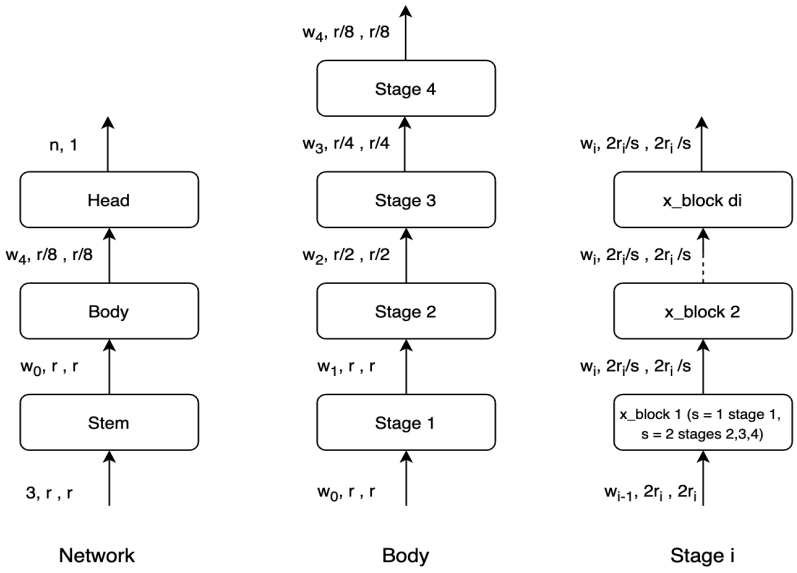
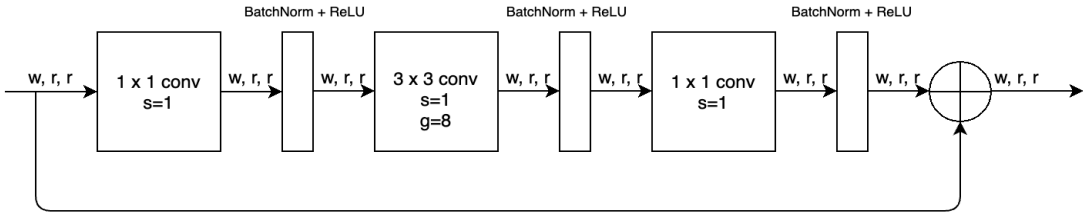
1. Design

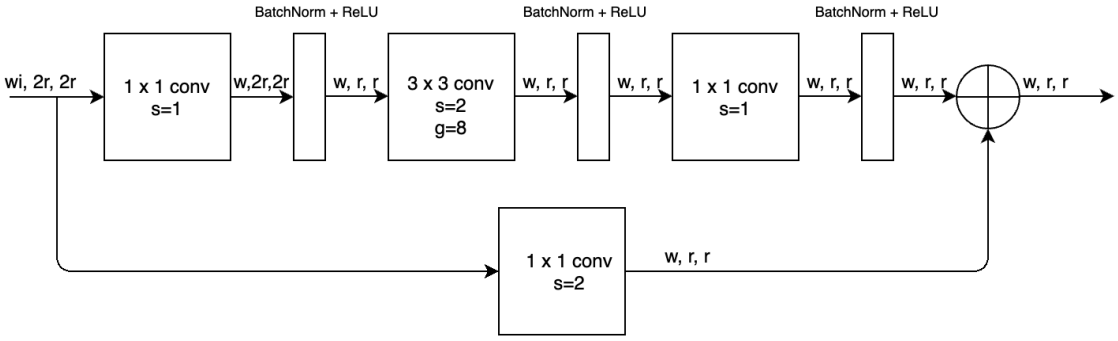
A. Generic Network Structure:



B. Standard Building Block (X Block, s=1):



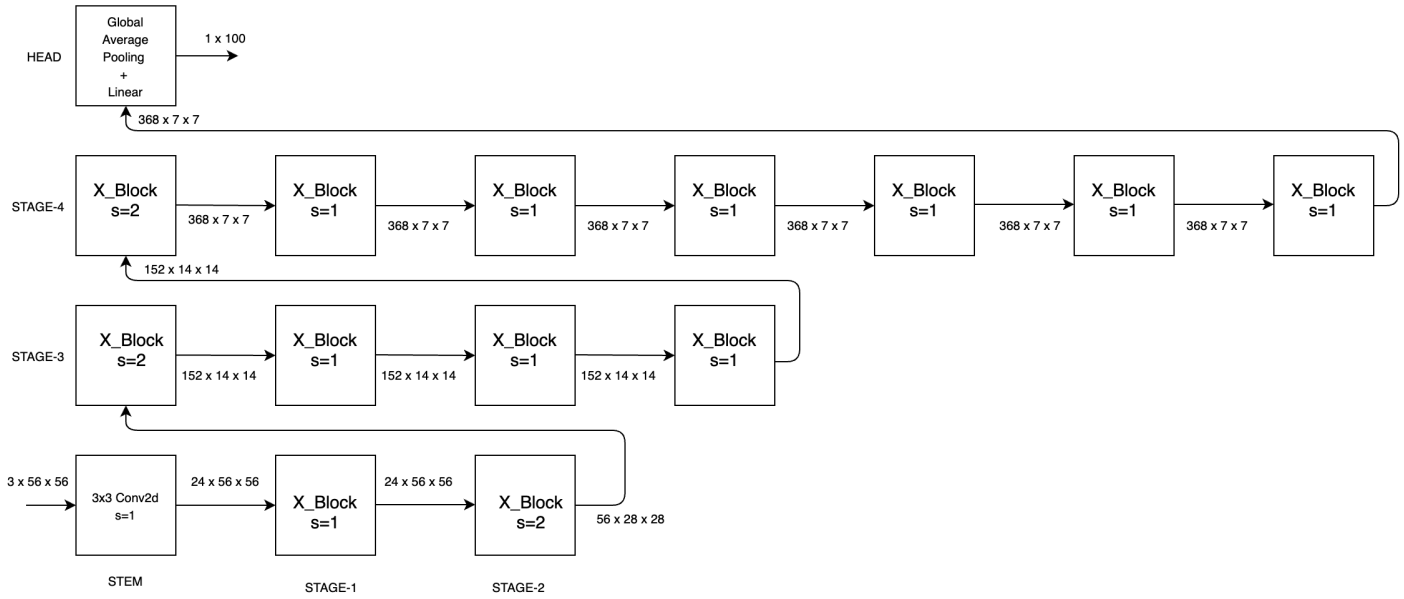
C. Downsampling Building Block (X Block, s=2):



D. RegnetX 200MF Table and Final Network

Parameter	RegnetX 200 MF Value	Description
Stem	3x3 Conv2d, s =1, Ni = 3, No =24	Converts 3 x 56 x 56 to 24 x 56 x 56. Plug in w_0 = 24 in generic network structure
Stages	4	Encoder body has multiple stages
Depth	[1,1,4,7]	Number of blocks per stage. Plug in d_1=1, d_2=1, d_3=4, d_4=7 in generic network structure
Channels	[24,56,152,368]	Number of output channels per stage. Plug in w_1=24, w_2=56, w_3=152, w_4=368 in generic network structure
Group width	8	Channels in 3x3 Conv2d blocks in encoder body are divided into groups to reduce number of trainable parameters
X Block Stride	s=1, s=2	s=2 X block is the downsampling block in stages:2,3,4 s=1 X block is the repeat block in all stages
Head	Global Average Pool + linear	Converts 368 x 7 x 7 to 1 x 100

Final Modified RegnetX 200MF:

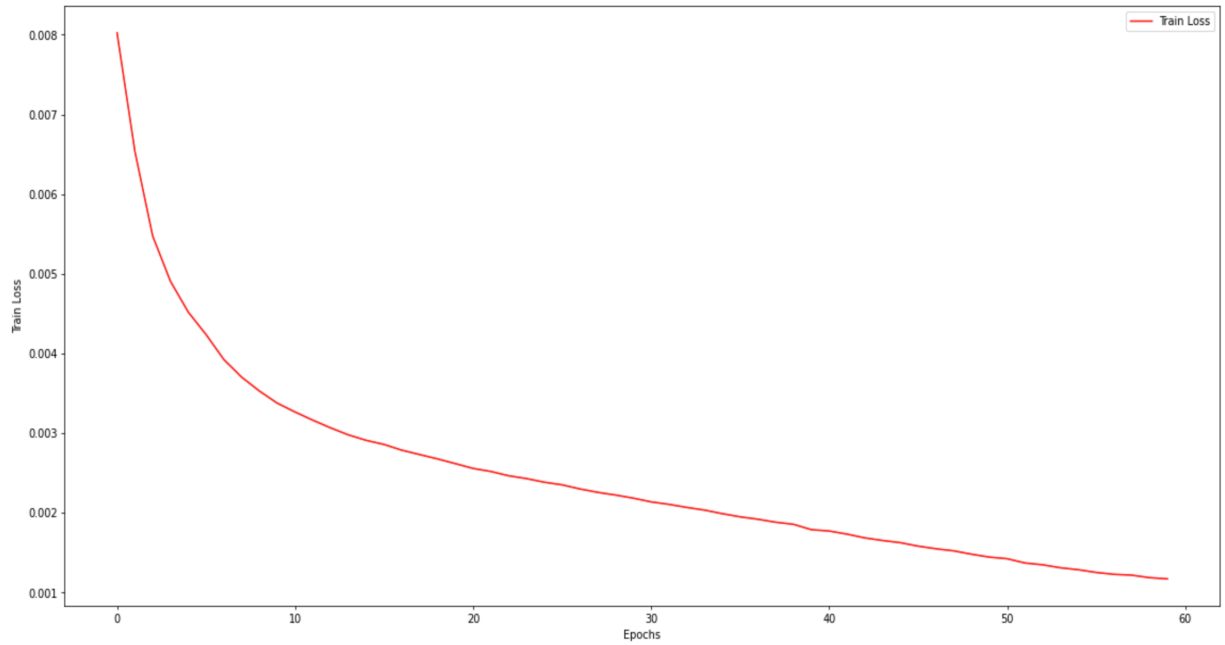


2. Training

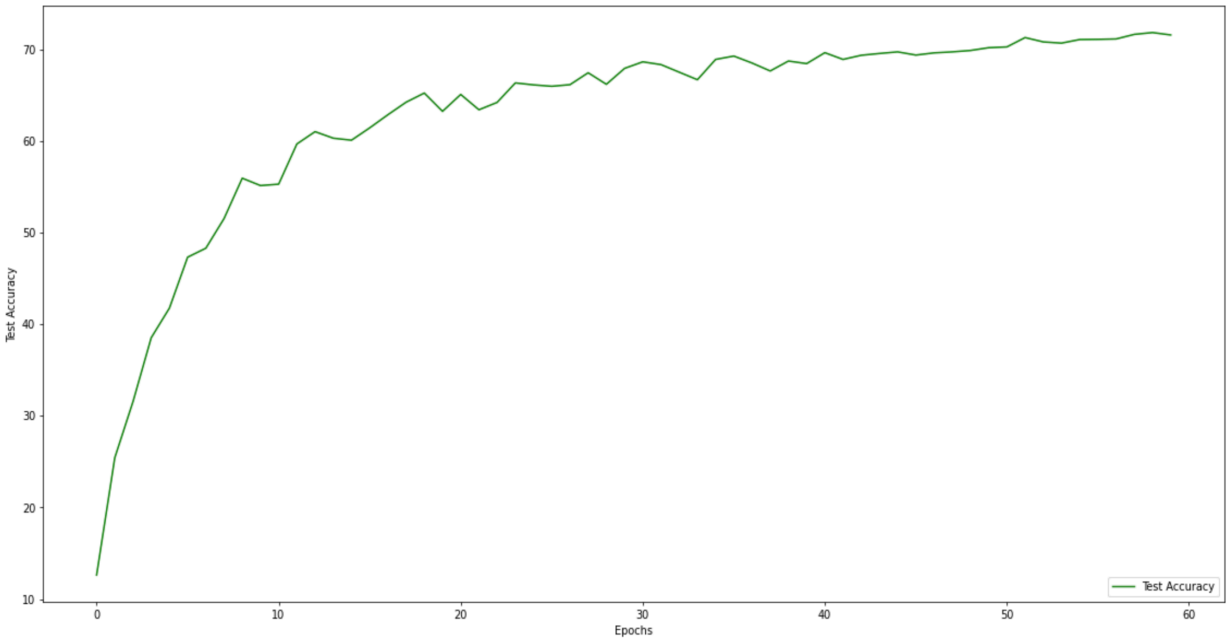
A. Hyperparameter table:

Hyperparameter	Value
DATA_BATCH_SIZE	512
DATA_NUM_CHANNELS	3
DATA_NUM_CLASSES	100
MODEL_TAIL_OUT_CHANNELS	24
MODEL_BLOCKS	[1,1,4,7]
MODEL_CHANNELS	[24,56,152,368]
MODEL_GROUP_WIDTH	8
MODEL_FILTER_SIZE	3
TRAINING_LR_FINAL	0.0001
TRAINING_LR_INIT_EPOCHS	5
TRAINING_LR_FINAL_EPOCHS	55
FILE_SAVE	1

B. Training Loss vs. Epoch



C. Test Accuracy vs. Epoch



D. Final Accuracy

Final Test Accuracy = 71.55%

3. Implementation

grp_width = 8 for 3x3 conv2d in Encoder body

num_grps = Ni/grp_width for 3x3 conv2d of Encoder body (num_grps =1 for all other conv2ds)

Parameters = Di * (Ni*No*Fr*Fc)/(num_grps) where Di = Depth of Stage i.

MACs = Di * (Ni*No*Fr*Fc*Mr*Mc)/(num_grps) where Mr, Mc are output feature map dims.

Conv2d	Di	Ni x No x Fr x Fc	Mr x Mc	Num_grps	Parameters	MACs
Stem, 3x3	1	3 x 24 x 3 x 3	56 x 56	1	648	2,032,128
Stage1, 1x1	1	24 x 24 x 1 x 1	56 x 56	1	576	1,806,336
Stage1, 3x3	1	24 x 24 x 3 x 3	56 x 56	3	1,728	5,419,008
Stage1, 1x1	1	24 x 24 x 1 x 1	56 x 56	1	576	1,806,336
Stage2, 1x1	1	24 x 56 x 1 x 1	56 x 56	1	1,344	4,214,784
Stage2, 3x3	1	56 x 56 x 3 x 3	28 x 28	7	4,032	3,161,088
Stage2, 1x1	1	56 x 56 x 1 x 1	28 x 28	1	3,136	2,458,624
Stage3 ,1x1	4	56 x 152 x 1 x 1	28 x 28	1	34,048	26,693,632
Stage3 ,3x3	4	152 x 152 x 3 x 3	14 x 14	19	43,776	8,580,096
Stage3 ,1x1	4	152 x 152 x 1 x 1	14 x 14	1	92,416	18,113,536
Stage4, 1x1	7	152 x 368 x 1 x 1	14 x 14	1	391,552	76,744,192
Stage4, 3x3	7	368 x 368 x 3 x 3	7 x 7	46	185,472	9,088,128
Stage4, 1x1	7	368 x 368 x 1 x 1	7 x 7	1	947,968	46,450,432
Total =					1,707,272	206,568,320

4. Extra

A. Created and compared performance and accuracy of different modified RegnetX variations. This can be accomplished by setting the parameter REGNETX in extra.py for each variation.

RegnetX	Input Image	Batch Size	Time/Epoch (s)	Test Accuracy (%)
200 MF	3 x 56 x 56	512	62	71.55
400 MF	3 x 56 x 56	512	94	72.05
600 MF	3 x 56 x 56	256	108	73.07
800 MF	3 x 56 x 56	256	143	73.48
1.6 GF	3 x 64 x 64	128	308	75.94

Common parameters:
Total Epochs = 60
Final Learning rate = 0.0001
Colab GPU runtime : Tesla V100-SXM2-16GB

B. Studied the effect of group width on performance and accuracy of a custom RegnetX
Custom RegnetX = Modified 200MF RegnetX with MODEL_CHANNELS = [32, 64, 128, 256].
This can be accomplished by setting the parameter REGNETX to “custom” in extra.py.

Group Width	Batch Size	Time/Epoch (s)	Test Accuracy (%)
1	512	44	69.29
2	512	45	69.31
4	512	51	68.71
8	512	53	69.05
16	512	49	69.62
32	512	49	68.92

Common parameters:
Total Epochs = 30
Final Learning rate = 0.0001
Colab GPU runtime : Tesla V100-SXM2-16GB