Mehler’s Formula, Branching Processes, and Compositional Kernels of Deep Neural Networks

Tengyuan Liang
Hai Tran-Bach

May 26, 2020
Motivations & Questions

DNN and Kernels (Rahimi, Recht ’08; Belkin et al ’18; Jacot et al ’19). What role do the activation functions play in the connections between DNN and Kernels?
Motivations & Questions

▶ DNN and Kernels (*Rahimi, Recht ’08; Belkin et al ’18; Jacot et al ’19*). What role do the activation functions play in the connections between DNN and Kernels?

▶ Interpolation and Generalization (*Zhang et al ’17; Belkin et al ’18; Liang, Rakhlin ’18*). How does the activation function interplay with depth, sample size, and input dimensionality in terms of memorization capacity?
Motivations & Questions

▶ DNN and Kernels (*Rahimi, Recht ’08; Belkin et al ’18; Jacot et al ’19*). **What role do the activation functions play in the connections between DNN and Kernels?**

▶ Interpolation and Generalization (*Zhang et al ’17; Belkin et al ’18; Liang, Rakhlin ’18*). **How does the activation function interplay with depth, sample size, and input dimensionality in terms of memorization capacity?**

▶ Is there hope to design activation functions such that we can ”compress” multiple layers?
Multi-Layer Perceptron with Random Weights

\textit{(Neal '96; Rahimi, Recht '08; Daniely et al '07)}

\[
\text{Input : } x^{(0)} := x \in \mathbb{R}^d
\]

\[
\text{Hidden Layers : } x^{(\ell+1)} := \sigma \left( \frac{W^{(\ell)} x^{(\ell)}}{\|x^{(\ell)}\|} \right) \in \mathbb{R}^{d_{\ell+1}} , \text{ for } 0 \leq \ell < L
\]

\[
\text{Random Weights : } W^{(\ell)} \in \mathbb{R}^{d_{\ell+1} \times d_{\ell}} , \quad W^{(\ell)} \sim \mathcal{MN}(0, I_{d_{\ell+1}} \otimes I_d)
\]

\[
\text{Regime : } d_1, \ldots, d_L \to \infty
\]
Duality: Activation and Kernel

Activation:

\[ \sigma(x) = \sum_{k=0}^{\infty} \alpha_k h_k(x), \]

with \( \sum_{k=0}^{\infty} \alpha_k^2 = 1. \)
Duality: Activation and Kernel

Activation:

\[ \sigma(x) = \sum_{k=0}^{\infty} \alpha_k h_k(x), \]

with \( \sum_{k=0}^{\infty} \alpha_k^2 = 1 \).

Dual Kernel:

\[ K(x_i, x_j) := \mathbb{E}_{w \sim \mathcal{N}(0, I_d)} \left[ \sigma(w^T x_i / \|x_i\|) \sigma(w^T x_j / \|x_j\|) \right] \]

\[ = \sum_{k=0}^{\infty} \alpha_k^2 \rho_{ij}^k =: G(\rho_{ij}); \quad \rho_{ij} := \langle x_i / \|x_i\|, x_j / \|x_j\| \rangle. \]
Duality: Activation and Kernel

Activation:

\[ \sigma(x) = \sum_{k=0}^{\infty} \alpha_k h_k(x), \]

with \( \sum_{k=0}^{\infty} \alpha_k^2 = 1. \)

Dual Kernel:

\[ K(x_i, x_j) := \mathbb{E}_{w \sim \mathcal{N}(0, I_d)} \left[ \sigma(w^T x_i/\|x_i\|) \sigma(w^T x_j/\|x_j\|) \right] = \sum_{k=0}^{\infty} \alpha_k^2 \rho_{ij}^k =: G(\rho_{ij}); \quad \rho_{ij} := \langle x_i/\|x_i\|, x_j/\|x_j\| \rangle. \]

Compositional Kernel:

\[ K^{(L)}(x_i, x_j) = \underbrace{G \circ G \circ \cdots \circ G}_{\text{composite } L \text{ times}}(\rho_{ij}) =: G^{(L)}(\rho_{ij}). \]
Branching Process and Compositional Kernels

Distribution: \( Y \), with \( P(Y = k) = \alpha_k^2 \) and PGF \( G \).

Galton-Watson Process: \( Z^{(L)} \), with off-spring \( Y \) and PGFs \( G^{(L)} \).
Rescaled Limit: Phase Transition

Theorem (Liang, Tran-Bach '20)

Define

$$\mu := \sum_{k \geq 0} a_k^2 k, \quad \mu^* := \sum_{k > 2} a_k^2 k \log k.$$  

Then, for all $t > 0$, we have

1. If $\mu \leq 1$,

$$\lim_{L \to \infty} K^{(L)}(e^{-t}) = \begin{cases} 1, & \text{if } a_1 \neq 1 \\ e^{-t}, & \text{if } a_1 = 1 \end{cases}$$

2. If $\mu > 1$,

$$\lim_{L \to \infty} K^{(L)}(e^{-t/\mu^L}) = \begin{cases} \xi + (1 - \xi) E[e^{-tW}], & \text{if } \mu^* < \infty \\ 0, & \text{if } \mu^* = \infty \end{cases}$$
Kernel Limits Example: centered ReLU

Unscaled Limit: $K^{(L)}(t)$
Kernel Limits Example: centered ReLU

Unscaled Limit: \( K^{(L)}(t) \)
Kernel Limits Example: centered ReLU

Unscaled Limit: $K^{(L)}(t)$

![Graph of ReLU with curves for different L values]

Rescaled Limit: $K^{(L)}(e^{-t/\mu^L})$

![Graph of Rescaled ReLU with curves for different L values]
Memorization Capacity

- "small correlation" \( \sup_{ij} |\rho_{ij}| \approx 0 \)
  - \( x_1, \ldots, x_n \overset{iid}{\sim} \text{Unif}(\mathbb{S}^{d-1}) \) and \( \log(n)/d \to 0 \)
- "large correlation" \( \sup_{ij} |\rho_{ij}| \approx 1 \)
  - \( x_1, \ldots, x_n \) maximal packing of \( \mathbb{S}^{d-1} \) and \( \log(n)/d \to \infty \)
Memorization Capacity

- **"small correlation"** \( \sup_{ij} |\rho_{ij}| \approx 0 \)
  - \( x_1, \ldots, x_n \overset{iid}{\sim} \text{Unif}(\mathbb{S}^{d-1}) \) and \( \log(n)/d \to 0 \)

- **"large correlation"** \( \sup_{ij} |\rho_{ij}| \approx 1 \)
  - \( x_1, \ldots, x_n \) maximal packing of \( \mathbb{S}^{d-1} \) and \( \log(n)/d \to \infty \)
Memorization Capacity Theorem

Theorem (Liang & Tran-Bach '20)

\[
L \gtrsim \begin{cases} 
\log(n\kappa^{-1}) + \log \frac{\log n}{d} & \quad \text{(small correlation)} \\
\exp(2\frac{\log n}{d})\log(n\kappa^{-1}) & \quad \text{(large correlation)}
\end{cases}
\]

\[
L \gtrsim \frac{\log(n\kappa^{-1}) + \log \frac{\log n}{d}}{\log a_1^{-2}}
\]

\[
L \gtrsim \frac{\exp(2\frac{\log n}{d})\log(n\kappa^{-1})}{\mu - 1}
\]

to memorize the data in the sense that \(1 - \kappa \leq \lambda_i \leq 1 + \kappa\), where \(\lambda_i\) are the eigenvalues of \(K := \{K(x_i, x_j)\}_{ij}\).
Memorization Capacity Theorem

Theorem (Liang & Tran-Bach ’20)

\[ L \gtrsim \frac{\log(n\kappa^{-1}) + \log \frac{\log n}{d}}{\log a_1^{-2}} \quad \text{(small correlation)} \]

\[ L \gtrsim \frac{\exp(2 \log \frac{n}{d}) \log(n\kappa^{-1})}{\mu - 1} \quad \text{(large correlation)} \]

to memorize the data in the sense that \(1 - \kappa \leq \lambda_i \leq 1 + \kappa\), where \(\lambda_i\) are the eigenvalues of \(K := \{K(x_i, x_j)\}_{ij}\).
Memorization Capacity Theorem

Theorem (Liang & Tran-Bach ’20)

$L \gtrsim \frac{\log(n\kappa^{-1}) + \log \frac{\log n}{d}}{\log a_1^{-2}}$ (small correlation)

$L \gtrsim \frac{\exp(2\frac{\log n}{d})\log(n\kappa^{-1})}{\mu - 1}$ (large correlation)

to memorize the data in the sense that $1 - \kappa \leq \lambda_i \leq 1 + \kappa$, where $\lambda_i$ are the eigenvalues of $K := \{K(x_i, x_j)\}_{ij}$. 
Memorization Capacity Theorem

**Theorem (Liang & Tran-Bach '20)**

\[
L \gtrsim \frac{\log(n\kappa^{-1}) + \log(n)}{\log a_1^{-2}} \quad \text{(small correlation)}
\]

\[
L \gtrsim \frac{\exp(2\log(n/d))\log(n\kappa^{-1})}{\mu - 1} \quad \text{(large correlation)}
\]

To memorize the data in the sense that \(1 - \kappa \leq \lambda_i \leq 1 + \kappa\), where \(\lambda_i\) are the eigenvalues of \(K := \{K(x_i, x_j)\}_{ij}\).
New Random Features Algorithm

- Activations $\sigma(\cdot)$
- Taylor Coefficients
- HERMITE Coefficients
- PGFs $G^{(L)}(\cdot)$
- Branching Processes
- Kernels $K(\cdot,\cdot)$
New Random Features Algorithm

![Diagram](image_url)

<table>
<thead>
<tr>
<th>Kernels</th>
<th>Activation</th>
<th>Sampling</th>
</tr>
</thead>
<tbody>
<tr>
<td>shift-invariant (Rahimi, Recht '08)</td>
<td>cos, sin</td>
<td>≈</td>
</tr>
<tr>
<td>inner-product (Liang, Tran-Bach '20)</td>
<td>≈</td>
<td>Gaussian</td>
</tr>
</tbody>
</table>

H. Tran-Bach

Compositional Kernels of Deep Neural Networks
### Experiment: MNIST & CIFAR10

<table>
<thead>
<tr>
<th>Activation</th>
<th>ReLU</th>
<th>GeLU</th>
<th>Sigmoid</th>
<th>Swish</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$</td>
<td>0.95</td>
<td>1.08</td>
<td>0.15</td>
<td>1.07</td>
</tr>
<tr>
<td>$a_1^2$</td>
<td>0.50</td>
<td>0.59</td>
<td>0.15</td>
<td>0.80</td>
</tr>
</tbody>
</table>
**Experiment: MNIST & CIFAR10**

<table>
<thead>
<tr>
<th>Activation</th>
<th>ReLU</th>
<th>GeLU</th>
<th>Sigmoid</th>
<th>Swish</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$</td>
<td>0.95</td>
<td>1.08</td>
<td>0.15</td>
<td>1.07</td>
</tr>
<tr>
<td>$a_1^2$</td>
<td>0.50</td>
<td>0.59</td>
<td>0.15</td>
<td>0.80</td>
</tr>
</tbody>
</table>

**MNIST: L=1**

**MNIST: L=2**

**MNIST: L=3**

**CIFAR10: L=1**

**CIFAR10: L=2**

**CIFAR10: L=3**
Conclusions

1. Additional Results:
Conclusions

1. Additional Results:
   - Eigenvalues of Compositional Kernels: Generalization Error
Conclusions

1. Additional Results:
   - Eigenvalues of Compositional Kernels: Generalization Error
   - Numerical Tricks with Guarantees: Stability under truncation
Conclusions

1. **Additional Results:**
   - Eigenvalues of Compositional Kernels: Generalization Error
   - Numerical Tricks with Guarantees: Stability under truncation

2. **Summary** of role of activation functions in DNN:
Conclusions

1. **Additional Results:**
   - Eigenvalues of Compositional Kernels: Generalization Error
   - Numerical Tricks with Guarantees: Stability under truncation

2. **Summary** of role of activation functions in DNN:
   - Composition of kernels as a Branching Process

Conclusions

1. **Additional Results:**
   - Eigenvalues of Compositional Kernels: Generalization Error
   - Numerical Tricks with Guarantees: Stability under truncation

2. **Summary** of role of activation functions in DNN:
   - Composition of kernels as a Branching Process
   - Functionals of activations govern kernel limits

Conclusions

1. Additional Results:
   - Eigenvalues of Compositional Kernels: Generalization Error
   - Numerical Tricks with Guarantees: Stability under truncation

2. Summary of role of activation functions in DNN:
   - Composition of kernels as a Branching Process
   - Functionals of activations govern kernel limits
   - Depth bounds depending on activations for memorization
Conclusions

1. **Additional Results:**
   - Eigenvalues of Compositional Kernels: Generalization Error
   - Numerical Tricks with Guarantees: Stability under truncation

2. **Summary** of role of activation functions in DNN:
   - Composition of kernels as a Branching Process
   - Functionals of activations govern kernel limits
   - Depth bounds depending on activations for memorization
   - New Random Features Algorithm

Conclusions

1. Additional Results:
   - Eigenvalues of Compositional Kernels: Generalization Error
   - Numerical Tricks with Guarantees: Stability under truncation

2. Summary of role of activation functions in DNN:
   - Composition of kernels as a Branching Process
   - Functionals of activations govern kernel limits
   - Depth bounds depending on activations for memorization
   - New Random Features Algorithm