An empirical evaluation of DenseNet

Kumar Shridhar
Technical University Kaiserslautern
Kaiserslautern, Germany
k_shridhar16.cs.uni-kl.de

Marcus Liwicki
Technical University Kaiserslautern
Kaiserslautern, Germany
liwicki@cs.uni-kl.de

Abstract

With the introduction of skip connections, deep neural network models can now be substantially deeper and can be trained more efficiently. This paper takes Dense Convolutional Network (DenseNet) [3], one of the state-of-the-art architectures into account, and shows the key advantages of connecting each layer to every other layer in a feed-forward fashion. The paper also discusses all the compelling advantages of DenseNet and evaluates the performance of it's architecture on a Kaggle dataset (Dog-Breed Identification) [5]. Additionally, it also discusses strategies to reduce the memory consumption during training.

1. Introduction

Convolution Neural Networks (CNNs) [10] have led to a paradigm shift in the field of Computer Vision. CNNs have recently achieved state-of-the-art results in various Image Recognition and Object Detection tasks. This has been made possible by the recent advancements in GPU hardware and availability of huge reliable datasets. Improvements in computer hardwares and network structures have enabled the training of very deep CNNs. It has also made the training process faster and more efficient. LeNet5 [11] introduced the first CNN architecture consisting of just 5 layers. Followed by which, VGGNet was introduced with 19 layers, and lately Residual Networks (ResNets) [2] have surpassed the 100-layer barrier.

As the number of layers in a deep neural network increases, it is accompanied by the vanishing gradient problem. ResNets [2] and Highway Networks [15] bypass the signal from one layer to the next via identity
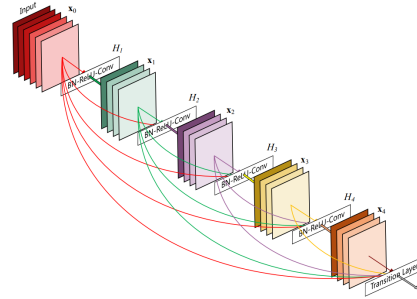

Figure 1. A 5-layer dense block with a growth rate of k = 4. Each layer takes all preceding feature-maps as input.

connections. This counters the issue of vanishing gradient. However, FractalNets (Ultra-Deep Neural Networks without Residuals) [9], without including any pass-through or residual connections achieve a comparable performance to deep Residual Networks. This demonstrates that residual representations may not be fundamental to the success of extremely deep convolutional neural networks.

Hunag *et al*. [3] proposed an architecture that addresses the vanishing gradient issue with a simple connectivity pattern to ensure maximum information flow between layers in the network. In this method, all layers (with matching feature-map sizes) are connected directly with each other. To preserve the feed-forward nature, each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers.

Figure 1 illustrates this layout schematically. Crucially, in contrast to ResNet, the features are never combined through summation before they are passed into a layer, instead the features are combined by con-

1

Figure 2. A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change the feature-map sizes via convolution and pooling.

catenation. Hence, the `lth` layer has `l` inputs consisting of the feature-maps of all preceding convolutional blocks. Its own feature-maps are passed on to all `L-l` subsequent layers. This introduces $L(L + 1)/2$ connections in an `L`-layer network, instead of just `L` connections, as in traditional architectures. Because of its dense connectivity pattern, the architecture is named as Dense Convolutional Network (DenseNet).

2. DenseNet

Traditional convolutional feed-forward network connects the output of one layer as the input to the next layer. ResNet [2] further adds a skip connection to bypass the layers with an identity function. This helps the gradient to flow directly through the identity function.

DenseNet on the other hand introduces direct connections from any layer to all subsequent layers. Consequently, the `lth` layer receives the features from all preceding layers, `x0, . . . , xl-1`, as input:

$$x_l = H_l([x_0, x_1, ..., x_{l-1}])$$

where `[x0, x1, . . . , xl-1]` refers to the concatenation of the feature-maps produced in layers `0 . . . . . l-1` and `Hl` is any non linearity function.

3. Experiments

3.1. Performance Evaluation

DenseNet obtains significant improvements over the state-of-the-art results on four highly competitive object recognition benchmark tasks (CIFAR-10[6], CIFAR-100[7], SVHN[12], and ImageNet [14]). To

Figure 3. The architecture has 2 Transition Up (TU) blocks an 2 Transition Down (TD) blocks and gray arrows represent the skip connections from TU blocks to TD blocks.

verify that the DenseNet is as good as mentioned by the authors, it was tested on a Kaggle competition task (Dog-Breed Identification [5]). The results were compared with ResNet [2] and InceptionNet[16] that won the ImageNet [14] Image Recognition Challenge in the past.

As seen in Table 1, DenseNet performs significantly better (89.9% accuracy) than ResNet and Inception-Net for Dog-Breed Identification [5]. The performance of DenseNet was even significant than other models when no preprocessing was performed (as seen in Table 2). It shows that connecting every layer's output to each subsequent layers allowed the model

Model	Data Augmentation
ResNet50	88.2%
InceptionNet	87.4%
DenseNet	89.9%

Table 1. Accuracy of different model architectures on Dog-Breed Identification task.

Model	Without Data Augmentation
ResNet50	82.7%
InceptionNet	81.9%
DenseNet	86.4%

Table 2. Accuracy of different model architectures on Dog-Breed Identification task without data augmentation.

to learn the features more effectively. Hence, lesser data augmentation is needed for DenseNet as compared to other models for the given task. The implementation of DenseNet on Dog-Breed Identification challenge is available at: `https://github.com/kumar-shridhar/CNN_Architectures`.

Furthermore, to evaluate the performance of DenseNet, it was applied to the task of semantic segmentation. CNNs are used in the state-of-the-art methods for semantic segmentation. The architecture for most tasks perform (1) a downsampling of the image to get the features, (2) then an upsampling to get the image back, (3) and post processing like Conditional Random Field (CRF)[8] and pre processing modules like data augmentation is done to get better results. A lot of different CNN architectures were tried and DenseNet achieved the state-of-the-art results as observed by Jegou *et al.* [4]. Additionally, they also achieved state of the art results on urban scene benchmark datasets, such as CamVid[1] and Gatech, without any post-processing or pre-training. This shows the effectiveness of DenseNet on a task other than Image Recognition or Object Detection.

3.2. Memory Consumption

DenseNet implementation in general require higher memory as seen in Figure 4. If not properly managed, pre-activation, batch normalization and contiguous convolution operations can produce feature maps that grow quadratically with the network depth. To counter these problems, the concept of Shared Mem-

ory Storage was proposed by Pleiss *et al.* [13]. During the forward pass, all intermediate outputs were assigned to shared memory blocks. During back-propagation, the concatenated and normalized features were computed dynamically.

3.3. Training Time

DenseNet's training time per epoch is significantly lower than ResNet and InceptionNet on the same dataset. The experiments showed a decrease of 50% training time for DenseNet per epoch. However, to achieve the best possible results, DenseNet needs to be trained over 100 epochs as compared to 20 epochs for ResNets. This is 5 times more than the other models. So, a tradeoff exists between training time and accuracy for DenseNet.

4. Conclusion

A large community of researchers believed that the approach of DenseNet was not very novel and it has been directly/indirectly applied to a lot of tasks earlier. But when observed closely, a lot of thought has been put in by the authors to design the architecture. This has been proved by the state-of-the-art results achieved by DenseNet in a variety of Image Recognition competitions. The concept has been further tested by a series of experiments performed in this paper and by several other researchers[4]. Overall, DenseNet is not a groundbreaking idea but the way it is implemented has achieved desired results and has proven useful for various tasks.

5. Future Scope

The core idea of DenseNet has opened a lot of research areas. It has already been applied in many domains (Document Analysis, Image Recognition, etc.) and it is believed it will be applied in many other areas in the near future. It would be interesting to see if an architecture similar to DenseNet can solve the need of huge amount of data for training deep learning models. The idea of DenseNet can be combined with the idea of ResNet (introduce skip connections), to avoid learning of redundant features over the layers. This is hypothesised to improve the overall training time for DenseNet and to solve problem of overfitting.
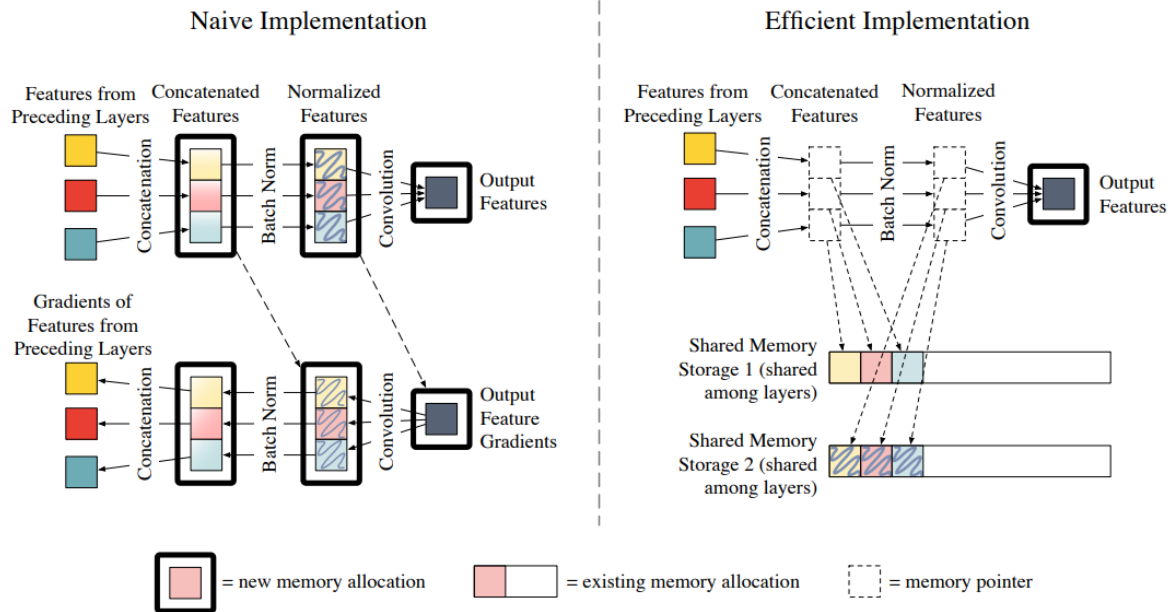
Figure 4. Original DenseNet implementation is on the left and its memory efficient implementation is on the right. The memory efficient implementation stores the output of the concatenation, batch normalization, and ReLU layers in temporary storage buffers, whereas the original implementation allocates new memory.

References

[1] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV (1)*, pages 44–57, 2008.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[3] G. Huang, Z. Liu, and K. Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[4] S. Jégou, M. Drozdzal, D. Vázquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. *CoRR*, abs/1611.09326, 2016.

[5] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, June 2011.

[6] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research).

[7] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-100 (canadian institute for advanced research).

[8] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[9] G. Larsson, M. Maire, and G. Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *CoRR*, abs/1605.07648, 2016.

[10] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, Dec. 1989.

[11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998.

[12] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[13] G. Pleiss, D. Chen, G. Huang, T. Li, L. van der Maaten, and K. Q. Weinberger. Memory-efficient im-

plementation of densenets. *CoRR*, abs/1707.06990, 2017.

[14] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[15] R. K. Srivastava, K. Greff, and J. Schmidhuber. Training very deep networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2377–2385. Curran Associates, Inc., 2015.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.