DL_Project_Contents

Lingfeng Zhang

December 2019

1 Experiment Setup

In this experiment, relationships between the existence of university perturbation and the dataset complexity, as well as the training model complexity should be found. There are three main steps to process experiments:

- Generating images with different complexity levels
- Training models with different number of layers
- Generating the universal perturbation based on a certain complexity level dataset and a certain number of layers training model.

1.1 Generating images

Some assumptions setting:
images size and channel: 28*28*1
number of classes: 4
images number for each class: 1000

Firstly, 4 class baseline images should be created. See Figure 1.To generate these images, setting 1 as the pixel value on separated locations and uniform distribution random number ranged [0,1) on other unfilled locations. Making sure that the training model can recognise different class images. In this case, setting 1 on upper left corner 7*7 pixels and random number on other pixels represents class 0 images. And by the same logic, images of class 1, 2 and 3 are created.

While getting more complex dataset, more convolutional layers in the generating images model(called "images generator") should be added. The relation between them see formula (1). The weights and bias in convolutional layers do not need be learned in the images generator. These parameters are initialized and fixed by random values, uniform distribution random value for weights and normal distribution for bias.

$$number\ of\ layers\ in\ generator\ \propto dataset\ complexity\ level \tag{1}$$

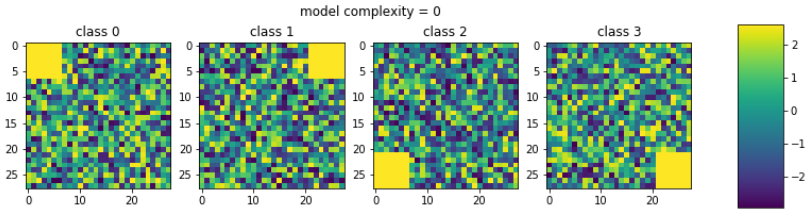

Figure 1: 4 class baseline images

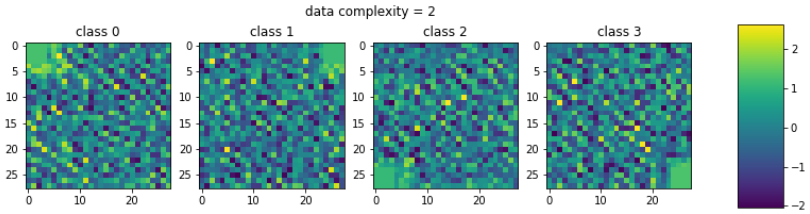

Figure 2: 4 class images after 2 convolutional computation

After 2 layers convolutional computation, the baseline images are transferred into Figure 2. These images look like more complex than baseline images intuitively.

It should be mentioned that all pixel values of one image will be close to a certain number(in most cases, close to 0) after original images passing the images generator with many convolutional layers. This is probably because feature boundaries of different class images fades with increasing times of convolutional computations. So, these images pixel values should be scaled. Feature boundary in this experiment is represented by the location of pixels with value 1.

There are several comments and tricks about the images generator model

- To make sure that adding more layers in the images generator can get more complex dataset, weights in less layers of the images generator should be stored and used again on lower layers of the more layers images generator. Otherwise, if these weights are totally different random values, it will occur that the more layers generator may get lower complex dataset than the less layers generator. This is because all weights of the more layers generator are probably overall smaller than these weights of the less layers generator.

- This method to generate different complexity level dataset is reasonable because convolutional computation can gradually blur the decision boundary from baseline images dataset with increasing the convolutional layers.

- The number of convolutional layers in the images generator is in range from 0 to 9. When 10 convolutional layers are used in the generator,

Height	Width	Depth	filter Height	filter Width
36	36	1	3	3
34	34	32	3	3
32	32	32	3	3
30	30	32	3	3
28	28	1	None	None

Table 1: The structure of images generator with 4 convolutional layers

training model could not learn the decision boundary of these classes well, meaning testing accuracy is not good enough. This is probably because too many convolutional computations mix up the location information which can be recognised by the training model. In another word, too much layers in images generator will generate "garbage" images which can not be classified by training models well. After 9 layers convolutional computation, the baseline images are transferred into Figure 3. It is difficult to recognise which classes these images belong to by human eyes.

- To compare the size of universal perturbation generated from different complexity level dataset later, these images pixel values should be expanded or shrank to the same scale. So, these images pixels should be normalized after passing these convolutional layers.

- To avoid too much loss of baseline images' information, the activation function in each convolutional layers in images generator is LeakyRelu because other activation functions like ReLu, sigmoid, tanh or others, may induce values into saturation field, which cause the information loss.

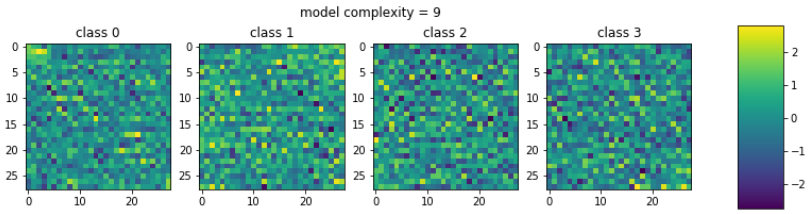

Figure 3: 4 class images after 9 convolutional computation

The structure of images generator with 4 convolutional layers(dataset complexity level 4) is shown in Table 1 and Figure 4.

To find the relation between the dataset complexity and the universal perturbation, the number of training model layers should be fixed on 4.

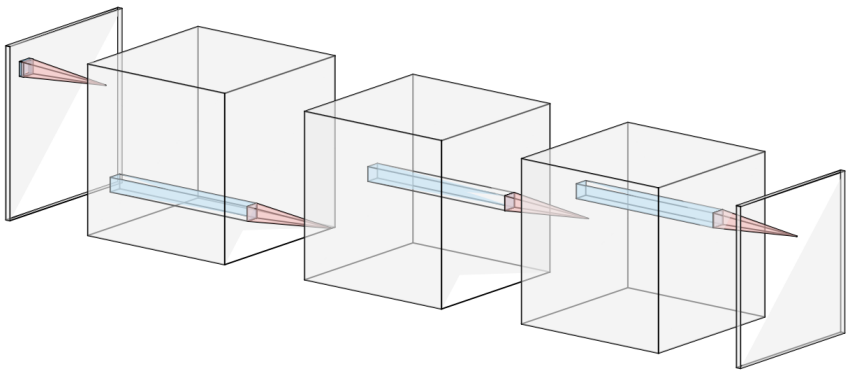

Figure 4: The structure of images generator with 4 convolutional layers

1.2 Training models

To get different complex training models, adding different number of layers can achieve this goal. Seven kinds of layers number are record: 3, 5, 8, 11, 14, 17 and 20. 3 layers training model includes 1 convolutional layer and 2 fully connected layers. Similarly, for 3, 5, 8 and 11 layers training models, they include 2 fully connected layers and layers-2 convolutional layers. With increasing too many convotional layers, the traning model can not perform well, but adding more fully connected layers based on sufficient number of convolutional layers can improve the training model performance again. So, for 14, 17 and 20 layers training models, they include 11 convolutional layers and layers-11 fully connected layers.

In addition, with increasing the number of layers in the training model, smaller learning rate should be applied because large learning rate could not help complex training models converge.

After fine tuning, all training models can reach up 99% testing accuracy within small epochs. The activation function for all layers in all training models are ReLu.

The structure of training model with 4 convolutional layers is shown in Table 2 and Figure 5.

To find the relation between the training model complexity and the universal perturbation, the dataset complexity level(number of layers in images generator) should be fixed on 4.

1.3 Generating the universal perturbation

Some hyper-parameters setting

Layer	Height	Width	Depth	filter Height	filter Width
Convolution	28	28	1	3	3
Convolution	26	26	32	3	3
Dense	24	24	128	None	None
Dense	24	24	32	None	None
Output	4		1	None	None

Table 2: The structure of training model with 4 convolutional layers

Data complexity	Model layers	ξ
0		4.2
1		2.0
2		1.8
3		1.8
4		1.8
5	4	1.6
6		1.6
7		1.6
8		1.1
9		0.9

Table 3: Relation between dataset complexity and universal perturbation

- Images used to train the universal perturbation: 400 and 100 for each class
- Baseline Method to generate perturbation: Deep Fool
- Fooling rate $> 70\%$
- Metric to calculate the magnitude of universal perturbation: Infinity Norm $||\xi||_\infty$
- Maximum number of iterations to train the universal perturbation: 10

2 Result Analysis

To visualize the Table 3, see Figure 6.

As shown in table, for example, when the data complexity is 0 and the number of training models layers is 4, if adding the universal perturbation is smaller than 4.2, then the fooling rate will less than 70%. The value of ξ is the boundary to determine whether fooling rate is greater or smaller than 70%. So this value can show how easy the universal perturbation exists.

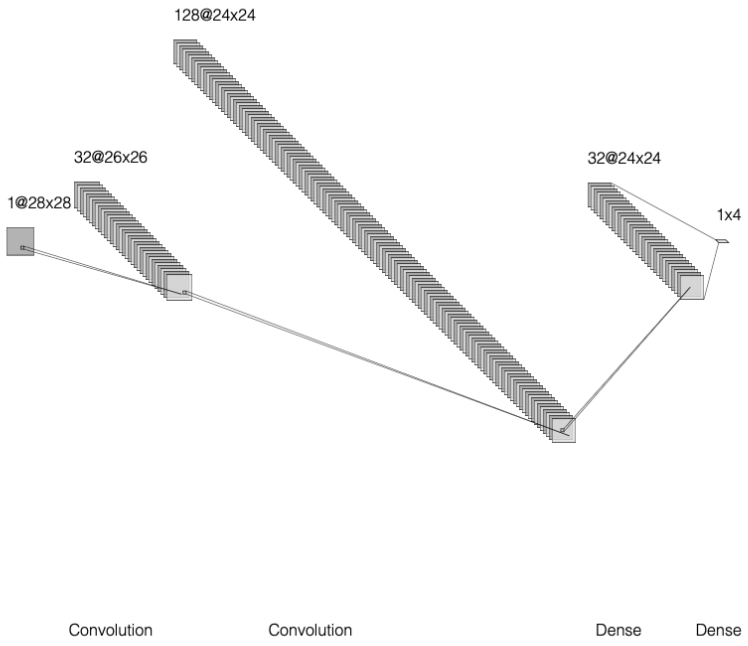

Figure 5: The structure of training model with 4 convolutional layers

See Figure 7, when dataset complexity level is 1, adding universal perturbation with infinity norm 2.0 can fool training model to recognise one image of class 2 as class 3.

See Figure 8, when dataset complexity level is 9, adding universal perturbation with infinity norm 0.9 can fool training model to recognise one image of class 1 as class 3.

To compare these two universal perturbations, the universal perturbation with larger infinity norm changed the original images information more, like brute force to change one class image to another class. This kind of universal perturbation is relatively meaningless than slight perturbation in real world.

When the dataset become more complex, smaller universal perturbation can be added on images to fool the training model well.

To visualize the Table 4, see Figure 9.

Overall, when the training model become more complex, smaller universal perturbation can be added on images to fool the training model well. In this experiment, the universal perturbation generated from 20 layers training model is slightly larger than those generated from 14 and 17 layers training models.

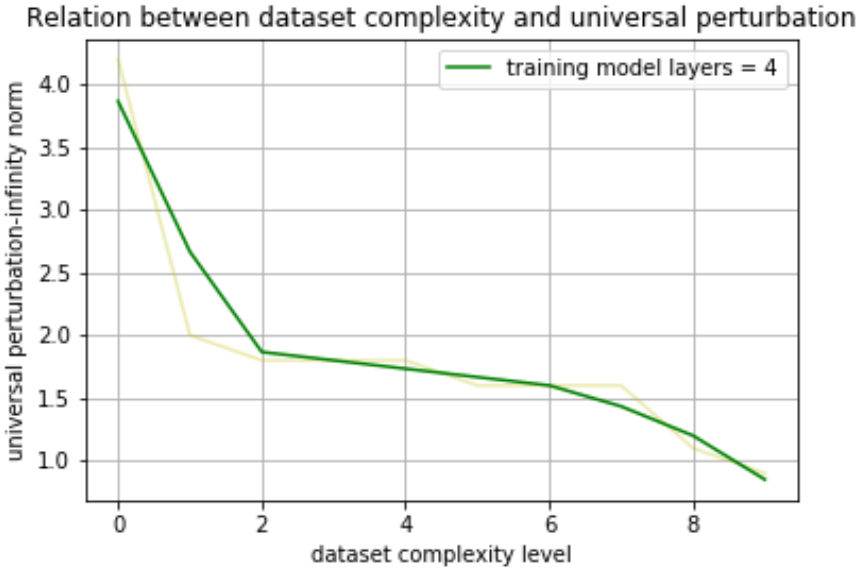

Figure 6: Relation between dataset complexity and universal perturbation

This is probably because some bias of this experiment occur.

3 Discussion and Future Work(addition)

For the bias in the experiment, more experiments should be done to analyse why this noise happen. For example, adding more than 20 layers in training model to analyze the infinity norm of the universal perturbation.

In this experiment, adding more convolutional layers in images generator can control the complexity level of the datasets. But in real world, some methods should be found or applied to evaluate semantic datasets complexity, such as

Model layers	Data Complexity	ξ
3		2.7
5		2.4
8		1.5
11	4	1.5
14		1.3
17		1.2
20		1.5

Table 4: Relation between training model complexity and universal perturbation

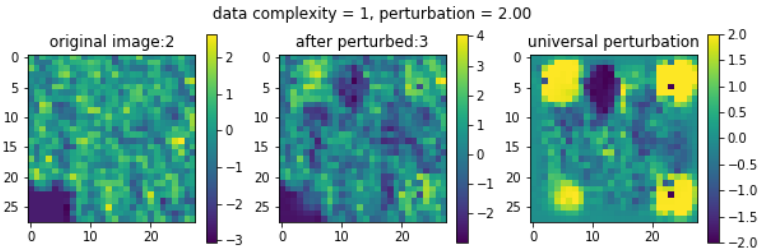

Figure 7: Example of universal perturbation with infinity norm 2.0

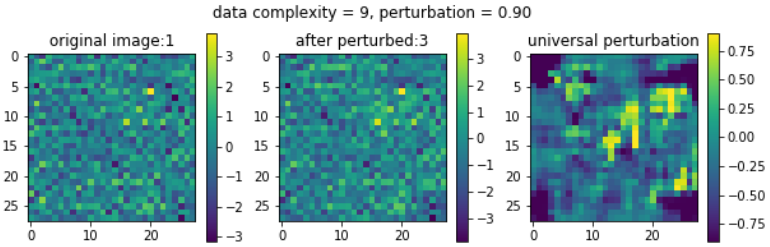

Figure 8: Example of universal perturbation with infinity norm 0.9

spectral metric [1] or other complexity measures.

Calculate the difference of different class images in various complexity level datasets by using Kullback–Leibler divergence of other metrics. Find whether the difference of different class images in complex datasets is smaller than that in simple datasets.

Test the universality of universal perturbation and find whether there is a "global" universal perturbation. In this experiment, universal perturbation is "locally" universal, meaning it is a pattern can fool the training model trained on a certain dataset and may not fool the training model trained on other datasets in a high fooling rate.

4 Conclusion(addition)

To summarise, when the dataset and training model become more and more complex, the decision boundary gradually become unclear and difficult to be learnt by classifiers. So, there may exist an universal perturbation or a pattern to disturb original dataset information, inducing one class images can be added a small perturbation to "skip" into other decision areas.

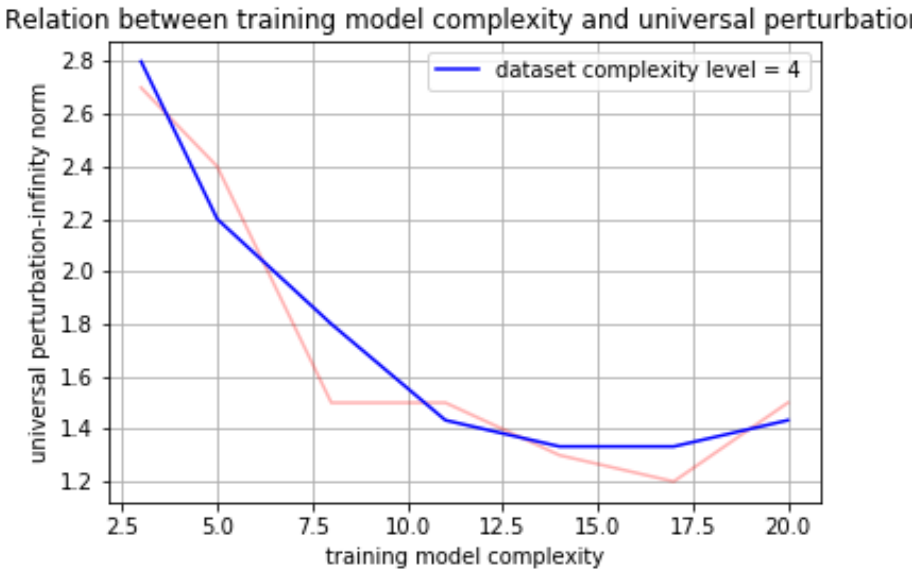

Figure 9: Relation between training model complexity and universal perturbation

References

[1] Frederic Branchaud-Charron, Andrew Achkar, and Pierre-Marc Jodoin. Spectral metric for dataset complexity assessment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.