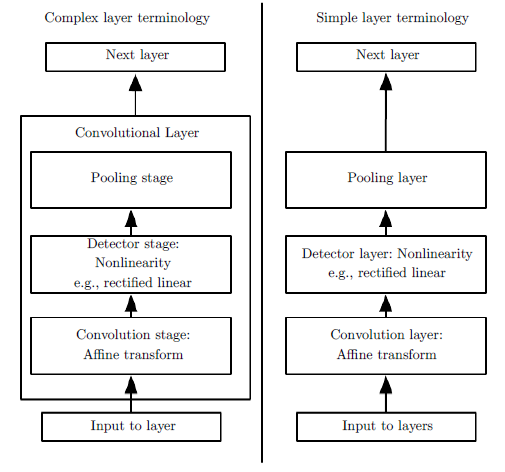


Figure . The components of a typical convolutional neural network layer. There are two commonly used sets of terminologies for describing these layers. Complex layer terminology can contain many stages like simple layer inside a single stage

## Variants of Basic CNN

When we use single kernel, it can only extract one kind of feature at many spatial locations. Usually, we want to extract more features at different spatial locations. Sometimes, when using multiple features, we may want to skip some positions of the kernel to reduce the computational cost (at the expense of not extracting all features). This can be done by down-sampling function. CNNs are mostly being used for image recognition tasks. When working with color images, we consider input/output of the convolution as being 3D-tensors, with one index being different channels (colors RGB), and remaining two indices being spatial coordinates of each channel.

One essential feature of any convolutional network implementation is the ability to implicitly zero-pad the input to make it wider. Without “zero padding” the width of the network shrinks by one pixel less than the kernel width at each layer. Without zero-padding we are forced to choose between shrinking the spatial extent of the network rapidly and using small kernels. -1 and -2, shows the representation of each cases described.

The convolution is a linear operation. The three operations: convolution, backpropagation from output to weights and backpropagations from output to inputs are required to compute all the gradients needed to train any depth of feedforward convolutional network.

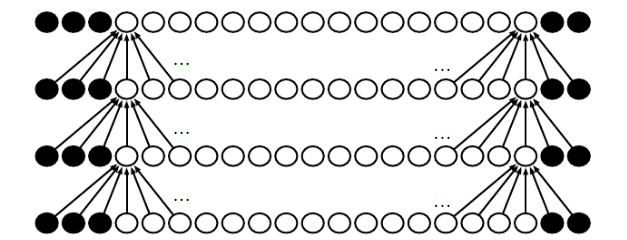


Figure -2 Kernel of width six at each layer with zero padding

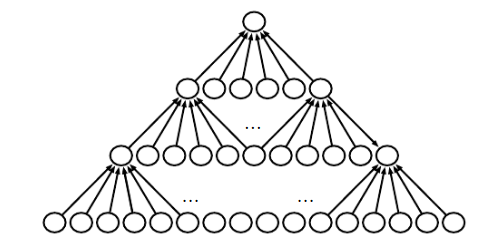


Figure -1 Kernel of width six at each layer without zero padding. Network shrinks by five pixels at each layer

For connected layers in Convolutional Networks, it is typical to assign each unit its own bias. It is common to assign one bias per channel of the output and share it across locations within each convolution map. In other cases, it is more efficient to share biases in a non-linear form. For example, when using implicit zero padding. Detector units at the edge of the image receive less total input and may need larger biases.

## The Neuro-Scientific Basis for Convolutional Networks

Some of the key design principles of neural networks have been drawn from neuroscience. The Convolutional Neural Networks are inspired from the observation of functioning capability of Primary Visual Cortex.

* Visual cortex is arranged in a spatial map. It has 2-D image structure on retina. Convolutional networks capture this property by having their features defined in terms of two dimensional maps.
* Visual cortex contains simple cells. These cell functioning can be characterized by a linear function of the image in a small, spatially localized receptive field. The detector units of a convolutional networks are designed to emulate these properties.
* Visual cortex also contains complex cells. These cells are invariant to small shifts in the position of the feature. This inspires the pooling units of convolutional networks.

# Doubly convolutional neural networks

## DCNN implementation and Algorithm development

The third paper of my exam is related to the DCNN (Doubly Convolution Neural Network). Which is just simply two step convolution procedure. In a well-trained CNNs many of the learned filters are slightly translated version of each other. To quantify the correlation between filters, we include one parameter which is called “k-translation correlation value” between the two convolutional filters within the same layer (Zhai et al, 2016, p. 1082). We define this as:

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

Where, T( ∙ , x, y) denotes the translation of the first operand by (x, y) along its spatial dimensions, with proper zero padding at border to maintain the shape; denotes the flattened inner product, where the two operands are flattened into column vectors before taking the standard inner product.

The correlation between adjacent filters are being observed by absolute images and by adding standard Gaussian samples from the initial layers which indeed shows very high correlation between channels. DCNN allocates a set of meta filters which has filter sizes that are larger than the effective regular CNN filter sizes (). In DCNN, effective filters can be extracted from each meta filter, which corresponds to convolving the meta filters with an identity kernel. All the extracted filters are then concatenated, and convolved with the input.

Let’s consider an image as a real-valued 3D-tensor, where c is the number of channels; w, h are the width and height, respectively. We define the convolution operation, denoted by ,as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

Here, is the input image; is a set of filters, with each filter of shape ; is the output image. The spatial dimensions of the output image are by default + z – 1 and , respectively. One can also pad number of zeros at the borders of to achieve different output spatial dimensions.

Comparing this with DCNN network, double convolution operation, denoted by × as follows:

|  |  |  |
| --- | --- | --- |
|  |  | (12) |

Here, and are the input and output image, respectively. are a set of meta-filters, with filter size , ; is the intermediate output of double convolution; Defines a spatial pooling function with pooling size .

The image shown in and show a good representation of how DCNN works in contrast with CNN by applying meta-filters within the same layer to find out correlation between two adjacent filters.

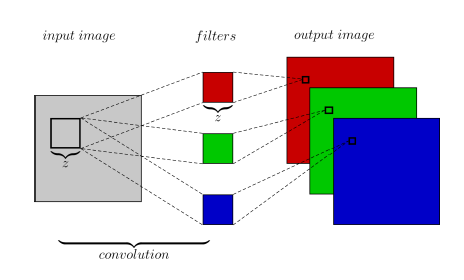


Figure ‑ Architecture of a convolutional layer

After the two-step convolution phase, a spatial pooling of size is then applied along this resulting output map, whose output is then flattened in to a single column vector. Which is a feature map with channels. It is also possible to use different variants of DCNN such as ConcatDCNN and MaxoutDCNN.

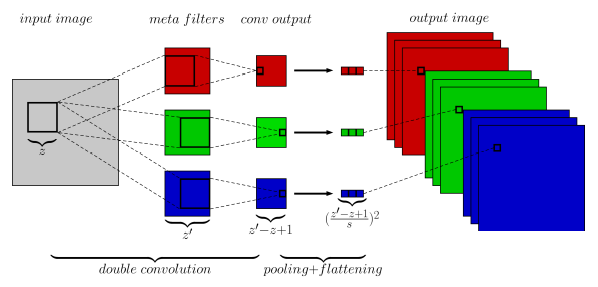


Figure ‑ The architecture of Doubly Convolutional Neural Network (DCNN) layer

## Results of DCNN

The results using DCNN, CNN and a CNN variant (MaxoutCNN) were being compared on an open source database CIFAR-10 and CIFAR-100 with and without data augmentation. The results are shown in

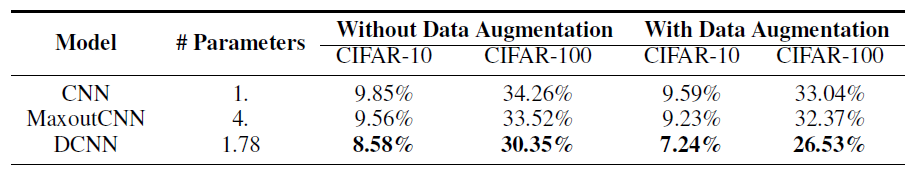


Table Test errors on CIFAR-10 and CIFAR-100 with and without data augmentation

The numbers shown in are test errors in percentage. It seems clear that error percentage by DCNN on CIFAR database is relatively low using small number of parameters.

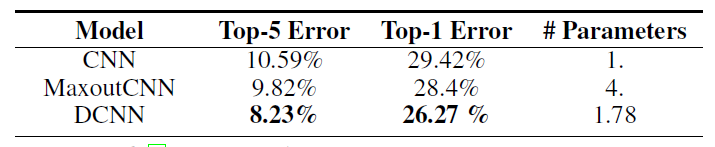


Table Test errors on ImageNet

shows similar results are obtained from ImageNet database.

We can see that reduction in error ranges from 0.98% up to 3.15% for data without augmentation. With dataset with data augmentation, reduction of error is observed by 2.35% up to 6.51%.

# Conclusion

From the paper studied for this exam, following conclusions have been made. With increasing number of Epilepsy patients and with advancement in EEG recording technology. Neurologists are in need of tools such as QEEG or automatic Seizure detection algorithm which can fasten the process of diagnoses of patients. The preprocessing approaches used in paper “Multitask Seizure Detection: addressing intra-patient variation in seizure variabilities” are very innovative to isolate the events of interest and to prevent overfitting on common seizure types; but training and testing models on CHB-MIT database is not an ideal database for testing tool for empirical world application. DCNN seems like a very good approach for seizure detection algorithm because it looks for correlation between primary feature maps. If we develop seizure detection algorithm using DCNN, it would check correlation between spatial locations on scalp, which can generate feature maps that can focus on artifact reduction.

# References

Annegers, J. F. (1997). *The treatment of epilepsy: Principle and practice*. Baltimore: Williams and Wilkins.

Van Esbroeck, A., Smith, L., Syed, Z., Singh, S., & Karam, Z. (2016). Multi-task seizure detection: addressing intra-patient variation in seizure morphologies. *Machine Learning*, *102*(3), 309–321.

Shoeb, A., Kharbouch, A., Soegaard, J., Schachter, S., & Guttag, J. (2011). An algorithm for detecting seizure termination in scalp EEG. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS* (pp. 1443–1446).

Goodfellow, I., Bengio, Y., & Courville, A. (2017). Deep Learning (1st ed.). Cambridge, MA, USA: MIT Press.

Zhai, S., Cheng, Y., Zhang, Z. (Mark), & Lu, W. (2016). Doubly Convolutional Neural Networks. In *NIPS* (pp. 1082–1090).

Haider, H. A., Esteller, R., Hahn, C. D., Westover, M. B., Halford, J. J., Lee, J. W., … Pargeon, K. (2016). Sensitivity of quantitative EEG for seizure identification in the intensive care unit. *Neurology*, *87*(9), 935–44.