INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Meta Learning for Hyperparameter Optimization in Dialogue System

Jen-Tzung Chien, Wei Xiang Lieow

Department of Electrical and Computer Engineering, National Chiao Tung University, Taiwan

Abstract

The performance of dialogue system based on deep reinforcement learning (DRL) highly depends on the selected hyperparameters in DRL algorithms. Traditionally, Gaussian process (GP) provides a probabilistic approach to Bayesian optimization for sequential search which is beneficial to select optimal hyperparameter. However, GP suffers from the expanding computation when the dimension of hyperparameters and the number of search points are increased. This paper presents a meta learning approach to carry out multifidelity Bayesian optimization where a two-level recurrent neural network (RNN) is developed for sequential learning and optimization. The search space is explored via the first-level RNN with cheap and low fidelity over a global region of hyperparameters. The optimization is then exploited and leveraged by the second-level RNN with a high fidelity on the successively small regions. The experiments on the hyperparameter optimization for dialogue system based on the deep Q network show the effectiveness and efficiency by using the proposed multifidelity Bayesian optimization.

Index Terms: dialogue system, meta learning, Bayesian optimization, recurrent neural network

1. Introduction

Deep reinforcement learning (DRL) provides an appealing solution to learn an advanced dialogue policy which enables human-computer interaction via speech in a spoken dialogue system [1–3]. In general, DRL as a dialogue agent has been successfully developed to explore the interaction between agent and user based on the deep Q network (DQN) which typically explores the interaction between agent and user via the ϵ-greedy heuristics. However, a good success in task-oriented reinforcement learning using DQN relies on an efficient exploration where the topology of deep neural networks and the selection of ϵ are properly selected in the inference for dialogue policy. It becomes crucial to develop a desirable spoken dialogue system where the hyperparameters in DRL model are optimally selected [4]. This paper presents a meta learning for hyperparameter optimization in construction of DRL model for task-oriented dialogue systems.

Basically, the selection of hyperparameters can be formulated as an Bayesian optimization (BO) problem which is tackled according to a sequential search strategy for global optimization over a black-box function [5–7]. BO adopts a Bayesian strategy which treats the unknown objective as a random function and places a prior over it. Sequential search is seen as the continuous learning process for updating the posterior distribution over the objective function. Traditionally, BO for hyperparameter tuning was solved by using the Gaussian process (GP) where an uncertainty model was characterized [8] and the tradeoff between exploration and exploitation was treated during sequential search [9]. However, a critical drawback of GP-based BO is that the inference time increases cubically by the number of observations due to the requirement of a costly computation for the inverse of covariance matrix. Accordingly, the growing complexity of learning model results in the expanding search space along with the increasing number of hyperparameter configurations which are required to be evaluated before finding the solution with sufficient quality or confidence.

This paper tackles the weakness of inefficiency using GP and presents a neural meta learning for hyperparameter tuning. The recurrent neural networks are adopted in sequential search instead of using GP so that the calculation of inverse matrix is disregarded. In particular, the process of hyperparameter tuning is further speed up by means of a multifidelity search process instead of the traditional single fidelity search with expensive evaluation. This multifidelity BO starts from a cheap and low-fidelity evaluation over a global search space and then activates the high-fidelity search only for a local certain region. In implementation of varying fidelities, the auxiliary function or target objective based on the hyperparameters as the inputs and the validation accuracy or meta loss as the output is systematically evaluated. The overall computation time in sequential search is significantly reduced by this neural meta learning. The multifidelity BO does not only identify the informative hyperparameters but also evaluate the belief for a global target optimizer. Experiments on dialogue system show the merit of the efficiency and effectiveness of the proposed hyperparameter optimization based on meta learning using recurrent neural networks.

2. Background survey

This study develops the meta learning for multifidelity Bayesian optimization for hyperparameter tuning in dialogue system.

2.1. Bayesian optimization

Bayesian optimization (BO) [10, 11] seeks to find a global minimizer over an unknown or black-box function f via

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{X}} f(\mathbf{x}) \tag{1}$$

where $\mathcal{X}$ denotes the search space of interest. BO is known as a model-based sequential approach to search the best $\mathbf{x}^*$. There are two processing components. The first component is to calculate a probabilistic model, consisting of a prior distribution that captures our beliefs about the behavior of the unknown objective function and an observation distribution that measures the data generation. The second component is to find an acquisition function, which is optimized at each time step so as to trade-off between exploration and exploitation. BO can be solved by a variety of search strategies including Thompson sampling, information gain, expected improvement, upper confidence bounds and so on. Gaussian process (GP) is popularly used as the probabilistic representation in these strategies. However, the evaluation of black-box function using GP prediction or validation accuracy is too expensive to apply for adjusting

http://dx.doi.org/10.21437/Interspeech.2019-1383

the practical dialogue systems with high-dimensional hyperparameters. The high-cost and single-fidelity evaluation could not work well for hyperparameter optimization. In many cases, in addition to the original expensive evaluation function, the cheap approximation to objective function f is available. Multifidelity BO is implemented to reduce the computation cost by evaluating the low-fidelity auxiliary function globally in search space and then calculating the high-fidelity function in a small but promising region of interest. In [12], the multifidelity bandit optimization was proposed under the probabilistic setting using GP. Instead of using GP prediction with high computation in matrix inversion, this study presents the neural sequential learning where two recurrent neural networks as optimizer and optimizee are organized in a joint framework to perform the hierarchical multi-fidelity evaluation for hyperparameter optimization. The meta learning is implemented by minimizing the meta loss which is integrated with different fidelities.

2.2. Meta learning

Meta learning aims to use metadata to acquire knowledge and understand how automatic learning can become flexible in solving learning problem. Basically, the scope of meta learning is broad and the problem is challenging. In general, meta learning is to build an agent for *learning to learn* [13–15]. To meet this goal, the learner is implemented and trained across different learning algorithms, e.g. gradient descent, simulated annealing and reinforcement learning, which lead to a large space with strong capability for exploration and exploitation. Meta learning can even work out different valuable learning approaches. In [16, 17], meta learning was fulfilled to obtain a trained recurrent neural network (RNN) [18, 19] which was subsequently used as an optimization algorithm by maximizing a differentiable objective to match different models with the observed data. In contrast, in [20], the outputs of meta learning were seen as an RNN which was employed as a model for fitting data by using classical optimizer. In [21], meta learning was developed as an algorithm for globally optimizing the black-box functions. Accordingly, meta learner was trained by using the gradients for distillation. Using the above-mentioned works, the output of meta learning was represented by an RNN. This RNN was interpreted and applied as a model or even an algorithm. Different from previous methods, this study presents the meta learning for hyperparameter optimization in deep reinforcement learning for dialogue system. The output is seen as a multilayer RNN which is applied as an algorithm for fulfilling the multifidelity BO.

3. Multifidelity Meta Learning

In particular, we present an advanced framework for hyperparameter tuning where the multifidelity Bayesian optimization is implemented via neural meta learning. Using this framework, we have access to $M-1$ successively approximations $f^{(1)}, \ldots, f^{(M-1)}$ to the expensive black-box function $f = f^{(M)}$. These approximations are referred as the fidelities which range over different costs with different degrees of approximation accuracy. In the search process, assuming that a query $\mathbf{x}$ at fidelity m spends a cost $c^{(m)}$ for a resource, e.g. computational time or expense of money. As the fidelity m increases, the approximations become more accurate but at the same time the expense of evaluation also becomes more expensive. An algorithm for multifidelity bandits is derived by using a sequence of query-fidelity pairs $\{(\mathbf{x}_t, m_t)\}_{t\geq 0}$, where at time T, the algorithm chooses $(\mathbf{x}_T, m_T)$ using the information from previous query-observation-fidelity triples $\{(\mathbf{x}_t, \mathbf{y}_t, m_t)\}_{t=1}^{T-1}$. This paper develops two layers of RNNs where one RNN is used to explore the search space via the cheap and low-fidelity evaluation and the other RNN is designed to leverage the search by high-fidelity evaluation on those successively smaller regions so as to achieve better regret when compared with the evaluation under single fidelity. Figure 1 depicts the network architecture of meta learning for multifidelity BO. The hyperparameter tuning in dialogue system based on deep Q network is investigated.

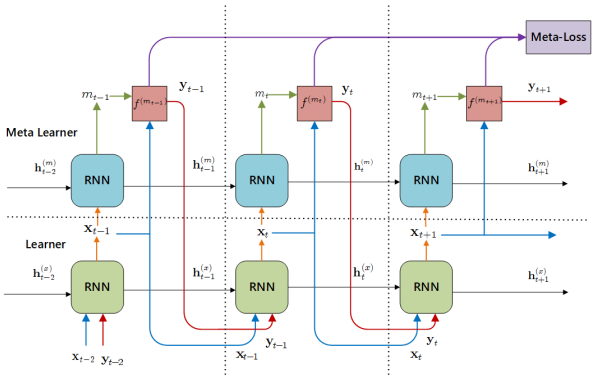

Figure 1: *Computational graph for meta learning of RNN optimizer and optimizee which are unrolled over multiple times.*

Algorithm 1 Training procedure for meta learning

initialize $\mathbf{h}_0^{(x)}$, $\mathbf{h}_0^{(m)}$, $\boldsymbol{\theta}_x$, $\boldsymbol{\theta}_m$, require learning rate η
approximate $\{f^{(m)}\}_{m=1}^{M-1}$ to the expensive function $f = f^{(M)}$
for $t = 1, \ldots, T$ **do**
  $(\mathbf{x}_t, \mathbf{h}_t^{(x)}) = \text{RNN}_{\boldsymbol{\theta}_\mathbf{x}}(\mathbf{h}_{t-1}^{(x)}, \mathbf{x}_{t-1}, \mathbf{y}_{t-1})$
  $(m_t, \mathbf{h}_t^{(m)}) = \text{RNN}_{\boldsymbol{\theta}_m}(\mathbf{h}_{t-1}^{(m)}, \mathbf{x}_t)$
  $\mathbf{y}_t \leftarrow$ query $f^{(m_t)}$ at $\mathbf{x}_t$
  accumulate the objective $\mathcal{L}$
  compute the gradient $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_x}$, $\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_m}$ with REINFORCE
  update the parameters
    $\boldsymbol{\theta}_x \leftarrow \boldsymbol{\theta}_x - \eta \odot \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_\mathbf{x}}$ $\boldsymbol{\theta}_m \leftarrow \boldsymbol{\theta}_m - \eta \odot \frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_m}$
end for

3.1. Multifidelity Bayesian optimization

The procedure of sequential search using multifidelity Bayesian optimization is shown in Figure 1 and summarized by

- given the current hidden state $\mathbf{h}_{t-1}^{(x)}$ and inputs $\mathbf{x}_{t-1}$, $\mathbf{y}_{t-1}$, propose a query point $\mathbf{x}_t$ in first-layer RNN
- input $\mathbf{x}_t$ to second-layer RNN and propose a fidelity m_t which to query at.
- observe the response $\mathbf{y}_t$ from function $f^{(m_t)}(\mathbf{x}_t)$
- update the hidden states of two RNNs $\mathbf{h}_t^{(x)}$ and $\mathbf{h}_t^{(m)}$

This procedure performs the continuous updating of hidden state $\mathbf{h}_{t-1}^{(x)}$ of the first layer of RNN (also called the optimizee or learner) using the input data $\{\mathbf{x}_{t-1}, \mathbf{y}_{t-1}\}$ from the present time step t and then proposing a new query point $\mathbf{x}_t$. This query point is used to select a fidelity m_t using the second layer of RNN (also called the optimizer or meta-learner) which is also continuously updated with hidden state $\mathbf{h}_{t-1}^{(m)}$. A target value $\mathbf{y}_t$

is then collected from $f^{(m_t)}(\mathbf{x}_t)$. The meta loss $\mathcal{L}$ is accumulated from different fidelities m and time steps t for meta optimization. Algorithm 1 illustrates the calculation of gradients of meta loss with respect to two RNNs $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_m$ for parameter updating according to the stochastic gradient descent algorithm. However, the derivatives of black-box objective f are not available, we apply the REINFORCE [22] for optimization.

3.2. Meta learning objective

Meta learning involves two levels of loss function. which closely affect the estimated parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}\}$ for system performance. One is the loss function $\mathcal{L}^o$ for searching global optimum while the other is the loss function $\mathcal{L}^c$ for querying different fidelities for cost minimization given with the distribution of function $p(f)$. There are several choices of loss functions $\mathcal{L}^o$ as well as $\mathcal{L}^c$. A simple global loss function can be defined as the expected loss due to the sample point with the lowest function values which will happen at the final search time T

$$\mathcal{L}^o_{\text{min}}(\boldsymbol{\theta}) = \mathbb{E}_f[f(\mathbf{x}_T)]. \tag{2}$$

The amount of information conveyed in this loss function is temporally sparse because only one sample point is considered. To compensate the sparse samples, a sum of loss functions can be utilized to provide information from a trajectory of time steps

$$\mathcal{L}^o_{\text{sum}}(\boldsymbol{\theta}) = \mathbb{E}_f[\textstyle\sum_{t=1}^T f(\mathbf{x}_t)]. \tag{3}$$

However, optimizing the sum of losses likely chooses a greedy search strategy since the exploration to the region with high function values will be penalized. Optimizing the loss functions in these two extreme cases may be difficult due to the fact that nothing explicitly encourages the optimizer to explore. Accordingly, we may directly incorporate an exploration force into loss function so as to stimulate the exploration in meta learning which is similar to the learning objectives in bandit and BO algorithms. A popular example is the loss function based on the expected improvement (EI)

$$\mathcal{L}^o_{\text{EI}}(\boldsymbol{\theta}) = -\mathbb{E}_f[\textstyle\sum_{t=1}^T \text{EI}(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{y}_{1:t-1})] \tag{4}$$

where $\text{EI}(\cdot)$ denotes the expected posterior improvement of querying $\mathbf{x}_t$ given observations up to time t. Based on this policy, one can encourage exploration by giving an explicit bonus to the optimizer based on posterior improvement rather than just implicitly doing so by means of function evaluations. However, the downside of this search strategy is caused by high computation cost based on the GP probabilistic inference. Alternatively, the observed improvement (OI) is yielded as the objective by

$$\mathcal{L}^o_{\text{OI}}(\boldsymbol{\theta}) = \mathbb{E}_f\left[\sum_{t=1}^T \min\left\{f(\mathbf{x}_t) - \min_{i<t}\left(f(\mathbf{x}_i), 0\right)\right\}\right]. \tag{5}$$

Next, the loss function for querying the fidelity m_t is investigated. Our goal is to achieve a function value after spending the predefined budget or capital Λ of a resource [23]. To meet this goal, we provide the budget bound which implies that the game is assumed to be played infinitely with a bound of regret for all values of Λ. Conceptually, this is similar to the variable time analysis in single-fidelity bandit methods as opposed to the fixed time analysis in conventional methods. Let $\{m_t\}_{t\geq 0}$ denote the fidelities queried by a multifidelity method at each time step t. The cumulative cost is computed by $C_t = \sum_{i=1}^t c^{(m_i)}$. The low fidelity functions $\{f^{(m)}\}_{m=1}^{M-1}$ are cheap for approximation to the expensive objective $f = f^{(M)}$. However, there may be no reward for optimising cheap approximations. A straightforward approach is to minimize the loss for querying the fidelities of inputs $\{\mathbf{x}_t\}_{t=1}^T$ by following the cumulative cost

$$\mathcal{L}^c(\boldsymbol{\theta}) = \mathbb{E}_m[\textstyle\sum_{t=1}^T c^{(m_t)}]. \tag{6}$$

Furthermore, the fidelity loss can be refined as

$$\mathcal{L}^c(\boldsymbol{\theta}) = \mathbb{E}_m\left[\max\left(\Lambda, \sum_{t=1}^T c^{(m_t)}\right) - \Lambda\right] \tag{7}$$

where the budget bound Λ is considered. The loss due to the excess over the predefined budget is penalized in meta learning. Overall, the global loss and fidelity loss is integrated by

$$\mathcal{L} = \lambda\mathcal{L}^o + (1-\lambda)\mathcal{L}^c. \tag{8}$$

λ is a tradeoff parameter between exploration and exploitation for minimization of overall loss. Considering this integrated meta loss, the proposed method can explore the global space with low-fidelity auxiliary function and successively exploit the specific region with high-fidelity evaluation so as to converge on the optimum in a predefined budget.

4. Experiments

4.1. Experimental setup

Neural meta learning based on multifidelity Bayesian optimization was evaluated for hyperparameter optimization in deep reinforcement learning (DRL) for task-oriented dialogue system [24]. The open source end-to-end statistical spoken dialogue system toolkit, Pydial [25], was used. Pydial provided a benchmark environment with different dialogue modules where DRL based on deep Q network (DQN) or other algorithms could be evaluated. The dialogues for seeking the restaurants in San Francisco were evaluated. The hyperparamter ϵ in ϵ-greedy learning and the sizes of first and second hidden layers in policy network under DQN framework were tuned in the ranges [0.1, 1.0] [100, 500], [30, 200], respectively. All the other settings were kept the same. DQN was trained over ten random seeds with 4000 training dialogues and then evaluated by 500 test dialogues. The results were shown by averaging over ten seeds. 20 optimization steps ($t \leq 20$) were fixed. The Adam optimizer with initial learning rate 0.001 was used. The maximum number of turns in dialogue was 25. The discount factor was 0.99. The neural meta learning based on long short-term memory (LSTM) was implemented for Bayesian optimization (BO) and compared with BO using Gaussian process (GP) where the BO-GP package, GPflowOpt [26], was applied. Dialogue performance was assessed by using the metrics of success rate and reward for policy model with hyperparameters learned by different methods. Success rate was defined as the percentage of dialogues which were completed successfully. Reward was defined as $20 \cdot D - T$ where D was the success indicator and T was the dialogue length in turns. In meta learning phase, we used a large number of differentiable functions, which were generated by GPs with different kernel functions and parameters under a specified data range, to train LSTM optimizer using Algorithm 1. We therefore learn a *general* meta learner which is capable of searching the minimum point over any kinds of function distribution $p(f)$ under a specified parameter space. Such an BO using LSTM is different from BO using GP where no training phase is required but computation cost is very high in test time with large number of samples. If meta learner is

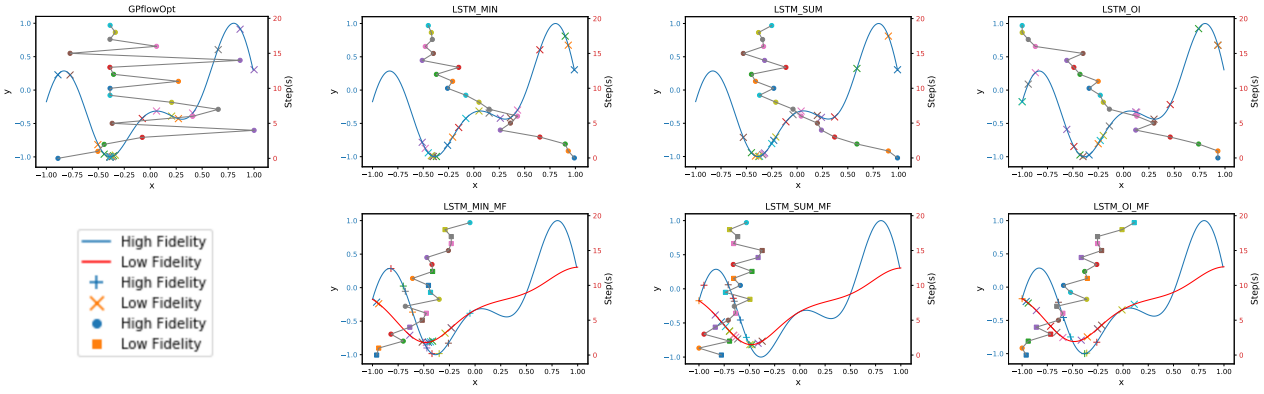

Figure 2: *How different methods tradeoff between exploration and exploitation in one-dimensional space. Blue: high-fidelity function being optimized. Red: low-fidelity function. Crosses: function values at query points. Gray trajectory: query points over 20 steps.*

trained with prior knowledge of objective function, the performance of optimizer using the proposed method can be further improved. In the experiments, GP refers to the function tested by GPflowOpt, LSTM-MIN refers to the LSTM trained by using $\mathcal{L}^o_{\text{min}}$, LSTM-SUM refers to $\mathcal{L}^o_{\text{sum}}$, and LSTM-OI refers to $\mathcal{L}^o_{\text{OI}}$, LSTM-MF refers to LSTM trained using with multifidelity. Cost budget Λ in Eq. (7) was 100. Table 1 compares the run time of 100 optimization steps in seconds by using GP and LSTM-MF where the dimension of parameter space d is varied from 1 to 6. A synthesis task is examined. In the evaluation of test time, the proposed LSTM optimizer runs about 10^4 times faster than GP optimizer over different parameter dimensions.

Table 1: *Run time (in seconds) for 100 optimization steps.*

Method	$d = 1$	$d = 2$	$d = 3$	$d = 6$
GP	1361	1376	1654	2063
LSTM-MF	0.04	0.05	0.08	0.1

Table 2: *Success rates and rewards by using different methods under number of training dialogues N being 2000 and 4000.*

	$N = 2000$		$N = 4000$	
Method	Success rate	Reward	Success rate	Reward
DQN	51.7%	3.5	63.6%	5.6
DQN-VIME	53.5%	3.7	67.8%	6.4
DQN-GP	55.1%	3.9	71.8%	7.5
DQN-LSTM	55.8%	4.1	72.9%	7.4
DQN-LSTM-MF	**56.2%**	**4.5**	**73.9%**	**8.1**

4.2. Experimental result

First of all, a synthesis task is introduced to illustrate how different methods sequentially search for the minimum point. Figure 2 shows the query trajectories $\{\mathbf{x}_t\}_{t=1}^{20}$, for different black-box optimizers in a one-dimensional space. Different optimizers explore initially, and then settle in one mode and later search more locally. GP performs well but the computation cost is much higher than LSTM. LSTM with direct function observations (LSTM-SUM) tends to explore less than the other optimizers and often misses the global optimum while the LSTM trained with the observed improvement (OI) keeps exploring even in later stages. LSTM-MF explores the space $\mathcal{X}$ with lower fidelity and uses the high fidelity in successively smaller regions to converge on the optimum. LSTM with two fidelities clearly performs better search than LSTM with single fidelity.

Table 2 reports the success rates and rewards by using baseline DQN and the extensions of DQN with different hyperparameter tuning. Numbers of training dialogues $N = 2000$ and $N = 4000$ are investigated. For comparison, the variational information maximizing exploration (VIME) [27] is carried out in DQN based DRL. VIME is considered as an exploration strategy based on maximizing the information gain about the agent's belief of environment dynamics. The resulting DQN-VIME is compared with hyperparameter optimization based on BO using different methods. As we can see, the optimizers trained on synthetic GP functions are able to transfer successfully to a very different black-box function for dialogue task. The multifidelity BO is performed to learn a meta learner or optimizer which has the capably of searching the minimum point. System performance is improved by using the number of training data. In this comparison, DQN with hyperparameter optimization attains higher success rate and reward than DQN with fixed hyperparameters and the trained policy using VIME exploration. DQN-GP performs as good as DQN-LSTM with single fidelity. The highest success rates and rewards are achieved by using DQN-LSTM where multifidelity is used in BO. Source codes are accessible at https://github.com/NCTUMLlab/.

5. Conclusions

This paper has presented a multifidelity Bayesian optimization for hyperparameter optimization in reinforcement learning for dialogue system. The black-box optimization was solved by using the hierarchical recurrent neural networks for learner and meta-learner which considerably tackled the computational difficulty by using Gaussian process. In particular, we proposed a multifidelity meta learning where the global search was run via the low-fidelity auxiliary function in the first layer of recurrent neural network while the local search was performed by high-fidelity evaluation using the second layer of recurrent neural network. An integrated loss of optimzee and optimizer was jointly minimized. Experiments on a synthetic task and a real-world dialogue task showed the merit of the proposed neural meta learning driven by multifidelity Bayesian optimization. Future works will be extended for other reinforcement learning algorithms and other speech applications.

6. References

[1] Z. Lipton, X. Li, J. Gao, L. Li, F. Ahmed, and L. Deng, "BBQ-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems," in *Prof. of AAAI Conference on Artificial Intelligence*, 2018, pp. 5237–5244.

[2] J.-T. Chien, "Deep Bayesian natural language processing," in *Proc. of Annual Meeting of the Association for Computational Linguistics : Tutorial Abstracts*, 2019.

[3] S. Watanabe and J.-T. Chien, *Bayesian Speech and Language Processing*. Cambridge University Press, 2015.

[4] F. Dernoncourt and J. Y. Lee, "Optimizing neural network hyperparameters with Gaussian processes for dialog act classification," in *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 406–413.

[5] S. Watanabe and J. L. Roux, "Black box optimization for automatic speech recognition," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 3256–3260.

[6] J. T. Springenberg, A. Klein, S. Falkner, and F. Hutter, "Bayesian optimization with robust Bayesian neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 4134–4142.

[7] J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, M. Patwary, M. Prabhat, and R. Adams, "Scalable Bayesian optimization using deep neural networks," in *International Conference on Machine Learning*, 2015, pp. 2171–2180.

[8] J.-T. Chien and H.-L. Hsieh, "Nonstationary source separation using sequential and variational bayesian learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 5, pp. 681–694, 2013.

[9] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, 2012, pp. 2951–2959.

[10] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[11] J. M. Hernández-Lobato, M. A. Gelbart, M. W. Hoffman, R. P. Adams, and Z. Ghahramani, "Predictive entropy search for bayesian optimization with unknown constraints," in *Proc. of International Conference on Machine Learning*, 2015, pp. 1699–1707.

[12] K. Kandasamy, G. Dasarathy, J. B. Oliva, J. Schneider, and B. Póczos, "Gaussian process bandit optimisation with multi-fidelity evaluations," in *Advances in Neural Information Processing Systems*, 2016, pp. 992–1000.

[13] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*, 2017.

[14] D. Russo and B. Van Roy, "Learning to optimize via posterior sampling," *Mathematics of Operations Research*, vol. 39, no. 4, pp. 1221–1243, 2014.

[15] O. Wichrowska, N. Maheswaranathan, M. W. Hoffman, S. G. Colmenarejo, M. Denil, N. de Freitas, and J. Sohl-Dickstein, "Learned optimizers that scale and generalize," in *Proc. of International Conference on Machine Learning*, 2017, pp. 3751–3760.

[16] M. Andrychowicz, M. Denil, S. Gomez, M. W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, and N. De Freitas, "Learning to learn by gradient descent by gradient descent," in *Advances in Neural Information Processing Systems*, 2016, pp. 3981–3989.

[17] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International Conference on Machine Learning*, 2016, pp. 1842–1850.

[18] J.-T. Chien and Y.-C. Ku, "Bayesian recurrent neural network for language modeling," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 361–374, 2016.

[19] J.-T. Chien and C.-W. Wang, "Variational and hierarchical recurrent autoencoder," in *Proc. of IEEE Internationl Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 3202–3206.

[20] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[21] Y. Chen, M. W. Hoffman, S. G. Colmenarejo, M. Denil, T. P. Lillicrap, and N. de Freitas, "Learning to learn for global optimization of black box functions," *arXiv preprint arXiv:1611.03824*, 2016.

[22] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[23] M. Hoffman, B. Shahriari, and N. Freitas, "On correlation and budget constraints in model-based bandit optimization with application to automatic machine learning," in *Artificial Intelligence and Statistics*, 2014, pp. 365–374.

[24] L. Chen, P.-H. Su, and M. Gasic, "Hyper-parameter optimisation of Gaussian process reinforcement learning for statistical dialogue management," in *Proc. of Annual Meeting of Special Interest Group on Discourse and Dialogue*, 2015, pp. 407–411.

[25] S. Ultes, L. M. Rojas Barahona, P.-H. Su, D. Vandyke, D. Kim, I. Casanueva, P. Budzianowski, N. Mrkšić, T.-H. Wen, M. Gasic, and S. Young, "PyDial: A Multi-domain Statistical Dialogue System Toolkit," in *Proc. of Annual Meeting of the Association for Computational Linguistics (ACL): System Demonstrations*, 2017, pp. 73–78.

[26] N. Knudde, J. van der Herten, T. Dhaene, and I. Couckuyt, "GPflowOpt: a Bayesian optimization library using Tensorflow," *arXiv preprint arXiv:1711.03845*, 2017.

[27] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, "VIME: Variational information maximizing exploration," in *Neural Information Processing Systems*, 2016, pp. 1109–1117.