CSCE 636 Deep Learning - Project

Jin Hyun Park
Texas A&M University
jinhyun.park@tamu.edu

Abstract

This paper introduces the model that combines Dense Convolutional Network (DenseNet) [5], Bottleneck structure, and Residual connection [4], namely, Dual Path Network [2]. After explaining the notion of the DenseNet, Bottleneck, and Residual connection in detail, the structure of the Dual Path Network will be investigated. The DenseNet received CVPR Best Award in 2017 and is known for its simple structure compared to its performance. The idea of the Bottleneck and Residual connection came from ResNet [4]. In this project, the test accuracy of 90% is acquired by implementing the Dual Path Network with CIFAR-10 testing dataset [7].

1 Preliminary Facts

1.1 Residual connection

Before the ResNet structure appeared for image recognition tasks, a large Convolutional Neural Network (CNN) structure suffered from having many layers [4; 1]. However, unlike other CNN structures introduced previously (e.g. GoogleNet [10], VGG [9], and AlexNet [8]), the ResNet used the idea of residual connection (a.k.a. skip connection or projection shortcut) and it helped the model to maintain high accuracy even when using deep layers. The residual connection enabled the model to reduce the effect of the vanishing gradient. This can be depicted by the following figures (See Figure. 1)

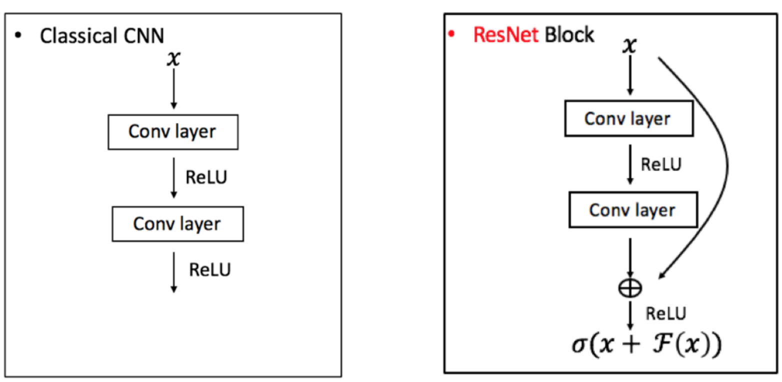

Figure 1: Comparison of basic CNN architecture and ResNet. Image from He et al. [3].

1.2 Bottleneck Structure

In short, the Bottleneck structure is a method to obtain *essential* image features by using a similar idea to AutoEncoder. In other words, the structure is similar to extracting latent features in the

AutoEncoder. The AutoEncoder encodes the important features of the image and the features are called latent vectors. Analogously, the Bottleneck structure is able to obtain latent representations of the input with reduced dimensionality.

The followings are the positive aspects of using the Bottleneck structure. Firstly, it enables us to use fewer parameters which leads to lower computational costs compared to the standard residual block. In addition, it can be used to obtain a representation with reduced dimensionality by using 1x1 convolution. This can be represented by the following figure (See Figure 2). On the other hand, there are also negative aspects of bottleneck structure. For example. we might lose some important features (or information) of the image because we force dimensionality to be reduced. However, it is generally known that the positive aspects of the bottleneck structure overwhelm the negative aspects[1].

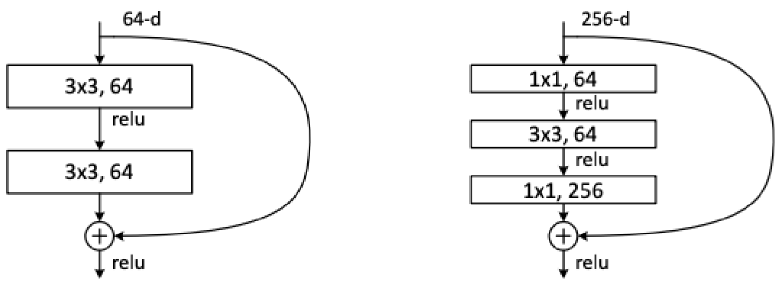

Figure 2: The Bottleneck structure from ResNet. Left: a standard residual network. Right: a standard residual network with the Bottleneck structure. Image from He et al. [4].

1.3 DenseNet

The idea of DenseNet is slightly different from ResNet. Firstly, in ResNet, we applied residual connection for each block. And the residual connection is added to the output of the next block (See Figure 1). However, in DenseNet, it applies residual connection not only to *the output of the next block* but also to *other outputs of the blocks*. Secondly, different from ResNet, it concatenates the residual connection and the output (e.g. `concat(x, f(x))`) rather than adding them up together (e.g. `add(x + f(x))`). To put it another way, original features are concatenated in a channel-wise manner to every filtered feature. This can be depicted by Figure 3.

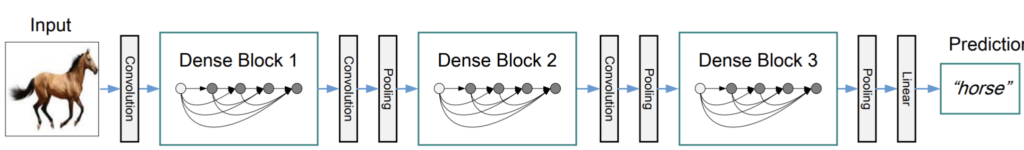

Figure 3: DenseNet structure. Image from Huang et al. [5]

There are many advantages to using the above structure. Firstly, the gradient of the error signal is stronger than the other models. Secondly, the number of parameters can be reduced and computational efficiency can be improved. Lastly, the softmax classifier uses features of all complexity levels, unlike other models. Consider other models: they use only the last feature for classification which implies that the softmax uses only high-level features. In contrast, DenseNet uses every level of features, from low-level to high-level features.

[1] I skipped the description about *projection shortcut*. However, it is used in my work.

1.4 Summary of the Residual connection and densely connected network

We can summarize the structure of the Residual Network and Densely connected Network (DenseNet) by the following. Take a look at Figure 4. There are two pipes.

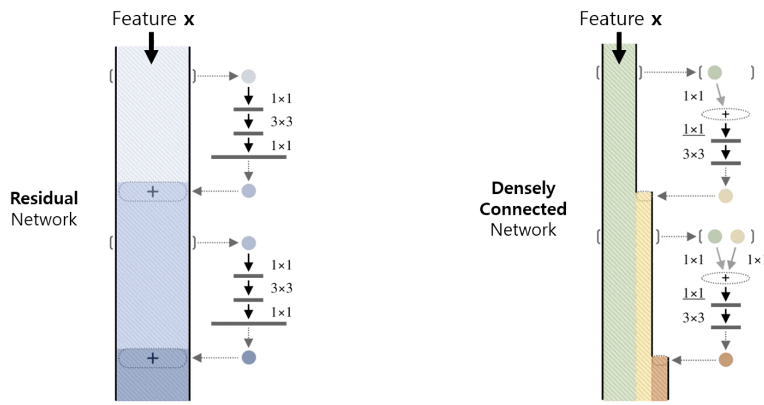

Figure 4: Residual network vs Densely connected network. Image from [6].

The Residual connection (= skip connection) with bottleneck structure is applied for the Residual network (LHS). It can be seen that the Residual connection is *added* to the original pipe. In the densely connected network, the residual connection is *concatenated* with the original pipe (RHS). Hence, the pipe becomes thicker for every residual connection. It can be also observed that we use the bottleneck structure. Due to the above structure, feature redundancy of the residual network is low compared to the densely connected network. And, the densely connected network is better than the residual network in terms of exploring new features. We can encapsulate the following facts in Table 1.

Table 1: The lower the feature redundancy the better and the higher the new feature exploration the better.

Structure	Feature Redundancy	Exploring New Feature
Residual Network	Low	Low
Densely Connected Network	High	High

2 Dual Path Network

As we have seen above, there are pros and cons for each model: residual network and densely connected network. In fact, by introducing the Dual path network, it is possible to enjoy the advantages of the residual network and the densely connected network at the same time. Namely, it is possible to decrease feature redundancy and increase new feature exploration. The following figure illustrates the structure of the Dual path network (See Figure 5).

Firstly, features from both dense and residual paths are added after 1x1 convolution. Secondly, recalling the idea of the Bottleneck structure, apply 3x3 convolution and 1x1 convolution. After that, feature vectors are split into two parts: one for the dense path and the other for the residual path. It is important to note that the channel size should be matched for the residual path. After splitting the feature vectors into two parts, we apply element-wise *addition* to the residual path and *concatenation* to the dense path. It can be noticed that the idea of concatenating features from DenseNet's paper is applied to the Dual path network as well. This implies that the following equation holds for every block (or stack) of the DenseNet.

$$\text{(current-pipe channel dimension)} \leftarrow \text{(previous-pipe channel dimension)} + \text{(\# of newly added channel dimension)}$$

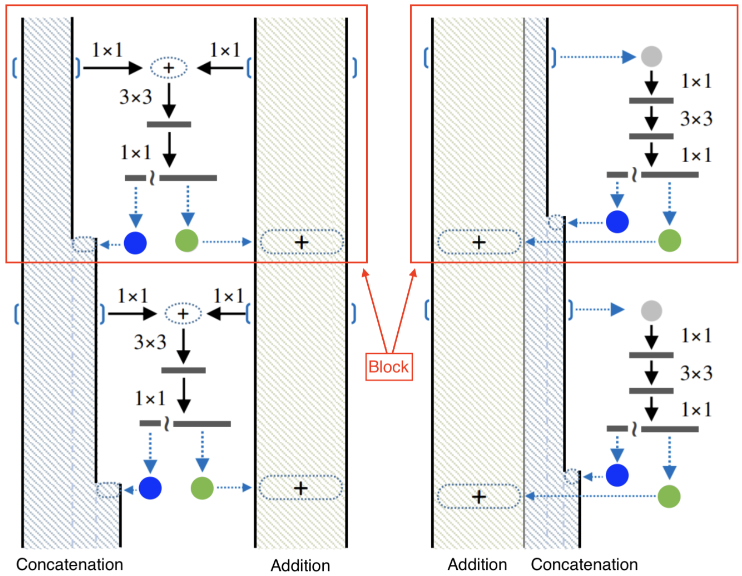

Figure 5: Dual Path Network. Baseline image from [2]. The LHS of the image and the RHS of the image are the same. They both depict the Dual path network. The feature dimensions are split into two parts: blue and green circles. The blue circle is used for concatenation (similar to DenseNet) and the green circle is used for addition (similar to ResNet).

where (# of newly added channel dimension) = (green circle in Figure 5)

The above figure only shows the mechanism of the Dual path network for only two blocks (or stacks). This can be stacked together to be similar to ResBlock (a mechanism that is used for the ResNet architecture that was discussed)

3 Experimental Setup

3.1 Development Environment

Pytorch version 1.12.1, CUDA version 11.3, Numpy version 1.12.6, and Python version 3.7.15.

3.2 Hyper-parameters

Batch size = 128, SGD optimizer with momentum = 0.9, weight decay = 1e-4, learning rate = 1e-2, cross-entropy loss, and finally learning rate scheduler with StepLR (step size = 30 and gamma = 0.01).

3.3 Image pre-processing

Applied random cropping, random horizontal flip with probability 0.5, random vertical flip with probability 0.5, and normalization with mean = (0.4914, 0.4822, 0.4465), standard deviation = (0.247, 0.243, 0.261). The mean and standard deviation are from the following URL: `https://github.com/kuangliu/pytorch-cifar/issues/19`. Applied the same normalization constant to the testing dataset.

3.4 Network Structure

Four dense blocks are used similar to the original DenseNet paper (See Figure 3). The first block contains 3 Bottlenecks, the second block contains 4 Bottlenecks, the third block contains 10 Bottlenecks, and the final block contains 3 Bottlenecks. More details can be found in `Network.py`.

4 Results

Validation (5000 samples) and testing (10000 samples) accuracy are obtained for every 10 epochs, starting from epoch 1 to 200 (See Figure 6). The highest accuracy for the testing data was about 90%. It can be seen that a bit of over-fitting aroused when comparing the validation accuracy and testing accuracy. Considering the original paper, I believe that it will be possible to get a better test accuracy.

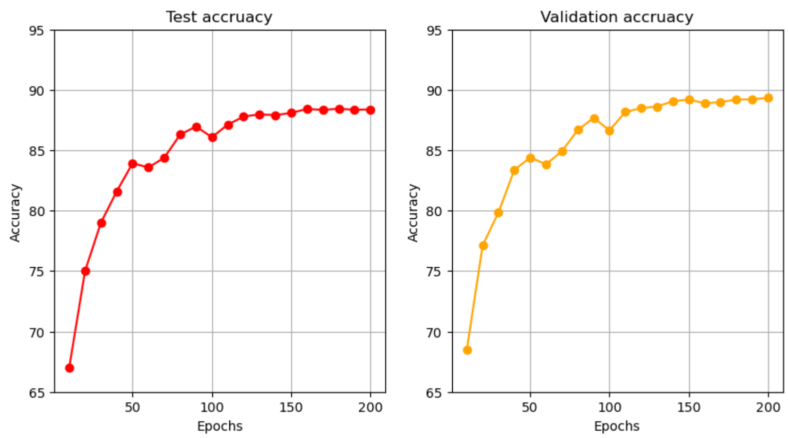

Figure 6: Validation and Test accuracy. The test accuracy is obtained from weights trained with the full dataset (= training + validation dataset).

5 Conclusion

This paper introduced the structure of the DenseNet, Bottleneck, and Residual connection in detail. The Dual path network was described based on the above. By using the Dual path network that combines the ideas of the Densenet, Bottleneck, and Residual connection, it was able to get better performance than the standard ResNet. In HW2 ResNet implementation, I got a test accuracy of 80% (90 epochs). On the other hand, in this project, I got a test accuracy of 90% (90 epochs).

References

[1] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8(1):1–74, 2021.

[2] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. *Advances in neural information processing systems*, 30, 2017.

[3] Juncai He, Jinchao Xu, Lian Zhang, and Jianqing Zhu. An interpretive constrained linear model for resnet and mgnet. *arXiv preprint arXiv:2112.07441*, 2021.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[6] Sanghyun Jung. Dual path networks, 2021. URL: `https://www.youtube.com/watch?v=9qpyryz9ppQ`.

[7] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL: `http://www.cs.toronto.edu/~kriz/cifar.html`.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.