# Optimization of Traffic Lights Using Supervised Learning

D.S. Carrilho[1,b)], C. Coelho[2,a)], M. Fernanda P. Costa[2,c)], L.L. Ferrás[2,d)] and A.J. Soares[2,e)]

[1]*Accenture Technology Center, Braga, Protugal*
[2]*Centre of Mathematics, University of Minho, Campus de Gualtar, 4710 - 057 Braga, Portugal*

[a)]Corresponding author: ceciliaeduarda58@gmail.com
[b)]diana.sofia.carrilho@gmail.com
[c)]mfc@math.uminho.pt
[d)]luislimafr@gmail.com
[e)]ajsoares@math.uminho.pt

**Abstract.** With the increasing number of vehicles, conventional traffic lights with fixed times are unable to prevent traffic congestion at intersections [1]. In this work, an intelligent traffic simulator is developed based on a Supervised Learning approach using artificial neural networks. The simulator makes decisions about which traffic light should be activated to minimise waiting times and maximise traffic flow, resulting in an intelligent management of green and red times.

## INTRODUCTION

This work aims to develop an intelligent solution using artificial neural networks, to reduce drivers' average waiting time and make the traffic at an intersection (with four traffic lights) smoother, thus avoiding traffic jams. The prototype intersection consists of vehicles located in the four lanes (Figure 1 (a)), represented by $L_i$, where $i = 1, ..., 4$. To solve this problem, we developed a traffic simulator and a neural network for real-time optimisation (called Intelligent Traffic Simulator - ITS).

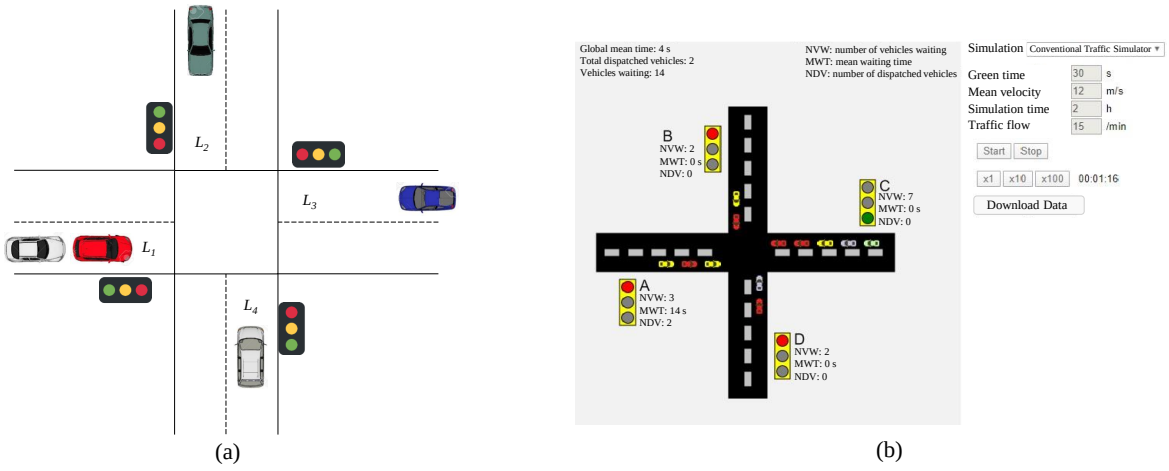
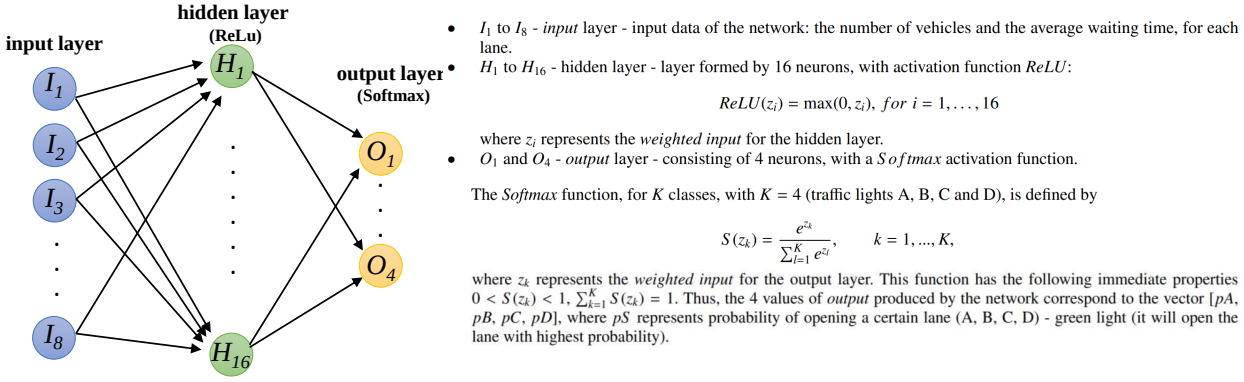



**FIGURE 1.** (a) Schematic of an intersection; (b) Conventional Traffic Simulator (CTS). The traffic can be seen in real time.

Before the development of the Intelligent Traffic Simulator (ITS), we developed a simple, fixed-time, traffic simulator - the Conventional Traffic Simulator (CTS) (see Figure 1 (b)). The CTS allows to model the behaviour

of each vehicle throughout the simulation time. To develop this simulator, we used the $p5.js$ library written in the $JavaScript$ language, as well as HTML and CSS. With these programming tools, it was possible to implement a dynamic graphical interface that represents both the intersection under study and the traffic flow. This simulator is based on the following assumptions: an intersection has four lanes; each traffic light has only two colours (green/red); when the traffic light for a given lane is green (the other three traffic lights become red), vehicles in that lane can travel in any direction; each green phase has a fixed time of 30 seconds; for each lane, there is a random process to generate the vehicles. By default, every 4 seconds a vehicle is randomly placed on one of the lanes.

**Intelligent Traffic Simulator:** To develop the intelligent simulator, we used *ML5.js*, a high-level *Machine Learning* library built on top of *TensorFlow.js*. It was decided to use an Artificial Neural Network (ANN) with supervised learning. The following figure illustrates the architecture of the ANNs used for the development of the intelligent simulator (together with details of the activation functions).




- $I_1$ to $I_8$ - *input* layer - input data of the network: the number of vehicles and the average waiting time, for each lane.
- $H_1$ to $H_{16}$ - hidden layer - layer formed by 16 neurons, with activation function $ReLU$:

$$ReLU(z_i) = \max(0, z_i), \ for \ i = 1, \ldots, 16$$

where $z_i$ represents the *weighted input* for the hidden layer.

- $O_1$ and $O_4$ - *output* layer - consisting of 4 neurons, with a $Softmax$ activation function.

The *Softmax* function, for $K$ classes, with $K = 4$ (traffic lights A, B, C and D), is defined by

$$S(z_k) = \frac{e^{z_k}}{\sum_{i=1}^{K} e^{z_i}}, \qquad k = 1, ..., K,$$

where $z_k$ represents the *weighted input* for the output layer. This function has the following immediate properties $0 < S(z_k) < 1$, $\sum_{k=1}^{K} S(z_k) = 1$. Thus, the 4 values of *output* produced by the network correspond to the vector [$pA$, $pB$, $pC$, $pD$], where $pS$ represents probability of opening a certain lane (A, B, C, D) - green light (it will open the lane with highest probability).

**FIGURE 2.** Architecture of the ANN used in this work. $I_1 - I_4$ are the number of cars in each lane and $I_5 - I_8$ are the mean waiting time for each lane. $O_1 - O_4$ are the traffics light to be activated (A, B, C or D)

To develop intelligent models, it was necessary to train the neural networks with the so called training datasets. To create these datasets, we used a procedure similar to the CTS, but now there were no fixed-time periods for the green lights. Instead, the user plays the role of a traffic warden and determines which lane will be activated (turn green) based on his perception of congestion (the user looks at the average wait time and number of vehicles in each lane to make his decision), by using the keyboard keys (called traffic warden simulator- TWS).

To test the trained models, we developed a script that passes a set of decision rules to an algorithm that automates traffic light decision tasks - this algorithm mimics the TWS, by playing the role of the user. The *inputs* are the number of vehicles and the average waiting time per lane, and the *output* is the traffic light to be activated. This algorithm is used on the fly. That is, while the ITS is running, the model makes predictions and the result of its prediction is immediately compared to the result given by this algorithm. For more details, see [2].

## RESULTS and DISCUSSION

**Conventional Traffic Simulator:** To evaluate the dynamics of the conventional simulator (CTS), a simulation with a duration of 1 h was performed, with a fixed green time of 30 s, and a flow of 15 vehicles per minute with an average speed of 12 m/s (the results are shown in Table 1).

Note: Let $t_1, ..., t_n$ be the set of waiting times of vehicles crossing the intersection during a cycle (one cycle consists in the activation of all four traffic lights at least once) on a given lane. The average waiting time of a lane's vehicles is given by $T_{av} = \frac{1}{n}\left(\sum_{k=1}^{n} t_k\right)$, where $n$ is the number of vehicles passing through the intersection in one cycle.

It was found that the vehicles' average waiting time, $T_{av}$, varied between 60 s and 64 s for the four lanes and the maximum waiting time, $T_{max}$, is different for each lane. The fixed green time (30 s) was not sufficient to allow all waiting vehicles to cross the intersection, so $T_{av}$ increased in each lane and consequently the corresponding maximum waiting time, $T_{max}$. Regarding the average number of stops in each lane ($S_{av}$), it was found that there was a slight increase in lane C and consequently a longer average waiting time is verified. The maximum number of stops ($S_{max}$) in all four lanes was 1. Note that $S_{av}$ is defined in the same way as $T_{av}$. The total number of vehicles crossing the

intersection during the green phase, $N_{vd}$, was somewhat higher in lane D. These values, the average number of stops, the average waiting time, and the number of vehicles crossing the intersection, vary from lane to lane because random factors (such as the number of injections in the simulator) affect real traffic conditions.

**TABLE 1.** Results obtained for the CTS. Simulation: 1 h; Average speed: 12 m/s; Flux: 15 vehicles/min.

| Lane | $T_{av}$ | $T_{max}$ | $S_{av}$ | $S_{max}$ | $N_{vd}$ |
|---|---|---|---|---|---|
| **A** | 64 s | 145 s | 0.78 | 1 | 209 |
| **B** | 60 s | 119 s | 0.75 | 1 | 187 |
| **C** | 64 s | 126 s | 0.81 | 1 | 247 |
| **D** | 61 s | 121 s | 0.77 | 1 | 248 |

The data obtained from this simulation allowed us to develop a useful empirical function to predict the waiting time of a vehicle based on its distance from the traffic light:

$$f(P_{vehi}) = \frac{(P_{vehi} - 1) \times C_{tvei}}{V_{av}} + T_{reac} + (P_{vehi} - N_{gini}) \times T_{reac} + T_{gini}, \tag{1}$$

where $C_{vehi}$ is the length of each vehicle (a random integer value in the interval [5,6]), $V_{av}$ is the average speed (set to 12 m/s in this example), $D_{btb}$ is the bumper-to-bumper distance (a random decimal value in the interval [0.38,1.8]), $C_{tvehi}$ is the total length of the vehicle ($C_{tvei} = C_{vehi} + D_{btb}$), $T_{reac}$ is the reaction time in seconds (a random decimal value in the interval [1.5,2]), $T_{gini}$ represents the first 10 s of the total green time, $N_{gini}$ is the number of vehicles crossing the intersection in the first 10 s (a random integer value in the interval [3,4]), and $P_{vei}$ is the position (1,2,3,...) of the vehicle in the queue ($P_{vei} > N_{gini}$). This function is useful to predict the waiting time of a vehicle in real situations.

**Intelligent Traffic Simulator:** To evaluate the dynamics of the intelligent simulator, four training datasets were created, based on different weights assigned to the various parts of a queue. The idea is to take into account the influence of the driver's reaction delay when the vehicle in front starts moving. The weighted waiting times are given by $T_{av}$ and the formulas (2) and (3):

$$T_{av-50\%,vec1} = t_1 \times 0.5 + \frac{1}{n-1}\left(\sum_{k=2}^{n-1} t_i\right) \times 0.5, \quad T_{av-70\%\,vec1} = t_1 \times 0.7 + \frac{1}{n-1}\left(\sum_{k=2}^{n-1} t_i\right) \times 0.3. \tag{2}$$

$$T_{av-70\%,mf} = \frac{1}{\lceil n/2 \rceil}\left(\sum_{k=1}^{\lceil \frac{n}{2} \rceil} t_i\right) \times 0.7 + \frac{1}{n - \lceil n/2 \rceil}\left(\sum_{k=\lceil \frac{n+2}{2} \rceil}^{n} t_i\right) \times 0.3, \tag{3}$$

where $\lceil y \rceil$ represents the *ceiling* function. The datasets assume, for each queue: all vehicles are assigned the same weight, DTraining-$T_{av}$; the first vehicle is assigned a weight of 50% and the remaining 50%, Dtraining-$T_{av-50\%\,vec1}$; a weight of 70% to the first vehicle and 30% to the remaining, Dtraining-$T_{av-70\%\,vec1}$; a weight of 70% to the first half of the queue and 30% to the other half, DTraining-$T_{av-70\%\,mf}$.

The datasets were created manually by running a 30-minute manual simulation using TWS, for each. The cardinality of these sets was as follows: #DTraining-$T_{av}$ = 110, #DTraining-$T_{med-50\%,vec1}$ = 112, #DTraining-$T_{av-70\%,vec1}$ = 107, #DTraining-$T_{med-70\%,mf}$ = 108. In each training dataset, the attributes are the number of vehicles and the average waiting time in each of the 4 lanes, and the label is the traffic light to be activated. To train the intelligent model we have considered 200 epochs, a batch-size of 12, and a 0.2 learning rate. In the end we obtained four different neural network models, denoted by: Model-$T_{av}$, Model-$T_{av-50\%\,vec1}$, Model-$T_{av-70\%\,vec1}$, Model-$T_{av-70\%\,mf}$. As mentioned earlier, the tests were performed on the fly using a specially developed algorithm that mimics the decisions of the user/traffic warden.

To evaluate the performance of the four models, we used three metrics, namely *Accuracy* (Acc.), *Precision* (Pre.) and *Recall* (Rec.), for each class $i$, $i = 1, ..., 4$:

$$Accuracy_i = \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}, \quad Precision_i = \frac{TP_i}{TP_i + FP_i}, \quad Recall_i = \frac{TP_i}{TP_i + FN_i}. \tag{4}$$

where TP is True Positive, TN is True Negative, FP is False Positive, and FN is False Negative. Since we are considering a multi-class problem, we have that $TP_i = \sum_{j=1}^{4} x_{j,j}$, $TN_i = \sum_{j=1,j\neq i}^{4} \sum_{k=1,k\neq i}^{4} x_{j,k}$, $FP_i = \sum_{j=1,j\neq i}^{4} x_{j,i}$, $FN_i = \sum_{j=1,j\neq i}^{4} x_{i,j}$. The $x_{i,j}$ are the entries of the 4 by 4 multi-class matrix with predictions for classes A ($j = 1$), B ($j = 2$), C ($j = 3$), D ($j = 4$) and Ground Truth for classes A ($i = 1$), B ($i = 2$), C ($i = 3$), D ($i = 4$).

**TABLE 2.** Performance metrics for the models: Model-$T_{av}$, Model-$T_{av-50\%\,vec1}$, Model-$T_{av-70\%\,vec1}$, Model-$T_{av-70\%\,mf}$.

| | Model-$T_{av}$ | | | Model-$T_{av-50\%\,vec1}$ | | | Model-$T_{av-70\%\,vec1}$ | | | Model-$T_{av-70\%\,mf}$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Lane | *Acc.* | *Pre.* | *Rec.* | *Acc.* | *Pre.* | *Rec.* | *Acc.* | *Pre.* | *Rec.* | *Acc.* | *Pre.* | *Rec.* |
| **A** | 94.05% | 0.83 | 0.83 | 95.09% | 0.88 | 0.92 | 97.81% | 0.93 | 0.98 | 95.45% | 0.88 | 0.94 |
| **B** | 91.45% | 0.85 | 0.81 | 98.49% | 1.0 | 0.94 | 95.99% | 0.85 | 0.98 | 93.01% | 0.90 | 0.84 |
| **C** | 93.68% | 0.97 | 0.81 | 95.09% | 0.88 | 0.94 | 93.07% | 0.97 | 0.79 | 95.1% | 0.97 | 0.85 |
| **D** | 94.08% | 0.83 | 0.97 | 96.23% | 0.95 | 0.90 | 96.35% | 0.92 | 0.94 | 94.06% | 0.80 | 0.95 |

Analysing the overall results presented in Table 2, we see that the performance values obtained for the models Model-$T_{av-50\%\,vec1}$, Model-$T_{av-70\%\,vec1}$ and Model-$T_{av-70\%\,mf}$ are high and very close to each other. The worst results are obtained for the Model-$T_{av}$ (equally distributed weights). For the four models, the mean waiting time (determined from Equation 1) was $\approx 36s$ in the 4 lanes. The number of vehicles that cross the intersection during the green phase was: 899 for Model-$T_{av}$, 914 for Model-$T_{av-50\%\,vec1}$, 917 for Model-$T_{av-70\%\,vec1}$ and 948 for Model-$T_{av-70\%\,mf}$. This shows that Model-$T_{av-70\%\,mf}$ was more efficient than the other three models.

## CONCLUSIONS

The intelligent simulator proved to be much more efficient than the fixed-time simulator, resulting in a reduction in vehicle's average waiting time of approximately 50%. The model Model-$T_{av-70\%\,mf}$ allowed more vehicles to cross the intersection than any other model.

The results obtained are promising and in the near future we will:

- perform more simulations with more diverse input parameters, i.e. using different values for average speed and traffic flow in order to obtain a more comprehensive overview of the functioning and performance of the models;
- apply this study to an actual intersection. To do this, it would be necessary to conduct a study of traffic behaviour at a specific intersection in a city, in order to collect real traffic data and consequently train the models proposed here using data from that intersection.

Finally, it should be noted that the implementation of a traffic simulator is not trivial, as it is a very complex and lengthy process that requires the performance of extensive studies to concretely capture all the dynamics of traffic conditions.

Regardless of how much technology is available for optimising traffic lights, we must always take into account the urban design of a city as a whole, from the architecture of the street on which we want to set up smart traffic lights to the type of vehicles that travel on it.

## ACKNOWLEDGMENTS

The authors acknowledge the funding by Fundação para a Ciência e Tecnologia (Portuguese Foundation for Science and Technology) through projects UIDB/00013/2020 and UIDP/00013/2020.

## REFERENCES

[1] A. K. Martin Treiber, *Traffic Flow Dynamics: Data, Models and Simulation* (Springer, 2013).

[2] D. Carrilho, *Smart Traffic Signals*, Master's thesis, University of Minho - Department of Mathematics, R. da Universidade, 4710-057 Braga (2021).