On the Random Conjugate Kernel and Neural Tangent Kernel

Zhengmian Hu [1], Heng Huang [1,2]

[1]Department of Electrical and Computer Engineering, University of Pittsburgh, Pittsburgh, PA 15213, USA.

[2]JD Finance America Corporation, Mountain View, CA 94043, USA.

Definition

A neural network: $N(x) = W\ h(x)$. W is the last fully connected layer weight, h is the last hidden layer.

Conjugate Kernel (CK):

$$\Sigma(x, y) = h(x)^\top h(y)$$

Definition

A neural network: $N(x) = W\ h(x)$. W is the last fully connected layer weight, h is the last hidden layer.

Conjugate Kernel (CK):

$$\Sigma(x, y) = h(x)^\top h(y)$$

Only train last layer, fix previous layers.
Use all previous layers as random features.
Gradient descent on input x for one step with step size t:

$$\Delta W = -t \frac{\partial L}{\partial N(x)} h(x)^\top$$

$$\Delta N(y) = -t \frac{\partial L}{\partial N(x)} h(x)^\top h(y)$$

Definition

Neural network has parameters W_i and b_i as weights and biases.

Neural Tangent Kernel (NTK):

$$K_{W_i}(x,y) = \sum_{j,l} \frac{\partial N(x)}{\partial [W_i]_{j,l}} \frac{\partial N(y)}{\partial [W_i]_{j,l}}$$

$$K_{b_i}(x,y) = \sum_{j} \frac{\partial N(x)}{\partial [b_i]_{j}} \frac{\partial N(y)}{\partial [b_i]_{j}}$$

$$K(x,y) = \sum_{i} (K_{W_i}(x,y) + K_{b_i}(x,y))$$

Definition

Neural network has parameters W_i and b_i as weights and biases.

Neural Tangent Kernel (NTK):

$$K_{W_i}(x, y) = \sum_{j,l} \frac{\partial N(x)}{\partial [W_i]_{j,l}} \frac{\partial N(y)}{\partial [W_i]_{j,l}}$$

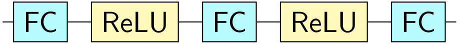
$$K_{b_i}(x, y) = \sum_{j} \frac{\partial N(x)}{\partial [b_i]_{j}} \frac{\partial N(y)}{\partial [b_i]_{j}}$$

$$K(x, y) = \sum_i (K_{W_i}(x, y) + K_{b_i}(x, y))$$

Train all layers for a small step t on input x:

$$\Delta N(y) \approx -t \frac{\partial L}{\partial N(x)} K(x, y)$$

Two types of neural network

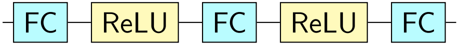
Feedforward network:


d - Depth

n - Hidden layer width

Two types of neural network

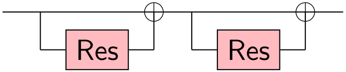
Feedforward network:



d - Depth

n - Hidden layer width

Residual network:



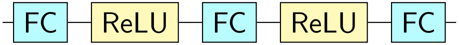
m - Number of branches

d - Sum depth of each branch

n - Hidden layer width

Two types of neural network

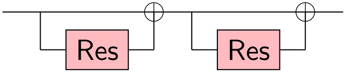
Feedforward network:



d - Depth

n - Hidden layer width

Residual network:



m - Number of branches

d - Sum depth of each branch

n - Hidden layer width

Both has Gaussian initialization for weights and set initial bias to 0.

Previous Results

Training a neural network is approximately kernel gradient descent. However, the kernel is not generally a constant.

When $n \to \infty$, d is fixed. CK and NTK converge to fixed values. (Cho, Y. & Saul, 2009, Jacot et al., 2018)

For finite width and depth case, the second order moments of random CK and NTK is bounded for both feedforward network (Hanin & Nica 2020) and residual network (Littwin et al. 2020).

CK of feedforward network converges to log normal distribution (Hanin & Nica 2019).

Our results

We study the diagonal elements of CK and NTK for both feedforward network and residual network with Relu activation:

- Derive every order moments of CK and NTK.
- Show that random CK and NTK converge to log normal distribution under certain constraints.

Moments

Feedforward Network CK:

$$\mathrm{E}[\Sigma(x_0, x_0)^r] = \|x_0\|^{2r} c^r \left(\exp\left(\binom{r}{2} \beta \right) + \mathcal{O}\left(\sum_{i=1}^{d-1} \frac{1}{n_i^2} \right) \right)$$

c depends on variance of Gaussian initialization.

Noise parameter:

$$\beta = \sum_{k=1}^{d-1} \frac{5}{n_i}$$

Moments

Feedforward Network CK:

$$\mathrm{E}[\Sigma(x_0, x_0)^r] = \|x_0\|^{2r} c^r \left(\exp\left(\binom{r}{2} \beta \right) + \mathcal{O}\left(\sum_{i=1}^{d-1} \frac{1}{n_i^2} \right) \right)$$

c depends on variance of Gaussian initialization.
Noise parameter:

$$\beta = \sum_{k=1}^{d-1} \frac{5}{n_i}$$

Feedforward Network NTK:

$$\frac{\mathrm{E}[K(x_0, x_0)^r]}{(\mathrm{E}[K(x_0, x_0)])^r} \leq \exp\left(\binom{r}{2} \beta \right) + \mathcal{O}\left(\sum_{i=1}^{d-1} \frac{1}{n_i^2} \right)$$

Distribution Convergence

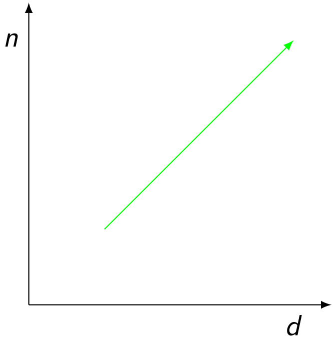


Figure: Different limiting behaviour of CK and NTK.

For feedforward network:

Green line: Diagonal elements of CK and NTK of each parameter converge to log-normal distribution.

Moments

Residual Network CK:

$$E[\Sigma(x_0, x_0)^r] = \exp\left(\left(r + \frac{4}{n}\binom{r}{2}\right)\sum_{i=0}^{m-1} c_i\right) + \mathcal{O}\left(\sum_{i=0}^{m-1} c_i^2\right)$$

c_i depends on variance of Gaussian initialization.

Moments

Residual Network CK:

$$\mathrm{E}[\Sigma(x_0, x_0)^r] = \exp\left(\left(r + \frac{4}{n}\binom{r}{2}\right)\sum_{i=0}^{m-1} c_i\right) + \mathcal{O}\left(\sum_{i=0}^{m-1} c_i^2\right)$$

c_i depends on variance of Gaussian initialization.
Residual Network NTK:

$$\frac{\mathrm{E}[K(x_0, x_0)^r]}{(\mathrm{E}[K(x_0, x_0)])^r} \leq \exp\left(\binom{r}{2}\left(\max_i \beta_i + \frac{4}{n}\sum_i c_i\right)\right) + \mathcal{O}\left(\sum_{i=0}^{m-1}\sum_{j=1}^{d_i}\frac{1}{n_{i,j}^2} + \sum_{i=0}^{m-1} c_i^2\right)$$

β_i is noise parameter for i-th branch

Moments

Residual Network CK:

$$\mathrm{E}[\Sigma(x_0, x_0)^r] = \exp\left(\left(r + \frac{4}{n}\binom{r}{2}\right)\sum_{i=0}^{m-1} c_i\right) + \mathcal{O}\left(\sum_{i=0}^{m-1} c_i^2\right)$$

c_i depends on variance of Gaussian initialization.
Residual Network NTK:

$$\frac{\mathrm{E}[K(x_0, x_0)^r]}{(\mathrm{E}[K(x_0, x_0)])^r} \leq \exp\left(\binom{r}{2}\left(\max_i \beta_i + \frac{4}{n}\sum_i c_i\right)\right) + \mathcal{O}\left(\sum_{i=0}^{m-1}\sum_{j=1}^{d_i}\frac{1}{n_{i,j}^2} + \sum_{i=0}^{m-1} c_i^2\right)$$

β_i is noise parameter for i-th branch

Moments

Residual Network CK:

$$\mathrm{E}[\Sigma(x_0, x_0)^r] = \exp\left(\left(r + \frac{4}{n}\binom{r}{2}\right)\sum_{i=0}^{m-1} c_i\right) + \mathcal{O}\left(\sum_{i=0}^{m-1} c_i^2\right)$$

c_i depends on variance of Gaussian initialization.
Residual Network NTK:

$$\frac{\mathrm{E}[K(x_0, x_0)^r]}{(\mathrm{E}[K(x_0, x_0)])^r} \leq \exp\left(\binom{r}{2}\left(\max_i \beta_i + \frac{4}{n}\sum_i c_i\right)\right) + \mathcal{O}\left(\sum_{i=0}^{m-1}\sum_{j=1}^{d_i}\frac{1}{n_{i,j}^2} + \sum_{i=0}^{m-1} c_i^2\right)$$

β_i is noise parameter for i-th branch

Distribution Convergence

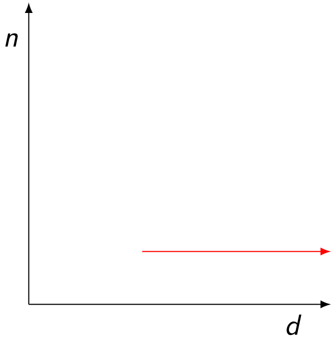


Figure: Different limiting behaviour of CK and NTK.

For residual network:

Red line: Diagonal elements of CK converges to log-normal distribution, and diagonal elements of NTK for each parameter converges in law to a log-normal distributed variable times CK of a feedforward network.

Experimental Verification

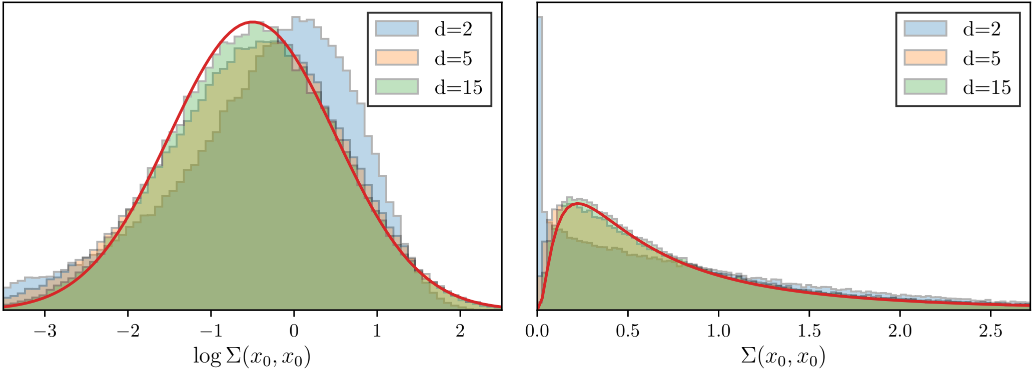
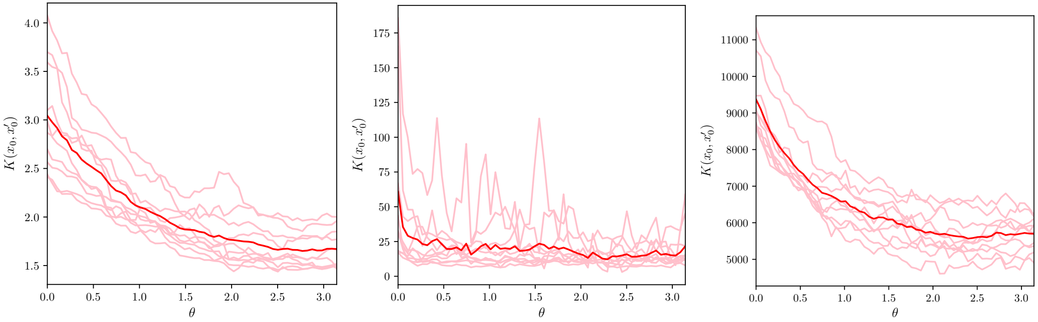


Figure: Distribution of CK for feedforward network. The red line is the theoretical limiting distribution given c and β.

Feedforward Network vs Residual Network



(a) FFNet (d=5) (b) FFNet (d=100) (c) ResNet (d=5,m=20)

Figure: NTK for different network with same hidden layer width n.

The NTK of deep residual network is less noisy and more informative, therefore deep residual network is easier to train than deep feedforward network.

Final Picture

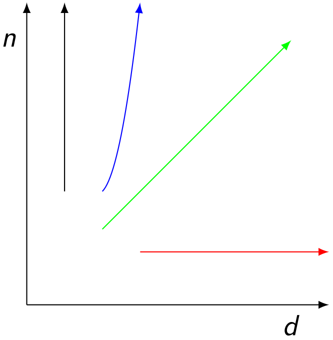


Figure: Different limiting behaviour of CK and NTK.

Black and Blue line: CK and NTK converge to fixed value.
Green line: For feedforward network, CK and NTK converge to log-normal distribution.
Red line: For residual network, CK converges to log-normal distribution and NTK converges to certain distribution.

Thank you