Individual Report #1

Student: Rahul Rachakonda – 0889474

Supervisor: Dr. Yimin Yang

Features Combined From Hundreds of Midlayers:Hirerachical Networks with Subnetwork Nodes

ABSTRACT:

In this report I am going to explain about the steps we have in the our project and my contribution in that and also the results of testing and training and the problems we have faced during the development of project.

Keywords: Classification, Feature extraction, Feature combination, Neural Networks**.**

1. Introduction

The past few years have witnessed rise of Machine learning. Deep learning which is a branch of machine learning also witnessed huge rise of huge performance increase, This is the field where artificial neural networks, algorithms designed are inspired by the human brain, They learn from large amounts of data. Similar to how we learn from the experience, the deep learning algorithm would perform a task repeatedly, each time tweaking it a little to improve the outcome. In the recent times we have seen various Deep Learning models such as AlexNet , AmoebaNet , RNN (Recurrent neural network) ,these ,models have advanced object recognition by discovering the hidden structures in high dimensional data, the only restrictions with these models is that they are iterative model of learning and due to this particular reason the training time is quite slower , further more because of the Back propaganda used in this kind of iterative learning the model suffers from the false local Minima and vanishing gradient problems and the change in learning rate has direct consequences with the accuracy. So in this paper we will be discussing.

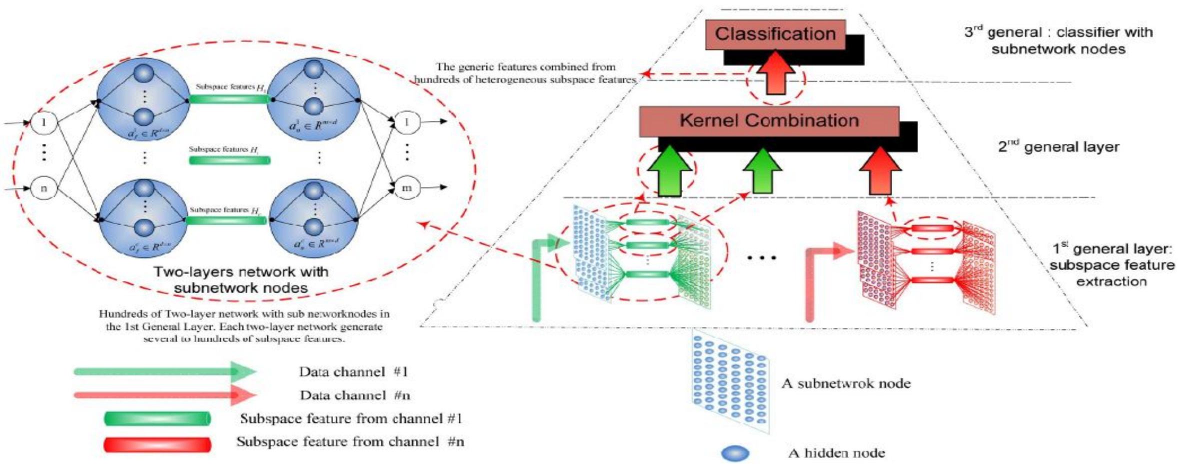
The parameters we consider in evaluating neural networks are mainly computational cost, learning speed, accuracy etc. Here we will be taking consideration of these factors while evaluating the neural network. Our project has two modules:

1. Implementing a feed-forward hierarchical three layer architecture with subnetwork nodes.

2. Patching OS-ELM algorithm into the above architecture.

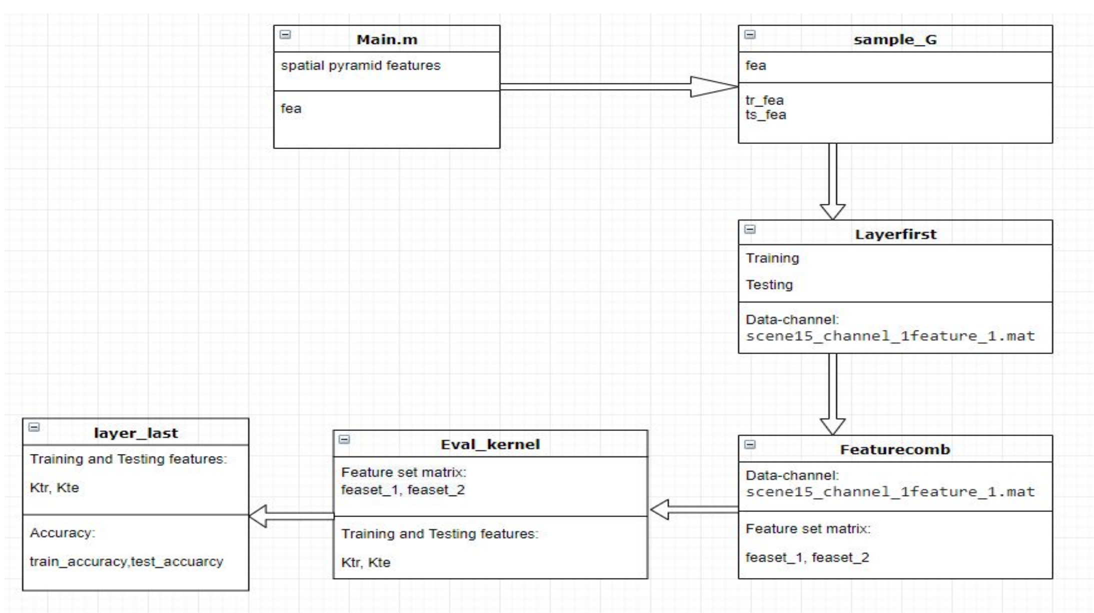
2.Architecture and Overview:

1. Hierarchical architecture with subnetwork nodes:



The above architecture has three layers, First layer has subspace feature extraction and second layer has feature combination and third layer has classifier applied on the combined features.

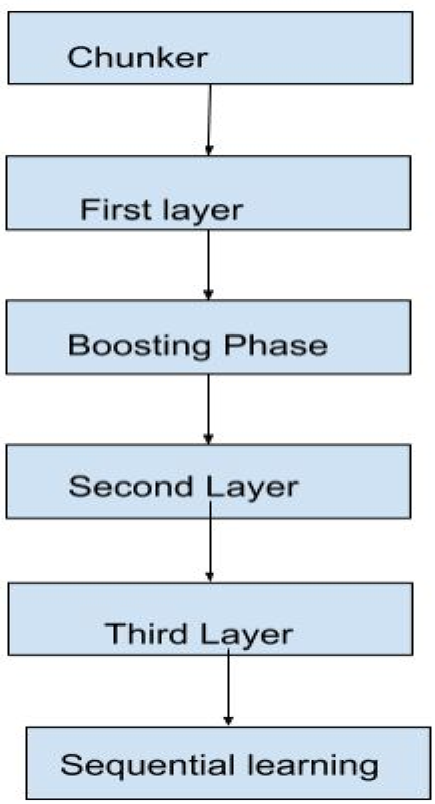
Flow chart:



The above flowchart represents the code flow and the data is manipulated in the three layers. As we can see the first layer(main.m and sample_G files) we have “Spatialpyramid.mat” file which contains features extracted from Scene15 Image dataset, From this file we will split the data into training and testing set(tr_fea and ts_fea) and send them to First layer. In the first layer we will do subspace feature extraction in eight steps and save them in a data channel(scene15_channel_1feature1.mat) as shown above. So we will repeat the first layer steps to extract many number of subspace features(H1, H2…..,Hn) and now we will send these features to second layer (Featurecomb and Eval_kernel) and combine the above features obtained from the first layer. For this project we are using “Linear Kernel” for combining the features. Now the combined features obtained from second layer(h_f) are sent to the last layer. In the last layer we will be applying classifier and calculating the error and accuracy.

2.OS-ELM:

Online sequential extreme learning machine(OS-ELM) can learn the data one-by-one or chunk-by-chunk with the fixed or varying chunk size. In this project we will use this OS-ELM technique of sending data chunk-by-chunk to the above hierarchical network and evaluate the performance. I will briefly discuss the algorithm and steps involved in it and how we will be patching it in the hierarchical network. Following represents the how the patching of os-elm done in heirarchical network.



3. Contribution:

Following are the contributions I have done for this project:

1. Writing and editing final report in latex IEEE format

1. Preprocessing and splitting into testing and training sets.

2. Writing code for the third layer which contains Classifier and error rate classification.

3. Calculating the training and testing accuracy.

4. Testing the code on before combing the OS-ELM to the above architecture and after combining.

5. Helped in combining the OS-ELM algorithm into the hierarchical architecture.

4. Results:

Index	*Method*	*Dataset*	*Accuracy*	*Time-Taken*
1	Hierarchical Method[a]	Scene15	97.80	46s
2	OS-ELM Combined Method[a]	Scene15	87.9	27s
3	Hierarchical Method[a]	Caltech101	80.01	270s
4	OS-ELM Combined Method[a]	Caltech101	50.20	290s
5	Hierarchical Method[a]	Caltech101(HMP+Spatial)	81.20	290s
6	OS-ELM Combined Method[a]	Caltech101(HMP+Spatial)	49.20	290s

[a]

5. Conclusion:

In this paper we present a hierarchical network with subnetwork in it and later combined OS-ELM approach to it evaluated the performance. The problem Is approached in two different ways: 1. Heterogeneous features extracted from two layers are combined and sent to the third layer and applied a classifier to evaluate the performance. 2) OS-ELM algorithm fused into architecture as Boosting phase attaching to first layer and using the Moore's-Penrose pseudo inverse calculated output in the third layer. Comparing the results of hierarchical networks to other state-of-art methods it gives a better performance. This method can be used as feature extractor(First layer) and a classifier(last layer) and also gives better performance.

We have tried to extend the work of the above hierarchical network to see how it performs for sequential learning. So instead of training all samples at once as mentioned in hierarchical architecture we tried to give the sample batch-by-batch for learning. As we can see eventhough the combined network didn't give the results as better as the original hierarchical network, It gave a decent performance. Further improvements or tweaks in this proposed method can give better performance.

6. References:

1. Yimin Yang, "Features Combined From Hundreds of Midlayers: Hierarchial Networks with Subnetwork nodes", Neural Netw. , Jan. 2019.
2. Nan Ying Liang, “A Fast and Accurate Online Sequential Learning Algorithm For Feedforward Networks”, Neural Networks, Vol 17, No 6, Nov. 2006
3. J. Schmidhuber, “Deep learning in neural networks: An overview,”.Neural Netw., vol. 61, pp. 85–117, Jan. 2015.
4. Y. Yang and Q. M. J. Wu, “Extreme learning machine with subnetwork hidden nodes for regression and classification,” IEEE Trans. Cybern.vol. 46, no. 12, pp. 2885–2898, Dec. 2016.
5. S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in Proc. IEEE Int. Conf. Comput. Vis. PatternRecognit., New York, NY, USA,Jun. 2006, pp. 2169–2178.
6. J. W. Cao and Z. P. Lin, “Extreme learning machines on high dimen-sional and large data applications: A survey,” Math. Prob. Eng.,vol. 2015, pp. 1–12, Mar. 2015.

Individual Report #2

Student: Abhishek Guntaka – 0891103

Supervisor: Dr. Yimin Yang

Feature Extraction from hundreds of mid-layers combined with OS-ELM

Abhishek Guntaka - 0891103

March 2020

1 Abstract :

In this project we tried to implement the OS-ELM (Online Sequential Extreme Learning Machine) algorithm on another algorithm that extracts features from a hundreds of mid-layers in a SLFN (Single Layer Feed forward Network)with sub network neurons, To improve the learning and to make the algorithm into a non-iterative learning algorithm.

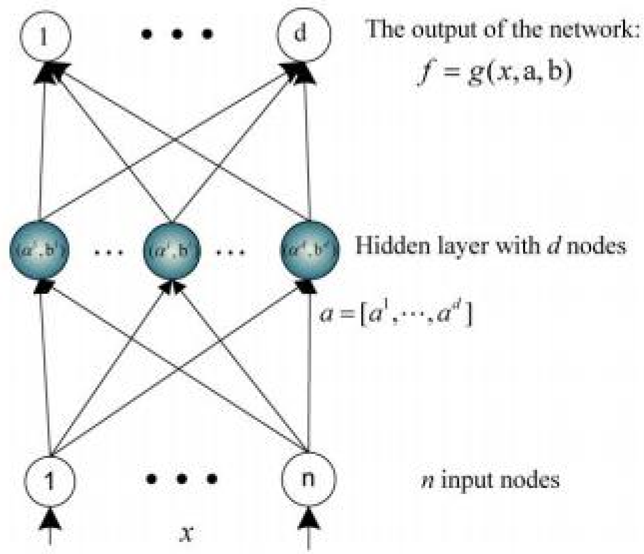
Keywords - OS-ELM,SLFN,Non-iterative Learning

2 Introduction :

Deep learning is being improved in an exponential rate in the past few years and there are various methods that have reached their Maximum potential and the majority of them use iterative method of learning and this method has been a paradigm for training a hierarchical neural networks, the learning e ectiveness and learning speeds of this method of learning is has not yet reached the required amount. Non-iterative learning on the other hand has not been explored to a big extent.

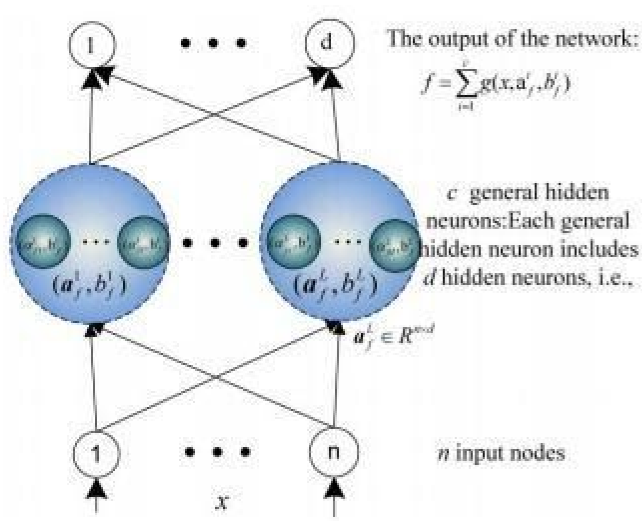
2.1 Feature Extraction from hundreds of mid-layers :

The number of layers of the network that we create also has a huge impact on the execution speed and the accuracy of the algorithm (complexity of the algorithm), In this project we are implementing the usage of deep three-general layer learning framework [1] to begin with.This is a modi cation over the existing single layer feed forward network, this single layer feed forward network has a general hidden layer in which there exists a sub-network of neurons, this sub-network plays a crucial role in feature generation.



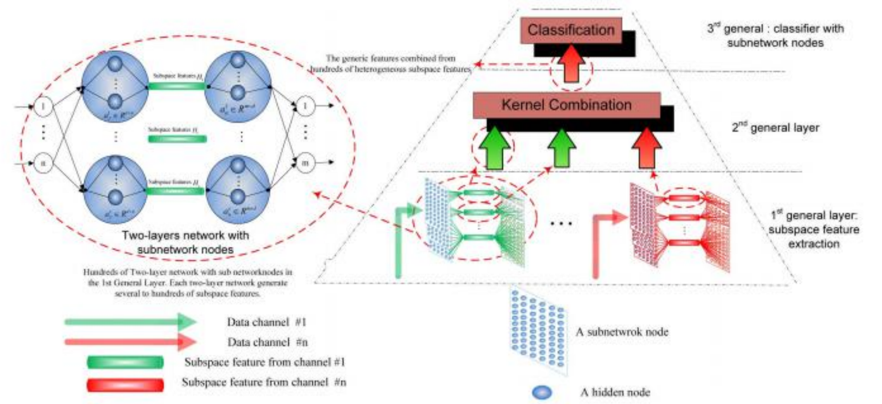
g 1 : Structure of single layer feed-forward network

This is the the general structure of a single layer feed forward net-work with input nodes, a hidden layer of nodes and the nal layer.



g 2 : Single layer feed-forward network with sub-network nodes.

This is the single layer feed-forward network that consists of a hidden layer which consists of a sub-network nodes.



g 3 : This is the method that was proposed in the paper for the multiple data channel structure.

This is the hierarchy of the layers that we have used to generate the nal classi cation of the pre-trained features.

The three layers that we used are the rst, middle and last layer. These layers are explained in detail in the following sections of the report.

2.2 Online Sequential Learning algorithm :

This algorithm is used for single hidden layer feed forward networks with additive or radial bias function (RBF), It can learn data one by one or chunk by chunk (varying or xed chunk size, This is mainly used for batch learning.

Activation functions used in this algorithm are any bounded non-constant piece wise continues functions (for Additive nodes) and in-tegrable piece wise continuous functions (for RBF nodes). In this method we can only choose no.o.nodes, no other parameter is chosen manually.

3 Algorithm :

3.1 Part 1 :

The following is the algorithm for using the modi ed SLFN from the 2.1 section -

Algorithm 1 Proposed Method

Given N features groups $(Q_1, \ldots, Q_N)$ generated from the same data set $Q_1 = \{(\mathbf{x}_k^1, \mathbf{y}_k^1)\}_{k=1}^M, \mathbf{x}_k^1 \in \mathbf{R}^{n_1}, \ldots, Q_N = \{(\mathbf{x}_k^N, \mathbf{y}_k^N)\}_{k=1}^M, \mathbf{x}_k^N \in \mathbf{R}^{n_N}$, positive coefficient C, and the number of subnetwork nodes L.

first general layer: Subspace feature extraction:

Set $n = 1$, and let $\{\mathbf{x}, \mathbf{y}\}$ equals Q_1.

while $n < N$ **do**

  while $c < L$ **do**

    Obtain a subspace feature $\mathbf{H}_f^{c+((n-1)\times L)}$ based on Section II-B Step 1–8.

  end while

  return Obtain L subspace features $\{\mathbf{H}_f^{1+((n-1)\times L)}, \ldots, \mathbf{H}_f^{L+((n-1)\times L)}\}$ from data group Q_n.

end while

return Obtain $L \times N$ subspace features $\{\mathbf{H}_f^1, \ldots, \mathbf{H}_f^{L\times N}\}$.

second general layer Feature combination:

Obtain combined features $\mathbf{H}$ as:

$$\mathbf{H} = \mathbf{H}^{1\oplus 2\oplus \ldots \oplus (N\times L)}$$

third general layer: Pattern learning: Given combined feature $\mathbf{H}$, set $c = 1$, $e_1 = \mathbf{t}$.

while $c < L$ **do**

  Step 1: Calculate the cth subnetwork node $(\mathbf{a}_p^c, b_p^c)$, and output weights $\boldsymbol{\beta}_p^c$ as:

$$\mathbf{a}_p^c = g^{-1}((\mathbf{e}_{c-1})) \cdot \mathbf{H}^T \left(\frac{C}{\mathbf{I}} + \mathbf{H}\mathbf{H}^T\right)^{-1}, \quad \mathbf{a}_p^c \in \mathbf{R}^{n\times m}$$

$$b_p^c = \mathrm{sum}(\mathbf{a}_p^c \cdot \mathbf{H} - h^{-1}((\mathbf{e}_{c-1})))/N, \quad b_p^c \in \mathbf{R}$$

$$\boldsymbol{\beta}_p^c = \frac{\langle \mathbf{e}_{c-1}, u^{-1}(g(\mathbf{a}_p^c \cdot \mathbf{H} + b_p^c))\rangle}{\|u^{-1}(g(\mathbf{a}_p^c \cdot \mathbf{H} + b_p^c))\|^2}$$

  Step 2: Calculate $\mathbf{e}_c = \mathbf{e}_{c-1} - \boldsymbol{\beta} \cdot g(\mathbf{a}_p^c, b_p^c, \boldsymbol{\beta}_p^c)$.

end while

2 Part 2 :

The following is the algorithm for implementing 2.2 section -

1. Training Observations are sequentially (one by one or chunk by chunk) with varying or xed chunk length feeded to learning algo-rithm.

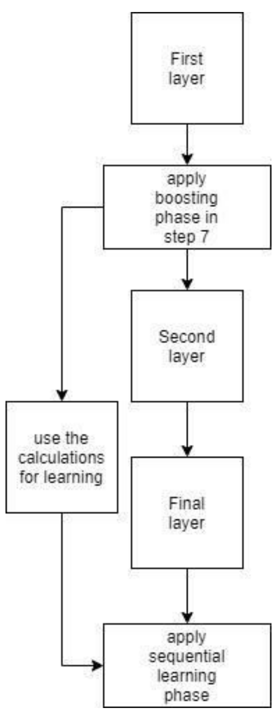
2. At any time, only the newly arrived single or chunk of observations are seen and learned.

3. After a chunk of training observations are discarded after the respective observations are complete.

4. Learning algorithm has no prior knowledge as to how many train-ing observations will be presented.

3.3 Combining part 1 and 2 :

Here is how we implemented the online sequential leaning in part 1 :



4 Contributions :

Code for features from hundreds of midlayers and implementation of chunking algorithm

5 References :

[1] J. Schmidhuber, \Deep learning in neural networks: An overview," Neural Netw., vol. 61, pp. 85{117, Jan. 2015.

[2] Y. Yang and Q. M. J. Wu, \Extreme learning machine with subnetwork hidden nodes for regression and classi cation," IEEE Trans. Cybern., vol. 46, no. 12, pp. 2885{2898, Dec. 2016.

[3] Y. LeCun et al., \Backpropagation applied to handwritten zip code recogni-tion," Neural Comput., vol. 1, no. 4, pp. 541{551, Dec. 1989.

[4] G. E. Hinton and R. R. Salakhutdinov, \Reducing the dimensionality of data with neural networks," Science, vol. 313, no. 5786, pp. 504{507, Jul. 2006.

[5] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, \Greedy layer-wise training of deep networks," in Proc. Adv. Neural Inf. Process. Syst., Vancou-ver, BC, Canada, 2007, pp. 153{160.

Individual Report #3

Features Combined from Hundreds ofMidlayers:Hierarchical Networks With Subnetwork Nodes.

Individual Report

Student:

Krisha Devi Kocherla (0892897)

Professor/ Supervisor:

Dr. Yimin Yang

Course:

Course title

Comp-5800-YH PROJECT

Articles and papers referred and researched

1. Yimin Yang, "Features Combined From Hundreds of Midlayers: Hierarchial Networks with Subnetwork nodes", Neural Netw. , Jan. 2019.
2. Nan Ying Liang, “A Fast and Accurate Online Sequential Learning Algorithm For Feedforward Networks”, Neural Networks, Vol 17, No 6, Nov. 2006
3. J. Schmidhuber, “Deep learning in neural networks: An overview,”.NeuralNetw., vol. 61, pp. 85–117, Jan. 2015.
4. Y. Yang and Q. M. J. Wu, “Extreme learning machine with subnetwork hidden nodes for regression and classification,” IEEE Trans. Cybern.vol. 46, no. 12, pp. 2885–2898, Dec. 2016.
5. Study of **Oselm** algorithm

Implementation

1. Implementation of OSELM algorithm
2. Contributed in implementing the three tier hierarchical layers in python
3. Contributed in understanding the implementation of the feature extraction layer with the OSELM parchment
4. Contributed in understanding the implementation method with feature extraction of different datasets performed by the team members.

Documentation of Report

1. Proposed flowchart
2. Motivation of OSELM
3. Experimental verification

4. Power point presentation