Abstract— This paper presents a static and dynamic economic dispatch study in electrical power systems using the artificial intelligence, or more precisely: the neural networks. Starting first of all by the load flow study to get an idea about the total demand, generation and losses, moving to the optimization of the power flow using the gradient method and the neural network as an modern option. Of course system constraints are included such as line losses and generators limits. Furthermore, a comparison between the results using the above mentioned methods is carried out at the end of this paper.

Index Terms— Economic Load Dispatch, Neural networks, Artificial Intelligence, Load flow.

I. INTRODUCTION

The economic dispatch problem is the determination of generation levels, in order to minimize the total generation cost for a defined level of load. It’s a kind of management for electrical energy in the power system in way to operate their generators as economically as possible [1].

In the other side we know that the factors having effects on the power generators cost are:

- Operating efficiencies of generators.
- Fuel cost.
- Transmission line losses.

Initially Neural networks objective was: patterns recognition, classification. Then it becomes very interesting in all domains.

II. LOAD FLOW STUDY -NEWTON RAPHSON-

The system of equations we need to study is the following [2]:

\[ P_i = V_i \sum_{j=1}^{n} V_j \left( \frac{g_{ij} \cos \delta_j + b_{ij} \sin \delta_j}{V_j} \right) \]

\[ Q_i = V_i \sum_{j=1}^{n} V_j \left( \frac{g_{ij} \sin \delta_j - b_{ij} \cos \delta_j}{V_j} \right) \]  

(1)

The aim is to get the phase and the magnitude of the voltage at any bus of generation. This can be obtained by using the jacobian matrix

\[
\begin{bmatrix}
    \frac{\partial P_1}{\partial \delta_1} & \frac{\partial Q_1}{\partial \delta_1} \\
    \frac{\partial P_2}{\partial \delta_2} & \frac{\partial Q_2}{\partial \delta_2} \\
    \vdots & \vdots \\
    \frac{\partial P_n}{\partial \delta_n} & \frac{\partial Q_n}{\partial \delta_n}
\end{bmatrix}
\]

\[
\begin{bmatrix}
    \Delta V_1 \\
    \Delta V_2 \\
    \vdots \\
    \Delta V_n
\end{bmatrix}
\]

\[
\begin{bmatrix}
    \frac{\partial P_1}{\partial V_1} & \frac{\partial Q_1}{\partial V_1} \\
    \frac{\partial P_2}{\partial V_2} & \frac{\partial Q_2}{\partial V_2} \\
    \vdots & \vdots \\
    \frac{\partial P_n}{\partial V_n} & \frac{\partial Q_n}{\partial V_n}
\end{bmatrix}
\]

\[
\begin{bmatrix}
    \Delta \delta_1 \\
    \Delta \delta_2 \\
    \vdots \\
    \Delta \delta_n
\end{bmatrix}
\]

(3)

(4)

(5)

III. ECONOMIC DISPATCH STUDY

III.1. The cost function

\[ F_i (P_{Gi}) = a_i + b_i P_{Gi} + c_i P_{Gi}^2 \]  

(6)

\[ F_i (P_{Gi}) : \text{is the function we need to optimize.} \]

Where:

- \( P_{Gi} \) = the real generated power in per unit on a common power base.
- \( F_i \) = the operating cost of unit in $/h.
- \( a_i, b_i \) and \( c_i \) are the cost coefficients of the generator \( i \).

Expressed in dollars per hour ($/h). [3][4][5][6]

III.2. Equality constraints

\[ \sum_{i=1}^{N} P_{gi} = P_D + P_L \]  

(7)

\[ P_D \]: total system demand.

\[ P_L \]: total system loss.

\[ N \]: total number of generators. [5][6][7]

III.3. Inequality constraints

\[ P_{i(min)} \leq P_i \leq P_{i(max)} \]  

(8)

\[ i = 1, \ldots, N \]
III.4. Losses formula

\[ P_L = \sum_{i=1}^{N} B_{ij} P_{Gi}^2 \]  

(9)

**B** \(_{ij}\)** are called the loss coefficients, which are assumed to be constant for a base range of load.

### III.5. Condition on the generated power

The Lagrange function can be constructed as shown bellow [10][11]:

\[ L = F_{Total} + \lambda(P_D + P_L - \sum_{i=1}^{N} P_{Gi}) \]  

(10)

Where: \( \lambda \) is called Lagrange multiplier. Or mathematically the incremental cost.

\[ \left( \frac{\partial L}{\partial P_{Gi}} = 0 \text{ and } \frac{\partial L}{\partial \lambda} = 0 \right) \]  

(11)

We can find the iterative compact form to get the minimum cost:

\[ P_i^{(k)} = \frac{\lambda^{(k)} - \lambda}{2(c_i + \lambda B_{Gi})} \]  

(12)

### IV. NEURAL NETWORKS

#### IV.1. Neuron Model

![Artificial neuron model](image)

**Fig.1 Artificial neuron model** [11]

\[ n = \sum_{j=1}^{R} w_{ij}p - b \]  

(13)

\[ a = f(\sum_{j=1}^{R} w_{ij}p - b) \]  

(14)

#### IV.2. Error correction learning

\[ Error_{global} = \frac{1}{2N} \sum_{p=1}^{P} \sum_{k=1}^{N} \left( y_k^{(p)} - d_k^{(p)} \right)^2 \]  

(15)

\[ \Delta w_{kj} = -\eta \nabla w_{kj} (Error_{global}) \Rightarrow \Delta w_{kj} = \eta y_j (d_k - y_k) \]  

(16)

Where, \( \Delta w_{kj} \) is the weight variation of the connection between the neurons \( j \) from the anterior layer and the output layer node \( k \).

\( d_k \): The desired neuron output \( k \), \( y_j \) and \( y_k \) are the output values produced in the neuron \( i \) and \( k \) respectively [11]

\[ w_{kj}^{\text{actual}} = \Delta w_{kj} + w_{kj}^{\text{anterior}} \]  

(17)

### V. APPLICATION AND COMPARISON

#### V.1. Results

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \phi )</th>
<th>( Delt (\degree) )</th>
<th>( P_g ) MW</th>
<th>( Q_g ) MVAR</th>
<th>( P_l ) MW</th>
<th>( Q_l ) MVAR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.06</td>
<td>0.000</td>
<td>260.983</td>
<td>-20.498</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>1.04</td>
<td>-5.503</td>
<td>40.000</td>
<td>47.014</td>
<td>21.70</td>
<td>12.700</td>
</tr>
<tr>
<td>3</td>
<td>1.02</td>
<td>-8.039</td>
<td>0.000</td>
<td>0.000</td>
<td>2.400</td>
<td>1.200</td>
</tr>
<tr>
<td>4</td>
<td>1.01</td>
<td>-9.701</td>
<td>0.000</td>
<td>0.000</td>
<td>7.600</td>
<td>1.600</td>
</tr>
<tr>
<td>5</td>
<td>1.01</td>
<td>-14.335</td>
<td>0.000</td>
<td>32.648</td>
<td>94.2</td>
<td>19.000</td>
</tr>
<tr>
<td>6</td>
<td>1.01</td>
<td>-11.451</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>7</td>
<td>1.00</td>
<td>-13.160</td>
<td>0.000</td>
<td>0.000</td>
<td>22.800</td>
<td>10.900</td>
</tr>
<tr>
<td>8</td>
<td>1.02</td>
<td>-12.235</td>
<td>44.012</td>
<td>22.800</td>
<td>30.000</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1.05</td>
<td>-14.451</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>10</td>
<td>1.04</td>
<td>-16.029</td>
<td>0.000</td>
<td>5.800</td>
<td>2.000</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1.07</td>
<td>-14.451</td>
<td>0.000</td>
<td>11.121</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>12</td>
<td>1.06</td>
<td>-15.323</td>
<td>0.000</td>
<td>0.000</td>
<td>11.200</td>
<td>7.500</td>
</tr>
<tr>
<td>13</td>
<td>1.07</td>
<td>-15.323</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>14</td>
<td>1.04</td>
<td>-16.209</td>
<td>0.000</td>
<td>10.137</td>
<td>6.200</td>
<td>1.600</td>
</tr>
<tr>
<td>15</td>
<td>1.04</td>
<td>-16.293</td>
<td>0.000</td>
<td>8.200</td>
<td>2.500</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>1.04</td>
<td>-15.893</td>
<td>0.000</td>
<td>3.500</td>
<td>1.800</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>1.04</td>
<td>-16.195</td>
<td>0.000</td>
<td>9.000</td>
<td>5.800</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>1.03</td>
<td>-16.895</td>
<td>0.000</td>
<td>3.200</td>
<td>0.900</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>1.02</td>
<td>-17.058</td>
<td>0.000</td>
<td>9.500</td>
<td>3.400</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>1.03</td>
<td>-16.858</td>
<td>0.000</td>
<td>2.200</td>
<td>0.700</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>1.03</td>
<td>-16.474</td>
<td>0.000</td>
<td>17.500</td>
<td>11.200</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>1.03</td>
<td>-16.460</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>1.03</td>
<td>-16.677</td>
<td>0.000</td>
<td>3.200</td>
<td>1.600</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>1.02</td>
<td>-16.846</td>
<td>0.000</td>
<td>8.700</td>
<td>6.700</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>1.02</td>
<td>-16.460</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>1.00</td>
<td>-16.875</td>
<td>0.000</td>
<td>3.500</td>
<td>2.300</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>1.03</td>
<td>-15.962</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>1.01</td>
<td>-12.120</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>1.01</td>
<td>-17.174</td>
<td>0.000</td>
<td>2.400</td>
<td>0.900</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>1.00</td>
<td>-18.043</td>
<td>0.000</td>
<td>10.600</td>
<td>1.900</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>300.983</td>
<td>124.434</td>
<td>283.400</td>
<td>126.200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Fig.2 Single line diagram of IEEE-30bus](image)

<table>
<thead>
<tr>
<th>Total demand (MW)</th>
<th>Inc cost ($/MWh)</th>
<th>Generation (MW)</th>
<th>Total Gen (MW)</th>
<th>Total Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>300.98 Classic</td>
<td>8.1245</td>
<td>52.3087 61.683</td>
<td>25.006 62.308</td>
<td>61.683 28.000</td>
</tr>
<tr>
<td>300.98 ANN</td>
<td>8.1243</td>
<td>52.3105 61.685</td>
<td>25.006 62.310</td>
<td>61.685 28.001</td>
</tr>
</tbody>
</table>

V.1.2. Dynamic economic dispatch using neural networks

- **Training phase**

  In our case we are studying a fitting problem this may explain the reason for which we are going to use Levenberg-Marquard algorithm.

   - The architecture of the neural network, so it contains two layers, the first one is hidden with two neurons and tangent sigmoid transfer function (tang). Whereas the second one is the output of the network containing seven neurons and a pure linear Transfer function (purelin).
   - The training algorithm is Levenberg-Marquard (trainlm).
   - The performance is the mean squared error, on matlab (mse)
   - Data division: divide random.
   - The convergence is satisfied at the ninth iteration number 100.

The best validation performance or the main squared error is about 0.0042872 at one hundred epochs. The total system (the training, validation and the test) is descending fast to reach the value mentioned earlier.

The regression is best when it reaches the value (1); and this is illustrated in figure (3), for the training, validation and the test.
After the training phase the network is ready to be used at real time. And this is the allocation of the generated power for a total demand varying from 909(MW) to 1589(MW).

<table>
<thead>
<tr>
<th>Total demand (MW)</th>
<th>T Gen (MW)</th>
<th>T Loss (MW)</th>
<th>T Cost ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>909</td>
<td>237.55</td>
<td>70.554</td>
<td>229.08</td>
</tr>
<tr>
<td>919</td>
<td>240.21</td>
<td>71.832</td>
<td>231.72</td>
</tr>
<tr>
<td>929</td>
<td>242.87</td>
<td>72.210</td>
<td>234.36</td>
</tr>
<tr>
<td>939</td>
<td>245.53</td>
<td>73.039</td>
<td>237.01</td>
</tr>
<tr>
<td>949</td>
<td>248.20</td>
<td>73.868</td>
<td>239.65</td>
</tr>
<tr>
<td>959</td>
<td>250.86</td>
<td>74.697</td>
<td>242.30</td>
</tr>
<tr>
<td>969</td>
<td>253.53</td>
<td>75.527</td>
<td>244.95</td>
</tr>
<tr>
<td>979</td>
<td>256.19</td>
<td>76.357</td>
<td>247.60</td>
</tr>
<tr>
<td>987</td>
<td>258.33</td>
<td>77.021</td>
<td>249.72</td>
</tr>
<tr>
<td>997</td>
<td>261.00</td>
<td>77.852</td>
<td>252.37</td>
</tr>
<tr>
<td>1006</td>
<td>263.40</td>
<td>78.600</td>
<td>254.76</td>
</tr>
<tr>
<td>1140</td>
<td>299.28</td>
<td>191.77</td>
<td>290.41</td>
</tr>
<tr>
<td>1280</td>
<td>336.96</td>
<td>214.54</td>
<td>327.84</td>
</tr>
<tr>
<td>1309</td>
<td>344.78</td>
<td>219.26</td>
<td>335.62</td>
</tr>
<tr>
<td>1430</td>
<td>377.54</td>
<td>238.9</td>
<td>368.16</td>
</tr>
<tr>
<td>1598</td>
<td>423.28</td>
<td>266.09</td>
<td>413.61</td>
</tr>
</tbody>
</table>

And we can see now the total losses, generation and cost.

VI. CONCLUSION

In this paper a static and a dynamic economic dispatch system has been developed using the artificial intelligence or more precisely the neural networks. This method has been tested on a test grid IEEE 30bus. Results obtained show that the neural networks method give better results than the classical method. Results shown previously illustrate us clearly the exactitude and the velocity of the neural network. This optimization on time is in another word an optimization on fuel.

VII. REFERENCES

[10] Lukman K. Walshf,R. Blackburn, "Loss Minimization In Industrial Power System Operation", School of Electrical and Telecommunication Engineering University of New South Wales,KensingtonSydney, NSW 2052, Australia.