

RESEARCH

Deep Learning algorithms for image classifications of breast cancer histopathology images

Yunus Emre Kurnaz, Aakash Nepal, Cuong Gia Pham and Evelina Gudauskayte — Group 10

Full list of author information is available at the end of the article

Abstract

Deep learning is a subset of machine learning algorithms that use multiple layers of neural networks to process data and compute large amounts of data. Deep learning algorithms are based on the functions and behavior of the human brain. Just like the human brain uses millions of neurons to compute information, deep learning algorithms rely on neural networks to operate, with the help of so-called layers. A distinction is made between supervised and unsupervised learning. However, this project is about classifiers with supervised learning.

Following classifiers were used in this project: Fully connected neural networks (FNN), residual neural networks (RNN), and convolutional neural networks (CNN). In this project, the data ("BreakHis") is classified based on biomedical (microscopic) images, i.e. breast cancer histopathology images.

Therefore three different Deep Learning (DL) topologies are executed and an evaluation is performed using the appropriate techniques.

The main result of this project is that all three deep learning topologies behave differently and thus achieve different results/evaluations. This project has shown that Deep Learning is a very complex area of machine learning.

As in the previous project, it is not possible to use one classifier for all research questions. This means that one has to look critically at the data, because with more layers, the complexity of the model also increases, and more data is needed for the model to perform correctly. However, if the amount of data is less than the complexity of the neural network, a different approach must be taken into account. With the help of this project, we were able to program three different approaches of an image classification algorithm in google collab with tensorflow and keras, and both refreshed or relearned important terminologies and procedures for dealing critically with the field of deep learning. A total of approx. 1-2 days were required to conclude all of the results.

1 Scientific Background

Deep learning is a sub-field of machine learning in which computer models perform classification tasks directly from images, texts or acoustic data. A special feature is that deep learning models are able to learn independently. This is done by the systems repeatedly linking what they have learned with new content and thus re-learning. This is done with the help of so-called neural networks. They are compared to "natural neural networks", which are a network of neurons in the nervous system of a human, the so-called layers.

In other words; the input feature/data is passed into the "input layer". The resulting output, serves as input for the next layer, the so-called "hidden layer". This is done until the final output is given to the "output layer". Depending on the complexity of the research question, a number of input layers, hidden layers or output

layers can be added.

That's why this learning process is called deep learning, because there are so many computations going on between the input and output layers.

The importance of deep learning algorithms is characterized by the fact that they can achieve higher accuracy and are therefore used in many areas of daily life. Well-known examples are autonomous driving, weather forecasts or predictions of diseases in medical researches. This however, requires a lot of classified data and high computational resources (GPU).

Since large data sets cannot be used for this project, due to lack of computing resources to process these data, the problem was limited to the "BreaKHis" database. This database was created in collaboration with the P&D Laboratory - Pathological Anatomy and Cytopathology, Parana, Brazil, and contains approximately 7909 clinically representative microscopic images of breast tumors provided by 82 patients with different magnification factors (40×, 100×, 200×, and 400×). The database contains 2480 benign and 5429 malignant samples.

This data will be used to classify data based on biomedical (microscopic) images with the help of three different deep learning topologies. The goal of this project is to become familiar with the specialized subfield of machine learning and to try three different deep learning (DL) topologies and perform an evaluation using the appropriate techniques, where one of the three DL topologies must be a CNN.

2 Goal of the project

As mentioned in the previous chapter, the goal of this project is to become familiar with the complex topic of Deep Learning and to apply the knowledge learned from the lecture to a real-life example. The focus is not only on programming an algorithm to classify breast cancer data based on biomedical (microscopy) images using TensorFlow and Keras, but also on learning important basics and concepts that are mandatory for this topic in order to create a good predictor. Depending on the size and complexity of the data and the research question, different approaches may be needed to provide suitable results. It is also important to understand what a neural network is and how it works, since it is the basis for a deep learning algorithm.

3 Data and Preprocessing

The BreaKHis database contains microscopic biopsy images of benign and malignant breast cancer tumors. These images were collected as part of a clinical study from January 2014 to December 2014. The samples were stained by with hematoxylin and eosin (HE), collected mainly by surgical (open) biopsies, prepared for histologic examination, and processed by P&D laboratory pathologists.

Preservation of the original tissue structure was important for observation of samples under a light microscope. Pathologists identify tumor areas on each slide by visually analyzing tissue sections under a microscope. A definitive diagnosis is made in each case by an experienced pathologist and confirmed by additional investigations such as immunohistochemical analysis.

Digital images of breast tissue sections were acquired using a microscope in combination with a high-resolution camera. To date, the database contains 7909 images classified into benign and malignant tumors. The dataset currently contains four

types of benign breast tumors: adenosis (A), fibroadenoma (F), phyllodes tumor (PT), and tubular adenoma (TA), and four malignancies: ductal carcinoma (DC), lobular carcinoma (LC), mucinous carcinoma (MC), and papillary carcinoma (PC). The resolutions of the biopsy images are available in the following resolutions: $40\times$, $100\times$, $200\times$ and $400\times$. However, three different models were programmed, where each model scaled the images as follows: 64×64 (CNN), 224×224 (RNN) and 64×74 (FNN). Thus, the only preprocessing step was suitably scaling the images for further analysis.

4 Methods

The following paragraph explains the three algorithms that were modeled. The focus is therefore on a briefly definition of the used classifier and additionally a description of how the classifier was handled during programming.

4.1 Fully Connected Neural Network

4.1.1 Definition

A Fully Connected Neural Network is a neural network that consists of a series of fully connected layers. Each fully connected layer consists of nodes that can also be called a Perceptron. Perceptron consists of weights and an activation layer. The input of each node is multiplied by weights and the result of multiplication is given to the activation function. Two other techniques that were used to train fully connected neural network are batch normalization and dropout. Batch-normalization layers normalize the contributions to a layer for every mini-batch. This has the impact of settling the learning process and drastically decreasing the number of training epochs required to train deep neural networks. Dropout layers randomly set input units to 0 with a frequency of given rate at each step during training time, which helps prevent overfitting.

4.1.2 Usage

To train fully connected neural network, classes "benign" and "malignant" were used, only $400\times$ images from all sub classes were taken. Architecture of network can be described as three blocks with linear layer, batch normalization and dropout and sigmoid function after these three blocks. Exact parameters can be found in the source code. Pytorch and torchvision libraries were used to implement such solution. All images were resized to the size 46×79 and then have been flattened. The training was performed for 10 epochs using Adam optimizer and BCE with logits loss function. Data was split such that 80 percents of data is in used for training and 20 percents used for validation purposes. During training values of f1 metric, sensitivity and specificity was monitored.

4.2 Residual Neural Network

4.2.1 Definition

A Residual Neural Network (ResNet) is an artificial neural network (ANN) based on a collection of connected units, the artificial neurons. A pre-trained version of the network, trained with more than one million images, can be loaded from the ImageNet database. This pre-trained network can be used for tasks such as object detection, image classification and object localisation. In a normal deep model, adding more layers leads to a correspondingly higher training error. To solve this problem, typical ResNet models are implemented in such a way that some layers are skipped by taking shortcuts, which simplifies the network and reduces the error. A residual neural network was used to win the ImageNet2015 competition and became the most cited neural network of the 21st century. ResNet-50 is a convolutional neural network with a depth of 50 layers, i.e. 48 convolutional layers with a MaxPool and an AveragePool layer.

4.2.2 Usage

First, the two classes "benign" and "malignant" were preprocessed based on all images (example: only 400x images from all sub classes were taken). The given images have higher image dimensions, so they were reformatted to a dimension of 240x240 pixels, which means that they are each 3x224x224 input data for the network. Using `tensorflow.keras.utils.image_dataset_from_directory()`, the image data was split into 80% for training and 20% for validation. A batch size of 32 was taken. The Matplotlib library was used to visualise some images in the training data, and `image.shape()` was used to check the shape of the image. After that, the ResNet50 model was imported and loaded. The precalculated weights from the Imagenet data were used so that the weights did not have to be recalculated to save computing time and memory. After importing the pretrained model, a fully linked and an outer layer with 2 output neurons were imported in which a softmax activation function was used as there were only two classes in the data. The model was then compiled with Adam optimizer (with default learning rate of 0.001) algorithm and ran for 10 epochs. The model then was evaluated on the basis of validation and training accuracy for each epoch in a plot using the Matplotlib library. Prediction was also performed by taking a random image from the validation data. To be able to determine, if the results are improving when the data is augmented, before running the model, the augmentation step was performed additionally and the model was reran with the same parameters as before. For this augmentation step the training set was applied transformations such as image flipping, rotation and zoom.

4.3 Convolutional Neural Network

4.3.1 Definition

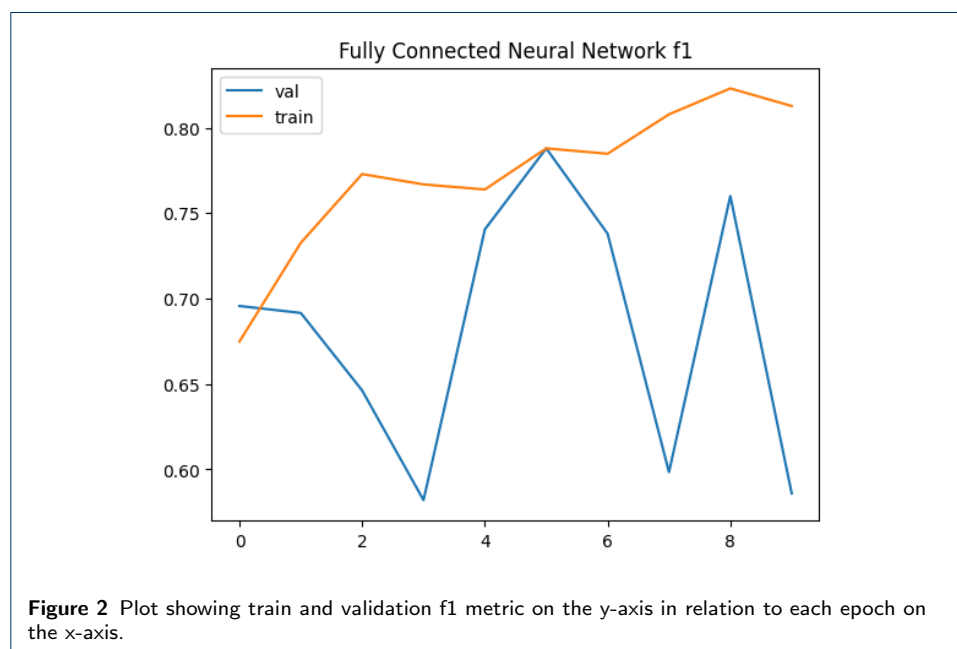
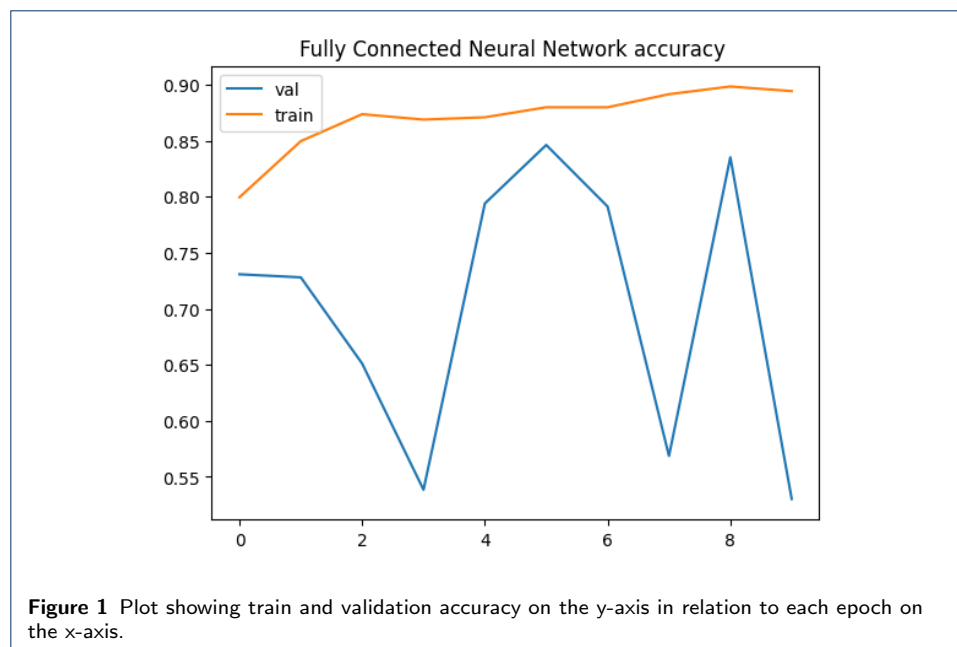
A Convolutional Neural Network (CNN) is a Deep Learning algorithm that can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and be able to differentiate one from the other. The pre-processing steps required in a CNN model are less, compared to other classification algorithms. Primitive methods require hand-engineered filters. However, with enough training, a CNN is able to learn these filters/characteristics. The architecture of a CNN is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the visual cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. A collection of such fields overlaps to cover the entire visual area.

4.3.2 Usage

First, a CSV file that contains paths to images and each image's label is generated by the script `generate_dataset.py`, to gather the chosen images. After that, the CSV file is used to split to the train and validation set by using the built-in function of TensorFlow `ImageDataGenerator`. Here, the train set accounts for 75%, and the validation set accounts for 25%. Besides that, the `ImageDataGenerator` function normalizes the generated matrix by dividing each pixel by 255. With the built-in function `flow_from_dataframe` of TensorFlow, the batch size is set to 32, and each image is cropped to 64x64 to improve the training process. The train set is then ready to fit in the model. Our CNN model is built based on the architecture that every two convolution layers come with one MaxPooling or one AveragePooling layer. The model is trained with ten epochs. At each epoch, the model shows the loss value, the AUC score and the accuracy of train set and validation set.

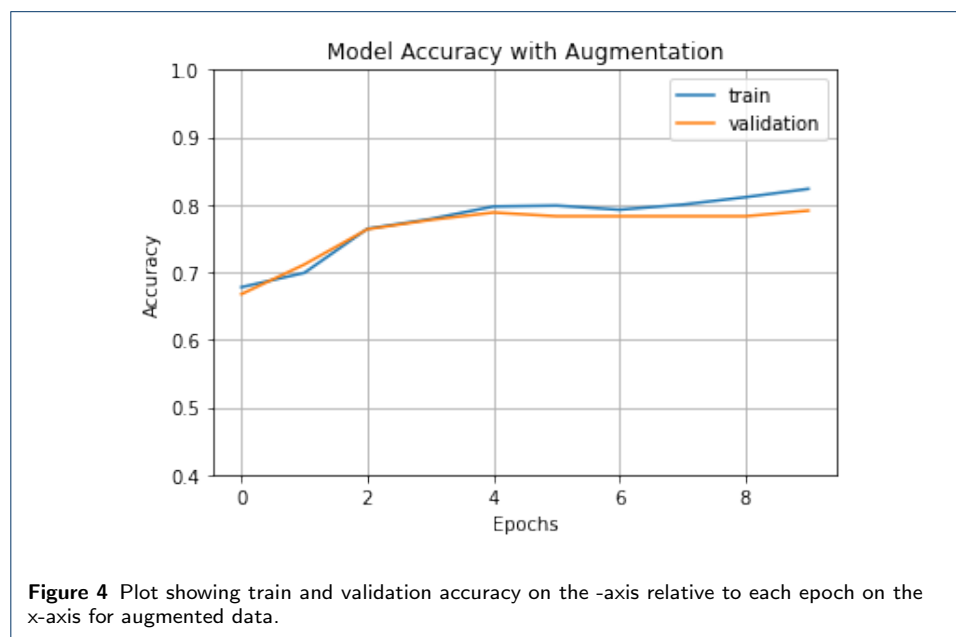
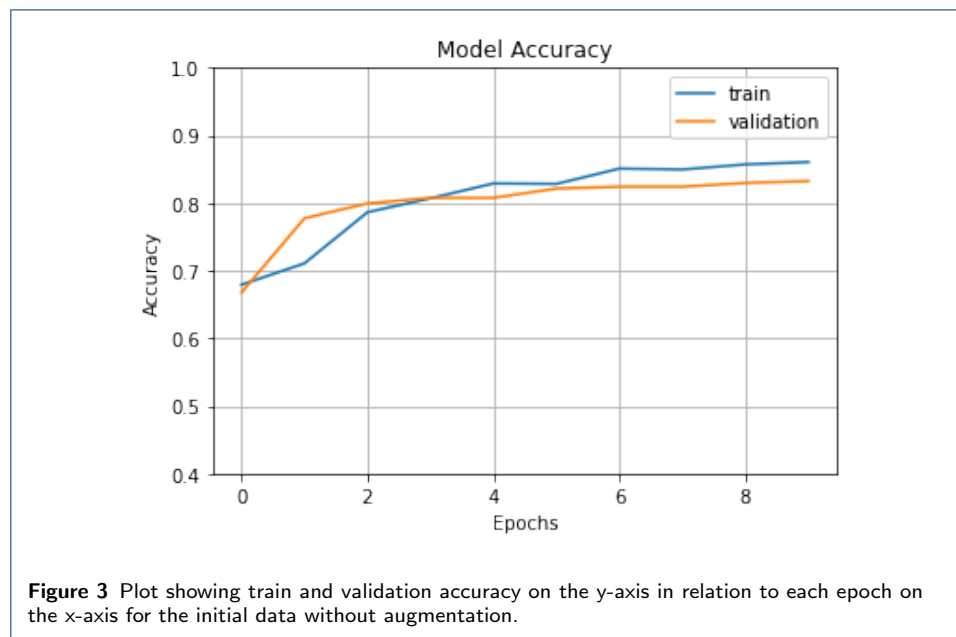
5 Results

5.1 Fully Connected Neural Network



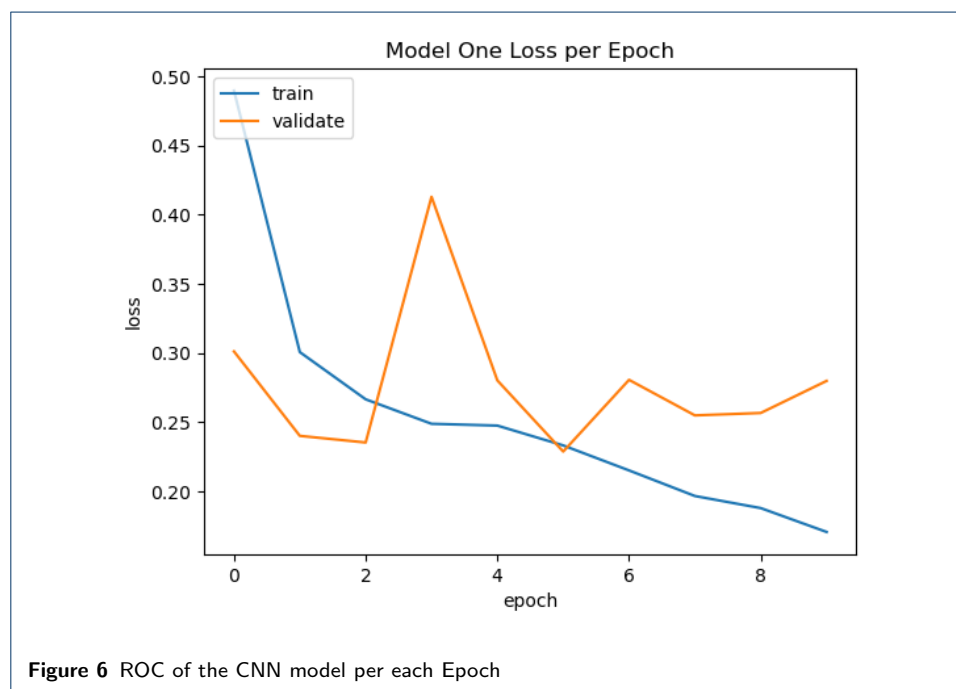
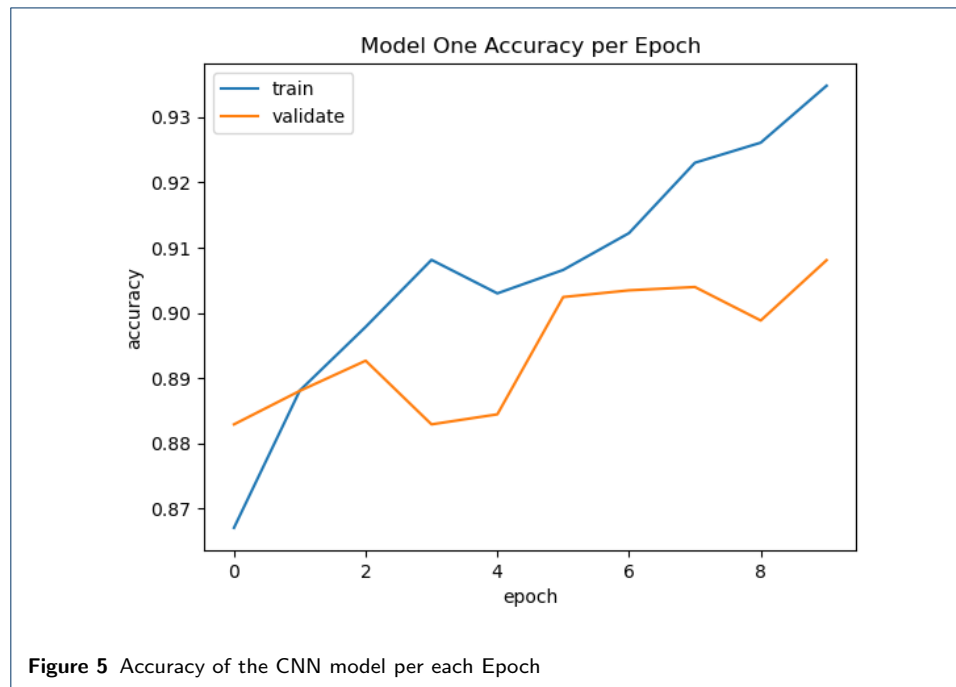
As can be seen in figure 2, validation metrics are the best at epoch 5, where f1 hits value 0.79. Overall metrics in figure 1 and in figure 2 are not very high, as when flattening applied, an image is represented as a vector and a lot of dimensional information is lost. Moreover, it is clearly to see, how the model overfits. Therefore the fully connected neural network is not the best choice for given data as it performs worse than other types of neural networks.

5.2 Residual Neural Network



In figure 3, the observed accuracy per epoch for the training set goes slightly higher than the accuracy of the model for the validation set, which means that the model seems to be somewhat overfitted. The highest validation accuracy reached is approximately 83%. But after augmentation, which can be seen in figure 4, both training and validation accuracy increase at about the very similar rate close to each other and reaching the highest validation accuracy at epochs 10 to be approximately 80%.

5.3 Convolution Neural Network



In figure 5, the accuracy of train and validation sets are fluctuating. It shows that some portion of the data set is classified randomly. Besides, in figure 6, the loss line of validation set tends to go up, which means that the model seems to be overfitted.

6 Discussion

As can be seen from the previous chapter, all of the models used, tend to overfit, which is also evident in the figures 1-6.

Overfitting is not always easy to avoid, since in most cases it is necessary to take into account what kind of data are available and what potential impact, methods to avoid overfitting, may have on the research question.

Probably the simplest approach is to train a classifier with more data. Since this cannot always be guaranteed and there is also the risk of adding additional "noisy data" into the classifier, this approach cannot always guarantee that overfitting can be avoided. There exist numerous methods: "early stopping", "regularization", "ensembling", "feature removal" or "augmentation", which can be used. However, none of them is always the solution to solve every problem. Each of them brings advantages and disadvantages, that must be weighted in order to avoid overfitting while obtaining a suitable result in the same time. This project, is an appropriate example for a data scientist, representing a possible real-world scenario where it is necessary not only to deal more critically with the data, but also to be aware of the consequences of overfitted and underfitted data before adopting a final result.

7 Appendix

Approximately the same amount of work was assigned among all group members. Yunus Emre Kurnaz wrote most of the report and also worked with Cuong Gia Pham on a convolutional neural network (CNN).

Aakash Nepal programmed a deep learning algorithm based on a residual neural network (RNN).

Evelina Gudauskayte programmed a fully connected neural network (FNN) for the above problem.