Artificial Superintelligence: Machine Consciousness Implementation Based On Computational Theory.

Poondru Prithvinath Reddy

June 22, 2020

Artificial Superintelligence : Machine Consciousness Implementation Based On Computational Theory.

Poondru Prithvinath Reddy

ABSTRACT

Artificial Super Intelligence or ASI that is more potent and refined than human's intelligence. ASI is based on the ideas that machines can imitate the human mind, their way of working to the extent that they can even supersede them. As a first step, ASI aims to improve the cognitive abilities of the machines and to achieve this, the ASI will have to become more conscious. In this paper, we define and implement machine consciousness as intelligence deduced by memory containing all information collected by sensors(or senses) and we have taken Self-Driving Car as "Artificially" created conscious object without any "living" properties. We have implemented Obstacle Detection using deep learning and Object Prediction using Kalman Filter, both relates to environment perception, for Self-Driving Car where it would have a variety of sensory inputs and be aware of its surroundings. The test results are encouraging and Self-Driving Car is therefore "conscious" of the given order since it has obeyed. So, the results show that it is possible to build intelligent machines artificially that are conscious.

INTRODUCTION

"**Superintelligence**" refers to the idea that steady advances in artificial intelligence, or machine (computer) intelligence, might one day result in creating a machine vastly superior to humans in reasoning and decision-making abilities.

Consciousness. Neuroscience hypothesizes that **consciousness** is generated by the interoperation of various parts of the brain, called the neural correlates of **consciousness** or NCC, though there are challenges to that perspective.

Consciousness - The state of being aware of and responsive to one's surroundings. **Consciousness** refers to one's individual awareness of own unique thoughts, memories, feelings, sensations, and environment. Individual **consciousness** is awareness of oneself and the world around him. This awareness is subjective and unique to individual, and **conscious** experiences are constantly shifting and changing.

Artificial Super Intelligence or ASI that has the capability to perform the tasks that are impossible for the human mind to think or do. It is that aspect of intelligence that is more potent and refined than a human's intelligence. Superintelligence is capable of outperforming human intelligence; it is extremely powerful in doing that. The human brain is made of neurons and is limited to some billion neurons. Superintelligence, therefore challenges this trait, which knows no limit.

The road to endless possibilities of Artificial Super Intelligence is paved by the ideas that machines can imitate the human mind, their way of working to the extent that shortly they can even supersede them. Under these circumstances, it is inevitable that ASI will be much better in concluding tasks that humankind would fail to achieve, and will function in better ways compared to the human.

In its first step, Artificial Super Intelligence aims to improve the cognitive abilities of the machine. In the future, the ASI will become more conscious, self-sustainable, and self-learning, developing, and improving constantly.

Every machine and every living thing is conscious. Consciousness is a memory at the disposal of the intelligence containing all informations collected by sensors (or senses) and those already deduced by intelligence about the moment. A machine starts, stops, manufactures, when asked. It is therefore "conscious" of the given order since it has obeyed. And it is conscious at every moment otherwise a security stops it. Machines that are conscious will also be possible to build. Artificial Intelligence would indeed be intelligent, and possibly conscious as well. It would have to have a variety of sensory inputs and be aware of it's environment. This is the stuff of "Artificial" fixes a created object by Human Being without any "living" properties at the moment, and likely for the near future as well.

METHODOLOGY

A **self-driving car**, also known as a **robot car**, a**utonomous car**, or **driverless car**, is a vehicle that is capable of sensing its environment and moving with little or no human input. Autonomous cars combine a variety of sensors to perceive their surroundings, such as radar, Lidar, sonar, GPS, odometry and inertial measurement units. Advanced control systems interpret sensory information to identify appropriate navigation paths, as well as obstacles .

A car capable of autonomous driving should be able to drive itself without any human input. To achieve this, the autonomous car needs to sense its environment, navigate and react without human interaction. A wide range of sensors, such as LIDAR, RADAR, GPS, wheel odometry sensors and cameras are used by self-driving cars to perceive their surroundings. In addition, the autonomous car must have a control system that is able to understand the data received from the sensors and make a difference between traffic signs, obstacles, pedestrian and other expected and unexpected things on the road .

For a machine to be called a robot, it should satisfy at least three important capabilities: to be able to sense, plan, and act. Self-driving cars are essentially robot cars that can make decisions about how to get from point A to point B. The passenger only needs to specify the destination, and the autonomous car should be able to take him or her there safely.

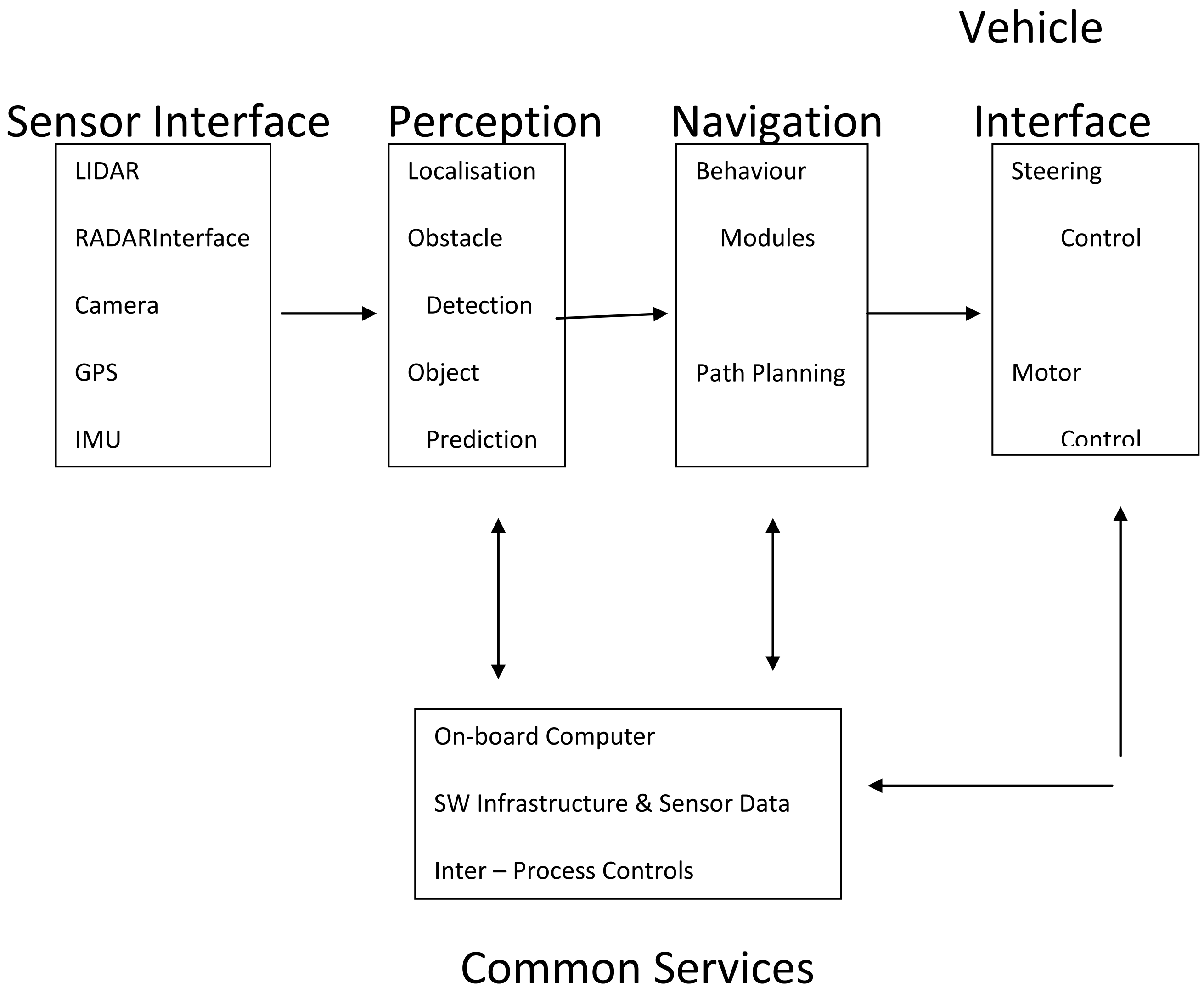


Figure 1 Illustrates the SW block diagram of the standard self-driving car.

Each block seen in Figure 1 can interact with other blocks using inter-process communication (IPC) and identified essential blocks for the SW block diagram of a typical self-driving car.

Perception modules. These modules process perception data from sensors such as LIDAR, RADAR and cameras, then segment the processed data to locate different objects that are staying still or moving.

Environment perception

Environment perception is the key module of a self-driving car. To provide necessary information for a car's control decision, the car is required to independently perceive surrounding environment. The major methods of environment perception include laser navigation, visual navigation and radar navigation.

During environment perception, multi-sensors are deployed to sense the comprehensive information from the environment, which are then fused to perceive the environment. Among the sensors, the laser sensor is utilized for bridging between the real world and data world, radar sensor is used for distance perception and visual sensor is for traffic sign recognition. The self-driving car fuses data from laser sensors, radar sensors and visual sensors, and generates the surrounding environment perception, such as road edge stone, obstacles, road marking and so on.

By measuring the reflection time, reflection signal strength and the data of target point can be generated, then the testing object information, such as location (distance and angle), shape (size) and state (velocity and attitude) can be calculated out.

Synergistic Combining of Sensors

All the data gathered by these sensors is collated and interpreted together by the car's CPU or in built software system to create a safe driving experience.

Programmed to Interpret Common Road Signs

The software has been programmed to rightly interpret common road behaviour and motorist signs. For example, if a cyclist gestures that he intends to make a manoeuvre, the driverless car interprets it correctly and slows down to allow the motorist to turn. Predetermined shape and motion descriptors are programmed into the system to help the car make intelligent decisions. For instance, if the car detects a 2 wheel object and determines the speed of the object as 10mph rather than 50 mph, the car instantly interprets that this vehicle is a bicycle and not a motorbike and behaves accordingly. Several such programs fed into the car's central processing unit will work simultaneously, helping the car make safe and intelligent decisions on busy roads.

Object detection is a computer technology related to computer vision and image processing that detects and defines **objects** such as humans, buildings and cars from digital images and videos. This technology has the power to classify just one or several **objects** within a digital image at once

Object detection is simply about identifying and locating all known **objects** in a scene. **Object tracking** is about locking onto a particular moving **object**(s) in real-time. **Objects** can be tracked based solely on motion features without knowing the actual **objects** being tracked.

KALMAN FILTER allows us to model tracking based on position and velocity of an object and predict where it is to be. **Kalman filtering** (KF) is widely used to **track** moving **objects**, with which we can estimate the velocity and even acceleration of an **object** with the measurement of its locations. However, the accuracy of KF is dependent on the assumption of linear **motion** for any **object** to be tracked.

Mathematical Formulation of Kalman Filter

The Kalman filter addresses the problem of trying to estimate the state $n\ x \in \Re^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$x_k = Ax_{k-1} + Bu_k + w_{k-1}$

with a measurement $y \in \Re^m$ that is

$y_{k=} Hx_k + v_k$

The random variables w_k and v_k represent the process and measurement noise respectively. They are assumed to be independent of each other and with normal probability distributions

$p(w) \approx N(0, Q)$

$p(v) \approx N(0, R)$

In practice, the process noise covariance Q and measurement noise R covariance matrices might change with each time step or measurement, however here we assume they are constant. The $n \times n$ matrix A relates the state at the previous time step to the state at the current step, in the absence of either a driving function or process noise. The $n \times 1$ matrix B relates the optional control input $u \in \Re^{l}$ to the state x. The $m \times n$ matrix H in the measurement equation relates the state to the measurement y_k .

The Methodology essentially consists of the following related to Environment Perception:-

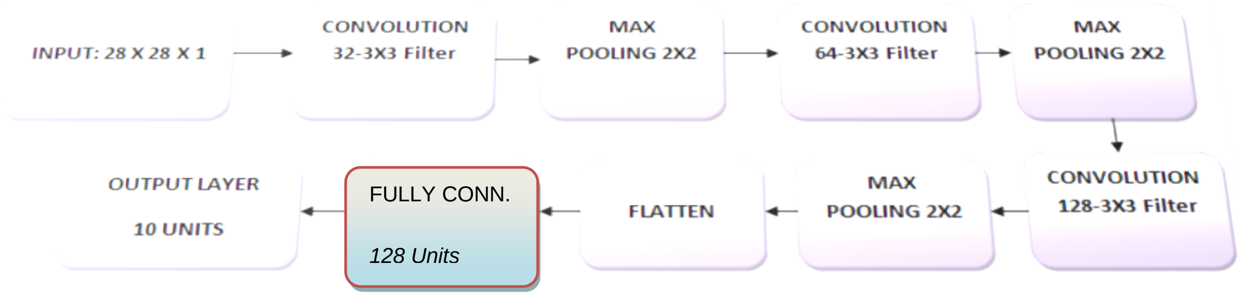
1. Obstacle Detection using Deep Learning
2. Object Prediction using Kalman Filter

ARCHITECTURE

IMAGE CLASSIFICATION USING DEEP LEARNING

Classification and object detection are the main parts of computer vision. Classification is finding what is in an image. When performing standard **image classification**, given an input image, we present it to our neural network, and we obtain a single class label and perhaps a probability associated with the class label as well.

This class label is meant to characterize the contents of the entire image, or at least the most dominant, visible contents of the image.



We have already heard of image or facial recognition or self—driving cars. These are real-life implementations of Convolutional Neural Networks (CNNs). We implement these deep, feed-forward artificial neural networks by overcoming overfitting with the regularization technique called “dropout”.

We have used the MNIST dataset for training and testing the image processing. The **MNIST database** (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems. The MNIST database contains 60,000 training images and 10,000 testing images. To load the data, we first need to download the data from the link and then structure the data in a particular folder format to be able to work with it.

From the above, we can see that the training data has a shape of 60000 x 784: there are 60,000 training samples each of 784-dimensional vector. Similarly, the test data has a shape of 10000 x 784, since there are 10,000 testing samples.

The 784 dimensional vector is nothing but a 28 x 28 dimensional matrix. That's why we will be reshaping each training and testing sample from a 784 dimensional vector to a 28 x 28 x 1 dimensional matrix in order to feed the samples in to the CNN model.

As a first step, we convert each 28 x 28 image of the train and test set into a matrix of size 28 x 28 x 1 which is then fed into the network.

The Deep Artificial Neural Network

We used three convolutional layers:

- The first layer will have 32-3 x 3 filters,
- The second layer will have 64-3 x 3 filters and
- The third layer will have 128-3 x 3 filters.

In addition, there are three max-pooling layers each of size 2 x 2.

We used a RELU as our activation function which simply takes the output of max_pool and applies RELU.

Flattening layer:

The Output of a convolutional layer is a multi-dimensional Tensor. We want to convert this into a one-dimensional tensor. This is done in the Flattening layer. We simply used the reshape operation to create a single dimensional tensor.

Fully connected layer:

Now, let’s define and create a fully connected layer. Just like any other layer, we declare weights and biases as random normal distributions. In fully connected layer, we take all the inputs, do the standard z=wx+b operation on it. The Fully Connected Layer has 128 Neurons.

We added Dropout into the network to overcome the problem of overfitting to some extent and also to improve the training and validation accuracy. This way, turning off some neurons will not allow the network to memorize the training data since not all the neurons will be active at the same time and the inactive neurons will not be able to learn anything.

OBJECT PREDICTION USING KALMAN FILTER

Imagine about a self-driving car and we are trying to localize its position in an environment. The sensors of the car can detect cars, pedestrians, and cyclists. Knowing the location of these objects can help the car make judgements, preventing collisions. But on top of knowing the location of the objects, the car needs to predict their future locations so that it can plan what to do ahead of time. For example, if it were to detect a child running towards the road, it should expect the child not to stop. The Kalman filter can help with this problem, as it is used to assist in tracking and estimation of the state of a system.

The self-driving car has sensors that determines the position of objects, as well as a model that predicts their future positions. **Kalman filtering** is used for many applications including **filtering** noisy signals, generating non-observable states, and predicting future states. An example application would be providing accurate, continuously updated information about the position and velocity of an object given only a sequence of observations about its position. The Kalman filter exploits the dynamics of the target, which govern its time evolution, to remove the effects of the noise and get a good estimate of the location of the target at the present time (filtering), at a future time (prediction), or at a time in the past (interpolation or smoothing). **Kalman filters** produce the optimal estimate for a linear system. Also, the Kalman Filter enhances the accuracy of tracking compared to the static least square based estimation.

A Kalman Filtering is carried out in two steps: Prediction and Update.

The Kalman filter process has two steps: the prediction step, where the next state of the system is predicted given the previous measurements, and the update step, where the current state of the system is estimated given the measurement at that time step. The steps translate to equations as follows :

- Prediction

$\ddot{X}_k = A_{k-1} - X_{k-1} + B_k U_k$

$\ddot{P}_k = A_{k-1} P_{k-1} A^T_{k-1} + Q_{k-1}$

- Update

$V_k = Y_k - H_k \ddot{X}_k$

$S_k = H_k \ddot{P}_k H^T{}_k + R_k$

$K_k = \ddot{P}_k H^T_k S^{-1}_k$

$X_k = \ddot{X}_k + K_k V_k$

$P_k = \ddot{P}_k - K_k S_k K^T_k$

Where

- $\ddot{X}_k$ and $\ddot{P}_k$ are the predicted mean and covariance of the state, respectively, on the time step k before seeing the measurement.

- X_k and P_k are the estimated mean and covariance of the state, respectively, on time step k after seeing the measurement.

- Y_k is mean of the measurement on time step k .

- V_k is the innovation or the measurement residual on time step k .

- S_k is the measurement prediction covariance on the time step k .

- K_k is the filter gain, which tells how much the predictions should be corrected on time step k .

X : The mean state estimate of the previous step (k −1).

P : The state covariance of previous step ($k-1$).

A : The transition n x n matrix.

Q : The process noise covariance matrix.

B : The input effect matrix.

U : The control input.

At the time step k , the update step computes the posterior mean X and covariance P of the system state given a new measurement Y . The function therefore performs the update of X and P giving the predicted X and P matrices, the measurement vector Y , the measurement matrix H and the measurement covariance matrix R .

The Kalman Filter for object tracking has been implemented with Python.

RESULTS

Image Classification is finding what is in an image and the test results show good accuracy between training and validation data.

The Kalman filter is a uni-modal, recursive estimator. Only the estimated state from the previous time step and current measurement is required to make a prediction for the current state. The Kalman Gain is calculated along with the observed data and the process covariance is also updated based on the kalman gain. These updates are used for the next round of predictions. The observations are compared to the prediction and the test results show close proximity between measured and estimated trajectory/values.

CONCLUSION

Artificial Super Intelligence(ASI) is based on ideas that machines not only imitate the human mind but can even supersede human’s intelligence. In order to achieve this, the ASI will have to be more conscious by improving cognitive abilities of the artificial machines. In this paper, we have taken Self-Driving Car as “Artificially” created conscious object and implemented Obstacle Detection(deep learning) and Object Prediction(Kalman Filter) where it would have a variety of sensory inputs and be aware of its environment. The test results show that it is possible to build artificial machines that are conscious and intelligent by collating information collected by different sensors.

REFERENCES

1. https://github.com/

2. Gustav Von Zitzewitz : "Survey of neural networks in Autonomous Driving".

   URL- https://www.researchgate.net/publication/

3. Mohamed LAARAIEDH IETR Labs, University of Rennes : "Implementation of Kalman Filter with Python Language" URL - https://arxiv.org/ftp/arxiv/papers/1204/1204.0375.pdf