RESEARCH

Introduction to the profile areas of data sciences: project 3

Rohan Sanjaybhai Chitte, Natalja Amiridze and Michael Onyebuchi Ohaya

Abstract

Goal of the project: To perform deep learning classification (CNN) on data based on breast cancer histopathaology images.

Main results of the project: Two deep learning models were build, trained and used for classification with AUC values of 70% and 74% respectively.

Personal key learnings: We learnt how to do classification on images using CNN and implement different topologies to optimize results.

Estimated working hours: 10

Project evaluation: this project is very useful for Data Science since we learn the steps for the development of a deep learning model to solve binary classification problems, how to track the performance of the model training and how to make predictions and evaluation.

Number of words: 1361

1 Scientific Background

The Convolutional Neural Network (CNN) is a class of deep learning neural networks. They are most commonly used to analyse visusl images and often work in image classification in everything from healthcare to security. The general idea of the convolutional neural network is to process an input image to classify the object space of values. CNNs consist of the input, output, and the hidden layers. The hidden layer of a CNN comprises of multiple convolutional layer(s), pooling layer(s), and the activation layer(s). The input image is hereby taken into a convolution layer and a series of filters are applied. The filtered data is then input to a pooling layer, which has a function to reduce the dimension size of the input data. Each input image passes through a series of convolutional layers with filters, where kernals are used to extract features from an input image. A rectified linear activation unit (ReLU) is used to introduce non-linear operation. The pooled feature map matrix may be flatten and densed to a vector to be fed into a softmax layer, which converts the output into a probability distribution to classify an object with probability values between 0 and 1.

2 Goal

The goal of the project is to develop, train and test two CNNs as Deep Learning classifiers with variable structure of layers or configuration on the dataset from the paper by Spanhol, Fabio A. et al[1]. The classification will be evaluated and based on that a better structured classifier will be identified.

3 Data and Preprocessing

A total of 1,824 microscopic images from the paper by Spanhol, Fabio A. et al[1], downloaded and stored on Google Drive, were used as input data for this project. These images were obtained from 82 patients. They consist of 400x magnification of breast tumor tissue, both benign and malignant type. This dataset was imported and mounted in Google Colab using the drive class of the google.colab package. An authorization code is required to mount the drive. The dataset was manually split into a training set and a test set. 1456 images belonged to the training set and 364 images to the test set. The Os package was used to define the test and train directories and to obtain the path of the current working directory. Furthermore, Tensorflow's Image DataGenerator class was used to implement image rescaling, random zoom, rotation, and horizontal and vertical flips for the training set. The test set was only rescaled.

4 Methods

First, all the necessary packages were imported. These included google colab, tensorflow, keras, numpy, pandas, sklearn, time and os. The directory containing the test and train data was mounted using the drive class of google.colab . Then the dataset containing the training and test images was read from the directory using flow. After successfully importing the data, 2 Sequential models were implemented (see Figures 1).Add function was used to add layers to our model. The model consisted of a first layer consisting of Conv2D layers with a filter number of 128 (model 1) and 32 (model 2), and kernal size of (3,3) corresponding to a filter matrix of 3x3. We used 'Relu' - Rectified Linear Activation as the activation layer. Then a maxPooling2D layer was added to select the maximum value and downsample the input representation by (2,2). Then a Dropout layer was implemented to perform regularization by model 1 and was left out by model 2. These 4 layers were then implemented two more times in sequence by model 1 and by model 2 with different number of filters (with 64 and 128). This is followed by the flatten layer, which serves as a connection between the convolutional layer and the dense layer. Furthermore, the dense layer is used for our output layer and activated with the relu activation layer. Another time, dense layer was implemented but activated with softmax activation layer. Next, the model was compiled using model.compile(). Compile takes three parameters: optimizer, loss and metrics. For optimizer, loss and metrics, we used adam, categorical crossentropy and accuracy respectively. Finally, we created a history variable and trained the model using the fit() function. During training, the values entered into the callbacks are recorded at the end of each epoch. To evaluate our model, we make predictions, compute a classification report, confusion matrix and a roc curve with an auc value.

5 Results

We have developed two deep learning CNN models capable of classifying images of breast cancer histopathology with a magnification factor of 400x into the classes "malignant" and "benign". Our trained models, which also include the architectural structures (see figure 1), show the following plots of accuracy/loss, validation accuracy/loss (figure 2).

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 128)	3584
activation (Activation)	(None, 222, 222, 128)	0
max_pooling2d (MaxPooling2D)	(None, 111, 111, 128)	0
dropout (Dropout)	(None, 111, 111, 128)	0
conv2d_1 (Conv2D)	(None, 109, 109, 128)	147584
activation_1 (Activation)	(None, 109, 109, 128)	0
max_pooling2d_1 (MaxPooling2	(None, 54, 54, 128)	0
dropout_1 (Dropout)	(None, 54, 54, 128)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	147584
activation_2 (Activation)	(None, 52, 52, 128)	0
max_pooling2d_2 (MaxPooling2	(None, 26, 26, 128)	0
dropout_2 (Dropout)	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense (Dense)	(None, 128)	11075712
activation_3 (Activation)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258
activation_4 (Activation)	(None, 2)	0

Total params: 11,374,722
Trainable params: 11,374,722
Non-trainable params: 0

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 222, 222, 32)	896
activation (Activation)	(None, 222, 222, 32)	0
max_pooling2d (MaxPooling2D)	(None, 111, 111, 32)	0
conv2d_1 (Conv2D)	(None, 109, 109, 64)	18496
activation_1 (Activation)	(None, 109, 109, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 54, 54, 64)	0
conv2d_2 (Conv2D)	(None, 52, 52, 128)	73856
activation_2 (Activation)	(None, 52, 52, 128)	0
max_pooling2d_2 (MaxPooling2	(None, 26, 26, 128)	0
flatten (Flatten)	(None, 86528)	0
dense (Dense)	(None, 128)	11075712
activation_3 (Activation)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258
activation_4 (Activation)	(None, 2)	0

Total params: 11,169,218
Trainable params: 11,169,218
Non-trainable params: 0

Figure 1 CNN Architecture of two developed classifiers (left: Model 1 ; Model 2)

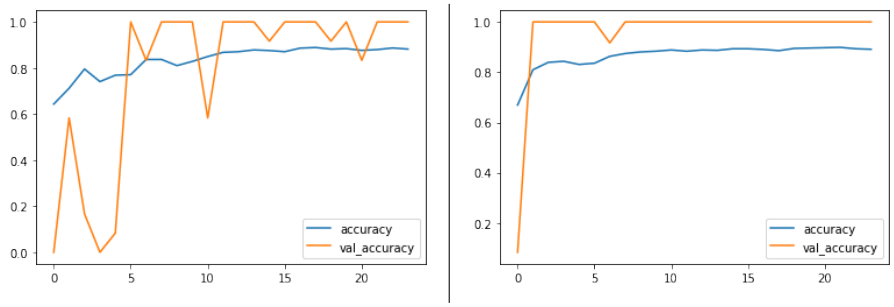

Figure 2 Accuracy Plot of two developed models (left: Model 1 ; right: Model 2)

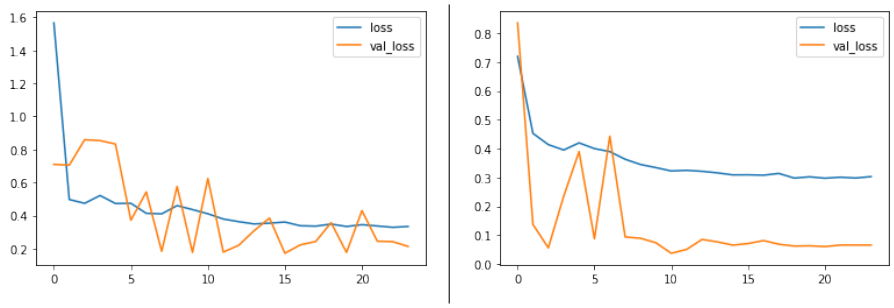

Figure 3 Loss Plot of two developed models (left: Model 1 ; right: Model 2)

It can be observed that both models have an accuracy value (y axis) of 80% and their curve (blue line) is nearly consistent and has no jumps during the epochs (x axis). The validation accuracy curve (orange line) increases at the beginning and remains almost at the same value of 100% for model 2. The validation accuracy curve is low at the beginning of training, increases at the beginning of training, has

large jumps and remains almost at the value of 100% for model 1. The loss curve (blue line) is high at the beginning and decreases for both models and remains nearly at value of 40%. The validation loss curve is zigzag especially for model 1 and for model 2 at the beginning and it has a decreasing trend. For model 1 it decreases nearly to value of 40-20 % and for model 2 to 10%. It can concluded that model 2 is slightly better than model 1 in terms of validation accuracy and loss values. Also this statement can be confirmed with estimated values for both models in the table 1.

	Model 1	Model 2
Accuracy	77%	81%
ValAccuracy	100%	100%
Loss	47%	45%
ValLoss	37%	14%

Table 1 CNN Classifier Accuracy and Loss Values of Model 1 and Model 2

In addition, Confusion Matrix (see figures 4), ROC curve with AUC value (see figures 5) and Classification Report (see figures 6) were created. The confusion matrix shows the 255 samples are correct identified vs. 109 missidentified for model 1 and 275 samples are correct and 85 are missidentified. The roc curve shows that model 1 with AUC value of 70% reliable and model 2 with AUC value of 74% slightly more reliable. Considering the classification report, it can be concluded that the

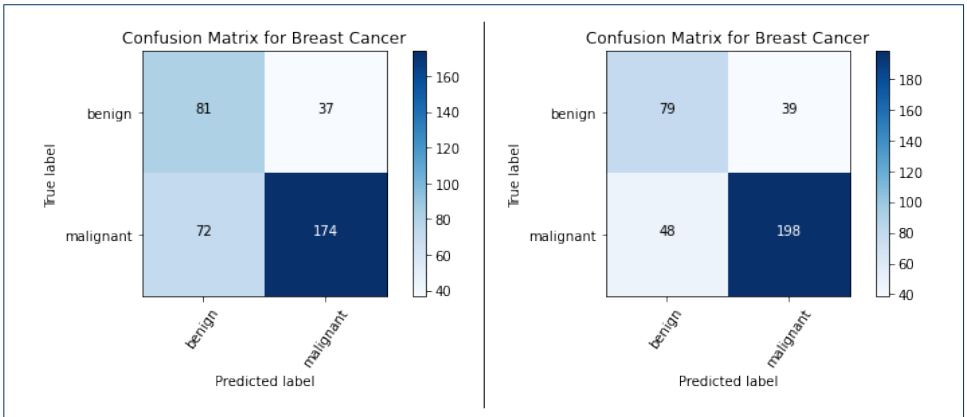

Figure 4 Confusion Matrix of two developed classifiers (left: Model 1 ; right: Model 2)

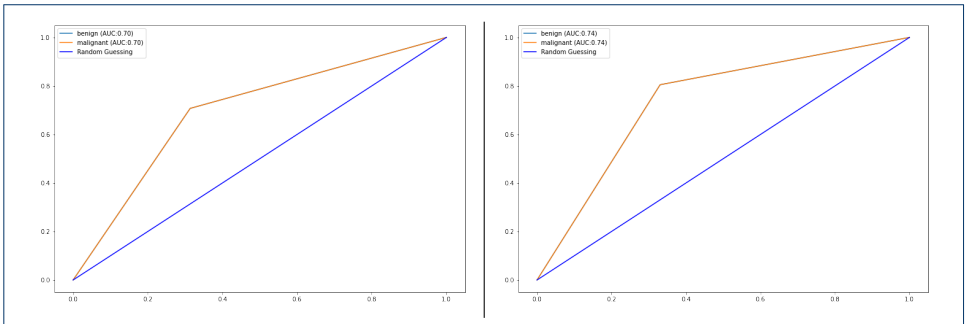

Figure 5 ROC Curve of two developed classifiers (left: Model 1 ; right: Model 2)

precision values for the benign class are low at almost 60% for both models and

better for the malignant class at almost 80% for both models. Also, the recall values are relatively good for the benign class at almost 70% for both models. The recall value for the malignant class is better for model 2 with 80% than 71% for model 1, so model 2 predicts the malignant class better than model 1. Although there

	precision	recall	f1-score	support
benign	0.53	0.69	0.60	118
malignant	0.82	0.71	0.76	246
accuracy			0.70	364
macro avg	0.68	0.70	0.68	364
weighted avg	0.73	0.70	0.71	364

	precision	recall	f1-score	support
benign	0.62	0.67	0.64	118
malignant	0.84	0.80	0.82	246
accuracy			0.76	364
macro avg	0.73	0.74	0.73	364
weighted avg	0.77	0.76	0.76	364

Figure 6 Classification Report of two developed classifiers (left: Model 1 ; right Model 2)

were missclassifications, both models were able to classify and distinguish between benign and malignant.

6 Discussion 1

Two models developed in this project could classify reliably with an auc value of 70% and 74% respectively. Misclassifications may be due to the high similarity of the cells of the benign tumor fibroadenoma with cells of some malignant tumors[1]

7 Discussion 2

In this project we learn the steps that lead to the development of a deep learning model for solving binary classification problems that are common in data science. We learned how to combine different layers like convolution, maxpooling, dense and other layers. We learned how to track the performance of the model training and how to select the model with a good performance in terms of accuracy and loss values and how to make predictions and evaluate it.

8 Appendix

Name	Work Description
Michael	Scientific Background, Goal Definition
Rohan	Data Preprocessing, Method Implementation
Natalja	Results Description, Discussion

Table 2 Task responsibilities

References

1. Spanhol, F.A., Oliveira, L.S., Petitjean, C., Heutte, L.: A dataset for breast cancer histopathological image classification. IEEE Transactions on Biomedical Engineering **63**(7), 1455–1462 (2016). doi:10.1109/TBME.2015.2496264