Supplementary Material for Scalable Bayesian Optimization Using Deep Neural Networks

Jasper Snoek[*] JSNOEK@SEAS.HARVARD.EDU
Oren Rippel[†*] RIPPEL@MATH.MIT.EDU
Kevin Swersky[§] KSWERSKY@CS.TORONTO.EDU
Ryan Kiros[§] RKIROS@CS.TORONTO.EDU
Nadathur Satish[‡] NADATHUR.RAJAGOPALAN.SATISH@INTEL.COM
Narayanan Sundaram[‡] NARAYANAN.SUNDARAM@INTEL.COM
Md. Mostofa Ali Patwary[‡] MOSTOFA.ALI.PATWARY@INTEL.COM
Prabhat[⋆] PRABHAT@LBL.GOV
Ryan P. Adams[*] RPA@SEAS.HARVARD.EDU

[*]Harvard University, School of Engineering and Applied Sciences
[†]Massachusetts Institute of Technology, Department of Mathematics
[§]University of Toronto, Department of Computer Science
[‡]Intel Labs, Parallel Computing Lab
[⋆]NERSC, Lawrence Berkeley National Laboratory

A. Convolutional neural network experiment specifications

In this section we elaborate on the details of the network architecture, training and the meta-optimization. In the following subsections we elaborate on the hyperparametrization scheme. The priors on the hyperparameters as well as their optimal configurations for the two datasets can be found in Table 2.

A.1. Architecture

The model architecture is specified in Table 1.

Layer type	# Filters	Window	Stride
Convolution	96	3×3	
Convolution	96	3×3	
Max pooling		3×3	2
Convolution	192	3×3	
Convolution	192	3×3	
Convolution	192	3×3	
Max pooling		3×3	2
Convolution	192	3×3	
Convolution	192	1×1	
Convolution	10/100	1×1	
Global averaging		6×6	
Softmax			

Table 1. Our convolutional neural network architecture. This choice was chosen to be maximally generic. Each convolution layer is followed by a ReLU nonlinearity.

A.2. Data augmentation

We corrupt each input in a number of ways. Below we describe our parametrization of these corruptions.

HSV We shift the hue, saturation and value fields of each input by global constants $b_H \sim U(-B_H, B_H)$, $b_S \sim U(-B_S, B_S)$, $b_V \sim U(-B_V, B_V)$. Similarly, we globally stretch the saturation and value fields by global constants $a_S \sim U(\frac{1}{1+A_S}, 1 + A_S)$, $a_V \sim U(\frac{1}{1+A_V}, 1 + A_V)$.

Scalings Each input is scaled by some factor $s \sim U(\frac{1}{1+S}, 1 + S)$.

Translations We crop each input to size 27×27, where the window is chosen randomly and uniformly.

Horizontal reflections Each input is reflected horizontally with a probability of 0.5.

Pixel dropout Each input element is dropped independently and identically with some random probability D_0.

A.3. Initialization and training procedure

We initialize the weights of each convolution layer m with i.i.d zero-mean Gaussians with standard devia-

tion $\frac{\sigma}{\sqrt{F_m}}$ where F_m is the number of parameters per filter for that layer. We chose this parametrization to produce activations whose variances are invariant to filter dimensionality. We use the same standard deviation for all layers but the input, for which we dedicate its own hyperparameter σ_I as it oftentimes varies in scale from deeper layers in the network.

We train the model using the standard stochastic gradient descent and momentum optimizer. We use mini-batch size of 128, and tune the momentum and learning rate, which we parametrize as $1 - 0.1^M$ and 0.1^L respectively. We anneal the learning rate by a factor of 0.1 at epochs 130 and 190. We terminate the training after 200 epochs.

We regularize the weights of all layers with weight decay coefficient W. We apply dropout on the outputs of the max pooling layers, and tune these rates D_1, D_2 separately.

A.4. Testing procedure

We evaluate the performance of the learned model by averaging its log-probability predictions on 100 samples drawn from the input corruption distribution, with masks drawn from the unit dropout distribution.

B. Additional figures for image caption generation

In Figures 1(a) and 1(b) we provide some additional visualization of the results from the image caption generation experiments from Section 4.2 to highlight the behavior of the Bayesian optimization routine. Both figures show the validation BLEU-4 Score on MS COCO corresponding to different hyperparameter configurations as evaluated over iterations of the optimization procedure. In Figure 1(a), these are represented as a planar histogram, where the shade of each bin indicates the total count within it. The current best validation score discovered is traced in black. Figure 1(b) shows a scatter plot of the validation score of all the experiments in the order in which they finished. Validation scores of 0 correspond to constraint violations. These figures demonstrate the exploration-versus-exploitation paradigm of Bayesian Optimization, in which the algorithm trades off visiting unexplored parts of the space, and focusing on parts which show promise.

C. Multimodal neural language model hyperparameters

C.1. Description of the hyperparameters

We optimize a total of 11 hyperparameters of the log-bilinear model (LBL). Below we explain what these hyperparameters refer to.

Model The LBL model has two variants, an additive model where the image features are incorporated via an additive bias term, and a multiplicative that uses a factored weight tensor to control the interaction between modalities.

Context size The goal of the LBL is to predict the next word given a sequence of words. The context size dictates the number of words in this sequence.

Learning rate, momentum, batch size These are optimization parameters used during stochastic gradient learning of the LBL model parameters. The optimization over learning rate is carried out in log-space, but the proposed learning rate is exponentiated before being passed to the training procedure.

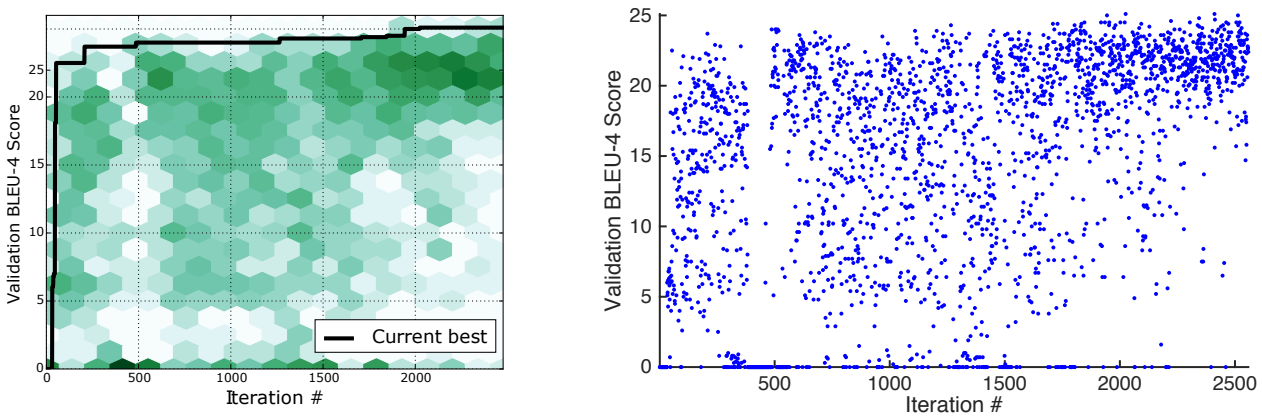
Hidden layer size This controls the size of the joint hidden representation for words and images.

Embedding size Words are represented by feature embeddings rather than one-hot vectors. This is the dimensionality of the embedding.

Dropout A regularization parameter that determines the amount of dropout to be added to the hidden layer.

Context decay, Word decay $\mathcal{L}_2$ regularization on the input and output weights respectively. Like the learning rate, these are optimized in log-space as they vary over several orders of magnitude.

Factors The rank of the weight tensor. Only relevant for the multiplicative model.



(a) Heatmap

(b) Scatter Plot

Figure 1. Validation BLEU-4 Score on MS COCO corresponding to different hyperparameter configurations as evaluated over time. In Figure 1(a), these are represented as a planar histogram, where the shade of each bin indicates the total count within it. The current best validation score discovered is traced in black. Figure 1(b) shows a scatter plot of the validation score of all the experiments in the order in which they finished. This projection demonstrates the exploration-versus-exploitation paradigm of Bayesian Optimization, in which the algorithm trades off visiting unexplored parts of the space, and focusing on parts which show promise.

Hyperparameter	**Notation**	**Support of prior**	**CIFAR-10 Optimum**	**CIFAR-100 Optimum**
Momentum	M	$[0.5, 2]$	1.6242	1.3339
Learning rate	L	$[1, 4]$	2.7773	2.1205
Initialization deviation	σ_I	$[0.5, 1.5]$	0.83359	1.5570
Input initialization deviation	σ	$[0.01, 1]$	0.025370	0.13556
Hue shift	B_H	$[0, 45]$	31.992	19.282
Saturation scale	A_S	$[0, 0.5]$	0.31640	0.30780
Saturation shift	B_S	$[0, 0.5]$	0.10546	0.14695
Value scale	A_S	$[0, 0.5]$	0.13671	0.13668
Value shift	B_S	$[0, 0.5]$	0.24140	0.010960
Pixel dropout	D_0	$[0, 0.3]$	0.19921	0.00056598
Scaling	S	$[0, 0.3]$	0.24140	0.12463
L2 weight decay	W	$[2, 5]$	4.2734	3.1133
Dropout 1	D_1	$[0, 0.7]$	0.082031	0.081494
Dropout 2	D_2	$[0, 0.7]$	0.67265	0.38364

Table 2. Specification of the hyperparametrization scheme, and optimal hyperparameter configurations found.

Hyperparameter	**Support of prior**	**Notes**	**COCO Optimum**
Model	{additive,multiplicative}		additive
Context size	$[3, 25]$		5
Learning rate	$[0.001, 10]$	Log-space	0.43193
Momentum	$[0, 0.9]$		0.23269
Batch size	$[20, 200]$		40
Hidden layer size	$[100, 2000]$		441
Embedding size	$\{50, 100, 200\}$		100
Dropout	$[0, 0.7]$		0.14847
Word decay	$[10^{-9}, 10^{-3}]$	Log-space	2.98456^{-7}
Context decay	$[10^{-9}, 10^{-3}]$	Log-space	1.09181^{-8}
Factors	$[50, 200]$	Multiplicative model only	-

Table 3. Specification of the hyperparametrization scheme, and optimal hyperparameter configurations found for the multimodal neural language model. For parameters marked log-space, the log is given to the Bayesian optimization routine and the result is exponentiated before being passed into the multimodal neural language model for training. Square brackets denote a range of parameters, while curly braces denote a set of options.