Learning to Transfer Initializations for Bayesian Hyperparameter Optimization

Jungtaek Kim[1], Saehoon Kim[1, 2], Seungjin Choi[1]

[1] Department of Computer Science and Engineering, POSTECH
[2] Altrics

BayesOpt 2017 Workshop @ NIPS 2017

Introduction and Motivation

- Suffer from a cold-start problem for finding the best configuration of hyperparameters.
- Learn to mimic human experts' behavior on selecting initial hyperparameters.
- Learn to transfer initializations for hyperparameter optimization.
- Transfer initializations via learned meta-features over datasets [1, 2] using convolutional bi-directional LSTMs.

Background

- Hyperparameter Optimization
  - ✓ Determine the best hyperparameter configuration by minimizing a validation error, given training and validation datasets.
- Sequential Model-Based Optimization (SMBO)
  - ✓ Referred to as Bayesian hyperparameter optimization (BHO).
  - ✓ Search minimum of validation error via BHO, gradually accumulating a pair of hyperparameters and validation error.
  - ✓ Use GP regression as surrogate function, and expected improvement (EI) [3] and GP upper confidence bound (GP-UCB) [4] as acquisition functions.
  - ✓ Enable BHO to use non-zero mean function for GP, in addition to zero mean function.

arXiv version is available.

- https://arxiv.org/abs/1710.06219

Selected References

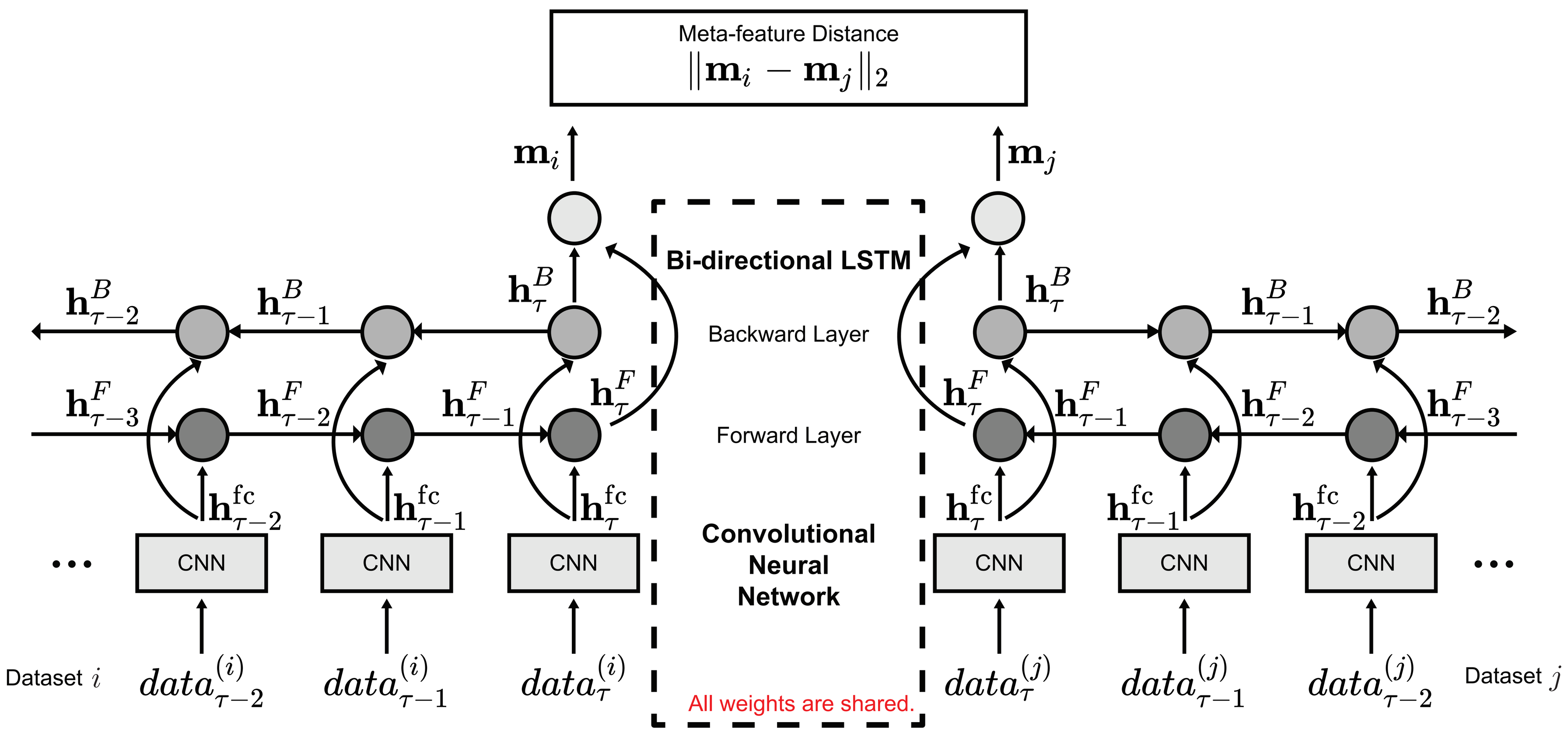
[1] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. Machine learning, neural and statistical classification. Ellis Horwood, 1994.

[2] M. Feurer, J. T. Springerberg, and F. Hutter. Initializing Bayesian hyperparameter optimization via meta-learning. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Austin, TX, USA, 2015.

[3] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. Towards Global Optimization, 2:117–129, 1978.

[4] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In Proceedings of the International Conference on Machine Learning (ICML), Haifa, Israel, 2010.

Proposed Method: Meta-Feature Learning with Siamese Architecture



- Extract a meta-feature from datasets via convolutional bi-directional LSTM, a wing of Siamese architecture.
- Minimize loss function of our network

$$\mathcal{L}(\mathcal{D}^{(i)}, \mathcal{D}^{(j)}) = \left[d_{\text{target}}(\mathcal{D}^{(i)}, \mathcal{D}^{(j)}) - \|\mathbf{m}_i - \mathbf{m}_j\|_2\right]^2.$$

- Select k-nearest datasets, after measuring the distance between new test dataset and known training datasets.
- Obtain the best previous configuration from each one of the nearest datasets and GP prior mean function from the average of the nearest datasets.

Algorithm 1 Meta-feature Learning over Datasets

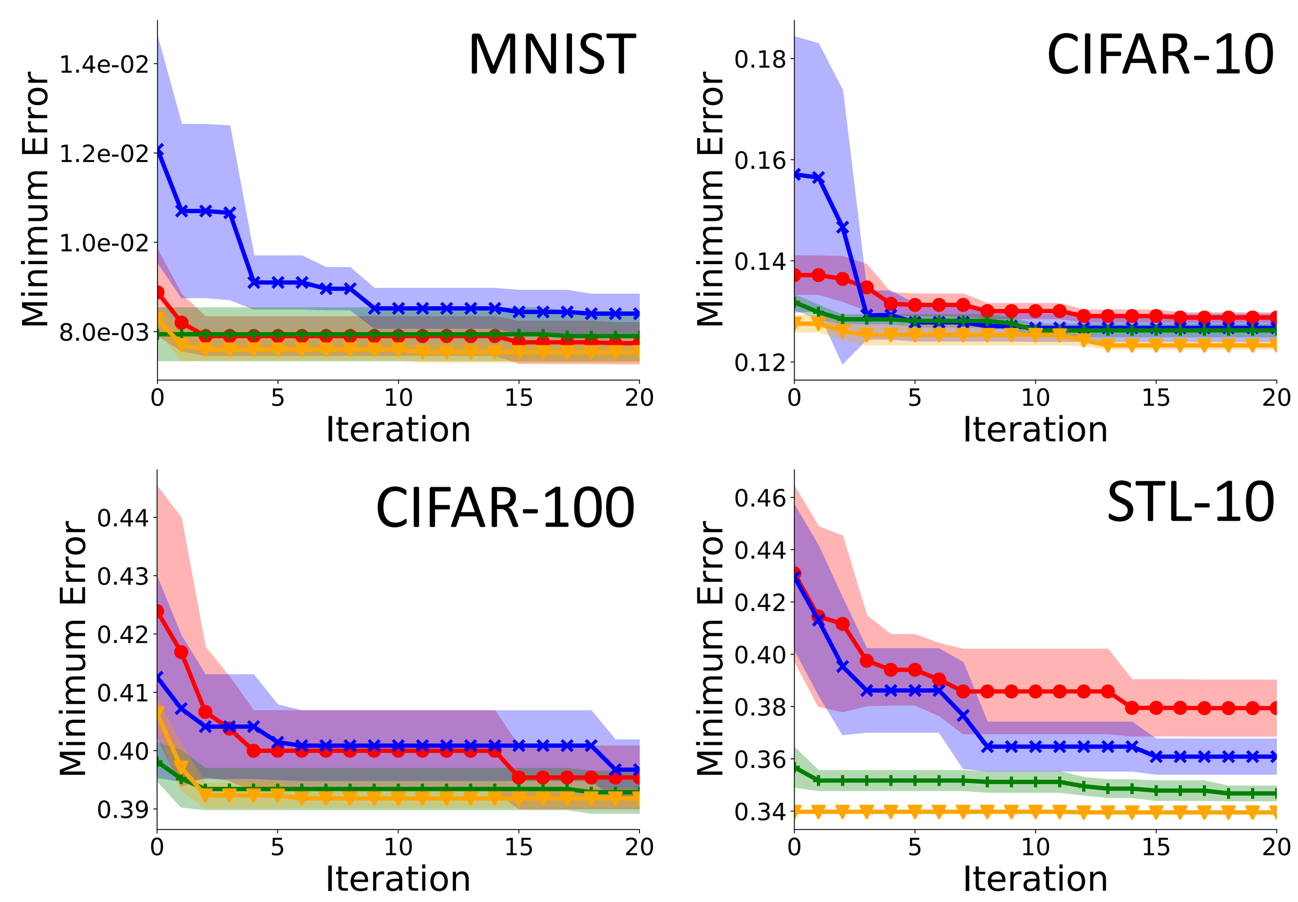
Input: A set of n datasets $\{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$, target distance function $d_{\text{target}}(\cdot,\cdot)$, batch size $\beta \in \mathbb{N}$, step size $\tau \in \mathbb{N}$, number of iterations $T \in \mathbb{N}$
Output: Siamese LSTM model $\mathcal{M}_{\text{S-LSTM}}$ trained over $\{\mathcal{D}_1, \ldots, \mathcal{D}_n\}$
1: Initialize $\mathcal{M}_{\text{S-LSTM}}$.
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Sample β different pairs of datasets, i.e., $\{(\mathcal{D}_i, \mathcal{D}_j)\}$ for $|i \neq j| = \beta, i, j = 1, \ldots, n$.
4: Sample τ data points from each dataset in the pair $\{(\mathcal{D}_i, \mathcal{D}_j)\}$ selected above, to make $|\mathcal{D}_i| = |\mathcal{D}_j| = \tau$.
5: Update parameters in $\mathcal{M}_{\text{S-LSTM}}$ using $d_{\text{target}}(\cdot,\cdot)$ and $\{(\mathcal{D}_i, \mathcal{D}_j)\}$ via backpropagation.
6: **end for**
7: **return** $\mathcal{M}_{\text{S-LSTM}}$

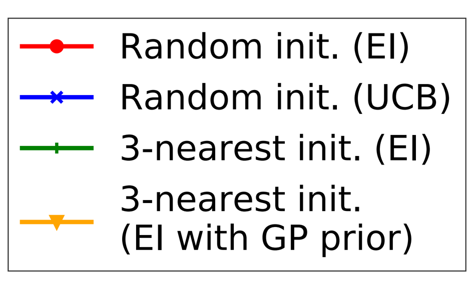
Algorithm 2 Bayesian Hyperparameter Optimization with Transferred Initial Points and GP Prior

Input: Learned Siamese LSTM model $\mathcal{M}_{\text{S-LSTM}}$, target function $\mathcal{J}(\cdot)$, limit $T \in \mathbb{N} > k$
Output: Best configuration of hyperparameters $\boldsymbol{\theta}^*$
1: Find k-nearest neighbors using the learned Siamese bi-directional LSTM, $\mathcal{M}_{\text{S-LSTM}}$.
2: Obtain k classification accuracy histograms over hyperparameters $\{\mathcal{H}_1, \ldots, \mathcal{H}_k\}$.
3: **for** $i = 1, 2, \ldots, k$ **do**
4: Find the best configuration $\boldsymbol{\theta}_i$ on grid of the i-th histogram $\mathcal{H}_i$.
5: Evaluate $\mathcal{J}_i = \mathcal{J}(\boldsymbol{\theta}_i)$.
6: **end for**
7: **for** $j = k+1, k+2, \ldots, T$ **do**
8: $\mathcal{M} \leftarrow$ GP regression with the prior mean function $\frac{1}{k}\sum_{h=1}^{k} \mathcal{H}_h$ on $\{(\boldsymbol{\theta}_i, \mathcal{J}_i)\}_{i=1}^{j-1}$.
9: Find $\boldsymbol{\theta}_j = \arg\max_{\boldsymbol{\theta}} a(\boldsymbol{\theta}|\mathcal{M})$.
10: Evaluate $\mathcal{J}_j = \mathcal{J}(\boldsymbol{\theta}_j)$.
11: **end for**
12: **return** $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}_j \in \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_T\}} \mathcal{J}_j$

Experimental Results

- We trained our network using a pair of subsampled datasets (5 classes, 10,000 images) from MNIST, CIFAR-10, ImageNet 200, and Places 205.
- The target distance was measured by L_1 distance between all configurations in previously observed mappings of subsampled datasets.





Conclusions

- We showed that the Siamese network can learn a distance function between two datasets.