Noise Suppression by Artificial Neural Networks

by

Garimella Ramamurthy

Report No: IIIT/TR/2017/-1

Centre for Security, Theory and Algorithms
International Institute of Information Technology
Hyderabad - 500 032, INDIA
August 2017

Noname manuscript No.
(will be inserted by the editor)

NOISE SUPPRESSION BY ARTIFICIAL NEURAL NETWORKS

Garimella Rama Murthy · Yaparla Ganesh

Received: date / Accepted: date

Abstract In this research paper, noise suppression ability of associative memories (e.g. Hopfield neural network) is briefly summarized. Motivated by this fact, noise suppression abilities of trained convolutional network is discussed. Introducing the concept of null vectors, noise suppression ability of Single Layer Perceptron, Multi-Layer Perceptron and Extreme Learning Machine are discussed.

Keywords Nulls, Artificial Neural Networks, Multi Layer Perceptron, Hybrid Neural Networks

1 Introduction

Models of Artificial Neural Networks(ANNs) are innovated to understand the capabilities of biological brain. Single Layer Perceptron(SLP) as well as Multi Layer Perceptron(MLP) provided highly successful ANNs employing supervised learning for classification tasks. In MLP, typically, the number of hidden layers was small as it served the purpose of classification in many applications.

In an effort to train an ANN for recognizing handwritten characters, Yann LeCun innovated the essential ideas of convolutional neural networks(CNNs). Typically the number of layers of neurons was larger than earlier ones employing MLP. To distinguish the learning paradigm from earlier ones, researchers coined the word "Deep Learning." Yann LeCun successfully derived the back propagation algorithm for CNNs.

Garimella Rama Murthy
Signal Processing and Communication Research Center, IIIT Hyderabad
E-mail: rammurthy@iiit.ac.in

Yaparla Ganesh
Signal Processing and Communication Research Center, IIIT Hyderabad
E-mail: ganesh.yaparla@research.iiit.ac.in

One of the authors proposed the idea of dynamic synapse modeled as an FIR filter[2]. ANNs based on such a model were innovated by the author and other researchers. From the theory of linear filtering, it is clear that such a dynamic synapse has the ability to suppress frequency selective noise(e.g. low pass filtering). To the best of knowledge of the authors, noise suppression abilities of CNNs were not investigated by the earlier researchers. This research is an effort in that direction.Also, noise suppression capabilities of arbitrary MLP and Extreme Learning machine(ELM) are not investigated. Bruck et.al related Hopfield associative memory and generalized neural networks with error correcting codes. This work explored noise suppression abilities of HAM. This effort motivated the authors to explore the noise suppression abilities of ANNs.

This research paper is organized as follows. In section2, Known literature on noise suppression abilities of associative memories such as Hopfield neural network is summarized. In section 3,4,5,6; based on the concept of null vectors, noise suppression abilities of CNNs, MLP, ELM, SLP are discussed.

2 Noise Suppression in ANNs: Review of Literature

2.1 Associative Memories: Noise Suppression

Bruck et.al showed that Hopfield neural network is naturally associated with a graph theoretic code, in the sense that the local optimum of quadratic energy function are related to the codewords. More generally, local optimum of higher degree energy function(of a generalized neural network), constitute the codewords of an error correcting code.

Every codeword of linear/nonlinear error correcting code constitute stable state and every stable state of generalized neural network constitutes a codeword[1]. Thus, when the patterns are quantized to finitely many values (e.g 0,1,2,....,'p', where 'p' is a prime number), error correcting code associated with a generalized neural network/associative memory performs "clustering" and renders noise suppression, i.e. correction of certain number of errors (determined by the minimum distance of code). Thus, such associative memories enable noise suppression.

Note 1: We expect connection between CNNs and convolutional codes.

2.2 Hybrid ANNs: Noise Suppression

In [3], the first author proposed hybrid neural networks by cascading an associative memory/encoder (of an error correcting code) with MLP(before MLP block or after MLP block). such ANNs perform clustering before or after classification by MLP (also hybrid neural network i.e feed forward and feedback neural network with encoder before and/or after MLP was discussed). Thus, as discussed earlier, hybrid neural networks perform noise suppression before and/or after classification by MLP or CNN.

2.3 Motivation for Noise suppression in MLP

In supervised learning, the class information of input pattern vectors is known. Most of the patterns in Euclidean space which belong to class are closer in Euclidean distance (or other distances such as mahalonibis distance). The components of pattern vectors are not necessarily quantized to integer values (unlike in case of block codes). The noise model that we consider is "additive noise vector" corrupting the pattern vector. The outputs of neurons, other than the first layer are determined by the trained network. As discussed in this research paper, the null space of the finite dimensional linear transformation(corresponding to trained synaptic weights) between input vector and the vector to whose components activation function is applied, characterizes the additive noise which is suppressed. It effectively specifies additive perturbations to input vector that are suppressed.

3 Linear Convolutional Layers: Finite Dimensional Linear Transformations

In this section, we explore the transformations effected by convolutional layers in a Convolutional Neural Network. We are motivated by the fact that linear filtering of the trained input signal corresponds to convolution of input with impulse response of the Linear Time Invariant System.

Let there be N neurons in a layer. Let the outputs of the neurons be incorporated into a 'N'- dimensional column vector(i.e $N \times 1$ vector, $\bar{U}$). The outputs of such neurons are convolved/correlated with the Mask Coefficients. Let the mask be of length M (i.e $\bar{H}$=[H(0).... H(M-1)] be convolved/correlated with input vector $\bar{U}$. Let the output vector (of length M+N-1) be denoted by $\bar{Y}$. For the sake of illustration, let N=6, M=3. The convolution/correlation of the input vector, $\bar{U}$ with mask vector, $\bar{H}$ effectively corresponds to finite dimensional linear transformation. i.e $\bar{Y}$=$\bar{H}$ $\bar{U}$, where, In general $\bar{H}$ is a $(M+N-1) \times N$ matrix.

$$\bar{Y} = \begin{pmatrix} y(1) \; y(2) \; ... \; y(8) \end{pmatrix}$$

$$\bar{U} = \begin{pmatrix} u(1) \; u(2) \; ... \; u(6) \end{pmatrix}$$

h(0)≠0

$$\bar{H} = \begin{bmatrix} h(0) & h(1) & h(2) & 0 & 0 & 0 \\ 0 & h(0) & h(1) & h(2) & 0 & 0 \\ 0 & 0 & h(0) & h(1) & h(2) & 0 \\ 0 & 0 & 0 & h(0) & h(1) & h(2) \\ 0 & 0 & 0 & 0 & h(0) & h(1) \\ 0 & 0 & 0 & 0 & 0 & h(0) \\ - & - & - & - & - & - \\ - & - & - & - & - & - \\ - & - & - & - & - & - \end{bmatrix}$$

In convolution even though $\bar{Y}$ is a vector of length 8, only first 4 components are considered(since last 4 components do not involve correlation with all the mask coefficients i.e only some mask coefficients are involved). It can be readily noted that $\bar{H}$ is a rectangular toeplitz matrix.

Note 2: The generalization to arbitrary dimensional input vector and mask vector is straight forward, avoided for brevity.

It should be noted that if pooling/sub-sampling operation is "linear" (for instance "average" computation or down sampling), then the output of pooling vector $\bar{Y}$ corresponds to a finite dimensional linear transformation, $\bar{P}$.

$$\bar{Z} = \bar{P}\bar{Y} = \bar{P}\bar{H}\bar{U} \tag{1}$$

Note 3: since only first four rows of $\bar{H}$ (8×6 matrix $\bar{H}$) are considered to determine the output $\bar{Y}$, it is clear that we have $\hat{Y}$ =$\hat{H}$ $\bar{U}$ (where $\hat{H}$ is 4x6 matrix). Thus the null space of $\hat{H}$ is of dimensional 2. In other words non trivial input $\hat{U}$ exists such that $\hat{H}$ $\hat{U}$=0.

Hence we have

$$\hat{Y} = \hat{H}(\bar{U} + \hat{U}) = \hat{H}\bar{U} \tag{2}$$

Equivalently vectors in the null space of $\hat{H}$ are the noise vectors which will be suppressed by the trained convolutional layer in a CNN. Thus, we are able to quantify certain noise suppression ability of a CNN. In general, the null space($\hat{H}$) is of dimension N-M-1. All such noise vectors are suppressed by a trained convolutional layer. $\hat{H}$ is also a toeplitz matrix.

$\hat{H} = [\bar{A} : \bar{B}], where$

$$\bar{A} = \begin{bmatrix} h(0) & h(1) & h(2) & 0 \\ 0 & h(0) & h(1) & h(2) \\ 0 & 0 & h(0) & h(1) \\ 0 & 0 & 0 & h(0) \end{bmatrix}$$

i.e. $\bar{A}$, being upper triangular is non singular if h(0)≠0.

Some properties of $\bar{A}$:

1. Eigenvalues of $\bar{A}$ are all equal to "h(0)."
2. The null space dimension is $'4'$ if h(0)=0 and is $'0'$ if $h(0) \neq 0$.

The columns of $\bar{B}$(2 of them) are linearly dependent on the columns of $\bar{A}$. In fact, if

$$\bar{B} = [\bar{b_1} : \bar{b_2}]; \tag{3}$$

$$\bar{A}\bar{c_1} = \bar{b_1}; \tag{4}$$

$$\bar{A}\bar{c_2} = \bar{b_2}; \tag{5}$$

$$\bar{c_1} = \bar{A}^{-1}\bar{b_1}; \tag{6}$$

$$\bar{c_2} = \bar{A}^{-1}\bar{b_2} \tag{7}$$

Note 1 Upper triangular Toeplitz matrix can be effectively inverted[4].

Example:

$$\bar{A} = \begin{bmatrix} h(0) & h(1) & h(2) & 0 \\ 0 & h(0) & h(1) & h(2) \\ 0 & 0 & h(0) & h(1) \\ 0 & 0 & 0 & h(0) \end{bmatrix}$$

Let's take h(0) to be 1, h(1) to be 0.5 and h(2) to be 0.3.
Then,

$$\bar{A} = \begin{bmatrix} 1 & 0.5 & 0.3 & 0 \\ 0 & 1 & 0.5 & 0.3 \\ 0 & 0 & 1 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\bar{A}^{-1}$ can be calculated using technique provided in [4]. $\bar{A}^{-1}$ is also an upper triangular Toeplitz matrix.

$$\bar{A}^{-1} = \begin{bmatrix} 1 & -0.5 & -0.05 & 0.175 \\ 0 & 1 & -0.5 & -0.05 \\ 0 & 0 & 1 & -0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Goal: precise determination of noise suppression ability of a trained convolutional layer i.e null space of $\hat{H}$.i.e. $\bar{f_1}$, $\bar{f_2}$ are the basis of null space of $\hat{H}$, then the set of noise vectors which are suppressed by the convolutional layer can be characterized

$$\bar{g} = \alpha\bar{f_1} + \beta\bar{f_2}, \text{ with } \alpha, \beta \in R. \tag{8}$$

Note 4: Each convolutional layer receives input from the previous layer. Thus, each layer provides certain amount of noise immunity.

Finite Dimensional Linear Filtering: Each convolutional/pooling layer removes noise in a certain linear space/subspace.

Problem Formulation: To determine the basis of null space of $\hat{H}$.

$\hat{H} = [\bar{A}$: $\bar{b_1}$: $\bar{b_2}]$ with $\bar{A}$ being uppertriangular nonsingular Toeplitz matrix. $\bar{A}\ \bar{c_1} = \bar{b_1}$, and $\bar{A}\ \bar{c_2} = \bar{b_2}$. Further, we have that

$$\bar{b_1} = \begin{bmatrix} 0 \\ 0 \\ h(2) \\ h(1) \end{bmatrix}$$

,

$$\bar{b_2} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ h(2) \end{bmatrix}$$

, Suppose, h(2)≠0, h(1)≠0.
Then $\bar{b_1}$, $\bar{b_2}$ are linearly independent. Thus, the basis of null space of $\bar{H}$ can easily be determined. Hence noise suppression ability of CNN can be determined.
Specifically we have

$$\hat{H} \begin{bmatrix} c_1 \\ -1 \\ 0 \end{bmatrix} \equiv \bar{0} \ \&\ \hat{H} \begin{bmatrix} c_2 \\ 0 \\ -1 \end{bmatrix} \equiv \bar{0}$$

4 Null Vectors of Artificial Neural Networks

We previously considered the vectors which are in null space of finite dimensional linear transformation associated with a Convolutional Neural Network(CNN). We realized that such vectors are suppressed by a CNN, which is trained.

We realize that the above idea has implications to a trained multi-layer perceptron.

The input from any layer to the next layer is effected through a linear transformation. For instance suppose there are 'M' neurons in a layer and the outputs are fed to 'N' neurons in the next layer. Then, a finite dimensional linear transformation, T i.e. an NxM matrix captures the transformation. Associated with layers of neurons, we define the concept of null vectors associated with a a linear transformation.

Consider a finite dimensional linear transformation $\bar{T}$ associated with the synaptic weights from a layer to the next layer.

Definition: Null vectors of a layer of MLP are the vectors in the null space of $\bar{T}$.

Then a signum operation is performed at each of the 'N' neurons on the weighted input. Then vectors which are in null space of T are suppressed.In other words, suppose $\bar{U}$ is the output vector of a layer and $\bar{Y}$ is obtained through $\bar{Y}$=$\bar{T}\ \bar{U}$. If there exists, non zero vector $\bar{V}$ (belonging to null space of T) such that

$\bar{T}\ \bar{V} = 0$, Then

$$\bar{T}(\bar{U} + \bar{V}) = \bar{T}\bar{U} = \bar{Y} \tag{9}$$

(If $\bar{V}$ is a noise vector, it is suppressed from propagating to next layer).

Even in a fully trained MLP, with transformation matrices whose null space is not empty, effective NOISE removal takes place.

Implications of the above idea to supervised learning by MLP: After training i.e. synaptic weights are freezed, "NOISY" patterns(those vectors which lie in the null spaces of transformation matrices) are suppressed.

5 Null vectors of Extreme learning machine

It should be noted that Extreme learning machine(ELM) is one type of Multi-Layer Perceptron. In ELM weights from Input layer to Hidden layer are assigned randomly and Hidden layer to Output layer are computed by a closed form expression. Input vector is $\bar{U}$ and $\bar{Y_1}$, $\bar{Y_2}$ are the outputs of layer one, two respectively.

$$\bar{Y_1} = \bar{T_1}\bar{U} \tag{10}$$

$$\bar{Y_2} = \bar{T_2}\bar{Z_1} \tag{11}$$

$\bar{Y_1}$ is output of the first layer of neurons before activation function is applied and $\bar{Z_1}$ is output of first layer after activation function is applied.

$\bar{T_1}$ and $\bar{T_2}$ are finite dimensional linear transformations at first and second layers respectively. i.e Null Spaces of $\bar{T_1}$, $\bar{T_2}$ determine the "noise" which is suppressed.

Note 5: There is no training in ELM. Null spaces of $\bar{T_1}$,$\bar{T_1}$ determine the perturbations"(could be noise) to $\bar{U}$,$\bar{Z_1}$ which will be suppressed/zeroed out.

From the training data available, it could be determined if the null vectors are like "NOISE" which should be suppressed or they have desired pattern vector information(which should not be suppressed), this holds true for any ANN.

Note 6: Given "NULL VECTOR" information of an arbitrary ANN, can a choice of initial synaptic weights be intelligently made(w.r.t null spaces of linear transformation) such that training time may be reduced.

Note 7: This idea may be of specific importance in design of ELM. i.e choice of synaptic weights from input to the hidden layer(single one). From the null space of $\bar{T_1}$, it is clear that certain patterns are suppressed/zeroed out.

Cases of Interest: $\bar{Y_1}$=$\bar{T_1}$ $\bar{U}$, where $\bar{T_1}$ is an 'MXN' matrix.

1. $M > N$: Null space of $\bar{T_1}$ is of interest for us.
2. $M = N$: i.e. number of input neurons in layer one is same as number of neurons in next layer.

3. $M < N$: i.e. number of input neurons is smaller than number of neurons in next layer.

Suppose, in the case of ELM, certain "DESIRED" pattern vectors are getting suppressed because they lie in null space of finite dimensional linear transformation/matrix i.e $\bar{T_1}$. Then, such vectors will not effect the training process.

Note 8: In the case of ELM, there is no backpropagation for changing the weights with whatever initialization of weights we choose from input neurons to neurons in hidden layer, the other weights are computed in closed form.

Suppose the NULL SPACE of $\bar{T_1}$ is empty. Then, none of the input noise vectors are suppressed. Then the entire input space effects the performance of ELM.

6 Single Layer Perceptron: Null Vectors

Goal: Classification of Linearly Separable Patterns, i.e. Training by perceptron convergence law is associated with finitely many transformation matrices, during convergence of learning law.

Lemma: Linear transformation preserves linear separability.

Proof: Refer[5]. . If the weights from inputs to the neurons in a single layer are updated in successive iterations, the finite dimensional linear transformation/matrix is changing. i.e rows of linear transformation/matrix correspond to the hyperplanes separating the classes.
In finitely many steps, the synaptic weight matrices are converging to a matrix which properly classifies the patterns.

Note 9: In case, there are maximum possible number of classes(i.e. number of neurons in the single layer + 1 e.g M hyperplanes that are parallel corresponding to 'M' classes) the rows of successive synaptic weight matrices(on learning) converge to a rank-one matrix.

Note 10: When the number of classes is maximum and the converged synaptic weight matrix is rank-one matrix (i,e hyperplanes are parallel), the null space has dimension 'M−1', where 'M' is the number of neurons in SLP. The converged synaptic weight matrix, is a

1. Square matrix if the number of inputs is equal to the number of neurons in SLP.
2. Rectangular matrix if the number of inputs is greater than number of neurons in SLP.

7 Conclusion:

In this research paper, noise suppression ability of trained CNNs is explored. The effective idea is utilized to characterize the noise suppression capabilities of Single Layer Perceptron(SLP), Multi Layer Perceptron(MLP) and Extreme Learning Machine(ELM). The noise suppression ability of Hopfield associative memory is reviewed. Also Hybrid neural networks for suppressing noise are proposed.

8 Conflict of interest:

Garimella Rama Murthy states that there are no conflicts of interest.

References

1. J. Bruck and M. Blaum, Neural Networks, Error Correcting codes and polynomials over the binary cube, IEEE transactions on Information Theory, vol.35,no.5,September 1989
2. G.Rama Murthy, Optimal Robust Filter Models of Synapse : Associated Neural Networks, Proceedings of International Conference on Soft Computing and Intelligent Systems, December 27-29, 2007.
3. G. Rama Murthy, Hybrid Neural Networks, Proceedings of International Conference on Power System Analysis, Control and Optimization(PSACO-2008), 13th-15th March 2008.
4. https://math.stackexchange.com/questions/786108/explicit-formula-for-inverse-of-upper-triangular-toeplitz-matrix-inverse.
5. Murthy, G. Rama. Multidimensional Neural Networks: Unified Theory. New Age International, 2008.