
Uncertainty Estimates for Efficient Neural Network-based Dialogue Policy Optimisation

Christopher Tegho*
Department of Engineering
Cambridge University
christegho@gmail.com

Paweł Budzianowski*
Department of Engineering
Cambridge University
pfb30@cam.ac.uk

Milica Gašić
Department of Engineering
Cambridge University
mg436@cam.ac.uk

Abstract

In statistical dialogue management, the dialogue manager learns a policy that maps a belief state to an action for the system to perform. Efficient exploration is key to successful policy optimisation. Current deep reinforcement learning methods are very promising but rely on ϵ -greedy exploration, thus subjecting the user to a random choice of action during learning. Alternative approaches such as Gaussian Process SARSA (GPSARSA) estimate uncertainties and are sample efficient, leading to better user experience, but on the expense of a greater computational complexity. This paper examines approaches to extract uncertainty estimates from deep Q-networks (DQN) in the context of dialogue management. We perform an extensive benchmark of deep Bayesian methods to extract uncertainty estimates, namely Bayes-By-Backprop, dropout, its concrete variation, bootstrapped ensemble and α -divergences, combining it with DQN algorithm.

1 Introduction

Statistical approaches to dialogue modelling allow automatic optimisation of the Spoken Dialogue Systems (SDS) [25]. A SDS is typically designed according to a structured ontology (or a database schema), which defines the domain that the system can talk about. The domain is presented using slots, which are variables the user can either specify or ask about in the domain.

The system also comprises various statistical components. This includes a spoken language understanding module, which takes a sentence as input and gives a dialogue act as output [9], (e.g. requesting information, informing constraints, saying good-bye), and a slot-value pair that specify arguments of the act. For example `inform(hotel=expensive)` is a dialogue act of type `inform`, where the user is informing the system that they would like to constrain their search to luxury venues. The other components of a SDS include a dialogue belief state tracker that predicts user intent and track the dialogue history, a dialogue policy to determine the dialogue flow, and a natural language generator to convert conceptual representations into system responses.

The POMDP framework mitigates the problem of noisy estimates of spoken language understanding by maintaining a distribution over all possible hypotheses, which is called the belief state [24]. A dialogue policy is then employed to map the belief state into an appropriate system action at every dialogue turn. The ability to generalise across different noise levels is essential for successful dialogue policy operation.

Although supervised learning from dialogue corpora can be used to learn human decision-making from data, the action selection does not take any future outcomes of the dialogue into consideration, which leads to a sub-optimal behaviour. As an alternative, Reinforcement Learning (RL), which maximizes the expected sum of rewards received over the course of a dialogue, is a much suitable

* Both authors contributed equally.

method for learning dialogue policies [14]. The reward in this case measures the degree to which the dialogue is successful.

Due to the need of learning with real users through online interactions, an efficient exploration of the state-action space is critical. The Q-function of each state-action pair can be augmented with an estimate of its uncertainty to guide exploration to achieve higher performance and efficient learning [4]. Uncertainty estimates in the policy allow the system to generalise across different noise levels and mitigate errors incurred by speech recognition, therefore resulting in a more robust dialogue manager.

Gaussian Processes (GPs) provide an explicit estimate of uncertainty, but are computationally intensive and preclude the use for large action spaces [8]. Deep neural network (DNNs), on the other hand, scale much better with data and are computationally less expensive than GPs. Many studies have shown that they are suitable for dialogue management tasks [3, 5, 23, 21]. However, their application in noisy environments is relatively under-explored [16, 2]. In this paper, we perform a benchmark of uncertainty estimates for a dialogue domain using Bayesian deep learning [1, 18, 15, 7]. We experiment with and without noise added to the simulated user input to examine generalization capabilities of different approaches and compare it to the state-of-the-art GPSARSA algorithm. The Bayes-by-Backprop algorithm achieves the best performance among the neural networks approaches.

2 Uncertainty estimates in deep reinforcement learning

2.1 Deep Q-learning

In the case of value-based reinforcement learning, we approximate the expected discounted sum of the future rewards received over the course of a dialogue given an action a in a belief state b :

$$Q_t(b, a) = \mathbb{E}_\pi \{r_t + \gamma r_{t+1} + \dots \mid b_t = b, a_t = a\},$$

where r_t is the one-step reward received at a given time t .

The Deep Q-Network (DQN) algorithm models action-value function $Q(b, a)$ using a deep neural network $\hat{Q}(b, a; \theta)$, with weight vector $\theta \in \mathbb{R}^n$, where we iteratively improve the prediction by minimizing the following loss:

$$L(\theta_t) = \mathbb{E} \left[(y_t - \hat{Q}(b_t, a_t; \theta_t))^2 \right],$$

where the targets y_t are:

$$y_t = r_t + \gamma \max_{a'} \hat{Q}(s_{t+1}, a'; \theta_t).$$

At testing time the system responds with an action that yields the highest Q-value for a given belief state b .

2.2 Uncertainty estimates in neural networks

To obtain uncertainty estimates from a neural network, Bayesian neural networks (BNNs) can be employed [17]. Instead of having single fixed value weights w in the neural networks, all weights are represented by probability distributions over possible values given observed dialogues \mathcal{D} , $P(w|\mathcal{D})$. Uncertainty in the hidden units allows the expression of uncertainty about predictions [1].

For exploration, Thompson sampling is used instead of ϵ -greedy, which consists of performing a single stochastic forward pass through the network every time an action needs to be taken. The Q-values given the input belief state b are given by:

$$P(Q, b) = \mathbb{E}_{P(w|\mathcal{D})} [P(Q|b, w)]. \quad (1)$$

Taking an expectation under the posterior distribution is equivalent to using an ensemble of an uncountably infinite number of neural networks, which is intractable [1]. We have to resort to sampling-based or stochastic variational inferences.

2.3 Benchmarked algorithms

We used in this benchmark five algorithms to extract uncertainty estimates from deep Q-Networks. Four of them can be casted within the variational inference framework:

2.3.1 Variational inference

The intractable posterior $P(w|\mathcal{D})$ is approximated with a variational distribution $q(w|\theta)$. The parameters are learnt by minimizing the Kullback-Leibler (\mathcal{KL}) divergence between the variational approximation $q(w|\theta)$ and the true posterior over the weights $P(w|\mathcal{D})$. The resulting cost function is termed as the variational free energy [11]:

$$\mathcal{F} = \mathcal{KL}[q(w|\theta)||P(w)] - \mathbb{E}_{q(w|\theta)} [\ln P(\mathcal{D}|w)]. \quad (2)$$

We test here four algorithms that can be casted in the variational approximate framework, namely Bayes-by-Backprop [1, 16], dropout [6], concrete dropout [7] and α -divergence [15, 10].

Deep BBQ-Learning. We implement the Bayes By Backprop method with DQN. To propagate the error through a layer that samples from $q(w|\theta)$, the reparameterization trick is used [13]. We choose $q(w|\theta)$ to be a Gaussian with diagonal covariance with a variational parameter set θ . Given the mean μ_i and covariance σ_i of q for each weight, a sample from q is obtained by first sampling $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon)$, then computing $w_i = \mu_i + \sigma_i \circ \epsilon_i$, where \circ is point-wise multiplication. To ensure all σ_i are strictly positive, the softplus function $\sigma_i = \log(1 + \exp(\rho_i))$ is used where ρ is a free parameter [16]. The variational parameters are then $\theta = \{\mu_i, \rho_i\}_{i=1}^D$ for D -dimensional weight vector w . The resulting gradient estimator of the variational objective is unbiased and has a lower variance. The exact cost in Eq. 2 can then be approximated as:

$$\mathcal{F}(\mathcal{D}, \theta) \approx \sum_{i=1}^n \log q(w^{(i)}|\theta) - \log P(w^{(i)}) - \log p(\mathcal{D}|w^{(i)}) \quad (3)$$

where $w^{(i)}$ is the i th Monte Carlo sample drawn from the variational posterior $q(w^{(i)}|\theta)$. For the likelihood term in the objective function Eq. 2, we use the expected square loss.

α -Divergences. The approximate inference technique described in the Bayes By Backprop method corresponds to Variational Bayes (VB), which is a particular case of α -divergence, where $\alpha \rightarrow 0$ [10]. The α -divergence measures the similarity between two distributions and can take the form:

$$D_\alpha[p||q] = \frac{1}{\alpha(\alpha - 1)} (1 - \int p(\theta)^\alpha q(\theta)^{1-\alpha} d\theta), \quad (4)$$

where $\alpha \geq 0$.

Hernandez-Lobato et al. [10] found that using $\alpha \neq 0$ performs better than the VB case, where an approximation with $\alpha \geq 1$ will cover all the modes of the true distribution, and the VB case only fits to a local mode, assuming the true posterior is multi-modal [10]. $\alpha = 0.5$ achieves a balance between the two and has shown to perform best when applied to regression or classification tasks.

We experiment with an objective function based on the black box α -divergence (BB- α) energy. We use the reparametrization proposed by [15] :

$$\mathcal{L}_\alpha \approx \tilde{\mathcal{L}}_\alpha = \mathcal{KL}[q(w|\theta)||P(w)] - \frac{1}{\alpha} \sum_n \log \mathbb{E}_{q(w|\theta)} [P(\mathcal{D}|w)], \quad (5)$$

where \mathcal{L}_α designates the BB- α energy, $\tilde{\mathcal{L}}_\alpha$ designates an approximation, and n corresponds to the number of datapoints in the minibatch.

Dropout. Another method to obtain uncertainty estimates in deep neural networks is Bayesian inference with dropout [6]. Dropout consists of randomly dropping units (with some probability d) from the neural network during training [20].

As in the previous methods, dropout can be analyzed from the variational inference perspective (Equation 2). This comes from the fact that applying a stochastic mask is equivalent to multiplying the weight matrix in a given layer by some random noise. The resulting stochastic weight matrix can be seen as draws from the approximate posterior over weights, replacing the deterministic weight matrix [6].

Concrete Dropout. To obtain well-calibrated uncertainty estimates with above method, a grid-search over the dropout probabilities is necessary. However, we can treat a dropout as a part of optimization task obtaining an automatic method of tuning the mask. One method is to continuously relax the

dropout’s discrete masks and optimize the dropout probability using gradient methods [7]. Dropout d probability becomes one of the optimized parameters. The concrete distribution relaxation z of the Bernoulli random variable becomes:

$$z = \text{sigmoid}\left(\frac{1}{t}(\log d - \log(1 - d) + \log u - \log(1 - u))\right)$$

with some temperature t which results in values in the interval $[0, 1]$ and $u \sim \mathcal{U}(0, 1)$.

2.3.2 Bootstrapped DQN

Uncertainty estimates can be obtained with random initialization of several neural networks which predict in ensemble uncertainty estimates for neural networks [18]. To improve efficiency, all networks share the same architecture with a different last layer (head) computing Q-values. The algorithm obtained the highest scores in its non-bootstrapped case when all networks share the same memory replay. Here we employ this ensemble variant.

2.3.3 Computation complexity

To obtain uncertainty estimates GPSARSA needs $O(nk^2)$ steps, where n is the total number of data points during training and k is the number of representative data points ($k \ll n$). Training complexity for dropout, concrete dropout and bootstrapped DQNs is $O(N)$ where N is the number of neural network parameters. Complexity for BBQN is tripled as it requires three sets of parameters.

3 Evaluation and results

Experiments are conducted using the Cambridge restaurant domain from the PyDial toolkit [22] with a goal-driven user simulator on the semantic level [19]. A user simulator replicates user behavior with sufficient accuracy to optimize model parameters to an acceptable level of performance [24], and is more cost-effective for development and evaluation purposes. We use an error model where confusions to the simulated user input are added. The error model outputs an N-best list of possible user responses.

The input for all models is the full dialogue belief state b of size 268 and the output action space consists of 14 possible actions. A linear kernel was used for GPSARSA. We used DQN with two hidden layers of size 130 and 50. The maximum dialogue length was set to 25 turns and γ was 0.99. The Adam optimiser was used with an initial learning rate of 0.001 [12]. The results are averaged over three different runs.

3.1 Comparison with baselines

Figure 1 shows learning curves for benchmarked models as a function of the training dialogues. From all analysed algorithms, only BBQN reached a performance comparable to state-of-the-art non-parametric approach in terms of the efficiency of exploration as well as the final performance. Moreover, thanks to implicit regularization due to the KL constraint, the learning becomes much more stable comparing to vanilla DQN.

Three other analyzed methods, dropout, concrete dropout and bootstrapped approach, did not help improving learning rate over the vanilla ϵ -greedy algorithm neither do they stabilize exploration. Although with concrete dropout tuning of the dropout probability is automatic, it did not help improve efficiency. We also optimize over number of heads with bootstrapped DQN, however, the performance did not vary substantially yielding the best results with 5 heads.

For α -divergences, we find all settings of α do not perform better than VI in general, for $k > 1$ samples. For clarity we did not show here learning curves. Taking more MC samples decreases the variance of the gradient estimates, and the averaged loss for most updates is closer to the loss obtained when taking a sample close to the mean of the variational distribution q . This implies more updates are necessary to move in the direction of the true posterior distribution p , resulting in slower convergence to an optimal policy.

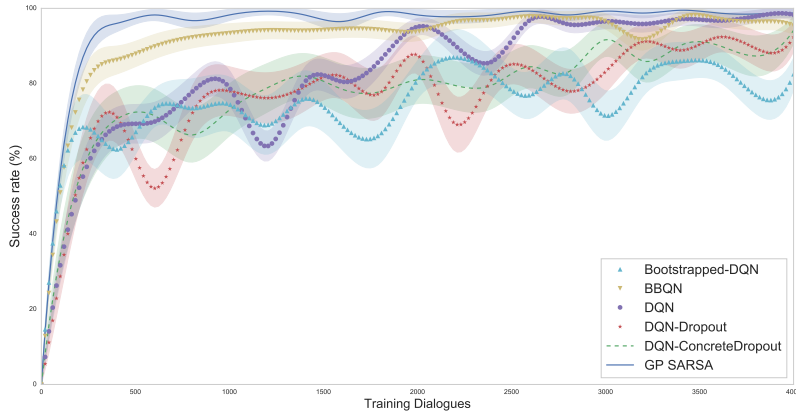


Figure 1: The success rate learning curves for all analyzed models under noise-free conditions.

3.2 Noise-robustness

We also investigated the impact of noise by training all models with the simulated user with a 15% semantic error rate, then evaluated on 45% semantic error rate to examine the generalisation capabilities of different algorithms. The final success rates are given in Figure 2 as a function of the training dialogues.

The results show that GPSARSA performs best in terms of success rate, followed closely by BBQN. This shows that BBQN generalizes better than ϵ -greedy algorithms. BBQN has the potential for robust performance, and performs well, even at conditions different from the training conditions. All other methods fall behind substantially with only vanilla DQN being able to reach similar performance at the end of the training.

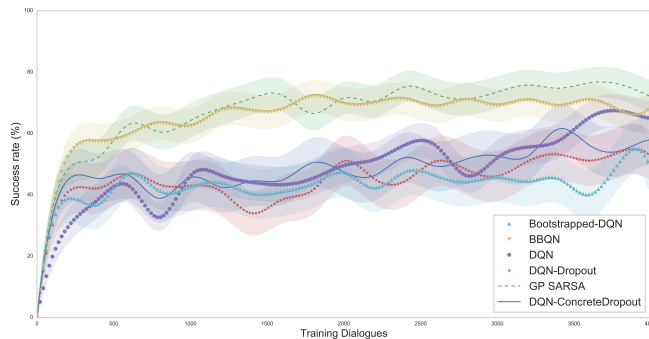


Figure 2: The success rate learning curves for all analyzed models with a 45% confusion rate at testing, and 15% confusion rate during training.

References

- [1] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *International Conference on Machine Learning*, 2015.
- [2] Lu Chen, Xiang Zhou, Cheng Chang, Runzhe Yang, and Kai Yu. Agent-aware dropout dqn for safe and efficient on-line dialogue policy learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2444–2454, 2017.
- [3] Heriberto Cuayáhuitl, Simon Keizer, and Oliver Lemon. Strategic dialogue management via deep reinforcement learning. *NIPS Workshop on Deep Reinforcement Learning*, 2015.
- [4] Lucie Daubigny, Milica Gašić, Senthilkumar Chandramohan, Matthieu Geist, Olivier Pietquin, and Steve Young. Uncertainty management for on-line optimisation of a pomdp-based large-scale spoken dialogue system. In *Interspeech 2011*, pages 1301–1304, 2011.
- [5] Mehdi Fatemi, Layla El Asri, Hannes Schulz, Jing He, and Kaheer Suleman. Policy networks with two-stage training for dialogue systems. *Proceedings of SigDial*, 2016.
- [6] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.
- [7] Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. *Neural Information Processing Systems*, 2017.
- [8] Milica Gasic and Steve Young. Gaussian processes for pomdp-based dialogue manager optimization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1):28–40, 2014.
- [9] Matthew S Henderson. *Discriminative methods for statistical spoken dialogue systems*. PhD thesis, University of Cambridge, 2015.
- [10] José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Daniel Hernández-Lobato, Thang D Bui, and Richard E Turner. Black-box α -divergence minimization. *International Conference on Machine Learning*, 2016.
- [11] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13. ACM, 1993.
- [12] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *The International Conference on Learning Representations*, 2015.
- [13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Proceedings of the 2nd International Conference on Learning Representations*, 2014.
- [14] Esther Levin, Roberto Pieraccini, and Wieland Eckert. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on speech and audio processing*, 8(1):11–23, 2000.
- [15] Yingzhen Li and Yarin Gal. Dropout inference in bayesian neural networks with alpha-divergences. *International Conference on Machine Learning*, 2017.
- [16] Zachary C Lipton, Jianfeng Gao, Lihong Li, Xiujun Li, Faisal Ahmed, and Li Deng. Efficient exploration for dialogue policy learning with bbq networks & replay buffer spiking. *NIPS Workshop on Deep Reinforcement Learning*, 2016.
- [17] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.
- [18] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In *Advances in Neural Information Processing Systems*, pages 4026–4034, 2016.

- [19] J. Schatzmann and S. Young. The hidden agenda user simulation model. *TASLP*, 17(4):733–747, 2009.
- [20] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [21] Pei-Hao Su, Pawel Budzianowski, Stefan Ultes, Milica Gasic, and Steve Young. Sample-efficient actor-critic reinforcement learning with supervised data for dialogue management. *Proceedings of SigDial*, 2017.
- [22] S Ultes, L Rojas-Barahona, P-H Su, D Vandyke, D Kim, I Casanueva, P Budzianowski, N Mrkšić, T-H Wen, M Gašić, and S Young. Pydial: A multi-domain statistical dialogue system toolkit. In *Proc. of ACL*, 2017.
- [23] Jason D Williams, Kavosh Asadi, and Geoffrey Zweig. Hybrid code networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. *Association for Computational Linguistics (ACL)*, 2017.
- [24] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.
- [25] Steve J Young. Talking to machines (statistically speaking). *InterSpeech*, 2002.