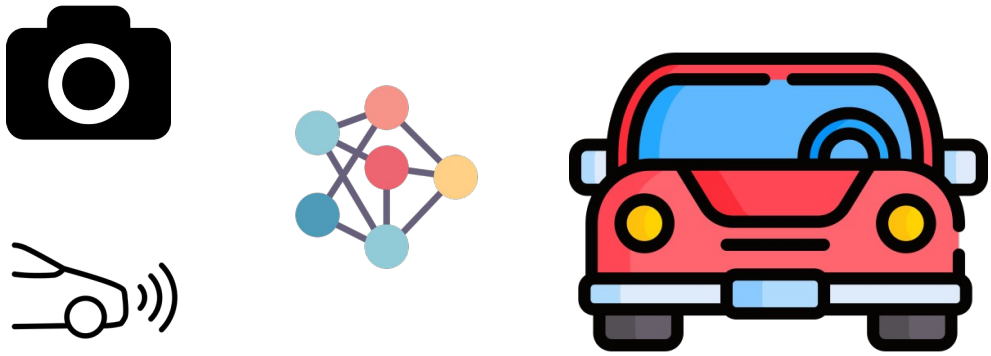
Self-driving car using NN



What is being controlled? (steering angle, throttle, and brake)
What is the input? Camera image, PointCloud Data from Lidar

To make things simple, let's only consider image from the Camera.

Let's now see how our model could be? How we can use the camera to control the car, also known as Self-driving car.

Input Layer: The input layer would take sensor data as input. Each sensor could be represented as a separate channel in the input layer. For example, cameras might have RGB channels, LIDAR could have depth information, and radar might have different frequency bands.

The number of nodes in the input layer would depend on the number of sensor channels and the resolution of each sensor. For example, a camera with a resolution of 1280x720 might have 1280x720x3 nodes for RGB channels.

Convolutional Layers: To process spatial information from sensors like cameras and LIDAR, convolutional layers would be used. These layers help in extracting features such as edges, shapes, and textures. The number of convolutional layers and their sizes would depend on the complexity of the environment and the resolution of the sensor data.

The number of nodes in each convolutional layer is determined by the filter size, the number of filters, and the stride. A common practice is to start with a small number of filters (e.g., 32 or 64) in the first few layers and increase the number in deeper layers.

Pooling Layers: After each convolutional layer, pooling layers can be added to reduce the spatial dimensions of the data, making the network more computationally efficient and helping prevent overfitting. Pooling layers typically don't change the number of nodes, as they reduce the spatial dimensions without altering the depth (number of channels).

Fully Connected Layers: Following the convolutional and pooling layers, fully connected layers can be added to learn high-level features and make decisions based on the extracted features. The number of nodes in these layers can vary based on the complexity of the task. The number of nodes in the fully connected layers can vary based on the complexity of the task. A common practice is to reduce the number of nodes in each successive fully connected layer. For example, the first fully connected layer might have 512 nodes, and the second might have 256 nodes.

Output Layer: The output layer would consist of nodes representing different actions or controls for the car, such as steering angle, throttle, and brake. For this case, the output layer would have 3 nodes.

Activation Functions: ReLU (Rectified Linear Unit) activation functions are commonly used in convolutional and fully connected layers for their simplicity and effectiveness. However, for the output layer, the activation functions would depend on the task. For example, a linear activation function might be used for regression tasks like steering angle prediction, while softmax might be used for classification tasks like traffic sign recognition.