A synopsis on

Handwritten Digit Recognition on MNIST dataset

Submitted By

A22 Eshan Kasliwal

A42 Pratham Solanki

A70 Pratik Jade

Guided By:

Mr. Ganesh Kadam

Department of Artificial Intelligence

G.H. Raisoni College of Engineering and Management, Wagholi, Pune

2021 - 2022

Contents:

Mr Ganesh Kadam — **Guide**

Prof. Rachna Sable — **H.O.D**

1. Introduction:

Artificial Intelligence, giving machines human-like abilities, has remained one of the most challenging areas in electronic sciences in the last few decades. Giving machines the power to see, interpret and the ability to read text is one of the major tasks of AI.
In the field of Machine Learning, the recognition of objects has become the most sought. Some of the examples of object recognition are Face recognition, Handwrite recognition, Disease detection etc. All these things can happen through a large set of the image data set. These image data sets will contain both positive and negative data regarding that domain. This helps the algorithm to classify the unknown data in better ways. Handwrite recognition is a new technology that will be useful in this 21st century. Handwritten digit recognition is the process of providing the ability to machines to recognize human handwritten digits. It is not an easy task for the machine because handwritten digits are not perfect, vary from person-to-person, and can be made with many different flavours.

a) **MNIST database**-

The MNIST database (Modified National Institute of Standards and Technology database) is a handwritten digit dataset. We can use it for training various image processing systems. The database is also widely used for training and testing in the field of machine learning. It has 60,000 training and 10,000 testing examples. Each image has a fixed size. The images are of size 28*28 pixels. It is a database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal effort on pre-processing and formatting. We will use this database in our experiment.

Neural networks offer complete independence of the recognition process and character set. This neural network is first trained by multiple sample images of each alphabet. Then in the recognition process, the input image is directly given to the neural network and the recognized symbol is outputted. The advantage of neural networks is that the domain of the character set can be very easily extended, one just needs to train the network over the new set. Another advantage is that neural networks are very robust to

noise. The disadvantage is that a lot of training is required which is very time-consuming. None of the above approaches thus used in their pure form yields good results on handwritten text. So generally, hybrid approaches are taken to achieve desired recognition rates.

b) **Convolutional Neural Networks**-

Convolutional neural networks are deep artificial neural networks. We can use it to classify images (e.g., name what they see), cluster them by similarity (photo search) and perform object recognition within scenes. It can be used to identify faces, individuals, street signs, tumours, platypuses and many other aspects of visual data. The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels) which have a small receptive field but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product, and producing a 2-dimensional activation map of that filter. As a result, the network learns when they see some specific type of feature at some spatial position in the input. Then the activation maps are fed into a downsampling layer, and like convolutions, this method is applied one patch at a time. CNN has also a fully connected layer that classifies output with one label per node.

2. Literature Review / Related Work

Levi & Hassner (2015), proposed CNN architecture to overcome the overfitting problem due to less number of images. The result shows that CNN provides improved gender and age classification result even in the case of the smaller sizes of contemporary unconstrained image sets. For large-scale image classification using CNN with a dataset of 1 million YouTube videos belonging to 487 classes is experimented CNN is used to learn features and classify RGB-D images task. The various parameter has an effect on the accuracy of training results. The authors presented the effect of mini-batch size on the training model. The model gave 93.33% accuracy on the test set with a minimum batch size of 10. CNN for document image classification is presented in a paper by Kang et al. (2014). Tang, Y. (2013), proved that the replacement of the softmax layer with SVMs is useful for classification tasks. The experimentation was carried out on MNIST and CIFAR-10 datasets. Using SVM, cross-validation accuracy for the testing phase is increased up to 68.9% which was 67.6% using softmax. Results of RNN for classification of FashionMNIST dataset presented by Zhang. This model was developed using the Long-Short Term Memory technique to reduce the risk of gradient vanishing the traditional RNN faces. Cross-validation detects and prevents overfitting and decreases in scores caused due to overfitting. The proposed model achieved an accuracy of more than 89% which is comparatively better than other models. Xiao et al. (2017) presented 89.70% and 97.30% accuracy for the fashion-MNIST dataset and MNIST dataset respectively using the SVC classifier.

Existing Systems

These days, an ever-increasing number of individuals use pictures to transmit data. It is additionally mainstream to separate critical data from pictures. Image Recognition is an imperative research area for its generally used applications. In general, in the field of pattern recognition, one of the difficult undertakings is the precise computerised recognition of human handwriting. Without a doubt, this is a very difficult issue because there is extensive diversity in handwriting from one individual to another individual. In spite of the fact that this difference does not make any issues to people, yet, anyway it is increasingly hard to instruct computers to interpret general handwriting. For the image recognition issue, for example, handwritten classification, it is essential to make out how information is depicted in images. Handwritten Recognition from the MNIST dataset is well known among scientists as by utilizing different classifiers for various parameters, the error rate has been decreased, for example, from linear classifier (1-layer NN) with 12% to 0.23% by a board of 35 convolution neural systems. The scope of this is to implement a Handwritten Digit Recognition framework and think about the diverse classifiers and different techniques by concentrating on how to accomplish close to human performance. For an undertaking of composing diverse digits (0-9) for various people the general issue confronted would be digit order issue and the closeness between the digits like 1 and 7, 5 and 6, 3 and 8, 9 and 8 and so forth. Additionally, individuals compose a similar digit from various perspectives, and the uniqueness and assortment in the handwriting of various people likewise impact the development and presence of the digits.

3. Proposed Work and Objectives

For the MNIST dataset, the best training accuracy and testing accuracy obtained are 98.86% and 98.96% respectively. The best result was obtained with 128 batch size, softmax activation function, adam optimizer, 0.25 dropout after each pooling layer, 10 epochs and 2x2 kernel size. For the Fashion-MNIST dataset, the best training accuracy and testing accuracy obtained is 92.02% and 92.76% respectively. The best result was obtained with 128 Batch size, softmax activation function, 0.25 dropout after each pooling layer, 10 epochs and 2x2 kernel size.

For the MNIST dataset, the best training accuracy and testing accuracy obtained are 99.60% and 99.37% respectively. The best result was obtained with 128 batch size, softmax activation

function, adam optimizer, 0.25 dropout after each pooling layer, 10 epochs and 2x2 kernel size. For the Fashion-MNIST dataset, the best training accuracy and testing accuracy obtained is 93.09% and 93.56% respectively. The best result was obtained with 128 Batch size, softmax activation function, adam optimizer, 0.25 dropout after each pooling layer, 50 epochs and 2x2 kernel size.

the optimal parameter is 128 batch size, 50 epochs, Softmax activation function, adam optimizer, 2x2 kernel size and 0.25 dropout after each pooling layer. For the MNIST dataset, the best-obtained training accuracy and testing accuracies are 99.02% and 99.03% respectively. For the Fashion-MNIST dataset, the best-obtained training accuracy and testing accuracies are 93.17% and 92.94% respectively.

Comparison of results- This subsection presents a comparison of obtained results of proposed CNN architectures. Further, the obtained results are compared with the literature. Results indicate that CNN is suitable to solve the MNIST dataset. The results obtained with all architectures are close to or better than 99%. Training and testing accuracy is the same.

Digit recognition is an excellent prototype problem for learning about neural networks and it gives a great way to develop more advanced techniques of deep learning. In future, we can develop character recognition and real-time person’s handwriting. Hnadwrittne digit recognition is the first step to the vast field of Artificial Intelligence and Computer Vision. As seen from the results of out project

`4. Methodology

CNN for image classification-

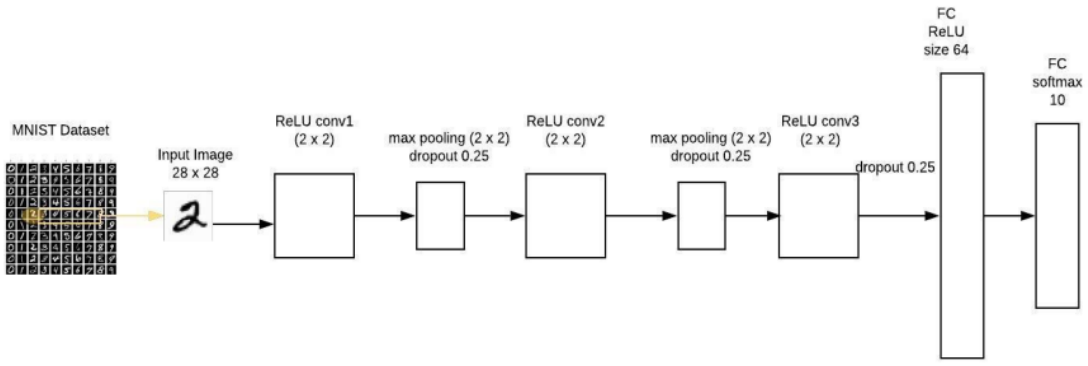


Fig. 1. CNN for image classification

There are three types of layers in CNN: convolutional layer, pooling layer, and fully connected layer. All these layers perform different task on the input data. In the convolutional layer, filters are applied to extract the features. Pooling layer perform the max pooling or average pooling, which extract the maximum value in the filter region or average value in filter region respectively. The fully connected layer aggregate the information from feature maps and generate the final classification.

Layers in our Neural Network-

1. The Convolution layer- The building block of CNN, which contains a set of filters (or kernels) parameters which are to be learned throughout the training.
2. The ReLU layer- Being one of the input layers, ReLU or the Rectified Linear Unit is the most common activation function, used to get a rectified linear map.
3. The Pooling layer- The pooling layer carries down the sampling operation to reduce the dimension of Rectified Feature Map to summarize the features.
4. The Fully Connected Layer- A fully connected layer is simply, feed-forward neural networks. Fully connected layers form the last few layers in the network.

a)Input Layer- Consists of CONVOLUTION LAYER(CONV2D) and RELU LAYER.

The input data is loaded and stored in the input layer. This layer describes the height, width and number of channels (RGB information) of the input image.

b) Hidden Layer- Consists of POOLING LAYER (MAX-POOLING 2D)

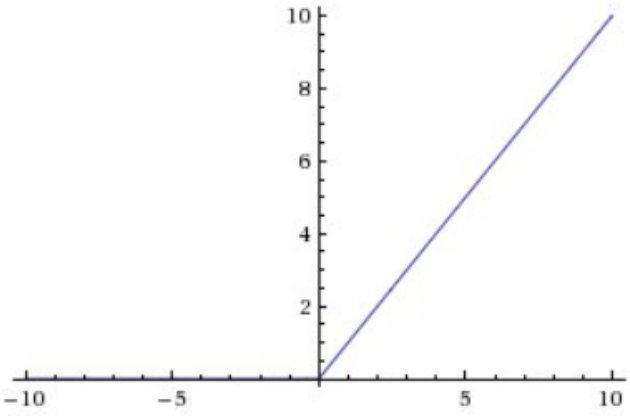
The hidden layers are the backbone of CNN architecture. They perform a feature extraction process where a series of convolution, pooling and activation functions are used. The distinguishable features of handwritten digits are detected at this stage.

- Convolutional Layer- The convolutional layer is the first layer placed above the input image. It is used for extracting the features of an image. The $n \times n$ input neurons of the input layer are convoluted with an $m \times m$ filter and in return deliver $(n - m + 1) \times (n - m + 1)$ as output. It introduces non-linearity through a neural activation function. The main contributors of the convolutional layer are receptive field, stride, dilation and padding, as described in the following paragraph. The output from the convolutional layer is a feature map, which is smaller than the input image. The output feature map contains more information on middle pixels and hence loses lots of information present on corners. The rows and the columns of zeros are added to the border of an image to prevent the shrinking of the feature map.

- ReLU layer- The most commonly used activation function in Neural Networks is ReLU or the Rectified Linear Unit. It helps the model account for the interaction effect. It performs element-wise operations in other words, the set of all negative values is converted to 0; this introduces non-linearity to the network. In simple terms, the ReLu helps in controlling the activation by not activating all the neurons at the same time.

It can be written as- $f(x)=\max(0,x)$

And Graphically it looks like this-



- Softmax Activation Function- The Softmax Activation is laso used in the model. The softmax is a combination of mutiple sigmoid function, it is mostly used in non-linear activation, basically, a sigmoid is an ‘s-shaped curve’ and softmax uses the sigmoid for the probability distribution of multiple values of 0’s and 1’s which are probabilities of data points of a particular class.

- Pooling Layer- A pooling layer is added between two convolutional layers to reduce the input dimensionality and hence to reduce the computational complexity. Pooling allows the selected values to be passed to the next layer while leaving the unnecessary values behind. The pooling layer also helps in feature selection and in controlling overfitting. The pooling operation is done independently. It works by extracting only one output value from the tiled non-overlapping sub-regions of the input images. The common types of pooling operations are max-pooling and avg-pooling (where max and avg represent maxima and average, respectively). The max-pooling operation is generally favourable in modern applications, because it takes the maximum values from each sub-region, keeping maximum information. This leads to faster convergence and better generalization.

- Flattening Layer- Flattening layer converts the resultant 2d array from pooled feature map into a single long and continuous linear vector. It basically converts spatial dimensions into channel dimensions. It is also considered the first fully connected layer.

- Dropout Layer- The Dropout is in the Fully Connected layer which randomly drops out some of the neurons to reduce the overfitting and coadaptation.

Overfitting: When we train our data without cleaning or there is a lot of noise in our data and our model starts to learn from that noise is called overfitting

Coadaptation: When there are multiple neurons, with the exact same feature and weight; such condition refers to coadaptation. this can happen when the connection weight of two different neurons is identical.

The Dropout randomly shuts down a fraction of the layer's neurons at each training step by zeroing out the weight of a neuron. The Dropout rate is given by r .

In our model, the Dropout rate is 0.25.

- Dense Layer- The Main fully Connected Layer or the Dense Layer is a layer that is deeply connected with its preceding layer which means the neurons of the layer are connected to every neuron of its preceding layer. In the background, the dense layer performs a matrix-vector multiplication. The values used in the matrix are actually parameters that can be trained and updated with the help of backpropagation. Backpropagation uses the gradient descent method to calculate the error in the accuracy of input and the weights of neurons. The output generated by the dense layer is an ‘m’ dimensional vector. Thus, the dense layer is basically used for changing the dimensions of the vector.

5. Mathematical Model

ReLu Function-

ReLU function is very simple, if it is greater than 0, it remains. Otherwise, it is 0, as shown in the figure given below.

Filters are tensor which keeps track of spatial information and learns to extract features like edge detection, smooth curve, etc of objects in something called a convolutional layer. The major part is to detect edges in the images and these are detected by the filters. It helps to filter out unwanted information to amplify images. There are high-pass filters where the changes occur in intensity very quickly like from black to white pixel and vice-versa.

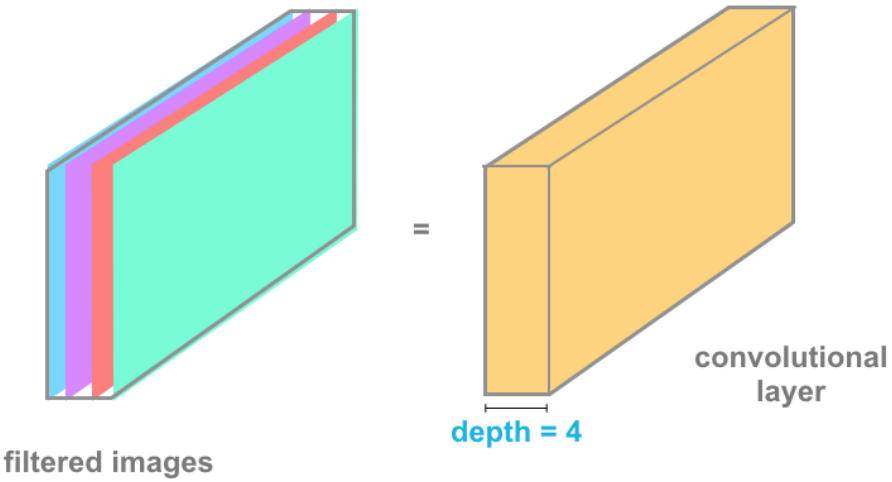


Image Credits: Deep Learning MachineLearning.ai

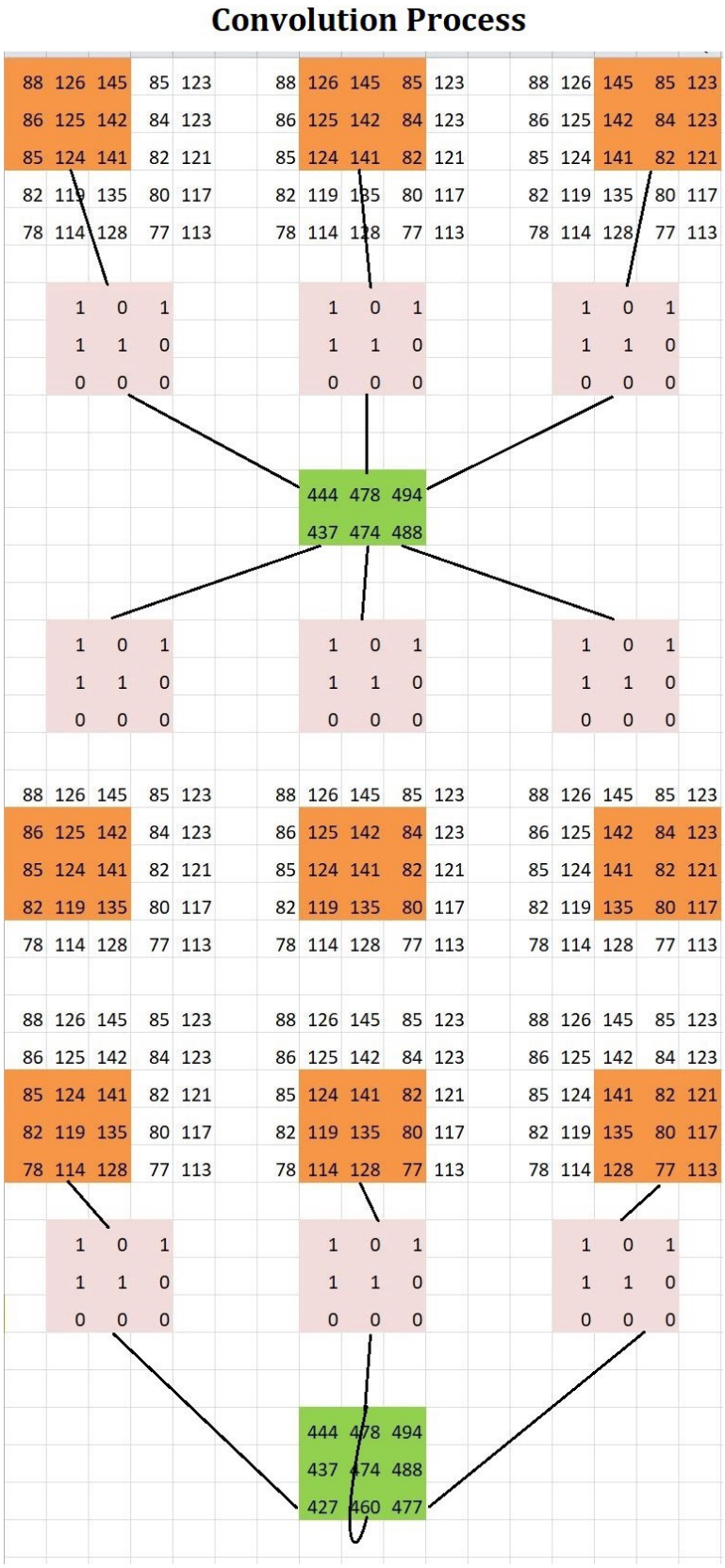
Consider any image, as you know every image represent some pixels in simple terms. We analyze the influence of nearby pixels in an image by using something called a filter (can be called as weights, kernels or features)

Let's consider an image of size 5X5 size and 3 filters (since each filter will be used for each color channel: RGB) of 3X3 size.

88	126	145	85	123												
86	125	142	84	123		1	0	1		1	0	1		1	0	0
85	124	141	82	121		1	1	0		0	1	0		0	1	0
82	119	135	80	117		0	0	0		1	1	0		0	1	1
78	114	128	77	113												
	5 * 5 Image					3 * 3 filter1				3 * 3 filter2				3 * 3 filter3		

Image Credits: Deep Learning MachineLearning.ai

For simplicity, we took 0 and 1 for filters, usually, these are continuous values.The filter convolute with the image to detect patterns and features.



Local Receptive Field

Filter

Output image

Ouput image value = LRF * Filter
(dot product of LRF and Filter)

Filter size = 3 X 3 --> 3
Input size = 5 X 5 --> 5
Stride = 1X1-->1 (1 cell move)
Padding = 0X0-->0 (No padding)

output size = (Input size - Filter size + 2 * Padding)* Stride + 1

output size = (I - F + 2P) * S + 1

output size = (5 - 3) * 1 + 1
output size = 3 -->3 X 3

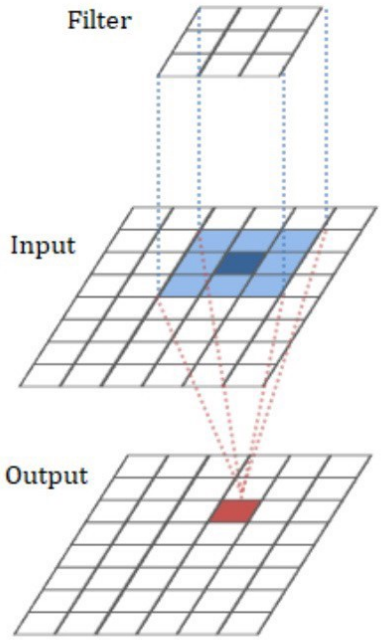


Image Credits: [Deep Learning MachineLearning.ai]

Then the convolution of 5 x 5 image matrix multiplies with 3 x 3 filter matrix which is called “Feature Map”. We apply the dot product to the scaler value and then move the filter by the stride over the entire image.

Sometimes filter does not fit perfectly fit the input image. Then there is a need to pad the image with zeros as shown below. This is called padding

Next, we need to reduce the size of images, if they are too large. Pooling layers section would reduce the number of parameters when the images are too large

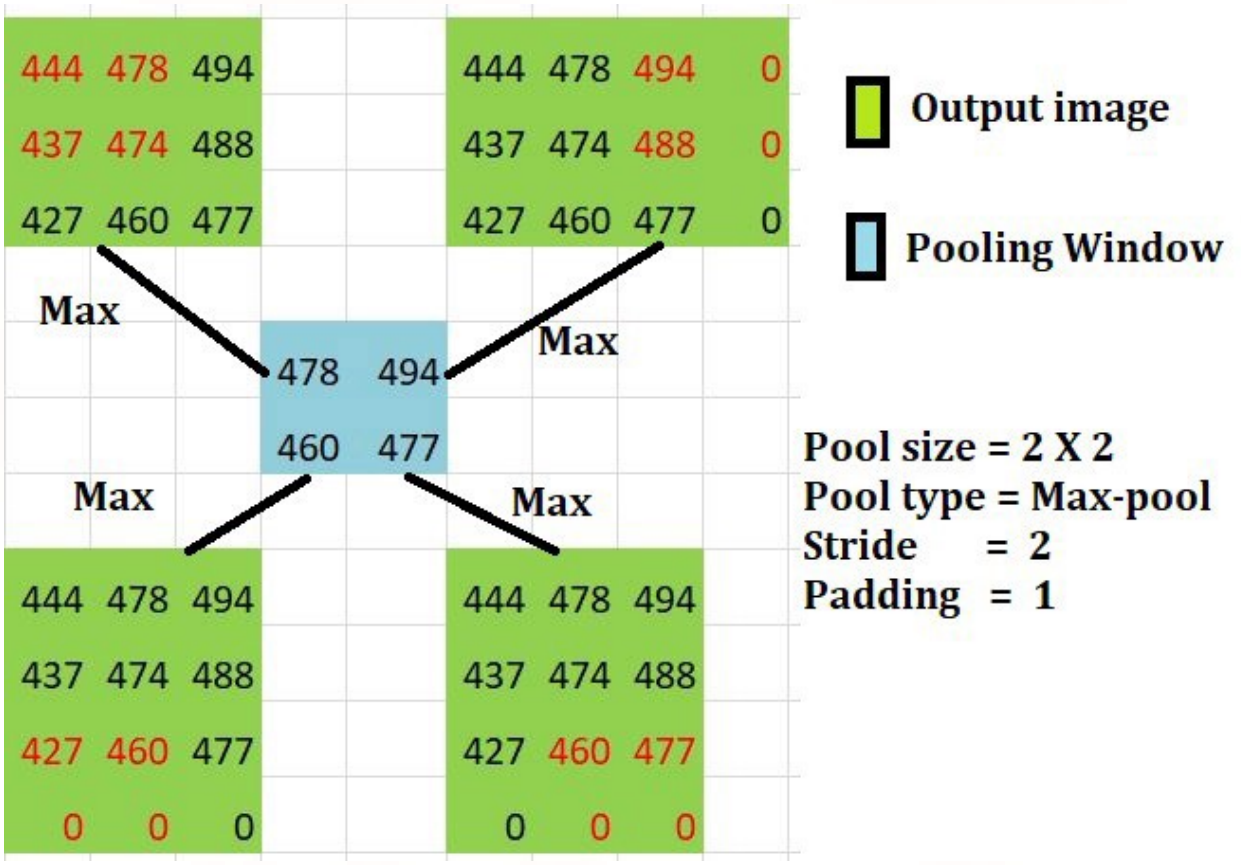


Image Credits: Deep Learning MachineLearning.ai

As shown in the above image, the padding is applied so that the filter perfectly fits the given image. Adding pooling layer then decrease the size of the image and hence decrease the complexity and computations.

Next Step, is Normalization. Usually, an activation function ReLu is used. ReLU stands for Rectified Linear Unit for a non-linear operation.

The output is $f(x) = \max(0,x)$.

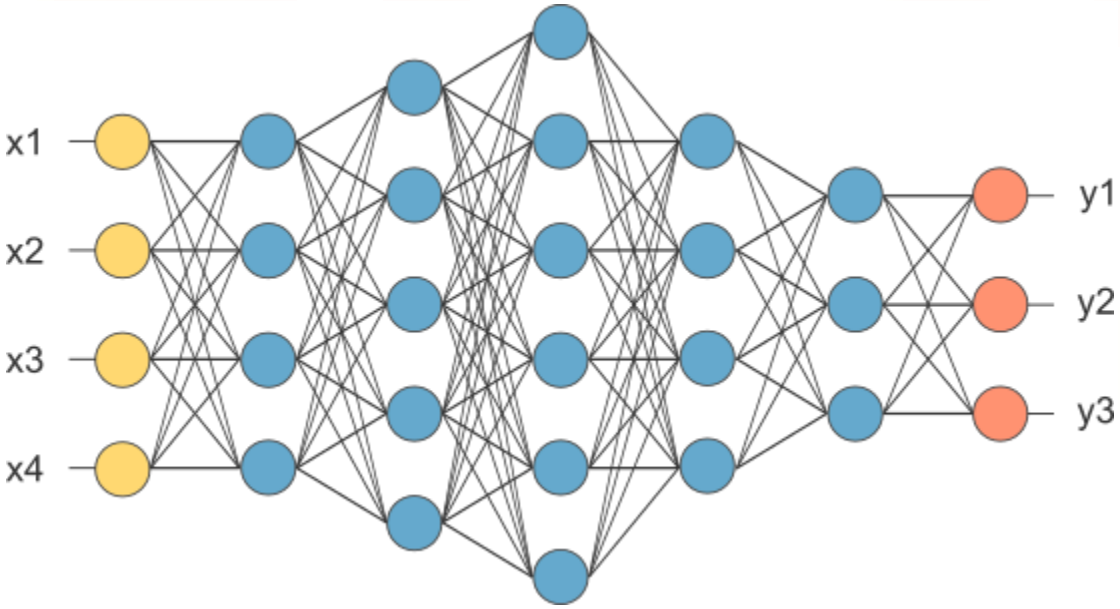
The purpose of ReLu is to add non-linearity to the convolutional network. In usual cases, the real-world data want our network to learn non-linear values.

A rectified linear unit has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. Here we are assuming that we have negative values since dealing with the real-world data. In case, if there is no negative value, you can skip this part.

$$RELU(x) = \begin{cases} 0 \ if \ x < 0 \\ x \ if \ x >= 0 \end{cases}$$

-478	494	0	494
460	-477	460	0

The final step is to flatten our matrix and feed the values to fully connected layer.



Next, we need to train the model in the same way, we train other neural networks. Using the ceratin number of epochs and then backpropagate to update weights and calculate the loss.

6. Desired Implications

Convolutional Neural Network is a type of deep learning neural network that is artificial. It is employed in computer vision and image recognition. This procedure includes the following steps:

- OCR and image recognition
- Detecting objects in self-driving cars
- Social media face recognition
- Image analysis in medicine

The term “convolutional” refers to a mathematical function that is created by integrating two different functions. It usually involves multiplying various elements to combine them into a coherent whole. Convolution describes how the shape of one function is influenced by another function. In other words, it is all about the relationships between elements and how they work together.

7. Conclusion

Here we demonstrate a model which can recognize handwritten digits using MNIST. There are many approaches to handwriting recognition. The highest accuracy is achieved from Convolutional Neural Network (CNN). When the images are trained with CNN, we will achieve good accuracy using more convolution layers and this is one of the successful methods for handwriting recognition and only disadvantage with this method is that the training time of the model is too high because a lot of image samples are included. Using the deep learning techniques, a high amount of accuracy can be obtained. This method focuses on which classifier works better by improving the accuracy of classification models by more than 99%. Using Keras as the backend and Tensorflow as the software, a CNN model is able to give an accuracy of about **98.72%**.

8. References

M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional neural networks,” arXiv:1311.2901, 2013

Gil Levi and Tal Hassner,” Age and Gender Classification using Convolutional Neural Networks”, Computer Vision and Pattern Recognition Workshops (CVPRW) IEEE, 2015

Aiming, He and Xiangyu, Zhang and Shaoqing, Ren and Jian Sun “Spatial pyramid pooling in deep convolutional networks for visual recognition” European Conference on Computer Vision, 2014.

Ciresan, Dan; Meier, Ueli; Schmidhuber, Jürgen (June 2012). "Multi-column deep neural networks for image classification". IEEE Conference on Computer Vision and Pattern Recognition (New York, NY: Institute of Electrical and Electronics Engineers (IEEE)). 2012

Yann LeCun, L´eon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedin x`gs of the IEEE, 86(11):2278–2324, 1998.

Qiuxing Chen; Lixiu Yao; Jie Yang “Short text classification based on LDA topic model”, IEEE

Vinoj Jayasundara; Sandaru Jayasekara; Hirunima Jayasekara; Jathushan Rajasegar;; “TextCaps: Handwritten Character Recognition With Very Small Datasets” ;; IEEE

Mahmoud M. Abu Ghosh; Ashraf Y. Maghari;; “A Comparative Study on Handwriting Digit Recognition Using Neural Networks” ;; IEEE

Yen-Min Su; Hsing-Wei Peng; Ko-Wei Huang; Chu-Sing Yang;; “Image processing technology for text rcognition” ;; IEEE

Amruta Pisa; S.D. Ruikar;; “Text detection and recognition in natural scene images”;; IEEE