BENCHMARKING UNCERTAINTY ESTIMATES WITH DEEP REINFORCEMENT LEARNING FOR DIALOGUE POLICY OPTIMISATION

Christopher Tegho, Paweł Budzianowski*, Milica Gašić*

Cambridge University Engineering Department, Trumpington Street, Cambridge CB2 1PZ, 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 (GP-SARSA) estimate uncertainties and sample actions 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 thorough analysis of Bayes-By-Backpropagation DQN (BBQN). In addition we examine dropout, its concrete variation, bootstrapped ensemble and α-divergences as other means to extract uncertainty estimates from DQN. We find that BBQN achieves faster convergence to an optimal policy than any other method, and reaches performance comparable to the state of the art, but without the high computational complexity of GP-SARSA.

Index Terms— dialogue management, reinforcement learning, Bayesian neural networks.

1 Introduction

Spoken Dialogue Systems (SDSs) allow human users to interact with computers through speech. SDSs have became a common deployment in the speech interfaces in mobile phones, and are gaining greater commercial use.

Statistical approaches to dialogue modelling allow automatic optimisation of the SDS behaviour. The Partially Observable Markov Decision Process (POMDP) framework [1] overcomes the problem of noisy estimates of spoken language understanding by assuming the dialogue state is only partially observable. A distribution over states is maintained, which is called the belief state, and a dialogue policy maps the belief state into an appropriate action at every dialogue turn. The ability to generalise across different noise levels is essential for successful dialogue policy operation.

Reinforcement leaning is used to learn a policy that maximizes the expected sum of rewards received after visiting a state [2], and an action-state value function Q is computed for this purpose. For dialogue systems, the state-action space is large, and a Q-function approximation is necessary. To explore the environment, an ε-greedy policy can be employed, where a greedy action is taken w.r.t. the estimated Q-function with probability $1-\varepsilon$, and a random one with probability ϵ. However, given the large action-state space, a randomly exploring Q-learner is not sample efficient. Convergence to an optimal policy is slow and successful dialogues are hard to achieve. This is especially the case in on-line learning, where the user is potentially subjected to poor behaviour.

The Q-function value of each state-action pair can be augmented with an estimate of its uncertainty to guide exploration, and achieve faster learning and a higher reward during learning [3]. Gaussian Processes (GPs) provide an explicit estimate of uncertainty [4], overcoming the above-mentioned problems. However, GP-SARSA requires computing the inverse of a Gram matrix K for determining the predictive posterior and estimating Q at new locations, which prohibits its use for large action spaces.

Deep neural network (DNN) models on the other hand scale well with data and are computationally less complex than GPs. They proved to be well suited for policy management task [5, 6, 7]. However, they do not directly provide an estimate of uncertainty, relying on ε-greedy exploration, which lowers the sample efficiency.

Building upon recent advancements in Bayesian deep learning [8, 9, 10, 11], we perform an extensive benchmark of uncertainty estimates in the dialogue domain. Specifically, following [12], this paper compares the Bayes By Backprop method for deep-Q-networks (BBQN) to GP-SARSA and other deep RL methods. We investigate how BBQN can be improved and be made competitive with GP-SARSA. We show that BBQN learns dialogue policies with more efficient exploration than other deep Bayesian methods, and reaches performance comparable to the state of the art in policy optimization, namely GP-SARSA, particularly in high noise conditions.

* Both authors contributed equally.

2 Uncertainty in DNNs

To obtain uncertainty estimates from a neural network, Bayesian neural networks (BNNs) can be employed [13]. Instead of having single fixed value weights in the neural networks w, 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 [8].

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

$$Q(b,a) = \mathbb{E}_\pi\{r_t + \gamma r_{t+1} + ... \ |b_t = b, a_t = a\},$$

where r_t is the one-step reward received at a given time t and γ is a discount factor. We model action-value function using a deep neural network by iteratively improving our guess by minimizing the loss:

$$L(w_t) = \mathbb{E}\left[(y_t - \hat{Q}^\pi(b_t, a_t; w_t))^2\right] \tag{1}$$

where targets are $y_t = r_t + \gamma \max_{a'} \hat{Q}^\pi(b_{t+1}, a'; w_t)$ and the expectation is usually taken with respect to ϵ-greedy policy [14].

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:

$$Q(b,a) = \mathop{\mathbb{E}}_{P(w|\mathcal{D})}[Q|b,a,w]. \tag{2}$$

Taking an expectation under the posterior distribution is equivalent to using an ensemble of an uncountably infinite number of neural networks, which is intractable [8]. We have to resort to sampling-based variational inference or stochastic variational inference.

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:

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-Lieber ($\mathcal{KL}$) divergence between the variational approximation $q(w|\theta)$ and the true posterior on the weights $P(w|\mathcal{D})$. The resulting cost function is termed as the variational free energy [15]:

$$\mathcal{F} = \mathcal{KL}[q(w|\theta)||P(w)] - \mathop{\mathbb{E}}_{q(w|\theta)}[\ln P(\mathcal{D}|w)]. \tag{3}$$

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 [16]. 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 [12]. The variational parameters are then $\theta = \{\mu_i, \rho_i\}_{i=1}^{\mathcal{D}}$ 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. 3 can then be approximated as:

$$\mathcal{F}(\mathcal{D}, \theta) \approx \sum_{i=1}^{n} \log q(w^{(i)}|\theta) - \log P(w^{(i)}) - \log p(\mathcal{D}|w^{(i)}) \tag{4}$$

where $w^{(i)}$ is the ith Monte Carlo sample drawn from the variational posterior $q(w^{(i)}|\theta)$. For the objective function in Eq. 3, we use the expected square loss. Note that least-squares regression techniques can be interpreted as maximum likelihood with an underlying Gaussian error model.

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 $\alpha \to 0$ [17]. The α-divergence measures the similarity between two distributions and can take the form:

$$D_\alpha[p||q] = \frac{1}{\alpha(\alpha-1)}(1 - \int p(\theta)^\alpha q(\theta)^{1-\alpha} d\theta), \tag{5}$$

where $\alpha \geqslant 0$.

Hernandez-Lobato et al. [17] 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 [17]. $\alpha = 0.5$ 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 [10] :

$$\mathcal{L}_\alpha \approx \check{\mathcal{L}}_\alpha = \mathcal{KL}[q(w|\theta)||P(w)] - \frac{1}{\alpha}\sum_n \log \mathop{\mathbb{E}}_{q(w|\theta)}[P(\mathcal{D}|w)], \tag{6}$$

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

DQN-Dropout. Another method to obtain uncertainty estimates in deep neural networks is Bayesian inference with dropout [18]. Dropout consists of randomly dropping units

(with some probability d) from the neural network during training [19].

As in the previous methods, dropout can be analyzed from the variational inference perspective (Equation 3). 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 [18].

DQN-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 [11]. Dropout d probability becomes one of the optimized parameters. The concrete distribution relaxation z of the Bernoulli random variable becomes:

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

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

Bootstrapped DQN. Another method to extract uncertainty estimates from DNNs is the bootstrapped method by Osband et al. [9]. Exploration can be improved with random initialization of several neural networks which in ensemble produce reasonable uncertainty estimates for neural networks at low computational cost. To improve efficiency, all networks share the same architecture with a different last layer (head) computing Q-values. Surprisingly, in its default case when all networks share the same memory replay, the algorithm obtained the highest scores. Here we employ this ensemble variant.

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 ($k << n$). Training complexity for dropout, concrete dropout and bootstrapped DQNs is $O(N)$ in every step where N is the number of neural network parameters. Complexity for BBQN is tripled as it requires three set of parameters.

3 Related Work

This work is motivated by the results obtained by Lipton et al. [12], which compares the performance of BBQN to DQN for policy optimisation in a dialogue system in a movie domain. Using more principled exploration, the agent learns a faster and better policy over standard ϵ-greedy and bootstrapped approaches.More recent work shows how uncertainty estimates obtained with dropout can improve safety and efficiency of policy optimization [20]. The authors proposed a student-teacher architecture where a data-driven student policy chooses to update its policy consulting a rule-based system based on uncertainty estimates.

4 Evaluation

Experiments are conducted using the Cambridge restaurant domain from the PyDial toolkit [21] with an agenda-based simulator on the semantic level. The Cambridge restaurant domain consists of a selection of about 150 restaurants, with 8 slots for every restaurant. The input for all models is the full dialogue belief state b of size 268, which includes the last system act and distributions over the user intention and the requestable slots. The summary action space consists of 14 actions.

We use the same DQN architectures with four ways of extracting uncertainty estimates - Bayes by backrop, dropout, concrete dropout and bootstrapped ensemble. All models are trained over 4000 simulated dialogues with mini-batches of 64. The experience replay pool size is 1000 for vanilla, dropout and BBQN DQNs and 6000 for bootstrapped and concrete models. Each sample is a state transition (b_t, a_t, r_t, b_{t+1}).

All deep RL models contains two reLU hidden layers of size 130 and 50. The Adam optimiser is used with a learning rate of 0.001 [22]. For vanilla DQN, an ε-greedy policy is used, which is initially set to a 0.75 value, and annealed to 0.0 after 4000 training dialogues.

4.1 Comparison with baselines

In Figure 1, we show the average success rate, and the average reward for BBQN, DQN, DQN with dropout, DQN with a concrete dropout, bootstrapped DQN and GPSARSA, in a noise-free environment.

We find that GPSARSA learns the fastest and is the most stable, benefiting from the ability of Gaussian Processes to learn from a small amount of data, exploiting the correlations defined by the kernel function. The results show that BBQN reach a performance comparable to GP-SARSA, and DQN in general. DQN reaches a higher final success rate than BBQN and a more stable performance at final stages of the training, but converges much slower, with high instability.

Three other analyzed methods, dropout, concrete dropout and bootstrapped approach, did not help improving learning rate over the vanilla ϵ-greedy algorithm neither they stabilize exploration. Although with concrete dropout tuning of

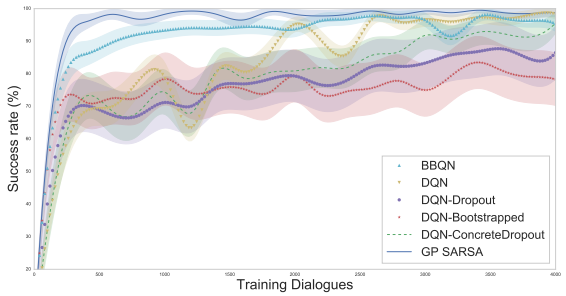

Fig. 1: The success rate learning curves for BBQN, GP-SARSA, DQN, DQN with dropout, DQN with concrete dropout and bootstrapped DQN under noise-free conditions, with two standard error bars.

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 (Figure 2), we find the value $\alpha = 0.5$ or other settings of α do not perform better than VI in general. Convergence to an optimal policy is slower with increasing number of samples. 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, trading off for reduced exploration, and slower learning.

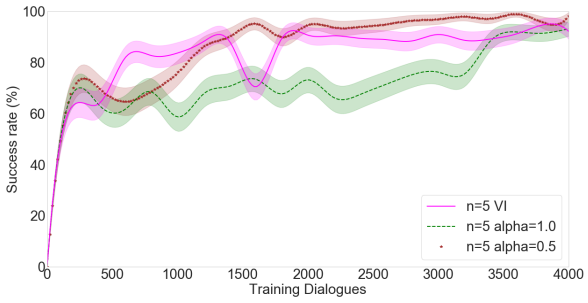

Fig. 2: $K = 5$ MC samples

4.2 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 Fig. 3 as a function of the training dialogues.

The results show that GP-SARSA 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 other methods fall behind substantially with only vanilla DQN being able to reach similar performance at the end of the training.

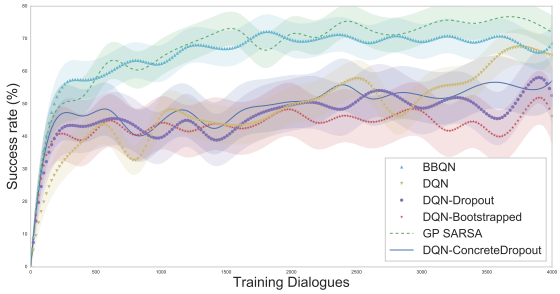

Fig. 3: The success rate learning curves for BBQN, GP-SARSA, DQN, DQN with dropout, DQN with concrete dropout and bootstrapped DQN with a 45% confusion rate at testing, and 15% confusion rate during training.

5 Conclusion

This paper has described how the Bayes-by-backprop method can be applied successfully to obtain uncertainty estimates in DQN (BBQN), when applied to POMDP-based dialogue management. The results obtained confirm that BBQN learns dialogue policies with more efficient exploration than ε-greedy based methods, and reach performance comparable to the state of the art in policy optimization, namely GPSARSA, especially when evaluated on more complex domains. BBQN is also almost as sample efficient as GP-SARSA, but without the computational complexity of GPs. When trained with a noise level of 15%, then evaluated at 45%, BBQN achieved higher performance at higher confusion rates than other deep RL methods. This shows that BBQN generalizes better than other deep RL methods and is as robust as GP-SARSA.

Future research in this area will need to address a number of issues. First, improvements to the uncertainty of estimates for BBQN are needed to improve its sample efficiency. A better stability at later stages of training needs also to be addressed with BBQN. Methods for better hyperparameter tuning need to be considered as well.

6 References

[1] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams, "Pomdp-based statistical spoken dialog systems: A review," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.

[2] 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*, vol. 8, no. 1, pp. 11–23, 2000.

[3] Lucie Daubigney, 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*, 2011, pp. 1301–1304.

[4] Milica Gašić and Steve Young, "Gaussian processes for pomdp-based dialogue manager optimization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 28–40, 2014.

[5] Heriberto Cuayáhuitl, Simon Keizer, and Oliver Lemon, "Strategic dialogue management via deep reinforcement learning," *NIPS Workshop on Deep Reinforcement Learning*, 2015.

[6] 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.

[7] 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.

[8] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra, "Weight uncertainty in neural networks," *International Conference on Machine Learning*, 2015.

[9] Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy, "Deep exploration via bootstrapped dqn," in *Advances in Neural Information Processing Systems*, 2016, pp. 4026–4034.

[10] Yingzhen Li and Yarin Gal, "Dropout inference in bayesian neural networks with alpha-divergences," *International Conference on Machine Learning*, 2017.

[11] Yarin Gal, Jiri Hron, and Alex Kendall, "Concrete dropout," *Neural Information Processing Systems*, 2017.

[12] 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.

[13] Radford M Neal, *Bayesian learning for neural networks*, vol. 118, Springer Science & Business Media, 2012.

[14] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[15] 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*. ACM, 1993, pp. 5–13.

[16] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *International Conference on Learning Representations*, 2014.

[17] 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.

[18] Yarin Gal and Zoubin Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*, 2016, pp. 1050–1059.

[19] 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*, vol. 15, no. 1, pp. 1929–1958, 2014.

[20] 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*, 2017, pp. 2444–2454.

[21] 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.

[22] Diederik Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 2015.