

Improving Weight Excitation for ConvNets and MLP Mixer

Ridwan Mahbub

180041230

Samaha Shafiq Anuva

180041137

Ifrad Towhid Khan

180041225



Department of Computer Science and Engineering
Islamic University of Technology
Organization of the Islamic Cooperation (OIC)

Dhaka, Bangladesh

May 19, 2023

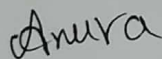
Declaration of Authorship

This is to certify that the work presented in this thesis, titled, “**Improving Weight Excitation for ConvNets and MLP Mixer**”, is the outcome of the investigation and research carried out by Ridwan Mahbub, Samiha Shafiq Anuva, Ifrad Towhid Khan, under the supervision of Professor Dr. Md. Hasanul Kabir, Lecturer Shahriar Ivan and Lecturer Md. Zahidul Islam. It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma, or other qualifications. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

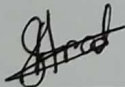
Authors:



Ridwan Mahbub
Student ID: 180041230

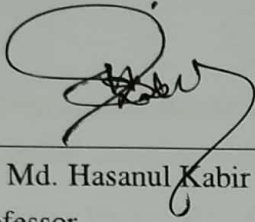


Samiha Shafiq Anuva
Student ID: 180041137



Ifrad Towhid Khan
Student ID: 180041225

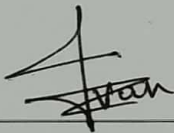
Supervisors:



Dr. Md. Hasanul Kabir
Professor

Department of Computer Science and Engineering
Islamic University of Technology


Co-Supervisors:



Shahriar Ivan

Lecturer

Department of Computer Science and Engineering
Islamic University of Technology



Md. Zahidul Islam

Lecturer

Department of Computer Science and Engineering
Islamic University of Technology

Contents

<i>Declaration of Authorship</i>	i
List of Figures	v
List of Tables	vii
<i>Abstract</i>	viii
1 Introduction	1
2 Related Works	4
2.1 Finding important Convolutional neural network parameters	4
2.2 Attention on Input Features	5
2.3 Attention on weights	7
2.4 Weight-based attention in different architectures other than CNN	9
3 Preliminary Experiments	10
3.1 Implementing Location-based Weight Excitation with different existing Attention Modules	10
3.1.1 Channel Attention Module: Revisiting Squeeze and Excitation	10
3.1.2 Channel and Spatial Attention Module : CBAM, BAM	11
3.1.3 Branch Attention / Cross Dimensional Attention Module: Rotate to Attend	14
3.2 Improving LWE with Novel Attention Module (Modified CBAM + Skip Connections)	15
3.3 Experiment with different activation functions for magnitude based weight excitation	17
3.3.1 Experiment with existing activation functions for MWE	17
3.3.2 Experiment with novel activation function for MWE	17
3.4 Paused Weight Excitation method	18
3.5 Weight Excitation in MLP Mixer	19
3.6 Weight Excitation as an alternative to Regularization	20

4	Proposed Method	22
4.1	Improving LWE with Novel Attention Module (Modified CBAM + Skip Connections)	22
4.1.1	Channel Attention Module	22
4.1.2	Spatial Attention Module	23
4.1.3	Placement of attention modules	24
4.1.4	Skip Connections	24
4.1.5	Activation Function	24
4.2	Global Weight Excitation	25
5	Datasets	26
6	Experiments and Result Analysis	31
6.1	Results for Location-based Weight Excitation using different Attention Modules	31
6.1.1	Results for Location-based Weight Excitation using SE	31
6.1.2	Results for Location-based Weight Excitation using CBAM	31
6.1.3	Results for Location-based Weight Excitation using Cross Dimensional Attention	32
6.2	Performance Analysis of our Novel Location-based Weight Excitation method	33
6.3	Results for Magnitude based Weight Excitation using Novel Activation Function	34
6.4	Results for Magnitude-based Weight Excitation on MLP Mixer	36
6.5	Experiment using different Activation Functions in SE module	39
6.6	Performance Analysis of Global Weight Excitation Method	40
6.7	Comparison between Weight Excitation and Regularization	41
7	Contributions	44
8	Future Works and Conclusion	46
	References	48

List of Figures

1.1	Preliminary working principle of weight excitation	2
2.1	NN Pruning	4
2.2	Three-step training pipeline for learning weights and connections	5
2.3	A Squeeze and Excitation Block	6
2.4	Convolutional Block Attention Module	6
2.5	Convolutional Triplet Attention Module. Observe that each branch is calculating attention along a particular plane, which is then broadcasted to meet the same dimensions as input. Later the three attention maps are averaged.	7
2.6	Magnitude-based weight excitation. The graph shows the activation function that has been used. Observe that the values at the extreme ends are increased compared to their identity values.	8
2.7	Location-based weight excitation. The flow chart shows how location-based weights are calculated. The portion is marked red in an SE module. It works differently as the input in this case is the filter kernel itself and the output is a modified filter kernel.	8
2.8	Dynamic convolution. Each convolution block has an attention magnitude multiplied with it. This results in a much better representation power by the CNN model.	9
3.1	The schema of the original Residual module (left) and the SE-ResNet module (right). [1]	11
3.2	CBAM Module	12
3.3	CBAM Flow Chart	13
3.4	CBAM Flow Chart	13
3.5	BAM Block Diagram	14
3.6	Triplet Attention Method: Rotate dimensions to capture cross-dimensional interaction	15

3.7	Comparisons with different attention modules: (a) Squeeze Excitation (SE) Module; (b) Convolutional Block Attention Module (CBAM); (c) Global Context (GC) Module; (d) triplet attention (ours). The feature maps are denoted as feature dimensions, e.g. $C \times H \times W$ denotes a feature map with channel number C , height H and width W . \otimes represents matrix multiplication, \odot denotes broadcast element wise multiplication and \oplus denotes broadcast element-wise addition.	16
3.8	Graph of our proposed activation function	18
3.9	Proposed Method	19
3.10	MLP Mixer with weight Excitation. The red marked regions show where magnitude based weight excitation has been applied.	20
3.11	(a)Over-fitting, (b) Appropriate fitting (c)Under-fitting in Machine Learning	20
3.12	Effect of bias and variance on Machine Learning Model	21
4.1	CBAM Module	23
4.2	Our Novel Attention Module	24
4.3	Global Weight Excitation	25
5.1	Cifar10 Dataset	27
5.2	Cifar100 Dataset	28
5.3	ImageNet Dataset	28
5.4	PASCAL VOC Dataset	29
5.5	CINIC-10 Dataset	30
6.1	Comparison of Original WE and our proposed novel methodology on ImageNet	35
6.2	Comparison between the learning curve of ResNet18 before and after applying MWE	36
6.3	Comparison between the learning curve of ResNet18 before and after applying Modified MWE	37
6.4	Training and Validation Accuracy of MLP Mixer on Cifar100 without MWE	38
6.5	Training and Validation Accuracy of MLP Mixer on Cifar100 with MWE .	39
6.6	Training and Validation Accuracy of MLP Mixer on Cifar10 without MWE	40
6.7	Training and Validation Accuracy of MLP Mixer on Cifar10 with MWE .	41
6.8	Performance analysis of Global LWE	42
6.9	The effect of applying Regularization on training curve	43
6.10	Weight Excitation has a similar effect on the learning curve as regularization	43

List of Tables

6.1	Performance Analysis of SE	31
6.2	Performance Analysis of CBAM	32
6.3	Performance Analysis of CBAM	32
6.4	Performance Analysis of Rotate to Attend on Cifar 100 Dataset	33
6.5	Performance Analysis of our Novel Location-based Weight Excitation method on Cifar 100	33
6.6	Performance Analysis of our Novel Location-based Weight Excitation method on Cinic-10	33
6.7	Performance Analysis of our Novel Location-based Weight Excitation method on Imagenet	34
6.8	Performance Analysis of Magnitude Based Weight Exciation	34
6.9	Performance Analysis of Weight Exciation on MLP Mixer	36
6.10	Experiment using different Activation Functions in SE module	40
6.11	Performance analysis of Global Weight Excitation on Imagenet	41

Abstract

To improve the representational power of convolutional neural networks, several attention mechanisms have been introduced in recent years. These attention mechanisms are calculated on input feature maps by enhancing some parts of the input data and diminishing other parts of the input data as all parts of the input do not contain important features for training. One exception can be seen where weights are used in place of input feature maps and this approach is known as weight excitation. Since the weights of a CNN get fine-tuned based on the input data, calculating attention on weights can be an alternative to calculating attention on input feature maps. One advantage of this method is that this doesn't introduce any additional computational cost at inference time. In this paper, we aimed to overcome the limitations of existing weight-based attention mechanisms. We have conducted several experiments to conclude whether weights can be used as an alternative to input feature maps for computing attention and if this applies to all existing attention mechanisms for Convolutional Neural Networks.

Keywords— Convolutional Neural Network, Weight Excitation, Attention Mechanisms, Feature Map

Chapter 1

Introduction

Convolutional neural networks [2] have proven to be very effective in analyzing visual data and have wide applications in all fields of computer vision. Convolutional Neural Networks or CNNs are made of up convolutional blocks that consist of filter kernels. These filters perform convolution operations on the visual data. They consist of weights or learnable parameters that are acquired backpropagation [3] during the training phase, depending on various factors like the dataset used, choice of the optimizer [4], and loss function.

It has been observed through various experiments that all weights or learnable parameters of a convolutional neural network do not contribute equally to obtaining the final result. Some weights seem to carry more importance than others. Various existing works have tried to use this phenomenon to either create a smaller architecture by ignoring or leaving out the unimportant weights [5] or by attending more to the important weights while suppressing the less important ones [6], which ultimately resulted in higher accuracy. It is also worth mentioning that these attention [7] mechanisms come with little to no extra computational expenses for the model during inference time or practical application after the training phase.

Apart from convolutional neural networks, recently alternative deep learning architectures like vision transformers [8] and MLP (Multi-Layer Perceptron) Mixer [9] have proven to be equally competent methods for the analysis of visual data. Transformer architecture can even surpass convolution-based models as it can capture the global context much better. MLP mixer is also very useful as it can almost match the accuracy of convolutional neural networks. MLP mixer-based architectures outperform other models when it comes to inference speed. No research work exists that takes into account the effectiveness of weights or learnable parameters in the case of the newer models. As the general principle

of learnable parameters is the same for the 3 mentioned types of architectures, if weight importance varies in the case of convolution-based models, the same should be the case for MLP mixer and vision transformer.

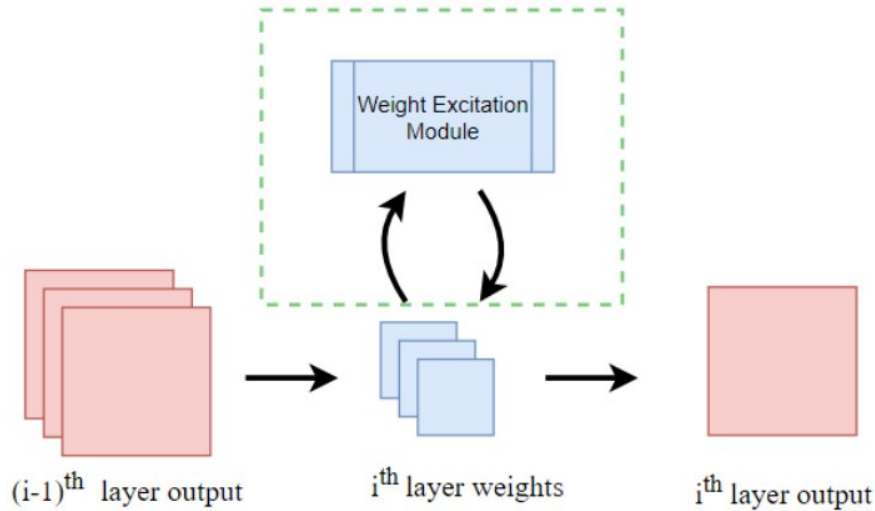


Figure 1.1: Preliminary working principle of weight excitation

In our works, we proposed new methods of weight excitation [6] that are more effective and maintain the same characteristics such as zero inference time overhead. We also extend weight excitation to other models of computer vision. Formally, we can summarize our contributions by the following points:

1. **Extension of Weight Excitation:** As no other work has been done to excite weights of MLP Mixer architecture, we are the first ones to implement magnitude-based weight excitation for it. In MLP Mixers, there are fully connected layers. So we applied the magnitude-based weight excitation to the fully connected.
2. **Improvement of Existing Weight Excitation Technique with a Novel Attention Module:** Just like the Squeeze and Excitation module has been used for the location-based weight excitation in the work of Quader, *et al.* [6], we took another approach for this weight excitation. We modified the CBAM (Convolutional Block Attention Module) [10] to use it as an attention module for location-based weight excitation and introduced skip connections. The idea behind using skip connections was that the model might perform better when some weights are not affected by weight excitation.
3. **Generalizing weights to be used as input for all attentions of CNNs** We have conducted extensive experiments to see if weight can be used as input in all types

of attention modules. We have used weights as input in spatial attention, channel attention, cross-dimensional attention, and branch attention and we saw that weights could be used as input instead of input feature maps in all of these attention modules.

4. **Checking the regularization effect of weight excitation:** We have conducted extensive experiments to see if weight excitation can be used as a regularization method. We have repeatedly observed that in our experiments weight excitation gives us much more smoother curves. This ultimately proves that weight excitation can be used as a regularization method.

Apart from these, we figured out some limitations and problems of the existing weight excitation method and proposed some solutions to those.

Chapter 2

Related Works

2.1 Finding important Convolutional neural network parameters

The earliest work of ConvNet Pruning [5, 11–13] came into existence by identifying important parameters and removing unimportant parameters to simplify networks, improve generalization, reduce hardware or storage requirements and increase the speed of training. These pruning methods included- pruning convolutional neural networks for resource-efficient inference [14], compressing deep neural networks with pruning, trained quantization and Huffman coding [13], and second-order derivatives for network pruning [12].

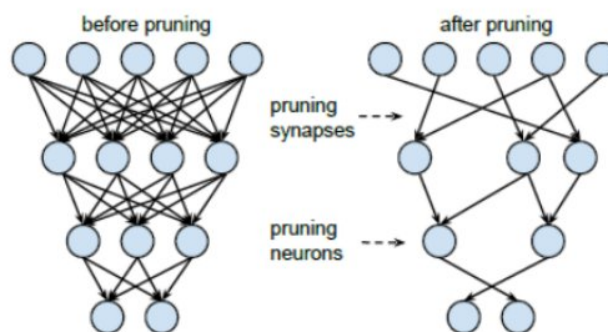


Figure 2.1: NN Pruning

We have found 3 common criteria for identifying important ConvNet parameters-

- Higher minimal increase in training error after removing a parameter indicates higher importance [11, 12]
- Higher magnitude parameters correspond to higher importance [14, 15]
- High or low importance of convolution filter weights depends on location [1]

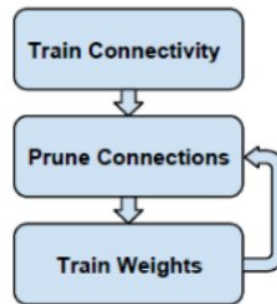


Figure 2.2: Three-step training pipeline for learning weights and connections

2.2 Attention on Input Features

In most of the earlier works, attention has been applied to activation maps/feature maps [1, 10, 16, 17]. These architectures calculate attention on feature maps to increase the models' representation capacity. Squeeze and excitation Network [1] is such an architecture that calculates the channel attention map of feature maps to produce refined feature maps. Squeeze and Excitation block is a simple and lightweight module that can be easily integrated into Convolutional Neural Networks with minimal additional computational cost. To limit model complexity and aid generalization, SE parameterizes the gating mechanism by forming a bottleneck [18] with two fully connected (FC) layers around the non-linearity. One limitation of the SE block is that it doesn't consider the spatial attention of inputs.

To overcome the limitations of SE [1], Convolution Block Attention Module [10] was introduced. Given an intermediate feature map, the CBAM module sequentially infers attention maps along two separate dimensions, channel and spatial, then the attention maps are multiplied by the input feature map for adaptive feature refinement.

After CBAM [10], Bottleneck Attention Module [16] was introduced. BAM infers two types of 3D attention maps - channel attention map and spatial attention map. In the chan-

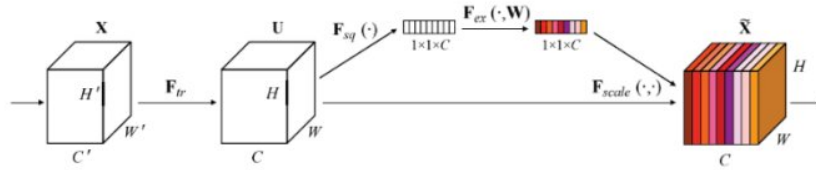


Figure 2.3: A Squeeze and Excitation Block

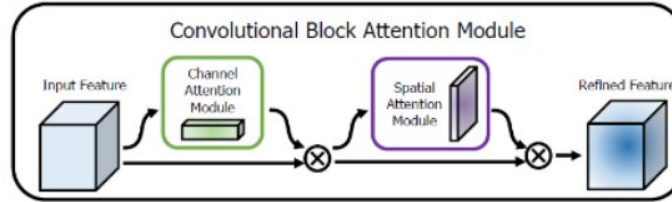


Figure 2.4: Convolutional Block Attention Module

nel attention module, the feature map in each channel is aggregated using global average pooling [19] to produce a channel vector that encodes global information in each channel. The spatial attention map is produced to emphasize or suppress features in different spatial locations. The feature is projected into a reduced dimension using 1×1 convolution [19] to integrate and compress the feature map across the channel dimension.

One of the glaring problems with the CBAM module was the relationship between channel attention and spatial attention was ignored. Since channel attention and spatial attention are being calculated separately, there is very little scope to capture the interdependence of the two. This problem was solved by Misra *et al.* [20]. The work proposed a Convolutional Triplet Attention Module [20] that calculated attention whilst considering the interrelations between the height, width, and channel dimension of the input tensor. This did 2 things very effectively, firstly it helped in capturing rich discriminative feature representations at a negligible computational overhead. Secondly, it captured the interaction between the spatial dimensions and the channel dimension of the input tensor.

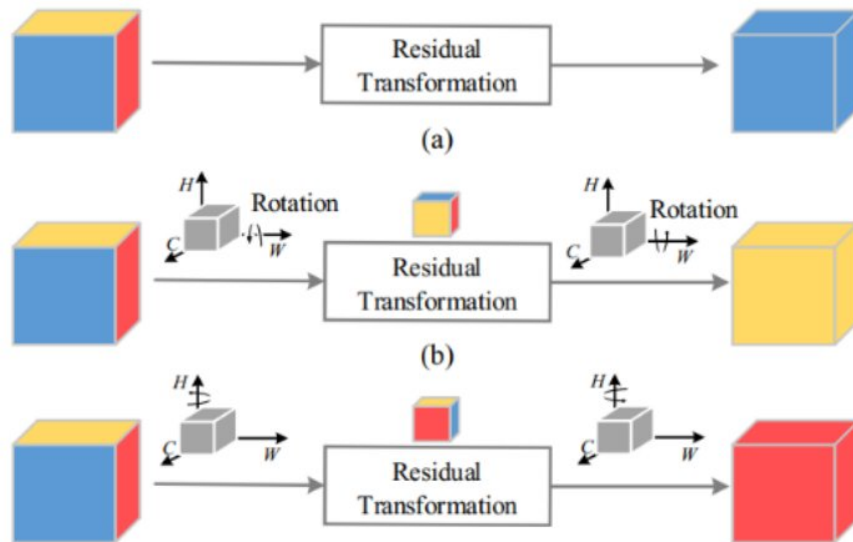


Figure 2.5: Convolutional Triplet Attention Module. Observe that each branch is calculating attention along a particular plane, which is then broadcasted to meet the same dimensions as input. Later the three attention maps are averaged.

2.3 Attention on weights

Prior works include Weight Normalization [21] which is a Weight Reparameterization technique used to decouple the length of weight vectors from their directions. After this other weight reparameterization techniques such as - Weight Standardization [22] and Spectral Normalization [23] were introduced. Han et al showed the importance of learning weights along with connections in a neural network [5]. The concept of weights having different levels of importance based on their location was first introduced in the ablation study of Squeeze and Excitation Networks [1].

The results found from [5] and [1] worked as the motivation to research weight-based attention. To provide more attention to important weights based on magnitude and location, the Weight Excitation [6] method was introduced. This weight reparameterization method emphasizes the important weights during training and suppresses the less important weights. This method increased the accuracy of Convolutional Neural Networks [20] without introducing any additional computational cost during the inference stage.

Weight Excitation proposed 2 new strategies to figure out the attention of weight kernels. In the first case, an observation was made that weights with higher magnitude weights had more importance. So, an activation function was designed to increase the important weights and suppress the less important ones.

Another approach was location-based weight excitation. This method identified weights

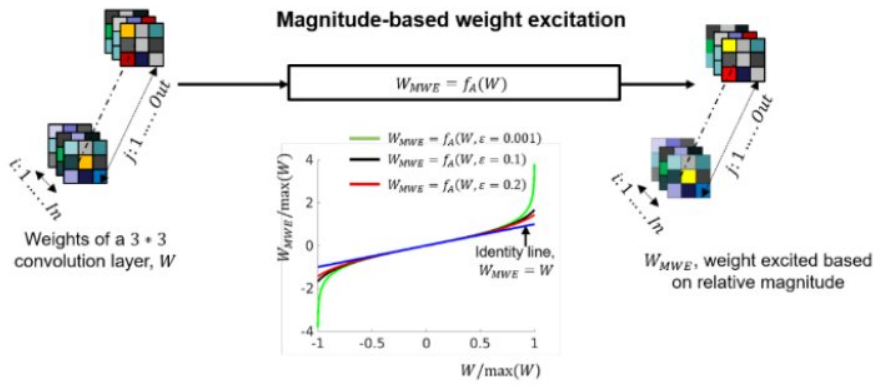


Figure 2.6: Magnitude-based weight excitation. The graph shows the activation function that has been used. Observe that the values at the extreme ends are increased compared to their identity values.

based on their location in the kernel block and assigned an attention map for every channel in the filter kernel.

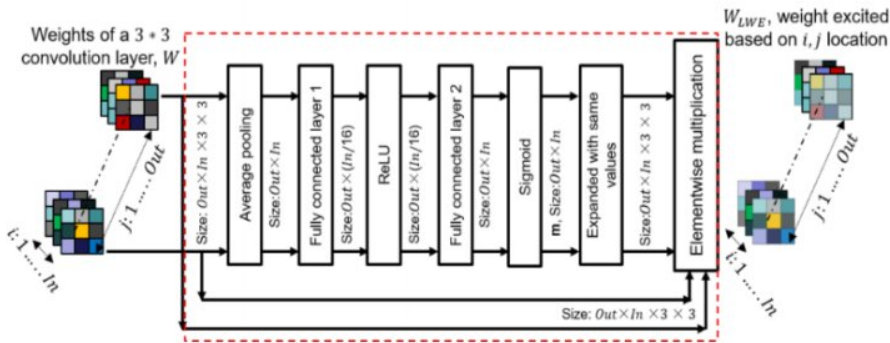


Figure 2.7: Location-based weight excitation. The flow chart shows how location-based weights are calculated. The portion is marked red in an SE module. It works differently as the input in this case is the filter kernel itself and the output is a modified filter kernel.

Recently, there have been many advances in the field of dynamic convolution [24]. Dynamic convolution is a concept where architectures have multiple kernels that are dynamic and weighted by attention maps based on the input features. The basic intuition is that instead of having just one set of weights, we can have multiple sets of weights that have different attention based on the input features. This work was advanced by the introduction of omnidirectional attention [25] that takes into account the attention of weight kernels from 4 different aspects like the number of parameters, spatial size, and the number of input and output channels.

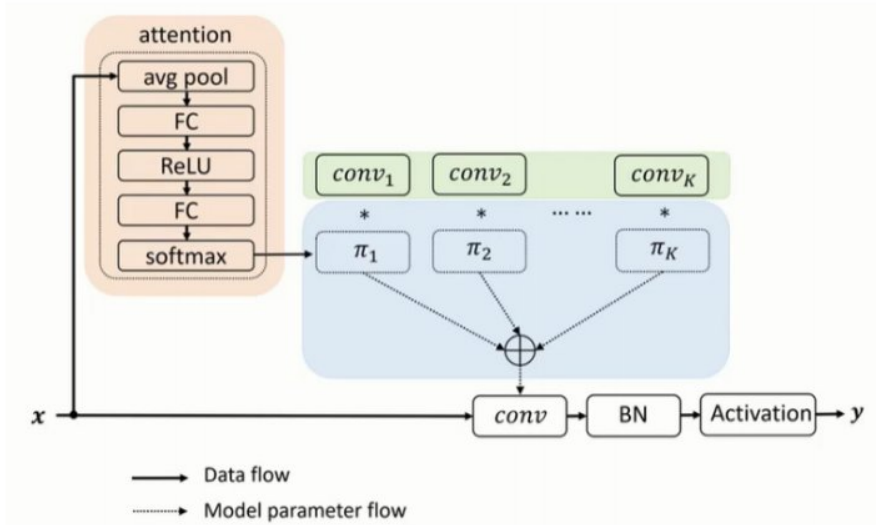


Figure 2.8: Dynamic convolution. Each convolution block has an attention magnitude multiplied with it. This results in a much better representation power by the CNN model.

2.4 Weight-based attention in different architectures other than CNN

Successful implementation of Weight Excitation in a Convolutional neural network and improved classification and detection accuracy motivated us to look into other architectures [8, 9] where Weight Excitation can be applied. MLP Mixer [9] is a simple architecture built entirely on multi-layer perceptrons [26]. Vision Transformer is an architecture that can very effectively take into account the global context of data. Applying weight excitation in such diverse architectures can be a very impactful contribution to the computer vision community.

Chapter 3

Preliminary Experiments

We have conducted several preliminary experiments based on our initial hypothesis. The experiments that were conducted covered different fields of work. Our initial plans consisted of improving weight excitation for convolutional neural networks and then applying weight excitation to other architectures like MLP Mixer and vision transformer. We ended up trying both types of experiments in parallel and have achieved preliminary success in both. Details of the experimentations have been discussed below:

3.1 Implementing Location-based Weight Excitation with different existing Attention Modules

3.1.1 Channel Attention Module: Revisiting Squeeze and Excitation

Squeeze and Excitation Module for LWE

Squeeze and Excitation module was used in the original weight excitation paper for location-based weight excitation. Two types of operations are performed in this module. In squeeze operation, we pass the input feature map to this module and it aggregates features across the spatial dimension. This gives us a feature descriptor that contains the global distribution of the channel-wise feature responses. The excitation operation helps to fully capture channel-wise dependencies by learning non-linear interaction between channels. The output of this block is a modified feature map that helps to boost feature discriminability.

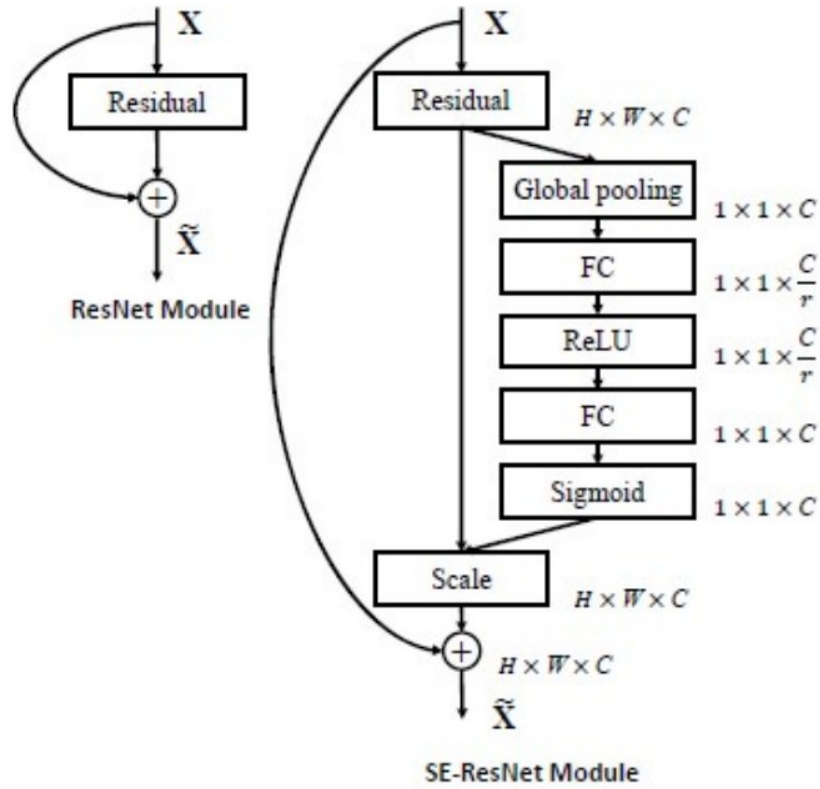


Figure 3.1: The schema of the original Residual module (left) and the SE-ResNet module (right). [1]

3.1.2 Channel and Spatial Attention Module : CBAM, BAM

Convolution Block Attention Module for LWE

CBAM module is an improvement to Squeeze and Excitation module. This applies both channel and spatial attention to the input in a sequential manner. In the channel attention module, Average pooling aggregates spatial information and Max-pooling gathers important clues about distinctive object features. The Spatial attention map in the channel attention module encodes where to emphasize or suppress weights. The following equations 4.1 and 4.2 are used here

$$W' = M_c(W) \otimes W \tag{3.1}$$

$$W'' = M_c(W') \otimes W' \tag{3.2}$$

Channel Attention Block: The input to this weight excitation module is the NxN convo-

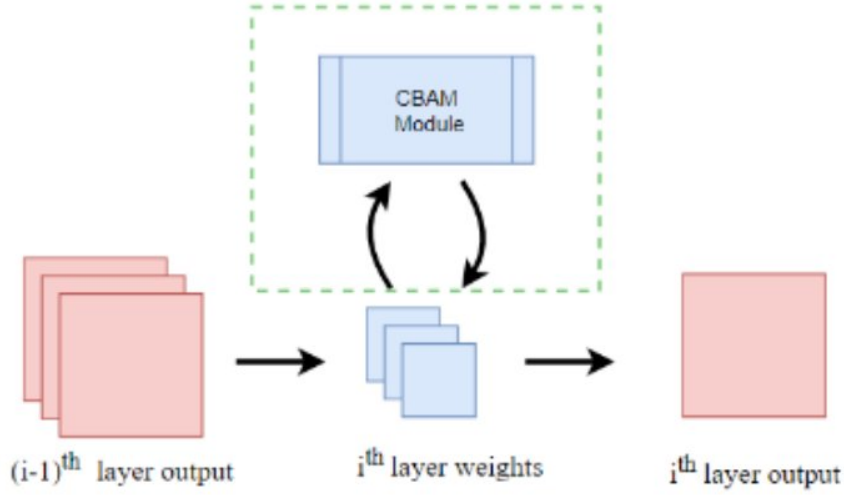


Figure 3.2: CBAM Module

lution filter kernels. At first, we generate two spatial context descriptors for aggregating spatial information of weights using average pooling and max pooling operation in parallel. Then both of the spatial context descriptors are passed through a shared network. The shared network/MLP consists of two fully connected layers and a hidden layer. The output of this shared network is passed through a sigmoid layer. The output of the sigmoid layer gives us our channel attention map. We perform channel-wise multiplication of this channel attention map with the original filter kernel to get modified weights. The following equation 4.3 is used here

$$\begin{aligned}
 M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\
 &= \sigma(W_1(W_0(F_{avg}) + W_1(W_0(F_{max})))
 \end{aligned}
 \tag{3.3}$$

Spatial Attention Block: The modified weights from the output of the channel attention module are passed as input to the spatial attention module. Here max operation and mean operation is performed on the input weights and their outputs are concatenated. Then we perform convolution on the output of the concatenation operation. The result of convolution is passed through a sigmoid layer which gives us our spatial attention map. Next, we perform element-wise multiplication of the spatial map with the output from the channel attention module to produce the final output of our location-based weight excitation module. The size of the filter kernel remains the same after applying weight excitation. The following equation 4.4 is used here

$$\begin{aligned}
 M_s(F) &= \sigma(f^{7 \times 7}([AvgPool(F); MaxPool(F)])) \\
 &= \sigma(f^{7 \times 7}([F_{avg}^s; F_{max}^s]))
 \end{aligned}
 \tag{3.4}$$

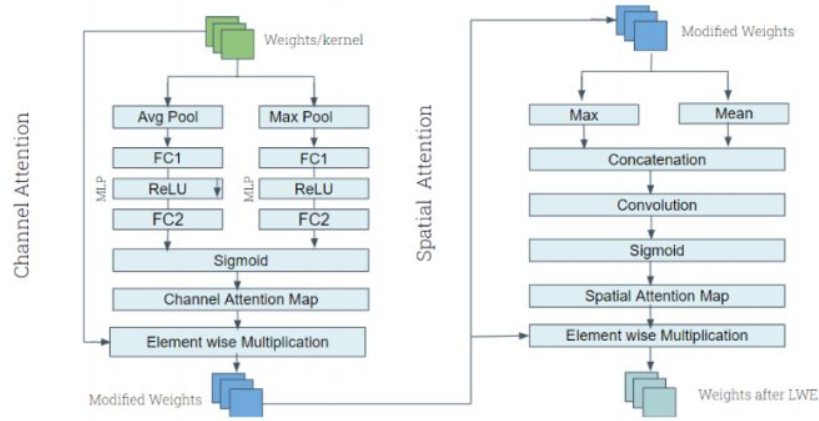


Figure 3.3: CBAM Flow Chart

Placement of Attention Blocks: Channel attention and spatial attention modules can be placed sequentially one after another or in parallel. In [10] it was shown that placing the two modules separately produced better results. In our weight excitation architecture, we placed the channel attention module first and then sequentially placed the spatial attention module.

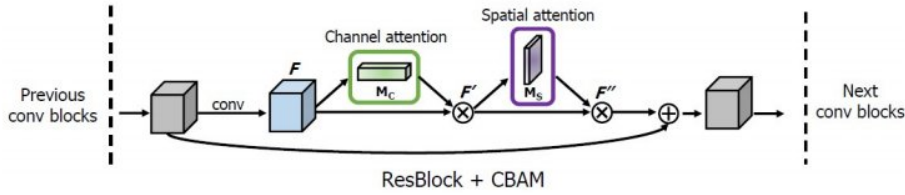


Figure 3.4: CBAM Flow Chart

Bottleneck Attention Module

Similar to CBAM, BAM [16] also infers both channel attention and spatial attention, unlike Squeeze and Excitation approach. Here, for a given input feature map $F \in R^{C \times H \times W}$, BAM infers a 3D attention $M(F) \in R^{C \times H \times W}$. The channel attention $M_c(F) \in R^C$ and the spatial attention $M_s(F) \in R^{H \times W}$ are computed at two separate branches. Finally, the attention map $M(F) = \sigma(M_c(F) + M_s(F))$ is computed. The refined feature map F' is computed as: $F' = F + F \otimes M(F)$

Channel Attention Block: Here as each channel contains a specific feature response, the feature map in each channel is aggregated using global average pooling on the feature map F and produces a channel vector $F_c \in R^{C \times 1 \times 1}$. This vector softly encodes global information in each channel. To estimate attention across channels from the channel

vector F_c , it uses a multi-layer perceptron (MLP) with one hidden layer. After the MLP, it adds a batch normalization (BN) layer to adjust the scale with the spatial branch output.

$$\begin{aligned} M_c(F) &= BN(MLP(AvgPool(F))) \\ &= BN(W_1(W_0 AvgPool(F) + b_0) + b_1) \end{aligned} \quad (3.5)$$

Spatial Attention Block: The spatial branch produces a spatial attention map $M_s(F) \in R^{H \times W}$ to emphasize or suppress features in different spatial locations. The feature $F \in R^{C \times H \times W}$ is projected into a reduced dimension $F \in R^{(C/r) \times H \times W}$ using 1×1 convolution to integrate and compress the feature map across the channel dimension. After the reduction, two 3×3 dilated convolutions are applied to utilize contextual information effectively. Finally, the features are again reduced to $R^{1 \times H \times W}$ spatial attention map using 1×1 convolution. For scale adjustment, a batch normalization layer is applied at the end of the spatial branch.

$$M_s(F) = BN(f_3^{1 \times 1}(f_2^{3 \times 3}(f_1^{3 \times 3}(f_0^{1 \times 1}(F)))))) \quad (3.6)$$

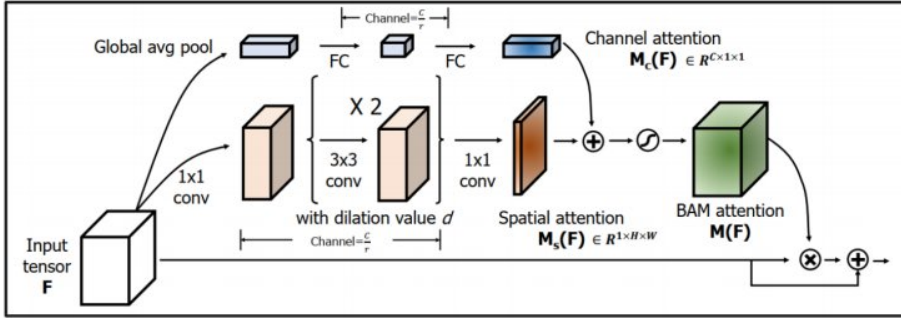


Figure 3.5: BAM Block Diagram

3.1.3 Branch Attention / Cross Dimensional Attention Module: Rotate to Attend

Triplet attention is a novel method to measure attention weights by capturing the cross-dimensional interaction of the input tensors. This attention module comprises of three branches each responsible for capturing crossdimension between the spatial dimensions and channel dimension of the input. It builds inter-dimensional dependencies by the rotation operation followed by residual transformations. After that it encodes inter-channel and spatial information with negligible computational overhead. It is lightweight and efficient. It ensures rich feature representation by capturing cross-dimensional interaction

at the time of computing attention weights.

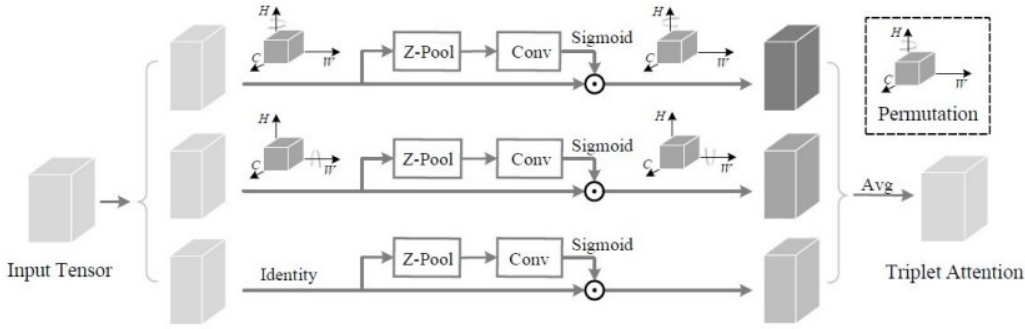


Figure 3.6: Triplet Attention Method: Rotate dimensions to capture cross-dimensional interaction

Upon receiving an input tensor, each of the branches of the triplet attention module captures the cross-dimension between the spatial dimensions and channel dimension of the input. If the shape of the input tensor is $(C \times H \times W)$, each branch aggregates cross-dimensional interactive features between either the spatial dimension H or W and the channel dimension C . This is done by permuting the input tensors in each branch and then passing the tensor through a Z-pool, followed by a convolutional layer with a kernel size of $K \times K$. Then sigmoid activation layer is applied to generate attention weights. These attention weights are then applied to the permuted input tensor. After this, the input tensor is permuted back to the original input shape. This way triplet attention method captures cross-dimensional interaction without any dimensionality reduction. It also eliminates indirect correspondence between channels and weights.

We modified the triplet attention module to take weights as input instead of feature maps. Experimental results confirmed that this module can be used for weight excitation as well.

3.2 Improving LWE with Novel Attention Module (Modified CBAM + Skip Connections)

Existing location-based weight excitation was implemented using the Squeeze and Excitation module. The problem with that module was that it used only average pooling features and didn't utilize the max pooling features. [27]. Only channel attention was applied for location-based weight excitation. To produce finer location importance maps, we used CBAM which utilizes both average pooled and max pooled features of weights. This weight excitation module sequentially infers a channel attention map and a spatial attention map. In the channel attention module, Average pooling aggregates spatial in-

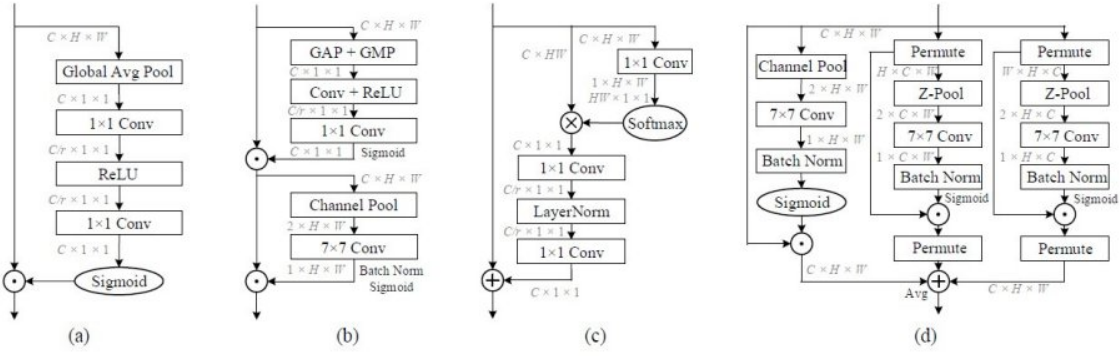


Figure 3.7: Comparisons with different attention modules: (a) Squeeze Excitation (SE) Module; (b) Convolutional Block Attention Module (CBAM); (c) Global Context (GC) Module; (d) triplet attention (ours). The feature maps are denoted as feature dimensions, e.g. $C \times H \times W$ denotes a feature map with channel number C , height H and width W . \otimes represents matrix multiplication, \odot denotes broadcast element wise multiplication and \oplus denotes broadcast element-wise addition.

formation and Max-pooling gathers important clues about distinctive object features. The Spatial attention map in the channel attention module encodes where to emphasize or suppress weights. Channel attention and spatial attention modules can be placed sequentially one after another or in parallel. In [10] it was shown that placing the two modules separately produced better results. In our weight excitation architecture, we placed the channel attention module first and then sequentially placed the spatial attention module.

Previously, we changed all of the model weights using the weight excitation method. But it's possible that the model's entire weight doesn't need to be changed. Some of the original weights were preserved using skip connections. The model was able to produce sharper attention and gain a better knowledge of the dataset by leaving some weights unchanged. As a result, the model was able to predict the training set data more precisely.

In our previous experiments, activation was accomplished through ReLU. We came to the conclusion that Leaky ReLU will function better for weight excitation after conducting experiments with various activation functions. Because of this, we used Leaky ReLU rather than ReLU.

3.3 Experiment with different activation functions for magnitude based weight excitation

3.3.1 Experiment with existing activation functions for MWE

In the original paper, the proposed activation function was differentiable and avoided vanishing and exploding gradient problems. We have tried out different types of activation functions for magnitude-based weight excitation. We did this by plotting the curve and seeing if the curve increases the higher magnitude value while decreasing the lower magnitude values. None of the existing activation functions showed promising results. The activation functions that we tried are as follows:

- Sigmoid Activation
- Tanh Activation
- ReLU Activation
- GELU Activation
- Leaky ReLU Activation
- SWish Activation
- Softsign Activation
- Softplus Activation
- Shifted Softplus Activation
- Lecun’s Tanh Activation
- StarReLU Activation

None of these existing activation functions gave us satisfactory results. Some of these were quite close to what we were expecting and some of these were not suitable for weight excitation experiments. So, we tried modifying some existing activation functions to get our desired results.

3.3.2 Experiment with novel activation function for MWE

After seeing no improvement in magnitude-based weight excitation after using existing activation functions, we tried to formulate an activation function that works well for MWE. We devised an activation function that does not decrease the lower magnitude weights as much as it increases the higher magnitude weights. Our reasoning behind this was that lowering the value of lower magnitude weights might remove some important feature descriptions. This ensured that no feature was neglected.

$$\frac{0.5x}{\log(1 + \exp^{-0.3x})} \quad (3.7)$$

In order to formulate new activation functions, we followed a number of steps. The first thing we did was plot various existing activation functions in Desmos which is a graph plotting software. Then we modified these activation functions and observed the resultant curves. After experimenting with different functions and modifying them, we came up with a function that increases the higher magnitude values but decreases the lower magnitude values by a very small margin. Then we used this function as our activation function for MWE. When we ran this experiment for 30 epochs, we saw an improvement in accuracy. But after training the model till convergence, no significant improvement in accuracy was seen.

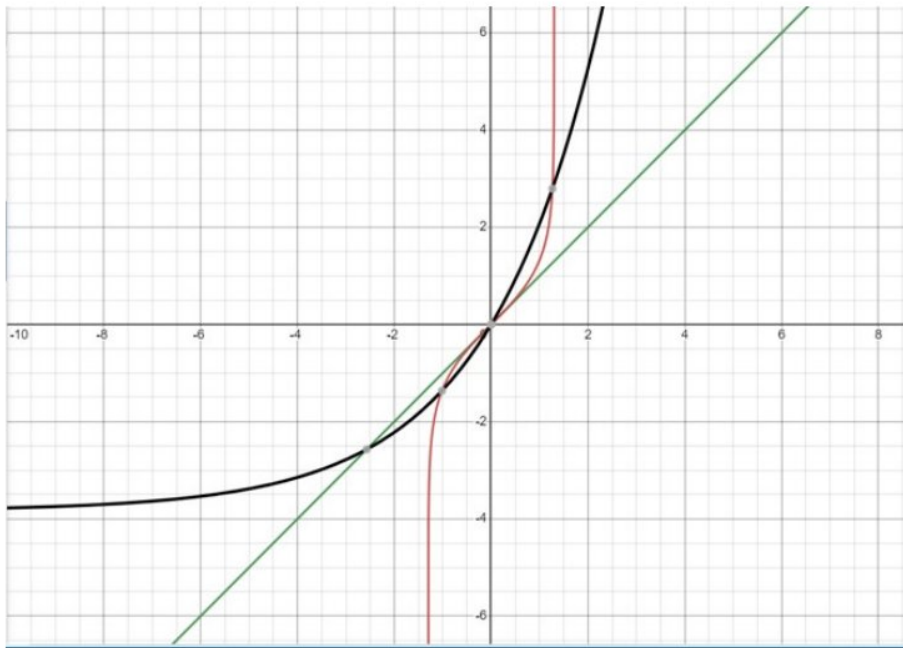


Figure 3.8: Graph of our proposed activation function

3.4 Paused Weight Excitation method

Some issues with the existing Weight Excitation method :

- If we re-calibrate [28] weights from the very first epoch, our model will fail to learn properly.
- Applying WE without the knowledge of important weights can produce erroneous results.

Proposed Solution : Training the model without any weight excitation for the initial few

epochs. Then we will apply weight excitation to emphasize the important weights and suppress the less important weights.

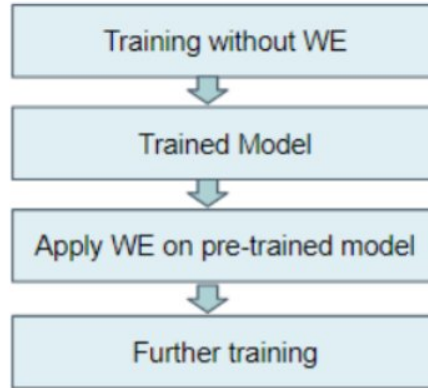


Figure 3.9: Proposed Method

3.5 Weight Excitation in MLP Mixer

MLP Mixer is a new and forthcoming deep learning model for visual data analysis. MLP Mixer was developed by the Google Research team. As mentioned earlier, it is a very simple model in terms of architecture but it can perform very well even when compared to state of the art architectures like ConvNets and Vision Transformers. That being said, unlike Convnets, MLP Mixer has not seen any attention modules being developed for it. Keeping this in mind, we extend the concept of weight excitation to MLP Mixers. We use the magnitude based weight excitation method for this purpose. We modify the fully connected layers of the MLP mixer to use the activation function given below in 3.8

$$\omega_{MWE} = f_A(W) = M_A \times 0.5 \times \ln \frac{1 + \omega/M_A}{1 - \omega/M_A} \quad (3.8)$$

Where, $M_A = (1 + \epsilon_A) \times M$, M being the maximum magnitude of weights in the kernels and ϵ_A is the hyper-parameter for modifying weight within the range $0 < \epsilon_A < 0.2$

multirow

This ensures that the weights of the MLP Mixer are changed according to their magnitudes, such that the weights with higher magnitudes are given more importance. MLP Mixer is entirely formed of fully connected layers. This means the weight excitation is being applied at all levels, from channel mixing layers, and token mixing layers all the way to the final fully connected classifier. MLP mixer only has fully connected layers.

So, the application of location-based weight excitation where we identify important channels, was not possible here. That is why only tried magnitude-based weight excitation. In order to perform well, MLP Mixer has to be trained on very large datasets(Imagenet 21k and JFT 300M). Which was out of our scope.

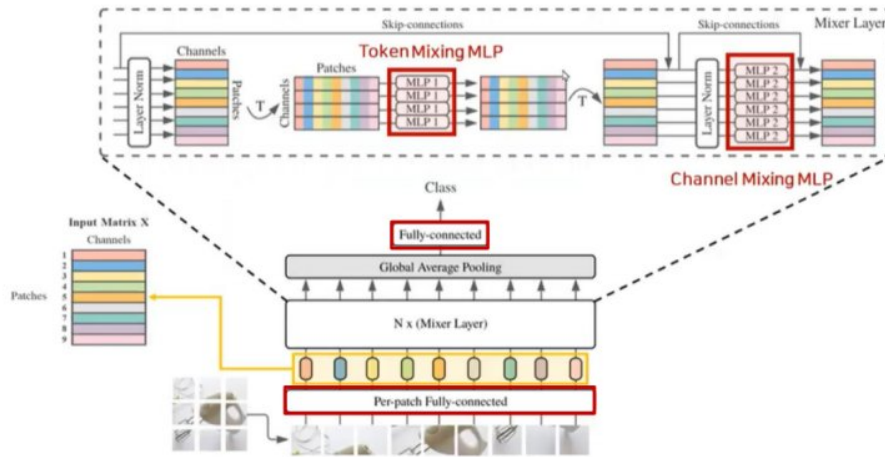


Figure 3.10: MLP Mixer with weight Excitation. The red marked regions show where magnitude based weight excitation has been applied.

3.6 Weight Excitation as an alternative to Regularization

Regularization [29] is an important step in model training which helps the model to avoid overfitting. Overfitting occurs when the model fits the training data very well but it fails to generalize on the test dataset or unseen dataset. This happens because the model learns the noise of the training data which helps the model to memorize the training data instead of learning the patterns in the training data. Overfitting results in poor accuracy.

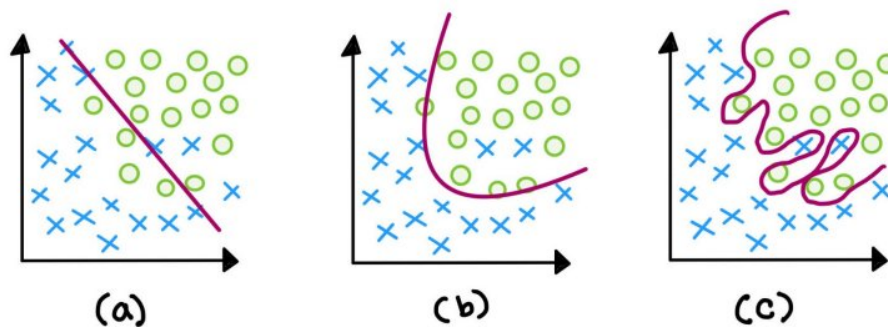


Figure 3.11: (a)Over-fitting, (b) Appropriate fitting (c)Under-fitting in Machine Learning

The weight Excitation method smoothens the learning curve, so the effect of applying weight excitation on a machine learning model is quite similar to the effect of apply-

ing regularization. It increases accuracy on test data, reduces noise, and smoothens the learning curve. If we plot the training vs test accuracy graph, the graph where we use regularization, and the graph where we use weight excitation both give a smoother curve compared to the graph where none of the two is used.

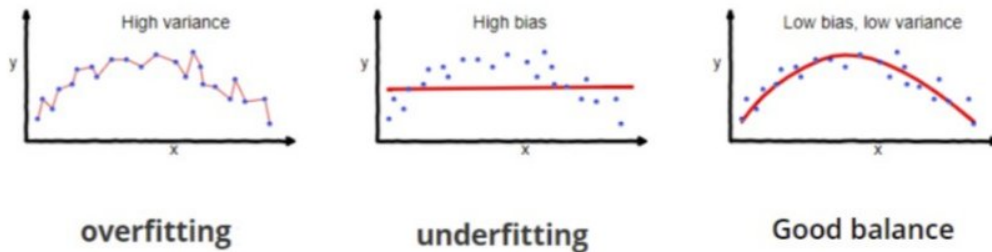


Figure 3.12: Effect of bias and variance on Machine Learning Model

In our weight excitation method, we have used weight standardization to make weight excitation more effective. This approach made the learning curve more stable than before and accelerated the learning process. So, it can be said that the effect of applying weight excitation to a model is similar to the effect of applying regularization to a model.

Chapter 4

Proposed Method

4.1 Improving LWE with Novel Attention Module (Modified CBAM + Skip Connections)

Existing location-based weight excitation was implemented using the Squeeze and Excitation module. The problem with that module was that it used only average pooling features and didn't utilize the max pooling features. [27]. Only channel attention was applied for location-based weight excitation. To produce finer location importance maps, we used CBAM which utilizes both average pooled and max pooled features of weights. This weight excitation module sequentially infers a channel attention map and a spatial attention map. In the channel attention module, Average pooling aggregates spatial information and Max-pooling gathers important clues about distinctive object features. The Spatial attention map in the channel attention module encodes where to emphasize or suppress weights. The following equations 4.1 and 4.2 are used here.

$$W' = M_c(W) \otimes W \quad (4.1)$$

$$W'' = M_c(W') \otimes W' \quad (4.2)$$

4.1.1 Channel Attention Module

The input to this weight excitation module is the $N \times N$ convolution filter kernels. At first, we generate two spatial context descriptors for aggregating spatial information of

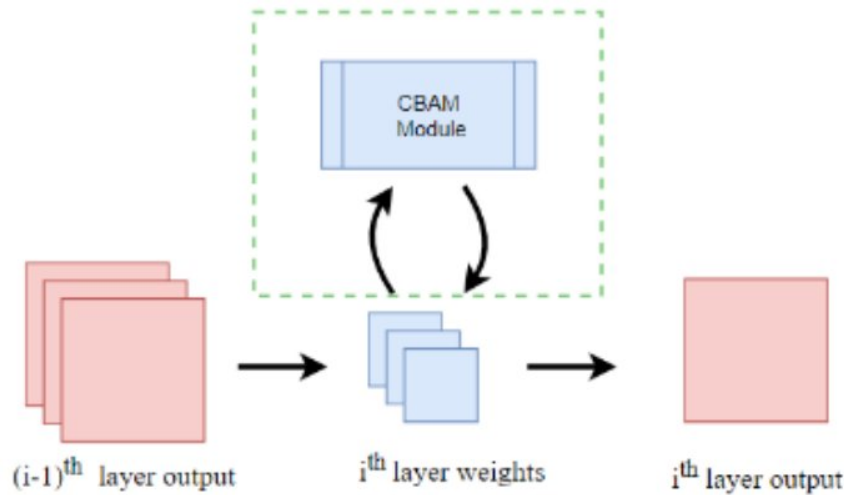


Figure 4.1: CBAM Module

weights using average pooling and max pooling operation in parallel. Then both of the spatial context descriptors are passed through a shared network. The shared network/MLP consists of two fully connected layers and a hidden layer. The output of this shared network is passed through a sigmoid layer. The output of the sigmoid layer gives us our channel attention map. We perform channel-wise multiplication of this channel attention map with the original filter kernel to get modified weights. The following equation 4.3 is used here

$$\begin{aligned}
 M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\
 &= \sigma(W_1(W_0(F_{avg}^c) + W_1(W_0(F_{max}^c)))
 \end{aligned}
 \tag{4.3}$$

4.1.2 Spatial Attention Module

The modified weights from the output of the channel attention module are passed as input to the spatial attention module. Here max operation and mean operation is performed on the input weights and their outputs are concatenated. Then we perform convolution on the output of the concatenation operation. The result of convolution is passed through a sigmoid layer which gives us our spatial attention map. Next, we perform element-wise multiplication of the spatial map with the output from the channel attention module to produce the final output of our location-based weight excitation module. The size of the filter kernel remains the same after applying weight excitation. The following equation

4.4 is used here

$$\begin{aligned}
 M_s(F) &= \sigma(f^{7 \times 7}([AvgPool(F); MaxPool(F)])) \\
 &= \sigma(f^{7 \times 7}([F_{avg}^s; F_{max}^s]))
 \end{aligned}
 \tag{4.4}$$

4.1.3 Placement of attention modules

Channel attention and spatial attention modules can be placed sequentially one after another or in parallel. In [10] it was shown that placing the two modules separately produced better results. In our weight excitation architecture, we placed the channel attention module first and then sequentially placed the spatial attention module.

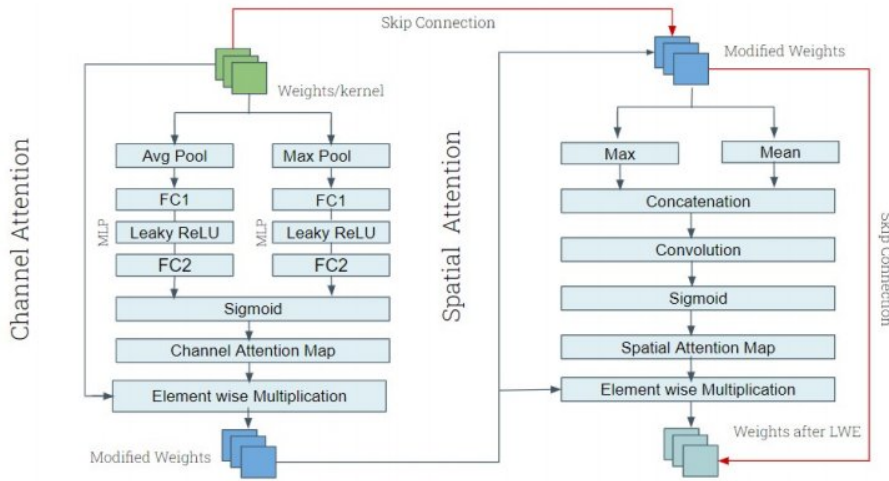


Figure 4.2: Our Novel Attention Module

4.1.4 Skip Connections

In the weight excitation method, we modified all the weights in the model. But all of the weight in the model might not need modification. We used skip connections to keep some of the original weights unchanged. Keeping some weights unchanged allowed the model to produce finer attention and get a better understanding of the dataset. This allowed the model to predict more accurately on the training set.

4.1.5 Activation Function

Previously ReLU was being used for activation. Our experiments with different types of activation functions led us to the conclusion that Leaky ReLU will perform better for

weight excitation. That is why we used Leaky ReLU instead of ReLU.

4.2 Global Weight Excitation

Global Weight Excitation is an approach to learn the global representations of the weights.

So the approach was that, instead of integrating the weight excitation module with the convolutional layer itself, the plan is to build a separate module which can be called by convolutional blocks according to their requirement.

This will help to the module to learn a global representations of the weight.

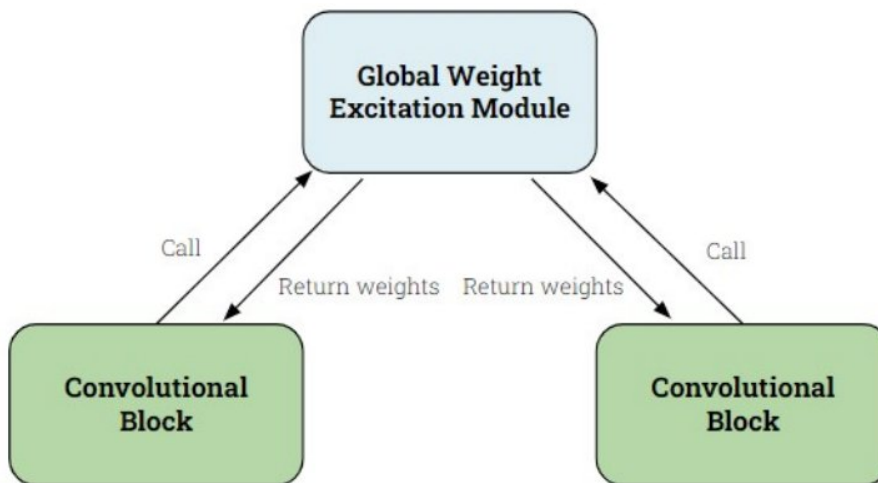


Figure 4.3: Global Weight Excitation

Chapter 5

Datasets

To test the effectiveness of the weight excitation technique, several datasets have been used for different tasks. For Image Classification, Cifar10 [30], Cifar100 [30], and ImageNet [31] datasets have been used. For Semantic Segmentation, the PASCAL VOC [32], and for Action Recognition, the Mini Kinetics dataset [21], and for Gesture Recognition, the Jester dataset [17] has been used.

CIFAR-10

The CIFAR-10 dataset (Canadian Institute for Advanced Research, 10 classes) is a subset of the Tiny Images dataset and consists of 60000 32x32 color images. The images are labeled with one of 10 mutually exclusive classes: airplane, automobile (but not truck or pickup truck), bird, cat, deer, dog, frog, horse, ship, and truck (but not pickup truck). This dataset is divided into five training batches and one test batch. Each of the batches contains 10,000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class. The classes are completely mutually exclusive. The total size of this dataset is 170 mb. We have conducted several experiments on this dataset to see the effects of weight excitation. We have come to the conclusion that this dataset is too small to benefit from weight excitation.

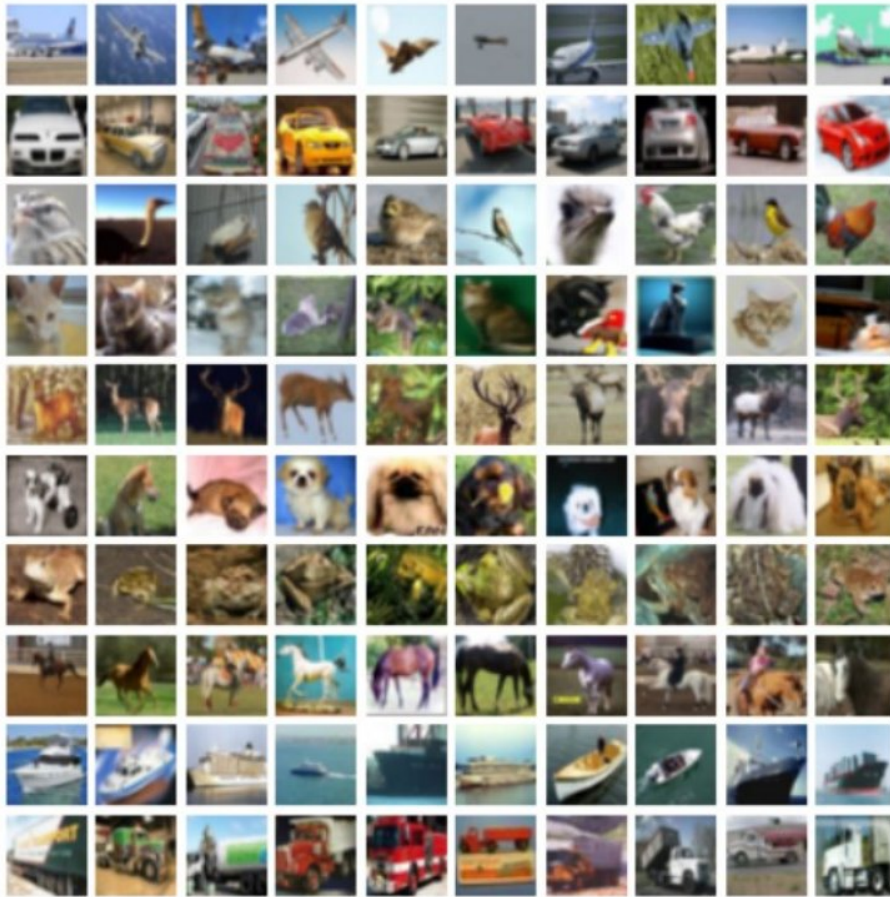


Figure 5.1: Cifar10 Dataset

CIFAR-100

Similar to CIFAR-10, the CIFAR-100 dataset also consists of 60000 32x32 color images but it has 100 classes. These 100 classes in the CIFAR-100 are grouped into 20 superclasses. There are 600 images per class. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class. The classes are completely mutually exclusive. Each image comes with a "fine" label (the actual class to which it belongs) and a "coarse" label (the superclass to which it belongs). The total size of this dataset is 170 mb. Since we did not have enough resources to fully train our model on Imagenet, we carried most of our experiments on this dataset.

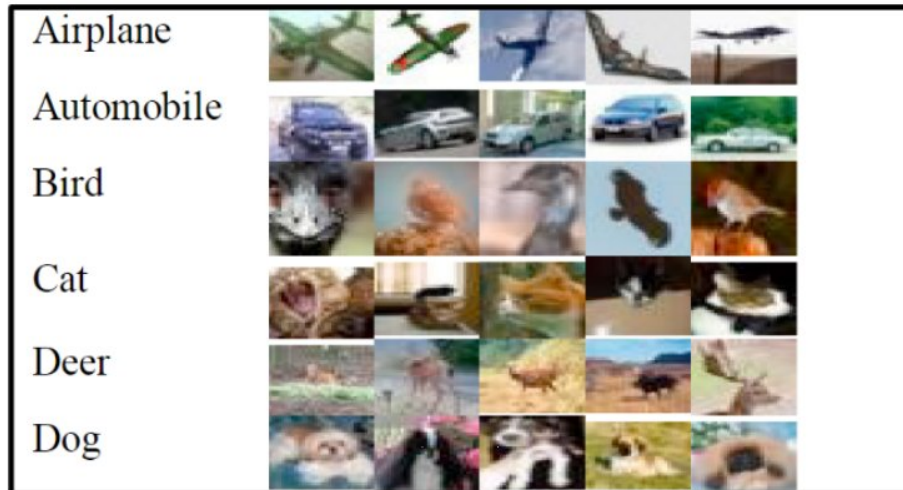


Figure 5.2: Cifar100 Dataset

ImageNet

The ImageNet dataset contains 14,197,122 annotated images according to the WordNet hierarchy. This dataset serves as a reference for object detection and image classification. It is a publicly available dataset that includes a collection of training photographs that have been carefully labeled. Additionally, a set of test photos is made available without manual annotations. The two types of ILSVRC annotations are (1) image-level annotations with a binary label indicating whether or not an object class is present in the picture, and (2) object-level annotations with a small bounding box and a class label enclosing an instance of an object in the image.



Figure 5.3: ImageNet Dataset

PASCAL VOC

The PASCAL Visual Object Classes (VOC) 2012 dataset contains 20 object categories. This dataset has been widely used as a benchmark for object detection, semantic segmentation, and classification tasks. The PASCAL VOC dataset has a total number of 2913 images.

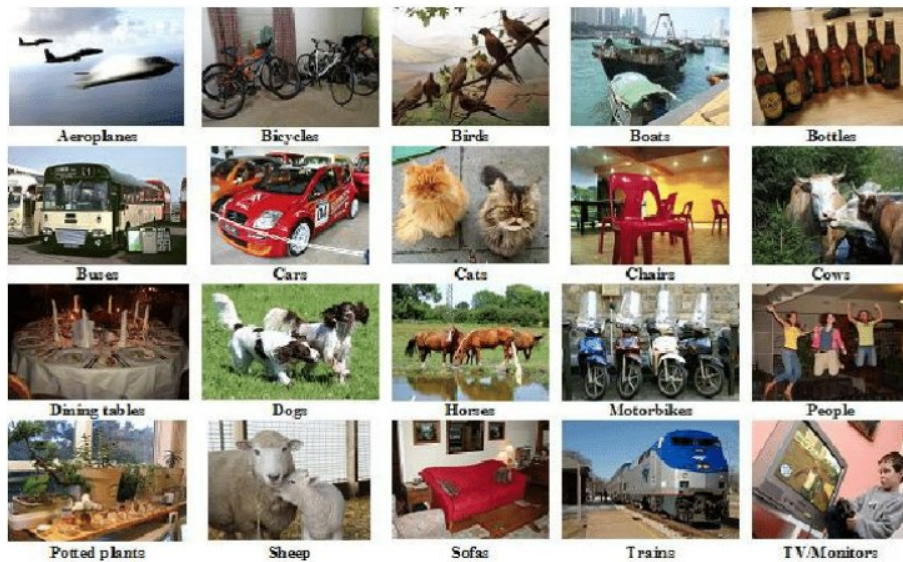


Figure 5.4: PASCAL VOC Dataset

Tiny ImageNet

This dataset contains 100,000 images belonging to 200 classes having 500 images each. All the images are downsized to 64x64 colored images. Each class has 500 training images, 50 validation images, and 50 test images. Since we did not have the resources to train our model on Imagenet dataset for more than 5 epochs, we went for the downsized version of Imagenet. We used this dataset to understand how weight excitation will perform on Imagenet. Using this model we were able to train our model till convergence.

CINIC-10

CINIC-10 dataset contains a combination of images from two benchmark datasets- Cifar-10, and Imagenet. It extends the Cifar-10 dataset by including downsampled images from the Imagenet dataset. This dataset contains 2,70,000 images belonging to 10 classes. The dataset is equally divided into train, test, and validation subsets. Each of the subsets

contains 90,000 images. This dataset is used for performing image classification tasks. It can be used to identify how well the models trained on the Cifar-10 dataset performs on the Imagenet dataset. We have experimented with this model to figure out how well weight excitation works when our dataset combines the features of Cifar-10 and Imagenet both.

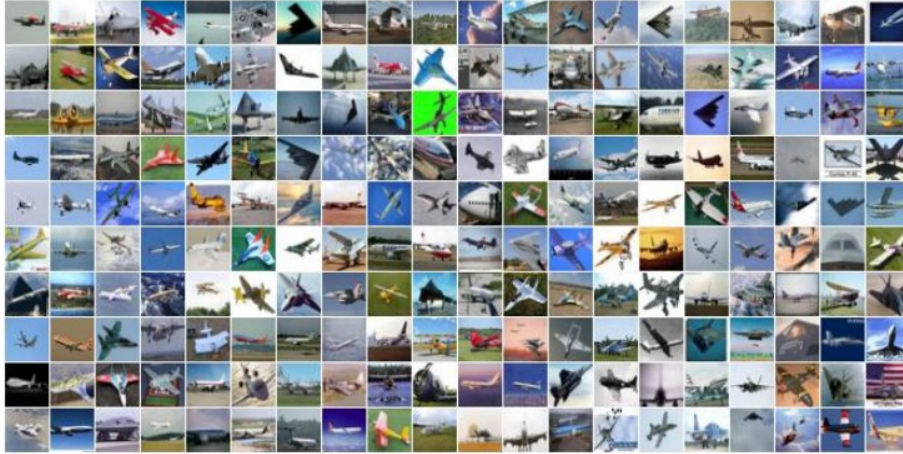


Figure 5.5: C100 Dataset

Chapter 6

Experiments and Result Analysis

6.1 Results for Location-based Weight Excitation using different Attention Modules

6.1.1 Results for Location-based Weight Excitation using SE

We used Resnext architecture as our convolutional neural network model. Our experiments were carried out on the CIFAR-100 dataset. We ran the model for 164 epochs as the model was run for 164 epochs in the original weight excitation paper. Our model reached convergence within 150 epochs. We got 80.2% accuracy on the test set using squeeze and excitation as the attention module. The accuracy in the original paper was 81.5% using the same model architecture. Using different resources for training might have caused the slight change in results.

Table 6.1: Performance Analysis of SE

Architecture	No. of Epochs	Dataset	Accuracy (%)	Comment
ResNext+LWE (SE)	164	Cifar 100	81.5	WE Paper
ResNext+LWE(SE)		Cifar 100	80.2	Experiment

6.1.2 Results for Location-based Weight Excitation using CBAM

We shall now look into the preliminary experiments that have been conducted with our objectives in mind. All our experiments were performed on the Kaggle website. Firstly, we look at the experiments that were conducted to improve the location-based weight excitation mechanism for convolutional neural networks. The original work made use

of a Squeeze and Excitation block for generated kernel-wise attention. Another module named the Convolutional Block Attention Module (CBAM) was experimented in place of the Squeeze and Excitation block. The result showed that the Convolutional Block Attention Module(CBAM) gave better results when tested on the Cifar100 dataset. The number of epochs was limited to 10 due to computational resource limitations.

Table 6.2: Performance Analysis of CBAM

Architecture	Epochs	Dataset	Accuracy (%)
ResNext	10	Cifar100	36.78
ResNext + LWE(SE)	10	Cifar100	37.86
ResNext + LWE(CBAM)	10	Cifar100	39.76

In this experiment, it is clear that the ResNext architecture does very well when paired with CBAM. It is performing much better than ResNext paired with the SE module and vanilla ResNext as well. Now that this combination is doing well in the cifar100 dataset, we intend to train it further for more epochs. Also, we plan to repeat the experiments with the ImageNet dataset for further analysis.

Table 6.3: Performance Analysis of CBAM

Architecture	No. of Epochs	Dataset	Accuracy (%)	Comment
ResNext+LWE (SE)	164	Cifar 100	81.5	WE Paper
ResNext+LWE(SE)		Cifar 100	80.2	Experiment
ResNext+LWE(CBAM)		Cifar100	80.1	Experiment

6.1.3 Results for Location-based Weight Excitation using Cross Dimensional Attention

We know that triple attention module outperforms CBAM and SE module when we use feature maps or images on input. However, when we modified the triple attention module to take weights or convolutional filters as input, we did not see any significant improvement. There was a slight increase in accuracy. Though it can be said that it performed similar to SE and CBAM with less number of parameters. It doesn't help to improve the accuracy much but it certainly helps to make the model lighter.

Table 6.4: Performance Analysis of Rotate to Attend on Cifar 100 Dataset

Architecture	No. of Epochs	Dataset	Accuracy (%)	Comment
ResNext	164	Cifar100	80.5	WE paper + Our Experiment
ResNext+LWE(SE)			81.5	WE Paper
ResNext+LWE(SE)			80.7	Our Experiment
ResNext+LWE (Rotate to Attend)			80.1	Our Experiment

6.2 Performance Analysis of our Novel Location-based Weight Excitation method

The original weight excitation method got an accuracy of 81.5% on Cifar100 dataset when trained upto convergence. But, when we cloned thier experiment in our labs, we got an accuracy of 80.7%. After conducting the experiment with our proposed methodology in the same machine we got an accuracy of 81.63% which is quite an improvement.

Table 6.5: Performance Analysis of our Novel Location-based Weight Excitation method on Cifar 100

Architecture	No. of Epochs	Dataset	Accuracy (%)	Comment
ResNext	164	Cifar100	80.5	WE paper + Our Experiment
ResNext+LWE(SE)			81.5	WE Paper
ResNext+LWE(SE)			80.7	Our Experiment
ResNext+LWE (Ours)			81.63	Our Experiment

We ran the experiment with our proposed novel attention module for 50 epochs in the Cinic10 dataset. Our proposed methodology similar to the original weight excitation method. The hypothesis for this is that the data points of this dataset were very close and are very specific and so, there is not much room for making much changes. The original weight excitation method got an accuracy of 78.52% and our proposed method got an accuracy of 78.53%.

Table 6.6: Performance Analysis of our Novel Location-based Weight Excitation method on Cinic-10

Architecture	No. of Epochs	Dataset	Accuracy (%)
ResNet18	50	Cinic 10	77.96
ResNet18+LWE(SE)			78.52
ResNet18+LWE(Ours)			78.53

We ran the experiment with our proposed novel attention module for 5 epochs in the ImageNet dataset. Our proposed methodology performed better than the original weight excitation method in every epoch. If continued upto convergence, we are hopeful that it will reach convergence faster with a possibility of higher accuracy. The original weight excitation method got an accuracy of 56.312% and our proposed method got an accuracy of 56.966%.

Table 6.7: Performance Analysis of our Novel Location-based Weight Excitation method on Imagenet

Architecture	No. of Epochs	Dataset	Accuracy (%)
ResNet18	5	ImageNet	55.256
ResNet18+LWE(SE)			56.312
ResNet18+LWE(Ours)			56.966

We can see from the graph that our novel attention module performs better than SE module. It gives a smoother learning curve than that of original ResNet18 model.

6.3 Results for Magnitude based Weight Excitation using Novel Activation Function

Significant improvement was seen when we experimented with few number of epochs. This gave us the motivation to train the model till convergence to ensure it actually works. But when we trained the model till convergence, negligible increase in accuracy was observed. The reason behind this can be using a small dataset. We are hopeful that this activation will perform way better if we try it out on Imagenet dataset. Small dataset can't utilize the effect of weight excitation properly like large datasets.

Table 6.8: Performance Analysis of Magnitude Based Weight Excitation

Architecture	Activation for MWE	No. of Epochs	Dataset	Accuracy (%)
ResNext	None	164	Cifar 100	77.961
ResNext+MWE	WE Paper			78.529
ResNext+MWE	Ours			77.719

We have also compared the learning curves of models on which magnitude-based weight excitation has been applied and on which magnitude-based weight excitation has not been applied. If we look at the learning curve of Resnet-18, we can see that this model achieves higher accuracy after using magnitude-based weight excitation. This happens because

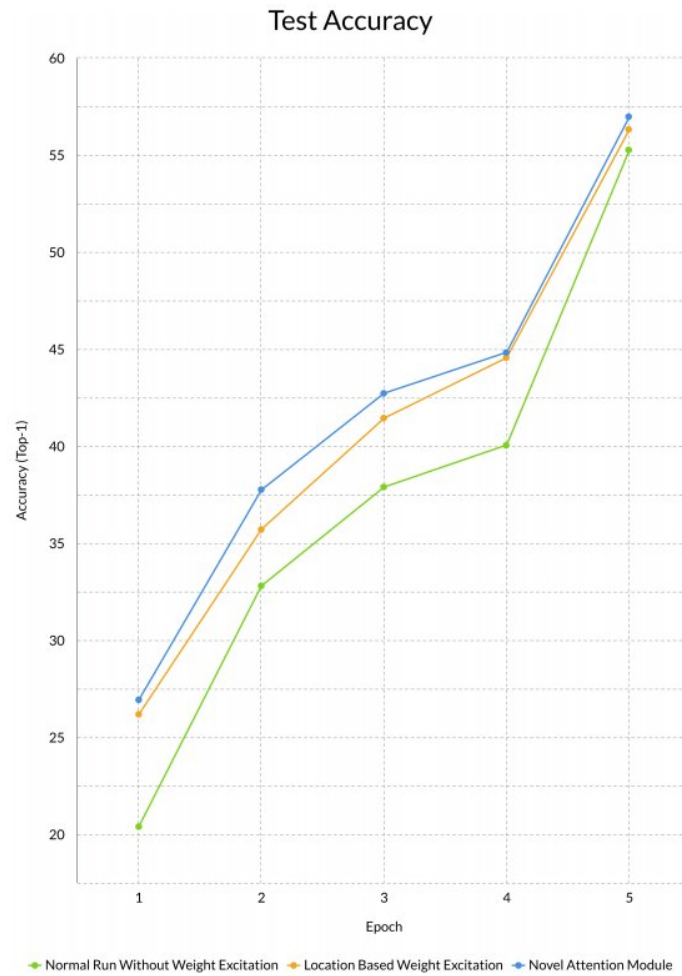


Figure 6.1: Comparison of Original WE and our proposed novel methodology on ImageNet

MWE helps the model to learn faster and better.

We have tried to come up with our own activation function for MWE. Our target was to devise an activation function that will increase the higher magnitude weights a little bit and decrease the lower magnitude weights by a smaller margin. Though we thought this method would give us better results, this resulted in lower accuracy than the original ResNet-18 model. One reason can be that our dataset was not large enough to realize the effects of this model. Another reason can be that weights should be increased and decreased by the same margin.

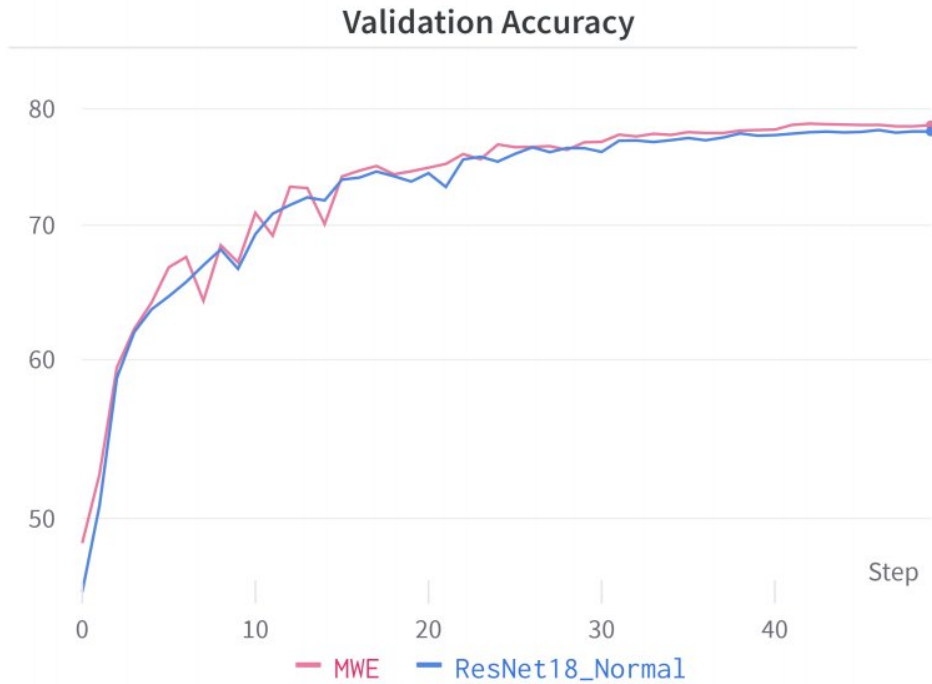


Figure 6.2: Comparison between the learning curve of ResNet18 before and after applying MWE

6.4 Results for Magnitude-based Weight Excitation on MLP Mixer

Now moving on to our next set of experiments, as mentioned in our contributions, we want to extend the concept of weight excitation to MLP Mixers and Vision Transformers. With that view in mind, we tested the magnitude-based weight excitation on MLP Mixers. The idea was that MLP mixer is made up of fully connected layers, which is the same as 1x1 convolution. So if weight excitation worked for convolutional neural networks with 1x1 convolution blocks, it would work for MLP mixers too. The result we obtained are as follows in the case of the cifar100 dataset. In the original weight excitation paper, the test was carried out on the Imagenet dataset and improvements were seen.

Table 6.9: Performance Analysis of Weight Excitation on MLP Mixer

Attempt	Epochs	Accuracy without MWE (%)	Accuracy with MWE (%)
01	30	54.27	54.63
02	30	53.71	54.74
03	30	53.50	55.42

The table shows that on multiple attempts, we are seeing improvement in accuracy. But

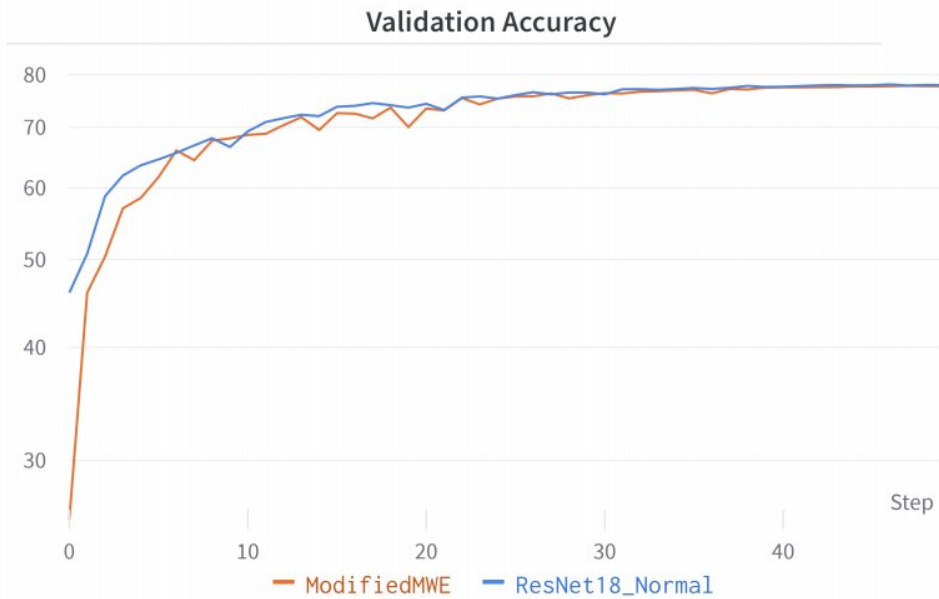


Figure 6.3: Comparison between the learning curve of ResNet18 before and after applying Modified MWE

the problem is that the model in general overfits the cifar 100 dataset. This makes it difficult to say whether the improvement in accuracy will hold on in the case of other datasets.

We also conducted the same test on the cifar 10 dataset.

In this case, the overfitting does not occur but the increase in accuracy is not found. This was found to be a problem with the dataset itself. Cifar10 is a very small dataset and the mlp mixer model that is being used seems to be tailored for cifar 10 only. This is why inconsistencies in the result are being observed.

After considering the result of all our experiments on MLP Mixer, we can conclude that it is not possible to successfully apply weight excitation on MLP Mixers. The reason behind these are as follows-

- In convolutional neural networks, the weight kernels have channels. This is the reason why applying location importance maps on the channels gives us better results. But MLP only contains fully connected layers that do not have any channels. So, the intuition of applying channel attention to the weight kernels can't be applied here. That is the reason we only tried applying magnitude-based weight excitation on MLP mixers. Though we are able to apply MWE to MLP Mixers, the results were not that satisfactory.

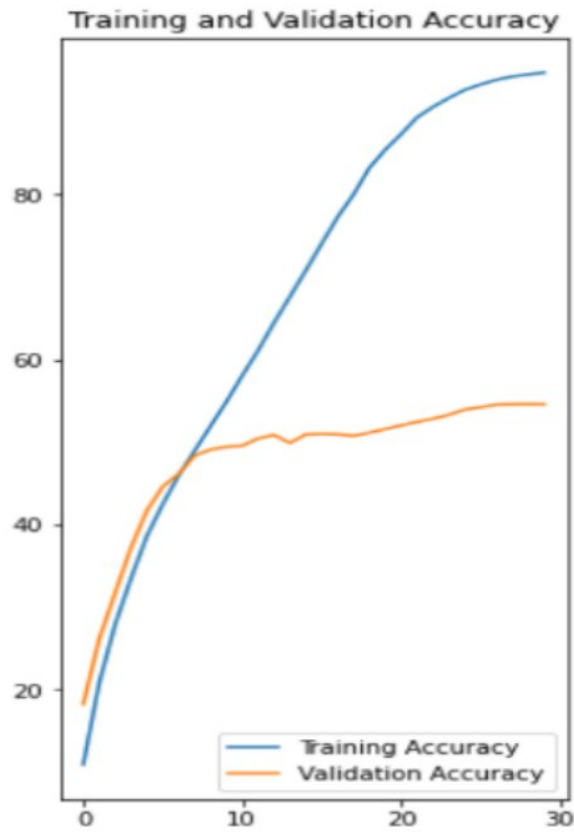


Figure 6.4: Training and Validation Accuracy of MLP Mixer on Cifar100 without MWE

- MLP Mixer works properly only when it is pre-trained on very large datasets like JFT-300M and Imagenet-21k. Otherwise, the original model faces problems of overfitting. Since we do not have the resources to fully train our models on these very large datasets, these experiments are out of our scope.

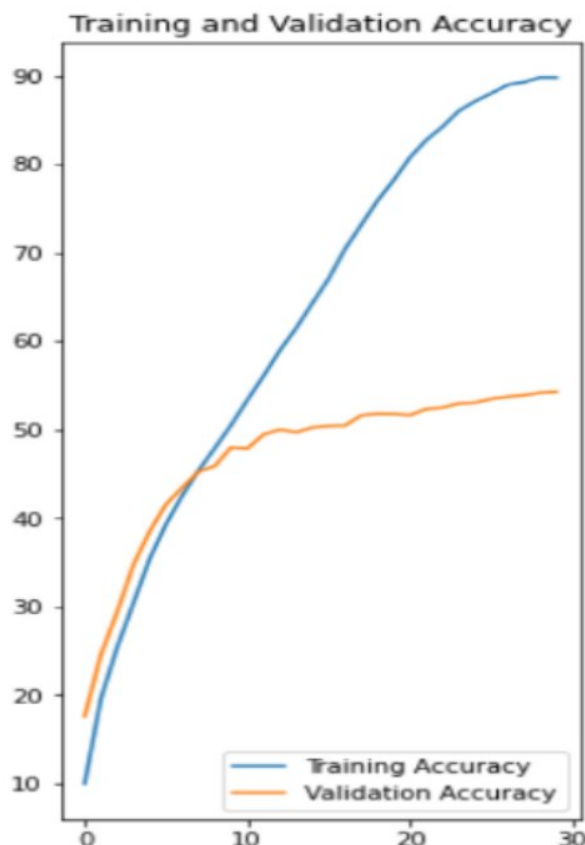


Figure 6.5: Training and Validation Accuracy of MLP Mixer on Cifar100 with MWE

6.5 Experiment using different Activation Functions in SE module

In this experiment, we have used different combinations of different activation functions for the Squeeze and Excitation module to compare the accuracies. Better accuracy was obtained when Leaky-Relu was used in place of the Relu activation function. The intuition behind this is that, in a neural network, the neurons stop learning when the value of their weights becomes less than or equal to zero when the Relu activation function is used. This limitation of Relu is handled by the Leaky-Relu activation function which allows the neurons to learn even after their weight becomes less than or equal to zero. Using of Leaky-Relu brings slight changes to the squeeze and excitation module. The modified squeeze and excitation module performs better than the others and original combinations as shown in the table. This also signifies that using squeeze and excitation module with Leaky-Relu will increase the performance of any model in general. We will be conducting further experiments on this. We can also draw a hypothesis that using this combination in other attention modules like CBAM and BAM can theoretically give higher accuracy as

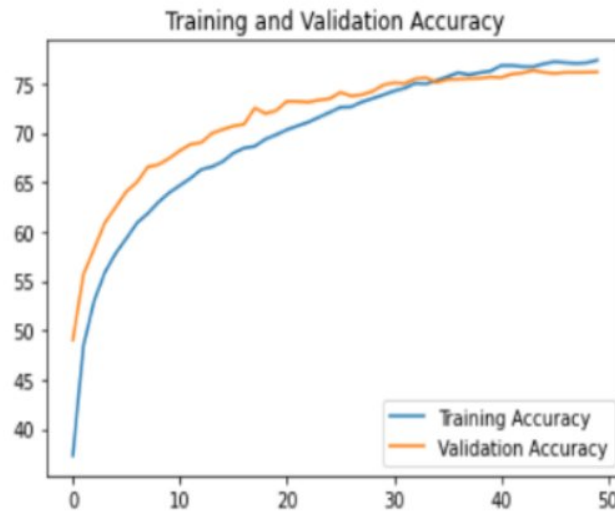


Figure 6.6: Training and Validation Accuracy of MLP Mixer on Cifar10 without MWE

Table 6.10: Experiment using different Activation Functions in SE module

Architecture	Weight Excitation	Activation Function	Accuracy (%)
ResNet18	None	None	80.75
	LWE	ReLU + Sigmoid	82.01
		Swish + ReLU	78.72
		ReLU + Tanh	80.68
		ReLU+Swish	80.72
		Swish+Swish	80.84
		Leaky Relu + Tanh	81.03
		Swish + Sigmoid	82.40
		Leaky ReLU+ Sigmoid	82.74

well. Other activation functions apart from leaky relu was tried as well. other than Swish and Leaky-Relu, none was better than the originally used Relu activation function.

6.6 Performance Analysis of Global Weight Excitation Method

We can see from our experiments that applying global weight excitation to ResNet-18 resulted in better accuracy but it could not beat the performance of the original location-based weight excitation method.

If we look at the learning curve of Global Location based weight excitation on Imagenet, we see that its learning curve converges faster than the original ResNet18 model, But it can not perform better than the existing weight excitation method.



Figure 6.7: Training and Validation Accuracy of MLP Mixer on Cifar10 with MWE

Table 6.11: Performance analysis of Global Weight Excitation on Imagenet

Architecture	No. of Epochs	Dataset	Accuracy (%)
ResNet18	5	ImageNet	55.256
ResNet18+LWE(SE)			56.312
ResNet18+LWE(Ours)			55.944

6.7 Comparison between Weight Excitation and Regularization

We know that regularization helps to decrease the difference between the loss curve of training and the validation set. Another noticeable change is that regularization smoothens the model's learning curves. The reason behind these effects is that regularization reduces the overfitting of learning curves and helps the model to learn better by tuning parameters. Regularization solves the overfitting and underfitting issue of models which is why we see that the loss curve of the training and validation set becomes closer.

When we plotted the learning curves of models with excitation and compared them to the models on which weight excitation has not been applied, we saw that the learning curve of the models with weight excitation seemed a lot smoother. So, from this, we came to the conclusion that the effect of applying weight excitation on models is quite similar to the effect of applying regularization. Both of the experiments result in smoother learning curves. Though the Weight Excitation method smoothens the learning curve, it can not be used as an alternative to regularization. In Figure 6.6, we can see that the learning curve

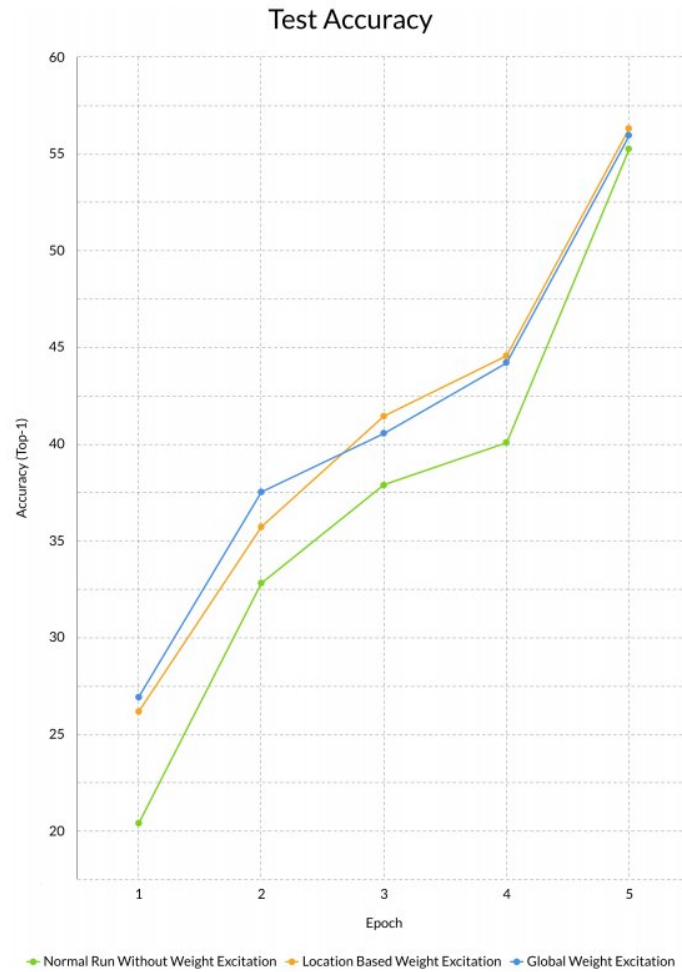
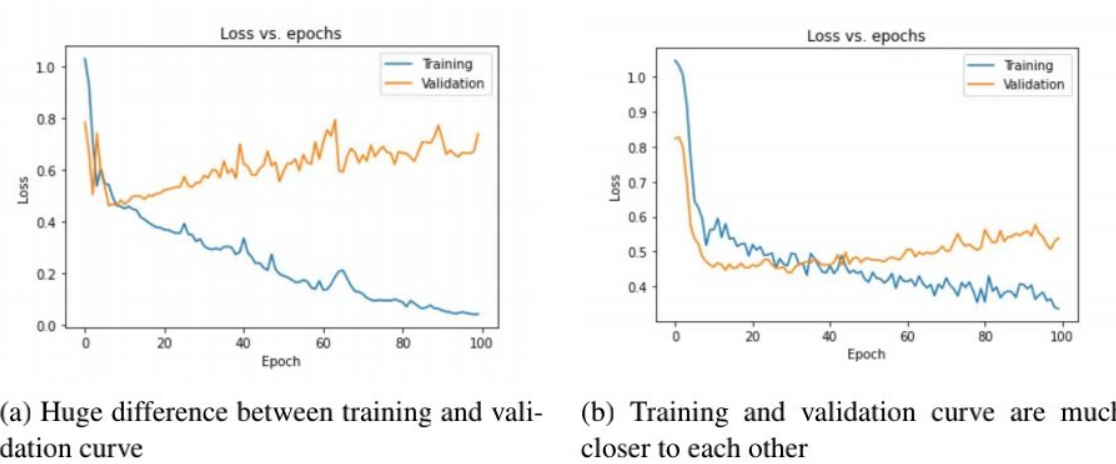


Figure 6.8: Performance analysis of Global LWE

of the Resnet-18 model fluctuates a lot. After applying location-based weight excitation to the model, the learning curve becomes much smoother. And we can see that the performance of ResNet-18 with weight excitation is better than the performance of normal ResNet-18 throughout the whole learning period. The learning curve is more stable and it converges faster than the original one.



(a) Huge difference between training and validation curve

(b) Training and validation curve are much closer to each other

Figure 6.9: The effect of applying Regularization on training curve

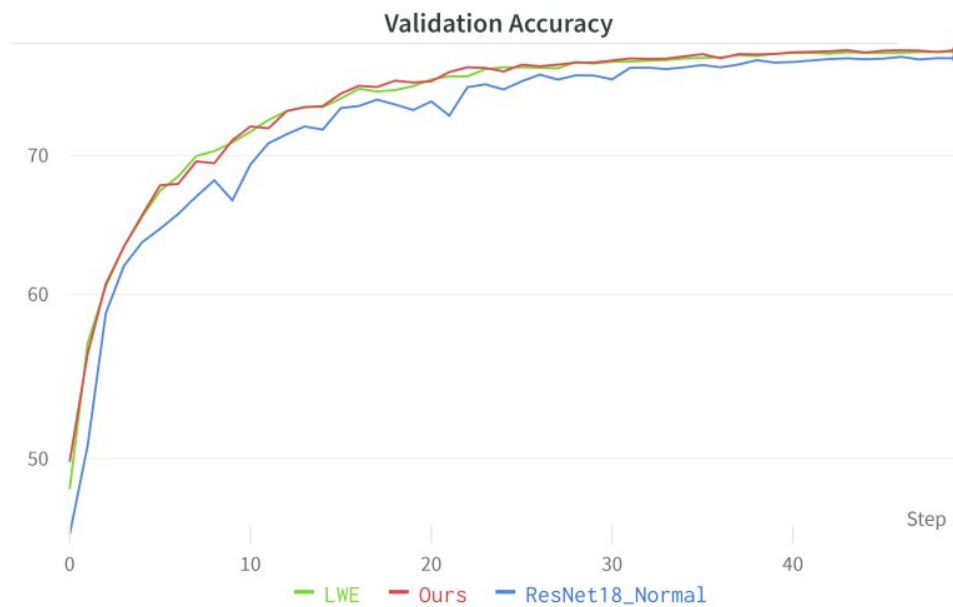


Figure 6.10: Weight Excitation has a similar effect on the learning curve as regularization

Chapter 7

Contributions

The contributions that we made in this work -

1. **Implementation of magnitude based weight excitation for the MLP Mixer architecture:** As discussed before, weight excitation's current approach is limited to convolutional neural networks only. We have expanded the concept to MLP mixer which are a new kind of model with weight layers consisting of fully connected layers only. That being said, the increase in accuracy by using weight excitation in MLP mixer can undoubtedly be used in other domains as well.
2. **Implemented new location based weight excitation approach:** Inspired by the CBAM module for attention in feature maps, we have designed a new weight excitation module that performs better than the existing module. We took the smart design choices from different modules and put these all in one place, at the same time added new ideas that we tested to have done better in other places.
3. **Improving accuracy of existing location based weight excitation:** With our new weight excitation module, we have successfully improved the performance of weight excitation. Our results on various datasets are a testament to our contribution.
4. **Finding problem with Weight Excitation method and proposing solution:** We have found some problems with the existing methods of weight excitation when applied to small datasets. More research work is needed to bring an effective method of weight excitation for small datasets.
5. **Weights as an alternative to input feature maps:** We have conducted several experiments to conclude whether weights can be used as an alternative to input

feature maps for computing attention and if this applies to all existing attention mechanisms for Convolutional Neural Networks.

6. **Introduction of global weight excitation:** We implemented Weight Excitation block as a separate module which can be called by convolutional layers as needed. This ensures the model to learn global representations of the weight. One of the next research endeavours can be that in which ways can global and local weight excitations can interact with each other in creating a more robust feature map.

Chapter 8

Future Works and Conclusion

There are some more experiments we plan to do for providing a more solid use of our work. We will implement the proposed methodologies on ImageNet upto convergence to get a clear idea about the affect of our work in large datasets.

Next, we will try to improve the previous magnitude weight excitation method by changing the weight excitation function to some learnable function. And finally, we will try to implement the weight excitation for the transformers.

There are a lot of experiments we are planning to do for improving the weight excitation effectiveness. The first idea is stacked weight excitation. So, the idea is that the weights in conv block in weight excitation goes through another weight excitation and so forth. This means that the weights that are already present in the conv block will go through another conv block and continue for a certain number of times. Hopefully, this will result in better accuracy as the internal weights of the conv block to get excited as well.

Our second idea is that the current weight excitation method starts exciting the weights from the first epoch. If the weights are not given a chance to converge before applying weight excitation, then the method cannot recognize the important weights properly. So, what we want to do is to pause the weight excitation for the initial epochs and allow the weights to converge by themselves at first, and then apply weight excitation in the later epochs. This should yield a better result according to our intuition as the important weights will be recognized better and focused more.

Thirdly, we will use the weights from a pre-trained model that already has well-learned weights in our weight excitation module which will complement our second idea.

Fourthly, as we have already used the design of CBAM/BAM in our weight excitation module and got better results, we are also planning to use a better submodule design to

improve further.

Next, we will try to improve the previous magnitude weight excitation method by changing the weight excitation function to some learnable function. And finally, we will try to implement the weight excitation for the transformers.

In conclusion, we will be devoting our next experiments for further improvement of weight excitation and extension of weight excitation to others vision architectures.

References

- [1] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.
- [2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [3] H. J. Kelley, “Gradient theory of optimal flight paths,” *Ars Journal*, vol. 30, no. 10, pp. 947–954, 1960.
- [4] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [5] S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network,” *Advances in neural information processing systems*, vol. 28, 2015.
- [6] N. Quader, M. M. I. Bhuiyan, J. Lu, P. Dai, and W. Li, “Weight excitation: Built-in attention mechanisms in convolutional neural networks,” in *European Conference on Computer Vision*, pp. 87–103, Springer, 2020.
- [7] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [9] I. O. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. Steiner, D. Keysers, J. Uszkoreit, *et al.*, “Mlp-mixer: An all-mlp architecture for vision,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 24261–24272, 2021.

- [10] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.
- [11] Y. LeCun, J. Denker, and S. Solla, "Optimal brain damage," *Advances in neural information processing systems*, vol. 2, 1989.
- [12] B. Hassibi and D. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," *Advances in neural information processing systems*, vol. 5, 1992.
- [13] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.
- [14] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *arXiv preprint arXiv:1611.06440*, 2016.
- [15] J. Liu, Z. Xu, R. Shi, R. C. Cheung, and H. K. So, "Dynamic sparse training: Find efficient sparse network from scratch with trainable masked layers," *arXiv preprint arXiv:2005.06870*, 2020.
- [16] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "Bam: Bottleneck attention module," *arXiv preprint arXiv:1807.06514*, 2018.
- [17] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *information retrieval*, vol. 4, no. 2, pp. 133–151, 2001.
- [18] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AICHE journal*, vol. 37, no. 2, pp. 233–243, 1991.
- [19] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.
- [20] D. Misra, T. Nalamada, A. U. Arasanipalai, and Q. Hou, "Rotate to attend: Convolutional triplet attention module," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3139–3148, 2021.
- [21] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 305–321, 2018.

- [22] S. Qiao, H. Wang, C. Liu, W. Shen, and A. Yuille, “Micro-batch training with batch-channel normalization and weight standardization,” *arXiv preprint arXiv:1903.10520*, 2019.
- [23] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” *arXiv preprint arXiv:1802.05957*, 2018.
- [24] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, “Dynamic convolution: Attention over convolution kernels,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11030–11039, 2020.
- [25] C. Li, A. Zhou, and A. Yao, “Omni-dimensional dynamic convolution,” *arXiv preprint arXiv:2209.07947*, 2022.
- [26] S. Haykin, “Neural networks,” *A comprehensive foundation*, 1994.
- [27] J. Nagi, F. Ducatelle, and G. Di Caro, “Max-pooling convolutional neural networks for vision-based hand gesture recognition. proceedings of the ieee international conference on signal and image processing applications (icsipa),” 2011.
- [28] S. Pereira, A. Pinto, J. Amorim, A. Ribeiro, V. Alves, and C. A. Silva, “Adaptive feature recombination and recalibration for semantic segmentation with fully convolutional networks,” *IEEE transactions on medical imaging*, vol. 38, no. 12, pp. 2914–2925, 2019.
- [29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [30] A. Krizhevsky, G. Hinton, *et al.*, “Learning multiple layers of features from tiny images,” 2009.
- [31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.
- [32] Z. Dong, K. Xu, Y. Yang, H. Bao, W. Xu, and R. W. Lau, “Location-aware single image reflection removal,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5017–5026, 2021.