CNN MODEL FOR PREDICTING IMAGES

Megha Trivedi
Machine Learning intern
AI Technology and Systems
meghat525@gmail.com
www.ai-techsystems.com

***Abstract*—In this paper, we discuss about making a CNN model for a fruits image dataset taken from Kaggle. The aim is to make a CNN model that can predict the class of images with maximum possible accuracy.**

Keywords—CNN, Softmax, Cross-Entropy

I. Introduction

In deep learning, a convolutional neural network is a class of deep neural networks, mostly used for analysing visual imagery. A convolutional neural network is just an artificial neural network on which we use the convolutional trick to add some convolutional layers.

Here we use Softmax activation function and cross-entropy loss function to achieve maximum accuracy.

II. Softmax

Softmax function is a generalization of the logistic function. Softmax is applied through a neural network layer just before the output layer. The Softmax layer must have the same number of nodes as the output layer. The mathematical equation of the Softmax activation function is:

$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

The softmax function squashes the outputs of each unit to be between 0 and 1, just like a sigmoid function. But it also divides each output such that the total sum of the outputs is equal to 1. The output of the softmax function is equivalent to a categorical probability distribution, it tells you the probability that any of the classes are true.

Softmax output is large if the score (input called logit) is large. Its output is small if the score is small. The proportion is not uniform. Softmax is exponential and enlarges differences - push one result closer to 1 while another closer to 0. It turns scores aka logits into probabilities. Cross entropy (cost function) is often computed for output of softmax and true labels (encoded in one hot encoding). the essential goal of softmax is to turn numbers into probabilities.

III. Cross-Entropy

A. Cross-entropy is a loss function that measures the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverges from the actual label. So predicting a probability of .012 when the actual observation label is 1 would be bad and result in a high loss value. A perfect model would have a log loss of 0. The mathematical equation of cross-entropy is:

$$Cross\ Entropy = -\overset{③}{\frac{1}{N}} \overset{②}{\sum_j} \overset{①}{y_j * \log(\hat{y}_j)}$$

① true label * log(predicted)

② sum over all sequences in each batch

③ divide by the number of samples

Cross entropy is always larger than entropy; encoding symbols according to the wrong distribution ŷ will always make us use more bits. The only exception is the trivial case where y and ŷ are equal, and in this case entropy and cross entropy are equal.

IV. Methodology

A. Dataset

We collected the dataset of images from Kaggle. It is a dataset of images containing fruits and vegetables.
The following fruits and vegetables are included: Apples (different varieties: Crimson Snow, Golden, Golden-Red, Granny Smith, Pink Lady, Red, Red Delicious), Apricot,

Avocado, Avocado ripe, Banana (Yellow, Red, Lady Finger), Beetroot Red, Blueberry, Cactus fruit, Cantaloupe (2 varieties), Carambula, Cherry (different varieties, Rainier), Cherry Wax (Yellow, Red, Black), Chestnut, Clementine, Cocos, Dates, Ginger Root, Granadilla, Grape (Blue, Pink, White (different varieties)), Grapefruit (Pink, White), Guava, Hazelnut, Huckleberry, Kiwi, Kaki, Kohlrabi, Kumsquats, Lemon (normal, Meyer), Lime, Lychee, Mandarine, Mango (Green, Red), Mangostan, Maracuja, Melon Piel de Sapo, Mulberry, Nectarine (Regular, Flat), Onion (Red, White), Orange, Papaya, Passion fruit, Peach (different varieties), Pepino, Pear (different varieties, Abate, Forelle, Kaiser, Monster, Red, Williams), Pepper (Red, Green, Yellow), Physalis (normal, with Husk), Pineapple (normal, Mini), Pitahaya Red, Plum (different varieties), Pomegranate, Pomelo Sweetie, Potato (Red, Sweet, White), Quince, Rambutan, Raspberry, Redcurrant, Salak, Strawberry (normal, Wedge), Tamarillo, Tangelo, Tomato (different varieties, Maroon, Cherry Red, Yellow), Walnut.

B. *Importing Libraries*

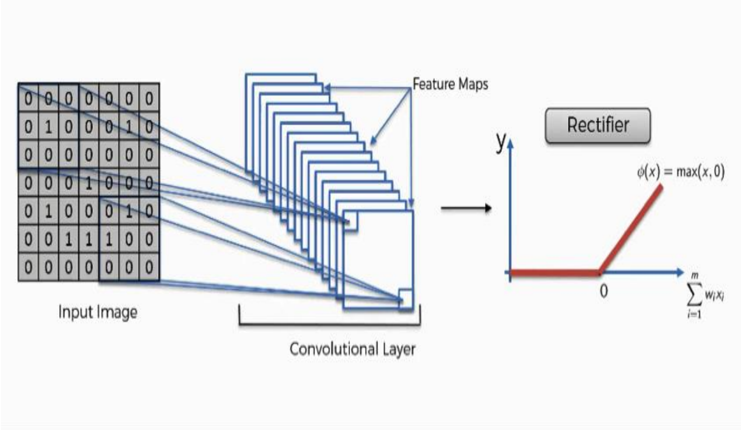
Imported the libraries:
Keras: An open-source neural-network library written in Python.
Keras is an API designed for human beings, not machines. It puts user experience front and center. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error.
A model is understood as a sequence or a graph of standalone, fully configurable modules that can be plugged together with as few restrictions as possible. In particular, neural layers, cost functions, optimizers, initialization schemes, activation functions and regularization schemes are all standalone modules that you can combine to create new models.
New modules are simple to add (as new classes and functions), and existing modules provide ample examples. To be able to easily create new modules allows for total expressiveness, making Keras suitable for advanced research.
No separate model's configuration files in a declarative format. Models are described in Python code, which is compact, easier to debug, and allows for ease of extensibility.

C. *Convolution Operation*

Here we have used a feature detector of 3x3 matrix.

D. *ReLU Layer*

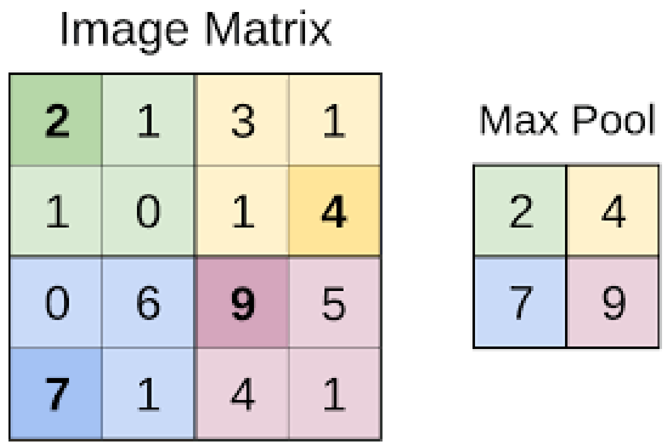
Applying rectifier over the convolution. The Rectified Linear Unit, or ReLU, is not a separate component of the convolutional neural networks' process.
It's a supplementary step to the convolution operation that we did earlier.



The rectifier serves to break up the linearity even further in order to make up for the linearity that we might impose an image when we put it through the convolution operation.

E. *Max Pooling*

The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned.
By reducing the information, we are preventing overfitting.



F. *Flattening*

Flattening transforms a two-dimensional matrix of features into a vector that can be fed into a fully connected neural network classifier.

G. *Full Connection*

Adding a whole ANN to our CNN.
An epoch is when you go through your dataset again and again and there are lots and lots of iterations.

V. EXPERIMENTAL RESUTS

We set the number of epochs equal to 7 and the model predicted images with an accuracy of 99.54%.

ACKNOWLEDGEMENT

I would like to thank AITS for giving me the opportunity to write a paper and for providing an environment where I can learn and grow.

Moreover, I would like to thank my parents for the constant support.

REFERENCES

1. Kaibo Duan1, S. Sathiya Keerthi, Wei Chu1, Shirish Krishnaj Shevade, and Aun Neow Poo "Multi-Category Classification by Soft-Max Combination of Binary Classifiers".

2. Zhilu Zhang and Mert R. Sabuncu "Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels".
3. Keras Documentation.