EECS 6322 Week 12, Paper 1
On Exact Computation with an Infinitely Wide Neural Net

Shivani Sheth (Student #: 218011783)
shivs29@yorku.ca

I. Summary and contributions

The paper proposes an efficient exact algorithm for computing the extension of Neural Tangent Kernel to convolutional neural nets, known as Convolutional Neural Tangent Kernel (CNTK), along with an efficient GPU implementation of the algorithm. It also proposes a novel non-asymptotic theoretical proof showing that a fully-trained sufficiently wide network is equivalent to the kernel regression predictor using Neural Tangent Kernel (NTK). The model, based on the framework proposed by the authors, achieves competitive results on the CIFAR-10 dataset with respect to the existing methods in the domain.

In the paper, the authors use weakly-trained networks to refer to networks whose layers receive random initialization and only the top layer is trained by gradient descent, and they use fully-trained networks to refer to those networks whose all parameters are trained by gradient descent. Weakly-trained convolutional networks have reasonable performance on MNIST and CIFAR-10, and the kernels defined by the weakly-trained networks are given by: $ker(x, x') = E_{\theta\sim\mathcal{W}}\ [f(\theta, x)\ f(\theta, x')]$ *where* x, x' *are two inputs,* θ *represents the parameters, and* $\mathcal{W}$ *is an initialization distribution over* θ *(usually Gaussian). On the other hand, the kernels defined by the fully-trained networks are known as Neural Tangent Kernels (NTK), which is given by* $ker(x, x') = E_{\theta\sim\mathcal{W}}\ \langle \frac{\partial f(\theta,x)}{\partial \theta}, \frac{\partial f(\theta,x')}{\partial \theta} \rangle$ *. This Neural Tangent Kernel characterizes the behavior of fully-connected infinite width neural networks whose layers have been trained by gradient descent. The NTK is different from the Gaussian process kernels (used by the weakly-trained networks), and is defined using the gradient of the output of the randomly initialized net with respect to its parameters. The NTK can also be generalized to convolutional neural nets, and we call the corresponding kernel Convolutional Neural Tangent Kernel (CNTK).*

A fully-connected deep neural net architecture with an infinite width limit can be trained with respect to the l_2 *loss that gives rise to a kernel regression problem involving the neural tangent kernel (NTK). The final NTK expression for the fully-connected neural network can be given by* $\Theta^{(L)}(x, x') = \sum_{h=1}^{L+1}(\sum^{h-1}\ (x, x')\ \prod_{h'=h}^{L+1}\ \sum^{h'}\ (x, x'))$ *where the authors define* $\sum^{h'}\ (x, x') = 1$ *for convenience. Further the training process can be incorporated and the equivalence between a fully-trained sufficiently wide neural net and the kernel regression solution using the NTK can be shown. This equivalence between trained net and kernel regression can be given by* $|f_{nn}(x_{te}) - f_{ntk}(x_{te})| \leq \epsilon$ *where* $x_{te}\epsilon R^d$ *with* $||x_{te}|| = 1$ *, and probability at least* $1 - \delta$ *over the random initialization.*

The bound on the equivalence between trained net and kernel regression is non-asymptotic and according to the equivalence, the model can have the same number of neurons per layer, which is closer to practice. The theorem is a more precise characterization of the learned neural network, where the prediction is essentially a kernel predictor. Thus, to study the properties of the over-parameterized networks, such as their generalization power, it is sufficient to study the NTK because a fully-trained wide neural net enjoys the same generalization ability as its corresponding NTK.

To study the convolutional neural networks (CNNs) and their corresponding Convolutional Neural Tangent Kernels (CNTKs), the paper studies two architectures which are vanilla CNN and CNN with global average pooling (GAP). The performances of CNNs and their corresponding CNTKs have been evaluated on the CIFAR-10 dataset. The CNTKs have also been tested with 2,000 training data to see whether their performances are consistent with CNTKs and CNNs using the full training set. The results of the experiments show that CNTKs are very powerful kernels, and the best kernel, a 11-layer CNTK with GAP, achieves 77.43% classification accuracy on CIFAR-10. This results in a significant new benchmark for performance of a pure kernel-based method on CIFAR-10, being 10% higher than the previous methods. The results also show that for both CNN and CNTK, depth can affect the classification accuracy. This observation demonstrates that depth not only matters in deep neural networks but can also affect the performance of CNTKs. Additionally, as seen by the experiments, the global average pooling operation can significantly increase the classification accuracy by 8% - 10% for both CNN and CNTK which imply that many techniques which improve the performance of neural networks can also benefit kernel methods.

Finally, the experiments show that performances of CNTK-V-2Ks and CNTK-GAP-2Ks are highly correlated to their CNN-V, CNTK-V, CNN-GAP, CNTK-GAP counterparts. The CNTK-GAP-2Ks also outperform CNTK-V-2Ks by a large margin (about 8% - 9%). This can guide the neural archi-

tecture search where the kernel can be computed on a small training data, tested on a validation set, and then choose neural network architectures based on the performance of this small kernel on the validation set. Thus, by giving the first practical algorithm for computing CNTKs, the paper allows investigation of the behavior of infinitely wide (hence infinitely over-parameterized) deep networks, which turns out to be competitive as compared to their finite counterparts. The authors also give a fully rigorous proof that a sufficiently wide net is approximately equivalent to the kernel regression predictor, thus yielding a powerful new off-the-shelf kernel.

II. Strengths

The paper gives an exact and efficient dynamic programming algorithm to compute CNTKs for ReLU activation, which tries to settle the question of the performance of fully-trained infinitely wide nets with a variety of architectures. The paper also gives a rigorous, non-asymptotic proof that the NTK captures the behavior of a fully-trained wide neural net under weaker condition than the previous existing proofs.

III. Weaknesses

There exists a 5% - 6% performance gap between CNTKs and CNNs, which show that theoretical work on over-parameterization that operates in the NTK regime do not fully explain the success of neural networks. Since CNTKs exactly correspond to infinitely wide CNNs, the performance gap implies that finite width in neural networks has its benefits.

IV. Correctness

The claims and empirical methodology as given by the authors are correct and are supplemented by strong theoretical grounding. The assumptions made in the theorems have also been stated, which are generally also noticed in practice. The comparison studies with respect to the existing methods also show that the framework proposed by the authors outperforms its infinite width counterparts by a large margin, and performs competitively with respect to its finite width counterparts, thus proving the efficiency of the framework.

V. Clarity

The paper is well written in general with sufficient theoretical proofs explained before the architectural comparisons between the models. The paper also provides a relative study with respect to the previous work which has already been implemented in the domain, and thus supplements the understanding of the paper.

VI. Relation to prior work

Prior work in Gaussian processes (GP) include methods that correspond between infinite neural networks and kernel machines and extend this correspondence to more general neural networks including shallow neural networks, along with deep and convolutional neural networks. However, these kernels represent weakly-trained nets, instead of fully-trained nets as discussed by the paper.

Prior work in kernel gradient descent also shows that if the number of neurons per layer goes to infinity in a sequential order, then the kernel remains unchanged for a finite training time, and hence derived the Neural Tangent Kernel (NTK). Additionally, a formula of Convolutional Neural Tangent Kernel (CNTK) was also derived in other works, along with other mechanistic ways to derive NTK for different architectures. Compared with these works, the CNTK formula as defined by the paper has a more explicit convolutional structure and results in an efficient GPU-friendly computation method.

VII. Reproducibility

Enough details are provided in terms of code implementations, model architectures and the theoretical changes from the previous models that can help reproduce the major details of the work proposed by the authors.

VIII. Additional comments

An additional study including the behaviour of these infinitely wide networks with respect to features such as Batch Normalization or Residual Layers would supplement the work proposed by the authors.