CMSC733 Homework 0 Report

Pyone Win
Department of Computer Science
University of Maryland, College Park
College Park, Maryland
Email: pwin17@umd.edu

I. Phase 2

A. Section 3.3

As per instructions, I constructed a basic neural network that consists 7 layers. The model takes the inputs in the input layer, then data is processed in two pairs of a conv2D layer and a max pooling layer. Then, the layer output is flattened and the final output was produced through the fully connected layer. Both convolution layers has (3,3) kernel size, has same padding-type and RELU activation is applied. Filter sizes used were 8 and 16 respectively. Max pooling layers has (2,2) pool size. The final dense layer is applied softmax activation. The model was trained for 20 epochs and training set accuracy of 68.96% and testing set accuracy of 65.74% was achieved at 17th epoch. The model has 11,642 parameters. Adam optimizer and categorical cross entropy were used in the model. The learning rate was 0.001 and batch size was 36.

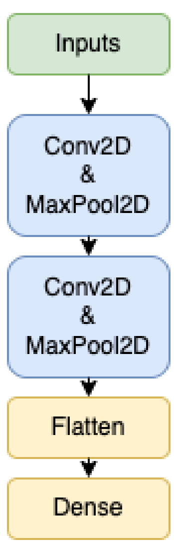

Fig. 1. Architecture of a basic model

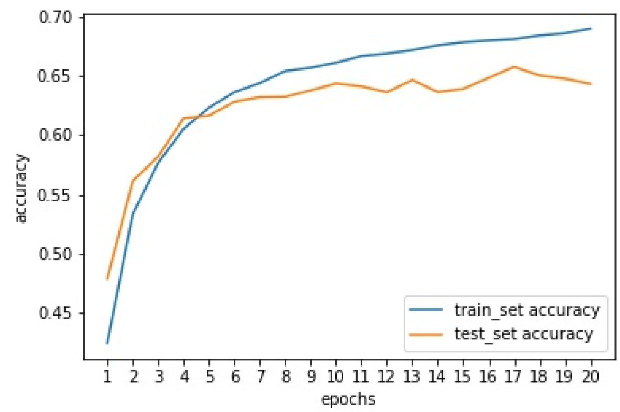

Fig. 2. Accuracy over epochs of the basic model

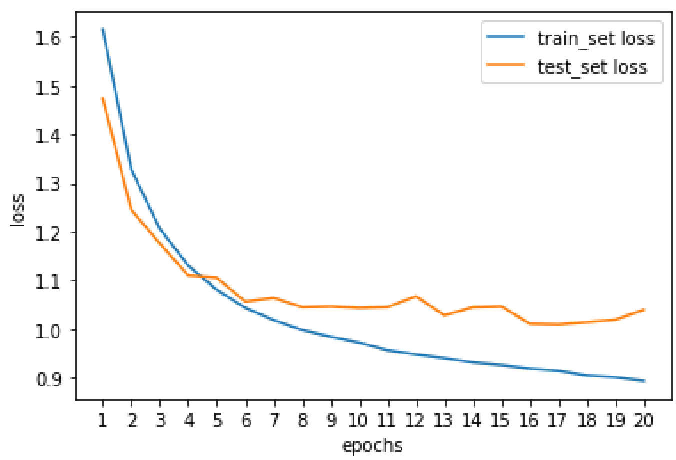

Fig. 3. Loss over epochs of the basic model

B. Section 3.4

To improve the basic model, I added batch normalization layer between convolution layers and max pooling layers. I have also three more layers of convolution layer, batch normalization layer and max pooling layer. For this model, I was able to achieve 79.85% and 72.19% accuracies for the training and testing sets respectively within only 10 epochs. I used batch size of 64 and utilized the same optimizer and loss functions as the basic model. I have also tried data augmentation by randomly taking 15% of the training set and 15% of the test set data to perform horizontal flip, vertical flip and random crop. I found that the results were quite similar with the data augmented version producing about 0.3% better accuracy. This might be due to less amount of data I augmented. It might also be due to the simplicity of the model.

	0	1	2	3	4	5	6	7	8
0	3542	104	288	54	118	47	77	39	590
1	96	4131	45	22	33	21	84	15	190
2	256	33	3010	145	507	236	580	66	119
3	112	50	424	1947	337	837	1017	79	139
4	138	25	455	169	3197	159	599	157	76
5	44	28	416	534	285	2977	484	132	63
6	24	24	192	77	120	61	4440	7	35
7	81	23	292	211	439	354	155	3330	48
8	262	176	62	30	42	24	66	14	4236
9	188	536	61	64	23	46	127	37	194

TABLE I

CONFUSION MATRIX OF THE TRAINED BASIC MODEL ON TRAINING DATA

	0	1	2	3	4	5	6	7	8	9
0	673	18	77	7	27	5	21	8	131	33
1	31	774	15	6	7	7	13	4	40	103
2	60	4	536	39	104	70	127	29	20	11
3	23	16	200	338	77	266	210	22	28	20
4	20	3	99	49	601	31	136	34	20	6
5	15	5	76	134	74	553	81	32	20	10
6	7	4	43	24	30	11	867	1	11	2
7	22	7	63	45	97	81	37	621	9	18
8	69	52	15	7	11	6	11	4	805	20
9	44	136	15	13	9	10	32	20	59	662

TABLE II

CONFUSION MATRIX OF THE TRAINED BASIC MODEL ON TESTING DATA

This model has 122570 parameters.

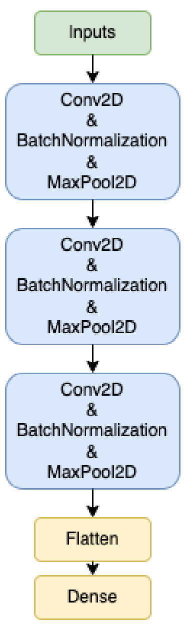

Fig. 4. Architecture of an improved model

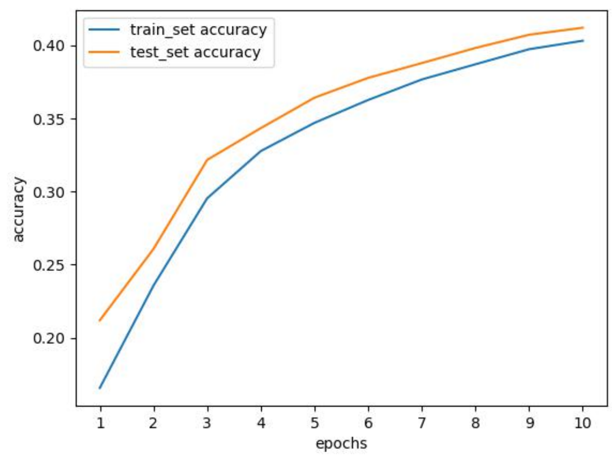

Fig. 5. Accuracy over epochs of the improved model

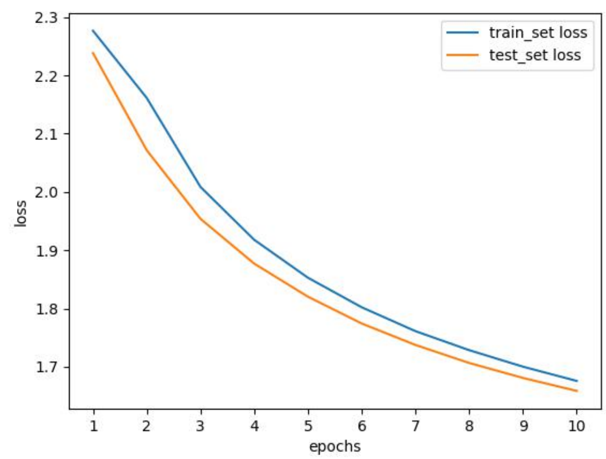

Fig. 6. Loss over epochs of the improved model

	0	1	2	3	4	5	6	7	8	9
0	800	30	37	14	26	8	3	23	27	32
1		9 857	8	4	7	8	7	8	16	76
2	66	7 636	45	90	68	29	46	5	8	
3	33	14	82 479	65 212	43	55	5	12		
4	17	4	66	49 718	48	18	73	6	1	
5	21	1	51 113	36 694	17	63	1	3		
6		5	8	63	66	62	43 721	21	6	5
7	10	0	31	25	46	44	1 839	0	4	
8	112	62	20	12	16	8	2	12 731	25	
9	21	90	13	14	4	10	5	31	11 801	

TABLE III

CONFUSION MATRIX OF THE TRAINED OPTIMIZED MODEL ON TRAINING DATA

C. Section 3.5

1) ResNet Model: I used ResNet34 model to train CIFAR10 dataset. I followed the implementation from the paper and also used the parameters mentioned in the paper. My resnet model has 2168278 parameters and 2154576 of them were trainable parameters. For this model, I trained for 50 epchs and I was able to achieve 100% and 79.10% accuracies for the training and testing sets respectively at 31st epoch. The results were similar after that. I used Stochastic Gradient Descent optimizer for the model. The learning rate was scheduled to be exponential decay. The initial learning rate was 0.1 and it decreased by 0.5 every 10000 steps. Batch size used was 128.

	0	1	2	3	4	5	6	7	8	9
0	673	18	77	7	27	5	21	8	131	33
1	31	774	15	6	7	7	13	4	40	103
2	60	4	536	39	104	70	127	29	20	11
3	23	16	200	338	77	266	210	22	28	20
4	20	3	99	49	601	31	136	34	20	6
5	15	5	76	134	74	553	81	32	20	10
6	7	4	43	24	30	11	867	1	11	2
7	22	7	63	45	97	81	37	621	9	18
8	69	52	15	7	11	6	11	4	805	20
9	44	136	15	13	9	10	32	20	59	662

TABLE IV
CONFUSION MATRIX OF THE TRAINED OPTIMIZED MODEL ON TESTING DATA

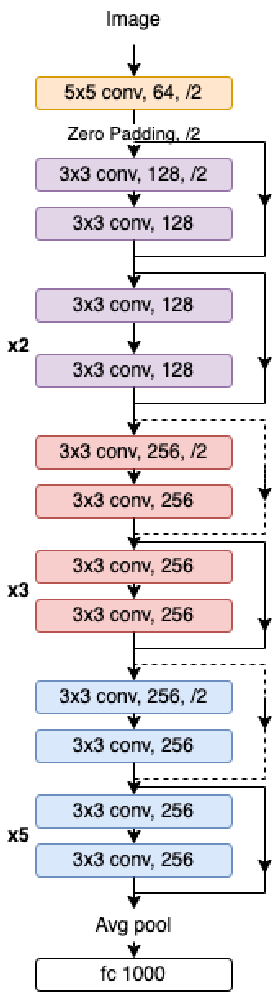

Fig. 7. ResNet Architecture

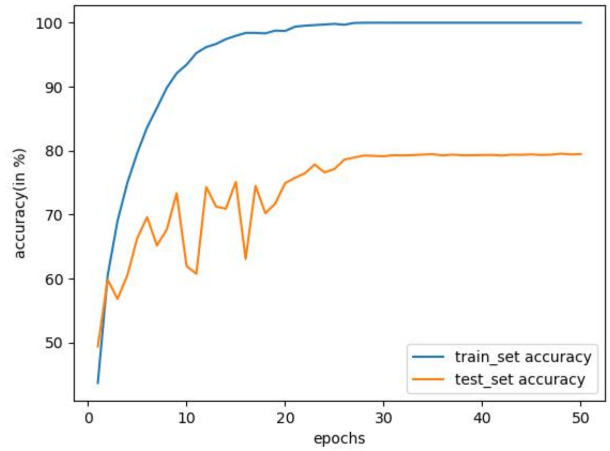

Fig. 8. Accuracy over epochs of the ResNet model

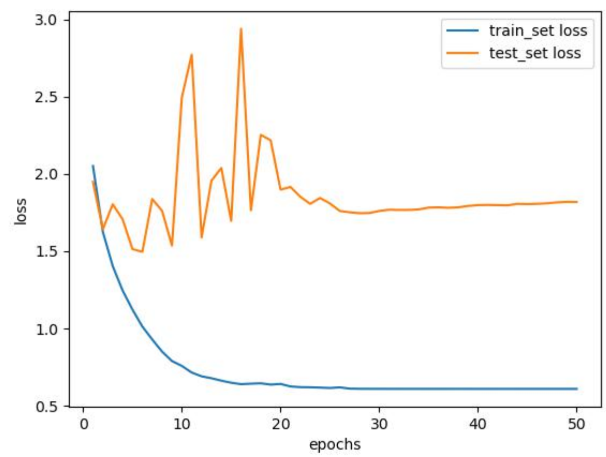

Fig. 9. Loss over epochs of the ResNet model

	0	1	2	3	4	5	6	7	8	9
0	5000	0	0	0	0	0	0	0	0	0
1	0	5000	0	0	0	0	0	0	0	0
2	0	0	5000	0	0	0	0	0	0	0
3	0	0	0	5000	0	0	0	0	0	0
4	0	0	0	0	5000	0	0	0	0	0
5	0	0	0	0	5000	0	0	0	0	0
6	0	0	0	0	0	5000	0	0	0	0
7	0	0	0	0	0	0	5000	0	0	0
8	0	0	0	0	0	0	0	0	5000	0
9	0	0	0	0	0	0	0	0	0	5000

TABLE V
CONFUSION MATRIX OF THE TRAINED RESNET ON TRAINING DATA

2) DenseNet Model: I followed the implementation of DenseNet121 from the paper but I reduced the growth rate and layer sizes since it was too big to train locally. My densenet model has 277866 parameters and 275562 of them were trainable parameters. For this model, I trained for 100 epchs and I was able to achieve 83.41% and 74.06% accuracies for the training and testing sets respectively at 97th epoch. I used Adam optimizer with fixed learning rate of 0.00001 for the model. Batch size used was 32.

The model begins with a convolution layer that with filter size that is twice the growth rate, (7,7) kernel size and (2,2) strides. The layer has relu activation. The output of the convolution layer is used as input for the max pooling layer with (2,2) pool size. This is followed by four dense blocks with a

	0	1	2	3	4	5	6	7	8	9
0	813	5	42	18	14	4	7	9	57	31
1	8	876	0	3	3	5	8	4	16	77
2	60	7	700	41	67	47	44	19	11	4
3	15	7	59	642	54	119	54	26	13	11
4	21	2	57	49	750	35	38	42	5	1
5	6	3	40	129	34	714	23	39	4	8
6	6	4	42	52	26	14	846	5	2	3
7	16	6	24	30	43	43	5	825	1	7
8	46	21	7	9	7	4	3	1	885	17
9	28	61	5	9	3	3	5	8	17	861

TABLE VI

CONFUSION MATRIX OF THE TRAINED RESNET ON TESTING DATA

transition layer in-between each block. Each dense block has 3 dense layers and each dense layer is comprised of convolution layer, relu activation, batch normalization, another covolution layer, relu activation and another batch normalization layer. All convolution layers has an L2 kernel regularization. Transition layer is composed of a batch normalization, a convolution layer, a dropout layer with 0.2 dropout rate, relu activation and average pooling layer with (2,2) pool size.

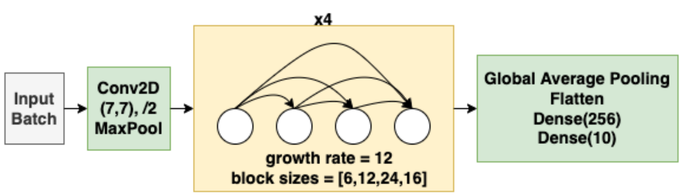

Fig. 10. DenseNet Architecture

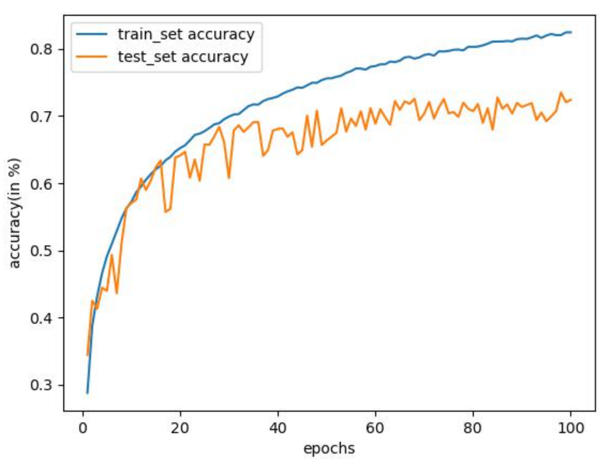

Fig. 11. Accuracy over epochs of the DenseNet Model

3) ResNeXt Model: The main difference between ResNet and ResNeXt is the concept of cardinality. I used cardinality value of 3 to build my model since that was the biggest I could use to train the model locally without GPU. Each resnext block has four layers and each layer has one pair of convolution and batch normalization layer, a convolution block as big as cardinality value, a batch normalization layer, and two pairs of convolution and batch normalization layers. I used batch size of 32, fixed learning rate of 0.00001, Adam optimizer and Cross Categorical Entropy loss. The model was trained for 50 epochs and I acheieved 81.80% and 50.36% accuracies for the training and testing sets respectively at 50th epoch. There are 2224244 parameters in the model and 2208048 parameters among them are trainable.

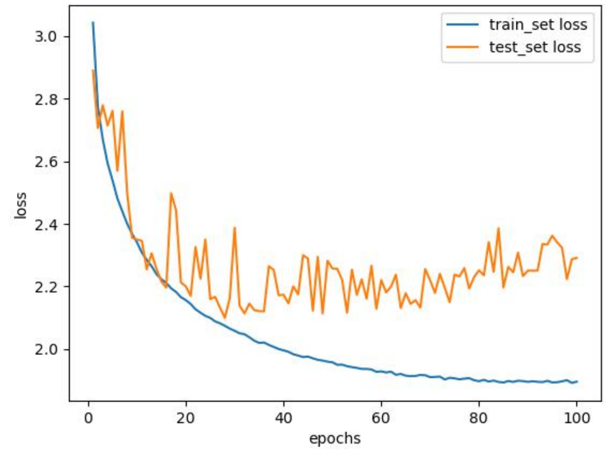

Fig. 12. Loss over epochs of the DenseNet Model

	0	1	2	3	4	5	6	7	8	9
0	4782	42	26	18	17	3	22	10	61	19
1	14	4907	1	3	0	1	10	2	14	48
2	222	16	4001	85	202	82	278	71	16	27
3	60	16	110	3521	190	397	519	107	23	57
4	81	7	114	65	4287	32	204	192	6	12
5	43	14	67	343	175	3823	261	213	15	46
6	15	7	30	36	41	14	4848	2	6	1
7	30	6	18	41	97	60	29	4682	8	29
8	167	69	3	5	13	5	12	8	4677	41
9	57	249	8	10	2	5	13	7	12	4637

TABLE VII

CONFUSION MATRIX OF THE TRAINED DENSENET ON TRAINING DATA

	0	1	2	3	4	5	6	7	8	9
0	822	37	24	13	15	3	16	10	40	20
1	13	879	3	3	3	1	15	6	19	58
2	87	7	561	47	95	51	91	35	11	15
3	31	17	45	462	66	145	135	56	11	32
4	39	4	51	29	664	18	99	80	12	4
5	20	5	33	143	44	594	72	71	4	14
6		9	6	27	20	19	12 890	9	4	4
7	24	8	18	24	53	51	17	782	5	18
8	78	38	12	9	14	3	12	3	811	20
9	47	123	5	8	2	6	11	9	14	775

TABLE VIII

CONFUSION MATRIX OF THE TRAINED DENSENET ON TESTING DATA

4) Analysis: Based on the results in Table X, ResNet has the best performance. Various factors such as batch size, epochs, learning rate and optimizer used can have influence on the model performances. Although my implementation of ResNet, DenseNet and ResNext did not have as many

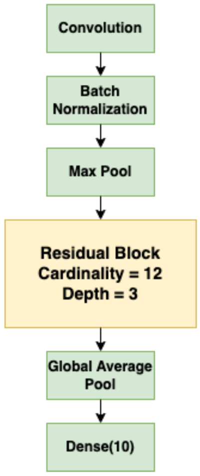

Fig. 13. ResNeXt Architecture

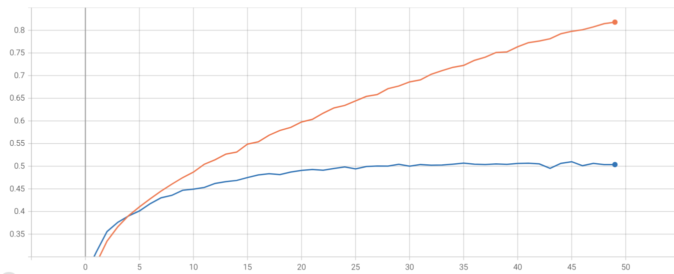

Fig. 14. Accuracy over epochs of the ResNeXt Model. The orange line represent the training accuracy and the blue line represents the testing accuracy. The X-axis is epochs and Y-axis is the accuracy values.

	0	1	2	3	4	5	6	7	8	9
0	4593	26	84	22	39	31	22	20	122	41
1	28	4729	16	15	6	3	14	7	44	138
2	60	15	4297	92	149	121	160	60	23	23
3	16	7	116	4172	87	344	149	57	26	26
4	31	6	134	84	4362	86	117	126	35	19
5	6	2	73	158	64	4543	81	55	8	10
6	15	5	51	67	57	56	4717	17	7	8
7	7	7	40	34	93	74	21	4693	9	22
8	42	33	20	8	30	12	12	6	4809	28
9	18	73	24	19	3	14	15	20	35	4779

TABLE IX
CONFUSION MATRIX OF THE TRAINED RESNEXT ON TRAINING DATA

	0	1	2	3	4	5	6	7	8	9
0	515	38	78	26	30	20	22	32	163	76
1	41	578	21	15	13	17	21	23	61	210
2	68	12	376	80	121	111	114	69	30	19
3	29	15	97	310	74	235	119	62	22	37
4	38	10	122	67	412	77	126	111	25	12
5	11	7	73	192	69	458	58	82	28	22
6	13	17	65	70	74	56	636	34	12	2
7	27	16	53	52	71	105	39	583	6	48
8	113	68	19	26	18	14	6	20	647	69
9	42	166	17	36	11	20	19	36	69	584

TABLE X
CONFUSION MATRIX OF THE TRAINED RESNEXT ON TESTING DATA

parameters as the models built in the original papers, we were

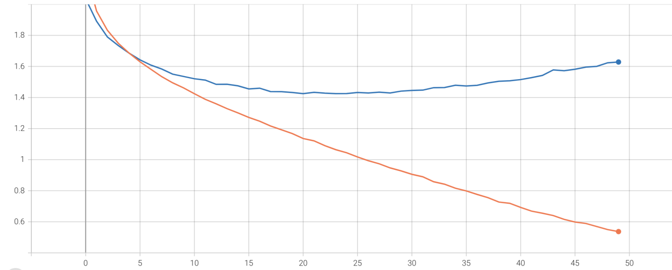

Fig. 15. Loss over epochs of the ResNeXt Model. The orange line represent the training loss and the blue line represents the testing loss. The X-axis is epochs and Y-axis is the loss values.

able to see that these models perform way better than basic model and optimized model.

Model	Parameters	Train Accuracy (%)	Test Accuracy (%)
Basic	11642	68.96	65.74
Optimized	122570	79.85	72.19
ResNet	277866	100	79.10
DenseNet	277866	83.41	74.06
ResNeXt	2224244	81.80	50.36

TABLE XI
PERFORMANCE SUMMARY OF ALL MODELS