FlexiBO: Cost-Aware Multi-Objective Optimization of Deep Neural Networks (Supplementary Materials)

Md Shahriar Iqbal
University of South Carolina
Columbia, SC, USA

Jianhai Su
University of South Carolina
Columbia, SC, USA

Lars Kotthoff
University of Wyoming
Laramie, WY, USA

Pooyan Jamshidi
University of South Carolina
Columbia, SC, USA

ABSTRACT

One of the key challenges in designing machine learning systems is to determine the right balance amongst several objectives. For example, when designing deep neural networks (DNNs), one often has to trade off multiple objectives such as accuracy, model size, energy consumption, inference time etc. Typically, there is not a single configuration that performs well in all objectives. Therefore, one is interested in identifying Pareto-optimal designs. Different multi-objective optimization algorithms have been developed to identify Pareto-optimal configurations. However, state-of-the-art multi-objective optimization methods do not consider the different evaluation costs for different objectives. This is, in particular, important for optimizing DNNs where the cost of training to evaluate accuracy is orders of magnitude higher than measuring inference time for a trained network. In this paper, we propose FlexiBO, a flexible Bayesian multi-objective optimization algorithm to address this issue. We formulate a new acquisition metric based on the improvement of Pareto hyper-volume that includes the measurement cost of each objective. Our hyper-volume improvement calculation uses the acquisition metric to select the next sample and objective with maximum information gain per cost for evaluation, rather than all objectives. We apply FlexiBO to optimize 8 state-of-the-art DNN architectures for object detection, natural language processing, and speech recognition. Our results indicate that, for the same evaluation cost, the Pareto-front obtained using FlexiBO has a 46.79% higher contribution to the true Pareto-front and a 33.45% better diversity compared to other state-of-the-art methods.

ACM Reference Format:
Md Shahriar Iqbal, Jianhai Su, Lars Kotthoff, and Pooyan Jamshidi. 2018. FlexiBO: Cost-Aware Multi-Objective Optimization of Deep Neural Networks (Supplementary Materials). In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/1122445.1122456

ICPE 2020, April 20–24, 2020, Edmonton, Canada

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

1 NETWORK HYPERPARAMETERS

FlexiBO runs the Bayesian optimization and guides the entire search process. Each pass through this 6-step search process produces one sample point, and the cycle is repeated to explore the design space.

Step 1—Update Model: The outputs from the performance measurements are fed back to FlexiBO. FlexiBO uses this information to update the posterior distribution of its surrogate model (i.e., the GPs, or RFs).

Step 2—Optimize Acquisition Function: Once the surrogate model has been updated, the acquisition function can then be recomputed and the next configuration will be determined by optimizing the updated acquisition function.

Step 3—Measure Information Gain: Each exploration iteration needs selecting values for each of the parameters. Configuration option values are chosen based on their ability to maximize expected utility. However, the objective to evaluate is selected based on the expected information gain along that particular objective dimension and the process is repeated.

Step 4—Train DNN:. FlexiBO is run on a separate machine than the training and profiling modules. The training of the target DNN is conducted on a cloud server.

Step 5-6—DNN performance measurements: The profiling and measurements are conducted on a target resource-constrained hardware for which we want to optimize the target DNN. FlexiBO contains a configuration service and message request service for communicating with model server and the profiler. The configuration service is used to set hardware/kernel configurations using Dynamic voltage and frequency scaling (DVFS) commands. We built an infrastructure for embedded hardware platforms such as NVIDIA TX1, TX2, Xavier, Nano that automatically set micro-architectural parameters. Message request service sends training configuration to the listener which is responsible to perform training with the network configuration. After the training is finished, the model is sent to the target hardware for performance measurements. FlexiBO initially checks the model server for pre-trained model for a network configuration before sending training request to the training server. Once the optimization budget is exhausted, the Pareto-optimal configuration of the DNN will be returned.

Table 1: Network hyperparameters for different architectures of object detection, NLP and speech recognition DNN systems.

Hyperparameter	**Object detection**					**NLP**	**Speech recognition**
	Xception	MobileNet	LeNet	resnet	SqueezeNet	BERT-base	Deepspeech
Num. channels, ζ	3	3	3	3	3	-	-
Num. classes, $\|C\|$	100	1000	1000	10	10	-	-
Epochs, ϵ	200	100	200	200	200	-	-
Batch size, b	20	32		32	32	-	-
Learning rate, η	0.00001	0.0001	0.0001	0.0001	0.0001	0.00003	0.0001
Decay, λ	$1e^{-6}$	$1e^{-6}$	$1e^{-6}$	$1e^{-6}$	$1e^{-6}$	$3e^{-6}$	$1e^{-6}$
Dropout, p	0.5	0.5	0.5	0.5	0.5	0.248	0.2367
Loss function, L	categorical cross-entropy	categorical cross-entropy	categorical cross-entropy	categorical cross-entropy	categorical cross-entropy	categorical cross-entropy	categorical cross-entropy
Width multiplier, α	1	1	1	1	1	-	-
Data augmentation	Yes	Yes	Yes	Yes	Yes	-	-
Default standard deviation, σ	-	-	-	-	-	-	0.046875
Doc stride	-	-	-	-	-	128	-