Prediction Under Uncertainty for Autonomous Driving

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider the problem of prediction in a highly uncertain environment.

1 Introduction

The contributions of this work are the following:

- We introduce a new open-source environment to test methods for autonomous driving, based on a large dataset of real-world driving data.
- We introduce a novel method for prediction under uncertainty which is simple to train, does not make assumptions about a prior distribution over latent variables or require sampling at training time, and is able to generate diverse predictions for hundreds of timesteps into the future.

2 Prediction Model

Our stochastic prediction model can be viewed as a conditional autoencoder paired with a non-parametric sampling procedure. The architecture consists of three neural networks: an encoder f_1, a decoder f_2 and a latent variable network ϕ. For each sample, the update equations are given by:

$$
\begin{aligned}
z_i &= \phi(x_i, y_i) \\
\tilde{y}_i &= f_2(f_1(x_i), z_i)
\end{aligned}
$$

and all networks are trained by gradient descent to optimize the following objective:

$$
\begin{aligned}
\mathcal{L} &= \sum_i \|y_i - \tilde{y}_i\|_2^2 && (1) \\
&= \sum_i \|y_i - f(x_i, \phi(x_i, y_i))\|_2^2 && (2)
\end{aligned}
$$

Note in particular that no sampling or reparamaterization is done at training time. After training, we extract all vectors z_i from the training set and use these as inputs to new inputs.

3 Related Work

In recent years, several works have explored prediction of complex time series such as video [1]. These typically train models to predict future frames with the goal of learning good representations

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

Algorithm 1 My algorithm

Input: Time series $\{s_1 ... s_T\}$.
Train latent model:
while not converged **do**
  $z_t = f_\phi(s_{1:t}, s_{t+1})$
  $\tilde{s}_{t+1} = f_\theta(s_{1:t}, z_t)$
  $\ell(\theta, \phi) = \|s_{t+1} - \tilde{s}_{t+1}\|_2^2$
  $\theta \leftarrow \theta - \eta \nabla \theta$
  $\phi \leftarrow \phi - \eta \nabla \phi$
procedure ESTIMATELATENTMANIFOLD
  $V \leftarrow \{\}$
  $E \leftarrow \{\}$
  for $t = 1 : T$ **do**
    $z_t = f_\phi(s_{1:t}, s_{t+1})$
    $V[t] = z_t$
  for $t = 1 : T$ **do**
    $E[t] \leftarrow$ list of k nearest neighbors of $V[t] = z_t$ in V
  return $G = (V, E)$
procedure GENERATE(s_0, s_1)
  Initialize $z_1 = f_\phi(s_0, s_1)$
  for $t = 1 : T$ **do**
    $\tilde{s}_{t+1} = f_\theta(s_{1:t}, z_t)$

which disentangle factor of variation and can be used for unsupervised learning [2**?** , 3] or learn action-conditional forward models which can be used for planning [4, 5, 6**?**]. Several works have included latent variables as a means to model the uncertainty, using the framework of Variational Autoencoders [7, 8]. In contrast to VAEs, our model does not place any priors on the latent variable distribution, which removes the need for an additional loss term enforcing consistency between the prior and posterior. Additionally, our approach does not require any sampling at training time which reduces the variance in the gradients.

Our method is closely related to Gated and Relational Autoencoders [9, 10], which were used to learn transformations between pairs of images in an unsupervised manner. The general architecture and loss is similar in both our works, however their focus was on representation learning for static images while ours is on video generation for use in planning.

- Video Prediction [1], stochastic: using the framework of Variational Autoencoders [11].

- Mixture Density Networks [12].

- VQ-VAE, GLO, Gated Autoencoders

4 Dataset

5 Experiments

- Histogram of distances between consecutive z vs random pairs.

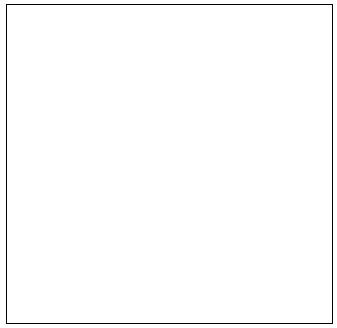

Figure 1: Sample figure caption.

Table 1: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

5.1 Figures

5.2 Tables

6 Final instructions

7 Appendix

Our model can also be viewed through the lens of Variational Autoencoders, as a type of conditional VAE with a zero-variance posterior network and a uniform categorical prior. In this case, the KL term reduces to a constant which can be ignored during training; details can be found in the Appendix.

Fixed Prior

$$p_\psi(z_j|x_i, y_i) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases} \tag{3}$$

$$p_\phi(z_j|x_i) = \frac{1}{M} \tag{4}$$

$$\tag{5}$$

The KL divergence is therefore:

$$\begin{aligned} \mathrm{KL}(p_\phi(z|x_i)||p_\psi(z|x_i, y_i)) &= \sum_j p_\psi(z_j|x_i, y_i) \log \frac{p_\psi(z_j|x_i, y_i)}{p_\phi(z_j|x_i)} && (6) \\ &= 1 \cdot \log \frac{1}{\frac{1}{M}} && (7) \\ &= \log M && (8) \end{aligned}$$

Learned Prior

$$p_\psi(z_j|x_i, y_i) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{else} \end{cases} \tag{9}$$

$$p_\phi(z_j|x_i) = \frac{h(z_j; x_j, \theta)}{\sum_{k=1}^{M} h(z_k; x_j, \theta)} \tag{10}$$

The KL divergence is therefore:

$$\begin{aligned}
\mathrm{KL}(p_\phi(z|x_i)||p_\psi(z|x_i, y_i)) &= \sum_j p_\psi(z_j|x_i, y_i) \log \frac{p_\psi(z_j|x_i, y_i)}{p_\phi(z_j|x_i)} && (11) \\
&= 1 \cdot \log \frac{1}{\frac{h(z_j; x_j, \theta)}{\sum_{k=1}^{M} h(z_k; x_j, \theta)}} && (12) \\
&= -\log \frac{h(z_j; x_j, \theta)}{\sum_{k=1}^{M} h(z_k; x_j, \theta)} && (13) \\
&= -\log h(z_j; x_j, \theta) + \log \sum_{k=1}^{M} h(z_k; x_j, \theta) && (14)
\end{aligned}$$

The last term is a constant and can be ignored.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

[1] Michaël Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *CoRR*, abs/1511.05440, 2015.

[2] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using lstms. *CoRR*, abs/1502.04681, 2015.

[3] Emily Denton and Vighnesh Birodkar. Unsupervised learning of disentangled representations from video. *CoRR*, abs/1705.10915, 2017.

[4] Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L. Lewis, and Satinder P. Singh. Action-conditional video prediction using deep networks in atari games. *CoRR*, abs/1507.08750, 2015.

[5] Chelsea Finn, Ian J. Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *CoRR*, abs/1605.07157, 2016.

[6] Pulkit Agrawal, Ashvin Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. *CoRR*, abs/1606.07419, 2016.

[7] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H. Campbell, and Sergey Levine. Stochastic variational video prediction. *CoRR*, abs/1710.11252, 2017.

[8] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. *CoRR*, abs/1802.07687, 2018.

[9] Roland Memisevic and Geoffrey Hinton. Unsupervised learning of image transformations, 07 2007.

[10] R. Memisevic. Learning to relate images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1829–1846, Aug 2013.

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. 12 2013.

[12] Christopher Bishop. Mixture density networks. Technical report, January 1994.

References

References follow the acknowledgments. Use unnumbered first-level heading for the references. Any choice of citation style is acceptable as long as you are consistent. It is permissible to reduce the font size to `small` (9 point) when listing the references. **Remember that you can use more than eight pages as long as the additional pages contain *only* cited references.**