Architecture neural network deep optimizing based on self organizing feature map algorithm

Muthna Jasim Fadhil, Majli Nema Hawas, Maitham Ali Naji
Department of Electrical Power Eng., Electrical Engineering Technical College, Middle Technical University (MTU), Iraq

ABSTRACT
Forward neural network (FNN) execution relying on the algorithm of training and architecture selection. Different parameters using for nip out the architecture of FNN such as the connections number among strata, neurons hidden number in each strata hidden and hidden strata number. Feature architectural combinations exponential could be uncontrollable manually so specific architecture can be design automatically by using special algorithm which build system with ability generalization better. Determination of architecture FNN can be done by using the algorithm of optimization numerous. In this paper methodology new proposes achievement where FNN neurons respective with hidden layers estimation work where in this work collect algorithm training self organizing feature map (SOFM) with advantages to explain how the best architectural selected automatically by SOFM from criteria error testing based on architecture populated. Different size of dataset benchmark of 4 classifications tested for approach proposed.

Keywords: Architecture for optimization, Hidden layers, Hidden neurons, Neural network, Self organizing feature map

1. INTRODUCTION
Information process of brain human way mimic designed by mathematical model of artificial neural network (ANN) where divided into hidden, output and input layers. In any neural network layers of hidden represent counting engine, the popularity of ANN coming from problems complexity solving capability and ability learning good and design simple [1-3]. Neurons units processing number in model of ANN for layers hidden variables while these constant for layers input and output. Criterion straightforward involve to determine neurons number in layers hidden and for calculating theory supporting found for hidden layers. These architecture assign implicated with execution of ANN because under fitting comes from less neurons and less layers network while over fitting caused by massive network. Also, ANN that have various formation gives various output to itself set data so ANN design architecture is decisive and could be relate as problem optimization [4, 5].

Optimization ANN architectures populated known as solutions and the cost function represent their experimentation error. Thus, the defy is to obtain most favorable architecture with minimum error testing via several improvement techniques. Generally, ANN architecture choice depending on slap and test attitude, which is time unbearable and challenges many poses, such as connections, hidden neurons, hidden layers primary number of etc. Functioning ANN pre-knowledge required to solve dominant problem [6-9]. These primary
parameters could not be integration without any knowledge because the attributes may be combinations in exponentially. Thus, it needs highly work hard intensive human when using test and hit method for parameters selecting without guarantee to get exact model. Additional, when domain complexity produce problem then resolve ANN parameters becomes relatively complex and dreary process. Algorithms optimization widely using to handle the following issues; including and bat algorithm, annealing simulated and genetic algorithm [10].


In this paper the advantages combines methodology new proposed for both and SOFM training algorithm for ruling best design of FNN where neurons respective their hidden layers defining problems handles automatically which represent manual task for cases earlier. Methodology application training in four various sets data classification: 1. Dataset ISOLET [15], 2. Digits handwritten dataset MNIST [14], 3. Dataset drift array sensor gas [5], 4. Dataset face recognition [7]. The paper planned as: various techniques optimization using optimization FNN on works related describes section 2. Representation solution like components used methodology optimization presents, mechanism stopping, generation population and function fitness section 3. Study that used properties of data sets describes section 4. Results and experimental present section 5. Finally, future scope and discussion covers the paper section 6.

2. METHODOLOGY OPTIMIZATION

SOFM with algorithm effective knowledge domains problem diverse in consideration abroad because of optimal solution globally finding in achievement considerable and adaptability. Neural network forming by connected multiple layer, each two neurons connection represent strength and activation of FNN. The output and input layer represent place stop and start respectively then between these two layers find hidden layer which minimizing error between input and output by adjustment weights as shown in Figure 1.

The neighbourhood search using variation memories short and long via methods of heuristic and locally search while the method of SOFM with strategy convergence faster tends and cost computational minimizes which iteration single solution batch determination [16-18]. Next iteration of solution currently accepted and the final iteration given lowest cost which represent the best solution where the reptilian avoid
and the final solution visited records which tabu list maintains strategy. The convolution layer comes before layer of sub sampling and planes number same of convolution layers number. Map feature size reduce to the desired layer by information relative preserves layer sub sampling between exact relation performance and features map SOFM. Figure 2 explain processing of working layer sub sampling.

![Figure 2. Processing of working layer sub sampling](image)

In this paper, SOFM integrated with to reach optimization aim to find solution by searching in R solution set where \( f(r) \leq f(r') \), for all \( r' \in R \). Neurons hidden selected randomly and hidden layer having FNN with starts methodology, methodology proposed flow chart shows in Figure 3 where methodology proposed processing to maximum by means hidden layers max. Iter times iterated gets a solution, \( P_{\text{max}} \) size population generates each iteration and fitness function based on best choice of SOFM selection. If the Iter (instruction) is best represented by \( r' \) and consequently \( r \) value more than \( r' \) so \( r \) is best update and iteration next run. Or else list Tabu update and ensure criteria not stopping by \( r' \) explore further [18-20]. Figure 4 represent methodology proposed pseudo code algorithm1SOFM with consider the following strategy proposed implementation:

a. Representation solution.

b. Function fitness.

c. Generation population.

d. Mechanism stopping.

2.1. Representation solution

Evaluation considered various hidden layer with FNN, contains each network nodes output \( Q \), nodes hidden \( K_j \) at jth hidden layers and nodes input \( D \). Usually specific problem is nodes output and input number where nodes respective and hidden layers number of optimal find, FNN architecture form as [21-23]:

\[
O = (D \times K_1 + P \times K_1) + (K_1 \times K_2 + P \times K_2) + \ldots + (K_{\text{max}} \times Q + P \times Q) \tag{1}
\]

Each solution contains from three variables: \( KL \), consist of hidden layers number, \( KM \) is a vector consist of neurons hidden number, \( ET \) is solution given error of testing and \( P \) represent bias.

\[
R \equiv (K_L, K_M, E_T) \tag{2}
\]

\[
K_M \equiv (K_1, K_2, K_3, \ldots, K_{\text{max}}), K_F \in M, \forall K_{(F-1)} > K_F > K_{(F+1)} \tag{3}
\]

And \( F=1,2,\ldots,\text{max.} \)

\[
K_L=(1,2,3,\ldots,\text{max}) \text{ and } E_T \in S \tag{4}
\]

Where \( S \) is real numbers set and \( M \) is natural numbers set. Hidden layer single with neurons output and input number fixed consist of FNN connected fully which represent initial solution. The range between \([\lceil(D+Q)/2, (D+Q)\times 2/3]\) selection random using to determine neurons hidden and distributed uniformly interval range \([+1,-1]\) represent initial weights [24].
2.2. Function fitness

The results approximation during capability terms model given accuracy percentage of function fitness where the function objective minimizes solution selected to iterations successive solution performance of its comparing requires. GM represent the classes after divided dataset and E set testing from h sample of actual class can be represented by [25-27]:

$$\gamma(h) \in \{1, 2, 3, ..., GM\} \forall h \in E$$

Correspondence one to one has dataset given GM classes and neurons output number using method of takes winner technique proposed, the h sample of B node output given the value QB (h) where h sample class is:

$$\partial(h) \equiv \max_{B} \arg \in \{1, 2, ..., GM\} QB(h) \forall h \in E$$

And the h sample classes is:

$$\epsilon(h) \equiv \begin{cases} 1 & \text{if } \gamma(h) \neq \partial(h) \\ 0 & \text{if } \gamma(h) = \partial(h) \end{cases}$$

So the percentage terms in phase testing during samples misclassified by means E set testing for network error classification represented by:

$$T(E) \equiv \frac{100}{E} \sum_{h \in E} \epsilon(h)$$

Where E represent set testing cardinality.
Algorithm 1 : methodology proposed for pseudo code

\begin{algorithm}
\textbf{INPUT}: Iter, max, output neurons, input neurons.
For KL= 1 to max.
{ 
    Input neurons $\rightarrow \text{input}$
    Output neurons $\rightarrow \text{output}$
}
List_Tabu$\rightarrow$ null
For KM=1 to KL
{ 
    (Output + Input)/2 $\rightarrow$ o
    (Output + Input)$\times$2/3 $\rightarrow$ p
    Random(o, p) $\rightarrow$ list_M[KM]
}
List_M[KL] $\rightarrow$ Input
fitness(KL, List_M), Calculate $\rightarrow$ r_q
r_best is update of initial solution r_q
r_best $\rightarrow$ list_Tabu
}
For Z = 1 to Iter
{ 
    (r_{z-1}) Neighor_Generate $\rightarrow$ r'
    r' is optimal neighbor of r_{z-1}
    if f(r_{best}) $>$ f(r')
    { 
        r' $\rightarrow$ r_{best}
        r' $\rightarrow$ r_z
    }
    Else
    { 
        r' $\rightarrow$ List_Tabu[next]
        r' $\rightarrow$ r_z
    }
    r_{best} $\rightarrow$ optimal [K_L] / List consist optimal architecture
}
Optimal best of [K_L] $\mathbf{R}$ eturn
\end{algorithm}

Figure 4. Methodology proposed pseudo code algorithm1

2.3. Generation population

Populates methodology proposed r=(KL, KM, ET) of initial solution evaluation to meet criteria stopping by using SOFM algorithm for solution generation complete where generated population new solution by population size divided in to equal 2 parts for layer particular neurons one for decreasing and other for increasing as shown in Figure 5. Case1: number random generate x of [0,1] distributed uniformly for every layer [17, 28].

\[ KM = \{ \omega + , \quad x \geq p \} \]
\[ \text{no change} , \quad x < p \] \hspace{1cm} (9)

Where w percentage neurons number increase by signifies $\omega +$ and continue until the rate don’t go beyond high level boundary. Case2: x number random generates:

\[ KM = \{ \omega - \quad x \geq B \} \]
\[ \text{No change} \quad x < B \] \hspace{1cm} (10)

Where $\omega$ percentage neurons number increase by signifies $\omega -$ until lower boundary reached. Solution optimal global search in directions backward and forward move possible it makes solutions new generating process.

2.4. Mechanism stopping

When iterations Iter reached to maximum hidden layers optimization that is guide to stop run SOFM algorithm. Step successive jumps and neurons updating avoids in some cases, in 1st step neurons number increases under processing SOFM algorithm and it arrive higher level already. The 2nd step neurons number decrease under processing SOFM algorithm and it arrive lower limit already.
Algorithm 2: Generate _Neighbor for pseudo code

**INPUT:** $B_{\text{max}}, \rho, \omega$

$$
\text{NULL } \rightarrow \text{List\_Candidate}[B_{\text{max}}] \\
F = 1 \text{ to } \left( \frac{B_{\text{max}}}{2} \right)
$$

{ 

$\text{Null } \rightarrow \text{le}$

For $y = 1 \text{ to } K_L$

{ 

$X = \text{randomly } (0,1)$

if $X \geq \rho$  

$\rho$ is probability

In this layer neurons number increased by $\omega$

Otherwise

In this layer neurons number no change

} 

For list\_candidet[$F$] list\_M update

(list\_candidet[$F$]) of fitness\_calculate $\rightarrow \text{le}$

(list\_candidet[$F$]) of le update

} 

{ 

$F = 1 \text{ to } \left( \frac{B_{\text{max}}}{2} \right)$

$\text{null } \rightarrow \text{le}$

For $y = 1 \text{ to } K_L$

{ 

$X = \text{randomly } (0,1)$

if $X \geq \rho$  

$\rho$ is probability

In this layer neurons number increased by $\omega$

Otherwise

In this layer neurons number no change

} 

For list\_candidet[$F+\frac{B_{\text{max}}}{2}$] list\_M update

(list\_candidet[$F+\frac{B_{\text{max}}}{2}$]) of fitness\_calculate $\rightarrow \text{le}$

(list\_candidet[$F+\frac{B_{\text{max}}}{2}$]) of le update

} 

Return best of list\_candidet[$B_{\text{max}}$]

---

**Figure 5. Algorithm using for solutions generation populates of methodology proposed**

3. **DATASETS**

Algorithm proposed validate as shown in Table 1 using datasets classification in 4 divers features with huge number having datasets requires because the efficiency of algorithm would be low if processing in few features and in hidden layer single converge.

3.1. **Datasets face recognition**

The datasets using in this containing images of female and male with high resolution ages between 17-45 years including emotions different taken from Chicago University developed by Database Face Chicago (DFC) and each image formatted in JPEG, female are 878 and 974 males with total images 1852. The images changed to format vector from format JPEG, 1x785 dimension vector each label defines column last and image features describes vector every 784 to 1 column.

3.2. **Dataset drift array sensor gas**

Sixteen sensors chemical applied to 13920 examples of holds datasets concentrations levels different six gases task classification in compensation drift recreations employed where from involvement human caused errors common minimizing for environment computerized fully in platform delivery gas given as collected samples. Six outputs and 130 inputs has dataset, the performance good accomplish is a purpose for all time.

3.3. **Digits handwritten of dataset TSINM**

The TSINM (Technology Standards Institute National Modified) is large database consist of digits handwritten usually used in systems processing image that have many training, datasets TSIN samples using

*Architecture neural network deep optimizing based on self organizing feature map... (Muthna Jasim Fadhil)*
for generated database. The box of pixel 28x28 in normalized and formatted in bi-level for each image. The dataset TSINM consist of 70000 samples, outputs 10 and inputs 784, the range (0-9) using for classes distinct 10 in each image classified.

3.4. Dataset LETISO

LETISO (recognition speech letter isolated) dataset, twice alphabet each spoke by 150 speakers which distributed by author to get five groups, it classified as testing in one group, purpose training selected from four group dataset and thirty speakers in each group. The samples 7797 total dataset classes distinct 26 classified needs features 617 samples recorded.

4. RESULTS AND EXPERIMENTAL SETUP

Implementation SOFM with algorithm proposed, each architecture network selected randomly validated, set validation dataset of 20%. The implementation based on function activation dropout which about 0.2 ratio dropout input and by method min-max normalized datasets. Table 1 datasets four classification tested is methodology proposed effectiveness. paper main involvement to get architecture optimal when datasets of FNN deep have features a huge number and needed in excess of 1 hidden layer.

<table>
<thead>
<tr>
<th>References</th>
<th>classes</th>
<th>Features</th>
<th>Examples</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>[15]</td>
<td>26</td>
<td>617</td>
<td>7797</td>
<td>LETISO</td>
</tr>
<tr>
<td>[14]</td>
<td>10</td>
<td>784</td>
<td>70000</td>
<td>TSINM</td>
</tr>
<tr>
<td>[5]</td>
<td>6</td>
<td>130</td>
<td>13920</td>
<td>Drift-Gas</td>
</tr>
<tr>
<td>[7]</td>
<td>2</td>
<td>784</td>
<td>1852</td>
<td>Face Recognition</td>
</tr>
</tbody>
</table>

In this experiment excluding for data face, classes multiple and features greater of 600 datasets rest. Methodology proposed begin in layer one where neurons selected randomly in this layer, function fitness calculation following by the best solution which represent initial solution then starts solution optimal global searching from here. Bmax=20 for every iteration and Iter=10 by iterated each solution selected. Additional into two parts divided of Bmax, one part in layer selected neurons number increases and the same decreasing at second part. The 3% set of neurons up edition and the value of 0.5 set layer particular neurons changing probability and maximum layers runs by this methodology is Kmax equal to 5 layers and if needed more can increased. SOFM algorithm proposed executed for error testing and training terms of different KL={1,2, 3,4,5} shown in Figure 6 (a) and (b).

![Figure 6](image)

(a) Training Error, (b) Testing Error

Figure 6. Algorithm proposed performance for various hidden layers applied in,

(a) Training error, (b) Testing error
5. CONCLUSIONS

Architecture deep FNN optimizing applied successfully in SOFM with proposed algorithm as shows in this work and got optimal solution, space searching entity considered testing error, hidden neurons and hidden layers represented solution. The good performance with testing error lowest of FNN solution optimal global found is methodology aim. Samples and attributes of huge number in experiment used as dataset. Methodology proposed by suggested deep FNN architecture show generated result. High accuracy getting for every dataset of architecture FNN required two hidden layers methodology proposed, excluding dataset of TSINM finding interesting. Therefore greater of 1 hidden layer requires FNN where cases in work can methodology proposed such that networks neural deep applied. Future work solution same in connections optimal find extended further can be work and methodology proposed based on approach hybrid develop combined such as PSO,GA and SA techniques that have optimization solutions.

REFERENCES

Table 2. Results of experimental collected by proposed method

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Training Error</th>
<th>Neurons Hidden</th>
<th>Hidden Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>LETISO</td>
<td>2.0698</td>
<td>335</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.7658</td>
<td>363,232</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2.3935</td>
<td>397,212,119</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4.6555</td>
<td>406,254,155,100</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>91.484</td>
<td>402,227,135,78,46</td>
<td>5</td>
</tr>
<tr>
<td>TSINM</td>
<td>1.745</td>
<td>517</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>1.832</td>
<td>548,224</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1.917</td>
<td>521,289,165</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2.155</td>
<td>532,302,130,75</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>3.053</td>
<td>490,274,158,103,64</td>
<td>5</td>
</tr>
<tr>
<td>Drift_Gas</td>
<td>6.5471</td>
<td>72</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>5.07398</td>
<td>91,63</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>7.2962</td>
<td>79,52,37</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>11.7081</td>
<td>75,78,38,23</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>23.8846</td>
<td>76,42,35,22,18</td>
<td>5</td>
</tr>
<tr>
<td>Face</td>
<td>11.133</td>
<td>515</td>
<td>1</td>
</tr>
<tr>
<td>recognition</td>
<td>10.3421</td>
<td>437,260</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>11.9585</td>
<td>506,261,140</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>11.3072</td>
<td>454,261,148,73</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>13.271</td>
<td>445,250,156,189,48</td>
<td>5</td>
</tr>
</tbody>
</table>


