INFLUENCE OF RECURRENT NEURAL NETWORKS (RNN) SUCH AS LSTM AND GRU ON AUTONOMOUS VEHICLES

1. INTRODUCTION

1.1. RECURRENT NEURAL NETWORK

Humans are gifted with the power of pattern recognition and the ability to identify and discard the necessary and unwanted information. Human neurons are one of the complex things to understand and deep learning mimics the working of these neurons. A traditional neural network will not be able to classify any information as useful or waste but with RNN we can train our models to do the same. A typical RNN structure is shown below.

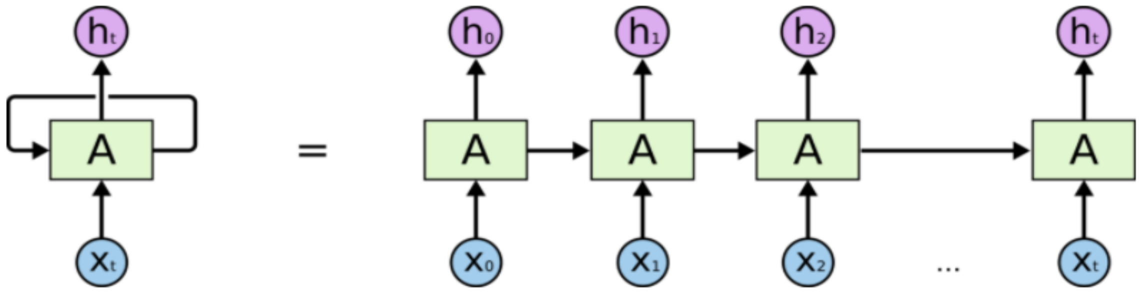

Fig 1(a) : Recurrent neural network

In the above figure, Xt is the input and ht is the output of that particular neuron which acts as a hidden layer (previous data) to the next RNN neuron. Notice that the loop is what makes RNN different and also they help us to pass the information from one step of the network to the next. Such similar networks are cascaded together and form the RNN network

1.2. INTRODUCTION TO LSTM AND GRU

During backpropagation of a model, RNN suffers from what is called the “vanishing gradient problem” i.e the gradient value becomes too small and doesn’t contribute to the learning and the weights won’t get updated. Hence, a type of RNN was introduced called LSTM

Long Short Term Memory networks or LSTM’s are capable of learning long-term dependencies unlike the regular RNN

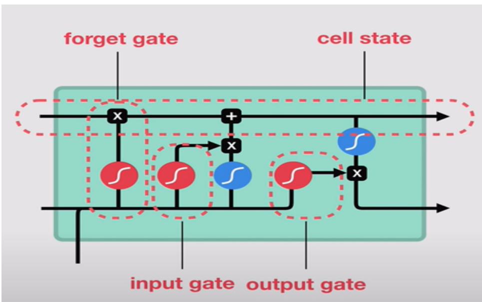

Fig 2(a): A typical LSTM

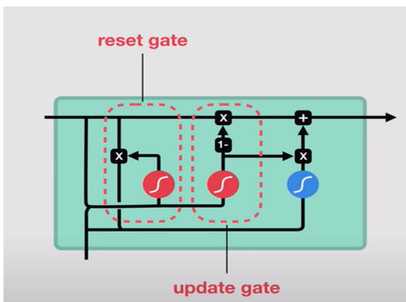

Fig 2(b): A typical GRU

- ❏ LSTM cells first decide which information is not necessary and can be thrown out of the cell state. A layer called the “forget gate layer” is a sigmoid layer that helps the cell in making that decision
- ❏ The “input gate layer” decides which values to be updated, and the tanh creates a vector and update the old cell state

- This output will be based on our cell state but will be a filtered version. First, we run a sigmoid layer which decides what parts of the cell state we're going to output. Then, we put the cell state through tanh and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to get.
- Gated recurrent units(GRU) are similar to LSTM and are the newer generation of RNN. Instead of a cell state, we have a hidden state and an update gate similar to the forget gate is LSTM. The reset gate decides how much past information is to be forgotten.

2. APPLICATION OF RNN ON AUTONOMOUS VEHICLES

RNN is used in many places like speech recognition, language modeling, translation, image captioning most importantly in Autonomous vehicles. When in a self-driving car or similar vehicle it is common to have some sensor failure during that the vehicle must safely maneuver and at the same time avoid collision with other vehicles on the road. This issue needs to be solved by predicting future semantic behaviors of other drivers, like lane changes, and their future trajectories using past sensor observations. This is where RNN will be used i.e it will be used to predict things like the future driving intent, for lane changes, of neighboring vehicles up to three seconds with a small-time gap of about one second. With their capacity to learn from large amounts of temporal data, RNNs have important advantages. Since they don't have to only rely on the pixel-based changes in the image obtained, they also increase prediction robustness for the motion of pedestrians and animals in the environment

RNN has to do two things for each object it detects, it must detect the future position and future velocity for that particular object. There are many approaches to this problem. The one I came across was using Radar and lidar sensors to measure object velocity whose data can be used to train the RNN model. This data is propagated in the camera domain which enables us to label camera images with velocity data. Now, we can use cross-sensor fusion and achieve "ground truth". Our RNN output will have time-to-collision (TTC), future position and future velocity predictions for each dynamic object detected while driving.

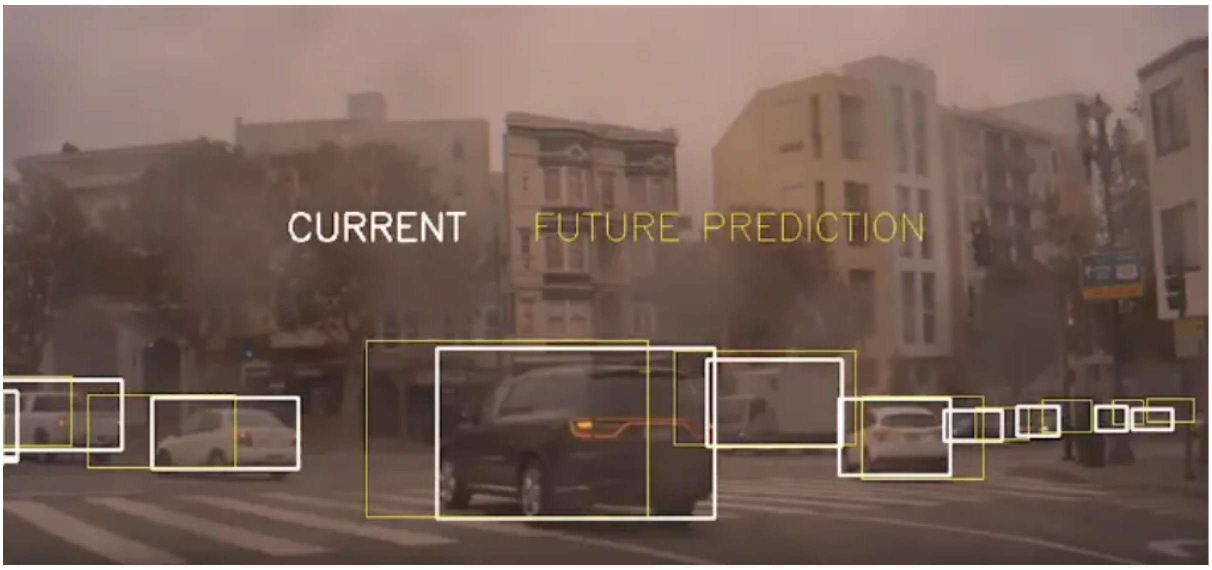

Fig 3(a)

In this image, White boxes indicate current object locations predicted by the RNN, while the yellow boxes show where the object will be in the near future with the help of the RNN

Another article I came across was the use of the famous Waymo Open Dataset. It is the largest, richest, and most diverse AV dataset ever published for academic research. This dataset, collected from Waymo level-5 autonomous vehicles in various traffic conditions, using radar, lidar, and camera data from 1000 20-second segments with labels. The three coordinate systems are provided in this dataset are the global frame, vehicle frame, and sensor frame. The dataset also provides vehicle pose VP, a 4 × 4-row matrix, to transform variables from one coordinate system to another.

Fig 4(a)

Fig 4(b)

As shown in Figure 4(a), the system checks if a hypothetical ray (green) starting from the position of the AV intersects with the front car (the red bounding box). But this procedure has a disadvantage as there might be many misses when a single ray is used. Hence, they came with the rectangular approach i.e as shown in Fig 4 (b). Many ensemble methods were used like XGBoost, Light Gradient Boosting, Stacked Linear Regressor. But the "encoder-decoder model" was used because acceleration is to be predicted and can be done based on the acceleration curve which is a trajectory based on past experiences.

The 12 basic features are packed into one single input channel in their model. These features are fed into an "encoder" module and this "encoder" extracts the key information from input features and generates an intermediate result. The intermediate result is then passed onto the "decoder" module, which decodes the information and gives the acceleration prediction as to the output.

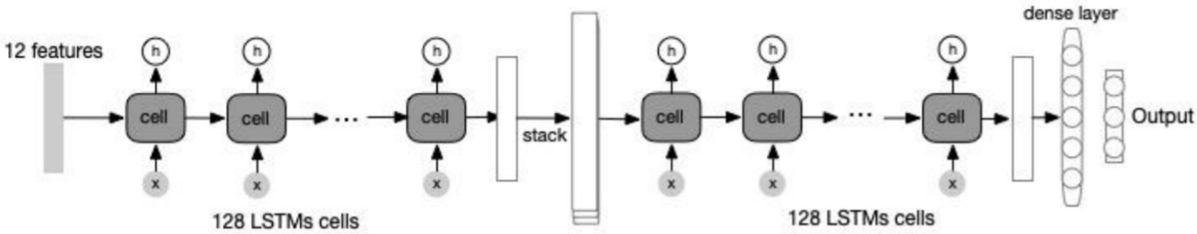

Fig 4(C)

Although this model has many advantages few of its disadvantages are the limitation of information provided by the 12 input features which are less. It is difficult for the model to collect and analyze its surrounding environment for the AV by these simple and limited numerical features, and the model trained on these features may fail to give a satisfying prediction. High-quality features with more key information are needed to improve this model's capacity.

3. REFERENCES

- Zhicheng Gu, Zhihao Li, Xuan Di and Rongye Shi 2 - “An LSTM-Based Autonomous Driving Model Using a Waymo Open Dataset”, 18 March 2020

- NEDA CVIJETIC - “DRIVE Labs: Predicting the Future with RNNs”, May 22, 2019

- DENNY BRITZ - “Recurrent Neural Network Tutorial, Part 4 – Implementing a GRU/LSTM RNN with Python”, OCTOBER 27, 2015

- Colah’s blog - “Understanding LSTM Networks”, August 27, 2015

- Sajan Patel, Brent Griffin, Kristofer Kusano, Jason J. Corso - “Predicting Future Lane Changes of Other Highway Vehicles using RNN-based Deep Models”, 16 May 2019