Project 2: Deep Learning and Feature Visualization

Xiangyu Lin

May 2019

1 Introduction

This project aims to apply deep learning models to multiple datasets and visualize the features. There are three tasks in this project: multi-category classification, image segmentation and image reconstruction. Also we need to visualize the features when doing these tasks.

Deep learning is a widely used method in this day. It achieves significant results in many fields, such as computer vision. The teacher has provided us with some widely-used datasets in image classification, image segmentation and image reconstruction. We will try out some well-known deep learning networks on these dataset and use feature visualization methods such as PCA, t-SNE and Grad-CAM to figure out how such model works.

2 Multi-category Classification(My Work)

Multi-category is a basic task in computer vision. Given an image as input, and a number of categories as output, we need to train a model which is able to classify the images correctly.

There are some well-known models which perform quite well on ImageNet dataset: AlexNet, VGG and ResNet. Due to limited computing resources, we are only able to use simple versions of these network and simple dataset such as MNIST.

The simple versions mainly simplified the number of filters on each layer and the number of units in fully connected layers. There are the results of simplified AlexNet, VGG and ResNet on MNIST dataset:

Model	Test Loss	Test Acc
Alexnet	0.0921	0.9797
Resnet	0.0798	0.9812
VGG	0.0144	0.9955

Figure 1: Results of classification models

Simplified VGG model outperforms ResNet in our experiment. The reason may be that the ResNet model is not good enough, but it is more possible that the image is relatively small(28×28 in MNIST dataset) and deep network is likely to have too many abundant parameters, which makes ResNet has no advantage over shallow well-structured simplified VGG network.

Here is our model for classification with input/output size for each layer shown on the right:

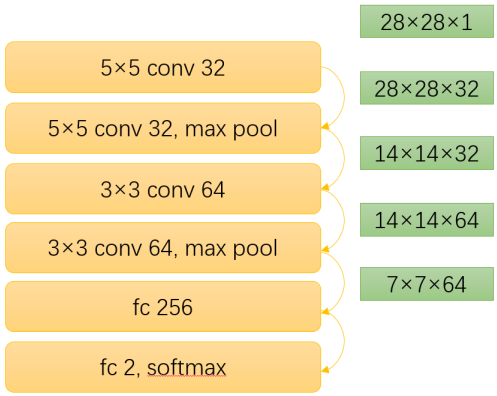

Figure 2: Model of classification

We refer to VGG model as the inspiration for designing the model, it is proved successful on MNIST dataset and can reach 0.9955 accuracy with fewer parameters comparing with original large network

In addition, we have tuned the number of filters of each convolutional layer in our model, here is the result:

filters	filters	filters	filters	loss	accuracy
16	16	16	16	0.0389	0.9872
16	16	32	32	0.0272	0.9914
32	32	32	32	0.0254	0.9918
32	32	64	64	0.0173	0.9943
64	64	64	64	0.0174	0.9941

Figure 3: Tuning of classification

We can see that as number of filters increases, the performance improves until overfitting when there are too many parameters in the model.

3 Image Segmentation(My Work)

Image segmentation in fact is pixel-level classification. The task is to classify each pixel as a category of object or background, and visualize the result.

We use augmented Pascal VOC 2012 dataset,which merges original Pascal VOC 2012 dataset with SBD dataset(http://home.bharathh.info/pubs/codes/SBD/download.html). We have 8829 training samples and 3748 testing samples in total.

We modify the structure of previous classification model to obtain our segmentation model. The input image is 96 × 96 with 3 channels so we add three more convolutional layers in the middle. The most significant change is the output layers. For classification model, the output layers are two fully-connected layers while in segmentation model, the output layers are replaced by combination of convolutional layers and transpose convolutional layers. We refer to FCN8 to design our output layers, and our model is shown as follows:

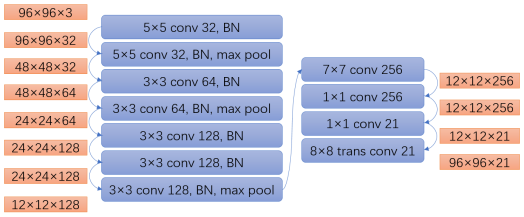

Figure 4: Model of image segmentation

Suffering from lack of computing resouces, we can only use images with size 96 × 96, so our accuracy is only about 0.78. But there are still many well-segmented images. In general, our model is better at segmenting human than animals and other objects. The main reason is that people in the dataset usually face the camera and are generally in the center of the picture, while other targets such as animals, planes and cars, may have different postures.

Here are some of the results, black pixels are classified as background, white pixels are classified as target object while red pixels are wrongly classified as other objects:

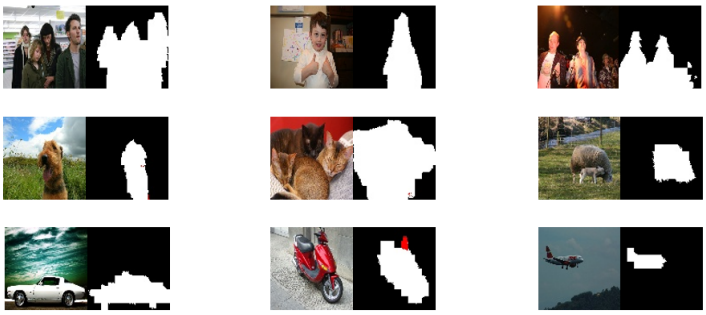

Figure 5: Results of image segmentation

4 Image Reconstruction

In neural net language, a VAE(Variational auto-encoder) consists of an encoder, a decoder, and a loss function. In probability model terms, the variational autoencoder refers to approximate inference in a latent Gaussian model where the approximate posterior and model likelihood are parametrized by neural nets.

The following figure showes the network structure of VAE,for image reconstruction, the input is the original image, we will first construct the encoder network, get the mean vector and standard deviation vector and then derive the latent vector. For decoder network, the input is latent vector and output is the reconstructed image.

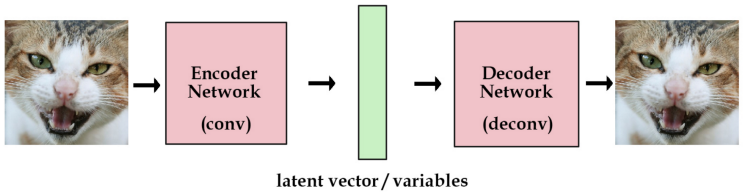

Figure 6: Structure of VAE

We have developed three models for image reconstruction.

4.1 Simple Encoder-Decoder Model

This model just use one fully connection layer as encoder and another fully connection layer as decoder. We tested this model on mnist dataset. The result

is shown in the following figure:

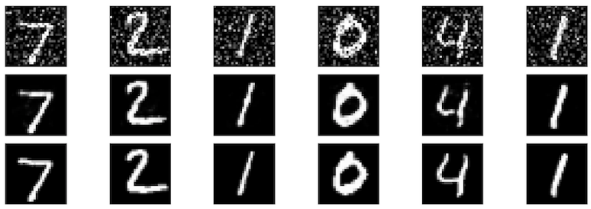

Figure 7: Result of simple encoder-deconder model on MNIST dataset

The first line is the image with noise which is used as the input of model, the second line is the reconstructed images and the third line is original images.

The simple model performs quite well on MNIST dataset because the dataset is very simple, but on more complicated dataset, it fails us. So we move on to more sophisticated models.

4.2 VAE

The difference of this model from the previous one is that it uses convolutional layers and batch normalization layers in encoder and decoder. The detailed structure of VAE can be seen on my teammate's report, here I just introduce its result.

The following figure is the result of VAE on MNIST dataset, we can see that the model still performs very well on simple dataset.

Figure 8: Result of VAE on MNIST dataset

What's more, VAE outperforms simple model on more complicated dataset like CIFAR-10. Following is the result of simple model and VAE, we can see the difference.

The first line is the image with noise which is used as the input of model, the second line is the reconstructed images and the third line is original images:

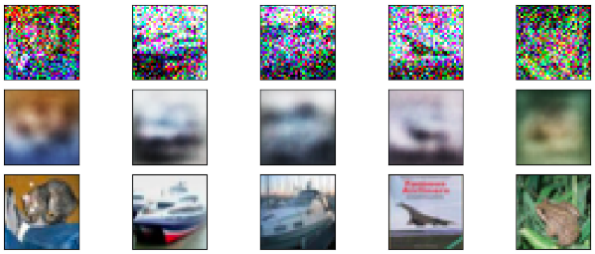

Figure 9: Result of simple model on CIFAR-10 dataset

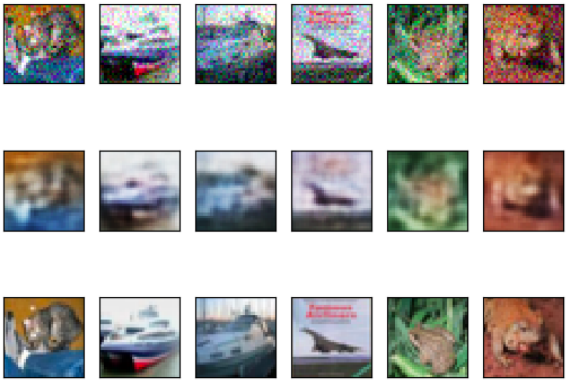

Figure 10: Result of VAE on CIFAR-10 dataset

4.3 Modified-VAE

This VAE model has more convolution layers and uses LeakRelu as encoder activation function and uses a Batch Normalization layer to let of each layer's output follow Gaussian distribution.We tested this model on Stanford dogs datasets which is even more complicated and get good results:

Figure 11: Result of modified-VAE.

5 Feature Analysis

5.1 PCA

Principal component analysis (PCA) is a widely-used dimension reduction algorithm. It can project n-dimensional vector to m-dimension.

Here we choose the first fully connected layer in the vgg net shown in the classification section above as the intermediate layer. Then we reduce the dimension of the vectors and then cluster them. First we reduce the vectors to two dimensions, since we use MNIST dataset here, we cluster the vectors into 10 clusters:

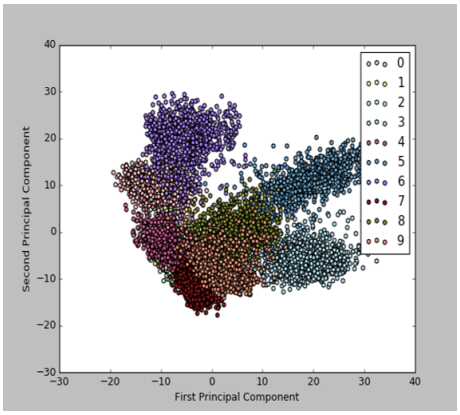

Figure 12: Clustering Result of Keeping Two Principal Components

We can see that the 10 categories cannot be divided very clearly if we only keep two principal components. Then we reduce the n-dimensional vectors to three dimensions:

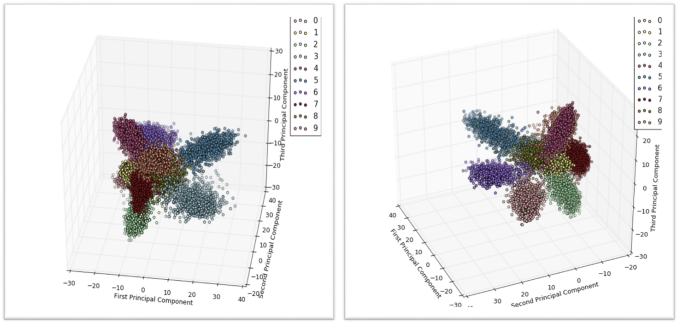

Figure 13: Clustering Result of Keeping Three Principal Components

We can see from the figure that the different categories are distinct from the figure now, which means that the 3-dimensional features keeps enough information to divide the data clearly.

5.2 t-SNE

t-SNE provides a way to reduce dimension, making it possible for us to show the clustering result of high dimensional in the form of two dimensions:

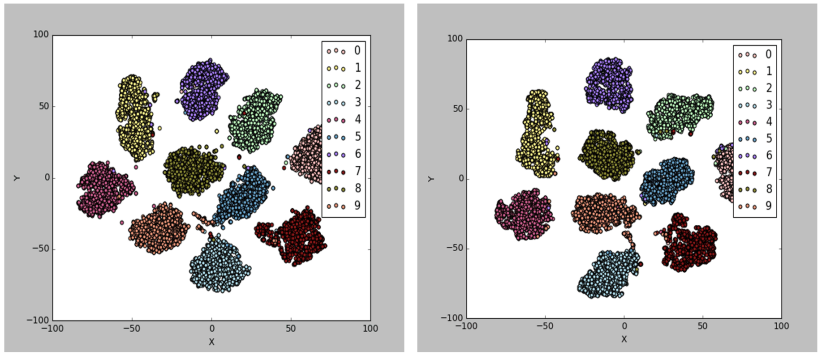

Figure 14: Using t-SNE to visualize the intermediate-layer feature f=g(x) & m-dimentional features of PCA

The right figure is the result of using t-SNE to visualize the m-dimentional features of PCA. One of the advantages of t-SNE is that it can enlarge the distances between the clusters, so we can see the clustering results much more

clearly. Comparing to the clustering result of PCA, we can find that the clustering result of t-SNE has clearer boundaries, which makes the different categories more distinct in the figure.

5.3 Grad CAM

Grad CAM a technique for producing 'visual explanations' for decisions from a large class of Convolutional Neural Network (CNN)-based models, making them more transparent. Grad CAM uses the gradients of any target concept, flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept.

Our group member implemented Grad CAM using three different neural networks: VGG16, VGG19 and our simpler network used in the classification section.

The following figures are results of Grad CAM on VGG16 and VGG19.

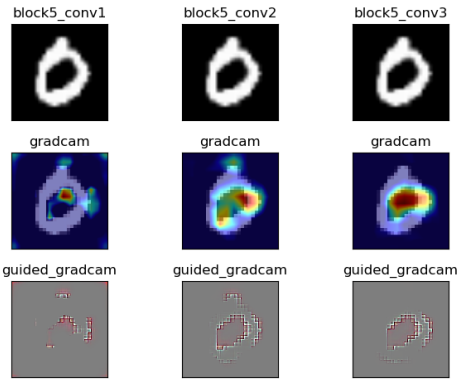

Figure 15: Grad CAM and Guided Grad CAM of VGG16

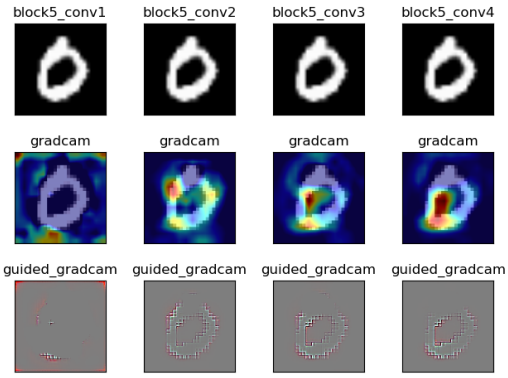

Figure 16: Grad CAM and Guided Grad CAM of VGG19

The third model is the neural network designed by myself. We output the grad cam and guided grad cam of these layers to compare the attention place between VGG and our model. As the Fig 17 shows, our model's attention is mainly on the track of the number, while VGG models' attention is more concentrated on a circle.

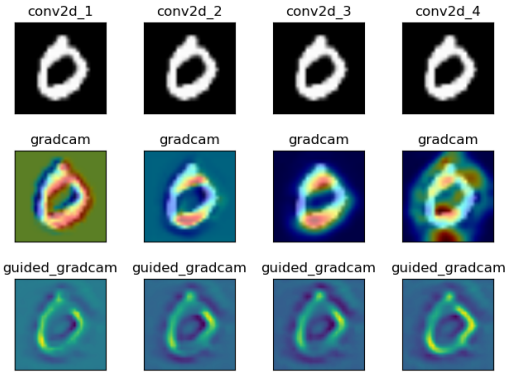

Figure 17: Grad CAM and Guided Grad CAM of our model

6 Challenges and Thinking

I work on most part of classification and segmentation in this project, the major challenge I encountered is the lack of computing resources. Due to there's no available server for me to use, I can only use simple dataset and simple network. So for classification task, I only test on MNIST dataset, which is the simplest of all.

Also, I need to design a network which is both simple and good. So I simplify

the three mainstream networks in this project: AlexNet, VGG and ResNet, and use the simplified networks to test. It proved that the structure of VGG network is more suitable for the dataset(As shown in the classification section above).

We notice that the structure of VGG is quite in order as it gradually increase the number of filters as the depth increases, and use multiple convolutional layers before pooling layer. Though I don't really get the idea of such design, I refer to it to design our model, and it proved useful.

Based on this model, I finished the classification and segmentation tasks with some help from another teammate. For segmentation, again, due to limited computing resources, we can only use smaller images and smaller network, but the performance is still acceptable.

In my understanding, the segmentation task is another form of image construction. It first use convolutional layers and pooling layers to abstract the image to a different form of parameters and use transpose convolutional layer to "restore". The process can be viewed as first get the information of the whole image then use the information to "get back to" the image and analyse the category of each pixel. What's more, the segmentation task is an "update" of classification task. So the three tasks of the project is actually correlated.

During this project, we as a team work together and it is a valuable experience. We learnt a lot from out tasks and even more from each other. Thanks to the teacher and TAs for this excellent lesson !