Netlogo reinforcement-learning model for traffic simulation:   
  
This model models the movement of cars on a highway. Each car follows a simple set of rules: it slows down (decelerates) if it sees a car close ahead and speeds up (accelerates) if it doesn't see a car ahead. The model demonstrates how traffic jams can form even without any accidents, broken bridges, or overturned trucks. No "centralized cause" is needed for a traffic jam to form. Originally the model was designed to handle only one lane highway and one type of car, this has been improved, the simulation now takes place on three lanes and with three different types of cars, each with its velocity, acceleration and deceleration rates.

Requirements:

To run this model it is necessary to install:

* Python (version 3.9)
* numpy (version 1.24)
* Netlogo Desktop
* Tensorflow (version 1.13)
* Keras (version 2.14)

These dependencies can also be installed all together running “pip install -r” on the requirements file in the project folder.

How to set up:

Cick on the SETUP-TF button to setup the python environment for the simulation.

Click on the SETUP button to set up the cars.

Set the NUMBER-OF-CARS slider to change the number of cars on the road.

Click on GO to start the cars moving. Note that they wrap around the world as they move, so the road is like a continuous loop.

The ACCELERATION slider controls the rate at which cars accelerate (speed up) when there are no cars ahead.

When a car sees another car right in front, it matches that car's speed and then slows down a bit more. How much slower it goes than the car in front of it is controlled by the DECELERATION slider.

Things to notice:

Traffic jams can start from small "seeds." These cars start with random positions and random speeds. If some cars are clustered together, they will move slowly, causing cars behind them to slow down, and a traffic jam forms.

Even though all of the cars are moving forward, the traffic jams tend to move backwards. This behavior is common in wave phenomena: the behavior of the group is often very different from the behavior of the individuals that make up the group.

The “Car speeds” plot shows two values as the model runs:

* the fastest speed of any car (this doesn't exceed the speed limit!)
* the slowest speed of any car

Notice not only the maximum and minimum, but also the variability -- the "jerkiness" of one vehicle.

The “selected-actions” plot shows the number of cars that choose one of the three actions through the simulation:

* decelerate (red)
* maintain velocity (yellow)
* accelerate (green)

Notice that the default settings have cars decelerating much faster than they accelerate. This is typical of traffic flow models.

Even though both ACCELERATION and DECELERATION are very small, the cars can achieve high speeds as these values are added or subtracted at each tick.

The “Emissions over time” plot shows:

* The sum of the emissions of the cars on the lanes (red)
* The average of the emissions of the cars through the simulation (green)

Things to try:

In this model there are three sliders that can affect the tendency to create traffic jams: the initial NUMBER-OF-CARS, ACCELERATION, and DECELERATION.

Look for patterns in how these settings affect the traffic flow. Which variable has the greatest effect? Do the patterns make sense? Do they seem to be consistent with your driving experiences?

Set DECELERATION to zero. What happens to the flow? Gradually increase DECELERATION while the model runs. At what point does the flow "break down"?

Two other interesting parameters to play with are the exploration rate and the exploration rate decay which both determine how random the search for the best action is.