IMPROVING ON CIFAR-10 IMAGE CLASSIFICATION ACCURACY USING CONVOLUTIONAL NEURAL NETWORK (CNN)

FOLORUNSO, OLUBAYO DAVID

MSC COMPUTER SCIENCE, UNIVERSITY OF EAST LONDON

RESEARCH MOTIVATION

I have worked with graphical images and audiovisuals for more than a decade. One of the major challenges I identified was the inability of clients to recall names or faces of individuals in photographs after a considerably long period of time. This attracted me to the possibility of developing or improving the performance of an image classifier that could recognize faces and features in graphics objects.

OBJECTIVE OF THE RESEARCH WORK

To improve on CIFAR-10 image classification accuracy.

ALGORITHM USED FOR CLASSIFICATION

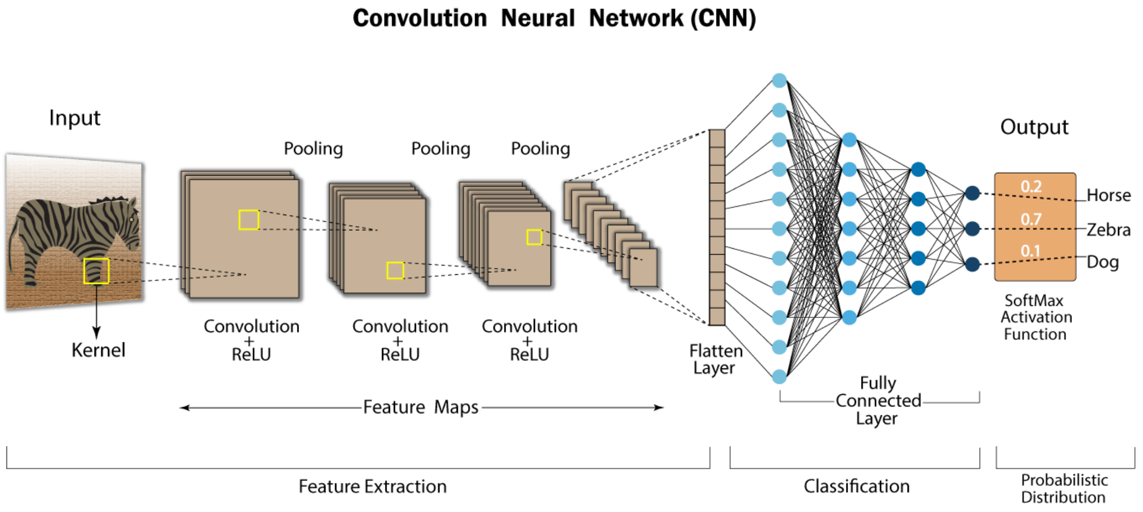
Convolutional Neural Network (CNN)

CONVOLUTIONAL NEURAL NETWORK (CNN)

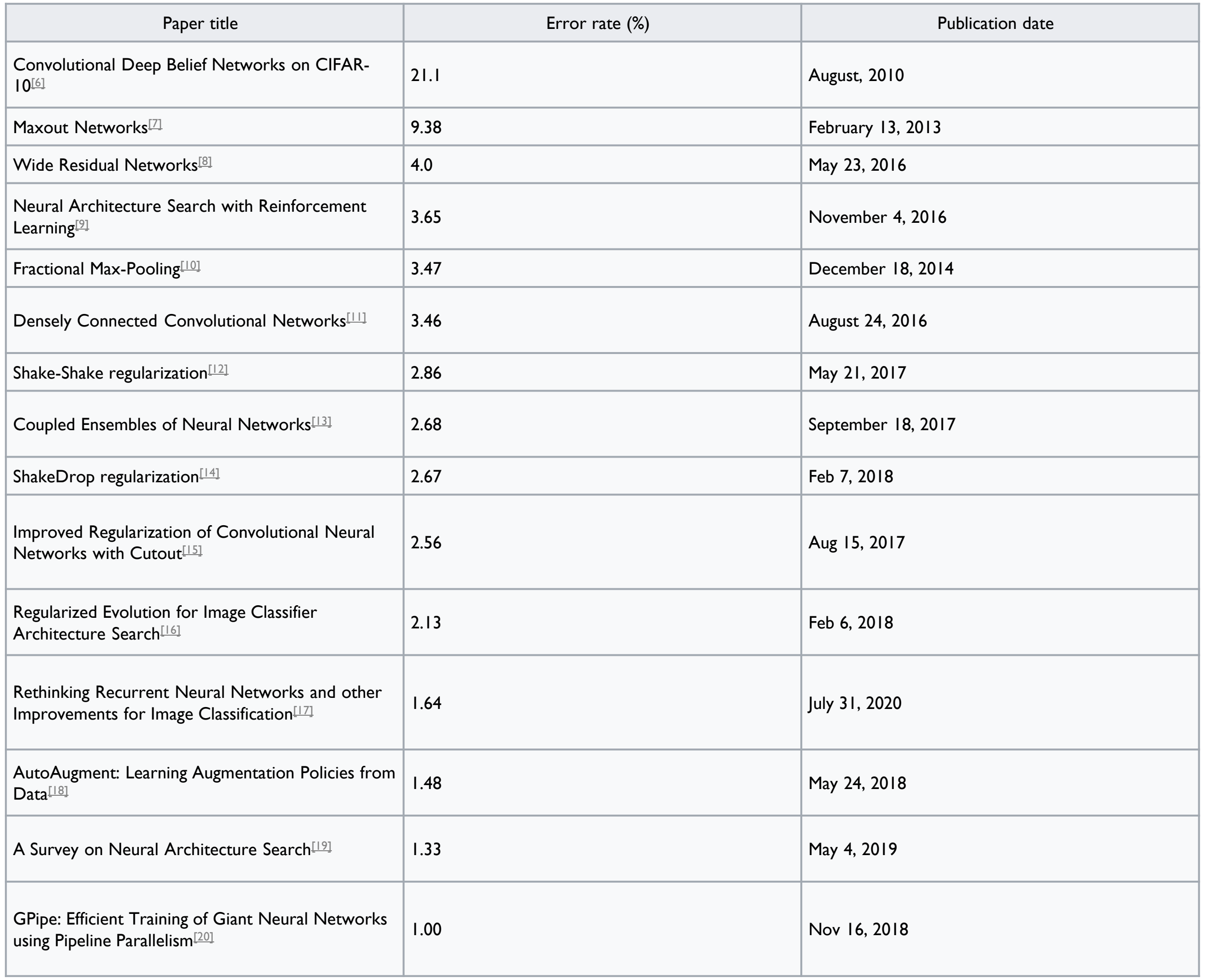
Convolutional neural networks (CNN) consist of neurons that have weights and biases. To create an effective learning model, these weights and biases are adjusted during the training process. Every neuron takes a set of inputs, goes through some sort of processing, and finally outputs a value.

ARCHITECTURE OF CNN

1. Convolutional layer
2. Rectified Linear Unit layer
3. Pooling layer
4. Fully connected layer



PREVIOUS RESEARCH PAPERS ON CIFAR-10



Paper title	Error rate (%)	Publication date
Convolutional Deep Belief Networks on CIFAR-10[6]	21.1	August, 2010
Maxout Networks[7]	9.38	February 13, 2013
Wide Residual Networks[8]	4.0	May 23, 2016
Neural Architecture Search with Reinforcement Learning[9]	3.65	November 4, 2016
Fractional Max-Pooling[10]	3.47	December 18, 2014
Densely Connected Convolutional Networks[11]	3.46	August 24, 2016
Shake-Shake regularization[12]	2.86	May 21, 2017
Coupled Ensembles of Neural Networks[13]	2.68	September 18, 2017
ShakeDrop regularization[14]	2.67	Feb 7, 2018
Improved Regularization of Convolutional Neural Networks with Cutout[15]	2.56	Aug 15, 2017
Regularized Evolution for Image Classifier Architecture Search[16]	2.13	Feb 6, 2018
Rethinking Recurrent Neural Networks and other Improvements for Image Classification[17]	1.64	July 31, 2020
AutoAugment: Learning Augmentation Policies from Data[18]	1.48	May 24, 2018
A Survey on Neural Architecture Search[19]	1.33	May 4, 2019
GPipe: Efficient Training of Giant Neural Networks using Pipeline Parallelism[20]	1.00	Nov 16, 2018

This is a table of some of the research papers that claim to have achieved state-of-the-art results on the CIFAR-10 dataset

Extracted from Wikipedia.

https://en.wikipedia.org/wiki/CIFAR-10

CIFAR-10 DATASET

CIFAR-10 dataset was used in this research, which is a labeled dataset containing 60,000 images from Canadian Institute for Advance Research.

CIFAR-10 contains ten classes: deer, ship, bird, cat, dog, automobile, frog, truck, airplane and horse. Each class contains exactly 6,000 images and the classes are completely mutually exclusive. It was collected by Alex Krizhevsky Vinod Nair, and Geoffrey Hinton. (Alex, 2010).

The class labels and their standard associated integer values are listed below.

0: airplane
1: automobile
2: bird
3: cat
4: deer
5: dog
6: frog
7: horse
8: ship
9: truck

COMPILATION SYSTEM CONFIGURATION

The training of the deep learning model was carried out on Windows PC with the following configurations.

Operating System: Windows 11 Pro 64-bit (10.0, Build 22000)

Processor: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (8 CPUs), ~1.8GHz

Memory: 16384MB RAM

IDE Environment: Jupyter Notebook running on Anaconda.

BUILDING THE CNN MODEL

1. CIFAR-10 dataset was imported from keras.datasets, which has train dataset and test dataset already prepared.
2. The train dataset (50,000) and test dataset (10,000), image dimension (32x32) and RGB (3) channels were confirmed with x_train.shape and x_test.shape
3. y_train and y_test were converted from 2D array to 1D array (y_train = y_train.reshape(-1,) and y_test = y_test.reshape(-1,)).
4. The classes of the dataset was introduced for proper labelling (classes = ["airplane","automobile","bird","cat","deer","dog","frog","horse","ship","truck"]

BUILDING THE CNN MODEL

5. Dataset was normalized (x_train = x_train / 255, x_test =x_test / 255
6. CNN model was built for image classification

```
      cnn = models.Sequential([
   layers.Conv2D(filters=512, kernel_size=(3, 3), activation='relu', input_shape=(32, 32, 3)), layers.MaxPooling2D((2, 2)),
   layers.Conv2D(filters=512, kernel_size=(3, 3), activation='relu’), layers.MaxPooling2D((2, 2)),
   layers.Conv2D(filters=512, kernel_size=(3, 3), activation='relu’), layers.MaxPooling2D((2, 2)),

   layers.Flatten(),
   layers.Dense(64, activation='relu'),
   layers.Dense(10, activation='softmax')
])
```

BUILDING THE CNN MODEL

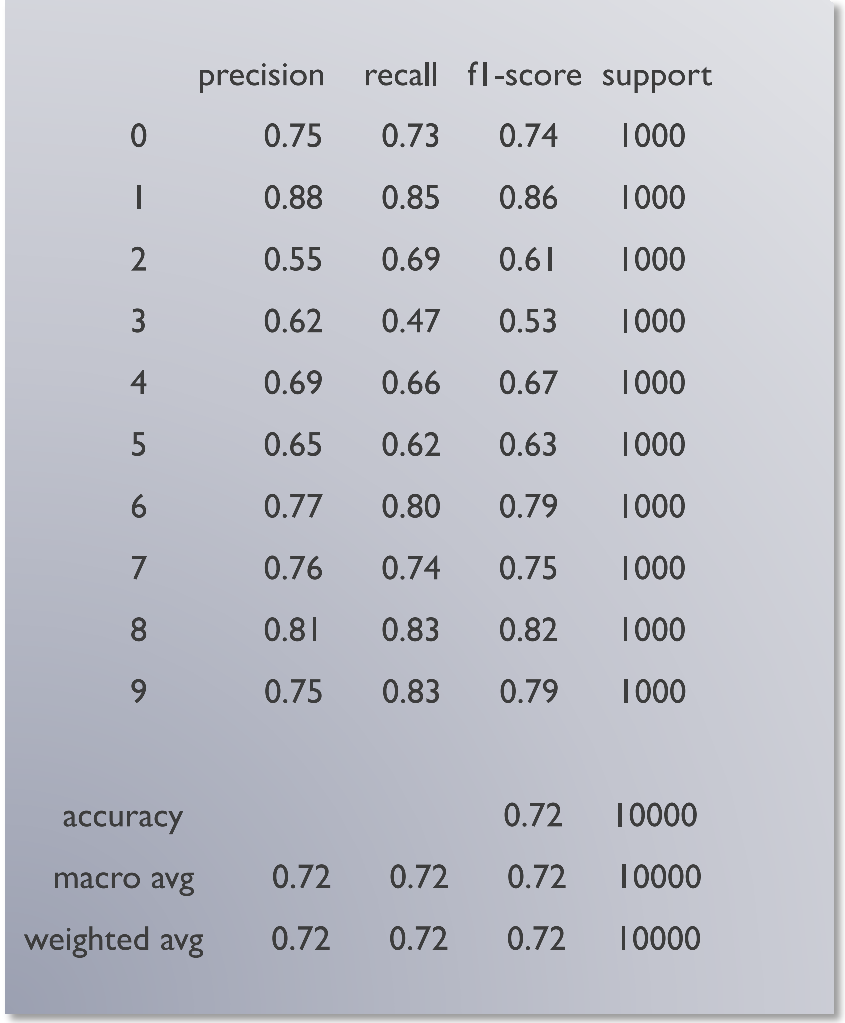
7. The compiler was configured.

```
cnn.compile(optimizer='adam’,
loss='sparse_categorical_crossentropy’,
metrics=['accuracy’])
```

8. The dataset was trained on 20 epochs.

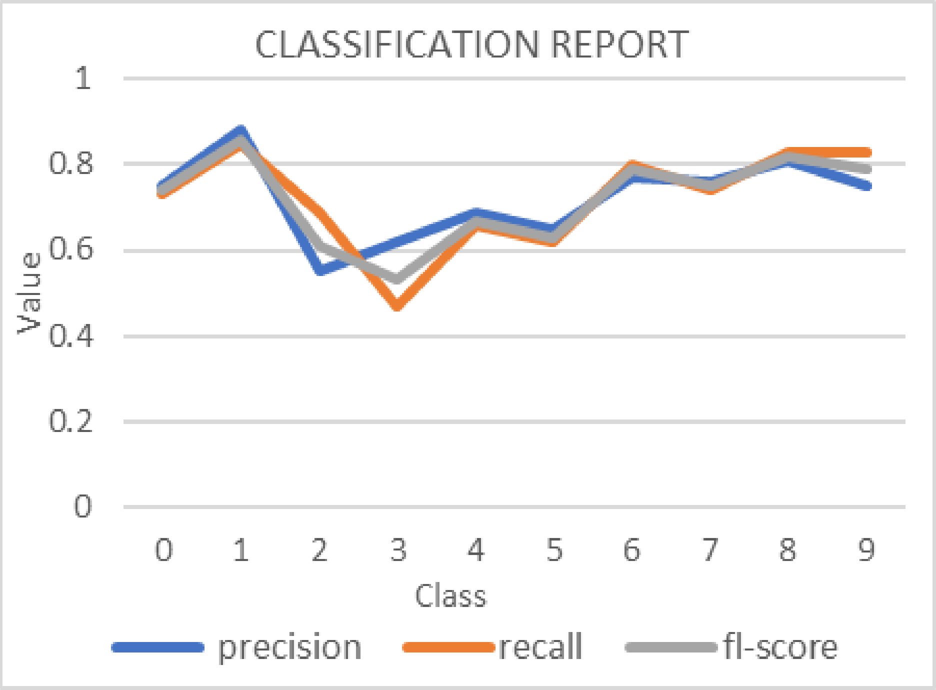
```
Epoch 1/20
1563/1563 [========================] - 1109s 709ms/step - loss: 1.5184 - accuracy: 0.4469
………
Epoch 20/20
1563/1563 [=============================] - 1136s 727ms/step - loss: 0.1057 - accuracy: 0.9632
```

CLASSIFICATION REPORT



	precision	recall	f1-score	support
0	0.75	0.73	0.74	1000
1	0.88	0.85	0.86	1000
2	0.55	0.69	0.61	1000
3	0.62	0.47	0.53	1000
4	0.69	0.66	0.67	1000
5	0.65	0.62	0.63	1000
6	0.77	0.80	0.79	1000
7	0.76	0.74	0.75	1000
8	0.81	0.83	0.82	1000
9	0.75	0.83	0.79	1000
accuracy			0.72	10000
macro avg	0.72	0.72	0.72	10000
weighted avg	0.72	0.72	0.72	10000

The dataset was trained using CNN (Convolutional Neural Network to provide the experimental findings listed in the table. The model was trained, and it now has a weighted average accuracy of 72%; the precision value of each of the 10 classes in the CIFAR-10 dataset, together with recall and f1-score, are listed also in the table.



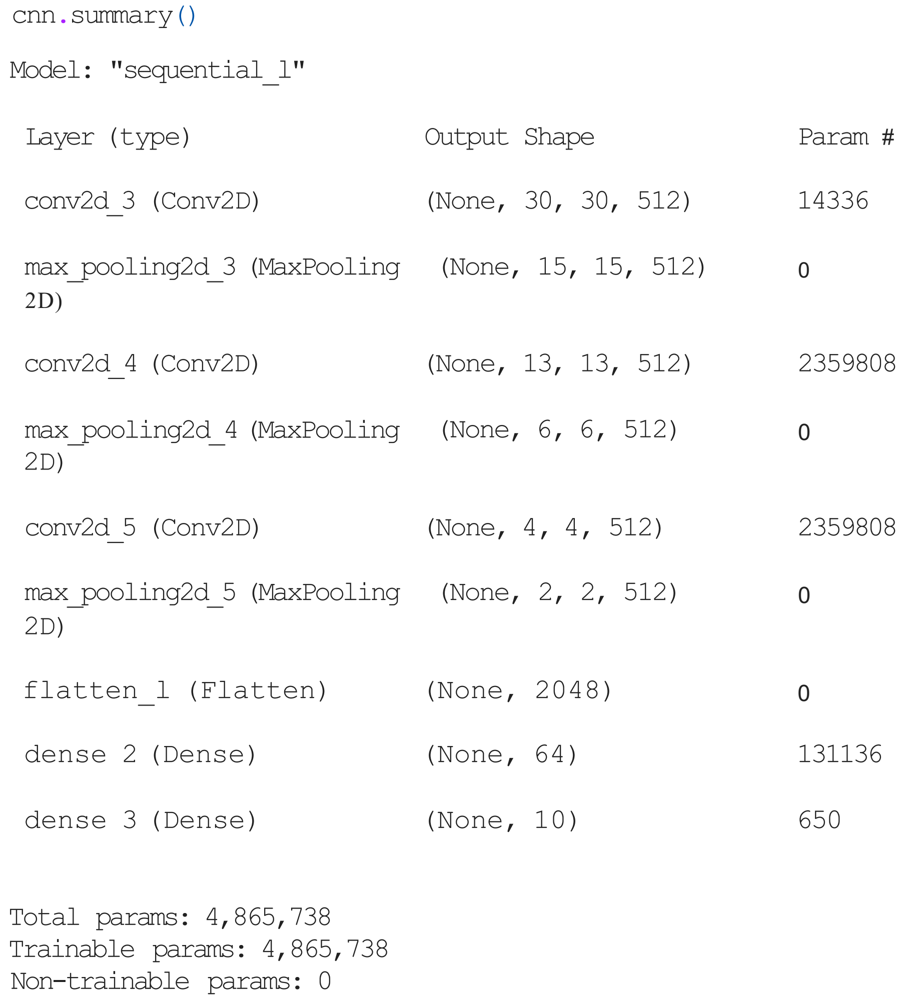
EVALUATING THE MODEL

The accuracy of the CNN model using test dataset for evaluation is 74%.
The model classifies correctly 7 out of every 10 images.

```
cnn.evaluate(x_test,y_test)

313/313 [==============================] - 7s
22ms/step - loss: 1.7730 - accuracy: 0.7402

[1.7730391025543213, 0.7401999831199646]
```

ANALYSIS OF THE CNN

In setting up the parameters for the convolutional layer, I had input_shape of 32. 512 filters were used for the layer. Kernel size was (3,3), the poling layer was Max Pooling with (2,2) parameters and the activation layer for the Conv2D was ReLU. Three convolutional layers were set up with the above parameters. The activation layer for the fully connected layer was ReLU and output layer was Softmax. This configuration brings the total number of model trainable parameters to 4, 865,738.

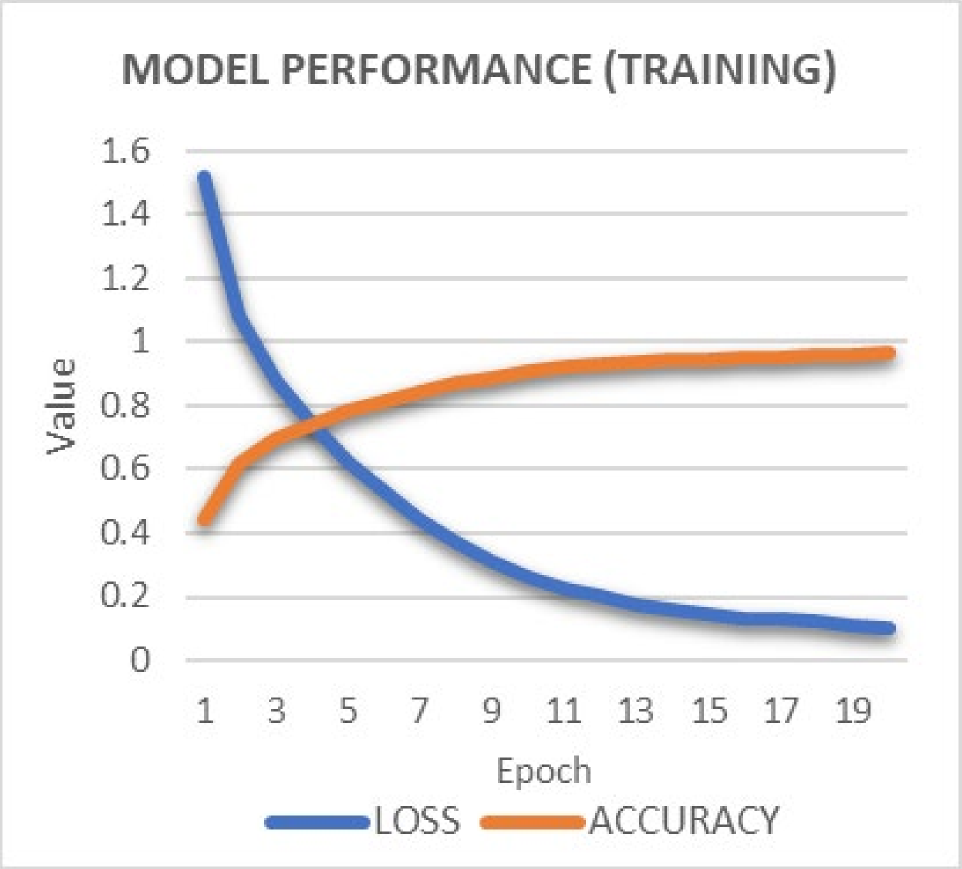


```
cnn.summary()

Model: "sequential_1"

 Layer (type)                Output Shape              Param #

 conv2d_3 (Conv2D)           (None, 30, 30, 512)       14336

 max_pooling2d_3 (MaxPooling  (None, 15, 15, 512)      0
 2D)

 conv2d_4 (Conv2D)           (None, 13, 13, 512)       2359808

 max_pooling2d_4 (MaxPooling  (None, 6, 6, 512)        0
 2D)

 conv2d_5 (Conv2D)           (None, 4, 4, 512)         2359808

 max_pooling2d_5 (MaxPooling  (None, 2, 2, 512)        0
 2D)

 flatten_1 (Flatten)         (None, 2048)              0

 dense 2 (Dense)             (None, 64)                131136

 dense 3 (Dense)             (None, 10)                650


Total params: 4,865,738
Trainable params: 4,865,738
Non-trainable params: 0
```

PERFORMANCE OF THE MODEL

The model performed well on training dataset, ending on 96.32% accuracy in correctly classifying images it was trained on with just 20 epochs. The accuracy climbs fast between epoch 1 and 10 but plateaued from epoch 11 to 20. The accuracy started 0.4469 on epoch 1 and climbed to 0.9073 on epoch 10; but ended on 0.9632 on epoch 20.

On the loss side, the value dropped from 1.5184 to 0.1057 on the last epoch. The drop in loss value also slowed down at epoch 10, reflecting what was happening to the training accuracy of the model at the same time.



EXPLANATIONS

The convolutional layer was setup with filter 512 been multiple of 64. The input size was 32 because the CIFAR-10 dataset has dimension of 32 x 32 coloured images. Max pooling was (2, 2) so that the corner pixels of the images can contribute to the feature extraction multiple times like the other parts of the images. ReLU was the activation function in other to avoid the complex computation involved in Sigmoid function activation; with ReLU the forward and backward propagations are just 'if' statement. It simplifies the computation unlike Sigmoid that computes an exponent.

Model performance was average for cat class with f1-ratio of 0.53 while it performed best in predicting automobile with 0.86 f1-ratio. It may be that the cat images were blurred or bad beyond recognition; it could also be that the parameters used did not well in classifying the cat class, though this is a rare situation.

CONCLUSION

This research work proposed an improvement on the Convolutional Neural Network (CNN) model to classify CIFAR-10 image dataset.

The proposed model was trained on a dataset that included the 60,000 images of 10 classes.

This model performs slightly lower than the best of research papers known and listed on Wikipedia.

The validated accuracy of the model was 74.01% while the error stood at 1.77.

Therefore, a new CNN architecture is proposed for future research.

THANK YOU