SURF Final Report

Zack Dugue

September 2023

1 Abstract

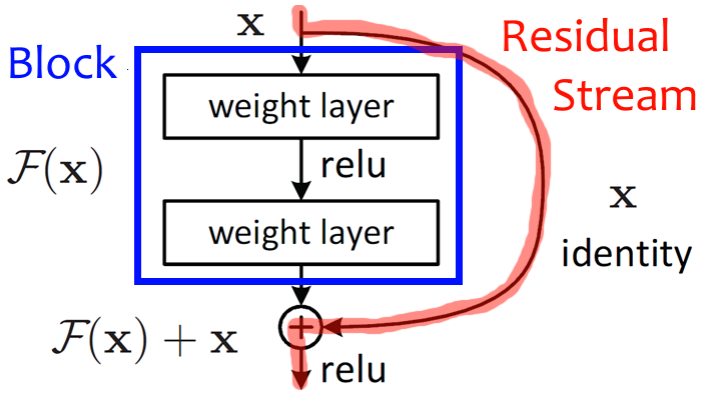
Skip Connections have been ubiquitous in deep neural networks since their inception in the ResNet architecture (He 2015). However, their relatively simple structure means that they have limited expressivity when compared to the standard feed forward approach that preceded it. we propose an architecture which uses Query Integrated Memory Interfacing Attention (QIMIA) as a middle ground between fully connected networks, and standard skip connections. QIMIA is an attention-based approach which uses learned queries to attend to the outputs of prior layers in order to construct the inputs to each layer. We evaluate QIMIA architectures against skip connection based architectures on both vision and language tasks, and find inferior performance when compared to standard skip connection Architectures.

2 Background

Deep Learning is a field of machine learning which uses Deep Neural Networks to learn from data how to do certain tasks. Despite the name, for a long time in Deep Learning research deeper didn't mean better. In fact, at this time, adding more layers often degraded performance[3].

The solution to this problem was the Residual Neural Network, AKA "Resnet" [2]. Rather than having the input of the next block be the output of the last block, Resnets use something called a "residual stream" to control the flow of information in the network. Every block's output is added to this residual stream (via something called a "skip connection") as , and every block's input is the value of the residual stream at that block (rather than simply the output of the prior block) (fig figure 2). At the end of the network some final block processes the value of this residual stream and then generates the output. Resnet architectures tend to smooth the optimization space of the neural network, allowing efficient learning of deeper models. Virtually all networks deeper than 3 hidden layers use a Resnet architecture.

Figure 1: A diagram representing the structure of a Residual Neural Network from [2], modified for clarity.



A drawback to this approach is that the residual stream must contain all the information relevant to the latter blocks of the neural network. For iterative tasks, this is not much of an issue. For example, imagine the steps necessary for multiplying out a factorial. You only need to remember what the running product is, and the number remaining left to multiply. But for tasks that involve parallel steps, like integration by parts, this means that you have to remember information from prior completed steps that are irrelevant to the current step, but necessary for a future step. In the case of a human brain, this might clutter up the short term memory, and in the case of a Resnet, this will clutter up the finite amount of information which can be held in the residual stream.

A solution to this problem would allow would allow the neural network to effectively "chunk up the problem", by analyzing relevant information while ignoring information produced by other, irrelevant, parallel steps.

3 The Architecture

Skip Connection Architectures have enabled much deeper neural networks than their generic Feed Forward Counter Parts. But this represents a strong inductive bias, since every block takes as its input the sum of the outputs of all other blocks, without concern for the extent to which those prior blocks are "relevant" to the current blocks task. This can cause the overall neural network to be less expressive. For example in the paper [8], they find much stronger performance with blocks that include several Attention, Feed Forward, and Mixture of Expert, based layers chained together into a single block. In this case then, the Neural Network improves performance by using fewer skip connections than are in standard architectures.

What is needed is a more expressive way to aggregate information from prior blocks. Our proposed solution here is Query Integrated Memory Interfacing Attention (QIMIA). The goal of QIMIA is an architecture that allows any block to "look up" the outputs of prior blocks using attention [7]. This solves our problem by creating an accessible "memory", such that current block only needs to take as input the relevant outputs of prior blocks, thus reducing the information clutter from irrelevant prior steps.

$$A_l = \sigma(\mathbf{q}\mathbf{K}^T) * \mathbf{V} = \frac{\sum_{i=0}^{l-1} \exp(\mathbf{q} \cdot \mathbf{k}_i)\mathbf{v}_i}{\sum_{i=0}^{l-1} \exp(\mathbf{q} \cdot \mathbf{k}_i)} \tag{1}$$

Equation 1: *This equation shows represents the input to a block via the attention mechanism at layer l. Where σ is the softmax operation, q is the learned query at block l, k_i is the output key for block i and v_i is the output value for block i.*

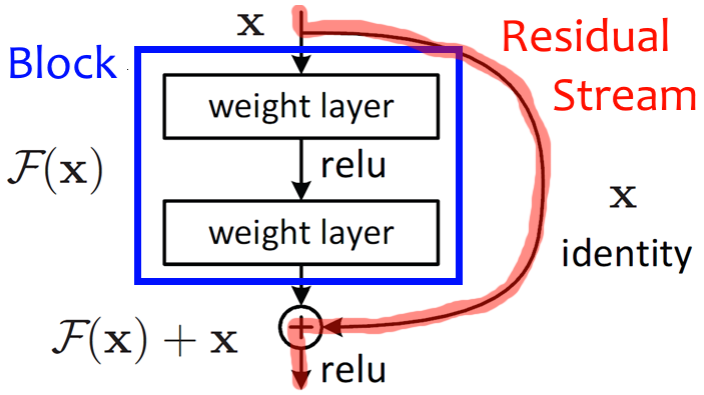
In this implementation, every block outputs a key and a value, and every block learns a query. This query is then used to do attention on the outputs of prior blocks. This allows each block to signal in their output key what they think their output value will be useful for. Each block also learns a query to search for the most relevant outputs of prior blocks. The queries are directly learned, and the keys and values of prior blocks acts as a sort of "memory", hence the name.

4 Background

Deep Learning is a field of machine learning which uses Deep Neural Networks to learn from data how to do certain tasks. Despite the name, for a long time in Deep Learning research deeper didn't mean better. In fact, at this time, adding more layers often degraded performance[3].

The solution to this problem was the Residual Neural Network, AKA "Resnet" [2]. Rather than having the input of the next block be the output of the last block, Resnets use something called a "residual stream" to control the flow of information in the network. Every block's output is added to this residual stream (via something called a "skip connection") as , and every block's input is the value of the residual stream at that block (rather than simply the output of the prior block) (fig 2). At the end of the network some final block processes the value of this residual stream and then generates the output. Resnet architectures tend to smooth the optimization space of the neural network, allowing efficient learning of deeper models. Virtually all networks deeper than 3 hidden layers use a Resnet architecture.

Figure 2: A diagram representing the structure of a Residual Neural Network from [2], modified for clarity.



A drawback to this approach is that the residual stream must contain all the information relevant to the latter blocks of the neural network. For iterative tasks, this is not much of an issue. For example, imagine the steps necessary for multiplying out a factorial. You only need to remember what the running product is, and the number remaining left to multiply. But for tasks that involve parallel steps, like integration by parts, this means that you have to remember information from prior completed steps that are irrelevant to the current step, but necessary for a future step. In the case of a human brain, this might clutter up the short term memory, and in the case of a Resnet, this will clutter up the finite amount of information which can be held in the residual stream.

A solution to this problem would allow would allow the neural network to effectively "chunk up the problem", by analyzing relevant information while ignoring information produced by other, irrelevant, parallel steps.

5 The Architecture

Skip Connection Architectures have enabled much deeper neural networks than their generic Feed Forward Counter Parts. But this represents a strong inductive bias, since every block takes as its input the sum of the outputs of all other blocks, without concern for the extent to which those prior blocks are "relevant" to the current blocks task. This can cause the overall neural network to be less expressive. For example in the paper [8], they find much stronger performance with blocks that include several Attention, Feed Forward, and Mixture of Expert, based layers chained together into a single block. In this case then, the Neural Network improves performance by using fewer skip connections than are in standard architectures.

What is needed is a more expressive way to aggregate information from prior blocks. Our proposed solution here is Query Integrated Memory Interfacing Attention (QIMIA). The goal of QIMIA is an architecture that allows any block to "look up" the outputs of prior blocks using attention. This solves our problem by creating an accessible "memory", such that current block only needs to take as input the relevant outputs of prior blocks, thus reducing the information clutter from irrelevant prior steps.

$$A_l = \sigma(\mathbf{q}\mathbf{K}^T) * \mathbf{V} = \frac{\sum_{i=0}^{l-1} \exp(\mathbf{q} \cdot \mathbf{k}_i)\mathbf{v}_i}{\sum_{i=0}^{l-1} \exp(\mathbf{q} \cdot \mathbf{k}_i)} \tag{2}$$

Equation 2: *This equation shows represents the input to a block via the attention mechanism at layer l. Where σ is the softmax operation, q is the learned query at block l, k_i is the output key for block i and v_i is the output value for block i.*

In this implementation, every block outputs a key and a value, and every block learns a query. This query is then used to do attention on the outputs of prior blocks. This allows each block to signal in their output key what they think their output value will be useful for. Each block also learns a query to search for the most relevant outputs of prior blocks. The queries are directly learned, and the keys and values of prior blocks acts as a sort of "memory", hence the name. Note that QIMIA, just like standard attention, can have multiple heads.

Due to the limitations of the architecture, all QIMIA models are transformer models. This is because the learned query for the whole blocks attends to each element (pixel in vision / token in NLP) of an input individually. For example, in a vision model, a learned QUERY for a block deep in the network, might attend strongly to the output of block 2, for one pixel, but for a neighboring pixel, it strongly attends to the output of block 3. Because of this, there aren't any obvious ways to "reduce" the number of keys / values from one block to another. So any methods which require down-sampling the outputs of a blocks, is incompatible with the QIMIA framework. This makes Transformers a natural fit.
Also, unless otherwise specified, all queries are initialized to be equal to the zero vector. This means that the model initializes in an identical state to a skip connection based architecture. This seems to greatly stabilize training over using randomly initialized queries.
Further implementation details can be found in the appendix.

Figure 3: A diagram of block l performing Query Integrated Memory Interfacing Attention, processing that input into a key and value, and then storing that key value pair in the set of key value pairs that serves as "memory".

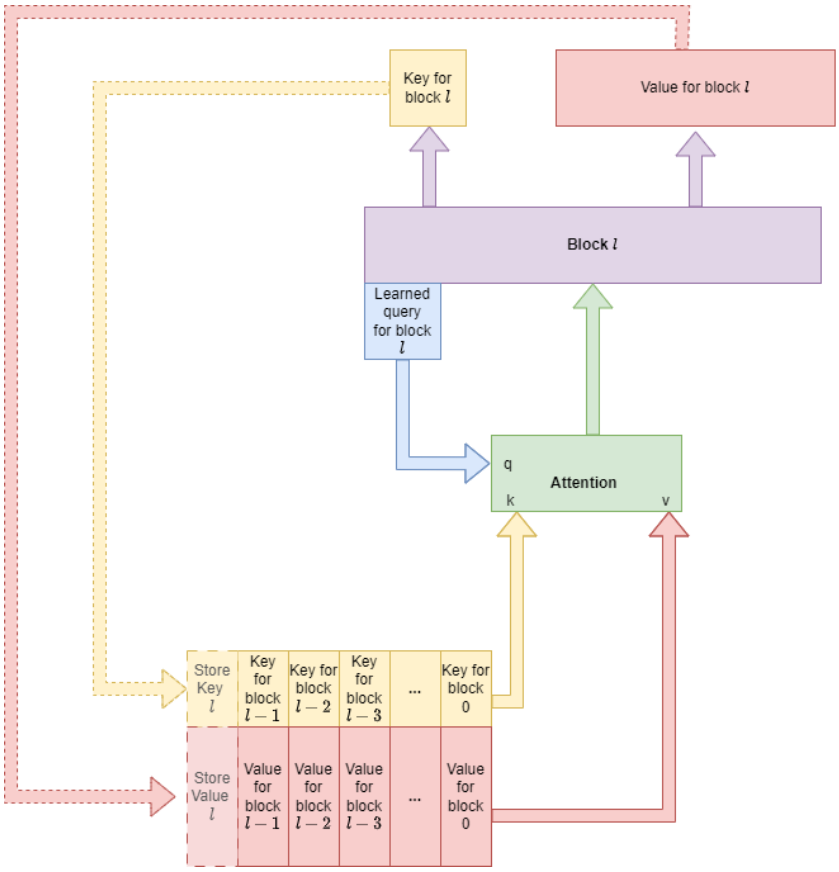
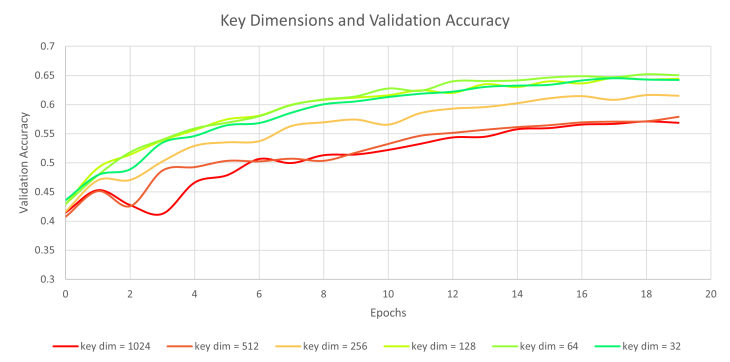


Figure 4: A graph of how the accuracy curves changed with the size of the key dimension.



6 Results

Out tests in the Vision Domain were primarily using the CIFAR10 dataset [cite CIFAR10 paper]. Initial experiments found that the QIMIA based model underperformed the skip connection based baseline. The QIMIA model introduces many hyper parameters however, and the effects of tuning these hyper parameters were explored to see if the gap could be closed.

In this first experiment we explored the difference in learning curves when changing the dimensionality of the keys/queries used in the QIMIA architecture (while holding the number of QIMIA Heads fixed). We found that larger key dimensionality resulted in a less accurate model overall (figure 4). This is counter intuitive given that larger key dimensional means more parameters overall. The "Entropy" metric is a metric which computes the average "entropy" of the QIMIA attention weight matrices of the model. Larger entropy means that blocks are attending equally to prior blocks, and are thus acting similarly to standard skip connection architectures. We find that models with smaller key dimensional tend to have lower entropy metrics (figure **??**, and are thus learning to imitate a skip connection (since the input is essentially an equally weighted sum of all prior outputs).

We then explored how altering the learning rate effects performance. We found that, lower learning rates did correlate with better performance (figure 6). But due to the zero initialization of queries, the low learning rate caused queries to be very small in magnitude, and so in practice the QIMIA model simply approximated a skip connection based architecture (figure 7).

We considered the possibility that learning rate warmup might be required for training. Early Transformer models used learning rate warmup in order to prevent the model from essentially over fitting on data points early in training and learning parameters that make learning unstable for future steps [1]. We found that learning rate warmup did not benefit performance at all. We also tried "Path Dropout" which is a regularization strategy sometimes used in very

Figure 5: A graph of how the entropy of the QIMIA model changed with the size of the key dimension.

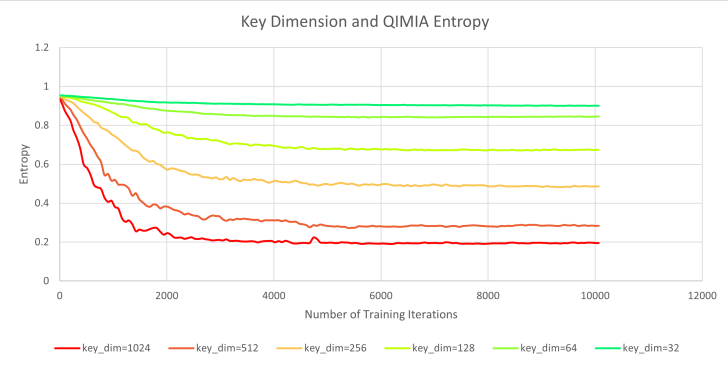


Figure 6: A graph of how the validation accuracy of the QIMIA model changed with the learning rate.

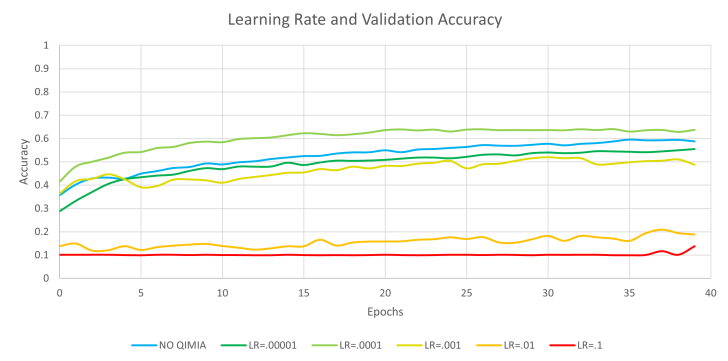
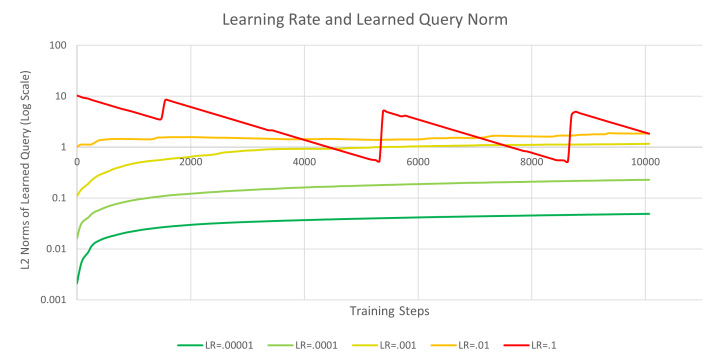


Figure 7: A graph of how the norm of the query of the QIMIA model changed with the learning rate as the model trained.



large transformer models that drops entire blocks from the models feed forward [6]. We implemented this by applying dropout to the attention weight matrix of the input to each block. The hope was that this would stabilize training by preventing any block from learning to depend too strongly on the output of one particular prior block. However we only found reduced performance when using this dropout.
Finally we explored some broader architectural changes including normalizing the keys to the unit vector, batch normalizing the values, and replacing the dot product attention of QIMIA with L2 attention. With Key Normalization, we normalize the keys such that each key has L2 norm of 1. This has the effect of stabilizing the gradient signals passing through the keys. We also test using a batch norm on the output of all the values (and getting rid of the Layer norm at the beginning of each block) [4]. Finally we test using L2 attention instead of Dot Product Attention. L2 attention was tested because, unlike dot product attention, L2 attention is Lipchitz Continuous, and so should have more stable gradients [5]. Unfortunately none of these architectural changes meaningfully improved the QIMIA model performance.
Tests were also performed in the language domain on the Wikitext103 dataset, but it was found that the standard skip connection based architecture outperformed the QIMIA architecture.

7 Conclusion

In general the pattern that we found was that QIMIA architectures appeared to be unstable during training. QIMIA architectures only performed well in situations where it was essentially imitating skip connection based architecutre. While the QIMIA model is much more expressive, it seems that much like the feed forward networks of old, that degree of expression makes it tend to bounce around the optimization space explored by Stochastic Gradient Descent. My exploration of potential hyper parameters was not exhaustive, and it may be possible that there is some combination of these hyper parameters (or architectural changes) that would cause QIMIA to perform on par (or better) than skip connection based architectures, and this remains an area to be explored in future work.

8 References

References

[1] Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation, 2018.

[2] Kaiming He and Jian Sun. Convolutional neural networks at constrained time cost. *CoRR*, abs/1412.1710, 2014.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[5] Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention, 2021.

[6] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals, 2017.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[8] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning, 2019.

9 Acknowledgements

I want to thank my PI and Mentor, Dr. Georgia Gkioxari. I would also like to thank the California Institute of Technology's Student and Faculty Programs office for hosting SURF and giving me the opportunity to research here!

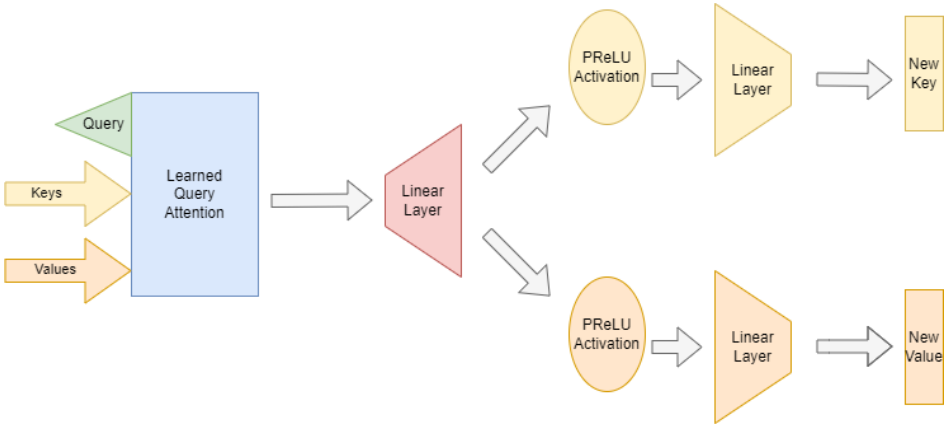


Figure 8: A graph of a feed forward QIMIA block. Note the two seperate paths for the keys and the values after the first linear layer.

10 Appendix

All QIMIA models tested used layer-normalization without an affine transform, and lacked the standard "Head Mixing Layer" that most attention modules typically contain. All of the CIFAR models were trained as vision transformers. With an embed dim of 256, a depth of 10, a path size of 8, and a hidden layer dimension for the feed forward layers of 1024. For the purposes of QIMIA the input embedding and the learned positional encoding are considered separate key value pairs, and thus later layers can attend to them individually.
Feed Forward blocks include a single shared Linear Layer that projects the input into the feed forward hidden dimension (figure 8). Then there are two splitting "paths". The Key Path and the Value Path. The Value path has a Parametric-ReLU activation, and then a linear layer which projects the hidden feed forward units back down into the embedding dimension. The key similarly has a Parametric-ReLU and a linear projection layer from the feed forward hidden dimension to the key dimension. Attention Blocks are similarly defined, but distinct from the standard transformer setup, they also contain Parametric-ReLU activations (figure 9).

Figure 9: A graph of a feed forward QIMIA block. Note the two seperate paths for the keys and the values after the Multihead Attention.