Applying ResNet and Inception Module Techniques for Skin Cancer Classification on Breast Histopathology Images Dataset with Convolutional Neural Networks

1st Pedro Almeida
Department of Ocean & Mechanical Engineering
Florida Atlantic University
Boca Raton, United States
palmeida2016@fau.edu

***Abstract*—**

***Index Terms*—artificial neural network, machine learning, convolutional neural network**

I. Introduction

Over the past 40 years [1], the field of object detection in machine learning has made massive strides in classification and detection capabilities. In addition to the expected improvements stemming from larger datasets, deeper models, and more powerful machines, much of the advancement owes to new and improved network architectures and novel algorithms. These techniques, often designed with specific datasets in mind, take advantage of the structure of the data to optimize the amount of information extracted per layer.The difficulty in designing the networks stem from the increasingly numerous design choices in determining hyper-parameters (width[1], filter sizes, strides, pooling layer parameters, dropout rates, *etc.*) and the large selection of techniques in constructing the network architecture [2]–[10].

The family of inception models [3], [11], [12] tackled the problem by introducing an extra dimension cardinality: the number of independent paths within a layer. The central method employed involves splitting the input of a layer into lower-dimensional embeddings whereby each is transformed by a set of specialized filters with different kernel sizes, which are then concatenated. The *split-transform-merge* method has been shown to approximate the representational power of large, dense layers, but at a substantially lower computational complexity. Despite showing good improvements in accuracy, the inception module suffers from a difficulty in tailoring filter numbers and sizes for different datasets/objectives; although the right combination can yield significant improvements to the accuracy, it is generally unclear as to how to adapt the Inception module architecture of a given problem, especially in combination with other complex transformations and hyper-parameters are involved.

Another strategy introduced by the family of ResNet models and its variants proposes the use of residual learning through stacking modules of the same topology. This simple rule creates a an effective strategy for constructing very deep network architectures that are applicable to a wide range of tasks, owing partially to the reduction of free choices for hyper-parameters.

The objective of the report is to explore the combination of some of these techniques [3], [4] through the Breast Histopathology Dataset [13]. The dataset contains 162 whole mount slide images of a breast cancer variant called Invasive Ductal Carcinoma scanned at forty-times magnification. As of 2019, breast cancer affects 1 in 8 [14] women during their lifetime, registering roughly 268,600 new cases every year. Invasive Ductal Carcinoma, also known as Infiltrating Ductal Carcinoma, is the most common type breast cancer, accounting for 80% [15] of breast cancer diagnoses. As is the case for most cancers, early detection accounts for most of successful treatments of the disease; as the cancer progresses to the later stages, the odds of spreading to nearby organs increases [16], thus furthering the risk of the death of the patient.

We propose a Convolutional Neural Network architecture

II. Related Work

Since the introduction of Convolutional Neural Networks (CNN), the model architecture off CNNs has remained largely static: a series of convolutional layers followed by optional normalization and pooling layers and finally one or more dense layers reducing to the output. This standard architecture and its variations have reigned dominant in accurately classifying objects in small and large datasets alike, from MNIST, CIFAR, ImageNet, MS-COCO, SVHN, and others. For larger datasets especially, the latest novel idea has been the introduction of deeper networks with dropout layers to correct for overfitting.

The tradeoff for adding more layers to a network is apparent in the time to train; as the model grows in size and increases the number of weights, the backwards propagation in updating the weights takes progressively longer, thus greatly

[1] Width – The number of channels in a layer.

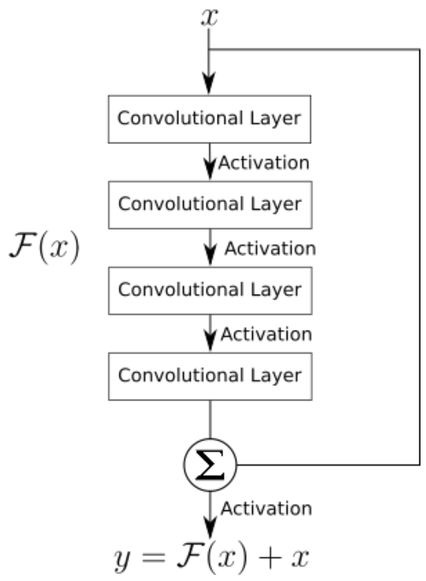

Fig. 1. Model of ResNet Layer with 4 Convolutions.

adding to the required time to train. Furthermore, although the development of newer, more powerful machines and better parallelization and multi-processing techniques can somewhat mitigate the additional computational cost.

III. Methods

A. ResNet Layer

The first CNN modification applied in the model architecture employs residual learning within the convolution layers. As described by He *et al.* [4], let us consider a residual layer within a neural network defined as:

$$y = \mathcal{F}\left(x_i, \{W_i\}\right) + x \tag{1}$$

In Eq. 1, x and y are the input and output vectors of the layer respectively. The function $\mathcal{F}\left(x_i, \{W_i\}\right)$ represents the mapping of inputs to outputs given a set of weights W_i for each of the feature maps of the inner convolutions [4]. For the example shown in Fig. III-A, the input x to the layer passes through 4 convolutional layers, which together constitute $\mathcal{F}(x)$ before the concatenation with the input x. The final concatenation of the output of the feature maps and initial input is a element-wise addition performed channel by channel, which is then passed through an activation layer. In the original paper by He, the residual function $\mathcal{F}\left(x_i, \{W_i\}\right)$ constitutes of 2-3 fully connected layers [4]. The proposed architecture slightly alters the core ResNet model by substituting the fully connected layers by convolutional layers followed by Rectified Linear Unit – also known as ReLU – activation functions.

The proposed architecture implements the ResNet layers as proposed by He *et al.* with an exponentially increasing number of filters in subsequent layers (32, 64, 128, 256). Each of the convolutions is followed by ReLU activation units.

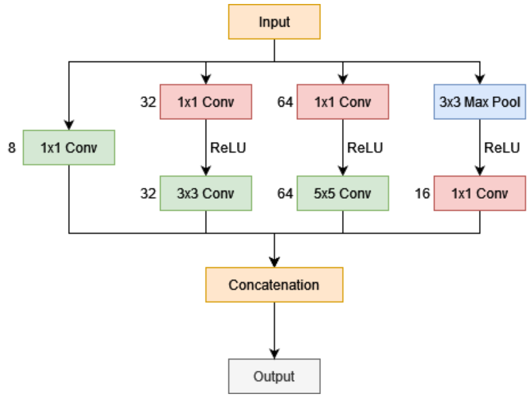

Fig. 2. Model of Inception Layer with 4 parallel paths.

B. Inception Module Layer

The second CNN modification applied in the network architecture is the use of the Inception Module [3] designed by Szegedy *et al.* Unlike the ResNet Layer, the inception module bypasses the deepening of neural networks for better performance through optimal local sparse structures. The structure of the module is founded on the parallelization of inputs into paths with convolutional layers of different kernel sizes.

In earlier[2] layers, one can expect the output of convolutional filters to concentrate in local regions, which could be covered by a layer of 1x1 convolutions in the next layer. However, over larger patches, the outputs of the convolutions will lead to fewer, more spread out clusters that cannot be easily covered by convoultuions. As such, in order to avoid the patch-alignment issues, Szegedy *et al.* proposes restricting the kernel size to 1×1, 3×3, and 5×5. The final addition to the inception module is the addition of 1×1 convolutions before the larger kernel size convolutional layers. The reason behind the choice is that, especially in the originally proposed architecture where inception modules are stacked to create the network, the numbre of 5×5 convolutions can be prohibitely expensive. As such, the merging of output layers in the module would inevitably lead to exponential growth, thus adding immense computational complexity to the model. The proposed 1×1 convolutions before the 3×3 and 5×5 would compute the reductions, thus preveting that growth.

The proposed architecture implements the Inception Module layers as proposed by Szegedy et al. with 8, 32, 64, and 16 filters as shown in Fig. III-B.

C. Network Architecture

D. Data Augmentation

Each of the 162 samples in the dataset have been partitioned to create 277,524 50x50-pixel patches, of which 198,738 are IDC negative and 78,786 are IDC positive. In preparation for the training cycle, the samples were shuffled and distributed into training, validation, and testing sets constituting 60%, 20%, and 20% of the samples respectively. To account for the

[2] Earlier – Refers to layers closer to the input image.

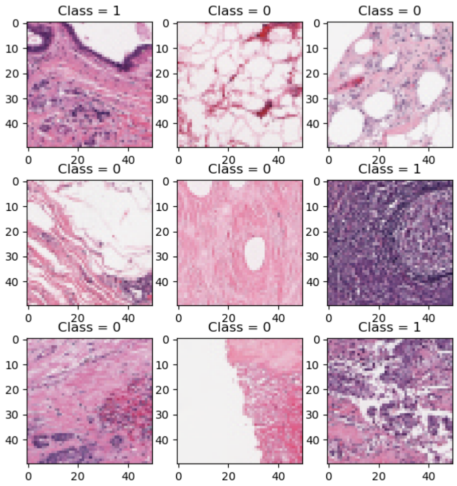

Fig. 3. Example of partitioned images found in dataset.

imbalance in negative and positive samples, the distribution executed ensures the ratio of negative to positive samples is consistent across all three datasets.

The images in each of the sections is then converted from to RGB for training with each color channel zero-centered. Since the spatial orientation and location of the features in the image is not relevant to the classification, the data loader for the model will randomly shuffle the images and choose wether to reflect the image horizontally, vertically, or any combination for each of the epochs, thus providing more variation to the dataset and hopefully generalizing the model.

E. Training the Network

The training of the model applies two important callbacks to maximize the efficiency of the model to converge to the lowest loss. The first of the methods applied is an adaptive learning rate as given in the Keras machine learning library. Using the built-in function *ReduceLROnPlateau*, the model will monitor the change in validation loss between epochs and automatically reduce the learning rate by a factor of 10 if the change in validation loss is lower than the threshold *min_delta* (given as 0.0001) for 3 consecutive epochs.

The CNN was trained over the course of 3 hours on a RTX-2070 GPU as shown in Fig. III-E and III-E.

IV. Results

V. Analysis and Discussion

VI. Conclusions

Acknowledgment

The author would like to thank Dr. Xingquan Zhu for valuable discussions and recommendations in writing this paper.

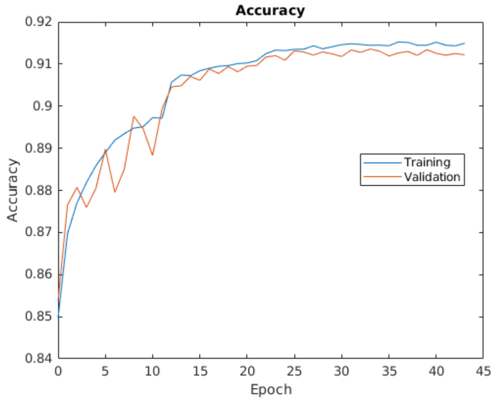

Fig. 4. Plot of training accuracy on train and validation data.

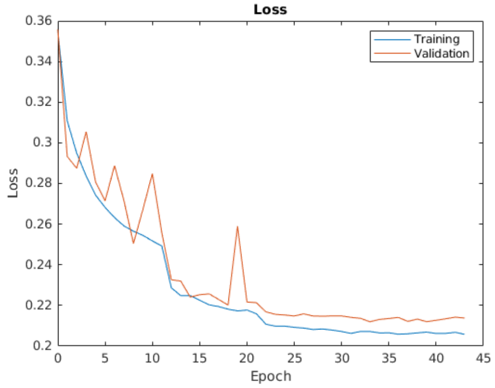

Fig. 5. Plot of training loss on train and validation data.

References

[1] K. Fukushima, “Biological cybernetics neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” p. 202, 1980.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks.” [Online]. Available: http://code.google.com/p/cuda-convnet/

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions.”

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” pp. 770–778, 2016.

[5] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” 8 2016. [Online]. Available: http://arxiv.org/abs/1608.06993

[6] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 9 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[7] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-excitation networks,” 9 2017. [Online]. Available: http://arxiv.org/abs/1709.01507

[8] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, and U. S. Diego, “Aggregated residual transformations for deep neural networks.” [Online]. Available: https://github.com/facebookresearch/ResNeXt

[9] G. Huang, Z. Liu, G. Pleiss, L. Van Der Maaten, and K. Weinberger, “Convolutional networks with dense connectivity,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[10] B. Zhao, H. Xiong, J. Bian, Z. Guo, C. Z. Xu, and D. Dou, “Como: Efficient deep neural networks expansion with convolutional maxout,” *IEEE Transactions on Multimedia*, vol. 23, pp. 1722–1730, 2021.

[11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” 12 2015. [Online]. Available: http://arxiv.org/abs/1512.00567

[12] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." [Online]. Available: www.aaai.org

[13] P. Mooney, "Breast histopathology images," Dec 2017. [Online]. Available: https://www.kaggle.com/paultimothymooney/breast-histopathology-images

[14] C. E. DeSantis, J. Ma, M. M. Gaudet, L. A. Newman, K. D. Miller, A. G. Sauer, A. Jemal, and R. L. Siegel, "Breast cancer statistics, 2019," *CA: A Cancer Journal for Clinicians*, vol. 69, pp. 438–451, 11 2019.

[15] G. N. Sharma, R. Dave, J. Sanadya, P. Sharma, and K. K. Sharma, "Various types and management of breast cancer: An overview," *J. Adv. Pharm. Tech. Res*, vol. 1. [Online]. Available: www.japtr.org

[16] M. Milosevic, D. Jankovic, A. Milenkovic, and D. Stojanov, "Early diagnosis and detection of breast cancer," *Technology and Health Care*, vol. 26, pp. 729–759, 2018.

[17] M. Moreira and E. Fiesler, "Neural Networks with Adaptive Learning Rate and Momentum Terms," *Technique Report 95*, vol. 4, pp. 1–29, 1995.