Neural networks to estimate pollutant levels in canyon roads

S. Amoroso & M. Migliore
Institute for Transport Studies, Palermo University, Italy.

Abstract

Many data can be exchanged between different places and in short time by using the new technology. New researches are developing in urban traffic management in order to elaborate data coming from the network in real time and to suggest on-line solutions of many problems.

Also in the environmental field the new technology can be used to develop an on-line air quality monitoring system by elaborating data.

In this paper neural networks have been used in order to develop a model able to estimate in real time the roadside pollutant levels in a canyon road by elaborating weather and traffic conditions.

Neural networks can carry out quantitative estimations in real time and with a higher precision than classical analytical models.

The survey is done in a Sicilian town. The vehicular flows, desegregated in different categories, and the flow average speed have been recorded by using videocameras. The weather conditions and the roadside pollutant levels in the canyon road have been recorded by the instrumented city facilities of the local authority.

Then the training of a multilayered feed-forward neural network using the back propagation algorithm is carried out. A comparison between the neural network results and the analytical model results is carried out.

1 Introduction

The quantitative estimation of roadside pollutant levels is very complex; in fact many variables influence this phenomena.

As an example it can be mentioned: different vehicle typologies (for example vehicle with particular antipollution devices and different for the fuel used), en-
engine's temperature, maintenance level of engines and antipollution devices, different cinematic conditions (André, [2], [3]), urbanistic structure of the site examined, weather conditions. Therefore it is important to develop scientific tools able to predict roadside pollutant levels in order to apply the best strategy to improve air quality in towns and to obtain atmospheric pollution levels consistent with laws in force (Crotti, [6]).

In the last years European research projects (Eggleston, [7]) have studied the problem giving emission factors for different vehicle categories and for different cinematic conditions. Many models have been also developed to estimate vehicle pollutant emissions and atmospheric pollution levels (Kohoutek, [11]), (Reynolds, [13]).

When these models are applied in our urban areas, there are consistent differences between forecasting and real pollutant levels (Festa, [8]). In fact, pollutant levels are very often influenced from local variables (composition and maintenance level of vehicles running in the town, driver behaviour, particular weather conditions of the town) difficult to be estimated or modelled by analytical formulas.

In this paper neural networks are used to correlate roadside pollutant levels (especially those concerning primary pollutants more directly correlated to vehicle traffic) with vehicle flow stratified for categories, vehicle average speed and weather parameters.

The application was done in a Sicilian town (Caltanissetta) and in a canyon road where prescriptive standards regarding air quality have been significantly exceeded.

2. Neural networks

Neural networks are composed of many simple elements operating in parallel, taking inspiration from biological nervous system, whose operation is determined largely by the connection between elements. A neural network can be trained to perform a particular function by adjusting the values of the connections between elements (Cammarata, [5]), (Rizzo, [14]).

Neural networks have been trained to perform complex functions in various application fields.

Training and production are the essential phases of a neural network application. (figure 1).

In the first phase, the neural network is trained by using an error minimising algorithm for calibrating the connection weights.

In the second phase, it is possible to run the neural network, obtaining outputs also for data sets unseen.

The neural network generally includes different neural layers (input layer, output layer and hidden layers), as it can be seen in figure 2.

The two-layer sigmoid/linear (as activation functions for neurones in each layer) network can represent any functional relationship between inputs and outputs if the sigmoid layer has enough neurones (Hornik, [10]).
The backpropagation algorithm can train multilayered feed-forward networks (where a neuron in a layer is linked only with the other neurons of a successive layer) with differentiable activation functions to perform function approximation. In particular for backpropagation, the training continues until the MSE (mean-square-error) designed for outputs (difference between predicted value and real value) is met, or the prefixed number of epochs has occurred.

![Diagram of neural network](image)

**Figure 1:** The essential phases of a neural network application: training and production

![Diagram of neural network](image)

**Figure 2:** The architecture of a multi-layered feed-forward neural network.
The architecture of a backpropagation network is not completely constrained by
the problem to be solved. The number of inputs to the network is constrained by
the problem, and the number of neurones in the output layer is constrained by the
number of outputs required by the problem.
However, the number of layers between network inputs and the output layer and
the sizes of the layers depend on the designer’s decision (using a test set for the
calibration of the network architecture and weights of the network).
In the training of the network is important to compare the forecast performance
on the test set to exclude an overfitting to the training set and, then, to test the
capability of the network to generalise the results (Bishop, [4]).

3. The application in Caltanissetta

Since June 99 we started a survey campaign in a canyon road of Caltanissetta
(Sicily), in order to record data about link flow (desegregated for categories), av-
erage speed, weather parameters and pollutant levels.
In order to record data about link flow and average speed, we have used two
video cameras. The data about weather have been carried out from a town facil-
ity that records temperature, relative humidity, solar radiation, wind direction
and wind speed every minute. The data about pollutant levels have been carried
out from a town facility that, in the same link of our surveys, records CO and
PM10 levels every minute.
Up to now about 120 data have been taken. The principal aim of this work has
been the building of a neural network able to predict the average pollutant level
per hour in canyon roads from a set of inputs (motorcycles, heavy vehicles, cars,
flow average speed, temperature, solar radiation, wind direction and wind speed).
Using the above 120 data, different neural networks have been designed for each
pollutant (CO and PM10).
For each neural network, a training set and a test set have been chosen by ran-
don extraction from experimental data. The inputs and the output are referred to
average value per hour.
For CO, after the training, a neural network has been chosen with 8 neurones in
the input layer, 4 neurones in the hidden layer, a sigmoidal activation function
for the hidden neurones, 1 neurone in the output layer and a linear activation
function for the output neurone.
For PM10, after the training, a neural network has been chosen with 8 neurones
in the input layer, 4 neurones in the hidden layer, a sigmoidal activation function
for the hidden neurones, 1 neurone in the output layer and a linear activation
function for the output neurone.
The Levenberg-Marquardt algorithm trained these multilayer feed-forward net-
works by using the Neural Network Matlab® Toolbox.
The Levenberg-Marquardt rule for updating parameters (such as weights and bi-
ases) is:

\[ \Delta W = (J^T J + \mu I)^{-1} J^T e \]
where \( J \) is the Jacobian matrix of derivatives of each error to each weight, \( \mu \) is a scalar, \( e \) is an error vector. If the scalar \( \mu \) is very large, the above expression approximates the gradient descent method, while if it is small the above expression becomes the Gauss-Newton method. The Gauss-Newton method is faster and more accurate near an error minimum, so the aim is to shift towards the Gauss-Newton method as quickly as possible. Thus, \( \mu \) is decreased after each successful step and increased only when a step increases the error. If \( \mu \) becomes too large no learning takes place and then an error minimum has been found. For this reason learning also stops when \( \mu \) reaches a maximum value [1].

After the training and the validation of the neural networks, classical analytical models have been used in order to carry out comparisons between these different techniques.

The emission levels have been estimated by using analytical formulas (Tartaglia, [15]) where the emission of pollutants depends from average speed, vehicular flow, desegregated for categories, and temperature. The levels of pollutants have been estimated by using the canyon formula (Hoydish, [9]). A calibration process of the experimental coefficients of the analytical formulas has been carried out by using the experimental data and the Gauss-Newton method as optimisation algorithm, in order to minimise the mean square error between the observed and calculated levels.

The comparison carried out, graphical and statistical (Lombardo, [12]), is reported in the figg. 3, 4 and in the tables 1 and 2.

As it is shown, the neural networks have understood the complexity of the problem better than the classical models examined. They have also approximated the correlation among the variables involved in the process.

Certainly the neural networks are slow in the training phase but very quick in the production of the output. Therefore, they are a useful tool in the forecast problems, because they can work in real time with a higher precision than classical models.

**Table 1: Statistical comparison among CO (mg/mc) observed values, estimations by neural network and by classical model**

<table>
<thead>
<tr>
<th>CO</th>
<th>Mean observed</th>
<th>Neural network</th>
<th>Classical model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>2,982</td>
<td>2.811</td>
<td>2,466</td>
</tr>
<tr>
<td>%RMSE</td>
<td>30 %</td>
<td>51 %</td>
<td></td>
</tr>
<tr>
<td>Pearson</td>
<td>0,78</td>
<td>0,53</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2: Statistical comparison among PM10 (\( \mu g/mc \)) observed values, estimations by neural network and by classical model**

<table>
<thead>
<tr>
<th>PM10</th>
<th>Mean observed</th>
<th>Neural network</th>
<th>Classical model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>48,08</td>
<td>48,08</td>
<td>43,77</td>
</tr>
<tr>
<td>%RMSE</td>
<td>22 %</td>
<td>57 %</td>
<td></td>
</tr>
<tr>
<td>Pearson</td>
<td>0,93</td>
<td>0,55</td>
<td></td>
</tr>
</tbody>
</table>
4. Conclusions

In this paper an experimental application in a Sicilian town was showed in order to evaluate the results obtained applying a neural network to correlate atmospheric pollution levels with flow stratified, average speed and weather parameters. A comparison between neural networks and classical models was also carried out.

The first results were quite good above all for PM10, but it is necessary to increase the experimental data set to improve the training of the neural network. The quality of the inputs is also important. In fact, the level of pollutants depends from many variables, that are not been included in this work because difficult to be estimated in real time or modelled (composition and maintenance level of vehicles running in the road, driver behaviour, particular weather conditions of the road).
Certainly, a higher number of input variables could explain the gap between estimated and measured values.

As preliminary conclusions of this first phase of the work, the neural networks have carried out estimations of pollutant levels in the experimental canyon road with a higher level of accuracy than examined classical models.

As further work, it could be possible to apply other neural network architectures to carry out a first input clustering in order to improve the forecast performances of neural model.

It is also necessary to do other surveys in other urban areas (intersections, squares, etc...) to generalise the results in other contexts.

Finally, it is also important to develop a tool able to use the forecast model in order to search the best strategies to reduce atmospheric pollution levels and to improve air quality in our towns.

References


