Assignment 3: Convolutional neural networks for classification

Project BIA group 1

By: J.N.A. Bergmans, M. Boone, J.N.A. van Mook, J.F. Peeters

Exercise 1:

The PatchCAMELYON challenge on Kaggle uses the area under the ROC curve as an evaluation measure. Describe the concept of ROC curve analysis and the area under the ROC curve (AUC) as an evaluation measure. Then, using methods from the sklearn Python toolbox (the required functions are alerady imported), perform ROC curve analysis and computation of the AUC for your model.

Note that you will have to do this analysis on the validation set (since you do not have access to the ground truth for the test set).

Answer:

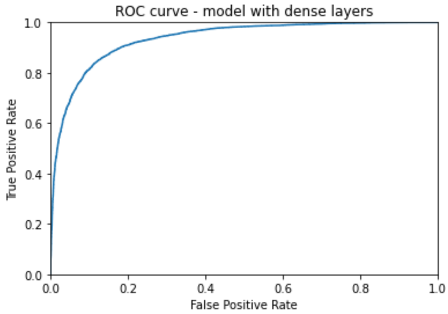
When 1-specificity (i.e. False Positive Rate, FPR) on the x-axis is plotted against sensitivity (True Positive Rate, TPR) on the y-axis, the plot is called the Receiver Operator Characteristic (ROC) curve (Hajian-Tilaki, 2013)*. To compare different classifiers, i.e. the same classifier with different learning parameters or completely different classifiers, the ROC curve can be useful* (Marsland, 2014)*. A single run of one classifier gives a single point on the ROC plot* (Marsland, 2014).

In medical context for instance, the basic principle of the ROC curve is to quantify how accurately medical diagnostic tests can discriminate between two patient states e.g. diseased or healthy (Hajian-Tilaki, 2013)*. By altering the decision threshold we will get different fractions of TP and FP* (Metz, 1978)*. For all these decision thresholds then the fractions are plotted in the ROC curve. A classifier is deemed to be 'perfect' when it is located at the point (0,1) on the ROC curve, i.e. when the classifier has a TPR of 1 and a FPR of 0 for a specific decision threshold. Moreover, when the ROC curve corresponds to the 45 degree line (y = x), the diagnostic test that belongs to this curve is as good as random guessing (i.e. chance level)* (Hajian-Tilaki, 2013)*. In other words, we have a test which yields positive or negative results unrelated to the true disease status* (Hajian-Tilaki, 2013)*. Hence, the closer to the top-left-hand corner the result of a classifier is, the better it has performed* (Marsland, 2014)*. Thus, to compare different classifiers, one can use the Area Under the Curve (AUC) to evaluate and compare model performance. Since the closer the classifier gets to the 'ideal' point of (0,1), the larger the AUC and the better the classifier performs* (Marsland, 2014).

The code for the ROC curves is implemented in the files cnn.py and assignment3_fullyconv.py that can be found in the zip file.

ROC curve for the model in cnn.py :

AUC score: 0.8903727499999999



The top left corner of the graph can be considered close to the ideal point of (0,1). Moreover, the area under the curve is close to 1 which is an indication that the model performs well on the validation set.

Exercise 2

It is possible to construct a neural network model that is equivalent to the model above, however, only using convolutional layers (i.e. without using any fully connected or "dense" layers). Construct and train such a model.

What would be the advantage of only using convolutional layers?

Answer:

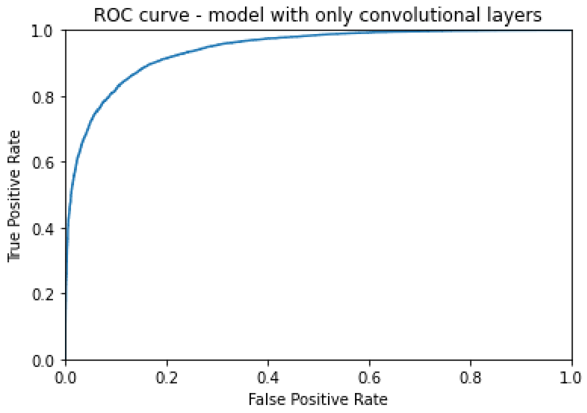
Yes, it is possible to construct a neural network model equivalent to the previous model by using only convolutional layers. The difference between fully-connected and convolutional layers is that the neurons in a convolutional layer are connected only to a local region in the input and the sharing of weights by the neurons (Thakur, 2021)*. This sharing is done by convolving the input image / array with a kernel. Hence, in order to convert a fully connected layer to a convolutional layer with equivalent output, one has to pay attention to the size of the kernel used in the convolutional layer. That is, the kernel size has to be the same as the size of the input feature array and the filter size must be equal to the input volume* (Thakur, 2021) (Raschka, 2022)*. Consequently, the output will be equal to the output we received from a fully-connected layer* (Thakur, 2021)*. In our particular case, this means that we do not flatten the array until after the convolutional layers since the kernels corresponding to the convolutional layers remain two dimensional. The two dense layers with respectively 64 and 1 neurons will be replaced by 2D convolutional layers with kernel size equal to the size of the input that the previous layer generates. From the model summary we find that the second max pooling layer gives an output shape of (None, 6, 6, 64) so the kernel of the first convolutional layer needs to be (6,6) with 64 filters. For the last convolutional layer we need only one output neuron, since we only want one final output from our model (i.e. the probability of the image belonging to class 1, binary problem). Hence, the last convolutional layer will have a kernel size of (1,1) with 1 filter (filter = 1 means the number of output filters in the convolution is equal to 1)* (Keras Team, n.d.)*. To receive one probability value, we flatten the array after all the layers. To check if the neural network consisting of only convolutional layer is equivalent to the previous fully-connected neural network, we can examine if the amount of total parameters is the same. This is the case since for both the fully-connected as well as the fully convolutional neural network, the amount of total parameters equals 166,977. Hence, we have created a network consisting of only convolutional layers that is equivalent to the first mentioned neural network.*

The architecture of a convolutional neural network, in comparison to an ordinary feed forward neural network, allows for better fitting to the image dataset caused by a reduction in the number of parameters involved and the reusability of weights by convolution with a kernel (Saha, 2021)*. Using only convolutional layers drastically reduces the amount of weights that needs to be calculated and updated during back propagation. Usually, an advantage then would be that the computation time is decreased and overall the convolutional neural network model would be less computationally expensive. However, in our specific case, the convolutional neural network is equivalent to the fully-connected version since it has the same amount of parameters. Hence this advantage does not apply here. A benefit of using only convolutional layers that does apply here, is the ability to automatically identify relevant features without the need for human supervision* (Alzubaidi et al., 2021)*. CNN's are designed to automatically and adaptively detect spatial hierarchies of features, which makes them especially suitable for image recognition* (Yamashita et al., 2018)*. In conclusion, the higher efficiency of CNN's due to the usage of kernels to extract features anywhere in the image, gives convolutional* *layers a substantial advantage over fully connected layers regarding image processing* (Yamashita et al., 2018)(Ganesh, 2021).

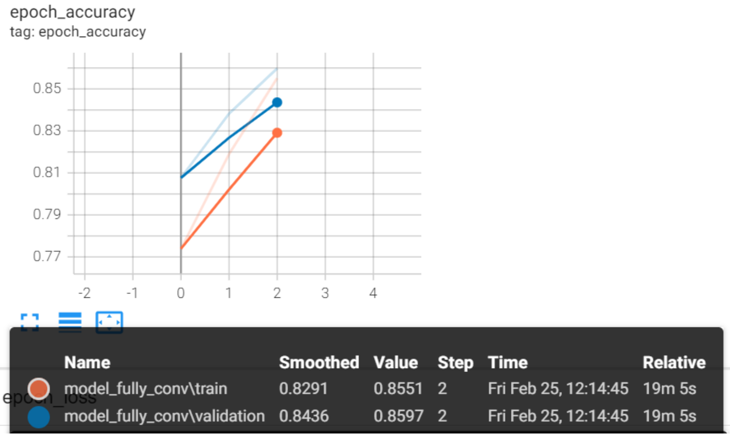
The code for the fully convolutional neural network can be found in the file assignment3_fullyconv.py together with the computation of the roc curve and the area under the curve for the fully convolutional neural network

Results

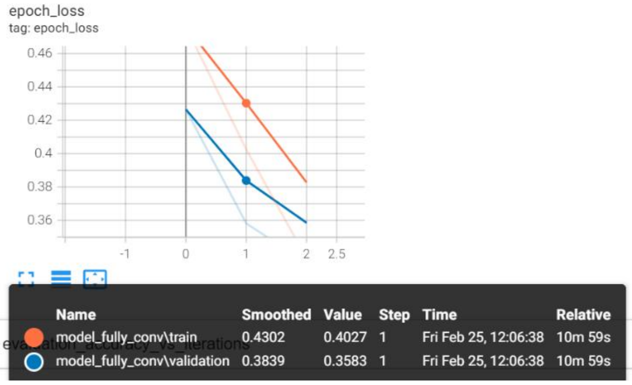
ROC curve : fully convolutional neural network (based on performance on validation set)



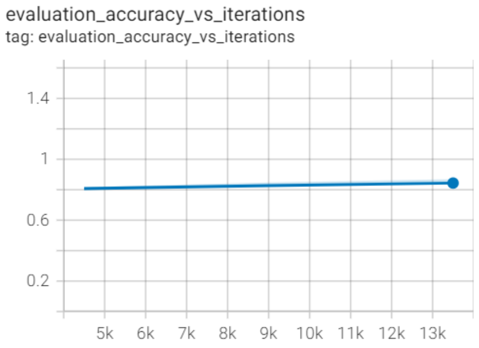
Loss curves – fully convolutional (training and validation set)

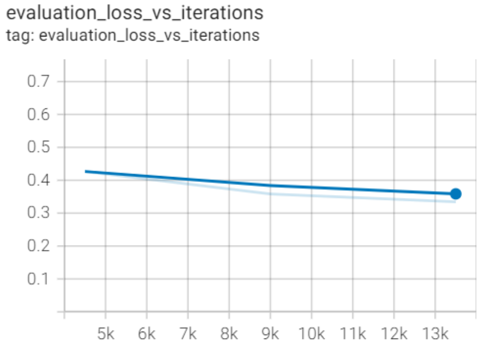


Name	Smoothed	Value	Step	Time	Relative
model_fully_conv\train	0.8291	0.8551	2	Fri Feb 25, 12:14:45	19m 5s
model_fully_conv\validation	0.8436	0.8597	2	Fri Feb 25, 12:14:45	19m 5s

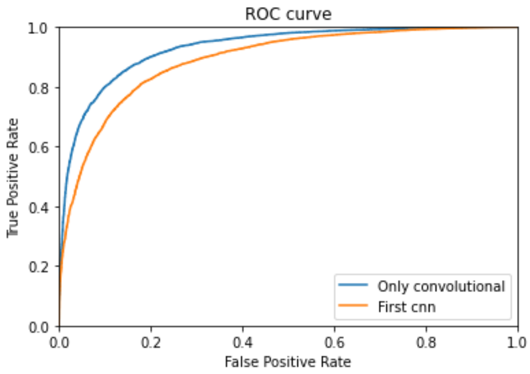


Name	Smoothed	Value	Step	Time	Relative
model_fully_conv\train	0.4302	0.4027	1	Fri Feb 25, 12:06:38	10m 59s
model_fully_conv\validation	0.3839	0.3583	1	Fri Feb 25, 12:06:38	10m 59s





With a value of AUC = 0.931 for the validation set, the fully convolutional neural network was more accurate at discrimination than the first CNN, which got a mark of AUC = 0.890. Comparison of the ROC curves suggests an identical inference.



Exercise 3

Use the kaggle_submission.py file to prepare and submit results for evaluation to Kaggle. What is the result of the evaluation on the test set? How does it compare to the evaluation that you performed on the validation set?

Answer

The private score on Kaggle for the fully convolutional model predictions on the test set equals 0.8985. This score is slightly lower than the AUC score of the fully convolutional neural network on the validation set (which had an AUC score of approximately 0.931)

Bibliography

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*(1), 53. https://doi.org/10.1186/s40537-021-00444-8

Ganesh, P. (2021). *Types of Convolution Kernels : Simplified - Towards Data Science*. https://towardsdatascience.com/types-of-convolution-kernels-simplified-f040cb307c37

Hajian-Tilaki, K. (2013). Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation. *Caspian Journal of Internal Medicine*, *4*(2), 627–635. https://pubmed.ncbi.nlm.nih.gov/24009950

Keras Team. (n.d.). *Conv2D layer - Keras documentation*. https://keras.io/api/layers/convolution_layers/convolution2d/

Marsland, S. (2014). *Machine Learning: An Algorithmic Perspective, Second Edition (Chapman & Hall/CRC Machine Learning & Pattern Recognition)* (2nd ed.). Chapman and Hall/CRC.

Metz, C. E. (1978). Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, *8*(4), 283–298. https://doi.org/https://doi.org/10.1016/S0001-2998(78)80014-2

Raschka, S. (2022). *Can Fully Connected Layers be Replaced by Convolutional Layers?* https://sebastianraschka.com/faq/docs/fc-to-conv.html#:~:text=Yes%2C you can replace a,exact same behavior or outputs.

Saha, S. (2021). *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Thakur, A. (2021). *Converting FC Layers to Conv Layers - Ayush Thakur*. https://mein2work.medium.com/converting-fc-layers-to-conv-layers-8a43880a44ed

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, *9*(4), 611–629. https://doi.org/10.1007/s13244-018-0639-9