Kernel Methods for Neural Architecture Search

Jasraj Singh

12 June, 2025

1 Summary

The aim of this project was to propose a principled (training-free) metric for scoring models on a given dataset, thereby introducing a new strategy for Neural Architecture Search (NAS). The score is defined as the (kernel) canonical correlation [1, 15] between the inputs, $\mathbf{X}$, and the outputs, $\mathbf{Y}$, with respect to the Reproducing Kernel Hilbert Spaces (RKHS) corresponding to the Neural Tangent Kernel (NTK) and the linear kernel, respectively. We provide a theoretical motivation for this metric, and aim to validate its efficacy on NAS benchmarks: NAS-Bench-201 [6] and DARTS [14].

2 Introduction

Selecting Neural Network (NN) architectures for a task has traditionally been a manual, time-intensive process requiring domain expertise and extensive trial-and-error. For example, cross-validation is a popular choice for model selection, involving training of a number of randomly initialized models. NAS addresses this challenge by automating the discovery of high-performing architectures within a predefined search space [16]. Most methods for NAS incur a high search cost in the form of (partially) training the architectures to score them, and/or training a *search model* that can make the architecture search efficient. Recently, there has been a increased focus on cheapening the search process, while retaining or improving the quality of the selected architectures. Some of these are NTK-based methods like TE-NAS [3], which uses the condition number of the NTK Gram Matrix, and KNAS [17], which uses the mean of the Matrix's entries. These studies report competitive performances on real-world computer vision tasks in NAS benchmarks [6, 14, 18].

Citing the success of NTK-based NAS methods that use correlation between architecture scores and the corresponding test accuracy as a measure of their strategy's efficacy, we aim to propose a scoring method that is closely correlated to the training loss. As with other NTK-based methods, it will utilize the Gram matrix associated with the NTK, but for the purpose of computing the kernel canonical correlation (KCC) [1, 15] between the inputs and the outputs.

3 Theory

For simplicity, we assume that we have a univariate regression task at hand. Consider the set of neural network functions, $\mathcal{A}$, parameterized by some architecture, $\mathbf{A}$. Most neural networks designed for regression have a linear output layer, and we assume the same for $\mathbf{A}$. In that case, minimizing the Mean Squared Error (MSE) between the network output and the regression labels, is equivalent to maximizing their correlation, since the parameters of the output layer

can be adjusted post-training to get the minimizer of the MSE [7]. Accordingly, we define the score for architecture $\mathbf{A}$ as

$$S(\mathbf{A}) \coloneqq \max_{f \in \mathcal{A}} \operatorname{Corr}(f(\mathbf{X}), \mathbf{Y}) = \max_{f \in \mathcal{A}, g \in \mathcal{L}} \operatorname{Corr}(f(\mathbf{X}), g(\mathbf{Y})) \tag{1}$$

where $\mathcal{L}$ is the space of linear functions on $\mathbf{Y}$. This optimization is, in general, hard to perform. <!– Furthermore, it is somewhat unprincipled since the subset of functions accessible to gradient descent is not the entirety of $\mathcal{A}$. –> Instead, if we were to optimize over some RKHS, then the optimization is equivalent to performing kernel canonical correlation analysis (KCCA) with the associated reproducing kernel on $\mathbf{X}$ and the linear kernel on $\mathbf{Y}$. Accordingly, we seek an RKHS that can approximate the space of functions the network can converge to.

3.1 Neural Tangent Kernel

Consider an L-layer fully-connected feed-forward network under the NTK parameterization:

$$\mathbf{x}^{l+1} = \mathbf{W}^{l+1}\mathbf{x}^{l} + \mathbf{b}^{l+1} \tag{2}$$

where $W_{ij} \sim \mathcal{N}(0, \sigma_w^2/n_l)$ and $b_i \sim \mathcal{N}(0, \sigma_b^2)$, n_l being the width of layer l. We define $\theta^l \coloneqq \operatorname{vec}\left(\left\{W^l, b^l\right\}\right)$ as the collection of parameters in layer l, and $\theta \coloneqq \operatorname{vec}\left(\cup_{l=1}^{L} \theta^l\right)$ as the collection of all parameters.

The parameter dynamics and the predictive dynamics for this model under gradient flow can be written as:

$$\dot{\theta}_t = -\eta \nabla_\theta \mathcal{L}(\mathcal{D}; \theta_t) = -\eta \nabla_\theta f(\mathbf{X}; \theta_t)^T \nabla_f \mathcal{L}(\mathcal{D}; \theta_t) \tag{3}$$

$$\dot{f}(\mathbf{X}; \theta_t) = \nabla_\theta f(\mathbf{X}; \theta_t)\dot{\theta}_t = -\eta \underbrace{\nabla_\theta f(\mathbf{X}; \theta_t) \nabla_\theta f(\mathbf{X}; \theta_t)^T}_{\triangleq \hat{\Theta}_t(\mathbf{X}, \mathbf{X})} \nabla_f \mathcal{L}(\mathcal{D}; \theta_t) \tag{4}$$

where $\mathcal{D}$ is the training data set, $\mathcal{L}$ is the loss function, η is the learning rate, and $\hat{\Theta}_t$ is the Empirical Neural Tangent Kernel (NTK) [12].

In the infinite-width limit, the NTK converges in distribution to an analytical limit, Θ, and the NNs evolve as linear models [13]. Under gradient flow, the predictive distribution of this wide network converges to a normal distribution [13], $f_{\theta_\infty}^{\text{lin}}(x) \sim \mathcal{N}(\mu_{\text{NN}}(x), \Sigma_{\text{NN}}(x, x))$, where

$$\mu_{\text{NN}}(x) = \Theta(x, \mathbf{X})\Theta(\mathbf{X})^{-1}\mathbf{Y} \tag{5}$$

$$\begin{aligned}\Sigma_{\text{NN}}(x, x') = \mathcal{K}(x, x') + \Theta(x, \mathbf{X})\Theta(\mathbf{X})^{-1}\mathcal{K}(\mathbf{X})\Theta(\mathbf{X})^{-1}\Theta(\mathbf{X}, x') \\ - \left(\Theta(x, \mathbf{X})\Theta(\mathbf{X})^{-1}\mathcal{K}(\mathbf{X}, x') + \mathcal{K}(x', \mathbf{X})\Theta(\mathbf{X})^{-1}\Theta(\mathbf{X}, x)\right)\end{aligned} \tag{6}$$

where $\mathcal{K}$ denotes the NN-GP kernel [5], defined as $\mathcal{K}(x, x') = \mathbb{E}\left[f_\theta(x) \cdot f_\theta(x')\right]$ which also converges in the infinite-width limit.

The covariance Σ_{NN} is inconvenient to deal with, involving two computationally expensive kernel computations, and a series of cubic-time matrix operations. To tackle this, we can augment the forward pass (denoted by $\tilde{f}_\theta$) by adding a random, untrainable function, which results in the distribution at convergence having a GP-posterior-like form, with Θ as the covariance kernel [10], $\tilde{f}_{\theta_\infty} \sim \mathcal{N}(\mu_{\text{NTK}}, \Sigma_{\text{NTK}})$, where $\mu_{\text{NTK}} = \mu_{\text{NN}}$ and:

$$\Sigma_{\text{NTK}}(x, y) = \Theta(x, x') - \Theta(x, \mathbf{X})\,\Theta(\mathbf{X})^{-1}\,\Theta(\mathbf{X}, x') \tag{7}$$

Importantly, in my Bachelor's thesis project [11], we showed that the ratio between $\Sigma_{\text{NN}}(x, x')$ and $\Sigma_{\text{NTK}}(x, x')$ can be tightly upper bounded, and hence, the NTK-GP posterior, $\mathcal{N}(\mu_{\text{NTK-GP}}(x), \Sigma_{\text{NTK-GP}}$ may be considered a reasonable approximation for the predictive distribution, $\mathcal{N}(\mu_{\text{NN}}(x), \Sigma_{\text{NN}}(x, x))$.

3.2 Kernel Canonical Correlation Analysis

We now consider the more practical architectures which have finite width, so that the feature mapping, $x \mapsto \nabla_\theta f_\theta(x)$, associated with the empirical NTK, Θ, is finite dimensional. Hence, the samples from the NTK-GP prior are almost-surely contained in the RKHS, $\mathcal{H}_\Theta$, associated with Θ (not sure about this part). Since the GP posterior does not have support where the prior does not, the posterior samples are also contained in this RKHS. Therefore, we can use the RKHS associated with the NTK to compute the KCC:

$$S(\mathbf{A}) \approx \max_{f \in \mathcal{H}_\Theta, g \in \mathcal{L}} \text{Corr}(f(\mathbf{X}), g(\mathbf{Y})) \tag{8}$$

This value is trivially equal to ± 1 when the kernel matrices associated with $\mathbf{X}$ and $\mathbf{Y}$ are full-rank [9]. A common practice is to add some regularization to this problem by penalizing rougher witness functions, f and g, which yields the following generalized-eigenvalue problem:

$$\begin{bmatrix} 0 & \tilde{\Theta}\tilde{L} \\ \tilde{\mathbf{L}}\tilde{\Theta} & 0 \end{bmatrix} \mathbf{u} = \lambda \begin{bmatrix} \tilde{\Theta}^2 + m\epsilon\tilde{\Theta} & 0 \\ 0 & \tilde{\mathbf{L}}^2 + m\epsilon\tilde{\mathbf{L}} \end{bmatrix} \mathbf{u} \tag{9}$$

where $\tilde{\Theta} = \mathbf{H}\Theta(\mathbf{X})\mathbf{H}$ and $\tilde{\mathbf{L}} = \mathbf{H}\mathbf{Y}\mathbf{Y}^T\mathbf{H}$ are the centered Gram matrices, $\mathbf{H} = \mathbf{I}_m - 1/m \cdot \mathbf{1}_{m \times m}$ is the centering matrix, m is the number of data points and ϵ is the regularization constant. The regularized canonical correlation is the maximum eigenvalue of this problem, $\gamma = \lambda_{\max}$.

4 Limitations and Extensions

The proposed scoring function is expected to be over-confident for two reasons:

1. Using $\Theta(\mathbf{X})$ means that we are optimizing the correlation over the NTK-GP prior's support, which is larger than the posterior's support.

2. The witness function, f, used for computing the canonical correlation is the best NN fitting the training set. This amounts to ignoring the probabilistic information in the prior/posterior altogether. Therefore, it could represent an over-fitting scenario.

3. Using a linear kernel might make sense for regression, but needs justification for classification tasks. What even is an appropriate kernel in the classification case? This is an important issue because most NAS benchmarks involve image classification tasks, like CIFAR-10/100 and ImageNet.

Another limitation is the lack of interpretability of KCC, which is crucial in many real-world applications. To address this limitation, we may explore alternate kernel-based measures, such as those based on Hilbert-Schmidt Independence Criterion (HSIC) [8], and the Kernel Target Alignment (KTA) [4], as proposed in [2].

References

[1] Akaho, S. (2007). A kernel method for canonical correlation analysis.

[2] Chang, B., Kruger, U., Kustra, R., and Zhang, J. (2013). Canonical correlation analysis based on hilbert-schmidt independence criterion and centered kernel target alignment. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 316–324, Atlanta, Georgia, USA. PMLR.

[3] Chen, W., Gong, X., and Wang, Z. (2021). Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. In *International Conference on Learning Representations*.

[4] Cortes, C., Mohri, M., and Rostamizadeh, A. (2012). Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13(28):795–828.

[5] de G. Matthews, A. G., Rowland, M., Hron, J., Turner, R. E., and Ghahramani, Z. (2018). Gaussian process behaviour in wide deep neural networks.

[6] Dong, X. and Yang, Y. (2020). Nas-bench-201: Extending the scope of reproducible neural architecture search. In *International Conference on Learning Representations (ICLR)*.

[7] Englisch, H. and Hiemstra, Y. (1994). The correlation as cost function in neural networks. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 5, pages 3170–3172 vol.5.

[8] Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. (2005a). Measuring statistical dependence with hilbert-schmidt norms. In Jain, S., Simon, H. U., and Tomita, E., editors, *Algorithmic Learning Theory*, pages 63–77, Berlin, Heidelberg. Springer Berlin Heidelberg.

[9] Gretton, A., Herbrich, R., Smola, A., Bousquet, O., and Schölkopf, B. (2005b). Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6(70):2075–2129.

[10] He, B., Lakshminarayanan, B., and Teh, Y. W. (2020). Bayesian deep ensembles via the neural tangent kernel. In *Advances in Neural Information Processing Systems*, volume 33, pages 1010–1022. Curran Associates, Inc.

[11] Hemachandra, A., Dai, Z., Singh, J., Ng, S.-K., and Low, B. K. H. (2023). Training-free neural active learning with initialization-robustness guarantees. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 12931–12971. PMLR.

[12] Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[13] Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. (2019). Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[14] Liu, H., Simonyan, K., and Yang, Y. (2019). DARTS: Differentiable architecture search. In *International Conference on Learning Representations*.

[15] Melzer, T., Reiter, M., and Bischof, H. (2001). Nonlinear feature extraction using generalized canonical correlation analysis. In Dorffner, G., Bischof, H., and Hornik, K., editors, *Artificial Neural Networks — ICANN 2001*, pages 353–360, Berlin, Heidelberg. Springer Berlin Heidelberg.

[16] Poyser, M. and Breckon, T. P. (2024). Neural architecture search: A contemporary literature review for computer vision applications. *Pattern Recognition*, 147:110052.

[17] Xu, J., Zhao, L., Lin, J., Gao, R., Sun, X., and Yang, H. (2021). Knas: Green neural architecture search. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11613–11625. PMLR.

[18] Ying, C., Klein, A., Christiansen, E., Real, E., Murphy, K., and Hutter, F. (2019). NAS-bench-101: Towards reproducible neural architecture search. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7105–7114, Long Beach, California, USA. PMLR.