Assignment 2: Covolutional Neural Networks for CIFAR-10 multiclass image classification, Deep Learning Fundamentals

Yuan Tian
a1818751@student.adelaide.edu.au

1. Introduction

ResNet CNN model is introduced in this paper to classify images on CIFAR10 dataset from Kaggle.com/quanbk/cifar10 and the aim of the paper is to understand the development of related CNNs with algorithms and implement it to improve the accuracy in the classification. The result of accuracy for each model of CNN has compared and illustrated based on changes of parameters or functions. The result has shown that ResNet achieved highest accuracy with faster running speed and the following paragraphs will address the methods of implementations behind.

2. Background

AlexNet, VGGnet, and ResNet are the three typical deep learning networks for enhancing ImageNet classification accuracy that are mainly examined in the existing literature.

AlexNet was suggested by Alex Krizhenvsky as one of the earliest deep networks for enhancing ImageNet classification accuracy significantly, compared to traditional methodologies. It has 5 convolutional layers preceding 3 fully connected (FC) layers [1]. AlexNet uses ReLU (Rectified Linear Unit), given by f(x)=max(0,x) for the non-linear part, rather than a Tanh or Sigmoid function that was the prior standard for traditional neural networks. The ReLu layer is located after each and every convolutional and FC layers. What makes the ReLU more advantageous than Sigmoid is that it trains much swifter. This is because the derivative of Sigmoid turns extremely small in the saturating region, and as a result, the updates to the weights nearly vanish, which is known as vanishing gradient problem. Another advantage of ReLU is that it helps to address the over-fitting issue by utilising a Dropout layer after each FC layer. Dropout layer involves a probability (P=0.5) which is separately applied each neuron of the response map. The activation is randomly switched off by the Dropout layer according to the probability P=0.5. Despite having a role in avoiding bad local minima, it requires a doubled number of iterations for convergence [2].

VGGNet was developed by VGG group, Oxford with the need of reducing the parameters in the convolutional layers and improving training time,. VGGNet improves the features of AlexNet by changing large kernel-sized filters with multiple smaller 3x3 kernels. Why these smaller-size kernel is better than the larger one is that the depth of the network is increased by multiple non-linear blocks which allows the network to learn more complex and representative features involving a lower cost. Using multiple same sized blocks is commonly applied in the networks after VGGNet. Meanwhile, finer level properties of the image are also retained with the top-5 accuracy of 92.3% on ImageNet with the help of the 3x3 kernels in VGGNet. This is achieved by starting the width of the network at a low value of 64 and increasing by a double after each sub-sampling/pooling layer [3].

While residual networks (ResNets) enable the training of deep networks by using residual models to construct the network. ResNets have similar structure as VGGNet that they are mostly made up of 3x3 filters. The residual models are formed by inserting shortcut connection.[4] Like Google Net, ResNets utilise a global average pooling preceding the classification layer. This allows ResNets to be learned with network depth of up to 152. Achieving 95.51 top-5 accuracies, ResNets have higher accuracy than VGGNet and GoogleNet and being more computationally-efficient than VGGNet. The usefulness of residual networks can be further demonstrated by the results of previous experiments that the degradation problem in terms of the higher validation error in the plain 34 layer network than the 18 layers plain network can become much lesser when being converted into the residual network.

3. A Description of Method

ResNet has been chosen as the CNN model to detect images on CIFAR10 data set for this assignment. The literature review above have addressed that the deeper CNN network will occur the vanishing gradient probloem as the gradient vanishingly small due to back propagating the derivative. The shortcut connections as shown in the image is use to solve the vanishing gradient problem which can provide residual connections straight to earlier layers.

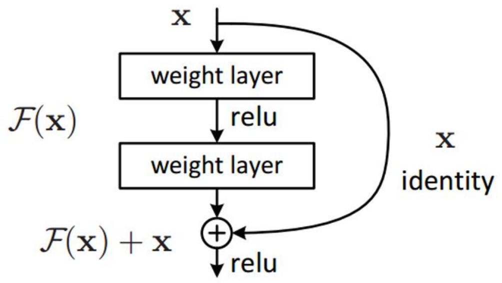

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 1. Residual learning: a building block [5].

In explanation of algorithms, the residual function of F(x): = H(x) -x is used then gives H(x):= F(x) + x. Therefore, F(x) represents the non linear layers and x is the identity function. This approach is used instead of direct mapping because the network is allowed to fit the residual mapping. This way can obviously skip those layers which hurt the architecture's performance, therefore the vanishing gradient problem will be solved when training deep neural network.

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}.$$

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}.$$

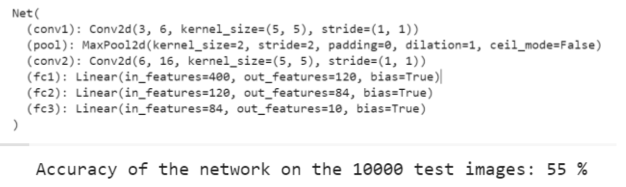
The first residual block function is the identity shortcuts which is used when output dimension is the same as the input dimension.The second residual block function is used when two dimension are not the same. The projection shortcut is to match the dimensions with extra zero entris padded. X is inpuit vector and y is output vector. σ in formula F = W2σ(W1x) represents

ReLU and shortcut connect is shown as F + x with addition elements. In this way, it can be obviously seen the comparison between plain and residual networks even with same parameters. The forma of F which is flexible residual function. It can contain 2 or 3 layers even more and the advantage of residual will be clearer.

4. Experimental Analysis

The experiment was conducted to improve the accuracy of CNN to detect images on CIFAR10 data set.

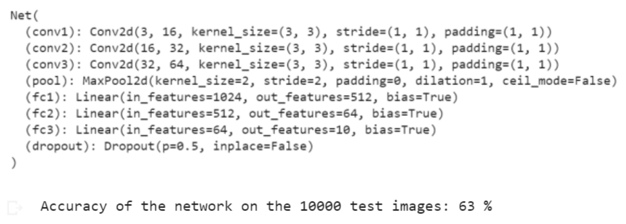
The structure of the basic CNN given by workshop is shown below. It consists two convolutional layers and one pooling layer between them, then followed by three full connected layers. The loss function and Optimization functions were used to calculate the difference between desired output and actual output. Stoichastic Gradient Descent optimizer is used in this case which can reduce the losses. Learning rate was set to 0.01 as it has huge influence to the optimizer.



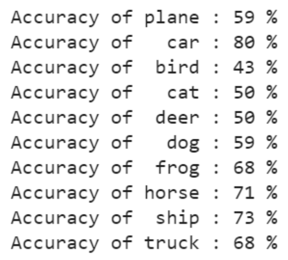
```
Net(
  (conv1): Conv2d(3, 6, kernel_size=(5, 5), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (conv2): Conv2d(6, 16, kernel_size=(5, 5), stride=(1, 1))
  (fc1): Linear(in_features=400, out_features=120, bias=True)
  (fc2): Linear(in_features=120, out_features=84, bias=True)
  (fc3): Linear(in_features=84, out_features=10, bias=True)
)

  Accuracy of the network on the 10000 test images: 55 %
```

The result of this basic CNN after 5 epochs has shown that the running loss of between 1.16-1.2. The accuracy of the network on 10,000 images testing is 55%. The accuracy of the basic CNN model is not very high and hyper parameters need to modify to get a better accuracy.

The second CNN model was introduced with added more layers and used larger batch size. It is assumed that more layers will have deeper network to learn and larger batch size will have more training examples which will result as higher accuracy. The second CNN model has structure shown below:



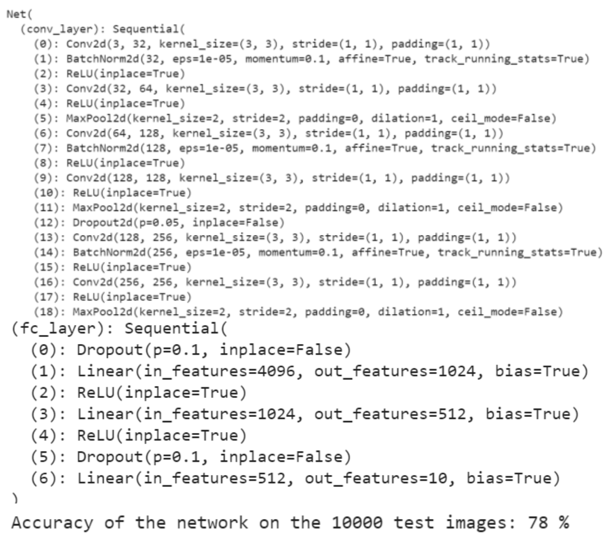
```
Net(
  (conv1): Conv2d(3, 16, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv2): Conv2d(16, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (conv3): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
  (pool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  (fc1): Linear(in_features=1024, out_features=512, bias=True)
  (fc2): Linear(in_features=512, out_features=64, bias=True)
  (fc3): Linear(in_features=64, out_features=10, bias=True)
  (dropout): Dropout(p=0.5, inplace=False)
)

  Accuracy of the network on the 10000 test images: 63 %
```



```
Accuracy of plane : 59 %
Accuracy of   car : 80 %
Accuracy of  bird : 43 %
Accuracy of   cat : 50 %
Accuracy of  deer : 50 %
Accuracy of   dog : 59 %
Accuracy of  frog : 68 %
Accuracy of horse : 71 %
Accuracy of  ship : 73 %
Accuracy of truck : 68 %
```

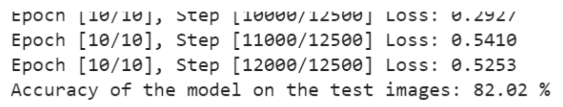
The modified CNN networks consist three conv layers and then followed by one pooling layer and three full connected layers. The key changes are the increase of channels for each layer. In third Conv layer, 32 and 64 channels were used which can have more complex matrices to do dot product during feature selection. The last layer is drop out layer which can reduce over fitting in case the model is too complex. The final accuracy number shows 63% which has improved 8%. The increase of conv layers and kernel size have helped to improve the overall accuracy and the best single class accuracy is 80% for car.

The next improvement was still based on increasing number of layers to increase the complexity and use more kernel to extract features. The learning rate will reduced to 0.001 which is 1/10 the original rate because the network is complex enough and it is expected to converge slowly. The third version CNN structure has shown as below with three convolutional layer blocks and each block includes batch normalization function and two ReLus. The batch normalization layer can let each layer work more independently in the network and therefore can improve the learning skills.



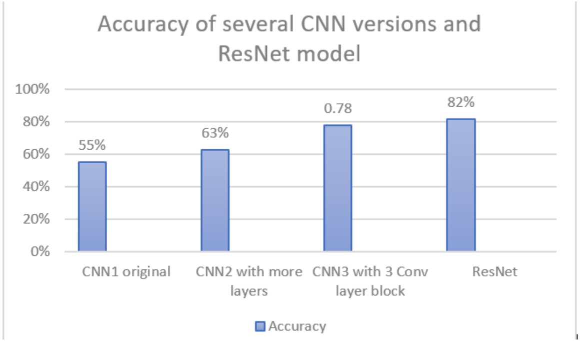
```
Net(
  (conv_layer): Sequential(
    (0): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): ReLU(inplace=True)
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (6): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (7): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (8): ReLU(inplace=True)
    (9): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (10): ReLU(inplace=True)
    (11): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (12): Dropout2d(p=0.05, inplace=False)
    (13): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (14): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (15): ReLU(inplace=True)
    (16): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (17): ReLU(inplace=True)
    (18): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
 (fc_layer): Sequential(
   (0): Dropout(p=0.1, inplace=False)
   (1): Linear(in_features=4096, out_features=1024, bias=True)
   (2): ReLU(inplace=True)
   (3): Linear(in_features=1024, out_features=512, bias=True)
   (4): ReLU(inplace=True)
   (5): Dropout(p=0.1, inplace=False)
   (6): Linear(in_features=512, out_features=10, bias=True)
 )
 Accuracy of the network on the 10000 test images: 78 %
```

The above network has well improved parameters with 18 Conv layers and 6 Fc layers. The channels were from 3, 32 to 256, 256. ReLU function was used to reduce gradient problems and bacthnormal from 32 to 256. The result of this improved CNN has accuracy of 78% which has 15% improvement than the previous version. However, it is believed that the accuracy can still be improved.



```
Epoch [10/10], Step [10000/12500] Loss: 0.2927
Epoch [10/10], Step [11000/12500] Loss: 0.5410
Epoch [10/10], Step [12000/12500] Loss: 0.5253
Accuracy of the model on the test images: 82.02 %
```

The deeper network will easily have vanishing gradient problem and also cause different dimension between inputs and out puts. The ResNet of 3 layers were used to compare the result to previous versions of CNN structures. The above running data shows that ResNet with 10 epochs has accuracy of 82% and the running speed is much faster than previous CNN model.



5. Code

https://github.com/Yuan-Tian2020/ResNet-CNN.git

git@github.com:Yuan-Tian2020/ResNet-CNN.git

6. Conclusion

This paper has analyzed deep learning networks of some CNN model and specifically focused on ResNet by showing the comparison of the accuracy of detecting image on CIFAR10 dataset. The idea of doing this experiment analysis is to understand the algorithms behind and understand how each model is developed. It can be seen that ResNet model achieved highest accuracy than other CNN models in this paper. ResNet has shown the ability to train large number of layers without increasing running error percentage and also reduce vanishing gradient problem by using identity mapping. Despite ResNet model has 82% accuracy with epochs 10, it still has a lot to improve such as adding more data augmentation or add more regularization. On the other hand, CNN3 has improved from 55% to 78% accuracy by adding more layers and regularization functions. The results have successfully proved the assumptions made and it is expected to test and compare more CNN networks such as VGG, AlexNet and GoogLeNet.

References

[1] N. Gomez Blas, L. F. de Mingo Lopez, A. Arteta Albert, and J. Martinez Llamas, “Image Classification with Convolutional Neural Networks Using Gulf of Maine Humpback Whale Catalog,” Electronics (Basel), vol. 9, no. 5, p. 731, 2020, doi: 10.3390/electronics9050731.

[2] Alom, Md Zahangir, et al., The history began from alexnet: A comprehensive survey on deep learning approaches, available at<https://arxiv.org/abs/1803.01164>.

[3] U. Muhammad, W. Wang, S. P. Chattha and S. Ali, "Pre-trained VGGNet Architecture for Remote-Sensing Image Scene Classification," 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 1622-1627, doi: 10.1109/ICPR.2018.8545591.

[4] A. Mahmood, M. Bennamoun, S. An and F. Sohel, "Resfeats: Residual network based features for image classification," 2017 IEEE International Conference on Image Processing (ICIP), 2017, pp. 1597-1601, doi: 10.1109/ICIP.2017.8296551.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.