Adding Quaternion Representations to Attention Networks for Classification*

Nazmul Shahadat and Anthony S. Maida

School of Computing and Informatics
University of Louisiana at Lafayette,
Lafayette LA 70504, USA
nazmul.ruet@gmail.com
maida@louisiana.edu
10/03/2021

Abstract. This paper introduces a novel modification to axial-attention networks to improve their image classification accuracy. The modification involves supplementing axial-attention modules with quaternion input representations to improve image classification accuracy. We chose axial-attention networks because they factor 2D attention operations into two consecutive 1D operations (similar to separable convolution) and are thus less resource intensive than non-axial attention networks. We chose a quaternion encoder because of they share weights across four real-valued input channels and the weight-sharing has been shown to produce a more interlinked/interwoven output representation. We hypothesize that an attention module can be more effective using these interlinked representations as input. Our experiments support this hypothesis as reflected in the improved classification accuracy compared to standard axial-attention networks. We think this happens because the attention modules have better input representations to work with.

Keywords: quaternion networks · axial-attention networks · representation learning · weight sharing.

1 Introduction

The past decade has witnessed drastic improvements in performance of computer vision applications such as image classification and recognition. Until quite recently, most improvements were based on using deep convolutional neural networks (CNNs). Although CNNs provide excellent inductive priors for image classification, they have trouble taking into account the importance of long-distance relationships between image features that may be relevant (e.g., relative configuration of eyes, nose, mouth for face recognition).

Encouraged by the dramatic success of self-attention transformer networks for sequence transduction in natural language processing (NLP) [13], which replace recurrence with self-attention, researchers explored the idea of augmenting existing CNNs with transformers as a means to learn relevant long-distance relationships between image features. It was surprisingly found that standalone stacked attention networks that did not incorporate CNNs at all could outperform both pure CNNs and hybrids if given enough training data and computing resources [10]. Axial-attention networks [14] were introduced to reduce the intense computational requirements of more direct attention models.

In a separate line of research, it was found that quaternion-based convolution modules could create better output representations for a color reconstruction task [12]. This was due to using a new type of weight sharing found in the quaternion convolution operation that did a better job at discovering inter-channel relationships for 4-channel color input images. We hypothesize that this effect may apply to a broader range of multi-dimensional input types, not just color.

Our experiments test this hypothesis using axial-attention networks for image classification. Our experiments confirm that when quaternion input representations are supplied to the axial-attention modules, classification performance improves as compared to standard axial-attention networks. We believe this happens because the attention modules have better input representations to work with.

* Partially supported by an Alfred and Helen M. Lamson endowed chair award.
2 Rationale for the Proposed Method

The main hypothesis of this paper is that feature map representations that are used as input to attention modules can be modified to improve their effectiveness. Our approach is to use the output of quaternion modules to provide improved input representations to the attention modules. The rationale follows.

Parchollet [8] showed that a quaternion-valued, auto-encoder can be trained to reconstruct color from grayscale input images whereas a real-valued autoencoder cannot. In contrast, a real-valued auto-encoder trained on the same data does not learn to do this reconstruction (more detail in Section 3.1). Thus, a trained quaternion-valued layer generates a richer representation than a trained real-valued layer because it allows implicit relationships between the color channel data to be captured and these relationships cannot be captured in a comparable real-valued auto-encoder [8].

Parchollet [8] attributes this functionality to a weight-sharing property found in the Hamiltonian product (quaternion multiplication) that is used to implement convolution. The reconstruction does not occur using real-valued networks which lack this weight-sharing property. The representation produced by the quaternion network captures more information about the interrelationships between the color input channels, and in this sense, we can say that it is a richer *interwoven or interlinked representation* of the information contained in the input channels.

We hypothesize that this improved ability to capture relationships between input channels is likely to apply to other multidimensional inputs besides color [3]. It may depend on whether the input data happens to contain *interwoven cross-channel relationships* of this kind to capture. In the quaternion context, it may apply to any model with three or four input dimensions (although this is still an empirical question). In the remainder of this paper, we use the term *interwoven/interlinked representation* or *interwoven feature map* to refer to the output of a quaternion layer.

3 Background and Related Work

3.1 Quaternion Convolution

A useful introduction to quaternion algebra and neural networks appears in [9].

The deep quaternion CNN was introduced in [2] as a hypercomplex extension to complex CNNs [12]. Deep quaternion CNNs were developed after deep complex networks [12]. The paper [12] extended the principles of convolution, batch normalization, and weight initialization from real-valued networks to complex-valued networks. The paper [2], in turn, extended these principles to quaternion-valued networks allowing the construction of deep quaternion-valued networks. Hamilton in 1833 attempted to extend multiplication of complex pairs to triples (a, b, c) but was initially unsuccessful. In 1843, he successfully showed how to multiply 4D objects at the cost of losing the commutativity property. These numbers became known as quaternions, the first of the hyper-complex numbers. For implementation purposes, quaternion calculations can be decomposed into operations on 4-tuples of real numbers and quaternion operations can be implemented on these 4-tuples. The take home message from this is that a quaternion convolution module must accept four channels of input. A quaternion layer can accept more than four input channels, say \( m \), as long as \( m \) is a multiple of four. In this case, the layer must hold \( m/4 \) separate quaternion convolution modules, each with their own weight sets.

The paper [2] showed that quaternion CNNs embedded in a ResNet [5] architecture could converge more quickly than real-valued and complex-valued CNNs, and with a reduced parameter count, while matching accuracy on CIFAR-10 and CIFAR-100 classification problems.

The paper [8] analyzed the Hamilton product, which underlies the quaternion convolution, to identify the mechanism for improved performance in quaternion models. Let us denote the Hamilton product as

\[
O_q = I_q \odot W_q, \tag{1}
\]

where the subscripts \( q \) indicate quaternion numbers. When decomposed into real-valued 4-tuples, this can be viewed as a linear mapping from a neural layer of four units, representing \( I_q \), to another layer of four units representing \( O_q \). Instead of using 16 independent weights to connect the layers, the four weights in \( W_q \) are repeatedly substituted within the 16 weights, so the mapping only uses four independent weights. This
weight sharing forces the model to learn cross-channel interrelationships in the data that are not normally
learned in fully connected models. The paper [8] confirmed this experimentally by comparing the ability of
a real-valued autoencoder with a quaternion-valued autoencoder on the task of reconstructing color images.
Both autoencoders were trained to reconstruct color images from grayscale input images by giving the colored
versions of the images as desired output. The quaternion network learned to accurately reconstruct the colors
in unseen images but the real-valued network did not.

More recently, the paper [314] showed that the properties of the quaternion convolution were due en-
tirely to weight sharing and had no dependence whatsoever on the associated quaternion algebra they were
embedded in. Specifically, implementation of the quaternion convolution does not use algebraic operations
unique to quaternions. When detached from the quaternion algebra, the constraint of using exactly four
dimensions is removed and a variant of the quaternion convolution can be used for multidimensional input
of any dimension size. These vector-map convolutions can be applied to data of any dimension. All that is
needed is a way to choose the vector map coefficients for the higher-dimensional multiplications (which was
given in [314]). More recently [15] reported a similar idea for the case of fully connected feedforward networks,
testing the model using LSTMs and Transformer Networks on various natural language understanding tasks.
There model used a learning algorithm to determine the multiplication coefficients.

The research presented in the current paper does not yet explore vector-map convolutions and restricts
itself to 1x1 quaternion convolutions. Just as 1x1 convolutions in a real-valued network can be represented
as a linear mapping from one layer in a fully connected network to another, the 1x1 quaternion convolution
can be represented by the Hamilton product used in Expression 1. The quaternion convolutions in our model
are 1x1, so they can be interpreted as fully connected layers with shared weights.

3.2 Axial-Attention Networks

As stated in the introduction, standalone attention networks consisting of stacked attention modules can
learn to outperform deep CNNs and hybrid CNN/attention models for image classification. Their advantage
stems from their ability to detect similarities, or affinities, between pixels that have large spatial separation in
the image. The drawback of directly using self-attention is that it is impractically computationally expensive,
consuming $O(N^2)$ resources for a sequence of length $N$ (using a flattened pixel set) and using a global window
to compare any pair of pixels in the image. For a 2D image of height, $h$, and width, $w$, $N = hw$, so the cost
is $O((hw)^2) = O(h^2w^2)$ to detect similarities for any pair of pixels in the image [6][14].

Axial-attention networks reduce the cost of computing attention. They were introduced in [6] for generative
modeling in auto-regressive models and use the assumption that images are approximately square so that
$h$ and $w$ are both much less than the total pixel count, $hw$. For simplicity, assume a square 2D image
where $h = w$, so $w^2 = N$. Axial attention only operates on one dimension at a time. It is first applied to, say,
the $h$ axis and then to the $w$ axis. When applied to the $h$ axis, then $h = w$ attention calculations are applied
to a 1D region of length $h$. Axial attention applied to the columns, $h$, performs $w$ self-attention operations
on each column whose total cost is $O(h \cdot h^2) = O(\sqrt{N} \cdot N)$. This bound is the same for using an axial column
module followed by an axial row module.

The paper [14] incorporated axial attention into a ResNet architecture [5] to develop a model called
Axial-ResNet, which reduced the computational requirements (described above) of the original standalone
attention networks [10][7] used for image classification. The conversion from ResNet to Axial-ResNet was
based entirely on modifying the bottleneck blocks within ResNet. Figure 1 (left) shows the bottleneck block
used in the original convolution-based ResNet. Figure 1 (middle) shows the axial-bottleneck block used in
Axial-ResNet. The 3x3 2D convolution used in the original ResNet is replaced by an axial attention module
(consisting of two 1D layers) that processes the $h$ axis followed by the $w$ axis.

4 Our Model: Axial-Attention with Quaternion Input Representations

Our model is the axial-attention model used in [14] and described above but modified to supply quaternion
input representations to the axial attention block.

Figure 1 (right) shows our modification to the axial-bottleneck block. It inserts a bank of 1x1 quaternion
conv2d modules in front of the axial-attention module. The number of input channels to the quaternion
bank is constrained to be a multiple of four. This is shown in Figure 1 (right) where the output channels of the top 1x1 conv2d module are split into groups of four. One quaternion 2D convolution is applied to each group of four channels. Each quaternion convolution accepts four channels of input and produces four channels of output. The set of output channels is merged (stacked) so that the number of input channels to the axial-attention module is unchanged from the original axial-attention block. In all other ways, the structure of our quaternion-modified model is identical to the axial-ResNet model.

To make the above clearer, if we have 64 channels of input to the quaternion bank, then the bank needs to hold 16 1x1 quaternion conv2d modules, each with their own weight sets. Four real-valued channels are mapped to each module. Thus, the weight-sharing is compartmentalized to groups of four input channels.

Although our model is created by modifying the axial-bottleneck module, it is also useful to view it as supplying a quaternion front end to the axial attention module that is used in the bottleneck module. The purpose of the front end is to generate potentially more useful interwoven/interlinked input representations for use by the axial-attention modules.

5 Experiment

We compare four models on a subset of the ImageNet dataset, called ImageNet300k, which we created (explained below). We use this dataset because we did not have the computing power to conduct simulations using the full ImageNet dataset [1].

5.1 Method

The models we compare are: the standard convolution-based ResNet [5], the quaternion-valued convolution-based ResNet [2], the axial-ResNet [14], and our novel quaternion-enhanced axial-ResNet. The main objective is to see if the representations generated by the quaternion front end improve classification performance of the axial attention model. These four models are depicted in Table 1. We used a 26-layer version of the models and a 50-layer version of the models. Table 1 shows the 26-layer versions. The bracketed expressions show the bottleneck blocks with the operations used and the number of output channels for each stage. The numbers to the right are the block multipliers. The multipliers shown in the figure are “[1, 2, 4, 1],” which gives 26 trainable layers. For the 50-layer version of the network uses the block multipliers “[3, 4, 6, 3].” For our quaternion enhanced model, these layer counts do not include the quaternion modules. If the quaternion
modules are counted then the layer counts for our model are 33 layers and 66 layers, respectively. We count the two-1D-layer axial-attention module as one layer because two 1D layers are equivalent to one 2D layer. The model can be found at https://github.com/nazmul729/representationalAxialNet.git.

Because of hardware limitations, our experiment conducts image classification on a subset of the ImageNet dataset [11]. Specifically, it took over two hours to train one of the axial-ResNet models for one epoch. This smaller dataset which requires lower computing resources, called ImageNet300k, uses the same 1,000 image categories as the original ImageNet [1]. The full ImageNet dataset has 1.28 million training images and 50,000 validation images. Our smaller dataset, which we call ImageNet300k, uses the same 1,000 image categories as the original ImageNet [1]. The full ImageNet dataset has 1.28 million training images and 50,000 validation images. Although this dataset is smaller, it still allowed us to train our 26-layer networks without overfitting. Overfitting was assessed by examining performance on the validation dataset in comparison to the training set. Validation dataset was the same as that used in the original ImageNet dataset.

All network models were trained using the same optimizer and hyperparameters. All networks were trained for 150 epochs using stochastic gradient descent optimization with momentum set to 0.9 and the learning rate which is warmed up linearly from $\epsilon$ near zero to 0.1 for 10 epochs [14]. The learning rate was then cut by a factor of 10 at epochs 20, 40, and 70. We adopt the same training protocol as [14] except for batch size. Our batch size is limited to ten because of memory limitations. Also, we used batch normalization with 0.00009 weight decay. For attention models, the number of attention heads is fixed to eight in all attention layers [14].

### 5.2 Results Analysis

The main results are seen in Table 2. The top half of the table shows the results for the 26-layer models (33 layers for the quaternion axial-attention model). The bottom half shows the results for the 50-layer models
Table 2. Image classification performance on the ImageNet300k dataset for 26 and 50-layer architectures. We include Top-1 training and validation accuracies.

| Architecture                  | Layers | Training Top-1 Acc. | Params | Validation Top-1 Acc. | Inference time |
|-------------------------------|--------|---------------------|--------|-----------------------|---------------|
| ResNet                        | 26     | 57.0                | 13.6M  | 45.48                 | 8.86 msec     |
| Quaternion ResNet             | 26     | 64.1                | 15.1M  | 50.09                 | 25.32 msec    |
| Axial-Attention               | 26     | 61.0                | 5.7M   | 54.79                 | 27.94 msec    |
| Quat Axial-Attention (ours)   | 33     | 78.2                | 6.0M   | 62.30                 | 31.75 msec    |
| ResNet                        | 50     | 65.8                | 25.5M  | 50.92                 | 14.70 msec    |
| Quaternion ResNet             | 50     | 73.4                | 27.6M  | 49.69                 | 50.01 msec    |
| Axial-Attention               | 50     | 63.6                | 11.5M  | 55.57                 | 52.35 msec    |
| Quat Axial-Attention (ours)   | 66     | 72.6                | 11.9M  | 59.71                 | 58.41 msec    |

The most important comparison is between axial-ResNet and quaternion-enhanced axial-ResNet because this directly shows the effect of the quaternion-generated interwoven/interlinked representations. There are two such comparisons, one for the 26/33 layer models and the other for the 50/66 layer models. In both cases, the quaternion enhanced versions of axial-ResNet produced higher classification accuracy performance. This was true for both the training and validation data. This supports our main hypothesis that quaternion modules can produce more usable interlinked/interwoven representations. Of course, there is the alternative explanation that the quaternion enhanced axial-ResNet had more layers and this is the reason for the better performance. This is addressed in Detailed Study.

Another result is that both of the axial-ResNet architectures give higher classification accuracy for the validation set than either of the convolution-based ResNets. This is true for both the 26-layer (33-layer) and 50-layer (66-layer) versions. This is noteworthy given that both axial-ResNet architectures use a much smaller parameter budget than the convolution-based architectures. However, the trend does not occur for the classification accuracy on the training set.

Finally, it is surprising that the 33-layer quaternion-enhanced axial-ResNet gives better classification accuracy performance than the 66-layer version. This is true for both the training and validation data. We have no explanation for this. If this were only true for the validation data, then overfitting would be a possible explanation.

6 Detailed Study

In the above experiment at table 2, adding the quaternion front-end to the axial attention module increased the number of layers from 26 to 33 for the small network and from 50 to 66 for the large network. This offered an alternative explanation for the improved classification results. The accuracy improvement might have been caused by the increased number of layers and not the added quaternion bank. This analysis tests the alternative explanation.

6.1 Method

This experiment increases the layer count of the the standard convolution-based ResNet [5], the quaternion-valued convolution-based ResNet [2], the axial-ResNet [14] from 26 to 35 layers. This was done by using a block multiplier of [2, 3, 4, 2] for these models. Although this did not give us exactly 33 layers, it preserved the bottleneck structure of the original design and enhances comparability.

6.2 Experimental Results

Table 3 shows the parameter count and accuracy for these 35-layer models. The most important comparison is the 35-layer Axial-Attention Network in Table 3 with the 33-layer Quaternion Axial-Attention Network in...
| Architecture     | Layers | Training Top-1 Acc. | Params | Validation Top-1 Acc. |
|------------------|--------|---------------------|--------|-----------------------|
| ResNet           | 35     | 63.8                | 18.5M  | 48.99                 |
| Quaternion ResNet| 35     | 70.9                | 20.5M  | 48.11                 |
| Axial-Attention  | 35     | 73.6                | 8.4M   | 60.49                 |

Table 3. Image classification performance on the ImageNet300k dataset for ResNet-35 architectures. We include Top-1 training and validation accuracies.

Table 2 Now the layer count for the non-quaternion version is slightly larger than that for the quaternion version but the quaternion version still gives slightly better performance on both the training and validation accuracy. The increased layer count did improve the performance of the 35-layer Axial Attention Network but not enough to overcome the quaternion version.

7 Conclusions and Future Work

The main conclusion of our work is that using quaternion convolutions as the front end to axial-attention-blocks in Axial-ResNets improves classification accuracy on the ImageNet300k task. This supports our hypothesis that the quaternion convolution bank provides better representations to be used by the axial-attention module.

A second conclusion is that both axial-ResNet architectures have better classification accuracy than comparable convolution-based ResNet architectures, at least for this data set.

Future work includes examining the performance of the quaternion-enhanced Axial-ResNet on other datasets. Most important would be to see how well it performs when trained on the full ImageNet dataset. This would allow us to assess how close its performance is to state-of-the-art.

Other future work involves introducing more fine-grained control over the amount of weight sharing for the quaternion front end. As stated earlier, the quaternion weight sharing is compartmentalized to groups of four input channels. If we were to use vector-map convolutions [3], we could have more fine-grained control over the amount of weight-sharing because the 4-channel constraint is not required allowing us to use an arbitrary number of dimensions. This would let us vary the amount of weight-sharing compartmentalization for different applications. It is conceivable that different applications would benefit from different amounts of weight sharing as has already been demonstrated in natural language understanding [15].

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. IEEE (2009)
2. Gaudet, C., Maida, A.S.: Deep quaternion networks. In: Intl. Joint Conf. on Neural Networks (IJCNN). IEEE (July 2018)
3. Gaudet, C.J., Maida, A.S.: Generalizing complex/hyper-complex convolutions to vector map convolutions. CoRR abs/2009.04083 (2020), https://arxiv.org/abs/2009.04083
4. Gaudet, C.J., Maida, A.S.: Removing dimensional restrictions on complex/hypercomplex networks. In: 2021 IEEE International Conference on Image Processing. IEEE Signal Processing Society (September 2021)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
6. Ho, J., Kalchbrenner, N., Weissborn, D., Salimans, T.: Axial attention in multidimensional transformers (2019)
7. Hu, H., Zhang, Z., Xie, Z., Lin, S.: Local relation networks for image recognition. In: Internation Conference on Computer Vision (ICCV) (2019)
8. Parcollet, T., Morchid, M., Linarés, G.: Quaternion convolutional networks for heterogenous image processing. In: IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 8514–8518 (2019)
9. Parcollet, T., Morchid, M., Linarés, G.: A survey of quaternion neural networks. Artificial Intelligence Review 53(2957-2982) (2020)
10. Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J.: Stand-alone self-attention in vision models. CoRR abs/1906.05909 (2019), http://arxiv.org/abs/1906.05909
11. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: ImageNet large scale visual recognition challenge. International Journal of Computer Vision 115(3), 211–252 (2015)

12. Trabelsi, C., Bilaniuk, O., Zhang, Y., Serdyuk, D., Subramanian, S., Santos, J.F., Mehri, S., Rostamzadeh, N., Bengio, Y., Pal, C.J.: Deep complex networks (2018)

13. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NIPS Proceedings (2017)

14. Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A., Chen, L.C.: Axial-deeplab: Stand-alone axial-attention for panoptic segmentation (2020)

15. Zhang, A., Tay, Y., Zhang, S., Chan, A., Luu, A.T., Hui, S.C., Fu, J.: Beyond fully-connected layers with quaternions: Parameterization of hypercomplex multiplications with 1/n parameters. CoRR abs/2102.08597 (2021). https://arxiv.org/abs/2102.08597