On Exact Computation with an Infinitely Wide Neural Net
Sanjeev Arora1,2 Simon S. Du2 Wei Hu1 Zhiyuan Li1 Ruslan Salakhutdinov3 Ruosong Wang3
1Princeton University 2Institute For Advanced Study 3Carnegie Mellon University

What happens when width (# of channels) is large?

• Recent papers [Li and Liang, Du et al., Allen-Zhu et al., Zou et al.] proved that NNs with sufficiently large width can achieve 0 training error via gradient descent.

• Over-parametrization doesn’t hurt generalization much. [Zhang et al., 17]

• [Jacot et al.] showed that as one increases the width to infinity, a certain limiting behavior, called neural tangent kernel (NTK), can emerge.

Main Questions:
1. Can we formally show that the prediction of NNs is equivalent to that of NTKs when width is sufficiently large?
2. How does NTK of classic CNNs (VGG or AlexNet) perform on standard datasets, such as CIFAR-10?

Fully-connected (FC) networks and Neural Tangent Kernel

\[ f(\theta, x) = W^{(L+1)} \cdot \sum_{d=1}^{d_L} \sigma \left( \sum_{j=1}^{n} W^{(j)} \sigma \left( \sum_{i=1}^{n} W^{(1)}x_i \right) \right) \]

where \( \sigma \) is activation, \( \sigma = \log \frac{1 + e^x}{1 + e^{-x}} \), \( \sigma' = \frac{1}{1 + e^x} \).

Square Loss:
\[ \ell(\theta) = \frac{1}{2} \sum_{t=1}^{n} (f(\theta, x_t) - y_t)^2 \]

Dynamics of Gradient Descent on \( \ell(\theta) \):
\[ \frac{d\theta(t)}{dt} = -H(t) \cdot \theta(t) - y \]

Here, \( \theta(t) \) is the parameters at time \( t \).

Implication: GD Trajectory \( \Rightarrow \ell_2 \) regression w.r.t. kernel \( \Theta(\theta, \theta) \)

\[ \Sigma(\theta) = \frac{1}{n} \sum_{i=1}^{n} (f(\theta, x_i) - y_i)^2 \]

Dependency on the derivative of non-linearity

Final output:
\[ \Theta(\theta, \theta) = \frac{1}{n} \sum_{i=1}^{n} (f(\theta, x_i) - y_i)^2 \]

In what sense does an ultra wide NN converge to NTK?

Existing results on asymptotic convergence:
• [Jacot et al.’18] sequential limit (\( d_1 \rightarrow \infty \rightarrow d_2 \rightarrow \infty \))
• [Yang’19] simultaneous limit (\( d_1 = \cdots = d_L \rightarrow \infty \))

Theorem (work): first non-asymptotic convergence result (m = width, n = # training data)

At initialization: Finite-width NTK converges to infinite-width NTK, i.e. \( H(0) \rightarrow \Theta(L) \), at the rate of \( O(m^{-0.5}L^{-1}\log n) \) for ReLU activation.

(During smooth phase, the rate could be in principle improved to \( O(m^{-0.5}L^{-2}\log n) \).

During training: The change of NTK over training, i.e. \( \|H(t) - H(0)\|_2 \) is bounded by \( O(m^{-1/6} \cdot \log n) \).

(Using Lemma from [Allen-Zhu, Li, Song])

Convolutional Neural Tangent Kernel (CNTK)

\[ CNTK \text{ formula: (1) Covariance } \Sigma(\theta) \text{ in NG-NN (Gaussian Process)} \]

For \( \alpha = 1, \ldots, M, (i,j), \beta \in \{1, 2, \ldots, L\} \), define

\[ \Sigma^{(\alpha, \beta)}_{(i,j)}(x, x') = \frac{1}{n} \sum_{i=1}^{n} \left( f^{(\alpha)}(\theta, x_i) \cdot f^{(\beta)}(\theta, x'_j) \right) \]

Final output (no GAP): \[ \Theta^{(\theta)}(x, x') = \left( \Theta^{(\theta)}(x, x') \right)_{i,j} \]

(2) Tangent Kernel \( \Theta^{(\theta)} \) by Dynamic Programming

\[ \Theta^{(\theta)}(x, x') = \sum_{i=1}^{n} \left( f^{(\alpha)}(\theta, x_i) \cdot f^{(\beta)}(\theta, x'_j) \right) \]

Without Global Average Pooling

Table 1: Classification accuracies of CNNs and CNTKs on the CIFAR-10 dataset. CNN-V represents vanilla CNN and CNTK-V represents the kernel corresponding to CNN-V. CNN-GAP represents CNN with GAP and CNTK-GAP represents the kernel corresponding to CNN-GAP.

Take-aways:

1. (CNTK are very powerful kernels)
2. (GAP significantly improves the test accuracy for both CNNs and CNTKs by 8%-10% in accuracy.)
3. There’s still a 5%-6% performance gap between CNTKs and CNNs.

• Can we explain the effect of Global Average Pooling?

Enhanced Convolutional Neural Tangent Kernels

Harnessing the Power of Infinitely Wide Deep Nets on Small-data Tasks

(NTK beats random forests, NN and GP!!)