**Kervolutional Neural Networks**

Chen Wang Jianfei Yang Lihua Xie Nanyang Technological University, Singapore

Junsong Yuan, State University of New York at Buffalo, USA

*Reviewed by Shubham Sharma, Roll No 190010065*

1. **Introduction**

*This work introduces a new operation, kervolution (kernel convolution) to extend convolution to non-linear spaces using the “kernel kick” while keeping the advantages of linear convolutions. It claims to enhance the model capacity, and capture higher order interactions of features, via patch-wise kernel functions, but without introducing additional parameters in the model. They back their claims with extensive experimentation that show that Kervolutional neural networks (KNN) achieve higher accuracy and faster convergence than the baseline CNN*.

1. **Motivation**

CNN’s have been greatly successful in many computer vision tasks. However not much work has been done on establishing convolution on non-linear spaces. Existing models mostly rely on activation layers to introduce non-linearity in the model. But these **activation layers** can only **provide** **point-wise non-linearity**. Here the authors argue that the network may perform better if convolution can be generalized to patch-wise non-linear operations via some kernel trick

1. **Background**

In recent years, researchers have been trying to extend convolution to non-linear spaces.

Directly introducing higher order terms introduces a large number of additional parameters thus increasing the training complexity exponentially.

Another method to introduce the high order features was to explode the pooling layer. However, it is not able to extract non-linear

features in a patch-wise manner. Moreover, the features (coordinates) are still needed to be

computer explicitly which increases computation complexity.

To solve these problems kervolution is defined to generalize convolution via the **kernel trick.**

1. **Approach**

On the mathematical side they basically define a kervolutional operator on their input vector and express element of the output as an inner product *(similar to what we do in a CNN).* What they do different here is instead of taking the inner product of the inputs and weights (like in a CNN) they take the inner products of a **non-linear mapping** function applied on the weights and the input vector.

This enables them to **extract features in a high dimensional space (d)** but in turn shoots up the computational complexity. In order to bypass this explicit calculation of mapping each element they use the kernel trick (mathematical trick).

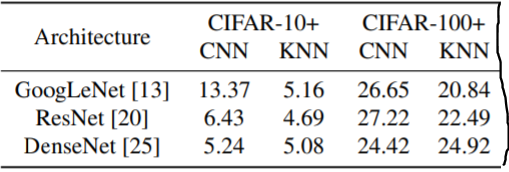
Using the kernel trick, they are able to **retain the complexity of**  i.e. same as of convolution. They establish convolution as the linear case of kervolution from their definition after applying the trick.

From their definition it is clear that the number of elements of w (weights) are not increased.

They also go on to prove easily that this kervolutional operation is **invariant to translation** (using the same proof of concatenating pooling layers as in a typical CNN).

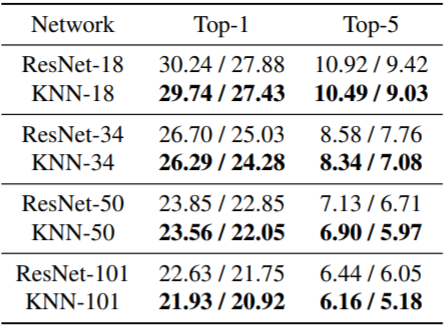
They study only the polynomial kervolution and gaussian kervolution for all their tests. They first attempt to only replace the convolutional layers with the kervolutional ones keeping the activation functions intact. And then finally they remove even the ReLU to test their network to get better performing results than CNN (99.11% as compared to 92.22% of CNN).

1. **Performance**
2. **CIFAR-10/100** dataset:

Was tested on a single GPU of Nvidia GeForce GTX 1080Ti using different architectures. In all architectures KNN outperformed CNN (different based on diff architectures)

Validation error (%) on diff architectures

1. **ImageNet** dataset :

This was tested on four Nvidia Tesla M40 again tested on sone diff architectures. Results are more or less the same with KNN outperforming CNN on almost every architecture by a **small** **margin**.

Validation errors(%) with single-crop/ 10-crop tresting

1. **Conclusions**

The paper introduced Kervolution to basically generalise convulation to non-linear space.

The key conclusins to take away were :

* Kervolution retains the advantages of convulation.
* The new useful features in

kervolutional layers are:

* + Translational equivalance
  + Sharing weights (no new parameters)
  + Increased model capacity
* They demonstrated that, with choosing appropriate kernels, the performance of the network can be significantly improved on CIFAR and ImageNet dataset by using a kervolutional layer.

1. **Limitations**

These are the points which I felt the paper lacked upon :

* It only tests **two kernels** for the layers i.e **polynimial** and **gaussian**, which may not be most optimal.
* In section 3.3 they claim that the network “enhances model capacity” but they make no effort in defining this “model capacity” and provide no explaination for the same. It is a kind of speculation from their part.
* The kervolutional layer does not outperform the convulational layer when we use the larger dataset (ImageNet) as compared to when we use the CIFAR dataset.