Student Internship Programme (SIP)

Final Project Report

at

GrabTaxi Holdings Pte. Ltd.

Reporting Period:

May 2020 to Jul 2020

by

New Jun Jie

Department of Computer Science

School of Computing

National University of Singapore

2020/2021

Project Title: Simkit - Generative and Probabilistic Modelling Framework for Reinforcement Learning

Project ID: SY192539200

Project Supervisor: Prof Stefan Winkler

Abstract

Simkit (Environment Model Kit) is a generalized framework for generative and probabilistic modelling for training reinforcement learning agents in TensorFlow. Many pricing and decision making problems at the core of Grab's ride-hailing and deliveries business can be formulated as reinforcement learning problems, with interactions of millions of passengers, drivers and merchants from over 65 cities across the Southeast Asia region. In my internship as a Machine Learning Engineer in the Marketplace team, I worked with scientists and engineers to develop Simkit in order to train reinforcement learning agents to be served in production services and make optimal, real-time decisions. This report presents a summary of the work and contributions I have made throughout the internship.

Subject Descriptors

Reinforcement Learning

Machine Learning

Neural Networks

Probabilistic Representations

Keywords

Reinforcement Learning, Machine Learning, Probabilistic Modelling, Bayesian Statistics

Software

TensorFlow 2, TensorFlow Probability, Ax, Pandas, Numpy, Matplotlib, Scipy, Python

Acknowledgement

I would like to express my deepest appreciation for my mentors from the Marketplace team at Grab. Their guidance, patience and continued support throughout my internship enabled me to progress and contribute at a faster pace than I had imagined.

I would like to thank Senior Data Science Manager (Econs Platform) Dmitry Bezyazychnyy for providing me with the opportunity and platform, enabling me to work on a challenging, engaging and fun project, and Lead Data Scientist Chongyu Zhou for guiding me towards theory and practical considerations in implementing generative and probabilistic models.

I would like to thank both Grab and the school for allowing for such a meaningful internship opportunity.

Table of Contents

Introduction

Background of Grab

Grab is Southeast Asia's leading Online-to-Offline (O2O) mobile platform. Grab is the region's everyday super app that provides services that matter most to customers. Through its open platform strategy, Grab works with partners to provide safe, accessible and affordable transport, food, package, grocery delivery, mobile payment and financial services to millions of Southeast Asians.[1] Grab aims to create 100 million micro-entrepreneurs through its ecosystems.

Position of Marketplace within Grab

The Marketplace team is at the core and centre of Grab's ride-hailing and deliveries business. The team's algorithms and models shape supply and demand in a market of passengers, drivers, and merchants through intelligent pricing and a variety of incentives. The team applies machine learning, forecasting and optimization techniques to huge datasets in order to gain a strong understanding of passenger and driver behaviours and continuously run large-scale live experiments to improve on our implementations.

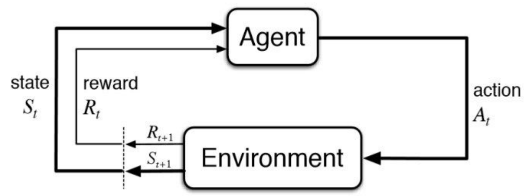
Background of Project

In the Marketplace team, many pricing optimization and decision making problems require knowledge of user response, which is difficult to predict using offline historical data. Online learning and reinforcement learning are useful techniques where algorithms continuously learn from experience to improve the policy gradually towards optimal. Simkit is a new generalized framework for building generative and probabilistic models that accompanies existing reinforcement learning frameworks to construct artificial environments for training reinforcement learning agents.

Internship Project - Simkit

Project Objective

Simkit (Environment Model Kit) is a generalized framework for generative and probabilistic modelling for training reinforcement learning agents in TensorFlow 2. Code is provided at *github.com/jetnew/Simkit*.



The reinforcement learning framework consists of an agent learning from its interactions with the environment.[2] With every action taken by the agent, the environment returns a state and a reward. Reinforcement learning is known to be unstable in training due to sample inefficiency and noisy observations and reward functions, therefore Simkit proposes 2 types of models: Conditional generative feature models that model the distribution of states, and probabilistic response models that model the distribution of rewards.

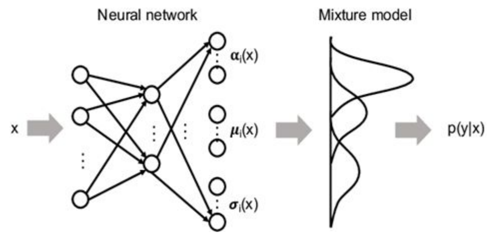
Researched & Implemented Models

Conditional Generative Feature Models

Feature models are used in reinforcement learning for generating features that represent the state during agent-environment interactions. For example, food-surge, a pricing algorithm in food delivery, considers independent parent features, time and merchant identifier, and dependent child features driver distance, basket value and merchant-to-eater distance. Computation of the joint distribution, $p(x_1, x_2, ..., x_N) = p(daxDist, basketValue, mexExDist \mid time, mexId) * p(time) * p(mexId)$, becomes modelling the conditional joint distribution, $p(x_{child}|x_{parent})p(x_{parent}) = p(daxDist, basketValue, mexExDist|time, mexId)$.

Gaussian Mixture Density Network

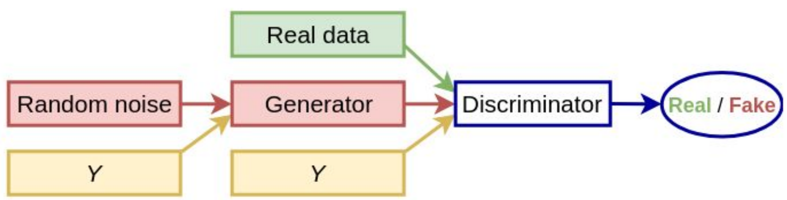
The Gaussian Mixture Density Network[3] consists of a neural network to predict the parameters that define the Gaussian mixture model, $p(x) = \sum_{i=1}^{K} \pi_i \, N(x|\mu_i, \Sigma_i)$, where $N(x|\mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^K|\Sigma_i|}} exp(-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1}(x-\mu_i))$.



Each parameter $\pi_k(x),\ \mu_k(x),\ \sigma_k(x)$ is approximated by a neural network as a function of input x, which represents the parent feature. π_k is transformed by softmax so that $\sum_{i=1}^{K} \pi_i = 1$. σ_k is transformed by exponential so that it is positive, where $Z = W_0 \tanh(W_h X + b_h) + b_0$ and $\Pi_k = \frac{\exp(Z_k)}{\sum \exp(Z)}$, $\sigma = \exp(Z),\ \mu = Z$.

Conditional Generative Adversarial Network

The Conditional Generative Adversarial Network consists of a generator network that generates candidate features and a discriminator network that evaluates them, both conditioned on parent features, that contest in optimisation via the loss function $\min_G \max_D (E_{y,x \sim p_{data}(y,x)}[\log D(y,x)] + E_{x \sim p_x, z \sim p_{z(x)}}[log(1 - D(G(z,x),x))])$.[4]



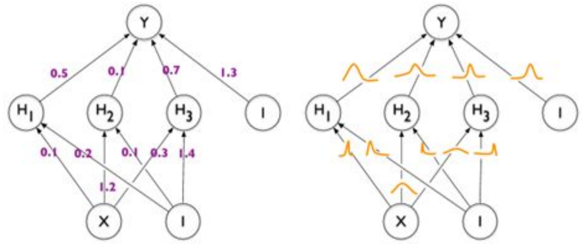
GANs are known to be unstable in training and therefore the improvements Wasserstein loss, label smoothing and diversity penalties are applied.

Probabilistic Response Models

Response models are used in reinforcement learning for the uncertainty modelling of distributional rewards instead of point estimations, to enable stable learning of the agent in cases of spiky responses.

Bayesian Neural Network

The Bayesian Neural Network is a neural network with weights assigned a probability distribution to estimate uncertainty and trained using variational inference.[5]

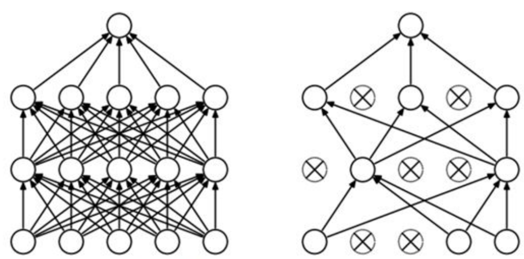


Variational Bayesian neural networks uses a prior distribution P(W) over weights and approximates the posterior distribution P(W|D) a P(W)P(D|W), where D denotes observed data. The evidence lower bound (ELBO) is maximised in variational inference:

$$F(D,\theta) = KL(q(w|\theta)\,||\,p(w)) - E_{q(w|\theta)} log\, p(D|w) \approx \frac{1}{N}\sum_{i=1}^{N}[log\, q(w^{(i)}|\theta) - log\, p(w^{(i)}) - log\, p(D|w^{(i)})]$$

Monte Carlo Dropout

The Monte Carlo Dropout is a method shown to approximate Bayesian inference.[6]

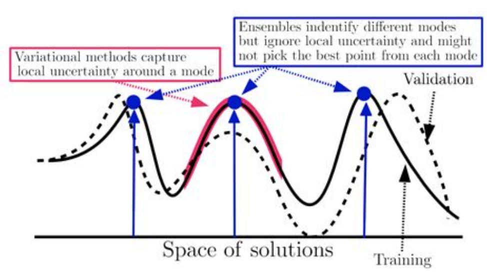


(a) Standard Neural Net (b) After applying dropout.

Dropout training in neural networks allows modelling of uncertainty without sacrificing computational complexity or test accuracy. During inference, Monte Carlo dropout performs stochastic forward passes through the neural network.

Deep Ensemble

The Deep Ensemble is an ensemble of randomly-initialised neural networks that performs better than Bayesian neural networks in practice.[7]



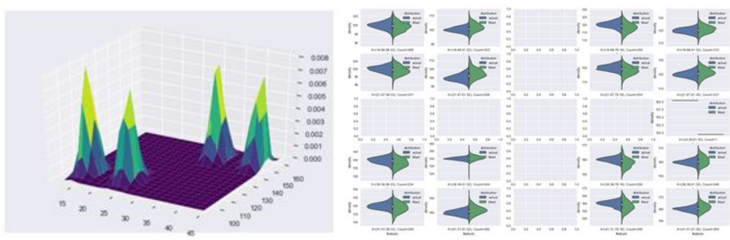
Variational Bayesian methods tend to focus on a single mode while deep ensembles tend to explore diverse modes in the function space due to random initializations, mapping the loss landscape.[8]

Implemented Utilities and Tools

Empirical Performance Metrics

The performance metrics computed are the Kullback-Leibler divergence and Jensen-Shannon divergence, $D_{KL}(P\|Q) = \sum P(x)\, log(\frac{P(x)}{Q(x)})$, $D_{JS}(P\|Q) = \frac{1}{2}D_{KL}(P\|M) + \frac{1}{2}D_{KL}(Q\|M)$, where $M = \frac{1}{2}(P+Q)$, computed by splitting the data into histogram bins. One limitation of empirical metrics is that values may be invalid when there are a lot of non-overlapping histogram bins.

Performance Visualisation Tools



The visualisation tools implemented include the probability density surface plot (left) that visualises the probability densities at each coordinate, and the grid violin relative density plot (right) that visualises the relative densities between the actual data and the generated data of the fitted model using histograms.

Hyperparameter Optimisation using Ax

Hyperparameter optimisation is implemented using Bayesian optimisation in the Ax framework, building a smooth surrogate model of outcomes using Gaussian processes from noisy observations from previous rounds of parameterizations to predict performance at unobserved parameterizations, tuning parameters in fewer iterations than grid search or global optimisation techniques.[9]

Model Evaluation

The models in Simkit are evaluated on 2 industry datasets: The Supermarket Sales dataset[10] that contains 1000 samples with 2 parent features (branch, time) and 1 child feature (spending), and the Wholesale Customers[11] dataset contains 440 samples with 2 parent features (channel, region) and 6 child features (spending in fresh, milk, grocery, frozen, detergents & paper, delicatessen). On both datasets, the Gaussian mixture density network performed the best (Supermarket: $D_{KL;} = 0.03530$, $DL_{JS} = 0.003320$; Wholesale: $D_{KL} = 0.0006931$, $D_{JS} = 0.0001699$) out of all the models.

Conclusion

Summary of Contributions

Over the course of my 3-month internship, I worked on the Simkit framework, implementing and documenting generative and probabilistic models for training reinforcement learning agents, created useful utility tools for performance metrics, visualisation and hyperparameter optimisation, and evaluated the models on industry datasets.

Acquired Knowledge, Skills and Experience

My internship with Grab is my first experience with conditional generative models, probabilistic models and Bayesian statistical concepts. While I initially had an introductory understanding of reinforcement learning before the internship, I now understand how the models interplay with the reinforcement learning framework to enable stable training. In the past 3 months, I had progressed from having no theoretical knowledge of the models to an eye-opening, uncertainty-modelling Bayesian perspective of machine learning. From implementing Simkit, I have acquired skills confident of combining probabilistic models and deep learning from research papers using TensorFlow Probability[12], a library for probabilistic reasoning and statistical analysis, and a clear understanding of Bayesian hyperparameter optimisation using the Ax framework created by Facebook, which is commonly used for training reinforcement learning agents. My experience with the Marketplace team at Grab taught me to work effectively with large teams comprising data scientists, machine learning and software engineers. I learnt the nuances of productionising machine learning solutions, and the different roles and responsibilities that are involved.

Concluding Remarks

As a Machine Learning Engineer, all the knowledge, skills and experience I have acquired in my internship are directly relevant and useful in my current and future studies and research. My experience replicating research papers that combine probabilistic models with deep learning will serve advantageous in my study of my research interest in reinforcement learning. Through my internship with Grab, I have gained valuable knowledge, skills and experience and have achieved much more than I thought possible in this short timeframe. My internship at Grab has been a very fulfilling experience and I look forward to future collaborations with Grab.

References

1. Mei, L. I. N., and Christopher William DULA. "Grab taxi: Navigating new frontiers." (2016): 40.
2. Sutton, Richard S., and Andrew G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
3. Bishop, Christopher M. "Mixture density networks." (1994).
4. Mirza, Mehdi, and Simon Osindero. "Conditional generative adversarial nets." *arXiv preprint arXiv:1411.1784* (2014).
5. Blundell, Charles, et al. "Weight uncertainty in neural networks." *arXiv preprint arXiv:1505.05424* (2015).
6. Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." *international conference on machine learning*. 2016.
7. Lakshminarayanan, Balaji, Alexander Pritzel, and Charles Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles." *Advances in neural information processing systems*. 2017.
8. Fort, Stanislav, Huiyi Hu, and Balaji Lakshminarayanan. "Deep ensembles: A loss landscape perspective." *arXiv preprint arXiv:1912.02757* (2019).
9. Chang, Daniel T. "Bayesian Hyperparameter Optimization with BoTorch, GPyTorch and Ax." *arXiv preprint arXiv:1912.05686* (2019).
10. Dataset at *https://www.kaggle.com/aungpyaeap/supermarket-sales*.
11. Dataset at *https://www.kaggle.com/binovi/wholesale-customers-data-set*.
12. Dillon, Joshua V., et al. "Tensorflow distributions." *arXiv preprint arXiv:1711.10604* (2017).