Sam Park

Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks

Paper:

Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, Alexandre Alahi. Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks: https://arxiv.org/abs/1803.10892

Summary:

Abstract

Understanding human motion behavior is important during navigation in a human-centric environment. However, this is challenging because human motion is multimodal.The authors combine sequence prediction and generative adversarial networks to tackle this challenge.

Introduction

Predicting pedestrians' behaviors is essential but challenging due to inherent features of human motion in a crowded environment.

- Inherent features of human motion that are challenging:
    1. Interpersonal: Each person's motion is affected by the people around them.
    2. Socially Acceptable: Pedestrians are governed by social norms (e.g. social distancing)
    3. Multimodal: There could be multiple plausible ways to reach the destination.

Although existing methods have made progress in tackling these challenges, there are still limitations. Limitations from existing methods are:

1. Local neighborhood: Do not model interactions between all people in a scene.
2. Commonly used loss function: Learn only the average behavior of people.

In contrast to these methods, the authors account for "good behaviors" (e.g. socially acceptable trajectories).

The authors proposed to use GAN to generate many socially acceptable trajectories based on a given dataset from the past observation. This GAN has 2 networks: 1. RNN Encoder-Decoder Generator 2. RNN Based Encoder Discriminator. The generator generates candidates and the discriminator evaluates the candidates. **The authors' GAN model, "Social GAN", has a global pooling vector that encodes the subtle cues for all people in the scene**.

Related Work

The authors' work is focused on human-human interactions. There have been methods that model human-human interactions and they were handcrafted. Over the past several years, data-driven methods based on RNNs have been used.

RNNs

- Lacks high level and spatio temporal structure. A Previous model based on RNNs did not model human-human interactions in crowded scenes.

Generative Modeling

- Good for tasks that have multiple possible outputs for a given input.

Methods

The authors want to predict multiple future trajectories. They propose to use GAN based encoder-decoder, novel pooling layer for modeling human-human interactions, and variety loss that produces multiple diverse future trajectories for the same observed sequence.

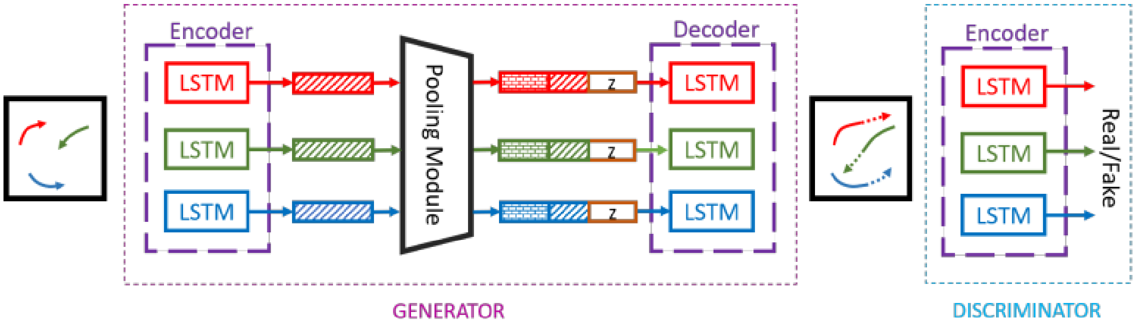

Figure 2: System overview. Our model consists of three key components: Generator (G), Pooling Module, and Discriminator (D). G takes as input past trajectories X_i and encodes the history of the person i as H_i^t. The pooling module takes as input all $H_i^{t_{obs}}$ and outputs a pooled vector P_i for each person. The decoder generates the future trajectory conditioned on $H_i^{t_{obs}}$ and P_i. D takes as input T_{real} or T_{fake} and classifies them as socially acceptable or not (see Figure 3 for PM).

The authors aim to jointly reason and predict the future trajectories of all the people in the scene simultaneously.

In a GAN, the 2 neural networks (the generator and the discriminator) are trained in opposition (adversarial) to each other. The authors embedded the location of each person using a single layer MLP (multi-layer perceptron) to get a fixed length vector. The discriminator will ideally learn subtle social interaction rules and classify trajectories which are not socially acceptable as “fake”.

Experiments

The authors computed the prediction error using 2 different error metrics:

1. Average Displacement Error (ADE) to compute mean euclidean distance over all estimated points at each time step in the predicted and true trajectories
2. Final Displacement Error (FDE) to compute the mean euclidean distance between the final predicted location and the final true location

The authors considered 3 different scenarios where people have to adjust their course to avoid collision.

1. People Merging
2. Group Avoiding (People avoiding each other while moving in opposite direction)
3. Person Following (a person walking behind someone)

There is a subtle difference between the authors' model and the real life situation where people in real life make trajectory decisions based on their field of view while the authors' model has access to ground truth positions of all people in the scene.

Conclusion

The authors tackle the problem of modeling human-human interaction and jointly predicting trajectories for all people in a scene. They propose a novel GAN based encoder-decoder framework for trajectory prediction capturing the multiple various trajectories.

The authors also propose a pooling mechanism that enables the network to learn social norms through a purely data-driven approach.

Glossary:

GAN (Generative Adversarial Networks)

- A generative adversarial network is a class of machine learning frameworks designed by Ian Goodfellow and his colleagues in June 2014 that, given a training set, learns to generate new data with the same statistics as the training set.
- GAN contains 2 networks: Generator & Discriminator
    - The generator and discriminator compete with each other. The generator creates random samples from Gaussian distribution and the discriminator (adversary to the generator) determines among training sets and random samples generated from the generative whether they are real or fake.

GP (Gaussian Process)

- In probability theory and statistics, a Gaussian process is a stochastic process, such that every finite collection of those random variables has a multivariate normal distribution.

L2 Distance (Euclidean Distance)

- The length of a line segment between the two points

ReLU (rectified linear activation unit)

- The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time

Loss function

- The function that computes the distance between the current output of the algorithm and the expected output

Hidden state

- E.g. A hidden state h_(t-1) stores the sequence information up to time step t-1.
- In general, the hidden state at any time step t could be computed based on both the current input x_t and the previous hidden state h_(t-1).

Markov Process

- A Markov process is a random process in which the future is independent of the past, given the present.

Loss (error)

- Error (loss) = t - y
- Error (loss) = t - W*x + b
    - t: real value
    - y: predicted value
        - W: weight (slope of the regression line)
        - b: bias (y-intercept of the regression line)
        - x: input

Loss function

- A function that accounts for all the losses (errors)

$$= \frac{1}{n} \sum_{i=1}^{n} [t_i - (Wx_i + b)]^2$$