Hermite Convolutional Networks

Leonardo Ledesma[1,2], Jimena Olveres[2], and Boris Escalante-Ramírez[2](✉)

[1] Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, Mexico City, Mexico
[2] Facultad de Ingeniería, Universidad Nacional Autónoma de México, Mexico City, Mexico
boris@unam.mx

Abstract. Convolutional Neuronal Networks (CNNs) have become a fundamental methodology in Computer Vision, specifically in image classification and object detection tasks. Artificial Intelligence has focused much of its efforts in the different research areas of CNN. Recent research has demonstrated that providing CNNs with a priori knowledge helps them improve their performance while reduce the number of parameters and computing time. On the other hand, the Hermite transform is a useful mathematical tool that extracts relevant image features useful for classification task. This paper presents a novel approach to combine CNNs with the Hermite transform, namely, Hermite Convolutional Networks (HCN). Furthermore, the proposed HCNs keep the advantages of CNN while leading to a more compact deep learning model without losing a high feature representation capacity.

Keywords: Hermite transform · Convolutional neural networks · Kernel modulation · Orientation · Scale · Feature map

1 Introduction

One of the challenges of image classification tasks is robustness against geometric transformations, such as rotation, deformation and scale. A large number of models and techniques of CNNs has been developed to deal with these issues, for example models based on attention mechanism or actively rotating filters [1], even hand-crafted filters with no learning processes involved. Moreover, DCNNs (Deep CNNs) have emerged as an alternative of feature extraction [10], however, these techniques derive in expensive training and complex model parameters, without solving the problem of geometric transformations.

Recently, a priori learning approaches have emerged as a way to help CNNs improve their performance by using mathematical transformations based on HVS (Human Visual System), such is the case of the Gabor CNN (GCNs) [2].

In [2] Gabor Orientation Filters (GoF) were proposed as the result of a new process called Modulation. This paper is inspired on the same idea, but instead of Gabor transform, we introduce the Hermite transform, which is also a human perception relevant image representation model [3, 4].

1.1 CNN Using Mathematical Transforms

Some important models of CNN's based on mathematical transforms have been described in the literature:

(1) *Scattering Networks* [5, 6]: This kind of architecture uses a wavelet scattering based on the expression of receptive fields in CNNs. The wavelets are another way to introduce mathematical transforms in an CNN.
(2) *Oriented Response Networks (ORN):* In [1] ORN resembles a hierarchical orientation encoder using Actively Rotation filters in order to discriminate structures.
(3) *Gabor filters:* Two approaches; the first one as a pre-processing model in order to improve the accuracy of CNNs [7], or by applying the Gabor filters in the 1st convolutional layer instead of using random filter kernels.

The second approach; modulating the learned convolution filters using the Gabor filters [2]. This idea inspires this paper.

1.2 Motivation

In [2], it was demonstrated that after the training process in an Alex Net architecture [8], the learned filters resemble very much Gabor filters oriented towards different directions, which are known to be good detectors of image structures, such as edges.

This idea is of great importance, since it suggests that the convolutional stages of a CNN aim at learning primitive image features, that we have known for years, to be relevant for image analysis and characterization, i.e., edges, lines, corners. These features can be obtained by designing appropriate filters for this task. Nevertheless, the learned filters behave like original Gabor filters after several epochs of training.

Recent research use Gabor filters in the convolution layers, for example; in [9], Gabor filters are used in the first or second convolution layers as a feature extractor such as a pre-processing stage in a CNN [10].

From the point of view of image processing this transform has been widely used in image segmentation, classification and texture detection.

In this paper, we introduce the Hermite transform as a model for building CNNs. The HT has the following characteristics:

- It allows multiscale analysis
- It is useful to perform directional analysis
- Its filter functions are based on Gaussian derivatives, which in turn are good detectors of image primitives, such as edges, lines, corners, etc.
- The HT allows perfect reconstruction and is shift invariant.

We present this work as a novel attempt to improve the feature extractor and performance of a CNN, taking advantage of Hermite transform and its benefits.

1.3 Contributions

We propose to use Hermite filters in a CNN basic architecture in such a way that the number of parameters is reduced and on the other hand, robustness of orientation and scale changes are enhanced.

Based on the idea given by [2], we will use the HT to generate a modulation process and a modified convolutional process to improve the performance of a CNN.

Therefore, the contributions of this papers are:

- The first attempt to incorporate Hermite filters into a CNN architecture in the state of the art.
- Prove the advantages of using a mathematical transform based on HVS in comparison to traditional methods.
- Develop modulation and modified convolutional processes that can be easily extended to more complex architectures like ResNet or Wide Residue Network (WRN) in the near future.

2 Related Work

2.1 Hermite Transform

The Hermite transform is defined by the analysis functions called D_n [3, 4] that comprise a set of filters obtained from the product of Gaussian window g and the Hermite polynomials H_n,

$$D_n(x) = \frac{(-1)^n}{\sqrt{2^n n!}} \frac{1}{\sigma\sqrt{\pi}} H_n\left(\frac{x}{\sigma}\right) e^{\left(-\frac{x^2}{\sigma^2}\right)} \tag{1}$$

where n is the polynomial order.

It is worth mentioning that the analysis functions are spatially separable and rotationally symmetric. A straight forward extension provides the 2D expression for the analysis functions:

$$D_{n-m,m}(x,y) = H_{n-m,m}(-x,-y) \cdot g^2(-x,-y) \tag{2}$$

where $(n-m)$ and m are the polynomial order in x and y axes for $n = 0, \ldots, \infty$ and $m = 0, .., n$.

The associated orthogonal polynomials are defined:

$$H_{n-m,m}(x,y) = \frac{1}{\sqrt{2^n m!(n-m)!}} H_{n-m}\left(\frac{x}{\sigma}\right) H_m\left(\frac{y}{\sigma}\right) \tag{3}$$

In the case of a discrete implementation of the Hermite filters, we use parameter N equal to the finite length of the filter that in turn is related to the scale, i.e., the standard deviation of the Gaussian window as follows: $N = 2\sigma^2$ (Fig. 1).

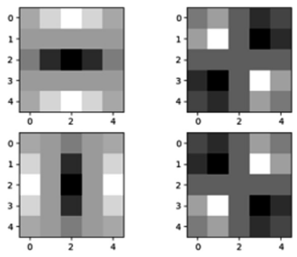

Fig. 1. The image illustrates of the Hermite Filters of length $N = 5$ especifically $D_{2,0}, D_{1,1}$, $D_{0,2}$ y $-D_{1,1}$ displayed as follows: $\begin{bmatrix} D_{0,2} & D_{1,1} \\ D_{2,0} & -D_{1,1} \end{bmatrix}$

3 The Proposed Method

Hermite Convolutional Networks (HCN) are created with the purpose of being used in the following tasks in computer vision:

- Image Classification
- Object Detection
- Texture Discrimination
- Segmentation

During the training stage learned convolution filters are adjusted followed by modulation process with a Hermite filter bank.

Similar to GCNs, Hermite orientation filters (HoFs) are the result of this modulation process and are used to generate feature maps in each layer.

The three main process are:

3.1 Hermite Orientation Filters (HoFs) and Modulation Process

HoFs are obtained by element product between Hermite Filter bank (see Fig. 2) and the learned filters in each convolutional layer. The HoFs have the size:

$$C_{out} \times C_{in} \times U \times N \times N$$

where C_{out}, C_{in} are the channel of output and input feature map respectively, and U is the number of orientation filters of the Hermite bank selected, in this paper $U = 4$.

Let the modulation process be:

$$C^{n}_{i,n-m,m} = C_{i,o} \cdot D_{n-m,m} \tag{4}$$

The $C^{n}_{i,n-m,m}$ is a modulated filter of the learned filter $C_{i,o}$ It is important to mention that the scale parameter N in one convolutional network is the same for all orientation filters in the Hermite bank but changes between layers.

During the modulation process Hermite filters adjust the kernels of the learned filters in order to extend the properties of the transformation to the back-propagation.

HoFs are defined by:

$$C_i^n = \left\{ C_{i,0,2}^n, C_{i,1,1}^n, \ldots, C_{i,2,0}^n \right\} \tag{5}$$

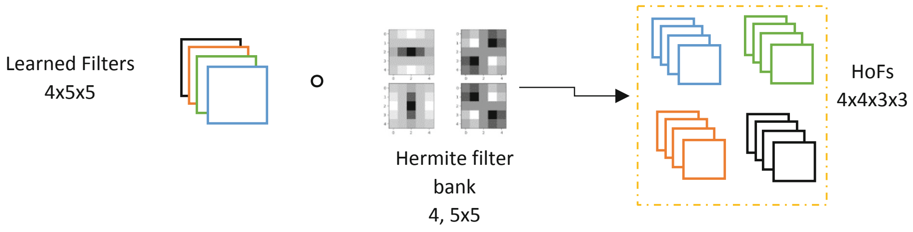

Fig. 2. The modulation process of HoFs

3.2 HCN Convolution Process

The HoFs are used in this process to produce an output feature map $\widehat{F}$, according to equation:

$$\widehat{F}_{i,n,n-m} = \sum\nolimits_{j=1}^{U} F^{(j)} \otimes C_{i,n,n-m}^{(j)} \tag{6}$$

Where F is an input feature map and $C_{i,nn-m}^{j}$ is set of HoFs (Fig. 3).

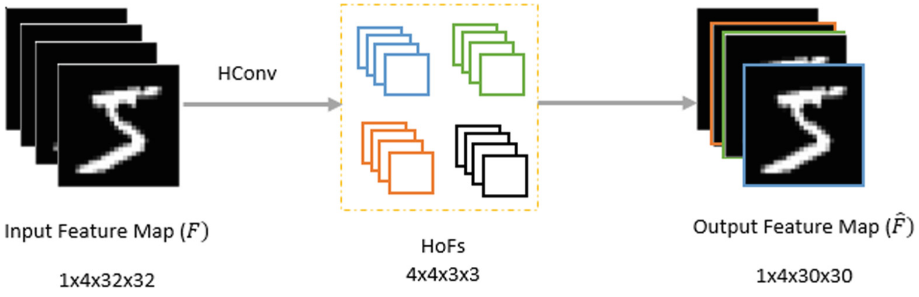

Fig. 3. The convolution process of HoFs and Feature Map

3.3 Back-Propagation Process

The HoFs only affect in forward calculation in HCNs. However, the back-propagation process updates the learned filters in each layer and the HoFs are recalculated in each epoch.

$$C_{i,o}^{t+1} = C_{i,o}^{t} - \eta \frac{\partial L}{\partial C_{i,o}^{t}} \tag{7}$$

Where L is the loss function, in this case, cross-entropy.

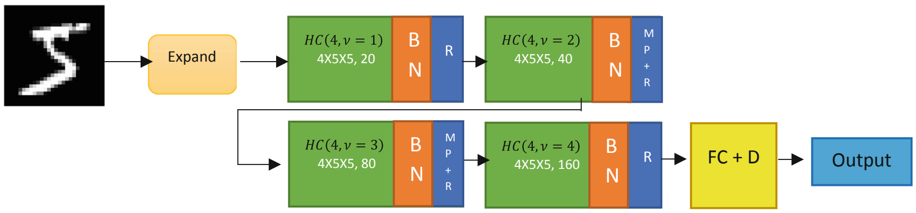

Fig. 4. Basic architecture of the Hermite Convolutional Network (HCN) where BN: Batch Normalization, MP: Max Pooling, FC: Fully Connect, D: Dropout, R: ReLu Activation, HC: Hermite convolutional layer.

4 Experiments and Implementation

The general implementation is based on [2] but is extended to support Hermite transform, so the module of Gabor Filter Bank is substituted by Hermite Filter Bank. The basic architecture HCN is shown in Fig. 4.

The proposed HCN architecture was evaluated with the datasets: MNIST [11, 12], Fashion-MNIST [13] y E-MINIST [14]. We evaluated these experiments in a GPU NVIDIA GeForce 1080 with an isolated environment in order to work with other versions of Python and PyTorch.

Algorithm 1 Hermite Convolutional Networks (HCN)

```
1.  Set hyper-parameters and network structure.
2.  Design the Hermite filters bank given a filter
    order $n$ keeping the rule: $[n-m, m], n \leq N-1$, and $m \leq n$
3.  Start training
4.  **Repeat**
5.          Select a mini batch set of training images.
6.          Produce HoFs using Eq. 4
7.          Apply Forward convolution using Eq. 6
8.          Calculate the cross-entropy and run back-propagation process using Eq. 7
9.          Update the learned filters
10. **Until** the maximum epoch
```

5 Results

5.1 MNIST

MNIST dataset was divided in 10,000 samples for validation and 50,000 samples for training. We evaluated different optimizers: SGD, ADAM and ADEDELTA. We used a batch size of 128 and 50 epochs for training and testing.

In the case of ADEDELTA optimization the initial learning rate was 0.01, the learning weight decay was set as 0.00003, these hyper-parameters were the same for ADAM and SGD.

Based on [2], we designed filters in four orientation and sizes of 5 × 5 and 7 × 7, we did not use 3 × 3, because it is too small to represent all the properties of Hermite transform.

In Table 1, we show the results for MNIST using different algorithms to calculate the gradient descent with filters 5 × 5.

Table 1. Results (error rate and accuracy vs gradient descent algorithm)

%Accuracy/%error	Gradient descent algorithm
99.51/0.49	SGD with one scale
99.46/0.54	SGD with four scales
99.42/0.58	ADAM with four scales
99.34/0.66	Adedelta with four scales

Table 2. Results (error rate and accuracy vs time and different filter sizes with SGD algorithm)

%Accuracy/%error	Time	Filter size per layer
99.51/0.49	9 min 30 s	SGD (5, 5, 5, 5)
99.39/0.61	9 min 10 s	SGD (3, 3, 3, 3)
99.31/0.58	25 min 9 s	SGD (7, 7, 7, 7)
99.27/0.66	12 min 10 s	SGD (3, 3, 5, 5)

Table 3. Results comparison on MNIST

% error	# network stage kernel	Architecture	Year
0.21	5 CNN/6-layer: 784-50-100-500-1000-10-10	Regularization of Neural Networks using DropConnect [15]	2013
0.23	35 CNNs, 1-20-P-40-P-150-10	Multi-column Deep Neural Networks for Image Classification [16]	2012
0.23	6-layer 784-50-100-500-1000-10-10	APAC: Augmented PAttern Classification with Neural Network [17]	2015
0.57	10-20-40-80	ORF4 (ORAlign) [1]	2017
0.48[a]	20-40-80-160	GCN4 (7 × 7) [2]	2018
0.49	10-20-40-80	HCN4 (5 × 5)	2019

[a]In this case the authors reported an error of 0.42 but we were not able reproduce it.

We also tested different filter sizes per layer in order to find the best performance, Table 2 shows the results for this experiment. Regarding computing time, as shown, the best result is obtained when filter size is 5 × 5.

The architectures reported in the state of the art have more millions of parameters and consequently need more computing time. In contrast, HCN and GCN reduce considerably the computing time and the complexity of parameters. Moreover, the HCN and GCN architectures are more compact in comparison with the others.

5.2 Fashion-MNIST and E-MNIST

Both datasets have the same number of samples as MNIST; we separated the sets into two groups of validation and training as in the first experiment; the hyper-parameters were the same (Fig. 5).

It was necessary to normalize the datasets before training to avoid gradient fading problems and divergence of the CNN. In Table 3, the state-of-the-art results are shown using the same this simple image database with their corresponding architectures and parameters (Tables 4 and 5).

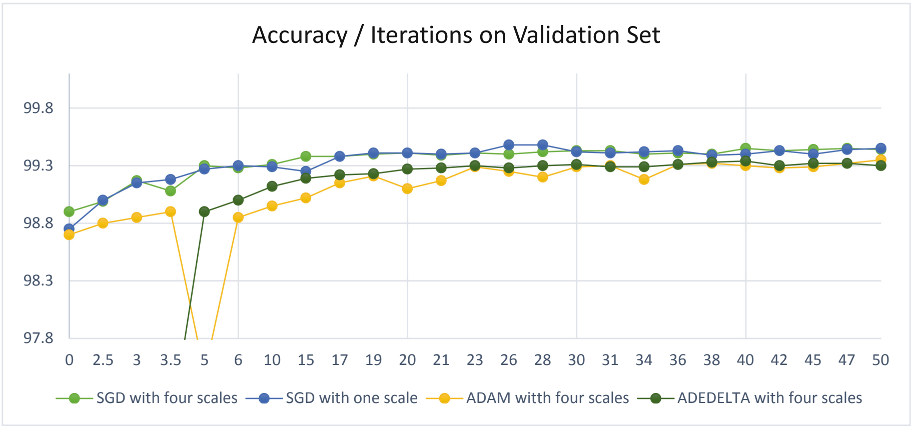

Fig. 5. In this graphic the x-axis is the number of iterations (x 1K) and y-axis is the accuracy for the test set, it's shown the performance of HCN with several algorithm gradient descent, being SGD with one scale the best performance.

Table 4. Results (error rate and accuracy rate vs time and different filter sizes with SGD algorithm on fashion MNIST)

%Accuracy/%error	Time	Filter size per layer
97.09/2.91	10 m 12 s	SGD (5, 5, 5, 5)
95.01/4.99	10 m 2 s	SGD (3, 3, 3, 3)
90.13/9.87	30 min 2 s	SGD (7, 7, 7, 7)
97.19/2.81	**14 min 14 s**	**SGD (3, 3, 5, 5)**

Table 5. Results (error rate and accuracy rate vs time and different filter sizes with SGD algorithm on E-MNIST)

%Accuracy/%error	Time	Filter size per layer
99.67/0.33	**9 m 50 s**	**SGD (5, 5, 5, 5)**
99.60/0.40	8 m 55 s	SGD (3, 3, 3, 3)
99.41/0.59	25 min 29 s	SGD (7, 7, 7, 7)
99.46/0.54	10 min 14 s	SGD (3, 3, 5, 5)

6 Conclusions

This paper presents a novel deep model incorporating Hermite filters to CNN, aiming at improving the performance with a high feature representation capacity. The proposed Hermite Convolutional Networks is advantageous tool for image classification. Results show that HCN got an accuracy near to state of the art of the MNIST database with lower number of parameters. As future work, more image databases will be tested on HCN, such as ImageNet and CIFAR-10. Furthermore, new hybrid architectures, e.g. similar to ResNet, will be developed in order to tackle more complex problems.

Acknowledgments. This publication has been sponsored by the grant UNAM PAPIIT IA103119, grant UNAM PAPIIT IN116917 and Consejo Nacional de Ciencia y Tecnología (CONACyT).

References

1. Zhou, Y., Ye, Q., Qiu, Q., Jiao, J.: Oriented response networks. In: 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, vol. 2017, pp. 4961–4970. IEEE, Honolulu (2017)
2. Chen, S.L., Zhang, B., Han, J., Liu, J.: Gabor convolutional networks. IEEE Trans. Image Process. **27**(9), 4357–4366 (2018)
3. Martens, J.-B.: The Hermite transform-theory. IEEE Trans. Acoust. Speech Signal Process. **38**(9), 1595–1606 (1990)
4. Martens, J.-B.: The Hermite transform-theory. IEEE Trans. Acoust. Speech Signal Process. **38**(9), 1607–1618 (1990)
5. Bruna, J., Mallat, S.: Invariant scattering convolution networks. IEEE Trans. Pattern Anal. Mach. Intell. **45**(8), 1872–1886 (2013)
6. Sifra, L., Mallat, S.: Rotation, scaling and deformation invariant scattering for texture discrimination. In: 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2013, pp. 1233–1240. IEEE, Portland (2013)
7. Calderón, A., Roa, S., Victorino, J.: Handwritten digit recognition using convolutional neural networks and Gabor filters. In: Proceedings of the International Congress on Computational Intelligence CIIC 2003, Medellin (2003)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Proceedings of the Advances in Neural Information Processing Systems, Nevada, vol. 1, pp. 1097–1105 (2012)
9. Sarwar, S.S., Panda, P., Roy, K.: Gabor filter assisted energy efficient fast learning convolutional neural networks. In: Proceedings of the IEEE/ACM International Symposium on Low Power Electronics and Design, Taipei, pp. 1–6 (2017)
10. Yao, H., Chuyi, L., Dan, H., Weiyu, Y.: Gabor feature based convolutional neural networks. In: Proceedings of the International Conference on Information Science and Control Engineering, pp. 386–390. IEEE, Beijing (2016)
11. LeCun, Y.: The MNIST Database of Handwritten Digits (1998). http://yann.lecun.com/exdb/mnist/. Accessed 22 Mar 2019
12. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)

13. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. https://github.com/zalandoresearch/fashion-mnist. Accessed 18 Mar 2019
14. The E-MNIST Database of Handwritten character. https://www.nist.gov/itl/iad/image-group/emnist-dataset. Accessed 18 Mar 2019
15. Wan, L., Zeiler, M., Zhang, S., LeCun, Y., Fergus, R.: Regularization of neural networks using dropconnect. In: Proceeding ICML 2013 Proceedings of the 30th International Conference on Machine Learning, Atlanta, vol. 28, pp. 1058–1066 (2013)
16. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3642–3649. IEEE Computer Society, Washington (2012)
17. Sato, I., Nishimura, H., Yokoi, K.: APAC: Augmented PAttern Classification with Neural Networks. https://arxiv.org/abs/1505.03229. Accessed 24 Mar 2019