DEEP LEARNING FOR EEG DATA ANALYSIS

CHEAH KIT HWA

A project report submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Engineering (Hons) Electronic Engineering

Faculty of Engineering and Green Technology
Universiti Tunku Abdul Rahman

April 2018

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : _________________________

Name : _________________________

ID No. : _________________________

Date : _________________________

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"DEEP LEARNING FOR EEG DATA ANALYSIS"** was prepared by **CHEAH KIT HWA** has met the required standard for submission in partial fulfillment of the requirements for the award of Bachelor of Engineering (Hons) Electronic Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : ___________________________

Supervisor : Dr. HUMAIRA NISAR

Date : ___________________________

Not very specially dedicated to
my beloved grandmothers, grandfathers, mother and father,
and relatives,
and friends of any kind of proven-or-unproven lifeforms,
and enemies if there are any,
and strangers, kind or mean or indifferently neutral,
and all other forms of energy that have directly or indirectly ever interacted with me,
and all the unthought of.

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Dr. HUMAIRA NISAR for her invaluable advice, guidance and her enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parents and friends who had helped and given me encouragement.

DEEP LEARNING FOR EEG DATA ANALYSIS

ABSTRACT

Electroencephalogram (EEG) is a multi-dimensional time-series brain signal that is highly information packed. While an EEG has high potential to serve in medicine (e.g. disease diagnosis, prognosis, pre-disease risk identification), psycho-physiology (e.g. mood classification, stress monitoring, alertness monitoring, sleep stage monitoring), brain-computer interface application (e.g. thought typing, prosthesis control), and many other areas, the classical design of EEG feature extraction algorithms and EEG classifiers is time-consuming and challenging to fully tap into the vast data embedded in the EEG. Deep learning (or deep neural network) which enables higher hierarchical representation of complex data has been strongly suggested by a wide range of recent research that these deep architectures of artificial neural network generally outperform the classical EEG feature extraction algorithms or classical EEG classifiers.

In this project, deep neural network architectures have been constructed to perform binary classification on an EEG dataset that was shown by traditional EEG feature extraction methods to have no significant difference between its two data pools (resting EEG recorded before and recorded after listening to music). The convolutional neural network (CNN) model constructed in this project has achieved a validation accuracy of $75 \pm 1\%$ using the same EEG dataset.

Using the top performing CNN architectures, short duration of relaxing music listening is found to affect the EEG signals generated by the frontal lobe more than the other lobes of the brain; and also to affect the EEG generated by the left cerebral hemisphere more than the right hemisphere.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

$J(\theta_0 \ldots \theta_n)$	cost function or objective function of a model for optimization with an arbitrary (n) number of parameters
θ_i	the i-th parameter of the cost function J
$\partial J / \partial \theta_i$	the partial derivative of the cost function w.r.t. the i-th parameter
w	the weight parameter
dw	change in parameter w
b	the bias parameter
db	change in parameter b
β_m	the exponential decay rate for the momentum of change
β_r	the exponential decay rate for RMSprop
α	the learning rate of a model
ε	a small value assigned to avoid division by zero

AF3, AF4	Antero-Frontal channels of 14-channel EEG
ANN	Artificial Neural Network
BCIC	Brain-Computer-Interface Competition
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture by NVIDIA
DNN	Deep Neural Network
EEG	Electroencephalogram
ELU	Exponential Linear Unit
FC-MLP	Fully-connected Multilayer Perceptrons
F, T, P, O	Frontal, Temporal, Parietal, Occipital EEG channels
MLP	Multilayer Perceptrons
MLPNN	Multilayer Perceptron Neural Network
ReLU	Rectified Linear Unit

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

1.1.1 Electroencephalogram

Electroencephalograms (EEG) are recordings of the electrical potentials of the brain typically measured from the scalp, as signal waveforms of varying frequencies and amplitude (in mV). The EEG is packed with information regarding the electrical activities of the brain, be it pathological or physiological.

Hence, EEGs are very useful in the medical field (such as diagnostic purposes, real-time monitoring of clinical progress of patients, prognostic purposes, and the pre-disease identification of prodromal neuro-pathological signals in preventive healthcare of increasing importance), for Brain-Computer-Interface (BCI) applications (such as thought-typing, prosthetic limbs control, and many others which can potentially improve the quality of life of the people with motor disabilities, as well as the normal), and myriad forms of other potential applications such as drowsiness warning system for drivers or lie detection for criminal investigation.

1.1.2 Machine Learning

Machine Learning is the set of soft computing techniques that allow the computers to have learning capability without being explicitly programmed. Machine Learning (or such soft computing paradigms) is particularly significant for studying, analyzing, and modeling solutions for very complex (usually real-life) tasks or phenomena. These groups of tasks or phenomena are usually too complicated to be possibly modeled or solved with conventional hard computing methods.

Soft computing and Machine Learning are not the recently discovered paradigms. The most early works related to soft computing can be dated back to as early as the 18^{th} century where the ideas of Bayes' theorem started to emerge. The first neural network machine was invented in the 1950's and the Rosenblatt's perceptron was invented in 1957. Even the Backpropagation algorithm had been published pretty early in 1970. Despite all these early exciting discoveries and invention, the field of Artificial Intelligence had its "winter" in the 1970's due to several reasons which were mostly due to insufficiency in the hardware computing capability. Even to date, the hardware processing power and memory requirements by the Machine Learning methods are still in a pressing need for improvement. Hence, it was very disheartening back in the 1970's and 80's that the soft computing algorithms were not being able to be realized or implemented. The recent re-emergence of "spring" of Machine Learning is at a large extent due to the significant progress in the speed of the hardware processors.

1.1.3 Deep Learning

Deep Learning is a paradigm of Machine Learning technique, which makes use of the artificial neural network. However, the Deep Learning differs from the earlier forms of conventional neural network in a way that Deep Learning adopts more hidden layers (thus the term "Deep"). Besides, Deep Learning architectures are capable of undertaking the supervised learning, or unsupervised learning, or even both (unsupervised pre-training followed by supervised tuning to one model).

1.1.4 Convolutional Neural Network

The Convolutional Neural Network (CNN) is a method of Deep Learning that incorporates the convolutional layers into the neural network. The architecture of the CNN is generally composed of the alternating layers of convolutional layers, rectifier layers (such as ReLU), and pooling layers, before passing the convoluted and pooled outputs into the Fully-Connected (FC layers) Neural Networks. The outputs of the convolution are termed feature maps, as they are regarded as containing the features extracted from the original input matrix by the convolutional kernels.

Fully-connected layers are Multi Layer Percpetrons (MLP). MLP are modelled in such a way that it is capable of being trained to recognize patterns and perform categorization. The training of fully-connected layers was achieved with backpropagating the output layer errors backward layer-by-layer, in order to gradually adjust the neuronal input weights for the most optimal values.

The final output of categorization is usually achieved with functions such as Softmax, which is defined as follows:

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K. \tag{1.1}$$

The output of Softmax function is a K-dimensional vector $\boldsymbol{\sigma(z)}$ of real values in the range of [0, 1] that add up to 1, giving a probability distribution of K different possible outcomes. This probability distribution can be regarded as the representation of a categorical distribution. For example, referring to Figure 1.1, among the four categories (dog, cat, boat, bird), the "boat" category has the highest probability and thus is the most likely input signal.

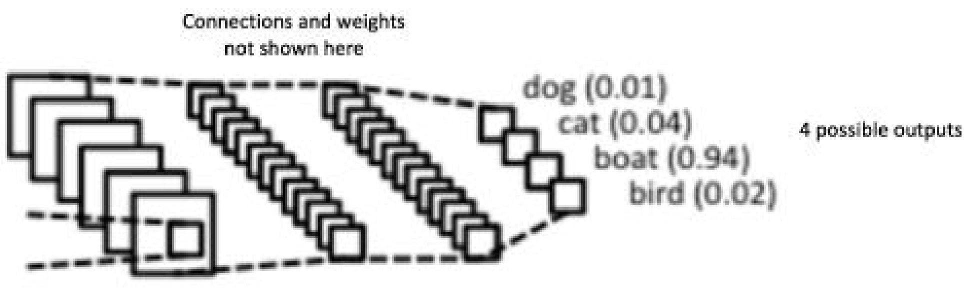

Figure 1.1: FC layers followed by Softmax function (Karn, 2016)

1.2 Problem Statement

EEG signals are packed with information and hence it requires lots of effort and time to perform manual analysis or decoding of these signals. Siuly and Li (2014) stated that to manually design the feature extraction model for multiple-class electroencephalogram (EEG) signal classification is an extremely challenging task because the true representative features/patterns have to be identified and extracted precisely from the multidimensional time series of EEG measured from the brain.

With the advances in the techniques for modelling the deep learning architecture, deep learning has revolutionized the computer's capability for processing information-packed data. For example, convolutional neural network for image processing has provided solutions to challenges previously encountered by the computer vision community, while recurrent neural network has resulted in much improvement in the processing of time-series signals such as speech processing.

It is thus very likely that deep learning will improve the analysis of EEG signals as well. A number of different studies (Ren and Wu, 2014; Behncke et al, 2017; Schirrmeister et al, 2017) trained and tested various architectures of deep learning for EEG data analysis and reported improved accuracies compared with the state-of-art EEG feature extraction methods. Yet, the research in the application of deep learning on EEG analysis is a new area of study and further analytical accuracy improvement is in need for much more reliable practical application.

In this project, various architectures of DNN for EEG analysis, feature extraction, and classification will be carried out.

1.3 Aims and Objectives

The objectives of this project are as follows:

i) To study the techniques of deep learning modelling and to construct deep learning models for EEG classification

ii) To investigate the performance of pure multilayer perceptrons (MLP) models for EEG signal classification

iii) To investigate the performance of convolutional neural networks (CNN) for EEG signal classification

iv) To interpret and analyze the results.

CHAPTER 2

LITERATURE REVIEW

2.1 Machine Learning for Epileptic EEG Pattern Recognition

The classification of EEG signals using various versions of artificial neural networks have been published with higher sensitivity, specificity, and accuracy than other traditional feature extraction and statistical methods.

Patnaik and Manyam (2008) applied the neural network for the identification of epileptic EEG segment from the non-epileptic EEG. They used discrete wavelet transform (DWT) for feature extraction, followed by a feed-forward backpropagating artificial neural network (ANN) for classification, with the training set for the ANN model being selected by a genetic algorithm instead of randomization. They improved their classification result by incorporating a post-classification stage using harmonic weights. The training and validation were done using the invasive pre-surgical EEG recording of 21 patients with medically intractable focal epilepsy. The average specificity of 99.19%, sensitivity of 91.29%, and selectivity of 91.14% were obtained. Each patient's EEG recording contained at least 50 min of pre-ictal and 50 min of post-ictal recording and the average duration of EEG with epileptic data was 7.73 min for one patient.

Subasi and Ercelebi (2016) compared logistic regression and neural network models for EEG signals (epileptic vs. normal data) classification. They obtained 89% accuracy using logictic-regression based classifier, which was lower than the two neural network models. The Multi Layer Perceptron Neural Network (MLPNN) trained with common error backpropagation algorithm achieved an accuracy of 92%; while the MLPNN trained with Levenberg-Marquardt (L-M) optimization method achieved an even higher accuracy of 93%. The MLPNN models were trained with a total of 300 EEG examples (102 epileptic and 198 normal EEG), and validated with another set of 200 EEG examples (88 epileptic and 112 normal EEG).

Satapathy, Dehuri and Jagadev (2017) performed classification of EEG for epileptic seizure identification using a version of neural network known as Radial Basis Function Neural Network (RBFNN). Their RBFNN was trained for mean square error optimization with a modified Particle Swarm Optimization (PSO). The improved PSO (termed IPSO in the paper) was designed for improving the searching speed of traditional PSO for global optimum. The RBFNN with IPSO had achieved a maximum accuracy of 99%.

2.2 Deep Learning for Sleep Stages Classification using EEG

Supratak et. al. (2017) constructed a Deep Learning model which utilizes:

1. convolutional neural network (CNN) to extract time-invariant features, and
2. bidirectional long-short-term-memory (bidirectional-LSTM) to learn transition rule among sleep stages from EEG epochs.

Their model was trained with a two step training algorithm which:

1. pre-trains the model using over-sampled data to lessen class-imbalance problems, and later
2. fine tunes the weights of the pre-trained model with sequences of EEG epochs to encode the model with necessary patterns for sleep stages classification.

The training dataset was from the F4-EOG channel of 62 subjects, giving rise to a total of 58600 EEG epochs, with the total recording duration close to 490 hours. The model achieved an accuracy of 86.2% and the macro F1-score of 81.7 as shown in Table 2.1.

Table 2.1: Confusion Matrix for the Performance of the DeepSleepNet (Supratak et. al., 2017)

	Predicted					Per-class Metrics		
	W	N1	N2	N3	REM	PR	RE	F1
W	**5433**	572	107	13	102	87.3	87.2	87.3
N1	452	**2802**	827	4	639	60.4	59.3	59.8
N2	185	906	**26786**	1158	499	89.9	90.7	90.3
N3	18	4	1552	**6077**	0	83.8	79.4	81.5
REM	132	356	533	1	**9442**	88.4	90.2	89.3

2.3 Cognitive/Mental State Interpretation using EEG with Deep Learning

Hajinoroozi, Mao and Huang (2015) applied Deep Learning to perform prediction of driver's drowsy or alert states using the EEG data. They introduced the Channel-wise Convolutional Neural Network (CCNN) and a variation of CCNN (termed CCNN-R in the paper) which adopted the Restricted Boltzmann Machine (RBM) in place of the convolutional filter/layers of conventional CNN models.

The EEG data set was collected from three studies of the driver's cognitive states using a virtual reality dynamic driving simulator. The simulated driving scenes were night time driving with 100 km/h with perturbation being injected into driving path every 8 to 12 seconds. The reaction times of the drivers were used to determine their alert/drowsy mental states. The dataset was collected from 70 sessions for 37 subjects. The EEG was recorded for 3 seconds before each perturbation was taken into consideration for CNN models training, with a total of 35074 non-overlapping 1s epochs (23074 alert and 15924 drowsy epochs).

In contrast to the conventional CNN which uses 2-D or multi-dimensional convolutional kernels for feature extraction, CCNN applies a 1-D kernel to convolve along each channel (hence channel-wise). After the feature extraction, the categorization with Fully Connected (FC) layers also uses backpropagation for weight optimization. The common kernels for CCNN include the Gaussian or Xavier filters.

A model variation mentioned above (CCNN-R) uses a more complicated feature extraction layers (RBM). The FC layers' backpropagation method has to be adjusted accordingly.

The algorithms' performance were evaluated using Az-score, with the CCNN having achieved the Az-score of 79.63% and the CCNN-R 82.78%. The prediction performance of other popular methods were investigated too, with the LDA achieving 52.81%, SVM achieving 50.38%, and CNN 71.41%.

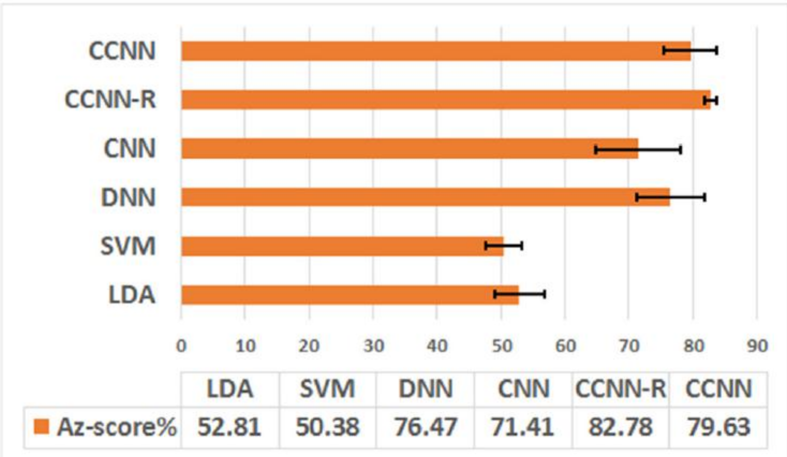

Figure 2.1: Performance (Az-score) of varous machine learning methods at predicting drivers' alertness using raw EEG data (Hajinoroozi, Mao and Huang, 2015)

Behncke et al (2017) attempted to classify the EEG signals of humans observing robot action into two classes (observing a successful robotic operation or observing a robotic failure). The classification task was performed with deep convolutional neural network (deep ConvNets), regularized Linear Discriminant Analysis (rLDA), and filter bank common spatial patterns (FB-CSP) combined with rLDA. Deep ConvNets achieved accuracies of 75% ± 9%, significantly higher than both the other two commonly used EEG classifiers, with the rLDA of 65% ± 10% and the FB-CSP combined with rLDA of 63% ± 6%, as shown in Table 2.2.

Table 2.2: Accuracies of ConvNet, rLDA and FB-CSP at identify EEG of human observing robotic failure (Behncke et. Al., 2017)

paradigm	interval	mean accuracy ± standard deviation		
		ConvNet	*rLDA*	*FB-CSP*
KPO error	2.5-5s	(78.2 ± 8.4) %	(67.5 ± 8.5) %	(60.1 ± 3.7) %
KPO error	3.3-7.5s	(71.9 ± 7.6) %	(63.0 ± 9.3) %	(66.5 ± 5.7) %
RGO error	4.8-6.3s	(59.6 ± 6.4) %	(58.1 ± 6.6) %	(52.4 ± 2.8) %
RGO error	4-7s	(64.6 ± 6.1) %	(58.5 ± 8.2) %	(53.1 ± 2.5) %

Ren and Wu (2014) also compared the performance of a deep learning architecture using Convolutional Restricted Boltzmann Machines (CRBM), to other state-of-art classical feature extraction methods including power band, multivariate adaptive autoregressive (MVAAR), and common spatial pattern (CSP). For 2-class and 4-class classification, the deep learning model achieved accuracies of 83% - 88% which is in general higher than the classical feature extraction methods (80% - 86%). The accuracy of the deep learning method in particular increased as the number of training samples increased from 80 to 240, as shown in Table 2.3 and Table 2.4.

Table 2.3: Mean 2-class motor imagery EEG classification accuracy of various methods (Ren and Wu, 2014)

Training Samples	Correct Rate (%)			
	CSP	*MVAAR*	*Band Power*	*CDBN*
80	**85.38±2.24**	80.35±4.07	81.90±2.75	83.63±1.82
120	85.25±1.94	84.88±3.87	83.59±2.90	**85.94±1.77**
160	85.74±2.33	85.46±2.54	85.00±2.55	**86.04±2.09**
200	85.56±3.10	84.81±4.06	84.63±4.24	**86.06±3.38**
240	85.75±6.18	85.88±5.80	86.13±6.36	**88.25±5.70**

Table 2.4: Mean 4-class motor imagery EEG classification accuracy of various methods (Ren and Wu, 2014)

Training Samples	Correct Rate (%)			
	CSP	*MVAAR*	*Band Power*	*CDBN*
140	80.45±1.62	81.08±1.50	80.54±2.21	**82.02±1.88**
160	81.02±1.50	80.70±2.00	81.53±1.61	**82.41±1.44**
180	85.47±1.44	85.64±2.36	86.09±2.36	**87.33±1.74**

All the research findings discussed above strongly support that deep learning is more powerful at decoding and analysis of the information-packed EEG data. A recent research (Schirrmeister et. al., 2017) indicates that the deep learning for EEG analysis is still having vast room for improvement with all the recent advances in the deep learning modelling techniques.

Literature review has been summarized in Table 2.5a and Table 2.5b.

Table 2.5a: Summary of Literatures Reviewed

Author	Year	Method	Data Input	Purpose / Types of EEG	Model's Performance
L.M.Patnaik, O. K. Manyam	2008	Artificial Neural Network, Genetic Algorithm, Resilient Backpropagation, Discrete Wavelet Transform	EEG	Detection of Epilepsy with EEG	Specificity=99.19%, Sensitivity=91.29%, Selectivity=91.14%
A. Subasi, E. Erç9elebi	2004	Logistic Regression, MLPNN with backprop, MLPNN with Levenberg–Marquardt (L–M)	EEG	Detection of Epilepsy	Specificity: 90.3% 91.4% 92.3% Sensitivity: 89.2% 92.6% 92.8%
S. K. Satapathy, S. Dehuri, A. K. Jagadev	2016	RBF trained with GD, RBF trained with general PSO, RBF Trained with improved PSO	EEG	Detection of Epilepsy	Accuracy of 70.0±0.089, 96.00±0.038, 99.0±0.019
A. Supratak, H. Dong, C. Wu, Y.K. Guo	2017	Convolutional Neural Network, bidirectional Long Short-Term Memory (bidirectional-LSTM)	raw single-channel EEG	Sleep Stages Classification	Average Accuracy of 86.2%
M. Hajinoroozi, Z.J. Mao, Y.F. Huang	2015	Channel-wise convolutional neural network (CCNN), CCNN with Restricted Boltzmann Machine (CCNN-R)	EEG	Predicting Drivers' Drowsy/Alert States	Az-score of 79.63%, 82.78% 9

Table 2.5b: Summary of Literatures Reviewed (cont.)

Author	Year	Method	Data Input	Purpose / Types of EEG	Model's Performance
J. Behncke, R. T. Schirrmeister, W. Burgard, T. Ball	2017	ConvNet, regularized Linear Discriminant Analysis (rLDA), filter bank common spatial patterns (FB-CSP) plus rLDA	EEG	to identify human observers who have noticed robotic action failure	ConvNet: 59.6-78.2% rLDA: 58.1-67.5% FB-CSP: 52.4-66.5%
Y. Ren, Y. Wu	2014	Conv. Deep Belief Net (CDBN), CSP, MVAAR, Band Power	EEG	to classify EEG dataset from BCI Competitions	CDBN: 83.63-88.25% CSP: 85.25-85.75% MVAAR: 80.35-85.88% Band Power: 81.90-86.13%
R. T. Schirrmeister, J. T. Springenberg, L. D. J. Feiderer, et al	2017	DeepConvNet, ShallowConvNet, HybridConvNet, ResidualConvNet, FBCSP	EEG	BCIC Dataset, High Gamma Dataset	DeepConvNet: 70.1-92.5% ShallowConvNet: 71.9-93.9% HybridConvNet: 66.2-92.4% ResidualConvNet: 60.8-88.9% FBCSP: 67.8-91.2%
R. T. Schirrmeister, L. Gemein, K. Eggensperger, F. Hutter, et al	2017	Deep ConvNet, Shallow ConvNet	EEG	TUH EEG Abnormal Corpus	Deep ConvNet: Accuracy 84.8%, Sensitivity 78.0%, Specificity 90.7% ShallowConvNet: Accuracy 84.7%, Sensitivity 74.8%, Specificity 93.0%

CHAPTER 3

METHODOLOGY

3.1 Overview

Two main varieties of deep learning architecture investigated in this project are pure multilayer perceptron (MLP) models and convolutional neural networks (CNN). The impact of modelling techinques and hyperparameters of deep learning models on the model's performance are also investigated, which include the effect of different optimizers, activation functions and dropout rates.

The EEG data used for training and validation of the deep learning models are 14-channel EEG of 26 participants of a music-based neurofeedback training previously conducted by a UTAR FYP student (Phneah, 2017).

3.2 EEG Dataset

3.2.1 Neurofeedback Training

In neurofeedback training, the measurement of brain activity (EEG in this case) is used as the feedback information to the participant for the purpose of attaining desired regulation of the brain function.

The EEG data used for this project is a portion of the EEG recorded at the very initial phase of the neurofeedback study, with each participant having undergone only a single short session of listening to favourite and relaxing music.

3.2.2 Training and Validation EEG Dataset

Three-minute EEG signal was recorded, at sampling frequency of 128Hz, before and during each of the 26 participants listened to their favourite and relaxing music, generating 52 EEG recordings (26 before listening to music and another 26 after listening to music). Each of the EEG recordings, after artifact removal and data cleaning, has different lengths ranging from 80-100 seconds. Hence, only the first 10000 sampling points (about 78 seconds) of each pre-processed EEG are used as the dataset of this project.

Each of the 52 cleaned EEG recordings is then split into 40 sub-segments, generating 2080 EEG recording segments (1040 before music and 1040 after music). Each of the sub-segments has the time span of about 1.95 seconds (250 sampling points). This dataset is shuffled and divided randomly into the training set and validation set at the ratio of 9:1, giving 1872 EEG segments (942 before music and 930 after music) as the training data set and 208 EEG segments (98 before music and 110 after music) as the validation set, as shown in Table 3.1.

Table 3.1: The numbers of categorized EEG data contained in the training set and the validation set.

EEG Dataset	Training set	Validation set
Before listening to music	942 (45.3%)	98(4.71%)
After listening to music	930(44.7%)	110(5.29%)
Total	1872(90%)	208(10%)

3.3 Project Equipment Utilized

3.3.1 Hardware

The computer system used for the training and validation of the models is a Dell Inspiron 7567 laptop, with the following specifications:

- CPU: Intel Core i5-7300HQ 2.50 GHz
- RAM: 4GB DDR4, plus an extra 8GB upgrade
- GPU: NVIDIA Geforce GTX 1050 4GB graphic RAM

The capability of GPU is of utmost importance because the fundamental design of GPUs allows huge amount of parallel computation of the same instructions. This suits the requirement of running deep learning models which are generally designed with large matrix of repetitive computational nodes.

In fact, the NVIDIA GTX 1050 GPU used in this project is designed for gaming purpose and is a rather low end GPU for deep learning research.

3.3.2 Software

The programming language used in this project is the Python language, version 3.6.4, under Anaconda distribution. Anaconda enables convenient creation and management of Python environment (conda environment), under which we can selectively run different tools specifically installed to the particular environment.

The scientific programming Python libraries used in this project include the numpy library, scikit-learn (sklearn) library, and matplotlib library. The Python 'os' library is used to move around, read from, and write to the system's directories. The 'mne' library is used to handle EEG data. And last but not of any less, the 'tensorflow' machine learning library is used for the constructing and running the deep learning models (Abadi, et al, 2016). Table 3.2 summarizes the Python libraries used.

Table 3.2: The Python libraries used in this project.

Python libraries	Usage/Purpose
numpy	to handle data as n-dimensional arrays
scikit-learn (sklearn)	for data preparation before model training
matplotlib	for visualization of data
os	to move around, read from, and write to the operating system's directories
mne	to read and handle EEG data
tensorflow	to construct, run, and analyse machine learning models

3.4 Supervised Learning

Deep learning models can learn or be trained through unsupervised or supervised learning process. Unsupervised learning of a deep learning model will enable the model to divide the dataset into classifiable clusters, without any indication as to which group any training or validation data belongs to. On the other hand, supervised learning, which is the training method used in this project, requires each example (x) of the training data to be associated or encoded with a label (y). After repeated observation of the paired examples of data x and label y, the model learns to predict y from data x.

3.5 Modeling, Training and Validation

Conceiving and constructing a deep learning model involve specifying the type of feature extraction operation to be incorporated (the application of convolutional kernels in convolutional neural network as in this project or the application of feedback loop in the neural network forming a recurrent neural network), the number of network layers (the depth of the model), the number of neuron at each layer (the width of the model), the type of activation function for the neural layers, the model regularization methods such as the drop-out mechanism to prevent overfitting, the choice of error back-propagation optimizer and the learning rate.

The training data set is divided from a total of 1872 segments of EEG into 16 smaller mini-batches, each containing 117 segments of EEG. During the model training stage, the mini-batches are fed batch-by-batch to the model-under-training.

The advantage of this splitting of training data into mini-batches is that (1) training with mini-batches requires less GPU memory, hence allowing to design more complex model with larger architecture; and (2) although mini-batches add noise to the training process, they may help the training process to avoid being trapped in local minima of the loss function.

3.6 Overall Project Flow

The overall project flow is illustrated in Figure 3.2 below, incorporating the “model design. training and validation cycle” illustrated in Figure 3.1 as a step within the overall project flow.

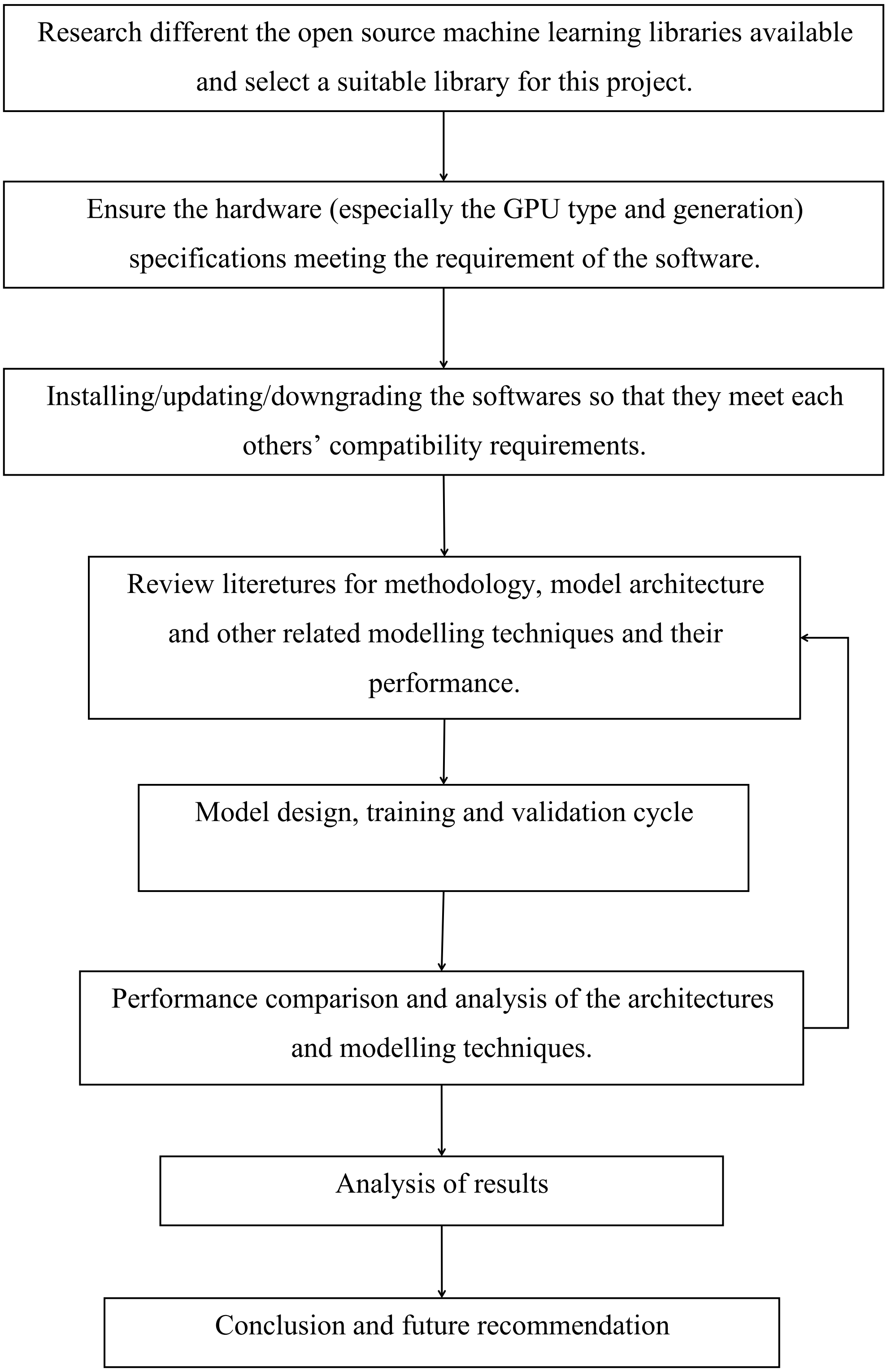

Figure 3.1: Overall methodology flow of the project

3.7 Project Gantt Chart

Timeline (month)	Jun 2017	Jul 2017	Aug 2017	Sep 2017	Oct 2017	Nov 2017	Dec 2017	Jan 2018	Feb 2018	Mar 2018	Apr 2018
Machine Learning & EEG background knowledge and concept study	■	■	■	■	■	■	■	■	■	■	■
Research, selection, upgrade, setup, compatibility adjustment of hardware and software	■	■	■	■	■	■	■				
Research Literature Review for Deep Learning Modelling Techniques		■	■	■	■	■	■	■	■	■	■
Learning Python and its data science libraries				■	■	■	■	■	■	■	■
Learning Tensorflow and its built-in functions						■	■	■	■	■	■
Construct, train, and validate models								■	■	■	■
Result analysis, conclusion, and report preparation									■	■	■

3.8 Project Cost and Sustainability

3.8.1 Cost

A new laptop (Dell Inspiron 7567) was purchased for the project because the old laptop of mine has a very old-generation GPU (NVIDIA Geforce 610M, with NVIDIA computational capability of only 2.1). Therefore, this old GPU is not CUDA-compatible. CUDA is is required for running gpu version of tensorflow.

The new laptop costs RM 3249. Additional RAM is also installed because the Dell Inspiron 7567 has only 4Gb original RAM. The additional 8GB of RAM costs RM385. Assembling a complete desktop workstation of equal capability should cost below RM3000.

All the softwares used are free and open-source.

The EEG data used in this project is from the previous project. Hence, no equipment or license fee is required for EEG collection at this stage.

3.8.2 Sustainability

The hardware power consumption, EEG data volunteers' privacy protection, and the cost of living of the researcher are the only three major areas of concern in this project, for its sustainability.

The hardware (Dell laptop Inspiron 7657) operates at 150W, which is very sustainable.

In details, the CPU (Intel i5-7300HQ) operates at around 3.30 GHz close to its max turbo frequency of 3.50 GHz, consuming 35-45W of power. The major concern is about the lifespan of the CPU is consistently being operated at over-clocked condition.

The GPU (NVIDIA GTX 1050) can have a maximum operational power consumption of 75W. Overheating of the processors should be taken note of with sufficient ventilation for lowering room temperature and processors' temperature.

All the EEG retrieved and used in this project does not contain any personal identification tags or details. No privacy issue should be of concern at this stage.

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Effect of Optimizer, Activation Functions and Dropout Mechanism

In order to investigate the impact of different optimizers, activation functions, and dropout rates on the progress of deep learning training process, all the different modelling techniques are independently tested on the same single convolutional neural network architecture.

4.2 Effect of Optimizer

Two optimization algorithms (the basic gradient descent algorithm and the adaptive moment estimation (Adam)) are tested for their effectiveness in searching the minimal point of the cost function of the deep learning model for EEG classification.

The activation function is fixed as ReLU and the dropout rate is fixed at 50% for either of the optimization techniques. This is to ensure that the changes in the performance of the model are all the result of the change in optimizer, instead of being the combined effect of changing various different modelling techniques.

4.2.1 Basic Gradient Descent Optimizer

The gradient descent optimizer is a very widely used algorithm to perform optimization of neural networks. As outlined by Ruder (2017), gradient descent has three basic variants, namely the batch gradient descent, the stochastic gradient descent and the mini-batch gradient descent. These three variants of gradient descent differ in the amount of data used for calculating gradient of the loss function of the neural network.

The batch gradient descent works as follows:

$$\begin{aligned}&repeat \quad until \quad convergence \quad \{\\ &\qquad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ \ldots, \ \theta_n), \qquad for \quad j = 0, \ \ldots, \ n\\ &\}\end{aligned} \tag{4.1}$$

where J is the cost function, θj is a parameter of the cost function, and α is the learning rate of the algorithm. The symbol := denotes an assignment operation. The gradient of the loss function is calculated using the complete training dataset (Ruder, 2017).

The stochastic gradient descent operation works as follows:

$$\begin{aligned}&repeat \quad until \quad convergence \quad \{\\ &\qquad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ \ldots, \ \theta_n; \ x^{(i)}; \ y^{(i)}), \qquad for \quad j = 0, \ \ldots, \ n\\ &\}\end{aligned} \tag{4.2}$$

where $x^{(i)}$ and $y^{(i)}$ denote any single training example (or input data) and its corresponding training label. The gradient of the loss function, at each training step, is calculated using only any random single training data (Ruder, 2017).

The mini-batch gradient descent works as follows:

$$
\begin{aligned}
&repeat \quad until \quad convergence \ \{ \\
&\qquad \theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J\left(\theta_0, \ \ldots, \ \theta_n; \ x^{(i:i+n)}; \ y^{(i:i+n)}\right), \ for \quad j = 0, \ \ldots, \ n \\
&\}
\end{aligned} \tag{4.3}
$$

where $x^{(i:i+n)}$ and $y^{(i:i+n)}$ denote a particular mini-batch of training dataset (input data and the corresponding labels). The superscript index i will be updated to $i+n$ for each next round of training, hence feeding in another mini-batch of training dataset. The gradient of the loss function is calculated using the mini-batch, instead of the complete training dataset (Ruder, 2017).

Using the above operation, with an optimal value of α (the learning rate), the value of θ_j will be updated towards a local or global converging point where the value of the cost function J will be at a local or global minimum, at which point the derivative of it δJ will be zero.

Algorithmic implementation of the gradient descent optimizer can be done as follow:

$$
\begin{aligned}
&do \ \{ \\
&\quad temp\_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \ \ldots, \ \theta_n) \\
&\qquad\qquad \vdots \\
&\quad temp\_n := \theta_n - \alpha \frac{\partial}{\partial \theta_n} J(\theta_0, \ \ldots, \ \theta_n) \\
&\quad \theta_0 := temp\_0 \\
&\qquad \vdots \\
&\quad \theta_n := temp\_n \\
&\} \ \ while(\partial J \neq 0);
\end{aligned} \tag{4.4}
$$

which ensures simultaneous update of all the parameters in the cost function.

The mini-batch gradient descent is the type of gradient descent investigated in this project. Hence, all the "gradient descent" terms that follow will refer to mini-batch gradient descent.

4.2.2 Adaptive Moment Estimation (Adam Optimizer)

The Adam optimizer incorporates the operation of both the momentum optimizer and the RMSprop optimizer. Both the momentum optimizer and the RMSprop optimizer allow speeding up of the optimization process towards the minimal loss, by ignoring noises in the parameter updating process.

Adam optimizer significantly outperformed the other optimizers (including stochastic gradient descent, RMSprop, AdaGrad and AdaDelta optimizers) in training both the multilayer neural network and convolutional neural network models, using MINST and CIFAR-10 data. Adam optimizer is able to achieve a much lower training cost (training error) than the other optimizers. Adam has also markedly increased the optimization convergence speed (Kingma and Ba, 2015).

The performance of the multilayer neural network and convolutional neural network with different optimizers, published by Kingma and Ba (2015) is shown in Figure 4.1.

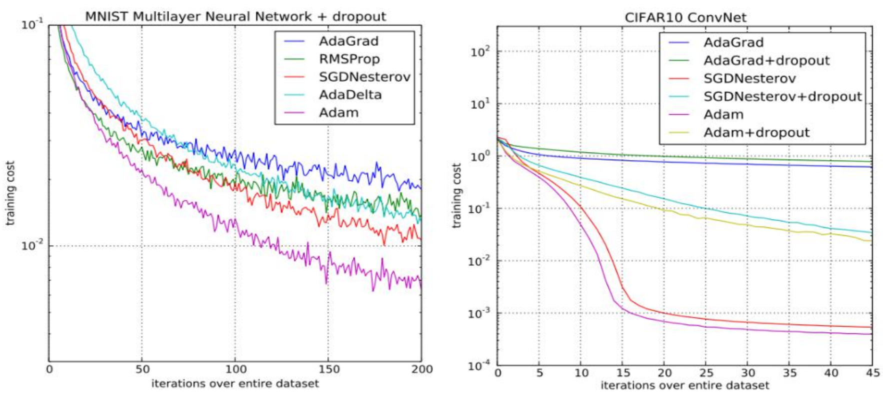

Figure 4.1: Performance of multilayer neural network (left) and convolutional neural network (right) working over MNIST and CIFAR-10 dataset respectively, using different optimizers (Kingma and Ba, 2015)

Adam optimizer works as follow:

$$
\begin{aligned}
& V_{dw} = 0,\ \ S_{dw} = 0,\ \ V_{db} = 0,\ \ S_{db} = 0 \\
& on\ \ training\ \ iteration\ \ t: \\
& \quad compute\ \ dw,\ db\ \ of\ \ the\ \ current\ \ batch \\
& \quad V_{dw} := \beta_m V_{dw} + (1-\beta_m)dw \ \ , \qquad V_{db} := \beta_m V_{db} + (1-\beta_m)db \\
& \quad S_{dw} := \beta_r S_{dw} + (1-\beta_r)dw^2 \ \ , \qquad S_{db} := \beta_r S_{db} + (1-\beta_r)db^2 \\
& \quad V_{dw}^{corrected} = \frac{V_{dw}}{1-\beta_m^t} \ \ , \qquad V_{db}^{corrected} = \frac{V_{db}}{1-\beta_m^t} \\
& \quad S_{dw}^{corrected} = \frac{S_{dw}}{1-\beta_r^t} \ \ , \qquad S_{db}^{corrected} = \frac{S_{db}}{1-\beta_r^t} \\
& \quad w := w - \alpha \frac{V_{dw}^{corrected}}{\sqrt{S_{dw}^{corrected}} + \varepsilon} \ \ , \qquad b := b - \alpha \frac{V_{db}^{corrected}}{\sqrt{S_{db}^{corrected}} + \varepsilon}
\end{aligned}
\tag{4.5}
$$

where

- *dw* denotes the change in parameter *w* (weight),
- *db* denotes the change in parameter *b* (bias),
- the terms V_{dw}, V_{db} are the momentum terms of the parameters *w* and *b* respectively,
- the terms S_{dw}, S_{db} are the RMSprop terms of the parameters *w* and *b* respectively,
- the terms β_m, β_r are the "friction" restriction (or the exponential decay rate) to the momentum and RMSprop respectively,
- α is the learning rate of the optimizer, and
- $\mathcal{E}$ is a parameter used to avoid division by zero.

The default values of the parameters in Adam optimizer tested to be good for machine learning tasks are α = 0.001, β_m = 0.9, β_r = 0.999, and $\mathcal{E}$ = 10^{-8}. These are also the values used in this project.

4.2.3 Comparison between Gradient Descent and Adam Optimizer

When the gradient descent optimizer is used, the model fails to learn from the EEG training dataset. The optimization process may have been trapped at a very early local minimum, or the deep model may have a cost function with extremely low gradient which has caused the gradient descent optimizer to learn too slowly.

On the other hand, when the Adam optimizer is used, the model has successfully learned and extracted the distinctive features between the two groups of EEG (before and after listening to music), enabling its classification accuracy to improve above 70% over the training iterations.

4.3 Effect of Activation Function

The purpose of applying nonlinear activation function to the neural network is to introduce non-linearity in the model, in order to model a non-linear representation of a complicated non-linear data domain. Without the nonlinear activation function, any arbitrary number of hidden layers would simply result in the formation of another linear representation. Hence, without non-linearity, the hidden layers in a model will not result in performance improvement.

A number of recent research (Clevert, Unterthiner, and Hochreiter, 2016; Ide and Kurita, 2017; Ramachandran, Zoph, and Le, 2018) showed that the proper selection of activation function has significant impact on the performance of deep learning models. Additional modifications made onto traditional activation functions such as the sigmoid and the ReLU functions have also resulted in improved performance (Ide and Kurita, 2017; Ramachandran, Zoph, and Le, 2018).

Figure 4.2 below shows the training log of the same model with the type of activation function as the only manipulated variable. Three different activation functions are examined: the exponential linear unit (ELU), rectified linear unit (ReLU), and sigmoid function.

ELU function has the formula

$$f(x)=\begin{Bmatrix} x \ , & x\geq 0 \\ e^{x}-1 \ , & x<0 \end{Bmatrix}, \quad -1<f<\infty \tag{4.6}$$

ReLU has the formula

$$f(x)=\begin{Bmatrix} x \ , & x\geq 0 \\ 0 \ , & x<0 \end{Bmatrix}, \quad 0\leq f<\infty \tag{4.7}$$

Sigmoid activation has the formula

$$f(x)=\frac{1}{1+e^{-x}}, \quad 0<f<1 \tag{4.8}$$

The sigmoid and ReLU activation functions both resulted in an increased classification accuracy, as compared to the model without activation function. ReLU activation function is the most suitable, among the tested functions, for the designed model to perform classification on the EEG data.

ELU activation function has negatively impacted the model's optimization, resulting in a performance worse than that without any activation function. However, the reason for ELU's negative impact is not clear.

4.4 Effect of Dropout Mechanism

One of the main challenges in the design of a machine learning model is the requirement for the model to perform with an almost equal accuracy on previously unobserved data (such as the test dataset), as on the training data set. This is a desired ability of the learning model, termed as generalization. The models that can generalize well are usually models with large capacity that are properly regulated.

The capacity of the model is defined as the ability of the learning model to fit or be modeled into a variety of complicated functions. Models with low capacity will not be able to recognize, memorize and learn from the huge pool of available features or parameters in the training data domain, resulting in underfitting where the model fails to approximate a fine/detailed representation of the training data set. Models with much higher capacity than the available effective parameters in the training data set will tend to overfit by memorizing all trivial characteristics of the training data set that might not be the true data-generating process. Such model is said to be overfitted to the training data set, and does not perform as well at the previously unobserved data (the validation or testing data).

Hence, for the design of deep learning training model, certain strategies are explicitly carried out for the purpose of reducing the test error, usually at the expense of increasing the training error.

Dropout mechanism is one of the regularization methods. In dropout method, a percentage of neurons (or computational nodes) of certain layers of the neural network is specified to be randomly blocked out during the training steps. Each training step will make a different combination of computational nodes available, instead of the full network, Hence, the model-under-training will not be able to rely too much on any selective few features propagated by certain computational nodes. Instead, every partial combination of the network will be more sufficiently trained, having their weights been updated more properly through backpropagation of error.

The best dropout rate among the examined is 40-50% dropout. The model with no dropout mechanism overfits the earlier. Extremely high dropout rate (such as 70% dropout) throttled the learning speed too much although overfitting is avoided.

4.5 Comparing the architecture and performance of pure FC-MLP models and various CNN models

A few different convolutional neural network (CNN) models are constructed and trained, in comparison with pure fully-connected-multilayer-perceptrons (FC-MLP) models without convolution mechanism. The performance of the CNN models and FC-MLP models at classifying the EEG dataset into 2 groups (before and after listening to music) are discussed in the following sub-section.

4.5.1 Performance of the CNN models and pure FC-MLP models

With the adoption of convolution mechanism, the CNN models are better at extracting the temporal relationship of the adjacent/successive sampling points and the spatial relationship across the EEG channels, contributing to higher EEG classification accuracy.

Using only the six frontal EEG channels, one of the CNN models with FC-MLP (classification accuracy of 71±1%) performs better than its counterpart without FC-MLP (accuracy of 65±3%). This shows that the additional hidden fully-connected layers of perceptrons have improved the classification accuracy of the CNN model by about 6%. This is probably due to the additional capability granted by the hidden layers to the CNN model to assume a more fine-tuned representation of the complicated EEG data domain.

Using the 1-path CNN model without FC-MLP to perform the classification based on all fourteen EEG channels has achieved 71±1% accuracy. While using the same 1-path CNN model without FC-MLP to perform the classification based on only six frontal channels has achieved a lower accuracy of 65±3%. This shows that the other EEG channels do also carry significant information, which can contribute to the EEG classification accuracy.

Apart from the above discussed, the pure FC-MLP models does not perform as good as the CNN models. In addition, the pure FC-MLP models' classification accuracy deteriorates with increasing depth (more hidden layers) of the model.

4.6 Which brain region's EEG changes more due to music listening

4.6.1 Comparing the degree of impact of short duration of music on the frontal lobes and the rest of the brain

The classification accuracy achieved using the temporal, parietal and occipital channels combined without the frontal channels is significantly lower than that achieved using six frontal channels. The model is able to classify frontal lobe EEG signals better than the signals from the other lobes.

This is probably because the short session of relaxing music listening has a greater impact on the frontal lobe than the other regions of the brain, causing the EEG generated by the frontal lobe to differ more significantly (before and after listening to music) than the EEG from the other regions.

4.6.2 Comparing the degree of impact of short duration of music on the left and right cerebral hemispheres

The CNN model trained and validated with the left hemisphere EEG signals has achieved significantly higher classification accuracy than the model trained and validated with the right hemisphere EEG.

This indicates that the short session of relaxing music has affected the left cerebral hemisphere more than it does to the right cerebral hemisphere. This finding is in contrary to our expectation that the right cerebral hemisphere, which is in charge of our emotion, should be affected more by the music than the left hemisphere.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

For the task of binary classification of EEG, one of the 14-channel CNN model has achieved the top validation accuracy of 75±1%. This performance is closely followed by another 6-frontal-channel CNN model, which has achieved validation accuracy of 71.5±2%.

This finding is significant as the models are operating on the EEG dataset that was shown by previous classical manual feature extraction methods to have no statistical significant difference.

5.1.1 Optimizer

Basic Gradient Descent Optimizer is not sufficient for training the deep learning models for the task of EEG data classification. Using basic gradient descent optimization algorithm to minimize the cost function, deep learning models have failed to learn from the EEG data, causing the validation accuracy to stay below 50%.

Adam Optimizer performs significantly better at training the deep learning model for EEG data classification, with the validation accuracy to reach up to and above 67±2%.

5.1.2 Activation Function

ReLU is the most suitable activation function for deep learning model for EEG classification, followed by the sigmoid function. The model with ELU activation function performs worse (with validation accuracy below 50%) than the model without any activation function.

5.1.3 Dropout Mechanism

The most suitable dropout rate is around 40% to 50%. Too low the dropout rate (0% to 30%) does not help much in preventing overfitting of the model to the training data. Too high the dropout rate (70%) will slow down the model learning speed excessively.

5.1.4 Effect of Convolution

Convolutional layers significantly improve the performance of the deep learning model for EEG classification, elevating the validation accuracy from below 64% up to above 75%.

5.1.5 Degree of impact of music on different brain regions

A short session of listening to relaxing music has greater degree of impact to the frontal region than the other regions of the brain, and also greater impact to the left cerebral hemisphere than the right, inferring from the discrepancy at the classification accuracy as discussed in Section 4.6.

5.2 Recommendations

5.2.1 Hardware

Future work on deep learning for EEG data classification should ideally be implemented on more powerful GPU with at least 12GB of graphic RAM.

5.2.2 Deep Learning Model Design

Variants of convolutional neural network such as residual network (ResNet), as well as other deep learning models such as recurrent neural network which is designed for processing time series data should be implemented for performing EEG data analysis.

More advanced techniques in modelling deep learning architecture and training should be adopted. For example, other model regularization techniques can be used in combination with dropout mechanism for potentially better results.

5.2.3 EEG Dataset

As the EEG dataset used in this project comprised of 2 groups of EEG signals (recorded before and after listening to a short session of music) which may indeed have no much difference in their data generation domain, other established EEG datasets that have shown distinct generalizable features can be used in the future work to first determine the performance of the deep learning model.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. and Zheng, X. (2016). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. [online]. Available at: https://arxiv.org/abs/1603.04467 [Accessed 15 Nov. 2017]

Behncke, J., Schirrmeister, R. T., Burgard, W. and Ball, T. (2017). The signature of robot action success in EEG signals of a human observer: Decoding and visualization using deep convolutional neural networks. [online]. *2018 6th International Conference on Brain-Computer Interface (BCI)*, GangWon, South Korea, January 15-17, 2018. IEEEXplore, pp 1-6. Available at: https://arxiv.org/ftp/arxiv/papers/1711/1711.06068.pdf [Accessed 4 Feb. 2018]

Clevert, D., Unterthiner, T. and Hochreiter, S. (2016). *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*. [online]. Available at: https://arxiv.org/abs/1511.07289 [Accessed 20 Nov. 2017]

Hajinoroozi, M., Mao, Z. and Huang, Y. (2015). Prediction of driver's drowsy and alert states from EEG signals with deep learning. In: *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. [online]. Cancun: IEEE Conference Publications, pp. 493–496. Available at: http://ieeexplore.ieee.org.libezp.utar.edu.my/document/7383844/ [Accessed 20 Jul. 2017]

Ide, H. and Kurita, T. (2017). Improvement of learning for CNN with ReLU activation by sparse regularization. [online]. *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, May 14-19, 2017. IEEEXplore, pp. 2684-2691. doi: 10.1109/IJCNN.2017.7966185 [Accessed 10 Dec. 2017]

Karn, U. (2016). An Intuitive Explanation of Convolutional Neural Networks. [online]. *the data science blog*. Available at: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/ [Accessed 1 Aug. 2017]

Kingma, D. P. and Ba, J. L. (2015). *Adam: A Method for Stochastic Optimization*. [online]. Available at: https://arxiv.org/abs/1412.6980 [Accessed 16 Aug. 2017]

Patnaik, L. M. and Manyam, O. K. (2008). Epileptic EEG detection using neural networks and post-classification. *Computer Methods and Programs in Biomedicine*. [online]. **91**(2), pp. 100–109. Available at: http://www.sciencedirect.com.libezp.utar.edu.my/science/article/pii/S0169260708000539 [Accessed 10 Jul. 2017]

Phneah, S. W. and Nisar, H. (2017). EEG-based alpha neurofeedback training for mood enhancement. [online]. *Australasian Physical & Engineering Sciences in Medicine*. **40**(2), pp. 325-336. Available at: https://link.springer.com/article/10.1007/s13246-017-0538-2 [Accessed 5 Apr. 2018]

Ramachandran, P,, Zoph, B. and Le, Q. V. (2018). Searching for Activation Functions. [online]. *6th International Conference on Learning Representations*, Vancouver Convention Center, Vancouver, BC, Canada, April 30 - May 3, 2018. Available at: https://openreview.net/forum?id=Hkuq2EkPf [Accessed 8 Apr. 2018]

Ren, Y. and Wu, Y. (2014). Convolutional Deep Belief Networks for Feature Extraction of EEG Signal. *2014 International Joint Conference on Neural Networks*. [online]. Beijing: IEEE Conference Publications, pp. . Available at: http://ieeexplore.ieee.org.libezp.utar.edu.my/document/6889383/ [Accessed 10 Jul. 2017]

Ruder, S. (2017). *An overview of gradient descent optimization algorithms*. [online]. Available at: https://arxiv.org/abs/1609.04747 [Accessed 25 Dec. 2017]

Satapathya, S. K., Dehuri, S. and Jagadev, A. K. (2016). EEG signal classification using PSO trained RBF neural network for epilepsy identification. [online]. *Informatics in Medicine Unlocked*, **6**(2017), pp. 1–11. Available at: http://www.sciencedirect.com/science/article/pii/S2352914816300387 [Accessed 5 Aug. 2017]

Schirrmeister, R. T., Gemein, L., Eggensperger, K., Hutter, F. and Ball, T. (2017). *Deep learning with convolutional neural networks for decoding and visualization of EEG pathology*. [online]. Available at: https://arxiv.org/abs/1708.08012 [Accessed 30 Mar. 2018]

Schirrmeister, R. T., Springenberg, J. T., Fiederera, L. D. J., Glasstetter, M., Eggensperger, K., Tangermann, M., Hutter, F., Burgard, W. and Ball, T. (2017). *Deep learning with convolutional neural networks for brain mapping and decoding of movement-related information from the human EEG.* [online]. Available at: https://arxiv.org/abs/1703.05051 [Accessed 4 Feb. 2018]

Siuly and Li, Y. (2014). A novel statistical algorithm for multiclass EEG signal classification. [online]. *Engineering Applications of Artificial Intelligence*, **34**(2014), pp. 154–167. Available at: http://www.sciencedirect.com.libezp.utar.edu.my/science/article/pii/S0952197614001092 [Accessed 6 Jul. 2017]

Subasi, A. and Ercelebi, E. (2005). Classification of EEG signals using neural network and logistic regression. [online]. *Computer Methods and Programs in Biomedicine*, **78**(2), pp. 87–99. Available at: http://www.sciencedirect.com/science/article/pii/S0169260705000246 [Accessed 10 Jul. 2017]

Supratak, A., Dong, H., Wu, C. And Guo, Y. (2017). DeepSleepNet: a Model for Automatic Sleep Stage Scoring based on Raw Single-Channel EEG. [online]. *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* **Volume PP**(99), pp.1–1. Available at: http://ieeexplore.ieee.org.libezp.utar.edu.my/document/7961240/ [Accessed 25 Jul. 2017]