Malaria Detection problem using deep learning

Abstract: Malaria caused by the Plasmodium parasites, is a blood disorder, which is transmitted through the bite of a woman Anopheles mosquito. With almost 240 million cases mentioned each year, the sickness puts nearly forty percentage of the global populace at danger.Here in this problem,we basically using a CNN model for checking whether a person is infected or uninfected by malaria.For using CNN model we getting much better accuracy and can easily classify.

Dataset source: https://www.kaggle.com/iarunava/cell-images-for-detecting-malaria

Steps behind this problem:

- Data Gathering
- Data Preprocessing
- Model Fiting

Data Gathering

Dataset collected from kaggle,data set contain two dataset ,one is test and other is train dataset.The dataset have two categories(parasite and uninfected)

Data Preprocessing

In data preprocessing , First we need to transform image in a much bug size as our orginal image for getting a good view, and i did a simple method to rescale this image means transform this pixel size to 0 to 1.

Then apply some data augmentation to training data to create more images with different angle this is on for training data to train our CNN model with large number of images,in this way can achieve more accurate results from our model.

Model Fiting

Creating CNN model

Here I created a simple CNN model from scratch

This is the mode summary

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_1 (Conv2D)            (None, 224, 224, 16)      208       
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 112, 112, 16)      0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 112, 112, 32)      2080      
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 56, 56, 32)        0         
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 56, 56, 64)        8256      
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 28, 28, 64)        0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 50176)             0         
_________________________________________________________________
dense_1 (Dense)              (None, 500)               25088500  
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 1002      
=================================================================
Total params: 25,100,046
Trainable params: 25,100,046
Non-trainable params: 0
```

Here last Dense layer have 2 node for two classes(Parasite and Uninfected)

Mostly used activation functions are :

Relu: The **rectified linear activation function** or **ReLU** for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero.

Softmax: Softmax is an activation function **that scales numbers/logits into probabilities(0 to 1)**

Categorical crossentropy: Categorical crossentropy is **a loss function that is used in multi-class classification tasks**. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one. Formally, it is designed to quantify the difference between two probability distributions.

Language used: Python

IDE : Jupyter Notebook,Pycharm

All the librarie:

Tensorflow ,Keras,numpy,matplotlib