Neural Network Image Classification in Fourier Space

Tabin Dharanikota
University of Chicago, Department of Physics

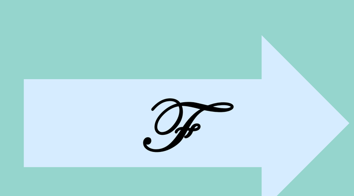
Background

Fourier analysis has been vital since its conception by Jean Baptiste Fourier for solving PDEs, signal processing, and, more recently, image analysis.

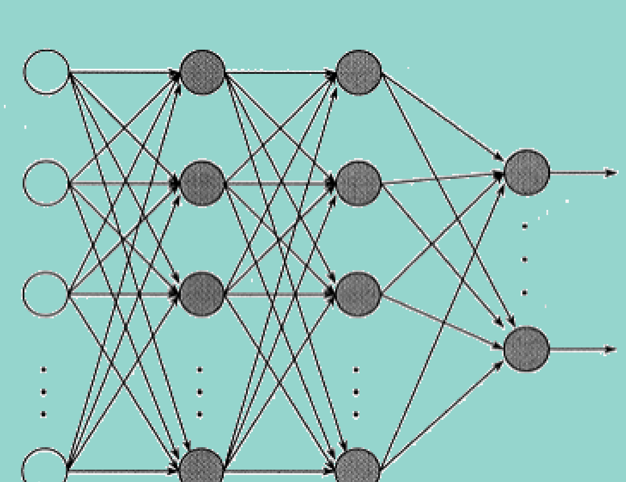
We can use the Fourier transform for image filtering, compression, noise reduction, and recognition [1].

Another field having a large impact on image analysis is deep learning—providing a robust method for image recognition with even simple feed forward networks. Both of these ideas were bound to intersect (for example: Wood, 1996 [2].)

I would like to see if our current methods of neural networks can be applied to the frequency space provided by the Fourier transform.



Example of Fourier Transform of Image.
The square wave analog.



Sketch of the Neural Network:
Input layer: 784 nodes (one for each pixel)
First Hidden Layer: 128 nodes
Second Hidden Layer: 128 nodes
Output Layer: 10 nodes

Objectives

Can we improve image recognition by moving to the frequency space?

Are there transformations that will improve accuracy in the spatial space or frequency space?

If we can or cannot improve the accuracy, what can the inner workings of our network tell us why it is the case?

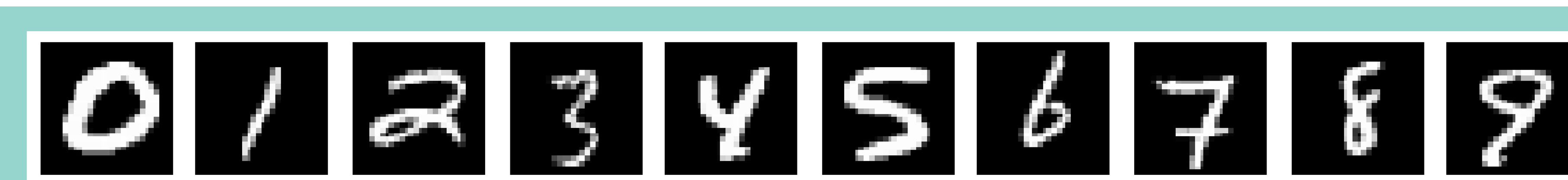
Methodology

Used the MNIST dataset: 60,000 28x28 images of handwritten digits.
Made 14 datasets with various transformations and amount of the transformations.

- Two different image transformations: blur and rotation
- Scaled amount transformed each image in three different ways (0.33, 0.67, and 1.00 scaling)
- Took those datasets also to frequency space

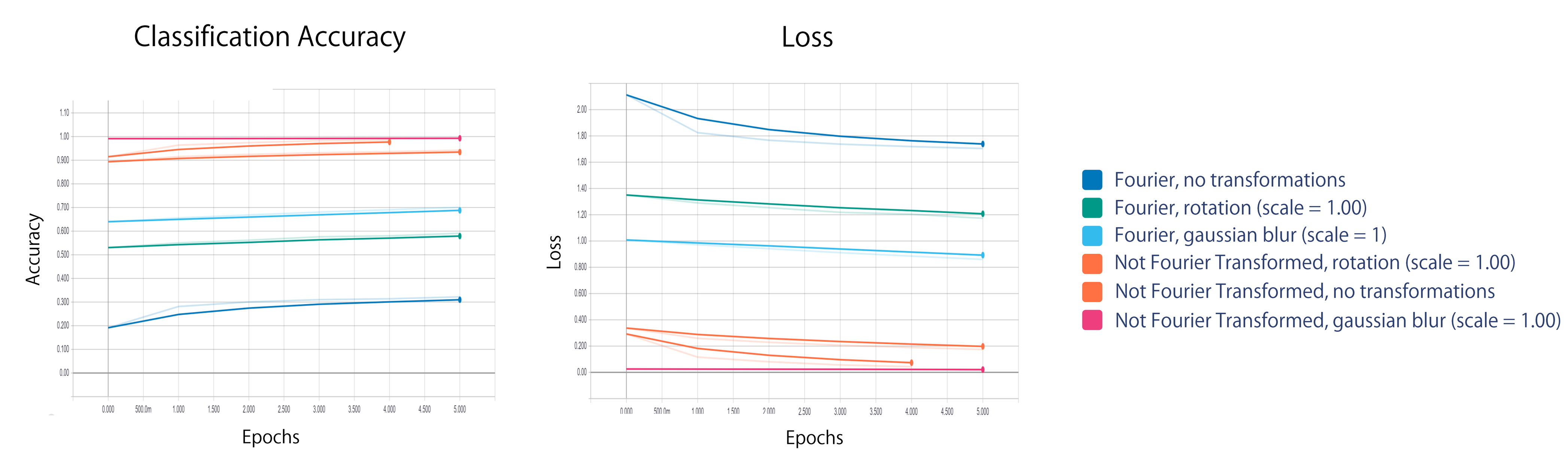
Model Specifications:

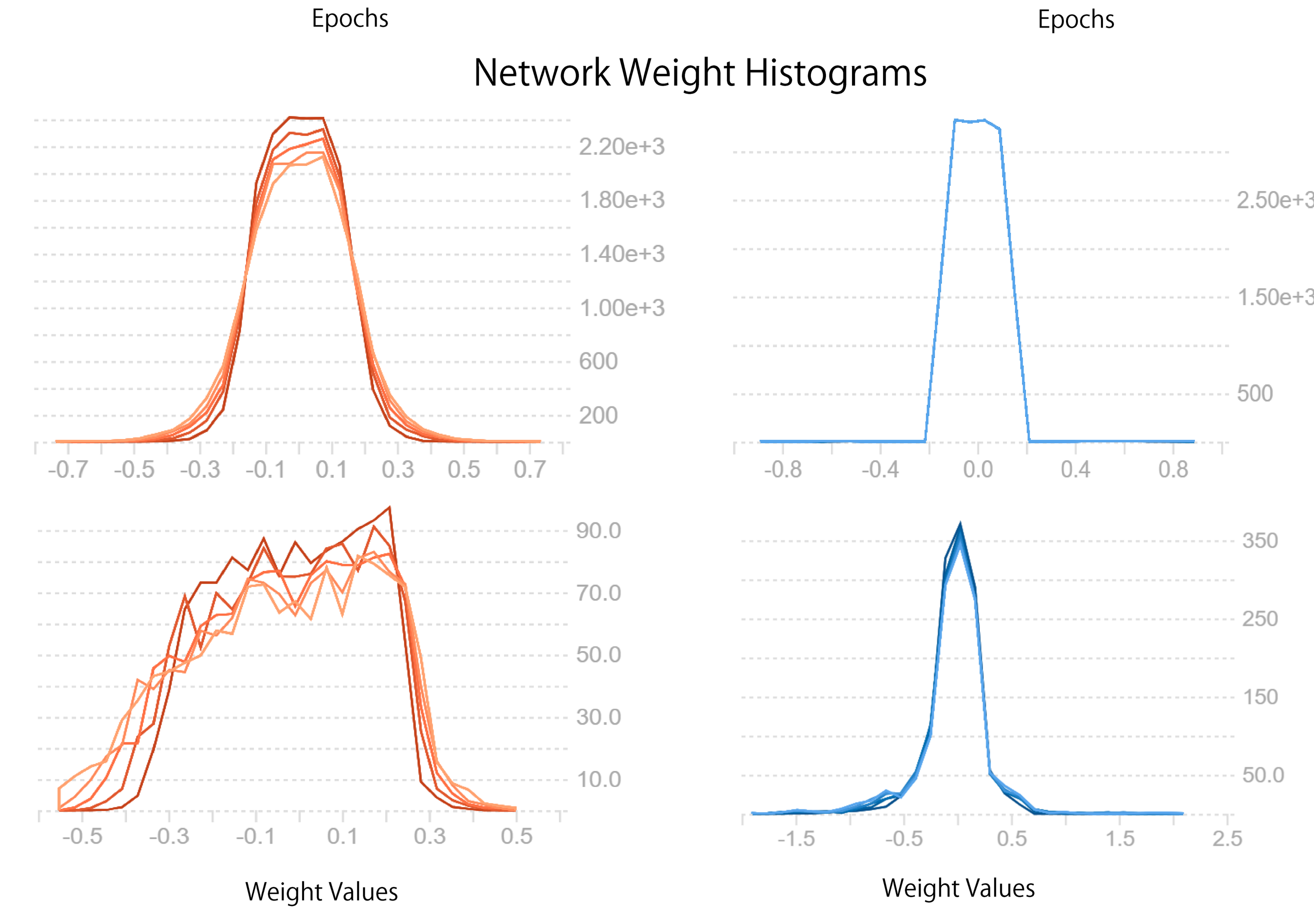
- Two dense hidden layers with a rectified linear units (ReLU) activation function.
- Each hidden layer has 128 nodes.
- Output layer has 10 nodes (one for each digit) with a softmax activation function.
- Sparse categorical cross entropy loss function with Adam optimizer.



Sample images from the MNIST dataset

Results





The top row shows the weights of the first hidden layer.
The bottom row the weights of the second hidden layer.

The orange graphs show the non-Fourier transformed data set without any image transformations.
The blue graphs show the Fourier transformed dataset without any image transformations.

The overlapping lines show the weight distribution at different epochs. The darker the line, the later the epoch.

Conclusions

The Fourier transformed results did significantly worse than the non-transformed images (with around a 40% accuracy difference.) We can see this fact manifest in the fact that even as the training went on, the Fourier space model's weights did not adjust much.

However, introducing the image transformations significantly improved the results in the frequency space (by about 30%).

Overall, when the blur transformation was applied to the data, for both the Fourier and non-Fourier images, the accuracy and loss improved. Perhaps this can be explained by the fact that blurring the data reduces the impact of noise and, with high blurring, leaves the more fundamental shape of the original image.

Acknowledgements

Citations:

[1]: Chen, Qin-Sheng, et al. "Symmetric Phase-Only Matched Filtering of Fourier-Mellin Transforms for Image Registration and Recognition." *IEEE*, vol. 16, no. 12, 1994, pp. 1156–1168.

[2]: Wood, Jeffrey. "Invariant Pattern Recognition: A Review." *Pattern Recognition*, vol. 29, no. 1, 1996, pp. 1–17.

Thank you to Professor David Miller for the immense amount of knowledge he has provided us. I know I will be using this experience for many years to come.

GitHub : https://github.com/TDharanikota/Phys250-Final-Project
(The QR code also links to this GitHub repository.)