Convolutional Neural Network Audio Classifier for Alarm Sound Detection

C. Ramesh Babu Durai, Kunj Yogesh Haria,G Sai Prashanth,

Abstract*artificial Neural Networks (ANN) has evolved through many stages in the last three decades with many researchers contributing in this challenging field. With the power of math complex problems can also be solved by ANNs. ANNs like Convolutional Neural Network (CNN), Deep Neural network, Generative Adversarial Network (GAN), Long Short Term Memory (LSTM) network, Recurrent Neural Network (RNN), Ordinary Differential Network etc., are playing promising roles in many MNCs and IT industries for their predictions and accuracy. In this paper, Convolutional Neural Network is used for prediction of Beep sounds in high noise levels. Based on Supervised Learning, the research is developed the best CNN architecture for Beep sound recognition in noisy situations. The proposed method gives better results with an accuracy of 96%. The prototype is tested with few architectures for the training and test data out of which a two layer CNN classifier predictions were the best.*

Keywords: CNN, Deep Neural Network , Adam Optimization, Sound processing, Backpropagation, Peak Detection.

I. INTRODUCTION

With the rapid development of technology and research, Artificial Intelligence has major plays in many industries like Search engines, Computer vision, Image retrieval, Business Analytics and a lot more. Artificial Intelligence brought out astonishing results like Self-Driving cars, Image Recognition, Speech processing, Cancer cells detection and Intuition and much more to come in the mere future. Among the popular machine learning algorithms Neural Networks are widely used my major data scientist and research engineers around the world. As of now majority of research works are based Supervised learning and Semi-supervised learning. In the upcoming years, it is targeted to produce high productivity using Reinforced learning and Unsupervised learning algorithms. Other popular algorithms used in machine learning include SVM (Support Vector Machines), KNN (K- Nearest Neighbors), Logistic Regressions, Decision Trees, Genetic Algorithm, Random Forest, Naïve Bayes etc. In this proposal it is focused on Supervised learning using Convolutional Deep Neural Network.

Revised Manuscript Received on August 30, 2019.

* Correspondence Author

C. Ramesh Babu Durai*, Department of EEE, Sri Sairam Engineering College, Chennai, India. Email : crbdurai.eee@sairam.edu.in.

Kunj Yogesh Haria, Masters in Electrical Engineering, N C University, Ralieigh. Email : kunj.haria@somaiya.edu

G Sai Prashanth, Department of ECE, Sri Sairam Engineering College, Chennai, Tamil Nadu, India. Email : donsai489@gmail.com.

II. DEEP NEURAL NETWORKS (DNN)

The mathematical theory of Artificial Neural Network was first proposed by Frank Rosenblatt in 1958.He assumed that the coordination of body activity and thoughts are the results of interactions among neurons in brain in beings. Neural networks commonly known as ANN (Artificial Neural Network) are the mathematical matrix operations consisting of input, hidden and output layers. It uses the concept of human neurons in brains that are gradually optimized over the course of time by interaction of humans with the environment. Human brain is estimated to have neurons in the order of billions. In the available firmware it is not possible to incorporate the power of such counts of perceptron. Researches are undertaken to improve hardware performance of Machine Learning algorithms especially ANNs. There are lot of DNN models in the industries that are mainly involved in cloud based image recognition, better search results, business analytics etc,. One of the example is Google's Deep Mind.

III. DATA PREPROCESSING

Supervised Learning is used for our classification model. The Training and Testing data consists of three hundred samples of noise and alarm (beep) sound and are labeled accordingly. One-Hot encoding is employed for output labels. In the case of alarm sound detection we have two classes as Noise, Beeps (Alarms). Neural networks are designed to deal with numbers not with strings or characters. So they must be encoded into numbers matching the output shape of the neural network. For e.g. if we have two classes as dogs and cats, labels for the network must be (0, 1) for cats and [1,0] for dogs and so on. As long as the model is trained on more data the accuracy gets better and it is tested experimentally.

The sampled sound is converted into a Mel spectrogram Image of data usually as Rectangular array of data. The mel-frequency scale is a quasi-logarithmic spacing roughly resembling the resolution of the human auditory system. The Matrix data has various patterns for various sounds like human voices, Air-conditioners, Fans, Environmental noises, Alarms etc., Normal Spectrogram is different from a Melspectrogram data. The mel-scale is closely related to biological hearing and has proven to be more successful in speech processing, segregation, recognition.

It produces unique patterns and textures in sound data by which Convolutional Neural Networks are good at detecting patterns in 2D or 3D matrix data. Lot of other sound features like Chroma, Pitch, Tonnetz, Mel Frequency Cepstral Coefficient(MCFC) may also be used for feature extraction for complex multi-feature multi-class classification problems.

A. MELSPECTROGRAM CALCULATION

In a normal spectrogram computation, the output spectrum is based on time series information of the frequency bins of each window of raw audio data.

Fast Fourier Transform is applied to each window of audio data of defined length and the audio frequencies are extracted mathematically and plotted to the vertical axis with value in the horizontal axis as time intervals which contribute to the Spectrogram. Spectrograms play an important role in frequency analysis in audio analysis.The figure 1 shows the plot of Mel spectrogram versus Normal Spectrogram on the right side.

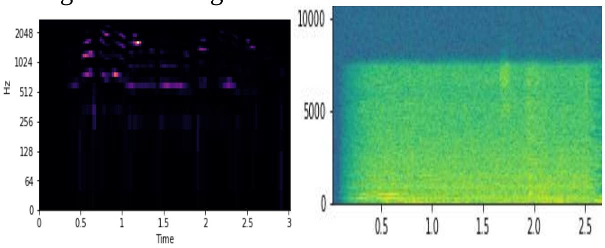
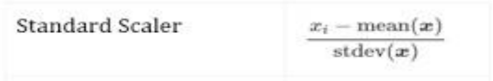


Fig.1 Plot of Mel spectrogram versus Normal Spectrogram

If a time-series input is provided, then its magnitude spectrogram S is first computed, and then mapped onto the mel-scale by mel_f.dot(S**power). The Mel spectrogram is the input data to the model. Python language provides inbuilt functions for Mel spectrogram calculations in Librosa library package. The data is passed through Standard Scalar where each data points are scaled between two points such that it well suits for the Neural Network.



Standard Scaler $\frac{x_i - \text{mean}(x)}{\text{stdev}(x)}$

B. DATA AUGMENTATION

Data augmentation plays a major role in improving the accuracy of the model. The prototype model is trained on a balanced data. When the data is unbalanced it is better practice to analyze the metrics using “f1score”. The audio data is augmented using the following methods.

i) Time Stretching
ii) Pitch Shifting
iii) Rolling series data

The 300 sample dataset has increased to a number of 900 sample dataset by applying the above methods in each sample dataset. As long as the training data has many and diverse datasets, the model becomes capable of learning and classifying the required outputs in noisy environments. The training dataset includes input data coupled with respective output labels. Some better practices include having diverse data features instead of repeating the same data and its feature in every iterations.The training is done is a powerful machine like personal Computer and the model trained and optimized can be deployed into the dedicated environments for example raspberry pi, voice bonnet and other embedded applications to deal with real time noisy data.

IV. CNN MODEL DEVELOPMENT

The model can be subdivided into two layers of Convolutional layers with pooling layers, a hidden perceptron layer (optionally be increased based on training data) and an output layer. The first Convolutional layer faces the input data, applies filters to extract features i.e. curves, lines, edges etc., The output shape of the first Convolutional Layer is defined by,

$$Out1 = [(N+2p-f)/s + 1]*[(N+2p-f)/s + 1]$$

for the input shape of order NxN and filter shape of order fxf. The notations ‘p’ and ‘s’ represents padding and strides in the Convolution operation and they are optional chosen during optimization and training. It is followed by a MaxPooling layer for further extraction. Further, the following Convolutional layer involves the same operation and MaxPooling layer. The output shape consists of 2D array of data. The 2D matrix is flattened into 1D array of data for feeding it into a Deep Neural network through synapses, which has perceptron or neuron units having ReLU(Rectified Linear Unit) activations f(x) = max(0,x). The output of this layer is passed through the synapses layer and other activation layer having “Sigmoid” activation. The two perceptron layers are densely connected with output of neuron. The output is categorized as either alarm or noise sounds based on the output data i.e. maximum(out1,out2). This is a basic primitive model for predicting beep sound patterns. For Multi-level audio classification high level CNN classifiers with huge datasets must be used. Model is tested under different testing conditions in real time and encountered a better accuracy results. Here is the CNN architecture for the prototype.

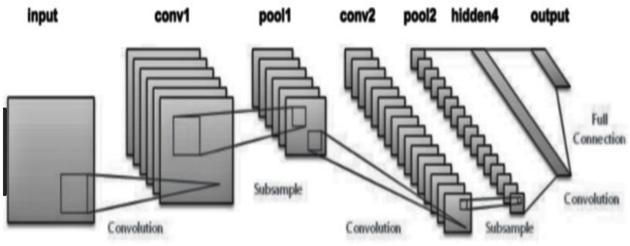


Fig. 2 CNN Architecture

V. OPTIMIZATION OF THE MODEL

The Backpropagation is deployed to optimize hyper parameters of the model. Backpropagation uses differential calculus to update weights in the model depending on the error value for each input i.e. how far away the predictions are from the actual value. The weights in the network are randomly initialized before training. During training process the error in weight values are gradually updated by Backpropagation process using chain rule.

If we iteratively reduce each weight's error, eventually we'll have a series of weights in the network that produce good predictions. Giants like Google trained their Deep learning models with some millions of data. The rate at which model learns depends on the learning rate. Higher the learning rate, lower the accuracy and vice versa. Each layer weights are updated based on the error obtained in the next layer during forward propagation of the input. The model gets optimized over the time during training and for each batches of training data, weights are updated

so that the model fits to the input and output labels. Backpropagation is done by applying "Chain rule" to the model.

Given a forward propagation function:

$$f(x) = A(B(C(x)))$$

Chain rule can be used to find f(x) with respect to x as,

$$f'(x) = f'(A) \cdot A'(B) \cdot B'(C) \cdot C'(x)$$

Assuming B(C(x)) to be a constant B and differentiated normally with respect to B. This technique is applied throughout the network and weights are modified or updated based on the cost function for the purpose of making good predictions as the input data propagates through the model. Adaptive Moment Estimation (**Adam**) is the state of the art optimization algorithm that integrates the ideas from RMSProp and Momentum. It is preferred comparing with the classical stochastic gradient descent procedure. It estimates adaptive learning rates for each hyper parameter as follows.

- First, the exponentially weighted average of past gradients is computed (vdW).
- Second, the exponentially weighted average of the squares of past gradients is computed (sdW).
- Third, these averages have a bias towards zero and to counteract this a bias correction is applied ($v^{corrected}_{dW}$, $s^{corrected}_{dW}$).
- Finally, the hyper parameters are updated using the data from the averages computed.

$$v_{dW} = \beta_1 v_{dW} + (1-\beta_1)\frac{\partial \mathcal{J}}{\partial W}$$
$$s_{dW} = \beta_2 s_{dW} + (1-\beta_2)(\frac{\partial \mathcal{J}}{\partial W})^2$$
$$v^{corrected}_{dW} = \frac{v_{dW}}{1-(\beta_1)^t}$$
$$s^{corrected}_{dW} = \frac{s_{dW}}{1-(\beta_1)^t}$$
$$W = W - \alpha\frac{v^{corrected}_{dW}}{\sqrt{s^{corrected}_{dW}}+\varepsilon}$$

The concept that the model is not able to classify input in test data but be able to classify input in train data is called Overfitting. The model is unable to generalize and accurately predict the output. Overfitting of the model can be avoided by having diverse data in training input by Data Augmentation, adding dropout and avoiding too much hidden layers in the model. Opposite to this is Underfitting. In this case, the model is unable predict desired output even in the training data itself. This can be reduced by reducing the dropout rate in the model, increasing hidden layers, adding more features to the input data. Fine-tuning is done by changing the trainable parameters of the model and training it accordingly to obtain the best results. Better predictions can be obtained in real time by having training data similar as that of real time data. When the number of classes for classification increases i.e. more features, complexity of the network should be arbitrary high and vice versa. Adding too much parameter (layers) for small class predictions can result in bad predictions as some weights in the network may not be optimized for minimal features. The detailed documentation is available in Python keras library for Adam Optimization of the model. Expected accuracy may be obtained by training the CNN model with new data every time (fine-tuning). The rest of the prototype has sound processing for peak sound detection in beep sounds for classifying whether the sound is abnormal or normal sound. The first part of the prototype is the CNN and the second part is Sound Processing .In our use case, again classifying Hospital instrumental beep sounds based on the interval between the peaks for abnormal and normal beep sounds. A heuristic method of peak detection is used to detect peaks in alarm signal. A sample n at a time is selected as peak if it satisfies the following conditions.

x[n] == maximum(x[n - pre_max (upto) n + post_max]),
x[n] >= mean(x[n - pre_avg (upto) n + post_avg]) + delta,
n – previous_n > wait.
Where, x -> array [shape=(n,)]
input signal to peak picks from
pre_max -> integer >= 0
no.of samples before n over which max is computed
post_max -> integer >= 1
no.of samples after n over which max is computed
pre_avg -> integer >= 0
no.of samples before n over which mean is computed
post_avg -> integer >= 1 [scalar]
of samples after n over which mean is computed
delta -> threshold offset for mean

This method is used to detect peaks in beep sounds. Detailed explanation is depicted in librosa.util.peak_pick function in python. This is fast and the parameters can be varied according to the environments. If the estimated number of peaks is more than the threshold for a particular time interval, it is considered as an abnormal alarm and an indication is set by general means.

V. EXPERIMENTAL INVESTIGATION

A test alarm sound mal spectrum after standard scaling is shown in fig.3.

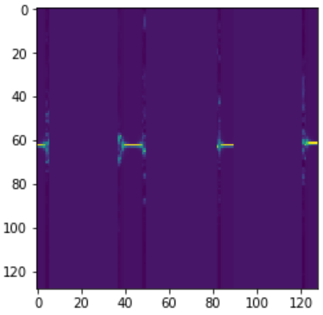


Fig. 3 Test alarm sound mal spectrum

Sound is recorded and sampled at same frequency as that of training data for a period of three seconds and converted into a linear array of data. The sampling rate at test conditions was 22100Hz which is the recommended sampling rate for raspberry pi.

The linear array of data is transformed into a Mel spectrogram matrix. It is set on its way towards the model. The matrix data must be reshaped as per the model's input shape. The Model classified the data, made some predictions and outputted probabilistic values. The maximum probability value is chosen as output prediction for that particular input. The data flow path i.e. forward propagation of input data for a particular class is defined during training.

VI. RESULTS AND DISCUSSION

The prototype acquired an accuracy of 96.5% as average in classifying Alarm sounds with environmental sounds. Without data augmentation the prototype encountered an accuracy of around 76%. In real time noisy test the model failed to predict some of test alarm sounds. After augmentation the accuracy is raised above 90%.

VII. SUMMARY

In this work, in order to predict Alarm sounds the implementation of a novel convolutional neural network has been proposed. Melspectrogram is considered as the input feature matrix for the model. The computation speed was also good and real time predictions were accurate for hospital situations. High accuracy is obtained only by data augmentation process as the model fits and gets optimized for the input features. For multi-class predictions various features like Chroma, Pitch, Tonnetz, Mel Frequency Cepstral Coefficients (MFCCs) may be used.

VIII. CONCLUSION

The proposed model predicts the alarms well and fine-tuned for different kind of alarm and beep sounds. Convolutional Neural Network detects pattern both is image and audio matrix data. Feature Engineering plays an important role in optimizing the weights in the network and suiting the role of predictive analyzer algorithm.

REFERENCES

1. Audio Classification Method Based on Machine Learning, Published in: 2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS).
2. L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone, "Classification and Regression Trees," Wadsworth, Belmont, CA, 1984.
3. Audio Recapture Detection With Convolutional Neural Networks, Published in: IEEE Transactions on Multimedia (Volume: 18 , Issue: 8 , Aug. 2016).
4. Website:https://www.kdnuggets.com/2017/12/audio-classifier-deep-neural-networks.html, blog post.
5. Shoji Kido ; Yasusi Hirano ; Noriaki Hashimoto. "Detection and classification of lung abnormalities by use of convolutional neural network (CNN) and regions with CNN features (R-CNN)". Published in: 2018 International Workshop on Advanced Image Technology (IWAIT).
6. Feng Rong. "Audio Classification Method Based on Machine Learning", Published in: 2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS).
7. Hertz, J., Krogh, A. and Palmer, R.G. "Introduction to the Theory of Neural Computation," Addison Wesley, 2001.
8. Begüm Demir ; Sarp Ertürk "Improving SVM classification accuracy using a hierarchical approach for hyperspectral images", Published in: 2009 16th IEEE International Conference on Image Processing (ICIP).
9. "Comparison of Classification Techniques used in Machine Learning as Applied on Vocational Guidance Data ",Halil Ibrahim Bulbul ; Özkan Unsal. Published in: 2011 10th International Conference on Machine Learning and Applications and Workshops.
10. V. Valev ; P. Radeva. "A method of solving pattern or image recognition problems by learning Boolean formulas", Published in: Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems.

AUTHORS PROFILE

C Ramesh Babu Durai is a Professor in Electrical and Electronics Engineering at Sri Sai Ram Engineering College, Chennai, Tamilnadu . He graduated from B.E. degree with Madurai Kamaraj University and M.E. and Ph.D. with Anna University,
Chennai, Tamil Nadu, India. His area of interest are. He has more than 22 years

Chennai, Tamil Nadu, India.

Kunj Yogesh Haria is doing post graduate in Calfornia University in the field Power Electronics. He graduated from B.Tech. degree with K.J. Somaiya College of Engineering, Mumbai. His area of interest are Power Electronics, Electrical drives and Control Systems.

Sai Prasanth G is doing Under graduate programme in Sri Sai Ram Engineering College in the field Electronics and Communication Engineering, Chennai, Tamil Nadu. His area of interest are Nano Physics, Electronics, Electromagnetic field and Signals and Systems.
Signal Processing , Image Processing , Electromagnetic
Field and Control Systems