Spectral Analysis of Kernel and Neural Embeddings: Optimization and Generalization

Abstract

We extend the recent results of (Arora et al., 2019) by a spectral analysis of representations corresponding to kernel and neural embeddings. They showed that in a simple single layer network, the alignment of the labels to the eigenvectors of the corresponding Gram matrix determines both the convergence of the optimization during training as well as the generalization properties. We generalize their result to kernel and neural representations and show that these extensions improve both optimization and generalization of the basic setup studied in (Arora et al., 2019).

1. Introduction

The well-known work of (Zhang et al., 2017) highlighted intriguing experimental phenomena about deep net training – specifically, optimization and generalization – and called for a rethinking of generalization in statistical learning theory. In particular, two fundamental questions that need understanding are:

Optimization. Why do true labels give faster convergence rate than random labels for gradient descent?

Generalization. What property of properly labeled data controls generalization?

(Arora et al., 2019) have recently tried to answer this question in a simple model by conducting a spectral analysis of the associated Gram matrix. They show that both training and generalization are better if the label vector aligns with the top eigenvectors.

However, their analysis applies only to a simple two layer network. *How could their insights be extended to deeper networks?*

A widely held intuitive view is that deep layers generate expressive *representations* of the raw input data. Adopting this view, one may consider a model where a representation generated by successive neural network layers is viewed as a *kernel* embedding which is then fed into the two–layer model of (Arora et al., 2019). The connection between neural networks and kernel machines has long been studied; (Cho & Saul, 2009) introduced kernels that mimic deep networks and (Tsuchida et al., 2018) showed kernels equivalent to certain feed–forward neural networks. Recently, (Belkin et al., 2018) also make the case that progress on understanding deep learning is unlikely to move forward until similar phenomena in classical kernel machines are recognized and understood. Very recently, (Jacot et al., 2018) showed that the evolution of a neural network during training can be related to a new kernel, the Neural Tangent Kernel (NTK) which is central to describe the generalization properties of the network.

Here we pursue this approach by studying the effect of incorporating embeddings in the simple two layer model and we perform a spectral analysis of these embeddings along the lines of (Arora et al., 2019). We can obtain embeddings in several ways:

i. We can use an unbiased kernel such as Gaussian kernel. This choice is consistent with the maximum entropy principle and makes no prior assumption about the data. Or use a kernel which mimics or approximates deep networks

ii. We could use data driven embeddings explicitly produced by the hidden layers in neural networks: either use a subset of the same training data to compute such an embedding, or transfer the inferred embedding from a different (but similar) domain.

While a general transformation $g(x)$ of the input data may have arbitrary effects, one would expect kernel and neural representations to improve performance. The interplay of kernels and data labellings has been addressed before, for example in the work of *kernel–target alignment* (Cristianini et al., 2001).

We do indeed observe a significant beneficial effect:

Optimization. Using kernel methods such as *random Fourier features* (RFF) to approximate the Gaussian kernel embedding (Rahimi & Recht, 2007) and neural embeddings, we obtain substantially better convergence in training.

Generalization. We also achieve significantly lower test error and we confirm that the data dependent spectral measure introduced in (Arora et al., 2019) significantly improves with kernel and neural embeddings.

Thus this work shows empirically that kernel and neural embeddings improve the alignment of target labels to the eigenvectors of the Gram matrix and thus help training and generalization. This suggests a way to extend the insights

of (Arora et al., 2019) to deeper networks, and possible theoretical results in this direction.

2. Spectral Theory

Network model. In (Arora et al., 2019), the authors consider a simple two layer network model:

$$f_{\boldsymbol{W},\boldsymbol{a}}(\boldsymbol{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \max(0, \boldsymbol{w}_r^T \boldsymbol{x}_i), \qquad (1)$$

with $\boldsymbol{x} \in \mathbb{R}^d$, $\boldsymbol{w}_1, .. \ \boldsymbol{w}_m \in \mathbb{R}^{d \times m}$ and $(a_1, .. \ a_m)^T \in \mathbb{R}^m$. These can be written jointly as $\mathbf{a} = (a_1, .., a_m)^T$ and $\mathbf{W} = (\boldsymbol{w}_1, .., \boldsymbol{w}_m)$. This network is trained on dataset of datapoints $\{x_i\}$ and their targets $\{y_i\}$.

They provide a fine grained analysis of training and generalization error by a spectral analysis of the *Gram matrix*:

$$\mathbf{H}_{i,j}^{\infty} := E_{\mathbf{W} \sim \mathcal{N}(0,\mathcal{I})} \left[\mathbf{x}_i^T \mathbf{x}_j 1[\mathbf{w}^T \mathbf{x}_i \geq 0, \mathbf{w}^T \mathbf{x}_j \geq 0] \right]$$

If $\mathbf{H}^{\infty} = \sum_i \lambda_i \mathbf{v}_i \mathbf{v}_i^T$ is the orthonormal decomposition of $\mathbf{H}^{\infty}$, (Arora et al., 2019) show that both training and generalization are better if the label vector y aligns with the eigenvectors corresponding to the top eigvalues of $\mathbf{H}^{\infty}$.

The two-layer ReLU network in this work follows the general structure as in (Arora et al., 2019) with the difference being the addition of an embedding ϕ at the input layer corresponding to a kernel $\mathcal{K}$. The corresponding model is:

$$f_{\mathbf{W},\mathbf{a}}(\mathbf{x}) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \max(0, \mathbf{w}_r^T \boldsymbol{\phi}(\mathbf{x}_i)). \qquad (2)$$

For a representation $(\boldsymbol{\phi}(\boldsymbol{x}_i), i \in [n])$ corresponding to a kernel $\mathcal{K}$, define the Gram Matrix

$$\mathbf{H}(\mathcal{K})_{i,j}^{\infty} := E_{\mathbf{W}} \left[\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) 1[\mathbf{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) \geq 0, \mathbf{w}^T \boldsymbol{\phi}(\boldsymbol{x}_j) \geq 0] \right]$$

and let its eigenvalues be ordered as $\lambda_0(\mathcal{K}) \geq \lambda_1(\mathcal{K}) \geq \cdots \geq \lambda_{n-1}(\mathcal{K})$ and let $\mathbf{v}_0(\mathcal{K}), \cdots, \mathbf{v}_{n-1}(\mathcal{K})$ be the corresponding eigenvectors.

A kernel $\mathcal{K}$ such that the corresponding eigenvectors align well with the labels would be expected to perform well both for training optimization as well as generalization. This is related to kernel target alignment (Cristianini et al., 2001).
Optimization. For the simple two layer network, (Arora et al., 2019) show that the convergence of gradient descent is controlled by

$$\sqrt{\sum_i (1 - \eta \lambda_i)^{2k} (\mathbf{v}_i^T \mathbf{y})^2} \qquad (3)$$

For our kernelized network, the corresponding convergence is controlled by

$$\sqrt{\sum_i (1 - \eta \lambda_i(\mathcal{K}))^{2k} (\mathbf{v}(\mathcal{K})_i^T \mathbf{y})^2} \qquad (4)$$

Generalization. For the simple two layer network, (Arora et al., 2019) show that the generalization performance is controlled by

$$\mathbf{y}^T (\mathbf{H}^{\infty})^{-1} \mathbf{y} \qquad (5)$$

For our kernelized two layer network, the corresponding data *and* representation dependent measure is:

$$\mathbf{y}^T (\mathbf{H}(\mathcal{K})^{\infty})^{-1} \mathbf{y} \qquad (6)$$

3. Experiments

We perform our experiments on two commonly-used datasets for validating deep neural models, i.e., MNIST and CIFAIR-10. These datasets are used for the experiments in (Arora et al., 2019). As in their work we only look at the first two classes and set the label $y_i = +1$ if image i belongs to the first class and $y_i = -1$ if it belongs to the second class. The images are normalized such that $\|\boldsymbol{x}_i\|_2 = 1$. This is also done for kernel embeddings such that $\|\boldsymbol{\phi}(\boldsymbol{x}_i)\|_2 = 1$.
The weights in equation (2) are initialized as follows:

$$\boldsymbol{w}_i \sim \mathcal{N}(0, k^2 \mathcal{I}), \ a_r \sim \text{Unif}(\{-1, 1\}), \forall r \in m. \qquad (7)$$

We then use the following loss function to train the model to predict the image labels.

$$\Phi(\mathbf{W}, \mathbf{a}) = 1/2 \sum_{(i=1)}^{n} (y_i - f_{\mathbf{W},\mathbf{a}}(\mathbf{x}))^2 \qquad (8)$$

For optimization, we use (full batch) gradient descent with the learning rate η. In our experiments we set $k = 10^{-2}, \eta = 2 \cdot 10^{-4}$ similar to (Arora et al., 2019).

3.1. Gaussian kernel method

We first use the Gaussian kernel $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) := \exp\left(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right)$. The corresponding embedding is infinite dimensional, hence we consider the fast approximations to the kernel given by *random Fourier features* (RFF) (Rahimi & Recht, 2007). The idea of random Fourier features is to construct an explicit feature map which is of a dimension much lower than the number of observations, but the resulting inner product approximates the desired kernel function. We use $\gamma = 1$ in all our experiments.
Optimization. We first investigate the use of Gaussian kernel for a more efficient optimization of the loss function on the training data. Figures 1(a) and 1(b) show the training loss at different steps respectively on MNIST and CIFAR-10 datasets. We consistently observe that the different Gaussian kernels (specified by various dimensions of the kernel) yields faster convergence of the optimization procedure on both datasets. MNIST is a simple dataset which

(a) MNIST training loss

(b) CIFAR-10 training loss

(c) CIFAR-10 projections
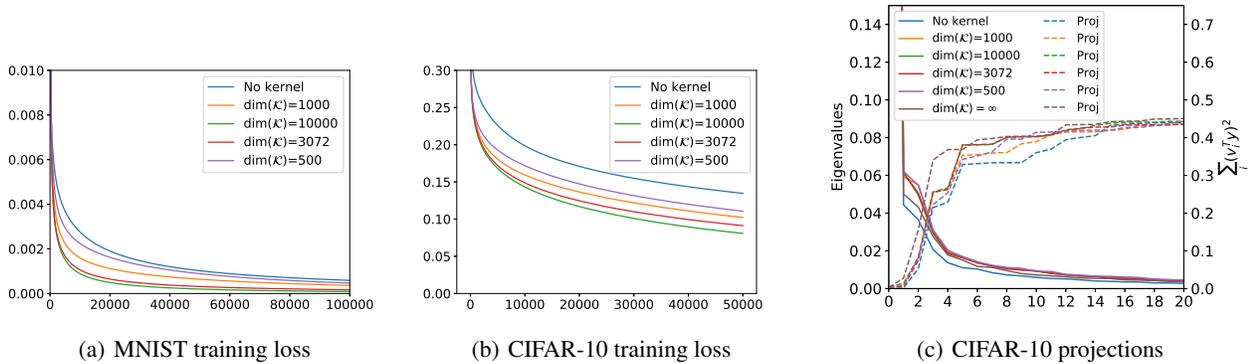
Figure 1. Performance on MNIST and CIFAR-10 training datasets. We observe that the different kernels yield faster convergence of the loss function on training data compared to non-kernel variant. Figure 1(c) demonstrates alignment of top eigenvalues and the projections of true labels on corresponding eigenvectors.

gives incredibly high score almost immediately, as shown by the train loss (Figure 1(a)) and by the accuracy on the test data (the table in Figure 2(c)) thus we will focus our analysis on the CIFAR-10 dataset. Similar to the setup in (Arora et al., 2019), in Figure 1(c), for different methods, we plot the eigenvalues of $\mathbf{H}(\mathcal{K})^\infty$ and the projections of the true class labels on the eigenvectors (i.e., the projections $\{(\mathbf{v}_i^T \mathbf{y})^2\}_{i=0}^{n-1}$). For better visualization, we plot the cumulative forms $\sum_{j=0}^{i}(\mathbf{v}_j^T \mathbf{y})^2$'s which are normalized such that $\sum_{i=0}^{n-1}(\mathbf{v}_i^T \mathbf{y})^2 = 1$. The results show that using kernels yield a better alignment of the projections with the top eigenvalues, leading to faster convergences. In other words, with kernels, we attain larger $(\mathbf{v}_i^T \mathbf{y})^2$'s for top eigenvalues.
Generalization. We next investigate the generalization performance of the Gaussian kernel method by analyzing the values of equations (5) and (6). Table 1 shows this quantity for different settings and kernels respectively on MNIST and CIFAR-10 datasets. We observe that in both datasets with several kernels we obtain a lower theoretical upper bound on the generalization error. It is clear that the bound improves as the dimension of the representations increases but also that the generalization bound seems quite sensitive to values of γ.

In addition to the theoretical upper bound, we measure the test error for the studied datasets. Figures 2(a) and 2(b) show respectively the test error and the test accuracy at different steps of the optimization by Gradient Descent for CIFAR-10. We observe that the kernel methods yield significant improvements of both the test error and the accuracy on the test dataset. We observe that the larger the kernel, the larger the improvement. Additionally, we can see a sharper reduction in test error compared to the no-kernel case. This sharp transition (after a small number of steps) is particularly interesting. Because, along such a transition, we observe a significant improvement in the accuracy on test dataset. Thus early-stopping that is commonly used in deep learning

Table 1. Quantification of $\mathbf{y}^T(\mathbf{H}^\infty)^{-1}\mathbf{y}$ (or $\mathbf{y}^T(\mathbf{H}(\mathcal{K})^\infty)^{-1}\mathbf{y}$ for kernels) for different experimental settings. For both datasets, most kernels yield smaller upper bounds on generalization error.

γ	Dimension	MNIST	CIFAR-10
0.1	∞	1519	71066
1	∞	442	16680
10	∞	9841	1236
1	500	790	55501
1	1000	679	51487
1	3072	534	47471
1	10000	478	44630
No kernel		789	74670

can be even more efficient when using kernel methods. Finally, similar to the no-kernel case in (Arora et al., 2019), by comparing the plots in Figures 1(b), 1(c) and 2(a) we find tight connections between, i) (training) optimization, ii) projection on the top eigenvalues, and iii) generalization. We can therefore improve both training and generalization with kernels since we can get better alignment of the eigenvectors belonging the largest eigenvalues and the target labels.

3.2. Neural embedding

Choosing a proper kernel and its parameters can be challenging (von Luxburg, 2007), as also seen in Table 1. Thus, we investigate a data-dependent neural kernel and embedding. For this purpose, we add a second hidden layer to the neural network with $m = 10000$ hidden units and ReLU activation. We pre-train this embedding using two different approaches. The first layer is then kept fix as an embedding where the rest of the network is reinitialized and trained. The first approach is to split the training data in half. We use the first subset to pre-train this three-layer network and the

(a) Test error on CIFAR-10 data.　　(b) Test accuracy on CIFAR-10 data.　　(c) Test accuracy on MNIST data.
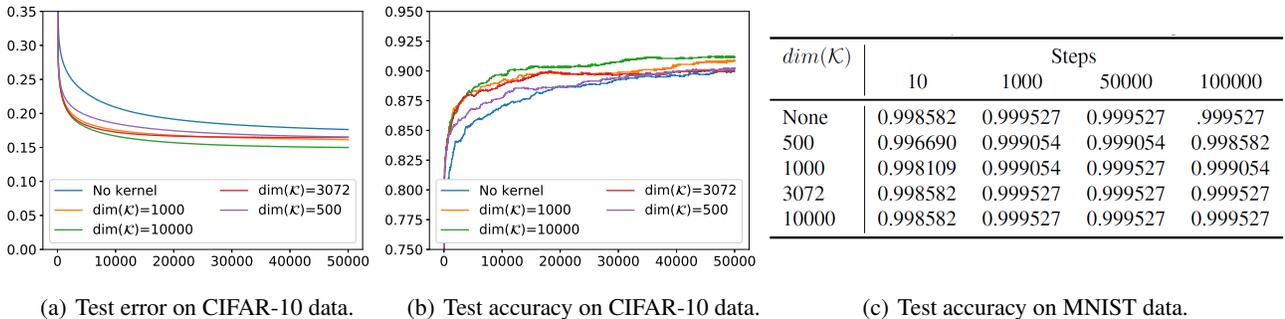
Figure 2. Experimental test errors and accuracy on the test set at the different steps of the Gradient Descent optimization algorithm for CIFAR-10 dataset. For MNIST, we report the accuracy at the different steps of the Gradient Descent optimization where performance is very good with different steps and parameters.

$dim(\mathcal{K})$	Steps			
	10	1000	50000	100000
None	0.998582	0.999527	0.999527	.999527
500	0.996690	0.999054	0.999054	0.998582
1000	0.998109	0.999054	0.999527	0.999054
3072	0.998582	0.999527	0.999527	0.999527
10000	0.998582	0.999527	0.999527	0.999527

(a) Training loss.　　　　　　(b) Test loss　　　　　　(c) Projections
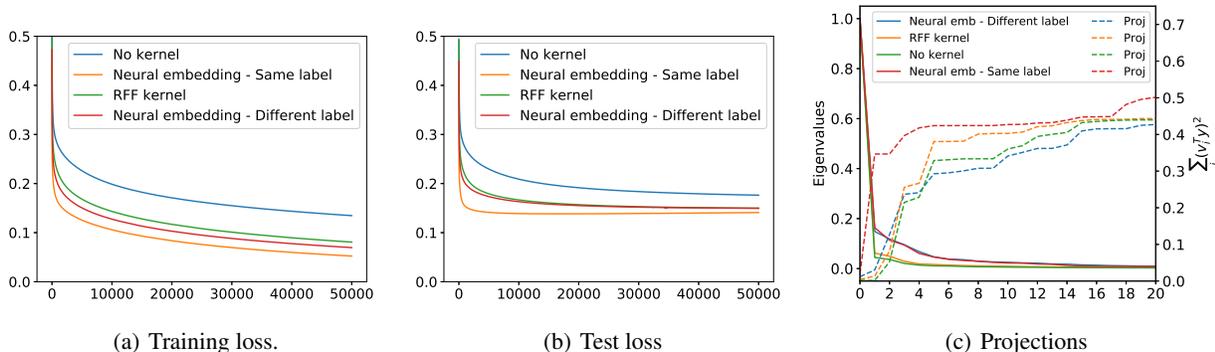
Figure 3. Experimental train and test errors at the different steps of Gradient Descent as well as eigenvector projections for the CIFAR-10 dataset. For the model pre-trained with the same labels, the training loss and projections are calculated based on the unseen subset of training data. We observe that neural embeddings improve the convergence, generalization and the alignment of eigenvector projections.

second subset to use for our optimization experiments. In this approach we double η to keep the step length the same. The other approach is to use data from a different domain for pre-training. For instance, we use the last two classes of the CIFAR-10 dataset for pre-training the embedding. We compare our results with not using any kernel and with using a RFF kernel with embedding of size 10000.

Optimization. Figure 3(a) shows the training loss for the CIFAR-10 dataset. We observe the neural embeddings achieve faster convergence compared to the previous methods. We report the training loss for neural embedding (same label) on the second (unused) subset of the data, whereas in the other cases we report the results on the full training data. If we use only the second subset for the other methods, we observe very consistent results to Figure 3. Figure 3(c) demonstrates the top eigenvalues as well as their eigenvector projections on the target labels. This shows that both variants of neural embeddings improve alignment of the labels to eigenvectors corresponding to larger eigenvalues (compared to the best RFF kernel). While the effect is un-

surprisingly larger when pre-training on the same labels, it is still significantly better when pre-trained on other labels. **Generalization.** In Figure 3(b) we report the test error on the CIFAR-10. This shows that the neural embeddings perform at least comparable with the best studied RFF kernel. If the pre-training is done on the same labels we obtain a clear improvement, even if the actual training is only done on a dataset with half the size.

4. Conclusions

We extended the recent results of (Arora et al., 2019) by a spectral analysis of the representations corresponding to kernel and neural embeddings and showed that such representations benefit both optimization and generalization. By combining recent results connecting kernel embeddings to neural networks such as (Tsuchida et al., 2018; Jacot et al., 2018), one may be able to extend the fine–grained theoretical results of (Arora et al., 2019) for two layer networks to deeper networks.

References

Arora, S., Du, S. S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *CoRR*, abs/1901.08584, 2019.

Belkin, M., Ma, S., and Mandal, S. To understand deep learning we need to understand kernel learning, 2018.

Cho, Y. and Saul, L. K. Kernel methods for deep learning. In Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems 22*, pp. 342–350. Curran Associates, Inc., 2009.

Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J. S. On kernel-target alignment. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pp. 367–373, 2001.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8571–8580. Curran Associates, Inc., 2018.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pp. 1177–1184, 2007.

Tsuchida, R., Roosta-Khorasani, F., and Gallagher, M. Invariance of weight distributions in rectified mlps. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pp. 5002–5011, 2018.

von Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.