IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 5, Ver. III (Sep.- Oct. 2017), PP 11-15
www.iosrjournals.org

Acoustic Event Detection Using T-DSN With Reverse Automatic Differentiation

*Rahna K.M , Baby C.J

Department of Computer science and Engineering Royal College of Engineering And Technology
Thrissur, India
Department of Computer science and Engineering Royal College of Engineering And Technology
Thrissur, India
Corresponding Author: Rahna K.M

Abstract: *The Tensor Deep Stacking Neural Network (T-DSN) enhance the conventional Deep Stacking Network by replacing one or more of its layers with a tensor layer, in which each input vector is projected into two nonlinear subspaces, and a tensor layer, in which two subspace projections interact with each other and jointly predict the next layer in the deep architecture. In addition, we implement a approach to find Derivatives, mostly in the form of gradients and Hessians. Automatic differentiation (AD) is a technique for calculating derivatives of numeric functions expressed as computer programs efficiently and accurately. Integrating AD with Deep Stacking Neural Network enables dynamic computational graph which increases the computational speed dramatically.*

Keywords: *Automatic Event Detection;Tensor Deep Stacking Network; Tensor;Gradient Methods*

Date of Submission: 13-09-2017 Date of acceptance: 02-10-2017

I. Introduction (Heading 1)

The audio and visual senses combine to give correlative information about the world. Our visual system gives us precise information about a small area of focus whereas auditory system provides generic information from all around, alerting us to things outside our peripheral vision. Audio information retrieval has been a popular research subject over the last decades and being a sub-field of this area, acoustic event classification has a considerable amount of share in the research. For many years, speech recognition technology has been dominated by a “shallow” architecture using many Gaussians mixture models associated with HMM states to identify events in acoustic event detection[1][2].

Deep Learning is a new area of Machine Learning research, which has been introduced with the objective of moving Machine Learning closer to one of its original goals: Artificial Intelligence. Deep Learning is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text[NN]. Deep learning refers to a class of machine learning techniques, where many layers of information processing stages in hierarchical architectures are exploited for pattern classification and for feature or representation learning. It is in the intersections among the research areas of neural network, graphical modeling, optimization, pattern recognition, and signal processing. Three important reasons for the popularity of deep learning today are drastically increased chip processing abilities (e.g., GPU units), the significantly lowered cost of computing hardware, and recent advances in machine learning and signal/information processing research. The essence of deep learning is to compute hierarchical features or representations of the observational data, where the higher-level features or factors are defined from lower-level ones.

In this paper, acoustic event classification using deep neural networks is investigated. A deep architecture Tensor Deep Stacking Network (T-DSN) with Graphical Process Unit (GPU) parallelization is presented in this system. The T-DSN comprise of multiple, stacked blocks, where each block contains a outer product of two hidden layers to the output layer, using a weight tensor to integrate higher-order statistics of the hidden binary features. T-DSNs used to classify the sounds into screams, shouts and other categories. For GPU parallelization Automatic Differentiation is used.

II. Deep Stacking Network

DSN is a discriminative approach of deep learning which can model arbitrarily complex posterior distributions[3][4]. Deep learning systems based on discriminative neural networks often work better when the input data is minimally preprocessed. It is hoped that the same applies as the output loss functions are more directly related to the final objective that the overall system aims to optimize, not a surrogate loss that is correlated with the overall aim of the system. As a side benefit, end-to-end training is simpler as there are no additional complexities arising due to system integration issues.

While the DNN just reviewed has been shown to be extremely powerful in connection with performing recognition and classification tasks including speech recognition and image classification, training a DBN has proven to be more difficult computationally. In particular, conventional techniques for training DNN at the fine tuning phase involve the utilization of a stochastic gradient descent learning algorithm, which is extremely difficult to parallelize across machines. This makes learning at large scale practically impossible. Deep Stacking, which attacks the learning scalability problem[4][5] .

The central idea of DSN design relates to the concept of stacking, where simple modules of functions or classifiers are composed first and then they are “stacked” on top of each other in order to learn complex functions or classifiers[6]. Various ways of implementing stacking operations have been developed in the past, typically making use of supervised information in the simple modules. The new features for the stacked classifier at a higher level of the stacking architecture often come from concatenation of the classifier output of a lower module and the raw input features.

*A. **Tensor Deep Stacking Networks***

The Tensor Deep Stacking Network (T-DSN) is an extension of DSN architecture. Tensor is a n-dimensional matrix, can run on GPUs. Rather than rely upon a single hidden layer in each block, it uses two parallel hidden representations in order to incorporate second order information from the non-linearly transformed input data into the model. This has the effect of moving more of the parameters within a block from the lower layer, the optimization of which is non-convex, to the upper layer, which has a convex, closed form solution, which in turn has the effect of making each layer more powerful. Hence the tensor consist of values from both hidden data along with its outer product.

III. Acoustic Event Detection

Sound event detection aims at processing the continuous acoustic signal and converting it into symbolic descriptions of the corresponding sound events present at the auditory scene. Input audio undergo preprocessing.

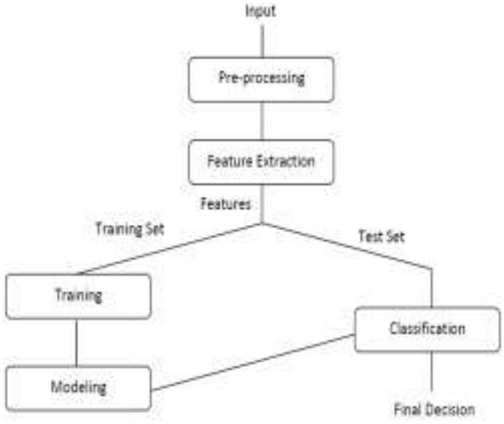

Fig. 1. System architecture.

Fig. 2.

*B. **Pre-processing***

Preprocessing involves taking a sound from the environment and loading it into a computer. A computer represents sounds in a digital format, which means that the analog signal produced by a microphone has to be converted into a digital format via sampling and quantization techniques. Sampling is a process of taking values of the wave at regular interval of time. By sampling smooth curve from the measurements is replaced by a finite set of numbers. Each pulse amplitude is then rounded to one of a finite number of levels, called quantization. Quantization and sampling is an encoding process that converts the analog waveform in to a binary representation.

*C. **Feature Extraction***

The important attributes of the data should be selected in such a way that, those would contain enough information to properly represent the similarities between the inter-class observations and variations between the intra-class observations. It is important to capture the variation of the frequency content over time, in order to fully characterize the sound. Mel Frequency Cepstral Coefficients(MFCC) features represent the frequency content of the sound at a particular instance in time, and therefore need to be combined with complex recognizers , which can model the temporal variation of these stationary features.

D. ***Classification***

The next step in the system is to recognize scream, shout, conversation and noise based on the extracted features. This task requires classifier training with a set of audios with particular events. Three principal issues in classification task are: choosing good feature set, efficient machine learning technique and diverse database for training. Feature set should be composed of features that are discriminative and characteristic for particular expression. Here use Deep Stacking Networks (DSN) intend to reproduce the mechanism with which mimics the human brain processes information. Importantly, higher-order hidden feature interactions are enabled in TDSN via the outer product construction for the large, implicit hidden layer. After converting input data to right format and divide it in batches.

Deep learning models operate in two modes: they either compute a prediction (or distribution over predictions) given an input, or, at training time when supervision is available, they compute the derivatives of a prediction error “loss” with respect to model parameters which are used to minimize subsequent errors on similar inputs using some variant of gradient descent. Since implementing a model requires both implementing the code that executes the model predictions as well as the code that carries out gradient computation and learning, model development is a nontrivial engineering challenge. The difficulty of this challenge can be reduced by using computational graphs which simplify implementation of neural network computations. It also have ability to express composite computations of a task-specific prediction and error that are then differentiated automatically to obtain the gradients needed to drive learning algorithms. This automatic differentiation is arguably most important labor-saving feature since changes to the function that computes the loss for a training input will require a corresponding change in the computation of its derivative. The flexible architecture allows deploy computation to one or more CPUs or GPUs. Generation of computation graphs either be done in statically or dynamically.

E. ***Static Vs Dynamic Computation Graph***

In the static declaration strategy defines a computation graph and then examples are fed into an engine that executes this computation and computes its derivatives[7][8]. Only once the graph is formed and then the data is fed to the same graph. Since the computation graph has a different shape and size for every input they are difficult to batch and any pre-defined static graph is likely excessive, wasting computation, or inexpressive.
In contrast to the two-step process of definition and execution used by the static declaration paradigm, the dynamic declaration model takes a single-step approach. In dynamic declaration strategy, there are no separate steps for definition and execution: the necessary computation graph is created, on the fly, as the loss calculation is executed, and a new graph is created for each training instance[7]. This requires very lightweight graph construction. Dynamic declaration thus facilitates the implementation of more complicated network architectures.

F. ***Reverse Automatic Differentiation***

Automatic differentiation (AD) is a technique for calculating derivatives of numeric functions represented as computer programs efficiently and accurately, used in fields such as computational fluid dynamics, nuclear engineering, and atmospheric sciences[9]. Automatic Differentiation (AD), which works by systematically applying the chain rule of differential calculus at the elementary operator level. Two approaches are: Forward mode Differentiation and Reverse mode Differentiation.

Forward mode differentiation starts at an input to the graph and moves towards the end. At every node, it sums up all the paths feeding in. Each of those paths equate one way in which the input affects that node. By adding them up, a total way in which the node is affected by the input, it’s derivative. It tracks down how one input affect every node. Reverse mode differentiation starts at an output of the graph and move towards the beginning. At each node it merges all paths which originated at that node. It tracks down how every node affects one output. As the reverse mode gives the derivative of with respect to all nodes in one swoop, hence it is more popular.

Reverse automatic differentiation is a generalization of back propagation. Back propagation models learning as gradient descent in neural network weight space, looking for the minimum of an error function, accomplished by the backwards propagation of the error values at the output utilizing the chain rule to compute the gradient of the error with respect to each weight. When a gradient needs to be calculated, the input is forward propagated through the computation graph with each node remembering its input. The graph is then traversed in reverse, calling the ”backward” method of each node with the gradient as its argument. Training of neural networks is an optimization problem with respect to a set of weights, via method gradient descent or stochastic gradient descent. As the highly successful back-propagation algorithm is only a specialized version, Reverse Mode Automatic Differentiation: by applying the reverse mode to any algorithm evaluating a network's error as a function of its weights, can readily compute the partial derivatives needed for performing weight updates.

To make deep learning techniques scalable to very large training data and optimization, GPU parallelizing can be achieved through on the fly dynamic computational graph batching.

IV. Result And Discussion

Deep learning is an emerging technology. T-DSN outperforms the DSN in terms of feature hierarchy. T-DSN along with reverse automatic differentiation builds a computational graph and execute the graph dynamically which gives result instantly. Dynamic computational graph with reverse automatic differentiation divide the building of graph and evaluation among GPUs. Transactions among GPUs done through tensors plays a crucial role for the performance.

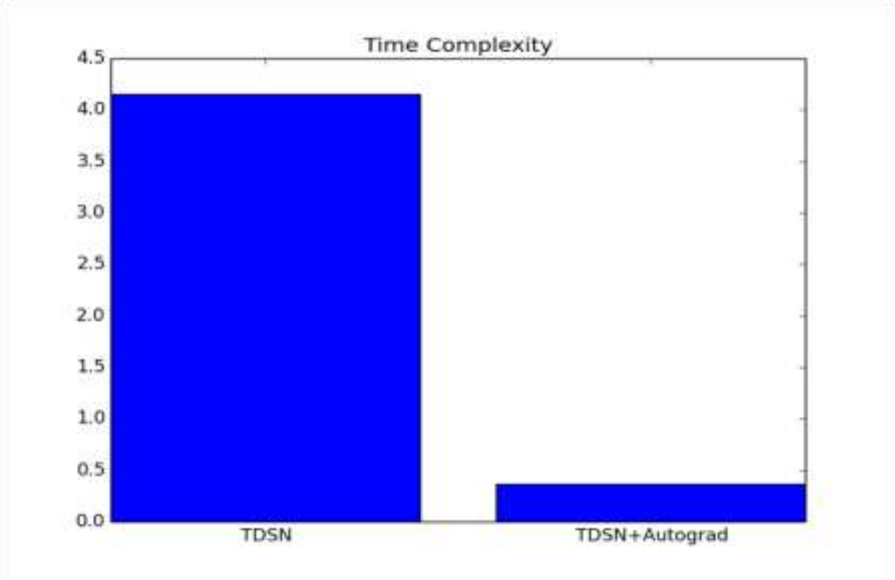

Fig. 3.comparison graph

Dynamic approach works faster than the static approaches used so far. The time-complexity graph shows the difference between the DSN with static computational graph versus T-DSN with dynamic computational graph. With a more powerful learning procedure, better recognition performance can be achieved.

V. Conclusion And Future Scope

This analysis shows that the T-DSN learns multiple views of the data. These different views capture a occurrences belonging to different classes at the same time in the input ,that would occur by random chance. Effective and scalable parallel algorithms are critical for training deep models with very large data. The common practice nowadays is to use graphical processing units (GPUs) to speed up the learning process which is also devised in the thesis efficiently. The main advantage of deep learning systems based on discriminative neural networks often work better when the input data is minimally pre-processed. As a side benefit, end-to-end training is simpler as there are no additional complexities arising due to system integration issues. The success of deep learning in unsupervised learning has not been determined as much as for supervised learning. The major motivation of deep learning lie right in unsupervised learning for automatically discovering data representations.

References

[1] Pierre L., David S., Charles T., Laurent G.,"Deep Neural Networks for Automatic detection of scream and shouted speech in subway trains",IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),Shanghai,pp. 6460-6464, 2016 .

[2] G. Rahna K M and Baby C J,"A Survey on Scream Detection Methods*"*,International Conference on Advanced Computing and Communication Systems (ICACCS), pp.1948-1952, 2017.

[3] D. Yu ; L. Deng : Deep learning and its applications to signal and information processing. *IEEE Signal Process. Mag*., 28 (2011), 145–154.

[4] L. Deng,"Three Classes of Deep Learning Architectures and their Applications." APSIPATrans. Signal Inf. Process. 2014,

[5] Hutchinson, B., Deng, L., and Yu, D, “Tensor deep stacking networks,” IEEE Trans. Pattern Analysis and Machine Intelligence, 2013.

[6] Wolpert, D, “Stacked Generalization,” Neural Networks, 5(2), pp 241-259, 1992.

[7] Graham N.,Yoav G.,Chris D., "On-the-fly Operation Batching in Dynamic Computation Graphs,"Neural Information Processing System ,2017.

[8] Moshe Looks, Marcello Herreshoff, DeLesley Hutchins, and Peter Norvig.,"Deep Learning with Dynamic Computation Graphs", International Conference on Learning Representations (ICLR),2017.

[9] Baydin, Atilim Gunes, and Barak A. Pearlmutter. "Automatic Differentiation of Algorithms for Machine Learning." arXiv digital library pp 1404-7456 ,2014.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Rahna K.M . “Acoustic Event Detection Using T-DSN With Reverse Automatic Differentiation.” IOSR Journal of Computer Engineering (IOSR-JCE) , vol. 19, no. 5, 2017, pp. 11–15.