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, e.g., 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:

\[ f_{W,a}(x) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \max(0, w_r^T x), \]  

with \( x \in \mathbb{R}^d \), \( w_1, .. w_m \in \mathbb{R}^{d \times m} \) and \( (a_1, .. a_m)^T \in \mathbb{R}^m \). These can be written jointly as \( a = (a_1, .. a_m)^T \) and \( W = (w_1, .. w_m) \). 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:

\[ H_{ij}^\infty := E_{x \sim \mathcal{N}(0, \mathcal{I})} [x_i^T x_j | w^T x_i \geq 0, w^T x_j \geq 0] \]

If \( H^\infty = \sum \lambda_i v_i v_i^T \) is the orthonormal decomposition of \( H^\infty \), (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 eigenvalues of \( H^\infty \).

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 \( \phi \) at the input layer corresponding to a kernel \( K \). The corresponding model is:

\[ f_{W,a}(x) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} a_r \max(0, w_r^T \phi(x_i)). \]  

For a representation \( (\phi(x_i), i \in [n]) \) corresponding to a kernel \( K \), define the Gram Matrix

\[ H(K)^\infty_{ij} := E_W [K(x_i, x_j)| w^T \phi(x_i) \geq 0, w^T \phi(x_j) \geq 0] \]

and let its eigenvalues be ordered as \( \lambda_0(K) \geq \lambda_1(K) \geq \cdots \geq \lambda_{n-1}(K) \) and let \( v_0(K), \cdots, v_{n-1}(K) \) be the corresponding eigenvectors.

A kernel \( K \) 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

\[ \sqrt{\sum_i (1 - \eta \lambda_i)^{2k}(v_i^T y)^2} \]  

For our kernelized network, the corresponding convergence is controlled by

\[ \sqrt{\sum_i (1 - \eta \lambda_i(K))^{2k}(v(K)^T y)^2} \]  

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

\[ y^T (H^\infty)^{-1} y \]

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

\[ y^T (H(K)^\infty)^{-1} y \]

3. Experiments

We perform our experiments on two commonly-used datasets for validating deep neural models, i.e., MNIST and CIFAR-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 \( ||x_i||_2 = 1 \). This is also done for kernel embeddings such that \( ||\phi(x_i)||_2 = 1 \).

The weights in equation (2) are initialized as follows:

\[ w_i \sim \mathcal{N}(0, k^2 \mathcal{I}), \ a_i \sim \text{Unif}(\{-1, 1\}), \forall r \in m. \]

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

\[ \Phi(W, a) = 1/2 \sum_{i=1}^{n} (y_i - f_{W,a}(x))^2 \]

For optimization, we use (full batch) gradient descent with the learning rate \( \eta \). 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 \( K(x_i, x_j) := \exp(-\gamma \|x_i - x_j\|^2) \). We perform our experiments on two commonly-used datasets for validating deep neural models, i.e., MNIST and CIFAR-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 \( ||x_i||_2 = 1 \). This is also done for kernel embeddings such that \( ||\phi(x_i)||_2 = 1 \).

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

\[ y^T (H^\infty)^{-1} y \]

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

\[ y^T (H(K)^\infty)^{-1} y \]
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 $H(\mathcal{K})^\infty$ and the projections of the true class labels on the eigenvectors (i.e., the projections $\{v^T_i y\}^n_{i=0}$). For better visualization, we plot the cumulative forms $\sum_{j=0}^\infty (v^T_j y)^2$’s which are normalized such that $\sum_{i=0}^{n-1} (v^T_i y)^2 = 1$. 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 $(v^T_i y)^2$’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 $\gamma$.

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 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
We compare our results with not using any kernel and with 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 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


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


