Wavelet Pooling Techniques in Convolutional Neural Networks

Alec Hon, Andrew Yeh
https://github.com/abhon/DSGA_1013_Final_Project

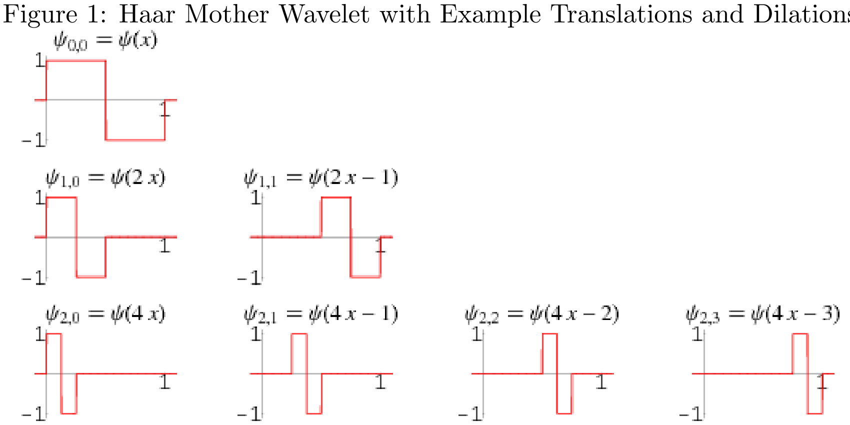
May 2020

1 Introduction

Wavelets, or 'small waves', are a family of zero mean functions that are used mostly in signal processing, with all wavelets being characterized with a small active support and integrating to one.

A wavelet is comprised of a mother wavelet and a father wavelet family, the first of which serves as a prototype function for the wavelet family, and the second of which serves as a scaling function to scale the wavelet to a determined time-frequency resolution. The Haar Wavelet, the most simple family of wavelets, is shown below:

Figure 1: Haar Mother Wavelet with Example Translations and Dilations



While Fourier Transforms are powerful in representing a function across the entire time domain by transforming a function into sinusoids, they struggle with studying functions over a local time space as many extra coefficients need to be added into the Fourier Transform of a function to effectively cancel out all the amplitude outside the area of study (Strang, 1993). On the other hand, wavelet analysis expands a function into translations and dilations of a mother wavelet function and allow the study of functions at a local time range, making it possible to recreate functions that include abrupt changes with fewer coefficients compared to a Fourier transform.

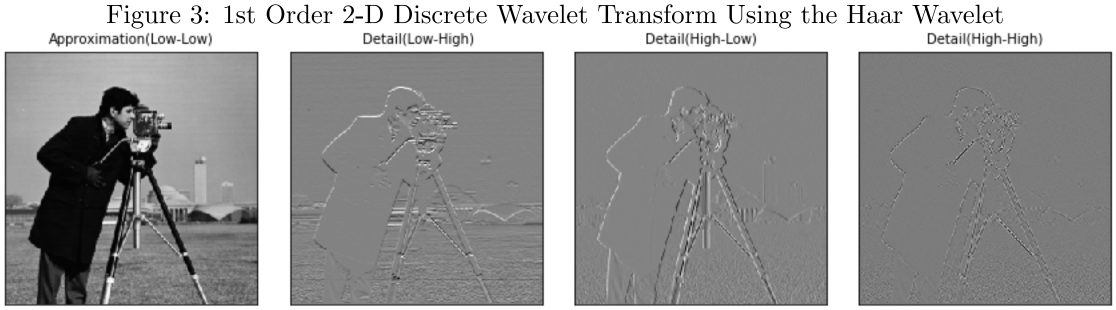
Wavelets have been used in image processing, with a large amount of study being focused on image compression and denoising (Tan, Khon, Mok, 2007; Goldberg, Pivovarov, et. al 1994).

This paper explores the use of the Discrete Wavelet Transform (DWT) as a pooling method in Convolutional Neural Networks (CNNs). Pooling in CNNs for image recognition is done to reduce the dimensionality of the training data for computational efficiency and regularization while still retaining the crux of the data for our neural network to learn from. Conventional CNN pooling techniques include max pooling and mean pooling, with both pooling techniques showing shortcomings, as details in images are either diluted or lost upon using either pooling algorithm as can be seen in Figure 2. As a result, a different pooling method could better approximate the image.

Figure 2: Deficiencies in Max and Mean Pooling(Image taken from Li, Williams 2018)

The Discrete Wavelet Transform (DWT) is a way to approximate images using a wavelet basis. Like the Discrete Fourier Transform, the DWT has an Inverse DWT, such that it is possible to revert transformations from the DWT. Many well-known wavelet families, such as Haar, Daubechies, Biorthogonal, etc., include filter banks consisting of high-pass and low-pass filters for each wavelet. High-pass filters resolve to the detail within images while low-pass filters lead to approximation, and these can be combined through the 2-D DWT. We use a combination of these filters due to the fact that utilizing a filter cuts half of the frequencies, forcing us to up-sample by two to avoid aliasing. An example of the first order 2-D DWT on an image is shown below:

Figure 3: 1st Order 2-D Discrete Wavelet Transform Using the Haar Wavelet



If the DWT can be used as a pooling operation for CNNs, it may result in stronger predictive power and higher classification accuracy.

2 State of the Art

Since the 2-D DWT can quickly provide an approximation for an image, it is possible that it can be used as a replacement for max and mean pooling. Previous work has shown that the Haar wavelet pooling can act on a suitable replacement for conventional pooling methods(Li, Williams 2018), where utilizing the 2nd Order 2-D DWT provided higher classification accuracies in certain datasets. The equations for the DWT used by papers for forward propagation are given below:

$$W_{\text{approximation}}(a,p,q) = \frac{1}{\sqrt{PQ}} \sum_{x=0}^{P-1} \sum_{y=0}^{Q-1} f(x,y)\Psi_{a,p,q}(x,y)$$
$$W_{\text{detail}}(a,p,q) = \frac{1}{\sqrt{PQ}} \sum_{x=0}^{P-1} \sum_{y=0}^{Q-1} f(x,y)\varphi_{a,p,q}(x,y)$$

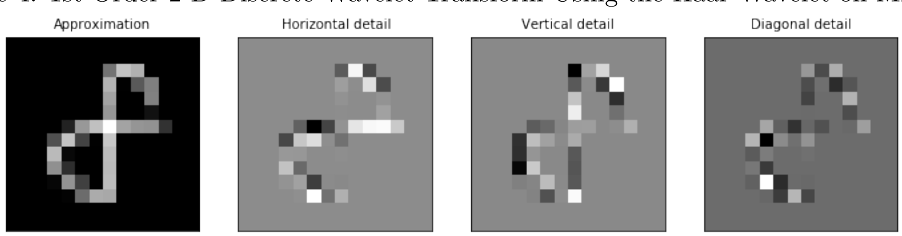
where $f(x,y)$ is an image with dimensions $P \times Q$, resolution level a, subband dimensions p and q, wavelet function $\Psi_{a,p,q}(x,y)$, and scaling function $\varphi_{a,p,q}(x,y)$. For backpropagation, each pooled value is up-sampled by 2 and then has the inverse Discrete Wavelet Transform applied on it to obtain the original kernel.

While many of the papers we read about this topic showed some levels of success with wavelet pooling, they only choose to pool with one or two wavelets and typically utilized the Haar Wavelet, which was implemented as a pooling step due to the wavelet's overall simplicity (Rosetto, Zhou 2019). As discussed in our methodology, we wanted to explore the use of different wavelets to use in our pooling step, as each wavelet can be interpreted differently in an image and could possible interpret an image better than another wavelet.

3 Methodology

For this project, we used the MNIST dataset originally and expanded to the CIFAR-10 dataset to explore color images afterwards.

Figure 4: 1st Order 2-D Discrete Wavelet Transform Using the Haar Wavelet on MNIST



We built a Convolutional Neural Network (CNN) framework with two convolutional layers, followed by a pooling step and then two fully connected hidden layers. In-between, we also implemented a dropout step where we dropped 25% and 50% of the nodes randomly in order to prevent overfitting. While we had originally planned to try to recreate Neural Networks in the papers that we had read, we decided to stick with simpler neural network as our goal was to compare the relative performance of wavelet pooling and max pooling without necessarily maximizing absolute accuracy.

We chose to use the Daubechies 2, Biorthogonal 1.1, and Coiflet 1 wavelets in addition to the Haar wavelet for our pooling step as popular wavelets for other applications. The Daubechies wavelet and Coiflets have vanishing moments that make them more complex and potentially better at representing complex signals with the tradeoff of longer support. Biorthogonal wavelets are invertible but not necessarily orthogonal.

With the data that results, we attempted to draw conclusions about whether wavelet pooling is worthwhile for our applications.

We tackled this question in two parts:

1. Experiment with different wavelets in the pooling step of our CNN.

MNIST: Dataset of handwritten digits with 60,000 training items and 10,000 test items

CIFAR: 60,000 32×32 images labelled into 10 classes.

2. Experiment with different configurations of Neural Network runs with wavelet pooling versus max pooling.

MNIST: Reversed image polarity

CIFAR: Small training set (5% of original)

The experiments we performed are an extension past our original proposal and a reaction to the results from our initial wavelet pooling results compared to max pooling which showed comparable performance. We were interested in whether we could distinguish them through manipulating the training data.

Firstly, suppose wavelet pooling does retain more details through the DWT than max pooling. However, test accuracy may still be comparable if the relative weakness of the feature is compensated for by a large training set such that the subsequent fully connected layers can interpret the right class.

To test this, we tried dramatically shrinking the CIFAR training set to 5% of the original training size. If our hypothesis is correct and wavelet pooling leads to smarter features through less loss of detail, wavelet pooling performance may be less degraded by the significantly reduced training set.

Second, max pooling may have an advantage specific to the design of the MNIST dataset. From Figure 2, we can see that max pooling is good at retaining the detail of the white line on the black background because the pooling steps keep the white pixels as bright spots. The MNIST dataset uses exactly that structure, with white digits on a black background. However, max pooling loses detail when interpreting the signal of a black line on a white background.

Therefore, to test this, we purposely reversed the MNIST training and test set's image coloration, making it a black digit on a white background. We tried reversing the entire training and test set first as well as flipping only half of the training an test images and leaving the other half the same as before.

4 Results

The first stage of our product was building a simple neural network and then trying different wavelets in our pooling step. Our results for the MNIST and CIFAR datasets are as follows:

Table 1: MNIST Test Accuracy

Pooling Algorithm	Test Accuracy
Max Pooling	97.28 %
Haar	98.41%
Daubechies 2	98.47%
Biorthogonal 1.1	98.25%
Coiflet 1	98.33%

As can be seen in Table 1, wavelet pooling performs slightly better than max pooling for the MNIST dataset. We thought that the relative simplicity of the MNIST classification problem might be explaining the low difference in test accuracy. Therefore, that motivated us to try and see if we could expand the difference in testing accuracy by trying a more complicated dataset like CIFAR.

However, as can be seen in Table 2, we found that wavelet pooling is still comparable to but does not outperform max pooling in CIFAR. Therefore, based on trying wavelet versus max pooling on two separate

Table 2: CIFAR Test Accuracy

Pooling Algorithm	Test Accuracy
Max Pooling	62.15%
Haar	61.39%
Daubechies 2	60.94%
Biorthogonal 1.1	61.13%
Coiflet 1	59.90%

datasets, our conclusion from this data was that wavelet pooling does not result in superior test accuracy compared to more conventional max pooling techniques.

Next, as mentioned above, we tried a few small experiments to get a better idea of whether there are some circumstances when wavelet pooling methods are superior. Additionally, these experiments under various pre-conditions help validate our initial results to see if they make sense.

For the CIFAR dataset, we tried shrinking the training dataset to 5% of its original size. Our hypothesis was that wavelet pooling features encoded more information and thus required less training data for accurate classification. However, the resulting test accuracy for max pooling was 37.7% while the resulting test accuracy for the Haar wavelet pooling was 37.9%, so this hypothesis was not supported by the results of our experiment.

For the MNIST dataset, we tried reversing the polarity of either some or all of the training and test set. We found that regardless of whether the whole training and test sets were flipped or whether only a portion of them were, max pooling and wavelet pooling continued to perform comparably. When half of the training and test set images had their polarities reversed, max pooling had 97.8% test accuracy. Haar wavelet pooling had 97.83% test accuracy. Therefore, we also did not find evidence that the structure of the MNIST dataset favored the max pooling approach relative to wavelet pooling.

5 Discussion

Our initial experiments with the MNIST dataset showed that the 2-D DWT was better as an approximation method in pooling in comparison to max pooling for our neural network. However, in expanding our neural network to handle more complicated datasets such as CIFAR-10, we were unable to obtain similar results with max pooling doing better overall. Similarly, our experiments did not lead to a distinguishment between wavelet pooling and max pooling test accuracies.

As noted in the results above, the accuracies obtained for the CIFAR-10 dataset were far from optimal. A next step we could attempt would be to use a neural network with a more complex architecture, such as ResNet(He, Zhang, et al. 2015), and replace the max pooling step with wavelet pooling to see if that will raise the accuracy of a more complex net–in essence, does wavelet pooling provide marginal gains when the model is more sophisticated? Other methods that we wish to explore is combining the use of different wavelets in a pooling step, where two approximations with two different wavelets are combined to approximate the image (Ferra, Agular, et al. 2018).

Another aspect that we wish to explore in the future is higher order wavelet decomposition on larger kernels as an approximation method for each sub-image. Many of the papers we read utilized the second order decomposition, and while we explored the first order decomposition of a 2×2 kernel, it is possible to expand each kernel to a larger size with different stride values for a higher order decomposition leading to a better approximation for that kernel for higher classification accuracy.

6 References

1. Strang. “Wavelet Transforms versus Fourier Transforms.” *American Mathematical Society*, 1993.

2. Tan, Khon, Mok, “Best Wavelet Function Identification System for ECG signal denoise applications.” *IEEE*, 2007.

3. Goldberg, Pivovarov, et. al. “Application of Wavelet Compression to Digitized Radiographs.” *Computers in Radiology*, 1994.

4. Ferra, Agular, et al. “Multiple Wavelet Pooling for CNNs”, *European Conference on Computer Vision* 2018.

5. He, Zhang, et al. “Deep Residual Learning for Image Recognition.” *Microsoft Research*, 2015.

6. Li, Williams. “Wavelet Pooling for Convolutional Neural Networks.” *ICLR*, 2018.

7. Rosetto, Zhou. “Improving Classification with CNNs using Wavelet Pooling with Nesterov-Accelerated Adam” *11th International Conference on Bioinformatics and Computational Biology*, 2019 pg. 84-93.