Documentation of the project Statistical Methods for Machine Learning

MARCO CAVALLARI
University of Milan
Marco.cavallari@studenti.unimi.it

BASHAR LOUIS
University of Milan
bashar.louis@studenti.unimi.it

July 17, 2022

ˆThis project aims at the binary classification of a big data set of images that contain cats and dogs by building a convolutional neural network and studying the impact of network architectures and training parameters on the final predictive performance. Furthermore, computing the risk estimates by applying the zero-one loss in the cross-validation. This research paper will study in-deep both the theoretical part of CNN and then describe the data set used and its processing for this project .The last part will cover the comparison between different implementations applied .

I. CONVOLUTIONAL NEURAL NETWORK

CNN is a deep learning neural network designed for processing structured arrays of data such as images. Convolutional neural networks are widely used in computer vision and have become the state of the art for many visual applications such as image classification, and have also found success in natural language processing for text classification. The patterns in the input picture, such as lines, gradients, circles, or even eyes and faces, are extremely well recognized by convolutional neural networks. Convolutional neural networks are extremely effective for computer vision because of this quality. Convolutional neural networks do not require any preparation and may function immediately on a raw picture, in contrast to older computer vision methods.

A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer. The convolutional layer performs a scalar product between two matrices (a.b) , where one matrix is the set of learnable parameters otherwise known as a kernel, and the other matrix is the restricted portion of the receptive field. This produces a two-dimensional representation of the image known as an activation map that gives the response of the kernel at each spatial position of the image. Mathematically, accepts an input volume of size $W_{input} H_{input} D_{input}$

with the four parameters , The number of filters K ,The receptive field size F ,The stride S and The amount of zero-padding P. The output of this layer will be $W_{output}\ H_{output}D_{output}$, where:

$$W_{output} = ((W_{input}F + 2P)/S) + 1$$

$$H_{output} = ((H_{input}F + 2P)/S) + 1$$

$$D_{output} = K$$

By calculating **an aggregate statistic** from the surrounding outputs, the pooling layer substitutes for the network's output at certain points. This aids in shrinking the representation's spatial size, which lowers the amount of computation and weights needed.

As in a conventional Fully-connected NN, all of the neurons in this layer are fully connected to all of the neurons in the layer before and after.The representation between the input and the output is mapped with the aid of the FC layer.

Mathematically speaking , for a given image and filter we have:

$$conv(I,K)_{x,y} = \sum_{i=1}^{n_H}\sum_{j=1}^{n_W}\sum_{k=1}^{n_C} K_{i,j,k} I_{x+i-1,y+j-1,k}$$

II. Pooling Layers

As we previously mentioned ,In general, there are three types of layer in a convolutional neural network, which are convolution layer (CONV), pooling layer (POOL) and fully connected layer (FC). Typically, several convolution layers are followed by a pooling layer and a few fully connected layers are at the end of the convolutional network.

. It was proven before that the convolutional product using the vertical-edge filter, the pixels on the corner of the image (2D matrix) are less used than the pixels in the middle of the picture which means that the information from the edges is thrown away. To solve this problem, we often add padding around the image in order to take the pixels on the edges into account. In convention, we padde with zeros and denote with p the padding parameter which represents the number of elements added on each of the four sides of the image. Therefore ,the function of pooling layer is to reduce the spatial size of the representation so as to reduce the amount of parameters and computation in the network and it operates on each feature map (channels) independently

There are two types of pooling layers, which are max pooling and average pooling. However, max pooling is the one that is commonly used.

There are two purpose of using the pooling layer :

- Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network.
- The pooling layer summarises the features present in a region of the feature map generated by a convolution layer.

pooling is especially helpful when you have an image classification task where you just need to detect the presence of a certain object

in an image, but you don't care where exactly it is located. The fact that pooling filters use a larger stride than convolutional filters and result in smaller outputs also supports the efficiency of the network and leads to faster training. In other words, location in-variance can greatly improve the statistical efficiency of the network.

III. Fully Connected Layer

The Fully Connected (FC) layer consists of the weights and biases along with the neurons and is used to connect the neurons between two different layers. These layers are usually placed before the output layer and form the last few layers of a CNN Architecture.

In this, the input image from the previous layers are flattened and fed to the FC layer. The flattened vector then undergoes few more FC layers where the mathematical functions operations usually take place. In this stage, the classification process begins to take place. simply , it could be defined as a fully connected layer is a function from m to n . Each output dimension depends on each input dimension

IV. Activation Functions

Activation functions are an extremely important feature of the artificial neural networks. They basically decide whether a neuron should be activated or not. Whether the information that the neuron is receiving is relevant for the given information or should it be ignored. The activation function is the non-linear transformation that we do over the input signal. This transformed output is then sent to the next layer of neurons as input.

- Linearity Layers:

  If we don't apply a activation function then the output signal would be simple linear function. A linear function is easy to solve but they are limited in their complexity and have less power t learn complex functional mappings from data. Also without activation function our neural networks would not be able to learn and other kinds of data such as images, videos, audio, speech

- Non-Linearity Layers:

  Non-linearity functions are frequently included right after the convolutional layer to add non-linearity to the activation map because convolution is a linear operation and pictures are everything but linear (This is the situation of our project).

  There are several types of non-linear operations, the popular ones being:

  – **Sigmoid:** The sigmoid non-linearity has the mathematical form

  $$\sigma(k) = 1 \div \left[1 + e^{K}\right]. \qquad (1)$$

  It takes a real-valued number and squashes it into a range between 0 and 1. However, a very undesirable property of sigmoid is that when the activation is at either tail, the gradient becomes almost zero. This is a smooth function and is continuously differentiable. The biggest advantage that it has over

step and linear function is that it is non-linear

- **Tanh:** Like sigmoid, the activation saturates, but — unlike the sigmoid neurons — its output is zero centered.

$$tanh(x) = 2sigmoid(2x) - 1 \qquad (2)$$

It basically solves our problem of the values all being of the same sign and as you can see It is continuous and differentiable at all points.

- **Relu:** The Rectified Linear Unit computes the function

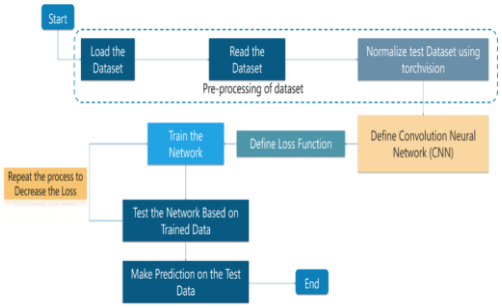
$$f(K) = max(0, K) \qquad (3)$$

In other words, the activation is simply threshold at zero.ReLU is more reliable and accelerates the convergence by six times comparing with the previous ones .It is clear that the ReLU function is non linear, which means we can easily back propagate the errors and have multiple layers of neurons being activated by the Re-LU function. The main advantage of using the ReLU function over other activation functions is that it does not activate all the neurons at the same time

- **Softmax function:** The softmax function is also a type of sigmoid function but is handy when we are trying to handle classification problems.The sigmoid function as we saw earlier was able to handle just two classes while softmax function could be used for multiple-classification issue.The softmax function would squeeze the outputs for each class between 0 and 1 and would also divide by the sum of the outputs.

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K.$$

The project could be abbreviated by the following image :



V. K-Fold Cross Validation

The original sample is randomly divided into k sub-samples of equal size for k-fold cross-validation. Out of the k sub-samples, k-1 sub-samples are utilized as training data, while the remaining k sub-sample is used as validation data for testing the model. The cross-validation procedure is then carried out k times, using the validation data from each of the k sub samples exactly once each time. After that, an estimation may be created by averaging the k outcomes.

in our project , we used stratified k-fold cross-validation, the partitions are selected so that the mean response value is approximately equal in all the partitions. In

the case of binary classification, this means that each partition contains roughly the same proportions of the two types of class labels.

Cross-validation implemented using stratified sampling ensures that the proportion of the feature of interest is the same across the original data, training set and the test set. This ensures that no value is over/under-represented in the training and test sets, which gives a more accurate estimate of performance/error.

so if we care about preserving the class ratio of our target and We have relatively fewer training examples so the stratified sampling cross-validation will be the best choice.

VI. Loss function

The Loss Function tells us how badly our machine performed and what's the distance between the predictions and the actual values. There are many different Loss Functions.

In general , Loss functions are used in optimization problems with the goal of minimizing the loss. Loss functions are used in regression when finding a line of best fit by minimizing the overall loss of all the points with the prediction from the line. Loss functions are used while training perceptrons and neural networks by influencing how their weights are updated. The larger the loss is, the larger the update. By minimizing the loss, the model's accuracy is maximized. However, the trade off between size of update and minimal loss must be evaluated in these machine learning applications.

VII. Binary Cross entropy

As our model is a kind of binary classification , we have used binary cross-entropy / log loss as your loss function. This value could be calculated by the following equation :

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

- y : the label which is 0 or 1 .
- p(y) : The predicted probability of the point to be 0 or 1

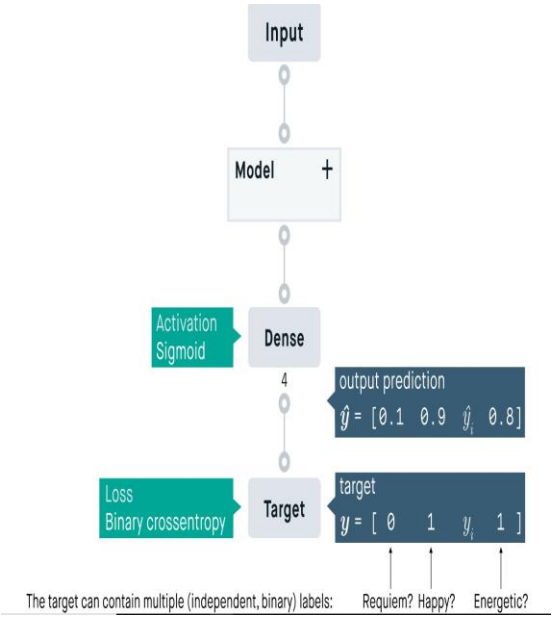
Since we're trying to compute a loss, we need to penalize bad predictions, so if the probability associated with the true class is 1.0, we need its loss to be zero. Conversely, if that probability is low, say, 0.01, we need its loss to be HUGE! The cross-entropy will have a BIGGER value than the entropy computed on the true distribution.

$$H_p(q) - H(q) >= 0$$

How to use binary cross entropy :

- **Activation functions:** Sigmoid is the only activation function compatible with the binary cross entropy loss function. You must use it on the last block before the target block.

This graph shows how this function is calculated :



VIII. Jump to the Code

i. pre-processing Data

our huge data set is mounted to google drive to be able to run our code in Co-lab and Google Colab is an excellent tool for deep learning tasks and it provides RAM of 12 GB with a maximum extension of 25 GB and a disk space of 358.27 GB.//[1ex] In order to find the corrupted images in the dataset ,Then we run a script to eliminate automatically the corrupted images from the folder by using the tensorFlow function **tf.io.decode-image()** to detects whether an image is a BMP, GIF, JPEG, or PNG, and performs the appropriate operation to convert the input bytes string into a Tensor of type dtype

The images of cats and dogs are labeled with the appropriate labels to be then split into training, validation, and testing sets with a ratio of 0.72 for training , 0.08 for validation and 0.2 for testing.

We applied a shuffle algorithm to serve the purpose of reducing variance and making sure that models remain general (preventing any bias during the training)and over-fit less and that is clear in the following defined function but we should be attention to preserve the images labeled with the appropriate label:

```
def unison_shuffled_copies(a, b):
    assert len(a) == len(b)
    p = np.random.permutation(len(a))
    return a[p], b[p]
```

The assert keyword lets us to test if a condition in our code returns True, if not, the program will raise an Assertion Error. while,the function random.permutation(x) will randomly permute a sequence, or return a permuted range.

ii. Some Statistics

- **cross validation function**:
  StratifiedKFold cross validation was used in our observations with fixed value of K=5 and every time for a different model in order to compare and have the best result.

- **accuracy of cross validation** :
  For each split of CV, the model is fit to the training data, and predictive accuracy is assessed using the validation

data. The results are then averaged over the splits.

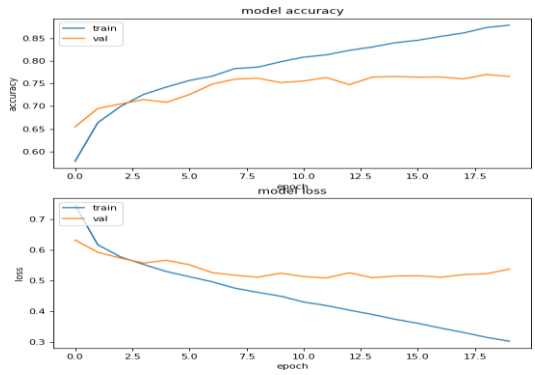
- **zero-one loss function** :
  a metric that we can measure the "cost" of an incorrect decision (miss-classification). It is a common loss function used with classification learning. It assigns 0 to loss for a correct classification and 1 for an incorrect classification

IX. Neural Network Models

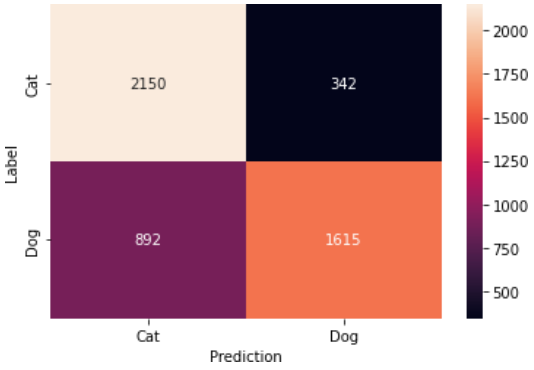
There are many different types of neural networks, varying in complexity. They share the intended goal of mirroring the function of the human brain to solve complex problems or tasks like classification or regression problems. The structure of each type of artificial neural network in some way mirrors neurons and synapses. However, they differ in terms of complexity, structure, and its suitability for the study-case . Many models were built in order to compare between the accuracy of each one and use the best one for our predictions

i. NN with single convolutions layer

A convolutional neural network consists of an input layer, hidden layers and an output layer while the activation's function of Rectified Linear Units(ReLu). The accuracy for this model was not so much high as desired (just 75.315%)



and the confusion matrix has the following shape :

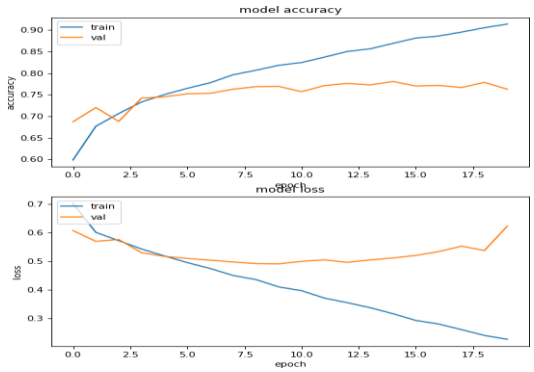


Applying the cross-validation for this model , provided us with average accuracy of 76.41 and average loss is: 1179.0 so we could say that both functions applied with this model presented **over-fitting** .

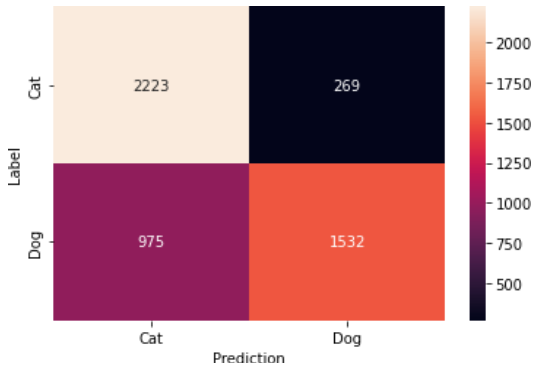
ii. Neural Network with some dropouts

Dropout randomly drops elements of its input, teaching the following layers not to rely on specific features or elements, but to use all information available. This enforces the network to generalize better and is a mean to reduce overfitting. in other words, dropout is used to regularize dense layers

which are very prone to over-fit. The accuracy for this model was not so much high as desired (just 75.115%) which is similar to the previous result. –> we still have over-fitting in our model.



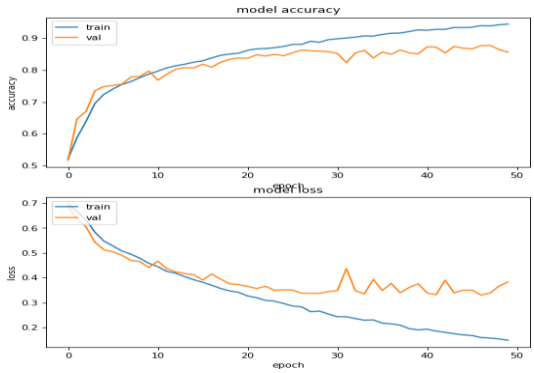
and the confusion matrix has the following shape :



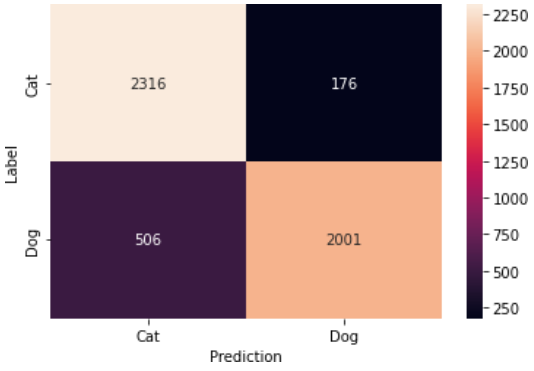
Applying the cross-validation for this model provided us with average accuracy of 77.09 and The average loss is: 1145.0 so we could say that the both functions applied with this model also presented **over-fitting** .

iii. Deep Model of Neural Network

Here,the network has more than one hidden layer so it can re-use the features computed in a given hidden layer in higher hidden layers. This enables a deep neural network to exploit compositional structure in a function, and to approximate many natural functions with fewer weights and units. This model is also called multi layer perceptrons and it's a quite good model for classification and prediction (as in our case).



and the confusion matrix has the following shape :

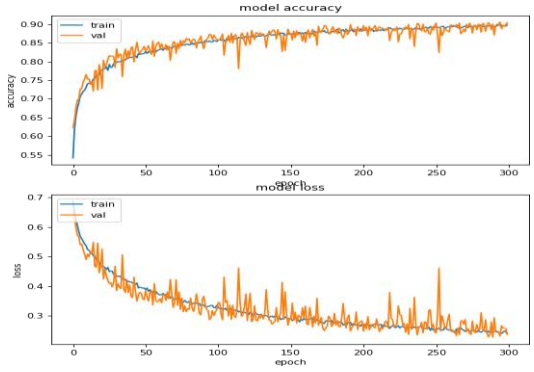


Applying the cross-validation for this model , provided us with average accuracy of: 88.3% and the average loss is: 585.0 so we could say that our model is getting enhanced in a noticeable rank and provide a good result and accuracy.

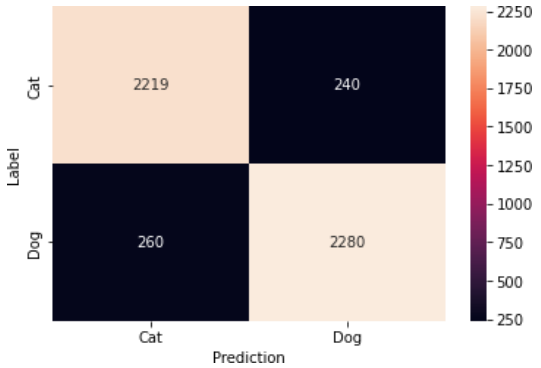
iv. Neural Network with augmented data

Training deep learning neural network models on more data can result in more skillful models, and the augmentation techniques can create variations of the images that can improve the ability of the fit models to generalize what they have learned to new images.

With more data available, deep learning neural networks frequently perform better. Data augmentation is a method for faking fresh training data out of old training data. To do this, domain-specific approaches are applied to instances from the training data to produce brand-new and distinctive training examples. This is mainly used in the image classification case and it involves creating transformed versions of images in the training data set.Transforms include a range of operations from the field of image manipulation, such as shifts, flips, zooms, and much more. Two techniques were used in this project for data augmentation: re-scaling and contrasting images.



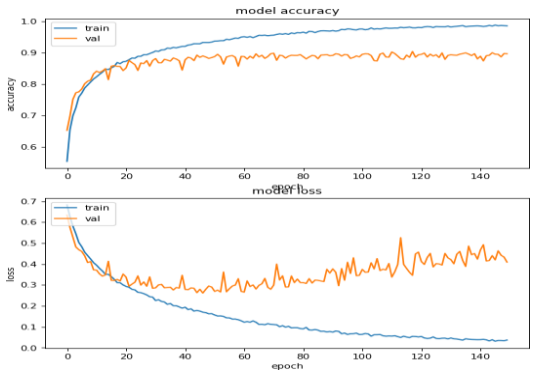
and the confusion matrix has the following shape :



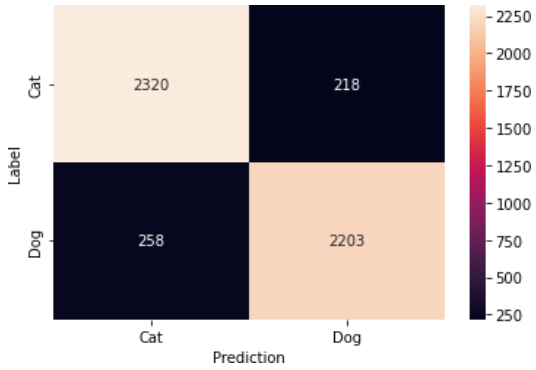
Applying the cross-validation for this model , provided us with average accuracy of: 87.8% and the average loss is: 610 so we could say that our model is getting enhanced in a noticeable rank and provide a good result and performance and our model will provide a higher accuracy by using 300 epochs which could lead us to a slightly better accuracy of (90%) as we saw in the experiment 1.

v. Neural Network without augmented data

This model is applied just to show how our model improved when the data is augmented . So, this model is applied to compare between the results and understand better the importance of augmented data in obtaining a higher resolution for the model.



and the confusion matrix has the following shape :



So, if we have looked carefully about the two previous models, we could detect that data augmentation model presented the same performance of the non-augmented data model but we preferably choose the augmented model in order to escape from the overfitting that has non-augmented one.

Cross-validation of non-augmented data ,provided us with average accuracy of: 88.58% and the average loss is: 571

X. Conclusion

CNN is a tough subject but a rewarding technique to learn. It teaches us how we perceive images and learn useful applications to classify images and videos. After learning CNN and using many different applications for the CNN algorithm in order to discover the model that provide us with the highest accuracy , I could confirm that the drawback of convolutional neural networks is that they can be sensitive to scaling and translation in the input image, which means it's difficult for them to handle an arbitrary orientation or size and another downside is that Convolutions neural networks can also suffer from over fitting if not enough regularization techniques are used during training and it requires a large number of labeled training data sets in order to work effectively.

According to provided dataset for this project , we could conclude that the best result we could obtain using the CNN algorithm is the one with augmented data as data augmentation is a powerful way to expand and improve your training dataset. It is a simple, low cost way to make your training data more robust and your model more effective in the field. Once we have augmented our data set to remove irrelevant biases and account for a wider array of inputs, there's only one more step before actually training your model.

References

[1] Very Deep Convolutional Networks for Large-Scale Image Recognition: Karen

Simonyan, Andrew Zisserman

[2] Neural networks and deep learning 2021/2022 , Nicolo' Cesa-Bianchi

[3] Neural Network Applications, J.G. Taylor

[4] Cross-validation: what does it estimate:Stephen Bates, Trevor Hastie, Robert Tibshirani

[5] Neural Networks for Classification: A Survey, Peter G. Zhang

[6] Loss Functions for Neural Networks for Image Processing:Hang Zhao, Orazio Gallo, Iuri Frosio, Jan Kautz

[7] Binary Classification from Multiple Unlabeled Datasets: Nan Lu, Shida Lei, Gang Niu, Issei Sato, Masashi Sugiyama

[8] Bias Plus Variance Decomposition for Zero-One Loss Functions, Ron Koha

[9] Machine Learning: A Review on Binary Classification:Roshan Kumari , Saurabh Kr. Srivastava

[10] Comparing the performance of different neural networks for binary classification problems,Publisher: IEEE , Conference Location: Bangkok, Thailand ,20-22 October 2009

[11] Performance Measures in Binary Classification,Matthias Kohl,Furtwangen University

[12] Augmenting weighted average with confusion matrix to enhance classification accuracy,VM Patro, MR Patra

[13] Documentation of the Keras.io https://keras.io/

Final Declaration:

We declare that this material, which We now submit for assessment, is entirely our own work and has not been taken from the work of others, save and to the extent that such work has been cited and acknowledged within the text of our work. We understand that plagiarism, collusion, and copying are grave and serious offences in the university and accept the penalties that would be imposed should I engage in plagiarism, collusion or copying. This assignment, or any part of it, has not been previously submitted by us or any other person for assessment on this or any other course of study.