APPLICATIONS OF NEURAL NETWORK MODELS IN KAGGLE COMPETITIONS

Grace Anulika Eze, Ukraine/Italy
Ivan Franko National University of Lviv/ University of L'Aquila
Faculty of Applied Mathematics and Informatics/DISIM
Intermaths program, Interdisciplinary mathematics
grace.eze@intermaths.eu
Supervisor: Vasyl Vavrychuk

1. Introduction

This paper focuses on implementing and analyzing a neural network model on the ”Malaria Cell Images Dataset” from Kaggle which contains over 27,588 blood smears image patches from patients which is equally separated into two distinct classes since there is no training label file. The difference between the uninfected and the parasitized red blood cells is that certain regions in varying locations of the cell contains the malaria parasite for the parasitized cells. We implemented a Convolutional Neural Network(CNN) with the Keras framework which we use for the medical image classification. CNN's are biologically inspired deep feedforward , backpropragate neural networks which are sparsely connected to the input layer and are designed to process pixel data of an image in an alternating manner till specific features of the problem image is learned.

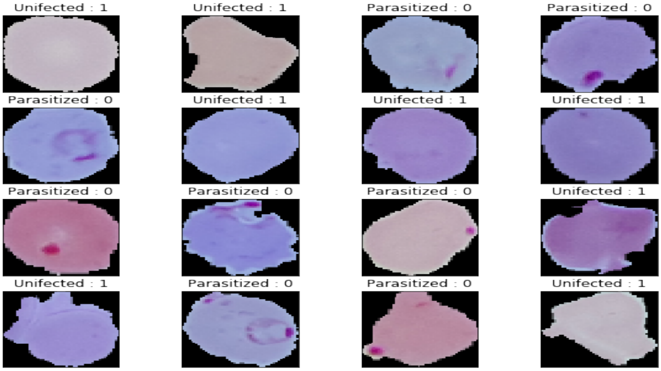


Figure 1: Cells from the Pre-processed Malaria Dataset

Considering a number of m training image samples, a given input image x which consists of an array of pixel intensities has input dimensions $x \in \mathbb{R}^{h\times w\times d}$ where h is the height, w is the width and d is the number of input channels. We are also given p number of filters, k with dimension $k \in \mathbb{R}^{f\times f\times d\times p}$ and for every filter a bias b having dimensions $b \in \mathbb{R}^{p}$. For the convolution layer, the input image from the sample is convolved with p filters where each filter aims to extract and learn specific features from the image. The filter is passed in equal and finite strides over local regions(receptive fields) in the image and at each location computes the sum of the element-wise dot product between it's element and the elements of the connected receptive field as well as adds the bias per filter. Every computation produces p extracted feature map of the input image as our output for the next layer. Mathematically, the process which generates the output of a convolution layer is given by

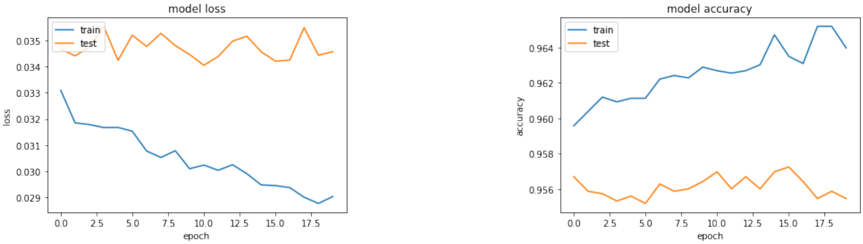
$$\begin{aligned}
(x * k)_{ij} &= \sum_{m=0}^{f-1}\sum_{n=0}^{f-1}\sum_{l=1}^{d} k(m,n)x(i-m,j-n)+b \qquad (1)\\
&= \sum_{m=0}^{f-1}\sum_{n=0}^{f-1}\sum_{l=1}^{d} k_{m,n,l}\cdot x_{i+m,j+n,d}+b \qquad (2)\\
&= \sum_{m=0}^{f-1}\sum_{n=0}^{f-1} k_{m,n}\cdot x_{i+m,j+n}+b \quad if \quad \#d=1 \qquad (3)
\end{aligned}$$

Then an activation function is then applied on the output of the convolutions to introduce non-linearity into the network. We use the ReLU activation to remove all non negative entries and replace them with 0.

For the pooling layer, the spatial size of the output gotten from the activations is reduced by reducing the amount of parameters and computation in the network without much loss of significant information. The output matrix is then vectorized and sent to a fully connected layer with a Softmax/Sigmoid classifier which trains and classifies the images using Back-propagation and predicts the label class of the input image.

2. Results

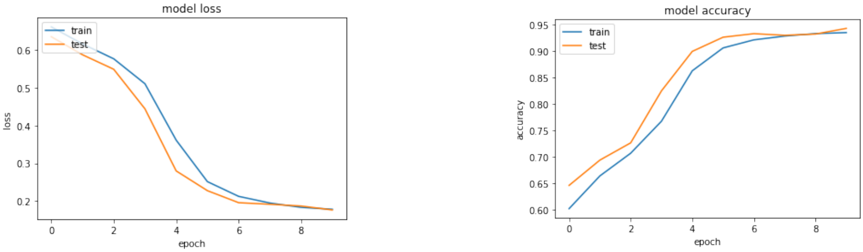
In Figure 2, the graphs showed some overfitting although the accuracy on the train was about 96% while the accuracy for the validation set was very low, therefore an $L1-$regularization parameter, reg $=0.001$ wass introduced to correct the overfitting.



(a) Loss graph

(b) Accuracy graph

Figure 2: The Loss and Accuracy for the first training of the Malaria Dataset



(a) Loss graph

(b) Accuracy graph

Figure 3: Visualizing the Loss and Accuracy for the Malaria Dataset

The accuracy after training the model after a number of iterations with a stochastic gradient descent optimizer and a categorical cross entropy loss function was 0.9348 and the loss was 0.2135. The validation accuracy was 0.9365 and validation loss as 0.2149. After computing the confusion matrix on the test sample to determine the performance of our classification model, we have that:

		Predicted		
		Negative	Positive	Total
Actual	Negative	TP: 2512	FP: 248	2760
	Positive	FN: 177	TP: 2575	2752
	Total	2689	2823	5512

Table 1: Confusion Matrix

class	Precision	recall	f1-score	support
0	0.93	0.91	0.92	2760
1	0.91	0.94	0.92	2752
average	0.92	0.92	0.92	5512

Table 2: Classification Report

References

[1] *Fouzia Altaf, Syed M. S. Islam, Naveed Akhtar, Naeem K. Janjua* Going Deep in Medical Image Analysis: Concepts, Methods, Challenges and Future Directions // February 2019.

[2] *Anthony L. Caterini* A Novel Mathematical Framework for the Analysis of Neural Networks. // Ontario, Canada, 2017.

[3] *Andres Calderon, Sergio Roa and Jorge Victorino* Handwritten Digit Recognition using Convolutional Neural Networks and Gabor filters // Proceedings of the International Congress on Computational Intelligence CIIC, 2003.