Research Summary July 2020

LN Pandey
Rahul Vashisht

July 17, 2020

1 Attention Model on Mosaic image

1. **Experiment setup and details:**
   Let X_i denote the ith class in CIFAR10.
   F denote the set of Foreground Classes such that $F = \{X_0, X_1, X_2\}$
   B denote the set of Background Classes such that $B = \{X_3, X_4, X_5, X_6, X_7, X_8, X_9\}$
   i^{th} Mosaic Image (Data Point) be $M_i = (x_i, y_i)$
   where x_i be set of images s.t $x_i = \{I_k : 1 \leq k \leq 9\}$ and $y_i = Class(I_1)$
   Images I_1 to I_9 are randomly shuffled, sampled s.t $Class(I_1) \in F$ and $Class(I_j) \in B$, where $2 \leq j \leq 9$.

2. **Model**
   Since the dimension of one Image is 3x32x32. Therefore,

   $f(x)$ be the function representing "Focus" network such that

   $$f : R^{3x32x32} \rightarrow R$$

   $g(x)$ be the function representing "Classification" network such that

   $$g : R^{3x32x32} \rightarrow F$$

   Let $h(x)$ be the function representing "Model" such that:

   $$h : R^{3x32x32x9} \rightarrow R^3$$

   Let $\hat{y}_i$ be the class predicted by the attention network, then

   $$\hat{y}_i = h(x_i) = g\left(\sum_{i=1}^{9} I_i f(I_i)\right)$$

2 Experiments done with the above settings:

1. **Testing "classification" Network with Different "Focus" Network on Mini Inception network.**

   (a) First, we have created 20k Mosaic images (x_i, y_i) as defined in 1.
       Each Dataset D_i will contain 10k weighted aggregated images (avgImg) generated from Mosaic images. Labels of avgImg will be same as y.
       avgImg in Dataset D_i is defined as
       $$avgImg(D_i) = \frac{i}{9} I_F + \frac{9-i}{9x8} I_B$$
       I_F is foreground Image in Mosaic Image x_i and I_B is background Image in Mosaic Image x_i

   (b) Model (Classification network) is trained on Dataset D_i, and Tested on all Dataset D_j, where $j \neq i, j \in [1, 9]$.

   (c) We want to see, as the Focus network perform better, i.e from Dataset1 to Dataset9 (Focus network learns to focus better), how the Classification network behaves.

2. **Testing "classification" Network with Different "Focus" Network on CNN network.**

   (a) This experiment is exactly the same as the previous experiment on CNN Network.
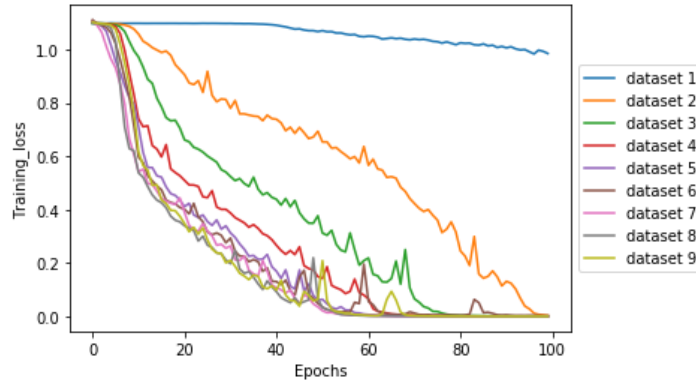

Figure 1: Train Loss for all Datasets when CNN Net is used

   (b) OBSERVATIONS:

   i. Even with 2/9 focus on foreground images, the classification network is still able to accurately classify 84% of the times.

   ii. Notice the faster convergence of classification network with more confident focus networks.

   iii. The classification network is powerful enough to pickup weak signals as well. This can be seen from the significant improvement from dataset2 to dataset3 (and from dataset3 to dataset4).

3. **Training Classification network by modelling output of Focus network as Dirichlet distribution with different alpha values**

   (a) Dataset Creation:
   First, we have created 20000 Mosaic images. Labels of every Mosaic image is the class of foreground image present in that image.
   Dataset 'i' will contain 10000 image. Where each image is weighted average of 9 images present in Mosaic image. Dataset 1 to Dataset 9 are made from same 10k Mosaic Images.
   Dataset 10 is made from 10000 to 20000 Mosaic Images which only contains the true Foreground Image present in every Mosaic Image. We have modeled the output of Focus network using the Dirichlet distribution with parameters alpha Since we know exactly where the foreground image is present in the Mosaic image
   therefore for Dataset 'i' and $\text{Dir}(\alpha)$,

$$
\begin{aligned}
w_{fg} &= a\_i * \alpha \\
w_{bg} &= \alpha \\
Avg\_Image &= w_{fg} * I_{fg} + \sum_{bg} w_{bg} * I_{bg}
\end{aligned}
$$

   where a_i is the parameter for Dataset 'i'.
   Where w_{fg} = weight_of_foreground_image,
   w_{bg} = weight_of_background_image,
   I_{fg} = Foreground_Image,
   I_{bg} = Background_Image

   (b) Paremeter a_i for different Datasets 'i'
   $a_i = (\frac{i}{9})(\frac{72}{9-i})$ for $i \in [1,8]$.
   For Dataset 9, $w_{fg} = \alpha$ and $w_{bg} = 0$.
   Dataset 10 will contain true foreground images of Mosaic Images from 10000 to 20000.

   (c) We have used $\alpha = [1e^{-2}, 0.5, 2, 1e^5]$
   $\alpha = 1e^{-2}$ denotes sparse samples and $\alpha = 1e^2$ denotes samples with equal weights.
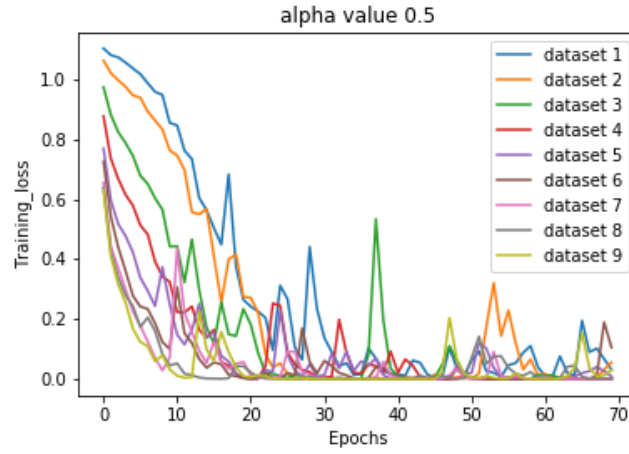
   (d) Observation:

Figure 2: Train Loss for all Datasets when $\alpha = 0.5$

    i. With alpha value less than one, occasionally background images will also get focused as input to classification network.

   ii. These results are similar to the experiments done with randomly fixing the Focus network.

4. **Stage wise learning with "classification" net first i.e training "classification" net for a fixed number of epochs then training "Focus" net for a fixed number of epoch.**

(a) Few Observations are:

Experiment No.	Total_Epochs	Train_network_for_Epoch	What_Learning_Rate	Where_Learning_Rate	Training Accuracy	Testing Accuracy
1	180	5	0.01	0.01	0.96	0.81
2	200	10	0.01	0.01	0.94	0.79
3	280	20	0.01	0.001	0.97	0.43
4	320	20	0.01	0.01	0.97	0.79
5	360	30	0.01	0.01	0.99	0.76
6	500	50	0.01	0.01	0.86	0.41
7	600	50	0.001	0.01	0.97	0.82
8	800	100	0.001	0.01	0.97	0.83

Table 1: Effect of Learning Rate on stage wise training

(b) The behaviour of accuracy and CE Loss for alternate training was a bit absurd when train alternatively for relatively larger number of epochs.

(c) It seems that a little improvement in Classification network rewards more as compared to a little improvement in Focus network.

(d) That is improvement in Classification will lead to more gains!

(e) "Classification" Network tries over fits the data while "focus" network was frozen, But when "focus" net starts training, it tries to restrict "classification" net in doing so and tries to learn focus correctly. hence every time after the "Classification" network finishes its training, There is a drop in accuracy and a sudden jump in CE Loss.
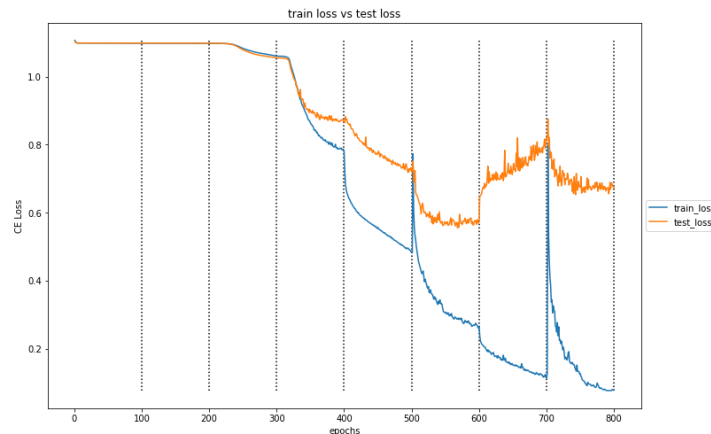
5. **Averaging at Convolution Layer 1**

   (a) Instead of taking a weighted average at pixel level, Averaging at Convolution layer 1.

   (b) Few observations are:

Experiment No.	Total Epochs	Learning Rate	Training Accuracy	Testing Accuracy
1	100	0.01	0.983	0.88
2	500	0.001	0.98	0.85

Table 2: Analysis of Training and Testing Accuracy

   (c) No major difference on testing data was observed as compared to when averaging of output images were taken of the "focus" network.

6. **Deep Dive into Attention** :
   We have 4 Datasets of Mosaic images (created from CIFAR 10 having classes $C = \{X_i : i \in [0,9]\}$) s.t in Dataset $D_i, F = \{i, i+1, i+2\}, B = C - F$. For all 4 datasets, we tried to analyse the FTPT (Focus True Prediction True), FFPT (Focus False Prediction True), FTPF(Focus True Prediction False), FFPF(Focus False Prediction False) for the following:

   (a) Data points (M_i s.t $(f, b_0) \in M_i$) having Foreground image f, s.t $\forall f \in F$ and Background image b_0, s.t $\forall b_0 \in B$.

   (b) Data points (M_i s.t $(f, b_0, b_1) \in M_i$) having Foreground image f, s.t $\forall f \in F$ and Background images (b_0, b_1) s.t $\forall b_0 \in B$ and $\forall b_1 \in B$.

   (c) Data points (M_i s.t $(f, b_0, b_1, b_2) \in M_i$) having Foreground image f, s.t $\forall f \in F$ and Background images (b_0, b_1) s.t $\forall b_0 \in B$, $\forall b_1 \in B$ and $\forall b_2 \in B$.

   (d) Data points (M_i s.t $f \in M_i$ and $(b_0) \notin M_i$) having Foreground image f, s.t $\forall f \in F$ and Background image b_0, s.t $\forall b_0 \in B$.

   (e) Data points (M_i s.t $f \in M_i$ and $(b_0, b_1) \notin M_i$) having Foreground image f, s.t $\forall f \in F$ and Background images (b_0, b_1) s.t $\forall b_0 \in B$ and $\forall b_1 \in B$.

   (f) Data points (M_i s.t $f \in M_i$ and $(b_0, b_1, b_2) \notin M_i$) having Foreground image f, s.t $\forall f \in F$ and Background images (b_0, b_1) s.t $\forall b_0 \in B$, $\forall b_1 \in B$ and $\forall b_2 \in B$.

3   Deep Network Training under random label noise:

1. Out of 50k data points 0,5,10,20,30,40,50,100% data points are corrupted:

   (a) On CIFAR10 dataset where the size of dataset is 50000, randomly assign labels to 0,5,10,20,30,40,50,100 percentage of the data and then train the network. Analyse the result for Training and Testing accuracy, CE Loss in all the scenarios.

   (b) Few observations are:

Corruption Percentage	Train accuracy on corrupted	Train accuracy on un-corrupted	Train accuracy on full	Test accuracy on corrupted	Test accuracy on un-corrupted	Test accuracy on full	Test accuracy on true data
0	-NIL-	1	1	-NIL-	-NIL-	-NIL-	0.83
5	1	1	1	0.0858	0.7830	0.7481	0.78
10	1	1	1	0.1028	0.7509	0.6841	0.75
20	1	1	1	0.1028	0.6659	0.5505	0.66
30	1	1	1	0.1019	0.5864	0.4405	0.58
40	1	1	1	0.1031	0.5058	0.3462	0.50
50	1	1	1	0.0970	0.4252	0.2625	0.42
100	0.99	-NIL-	0.99	0.1084	-NIL-	0.1084	0.10

Table 3: Analysis of Training and Testing Accuracy

2. Out of 50k data points, 25k data points were fixed as true data:

   (a) Keep the true training data constant as 25000, add randomise data (corrupted labels or corrupted training data) so that the final corruption percentages are given as 0,5,10,20,30,40,50. Analyse the result for Training and Testing accuracy, CE Loss in all the scenarios.

   (b) Few observations are:

Corruption Percentage	Train accuracy on corrupted	Train accuracy on un-corrupted	Train accuracy on full	Test accuracy on corrupted	Test accuracy on un-corrupted	Test accuracy on full	Test accuracy on true data
0	-NIL-	1	1	-NIL-	-NIL-	-NIL-	0.83
5	1	1	1	0.0978	0.7496	0.717	0.75
10	1	1	1	0.0844	0.7161	0.651	0.71
20	1	1	1	0.0961	0.6486	0.5353	0.64
30	1	1	1	0.1019	0.5755	0.4325	0.57
40	1	1	1	0.1047	0.5077	0.3479	0.50
50	1	1	1	0.1035	0.4254	0.2658	0.42
100	0.99	-NIL-	0.99	0.1084	-NIL-	0.1084	0.10

Table 4: Analysis of Training and Testing Accuracy

4 Systematic Corruption:

1. On CIFAR10 Dataset:
   Take CIFAR10 and create training data x_i, z_i as follows. Let x_i be the CIFAR10 images, and y_i be the CIFAR10 labels. The training labels z_i are simply given as follows:

   (a) $z_i = y_i$ if y_i is either 0 or 1 or 2, z_i is randomly 0 or 1 or 2 otherwise.

   (b) Train a network on this x_i, z_i data and predict labels on all test images. Check the accuracy on just the 0,1,2 labelled test images.

   (c) Training was done on CNN (Simple) and Mini-Inception Network (Complex models).

   (d) Few Observations are:

true training data	Corrupted Training data	Total Training data	Training accuracy	Test accuracy	Test accuracy 0-1-2
1000	35000	36000	1	0.11	0.38
2000	35000	37000	1	0.14	0.49
4000	35000	39000	1	0.16	0.56
6000	35000	41000	1	0.19	0.63
8000	35000	43000	1	0.20	0.67
10000	35000	45000	1	0.21	0.71
12000	35000	47000	1	0.22	0.73
15000	35000	50000	1	0.23	0.78

Table 5: Analysis of CIFAR Corrupted data on Mini-Inception Network

2. On MNIST Dataset:
   Take MNIST and create training data x_i, z_i as follows. Let x_i be the MNIST images, and y_i be the MNIST labels. The training labels z_i are simply given as follows:

   (a) $z_i = y_i$ if y_i is either 0 or 1, z_i is randomly 0 or 1 otherwise.

   (b) Train a network on this x_i, z_i data and predict labels on all test images. Check the accuracy on 0 and 1 labelled test images.

   (c) Training was done on CNN (Simple) and Mini-Inception Network (Complex models).

   (d) Few Observations are:

true training data	Corrupted Training data	Total Training data	Training accuracy	Test accuracy	Test accuracy 0-1
100	47335	47435	1	0.13	0.61
500	47335	47835	1	0.16	0.80
1000	47335	48335	1	0.17	0.83
2000	47335	49335	1	0.19	0.92
4000	47335	51335	1	0.20	0.95
6000	47335	53335	1	0.20	0.96
8000	47335	55335	1	0.20	0.96
12665	47335	60000	1	0.20	0.98

Table 6: Analysis of MNIST Corrupted data on CNN Network

5 Visualising Mosaic Data by Generating 2-Dim Separable Data

1. Generationtion of 10 Classes Using 2 Dimentional data :
   We sampled 5000 (500 for every class approx) 2-dim data points from multivariate Normal with 10 different Mean vectors and same Covariance Matrix. Covariance Matrix being the diagonal matrix with enteries [0.1, 0.1].

2. Generation of Mosaic Data

   (a) Available Classes = Class 0, Class 1, Class 2, Class 3, Class 4, Class 5, Class 6, Class 7, Class 8, Class 9.

   (b) foreground_classes = Class 0, Class 1, Class 2

   (c) background_classes = Class3, Class 4, Class 5, Class 6, Class 7, Class 8, Class 9

   (d) Every class will have a 2-dim Data Point. 1 data point was chosen at random from any foreground class, and rest 8 data points are chosen from background classes.

   (e) Now we have 9 data points which can be arranged randomly in 3 x 3 image matrix. In particular Dim of Matrix will be 3 x 6.

3. Following is the table of parameters and accuracy for different data sets used.
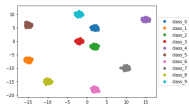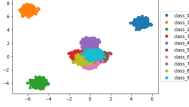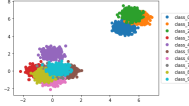
type of dataset used	epochs	Focus LR	Classification LR	Train Acc	Test Accuracy
	80	0.01	0.01	100	100
	10	0.01	0.01	100	100
	120	0.01	0.01	88	89

Table 7: Train accuracy and test accuracy

4. Our main goal is to find when the attention networks are interpretable from this experiment.

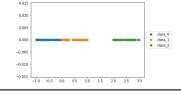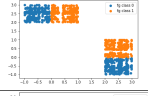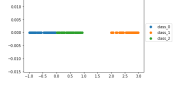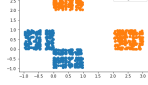5. Again another type of data distribution is used as below:

type of data	plot dataset used	Mosaic Data
linearly not seperable		
linearly seperable		

Table 8: Type of Data used for the experiment

6. Observation:

   (a) When mosaic data is not linearly separable The models needs attention to learn a good classifier. For higher dimension once it learns some good attention weights, the attention weight learning stops.

   (b) When mosaic data is linearly separable The model does not need attention to learn a good classifier. For higher dimension model need not require to learn the attention weights, but is sometimes learning good attention weights.

6 Correlation of Loss with Foreground and Background Classes

1. Generation of 10 Classes Using 2 Dim data :

   (a) We sampled 1000 (100 for every class approx) 2-dim data points from multivariate Normal with 10 different Mean vectors and same Covariance Matrix.

   (b) Covariance Matrix being the diagonal matrix with entries [0.01, 0.01].
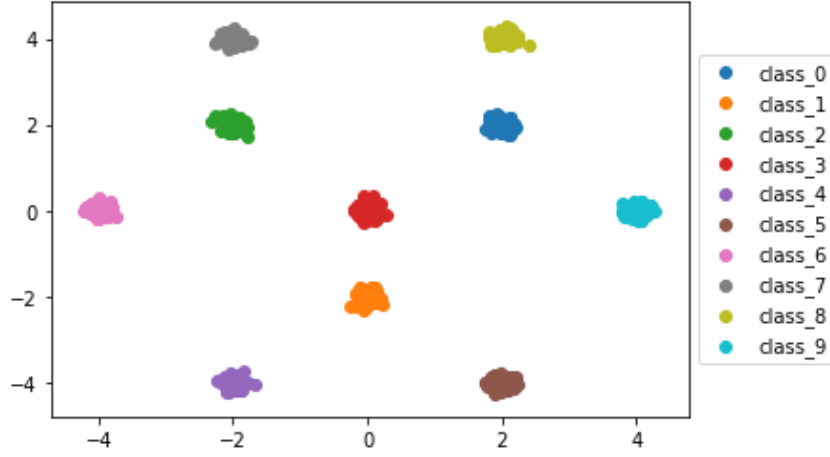
Figure 3: Generation of 10 Classes Using 2 Dim data

2. Generation of Mosaic Data

   (a) Available Classes = Class 0, Class 1, Class 2, Class 3, Class 4, Class 5, Class 6, Class 7, Class 8, Class 9.

   (b) foreground_classes = Class 0, Class 1, Class 2

   (c) background_classes = Class3, Class 4, Class 5, Class 6, Class 7, Class 8, Class 9

   (d) Every class will have a 2-dim Data Point. 1 data point was chosen at random from any foreground class, and rest 8 data points are chosen from background classes.

   (e) Now we have 9 data points which can be arranged randomly in 3 x 3 image matrix. In particular Dim of Matrix will be 3 x 6.

3. Model used :
   Model is developed as combination of 2 modules. "FOCUS" Module learns to "where" the foreground image is present out of 9 images in Mosaic image. "CLASSIFICATION" Module learns "what" is the class of this foreground image out of those 3 foreground classes.

4. Input to Model :
   Mosaic image is input to Module 1 i.e "FOCUS Network", and tries to focus on foreground image present in Mosaic Image. In Particular, Each image (2 x 1) is input to "FOCUS Network" and hence a 18 x 1 tensor (9 images) is input to "CLASSIFICATION Network". "FOCUS Network" tries to Focus on Foreground image and returns weighted average of all 9 images. This image is now input to "CLASSIFICATION Network" which finally predicts the Class label of foreground Image.

5. Gradient Definition :

$$\nabla\alpha_{fg} \quad = \quad \frac{1}{size(Datapoints)}\left[\sum_{datapoints} \nabla Loss_{\alpha}(Fg\_one\_hot)\right]$$

$$\nabla\alpha_{bg} \quad = \quad \frac{1}{size(Datapoints)}\left[\sum_{datapoints} \nabla Loss_{\alpha}((1 - Fg\_one\_hot)/8)\right]$$
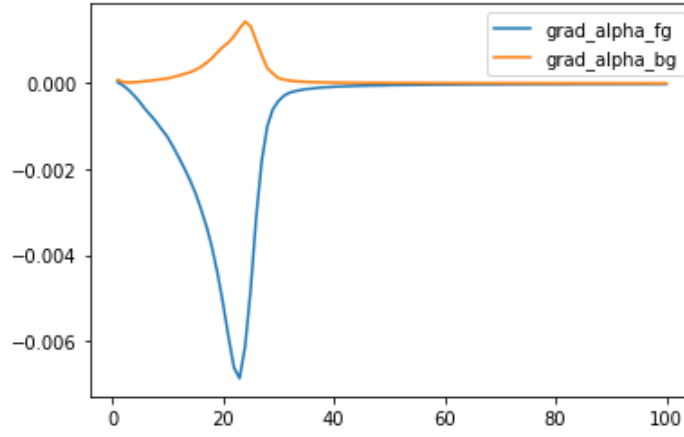
Figure 4: Gradient(Loss) w.r.t alphas for Foreground and Background Classes

7 Confounded Noise

1. Training Data : Take 2 Dim data and create training data x_i, y_i as shown in table below.
   Training data after Corruption : $z_i = y_i$ if y_i is either 0 or 1. z_i is randomly 0 or 1 otherwise. Train a network on this x_i, z_i data and predict labels on all test images. Check the accuracy on just the zero and one labelled test images.

Data Set	True training data	Corrupted training data
linearly separable at corner		
linearly separable at center		
moon data linearly separable at corner		

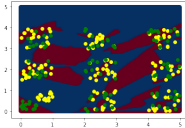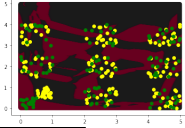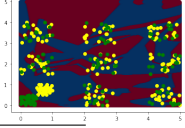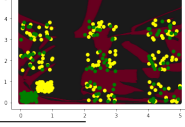**Table 9:** True Training data and after corruption

2. Observations :

8

true data	Corrupted Data	Training Accuracy	Test Accuracy	Data Accuracy with no noise	Figure
30	240	97	57	100	
60	240	96	61	100	
90	240	96	66	100	
120	240	98	67	100	

Table 10: training on noisy Linearly separable Data at corner with increasing nos. true data points

Similarly, the same experiments were performed on the moon data linearly separable at corner and data linearly separable at center and following interesting plots after training were observed.
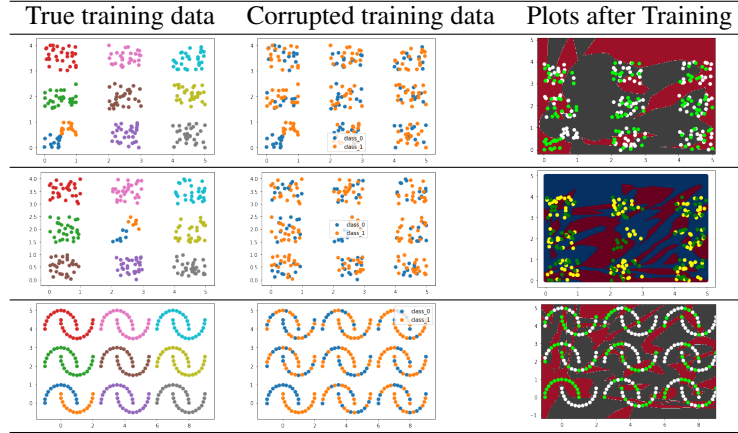
True training data	Corrupted training data	Plots after Training

Table 11: training on Different noisy dataset

Neural Networks can be generalised even in the presence of confounded noise (adding noise to the label). The experiment is similar to the experiment done in the paper "Model Neural Networks Generalize on Small Data sets". Only with small nos. of true data points, model is able to learn the classifier.

8   Averaging Noise

1. 3 dim Data points is generated s.t class 0,1,2 are in one plane (P_fg), and Class 3,4,5,6,7,8,9 are in other plane (P_bg). P_bg is fixed as x-y plane. Class 0,1,2 are Foreground classes, Class 3,4,5,6,7,8,9 are background classes.

2. Generation of Datasets

    (a) First, we have created 1000 Mosaic images (x_i, y_i) as defined in 1.
    Each Dataset D_i will contain 1000 weighted aggregated images (avgImg) generated from Mosaic images. Labels of avgImg will be same as y.
    avgImg in Dataset D_i is defined as
    $$avgImg(D_i) = \frac{i}{9}I_F + \frac{9-i}{9x8}I_B$$
    I_F is foreground Image in Mosaic Image x_i and I_B is background Image in Mosaic Image x_i

9

(b) Model (Classification network) is trained on Dataset D_i, and Tested on all Dataset D_j, where $j \neq i, j \in [1,9]$.

(c) We want to see, as the Focus network perform better, i.e from Dataset1 to Dataset9 (Focus network learns to focus better), how the Classification network behaves.

3. following are the observation with different angle between the planes. A small plane denote the plane of FG classes.
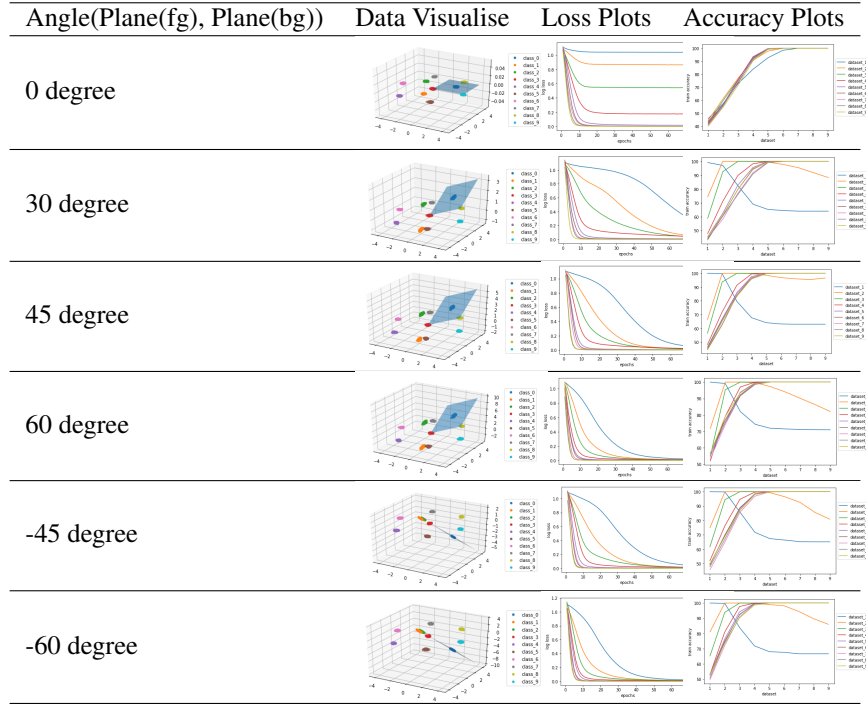
Angle(Plane(fg), Plane(bg))	Data Visualise	Loss Plots	Accuracy Plots
0 degree			
30 degree			
45 degree			
60 degree			
-45 degree			
-60 degree			

Table 12: training on Different noisy dataset