Improvement of accuracy and accelerate learning using Net2Net technique of classification datasets

Name : Basant Kumar(Undergraduate, IIT Bhubaneswar, batch 2016-2020, Roll:16MM01011)
Internship period: 14.05.2018 to 29.06.2018
Software used: MATLAB,ANACONDA,PYTHON
Guided by :Prof. Nischal K. Verma ,IIT Kanpur
Mentored by: Arun Kr. Sharma, Vikas Singh IDEA Lab, IIT Kanpur

Purpose of the project :To accelerate the learning based on the concept of function preserving transformations. This method accelerates the experimentation process by transferring the knowledge from a previous network to new deeper or wider network without altering the functionality.

Abstract: First of all I learnt some basics concepts of machine learning like backpropagation,different types of cost functions,training methods,classification methods,regression method.A brief overview of my project are following :

1. I implemented simple training using stochastic gradient descent backpropagation and tested with square function and other types of polynomials.
2. In the 2nd week of internship **LSTM** has been implemented and tested using polynomial functions and a **CBM dataset**.Through LSTM I came to know about the concept of memory in machine learning.How it store information and use it's information and make the network better.
3. Then I learnt autoencoder, sparsity, K-L divergence, PCA, whitening, regularization, batch learning. These are used in code

according to dataset for better accuracy. Then I implemented simple DNN using stacked AE. Regularizer has been used to avoid overfitting while autoencoder is for dimensionality reduction and to achieve global minima of cost function.

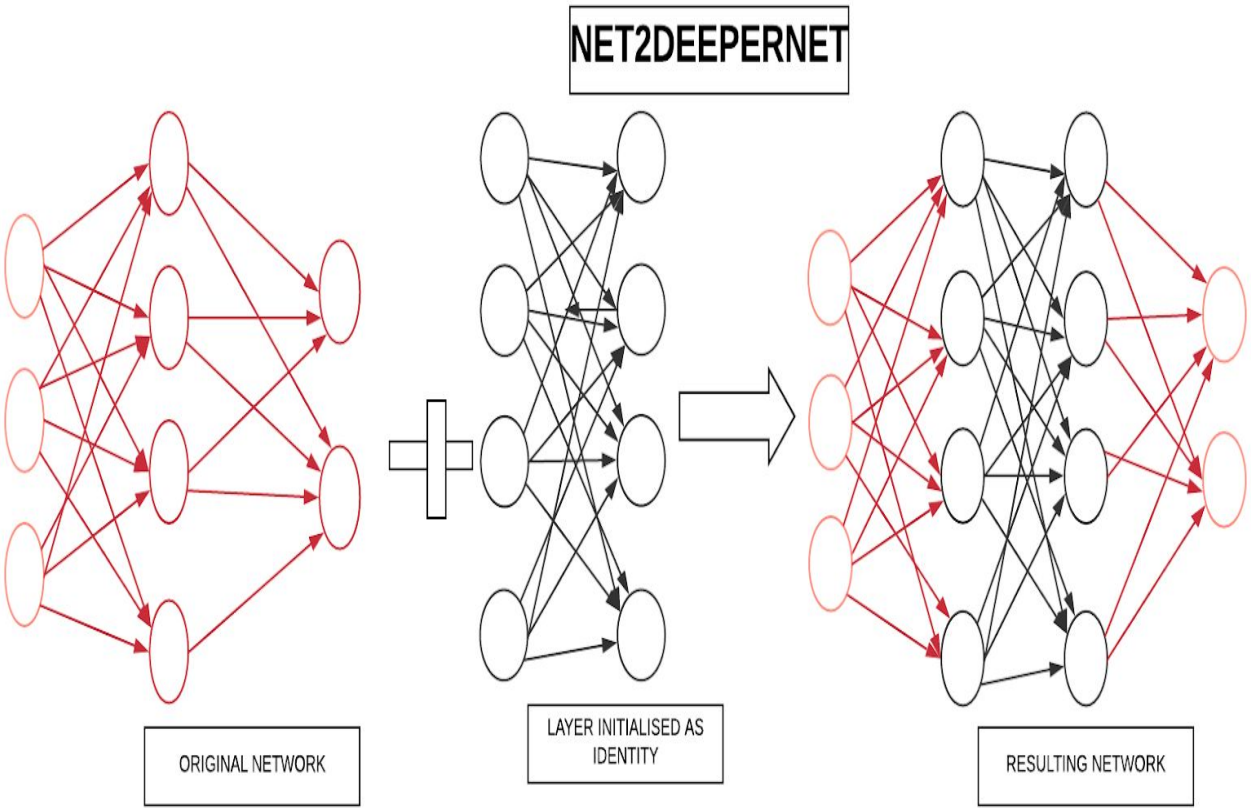
4. Solved a unsupervised learning problem given on stanford website(ufldl) of sparse autoencoder.I had to fulfill some piece of code.
5. Then I studied two types of transfer learning called **Net2DeeperNet** and **Net2WiderNet** for optimizing time complexity.In these cases network is first trained using some fraction of data and it work as accurate as if it's trained with all data.I used stacked AE as greedy layer-wise training for each hidden layer and then fine-tuned using backpropagation.In **Net2DeeperNet** network is first trained and then desired extra layers are added adjacent to a particular hidden layer with the same no of nodes as in that hidden layer, whereas in **Net2WiderNet** desired extra nodes are added to a particular hidden layer.
6. Now for classification I learned 4 methods:**Softmax, SVM linear, SVM with rbf and random forest classifier**. I implemented all these methods from scratch and not used any library.In **softmax** probability of each class is calculated for each test example and max of them is assigned to 1 and compared with the actual value whereas the concept of **svm** is based on that there should be certain margin between correct class and incorrect class and according to that svm loss is set-up.**RF** is base on decision tree method. In case of **random classifier** which has two types: one to all and one to one.In one to all for each test input output value of each class is calculated and max of them is considered whereas in one to one we find max of each

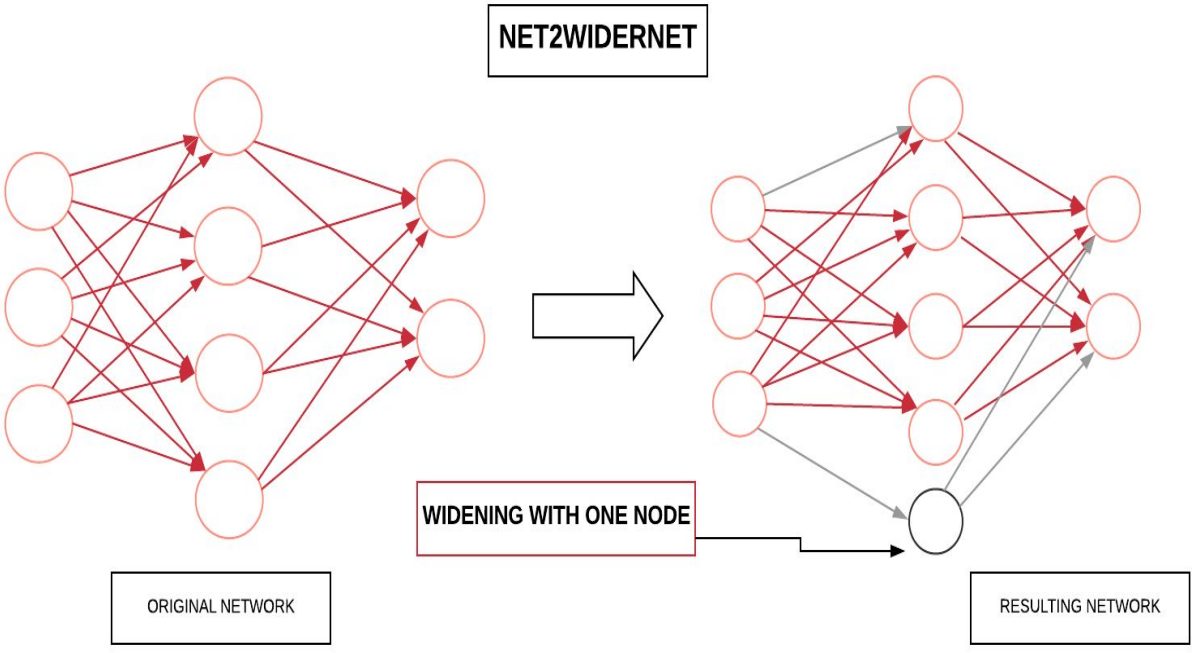
two class for all training inputs. So total of we need n*(n-1)/2 binary classifier for complete classification of n classes.

7. The **dataset** I used were :mnist, cwru(fault diagnosis bearing datasets, 4 sets) and NASA(fault diagnosis bearing datasets,3 sets).

   So I used 3 training methods(**DNN, Net2WiderNet, Net2DeeperNet**), 4 classification methods(**Softmax, svm linear, svm rbf and rf**) and 8 datasets(**1 mnist, 4 cwru and 3 NASA**). I got 3*4*8=96 results.

   Through following diagrams I have tried to make the concepts of Net2DeeperNet and Net2WiderNet more clear.


NET2DEEPERNET
ORIGINAL NETWORK
LAYER INITIALISED AS IDENTITY
RESULTING NETWORK


NET2WIDERNET
WIDENING WITH ONE NODE
ORIGINAL NETWORK
RESULTING NETWORK

References:

1. https://csegroups.case.edu/bearingdatacenter/pages/download-data-file
2. http://ufldl.stanford.edu/
3. https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/
4. http://www.mdpi.com/1424-8220/17/3/549/pdf-vor
5. http://openclassroom.stanford.edu
6. https://arxiv.org/pdf/1511.05641.pdf
7. https://arxiv.org/pdf/1804.09081.pdf
8. https://ac.els-cdn.com/S0165168416301797/1-s2.0-S0165168416301797-main.pdf?_tid=ef4cbf2c-c62a-403c-8a44-6c11e6f3555c&acdnat=1529132327_788dd1a714a925a7818287d77cce50fd
9. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7983338
10. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5375835/pdf/sensors-17-00549.pdf
11. http://colah.github.io/posts/2015-08-Understanding-LSTMs/