Neural Tangent Kernel
Convergence and Generalization of DNNs

Arthur Jacot, Franck Gabriel, Clément Hongler

Ecole Polytechnique Fédérale de Lausanne

October 28, 2019
Neural Networks

- $L + 1$ layers of $n_\ell$ neurons with activations $\alpha^{(\ell)}(x) \in \mathbb{R}^{n_\ell}$

\[
\begin{align*}
\alpha^{(0)}(x) &= x \\
\tilde{\alpha}^{(\ell+1)}(x) &= \frac{\sqrt{1 - \beta^2}}{\sqrt{n_\ell}} W^{(\ell)} \alpha^{(\ell)}(x) + \beta b^{(\ell)} \\
\alpha^{(\ell+1)}(x) &= \sigma\left(\tilde{\alpha}^{(\ell+1)}(x)\right)
\end{align*}
\]

- Parameters: connections weights $W^{(\ell)} \in \mathbb{R}^{n_\ell \times n_{\ell+1}}$ and bias $b^{(\ell)} \in \mathbb{R}^{n_{\ell+1}}$.
- Weights / bias balance: $\beta \in [0, 1]$.
- Non-linearity: $\sigma : \mathbb{R} \rightarrow \mathbb{R}$.

- Network function $f_\theta(x) = \tilde{\alpha}^{(L)}(x)$
Initialization: DNNs as Gaussian processes

- In the infinite width limit \( n_1, \ldots, n_{L-1} \to \infty \)
- Initialize the parameters \( \theta \sim \mathcal{N}(0, I d_P) \).
- The preactivations \( \tilde{\alpha}_i^{(\ell)}(\cdot; \theta) : \mathbb{R}^{n_0} \to \mathbb{R} \) converge to iid Gaussian processes of covariance \( \Sigma^{(\ell)} \) (Cho and Saul, 2009; Lee et al., 2018):
  \[
  \Sigma^{(1)}(x, y) = (1 - \beta^2)x^T y + \beta^2 \\
  \Sigma^{(\ell+1)}(x, y) = (1 - \beta^2)\mathbb{E}_{\alpha \sim \mathcal{N}(0, \Sigma^{(\ell)})} [\sigma(\alpha(x))\sigma(\alpha(y))] + \beta^2
  \]
- The network function \( f_\theta = \tilde{\alpha}^{(L)} \) is also asymptotically Gaussian.
Training: Neural Tangent Kernel

- Training set of size $N$: inputs $X \in \mathbb{R}^{N \times n_0}$ and outputs $Y \in \mathbb{R}^{N \times n_L}$.
- Gradient descent on the MSE $C(\theta) = \frac{1}{N} \| f_\theta(X) - Y \|^2$

\[
\partial_t \theta = -\nabla C(\theta) = \frac{2}{N} \sum_{i=1}^{N} \nabla f_\theta(x_i) (Y_i - f_\theta(x_i))
\]

- Evolution of $f_\theta$:

\[
\partial_t f_\theta(x) = (\nabla f_\theta(x))^T \partial_t \theta = \frac{2}{N} \sum_{i=1}^{N} (\nabla f_\theta(x))^T \nabla f_\theta(x_i) (Y_i - f_\theta(x_i))
\]

- Neural Tangent Kernel (NTK):

\[
\Theta^{(L)}(x, y) := (\nabla f_\theta(x))^T \nabla f_\theta(y)
\]
Asymptotics of the NTK

Theorem (Arora et al., 2019; Jacot et al., 2018)

Let \( n_1, \ldots, n_{L-1} \to \infty \), for any \( t \):

\[
\Theta^{(L)}(t) \to \Theta^{(L)}
\]

where

\[
\Theta^{(L)}(x, y) = \sum_{\ell=1}^{L} \Sigma^{(\ell)}(x, y) \dot{\Sigma}^{(\ell+1)}(x, y) \cdots \dot{\Sigma}^{(L)}(x, y)
\]

with

\[
\dot{\Sigma}^{(L)}(x, x') = (1 - \beta^2) \mathbb{E}_{\alpha \sim \mathcal{N}(0, \Sigma^{(L-1)})} [\dot{\sigma}(\alpha(x))\dot{\sigma}(\alpha(x'))].
\]
**Convergence**

The continuous-time dynamics of the outputs $Y_{\theta(t)} = f_{\theta(t)}(X)$ are described by the linear ODE

$$\partial_t Y_{\theta(t),k} = \frac{2}{N} \Theta^{(L)}(X, X) \left( Y - Y_{\theta(t)} \right)$$

with solution

$$Y_{\theta(t)} = Y - e^{\frac{2t}{N} \Theta^{(L)}(X, X)} \left( Y - Y_{\theta(0)} \right).$$

Convergence and loss surface (Jacot et al., 2019a):

1. **Eigenvalues:** speed of convergence along the eigenvector
2. **Narrow valley structure:**
   1. Large eig. are the 'cliffs'
   2. Small eig. are the 'bottom'
3. **Condition number** $\kappa = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}}$

![Narrow valley in the loss surface.](image)
Freeze and Chaos

Two regimes appear (as in Poole et al. (2016); Schoenholz et al. (2017))

Theorem (Jacot et al., 2019b)

Consider a twice differentiable $\sigma$ satisfying $E_{z \sim \mathcal{N}(0,1)}[\sigma(z)] = 1$ and two inputs $x, y$ with $\|x\| = \|y\| = \sqrt{n_0}$. The characteristic value

$$r_{\sigma, \beta} = (1 - \beta^2)E_{z \sim \mathcal{N}(0,1)}[\sigma'(z)^2]$$

determines two regimes:

FREEZE $(r < 1)$: For $x, y \in \mathbb{S}_{n_0}$

$$1 \geq \frac{\Theta^{(L)}(x, y)}{\Theta^{(L)}(x, x)} \geq 1 - CLr^L$$

CHAOS $(r > 1)$: If $x \neq \pm y$:

$$\frac{\Theta^{(L)}(x, y)}{\Theta^{(L)}(x, x)} \to 0$$
Freeze and Chaos: Properties

- **FREEZE (r<1):**
  - Almost constant NTK Gram Matrix
  - Bad conditioning $\lambda_{\text{max}}/\lambda_{\text{min}}$
  - Slow convergence

- **CHAOS (r>1):**
  - Almost identity NTK Gram Matrix
  - Fast convergence
  - Generalization?

- **EDGE:** For the RELU: $r = 1 - \beta^2 \approx 1$
  - Converges to $\text{Id} + c$
  - Strong constant mode for large $N$
  - Slow convergence

**Figure:** The NTK in the different regimes for $L = 6$. 
Chaos: normalization

- Normalize the non-linearity (over $z \sim \mathcal{N}(0, 1)$):

$$\bar{\sigma}(x) = \frac{\sigma(x) - \mathbb{E}[\sigma(z)]}{\sqrt{\text{Var}[\sigma(z)]}}.$$

- If $\sigma \neq id$ we have $\mathbb{E}[\dot{\bar{\sigma}}(z)^2] > 1$, and for small enough $\beta > 0$

$$r = (1 - \beta^2)\mathbb{E}[\dot{\bar{\sigma}}(z)^2] > 1.$$

- Asymptotically equivalent to Layer Normalization.
Chaos: normalization

- Normalize the non-linearity (over $z \sim \mathcal{N}(0, 1)$):
  $$\bar{\sigma}(x) = \frac{\sigma(x) - \mathbb{E}[\sigma(z)]}{\sqrt{\text{Var}[\sigma(z)]}}.$$  

- If $\sigma \neq \text{id}$ we have $\mathbb{E}[\dot{\bar{\sigma}}(z)^2] > 1$, and for small enough $\beta > 0$
  $$r = (1 - \beta^2)\mathbb{E}[\dot{\bar{\sigma}}(z)^2] > 1.$$  

- Asymptotically equivalent to Layer Normalization.

- For Batch Normalization we only have:
  - Applying BN after the last non-linearity controls the constant mode:
    $$\frac{1}{N^2} \sum_{ij} \Theta^{(L)}(x_i, x_j) = \beta^2$$  

Generative Adversarial Networks

- Two networks (Goodfellow et al., 2014)
  - Generator \( G \): generates data \( G(z) \) from a random code \( z \).
  - Discriminator \( D \): classifies real/generated data to guide the generator.

- If the generator lies in the **FREEZE**
  - **Mode collapse**: The generator \( G \) becomes constant.

- Solutions:
  - Batch Normalization
  - Chaotic generator
Deconvolutional Generator

For \( G \) a deconvolutional network with stride \( s \)

**Theorem**

*In the FREEZE (\( r > 1 \)):*

\[
\frac{1 - r^{v+1}}{1 - r^L} - C_1(v + 1)r^v \leq \frac{\Theta_p^{(L)}(x, y)}{\Theta_{p', p}^{(L)}(x, x)} \leq \frac{1 - r^{v+1}}{1 - r^L}
\]

for \( v \) the max. in \( \{0, L - 1\} \) s.t. \( s^v \) divides \( p - p' \)

- **Checkerboard patterns:** images which repeat every \( s^v \) pixels
- **Solution:** Batch Norm / Chaos
NTK PCA

FREEZE

CHAOS

BATCH NORM
Conclusion

1. Certain architectures lead to dominating eigenvalues:
   1.1 Constant mode
   1.2 Checkerboard patterns
2. Slows down training
3. Mode collapse in GANS
4. Use the NTK to identify and fix them
Bibliography I


