

DIGIT RECOGNITION

Project Smart Systems

References

https://www.tutorialspoint.com/keras/keras_dense_layer.htm

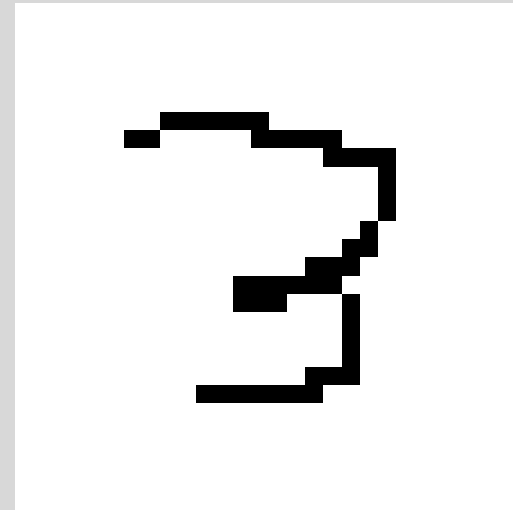
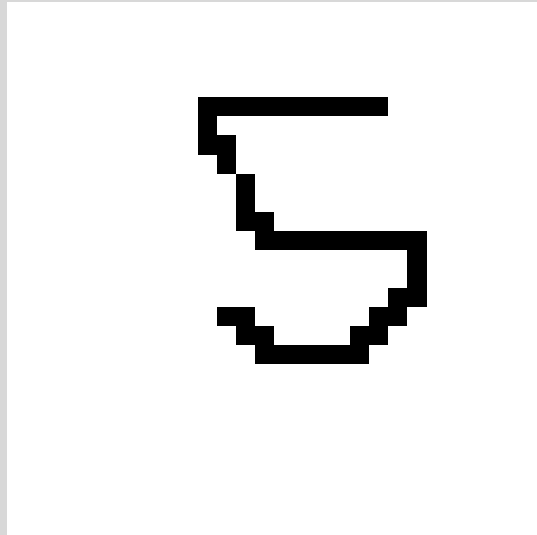
https://www.tutorialspoint.com/keras/keras_model_compilation.htm

<https://youtu.be/68BZ5f7P94E>

<https://youtu.be/m0pIIlfpXWE>

https://www.tensorflow.org/api_docs/python/tf/keras/datasets

Neural network has many applications. Today we will talk about a small project about neural network Which is digit recognition.



TRAINING PHASE

A close-up, low-angle shot of a reddish-brown running track. The track has white lane markings, including a prominent diagonal line running from the bottom left towards the top right, and several horizontal lines. The surface of the track is granular and textured. In the top left corner, the words "TRAINING" and "PHASE" are written in a large, white, sans-serif font, stacked vertically.

We will use minst dataset which is data consist of digits 28*28 . We will use bunch of libraries such as matplotlib, numpy , cv and tensorflow.We will download the keras data by tensorflow library which provides this function . We split the data to train and test data first phase which is training the data we get all the labeled data with all the digits and all the classification and we train the model to get the optimization then we use the testing data to assess the model. After we split the data we normalize it for example if we have grey scale rgb from 0 to 255 we normalize it to 0 to 1 . Normalization makes it easier for the neural network calculations. We will use The sequential API allows you to create models layer-by-layer for most problems. It is limited in that it does not allow you to create models that share layers or have multiple inputs or outputs.we will add the first layer which is the flattening layer to make it 1d so we will multiply 28*28 . Then we will dense layer which is a layer that consists of deeply connected neural network layer .

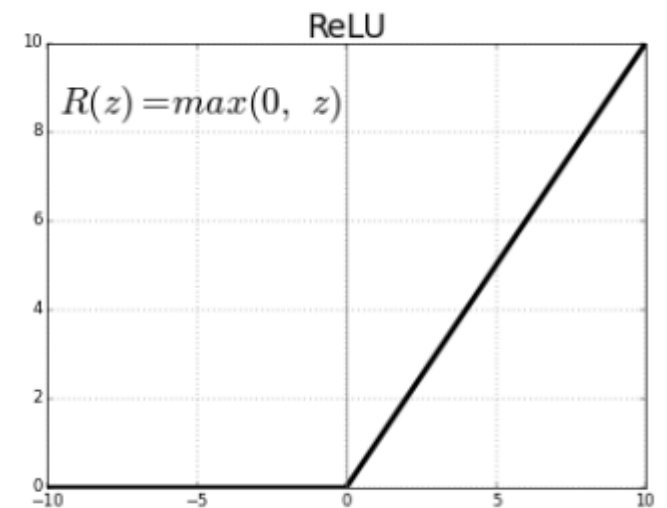
<https://www.youtube.com/watch?v=m0pIILfpXWE>. Relu it is a rectify linear unit is very basic just 0 in the negative and goes straight up linearly. $F(x) = \max(0, x)$

Compilation is the last step in creating a model. Once the compilation is done we Can move on to training phase .

Loss function is used to find the error in the learning process . In keras dataset it Requires loss function.

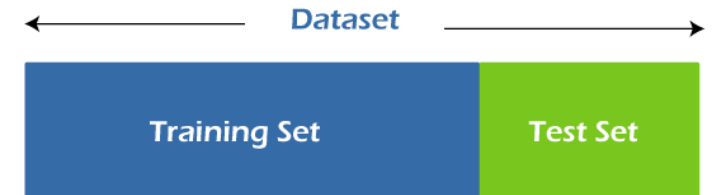
Optimization important process as it optimizes the input weights by comparing The pred and the loss we used adam optimizer which uses gradient descent Estimates first and sec order.

Metrics is used to evaluate the performance of your model.



Testing phase

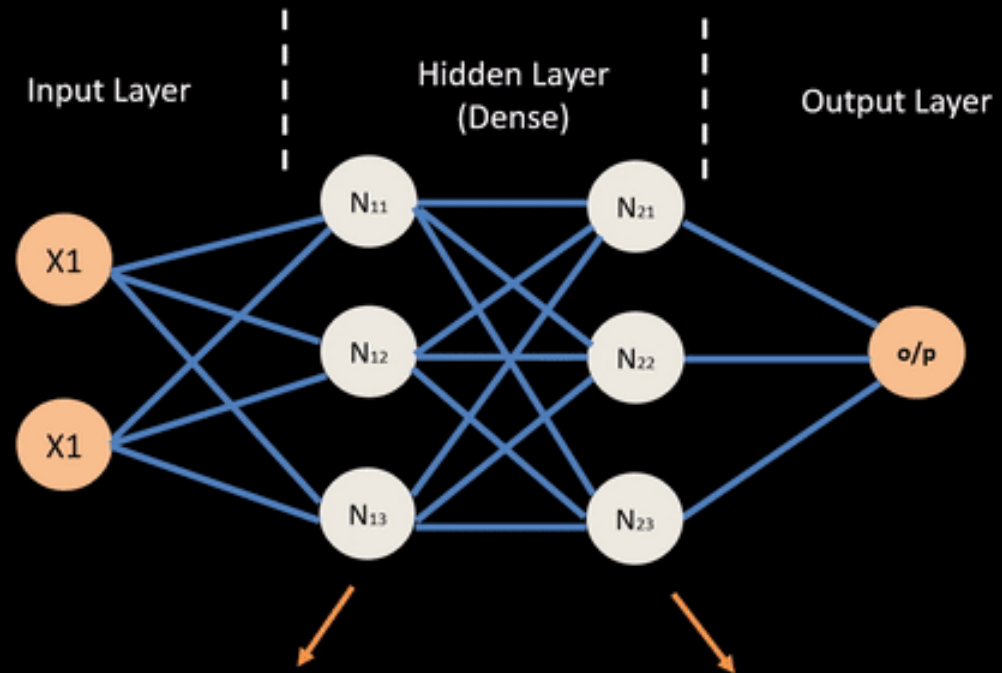
- After training the data its time for the testing phase to see the accuracy of the model and fitting it and it is used it to evaluate the performance and progress of your algorithms' training and adjust or optimize it for improved results.



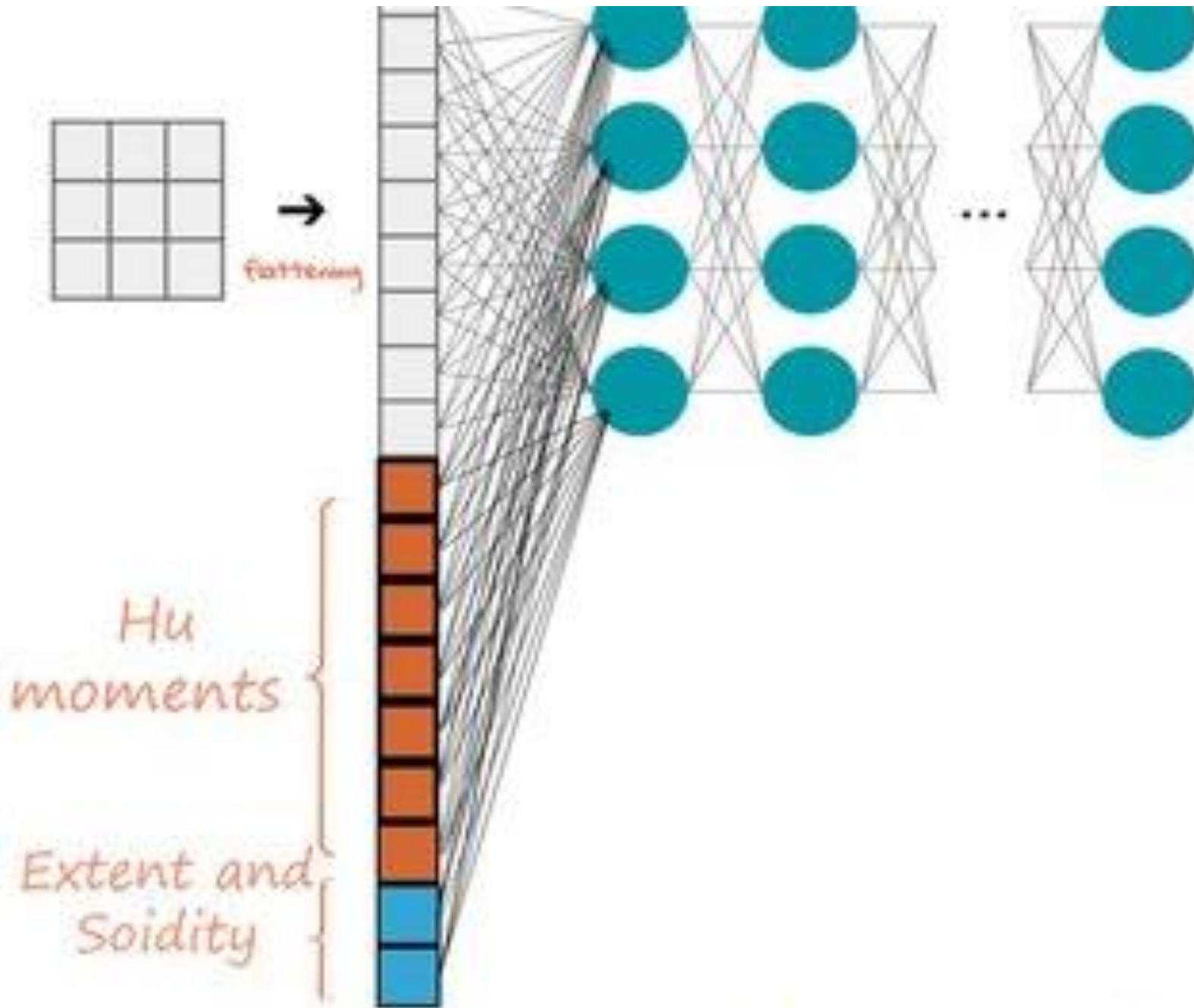
THE DENSE LAYER

will dense layer which is a layer that consists of deeply connected neural network layer

Dense Layer in Deep Neural Network



Each Neuron in Dense Layer receives input from all neurons of previous layer



Flattened layer

In order to save the layered image in a single-layer graphics format such as TIFF or JPEG, the image is said to be "flattened."

Data Values	Normalized
13	0
16	0.0517
19	0.1034
22	0.1552
23	0.1724
38	0.4310
47	0.5862
56	0.7414
58	0.7759
63	0.8621
65	0.8966
70	0.9828
71	1

To normalize the values in a dataset to be between 0 and 1, you can use the following formula:

$$z_i = (x_i - \min(x)) / (\max(x) - \min(x))$$

we normalize it for example if we have grey scale rgb from 0 to 255 we normalize it to 0 to 1 . Normalization makes it easier for the neural network calculations.

- Firstly we will check if there is a file in the following path by using the package cv then it will read the file we will invert it back like it was white on black not black on white we will put it in an array .We will predict the model and then show the image.

