Comparative study of performing features applied in CNN architectures

Sara LAROUI[1], Hicham OMARA[2], Mohamed LAZAAR[3], Oussama MAHBOUB[4]

[1,2,4]Abdelmalek Essaadi University, Tetuan – Morocco

[3]ENSIAS, Mohammed V University, Rabat, Morocco

saralaroui0@gmail.com, Hichamomara@gmail.com, lazaarmd@gmail.com, mahbouboussama@gmail.com

Abstract. Convolutional neural network (CNN), a class of artificial neural networks that has become dominant in various computer vision tasks, is attracting interest across a variety of domains, including Medical image analysis. CNN is designed to automatically and adaptively learn spatial hierarchies of features through back-propagation by using multiple building blocks, such as convolution layers, pooling layers, and fully connected layers. This paper presents an approach based on CNN for the classification of brain tumors, based on several characteristics that will be extracted automatically and some performing features that will be used in our CNN, This proposed approach provides efficient results at the level of automatic classification than the other usual methods.

Keywords: Deep networks, classification, convolutional neural network, brain tumor, Medical imaging.

1 Introduction

With advancements in computer vision (multimedia, computer-assisted editing, computer graphics) CNNs are finding their place in performing better, especially in medical imaging field. This progress is due to considerable work in this area and the availability of international image databases. CNN's applica tions in medical image analysis date back to the 1990s, when used in a variety of applications ranging from data analysis to decision systems [1]. Actually, by using CNNs the doctors are able to make more accurate and timely decision about patient's disease, its stage, and diagnosis.

We are interested in this article to problem of brain classification from MRI (Magnetic Resonance Imaging) [2]. There are many techniques available for diagnosis of brain tumor from the brain tissues, detection of brain tumor such as conventional radiology, ultrasonography, magnetic resonance imaging, computerized tomography and etc., but the process of diagnosing a number of CT-scan images manually becomes tiresome and also susceptible to error. Therefore, computer aided systems are used to assist the physicians as a second option to reduce the mistakes and errors, this raise the need of the automated com- puterized system, such as CNN [3]–[6]. In literature, there are many applications of CNN in the medical field and has played an important role for automated detection of cancerous cells from mammographic images, Computed topography (CT) and Magnetic resonance (MR) imaging are the most widely used radiographic techniques in diagnosis, clinical studies and treatment planning.

Many researchers has been applied CNN in medical image analysis to detect a brain tumor, such as: Verdhana et al. that were proposed in [3] a low-power architecture for edge detection to detect the bio- medical

ICCWCS 2019, April 24-25, Kenitra, Morocco
Copyright © 2019 EAI
DOI 10.4108/eai.24-4-2019.2284238

images are presented. They employed two major edge detection algorithms that are the Sobel and Prewitt algorithms using CMOS logic at 180 nm technology. The design is applied for various CT scan images of lungs, MR images of brain, X-ray images, etc. Another example, we see the work of Shuchao Pang et al. in [7]; they proposed a computational model for image classification based on end-to-end classifier using domain transferred deep convolutional neural networks or CNN. The proposed domain transferred deep CNN (DT-DCNN) showed significant increase in the accuracy while classifying the medical images with high precision. The new thought of the authors for simple data augmentation based on classification has improved the performance of this CNN based hybridized model. Another work has been proposed for classification of echocardiography by Xiaohong W.Gao et al. in [8]. They have pro- posed one hybridized network architecture with two CNNs along space and time directions both in spatial and temporal information fused to obtain classification scores from both networks. This spatial CNN work upon the normalized images to learn spatial information and pre-processing of the all the images are done in temporal CNN before training/learning starts and finally, to get the velocity and acceleration of images, the optical flow has been applied two times. Finally, we cite the proposition of Wei Shen et al. in [4], that they considered Multi-crop CNN (MC-CNN) to automatically extract the suspicious information based on an n multi-crop pooling strategy which crops different regions from convolutional feature maps without using segmentation to produce multi-sale features and applied on computationally effective single net- work. The goal of this work was to discover a set of discriminative features from hierarchical neural net- work to capture the suspicious information.

The main challenge addressed by the authors is to generate an image space including both the healthy tissues and specific suspicious nodules at different scales by ex- ploring the capabilities of deep learning architecture in an integrated fashion named as MC-CNN.

In this paper, we propose a method based on automatic two steps processing:

- Extraction of characteristics for the brain tumor.
- Choice of the best parameter that makes the neuron network performs.
- Classification of medical images containing brain tumors according to the characteristics extracted previously.

The rest of the paper has been organized as follows; section 2 and 3 describes the architecture and functionality of deep CNN. Section 4 describes some evaluation metrics for medical image analysis system that will be used in our experimentations, and a discussion of result obtained when applied the proposed model to classify brain tumors from DICOM images. Finally, we conclude the paper.

2 Convolutional neural networks (CNN)

A convolutional neural network (CNN) is a type of artificial neural networks, proposed by Yann Le Cun in 1988, most commonly applied to analyzing visual imagery[1]. Their multistage architectures are inspired from the science of visual cortex. One of the most popular and efficient uses of this architecture is image classification. The algorithm started by identifying low level features and then learns to recognize and combine these features to learn more complicated patterns. These different levels of features come from different layers of the network. And each layer has specific number of neurons and presented in 3 dimensions: height, width, depth.

The first part of a CNN is the convoluted part. It functions as a feature extractor of images. In this part, an image is passed through a succession of filters, or convolution kernels, creating new images called convolution maps. Some intermediate filters reduce the resolution of the image by a local maximum oper- ation.

A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalized layers. The description of each element is detailed in the rest of this part.

2.1 Convolution layer

A convolution layer is a fundamental component of the CNN architecture that used for feature ex- traction. Its purpose is to locate the presence of a set of features in the images received as input. It is com- posed of a stack of mathematical operations, such as convolution, a specialized type of linear operation.

In the case of digital images (whose pixel values are stored in a two-dimensional grid) a convolution- al filtering in the form of a small grid (called kernel) is applied at each image position to extract features. This procedure is repeated applying multiple kernels to form an arbitrary number of feature maps, which represent different characteristics of the input. The size of the kernels, number of kernels, padding, and stride are hyper-parameters that need to be set before the training process starts.

The individual grid outputs from the kernels are called feature maps (or activation maps). After each convolutional layer is applied, an activation function is applied to the 3-dimensional output. Typically, the activation function that is applied is called ReLU (or Rectified Linear Unit Layer) [8], [9]. ReLU gets rid of negative values, by setting any negative pixel values to 0. It is defined as follows:

$$ReLU(x) = max(x, 0)$$

2.2 Pooling layer

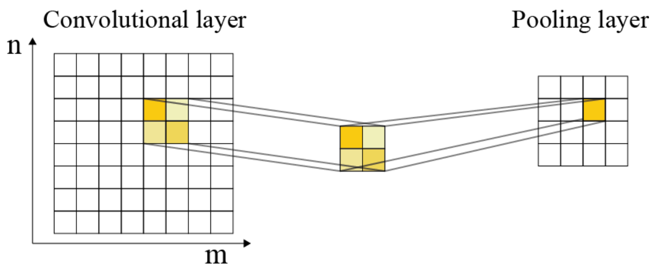

Fig.1. Convolution process

A pooling layer reduces the in-plane dimensionality of the feature maps in order to keep only the most relevant pixels. This process has a form of non-linear down-sampling. His main roles are to reduce progressively the spatial size of the representation, to reduce the number of parameters and amount of computation in the network, and hence to also control over fitting. It is of note that the filter size, stride, and padding are hyper parameters in pooling operations. The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 down samples at every depth slice in the input by 2 along both width and height, discarding 75% of the activations (as described in Fig. 1). In this case, every max op- eration is over 4 numbers. The depth dimension remains unchanged [10].

2.3 Max-pooling layer

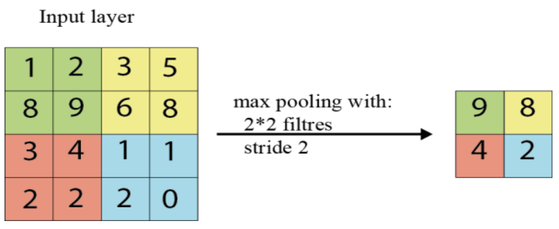

Fig. 2. A diagram showing a max pooling process.

The most popular form of pooling operation is max pooling (or down-sampling layer), which extracts patches from the input feature maps, outputs the maximum value in each patch, and discards all the other values. A max pooling with a filter of size 2 × 2 with a stride of 2 is commonly used in practice (Fig 2). This makes it possible to perform downscaling the dimension (height and width) of feature maps by a factor of 2, through pooling, without losing important information, however the depth dimension of fea- ture maps remains unchanged. Another example, in literature, consists to use the average pooling, which uses the average value from each of a cluster of neurons at the prior layer [5].

3 CNN training

The first step, in creating CNN, is to fix the architecture by fixing the number of chosen layers, their sizes and matrix operations that connect them. The training consists then, of optimizing the network's coefficients to minimize the output classification error. Similar to multilayer perceptron, CNNs are trained with the back-propagation algorithm by minimizing the following cost function with respect to the unknown weights W :

$$\zeta = -\frac{1}{|X|}\sum_{i}^{|X|} \ln\left(p\left(y_i \mid X_i\right)\right)$$

where $|X|$ denotes the number of training images, X_i denotes the ith training image with the corresponding label y_i, and $p\left(y_i \mid X_i\right)$ denotes the probability by which X_i is correctly classified. Stochastic gradient descent is commonly used for minimizing this cost function, where the cost over the entire training set is approximated with the cost over mini-batches of data. If $W_j(\mathrm{t})$ denotes the weights in j^{th} convolutional layer at iteration t, and L^ denotes the cost over a mini-batch of size N, then the updated weights in the next iteration is computed as follows:

$$\gamma^{t} = \gamma^{\left|\frac{tN}{|X|}\right|}$$

$$V_j(\mathrm{t+1}) = \mu V_j(\mathrm{t}) - \gamma^{t}\alpha_j \frac{d\zeta}{dW_j}$$

$$W_j(\mathrm{t+1}) = W_j(\mathrm{t}) + V_j(t+1)$$

where α_j is the learning rate of the j^{th} layer, μ is the momentum that indicates the contribution of the previous weight update in the current iteration, and γ is the scheduling rate that decreases learning rate α at the end of each epoch.

The iterative weight update begins with a set of randomly initialized weights. Specifically, before the commencement of the training phase, weights in each convolutional layer of a CNN are initialized by values randomly sampled from a normal distribution with a zero mean and small standard deviation. However, considering the large number of weights in a CNN and the limited availability of labelled data, the iterative weight update, starting with a random weight initialization, may lead to an undesirable local minimum for the cost function. Alternatively, the weights of the convolutional layers can be initialized with the weights of a pre-trained CNN with the same architecture. The pre-trained net is generated with a massive set of labelled data from a different application. Training a CNN from a set of pre-trained weights is called *fine tuning* and has been used successfully in several applications.

3.1 Proposed approach

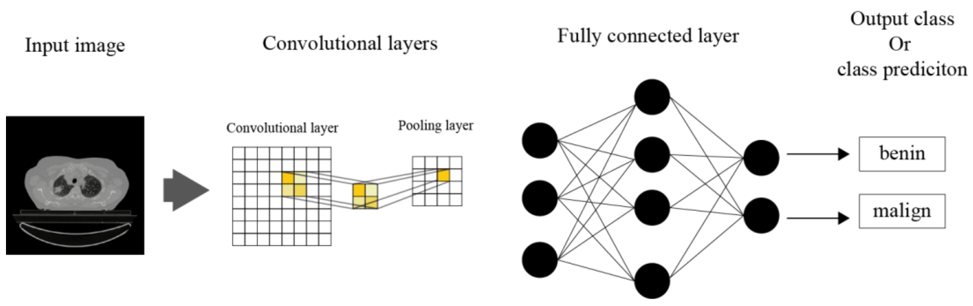

Fig.3. Global Architecture of the CNN used for medical image classification.

Fig.3 shows CNN architecture for classification of medical images. The network has convolutional, max pooling and fully connected layers. Each convolutional layer generates a feature map of different size and the pooling layers reduce the size of feature maps to be transferred to the following layers. The fully connected layers at the output produce the required class prediction. The number of parameters required to define a network depends upon the number of layers, neurons in each layer, the connection between neu- rons. The training phase of the network makes sure that the best possible weights are learned, that would give high performance for the problem at hand. The advancement in deep learning methods and computa- tional resources has inspired medical imaging researchers to incorporate deep learning in medical image analysis.

Some recent studies have shown that deep learning algorithms are successfully used for medical im- aging applications. A multi scale CNN based approach for automatic segmentation of MR Images is presented that classifies voxel into brain tissue classes. A total of five databases are used having T1 and T2 weighted images at a different postmenstrual age (PMA). A tri-planar CNN is used for segmentation of tibial cartilage in knee MRI images. Three 2D CNNs are integrated together for classification of a voxel in 3D images. A segmentation of brain tissue is presented through a CNN using multimodal MRI dataset by training the network on three patches that are extracted from the images. A deep learning based approach has been presented, in which the network uses a convolutional layer in place of fully connected layer to speed up the segmentation process. A cascaded architecture has been utilized, which concatenates the output of first network with the input of succeeding network. The network presented uses small kernels to classify pixels in MR image. The use of small kernels decreases network parameters, allowing to build deeper networks, without worrying about the dangers of over-fitting. Data augmentation and intensity normalization have been performed in pre-processing step to facilitate training process.

The approach proposed in this article avoids the design and the manual extraction of features. This approach is more typically faced with difficult problem such as:

- The design of features robust and easy to calculate.
- The evaluation of these characteristics and their relevance for the separation of classes,
- And the more general case of the selection of relevant characteristics.

Our approach based on convolutional neural networks avoids the delicate step related to the charac- teristics, and using multiple neural networks convolutional for the same problematic. The idea behind this is that each network will learn different characteristicsof the same input.

In order to have an efficient classification, we will create 3 CNN acting at different resolutions, choosing the most reliable activation function for this we use a medical images size to full resolution is from 512*512, the latter is successively divided by a factor √4 and √4² according to each axis so as to give images of sizes 256*256 and 64*64, respectively.

To learn features for our medical image classification task, we apply ConvNets combined with fine tuning technique. We used the pre-trained convolutional neural network AlexNet[11], [12] which is trained for 1000 possible categories on the large dataset ImageNet in the Large Scale Visual Recognition

Challenge (ILSVRC-2012), containing over then 1.2 million images. Results were remarkable and achieved a top- 5 error of 15.3%.

About our proposed algorithm, each image (or feature map) has dimensions W × H × D, where W is its width in pixels, H is its height in pixels and D is the number of channels (1 for a black and white im- age, 3 for an image in colors): The convolution layer has four hyper parameters:

- The number of filters K
- The size F of the filters: each filter has dimensions F × F × D pixels.
- The step S with which the window corresponding to the filter is dragged on the image. The zero-padding P is added to the input image of the layer a black outline of thickness P pix- els. Without this outline, the output dimensions are smaller. Thus, the more one stack of convolution layers with P = 0, the more the input image of the network shrinks. So we lose a lot of information quickly, which makes the task of extracting features difficult

And the pooling layer has only two hyper parameters:

- The size F of the cells: the image is divided into square cells of size F × F pixels
- Step S: The cells are separated from each other by S pixels

Our proposed algorithm is the gradient back propagation algorithm follows the Widrow-Hoff ap- proach. Define an error notion on an example and then calculate the contribution to this error of each of the synaptic weights. It is this second step that is not obvious, In order to be able to apply the gra- dient method, one needs to calculate derivatives.

4 Experiment

Medical imaging includes those processes that provide visual information of the human body. The purpose of medical imaging is to aid radiologists and clinicians to make the diagnostic and treatment process more efficient. Medical imaging is a predominant part of diagnosis and treatment of diseases and represents different imaging modalities such as computed tomography (CT), magnetic resonance imaging (MRI), positron emission tomography (PET), ultrasound, X-ray and hybrid modalities [7], [15]. These modali- ties play a vital role in the detection of anatomical and functional information about different body organs for diagnosis as well as research. Medical imaging is an essential aid in the modern healthcare and com- puter aided diagnosis (CADx) systems. Machine learning plays a vital role in CADx with its applications in medical image analysis, tissue or cancer detection and classification, image guided therapy, medical image annotation and medical image retrieval.

4.1 Dataset

We used in this article brain tumor image that working with DICOM files can be exchanged between two entities that are capable of receiving image and patient data in DICOM format.

Our work is about some of the brain MR images containing tumor taken for testing our proposed al- gorithm are shown. These images are taken from the cancer archive imaging. Our brain tumor dataset contains 160 contrast-enhanced images of 512*512 with three kinds of brain tumor of the type .dcm.

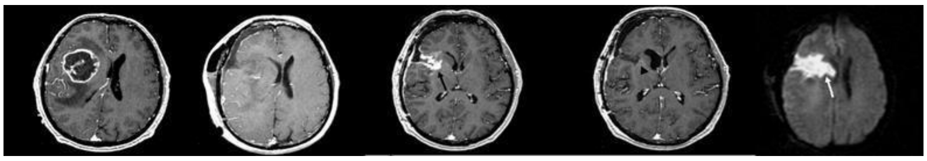

Fig. 5. Brain MR images containing tumor

4.2 Evaluation Metrics for Medical Image Analysis System

To evaluate proposed approach, we are used different key performance measures such as accuracy, precision, recall, sensitivity and specificity [13], [14]. Mathematically, these measures are calculated as,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + TN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Fscore = \frac{2 \times precision \times recall}{precision + recall}$$

where TP (True Positives) is correctly classified positive cases. TN (True Negative) is correctly classified negative cases. FP (False Positives) is incorrectly classified negative cases. FN (False Negative) is incor- rectly classified positive cases.

4.3 Result and discussion

In this section, we present experimental results about the effect of one of the hyper parameters of a deep neural network in our model that we will use it in our model proposed model.

The method we propose for our problematic is one of the best methods of deep learning; it is the convolutional neuron networks "CNN" have shown their effectiveness as a learning technique for the classification of data. They are indeed able to approximate complex non-linear functions in order to pro- cess large data. We are going to work on the classification of medical images, two approaches are merged within deep neural networks that realize both the automatic extraction of the characteristics and their classification, moreover, an approach is proposed using several networks depths at different resolutions, and whose outputs are merged, and to improve the performance of this model.

Our work is part of the analysis of medical images of MRI of the brain for the diagnosis of brain tu- mors (Glioma), the aim of our study is to provide physicians with a more precise automatic classification by distinguishing the types of these brain tumors, which prove to be very important during an early diag- nosis of cancers.

We will implement most of the characteristics extracted from each medical image of the base, and then we build networks of neurons with different resolutions by reducing the image in order to obtain more details by using one of the performing parameter obtained in those below tables (Table 2 -3-4). The last layer, in the case of supervised learning, contains as many neurons as desired, and a softmax activa- tion function is used to obtain probabilities of belonging to each class.

The function is:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad pour\,tout\ j \in \{1,\dots,K\}$$

That is, the component j of the vector $\sigma(z)$ is equal to the exponential of the component j of the vector z divided by the sum of the exponentials of all the components of z.

The following terms are consistently employed throughout this article so as to avoid confusion. A parameter in this article stands for a variable that is automatically learned during the training process. A hyper parameter refers to a variable that needs to be set before the training process starts. A kernel refers to the sets of learnable parameters applied in convolution operations. A weight is generally used inter- changeably with parameter; however, we tried to employ this term when referring to a parameter outside of convolution layers

Table 1. A list of parameters and hyper parameters in a convolutional neural network (CNN)

	Parameters	Hyperparameters
Convolution layer	Kernels	Kernel size, number of kernels, stride, padding, activation function
Pooling layer	None	Pooling method, filter size, stride, padding
Fully connected layer	Weights	Number of weights, activation function
Others		Model architecture, optimizer, learning rate, loss function, mini-batch size, epochs, regularization, weight initializa- tion, dataset splitting

Note that a parameter is a variable that is automatically optimized during the training process and a hyper parameter is a variable that needs to be set beforehand.

Activation Function:

The outputs of a linear operation such as convolution are then passed through a nonlinear activa- tion function. Although smooth nonlinear functions, such as sigmoid or hyperbolic tangent (tanh) func- tion, were used previously because they are mathematical representations of a biological neuron behavior, the most common nonlinear activation function used presently is the rectified linear unit (ReLU), which simply computes the function: $f(x)=\max(0,x)$.

Our problem type is classification, this why we implemented this experiment about extracting the most efficient hyper parameters as to use from our approach as a activation function and learning rate which aims to avoid over-learning and minimize the error rate, In this table below, we used the learning rate to 0.1 and fixing the epoch at 400 to get this results per number of hidden layers:

Table 2. The loss test according to the learning rate of the one hidden layer

Activation Function	Learning Rate	Test Loss	Training loss
Sigmoid	0,1	0.007	0.008
Tangh	0,1	0.003	0.003
ReLU	0,1	0.001	0.001

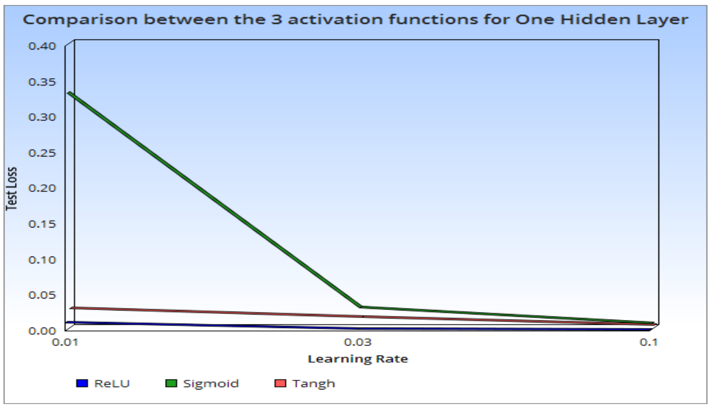

Fig. 6. Comparison between the 3 activation functions for one hidden layer

According to the table above, reproduces the results of the error test according to the learning rate in order to then know the most effective activation function when using a single masked layer, This **Fig.6** show more clearly the result of the table 2.

Now we will add the second hidden layer as you see in the table below:

Table 3. The loss test according to the learning rate of the two hidden layers

Activation Function	Learning Rate	Number of Neu- ron for The one hidden	Number of neu- rons for the two hidden	Test Loss	Training loss
Sigmoid	0,1	4	4	0.006	0.004
Tangh	0,1	4	4	0.000	0.000
ReLU	0,1	4	4	0.000	0.000

According to the table above, we added a second hidden layer and we obtain a more reduced error test than that of a single hidden layer on the table 2.

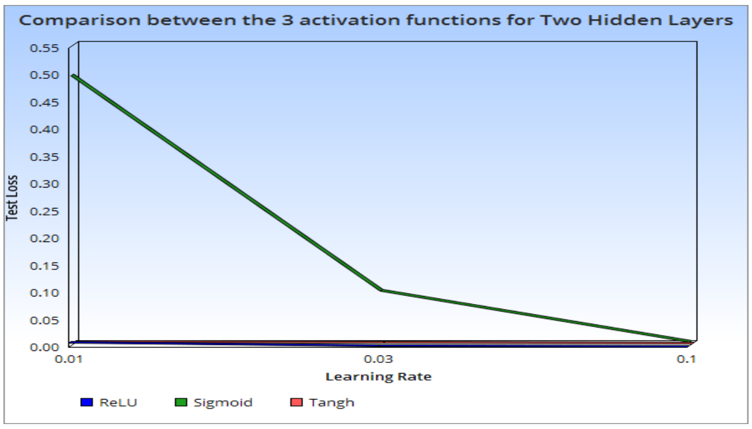

Fig. 7. Comparison between the 3 activation functions for two hidden layers

In order to obtain several precisions, we will add the third hidden layer as you see in the table below:

Table 4. The loss test according to the learning rate of the three hidden layers

Activation Function	Learning Rate	Number of Neu- ron for The one hidden	Number of neurons for the two hid- den	Number of neurons for the Three hidden	Test Loss	Training loss
Sigmoid	0,1	4	4	2	0.403	0.403
Tangh	0,1	4	4	2	0.000	0.000
ReLU	0,1	4	4	2	0.000	0.000

According to the table above, reproduces the results of the error test according to the learning rate in order to then know the most effective activation function when using a three masked layer is ReLU, This graph show more clearly the result of the table 4.

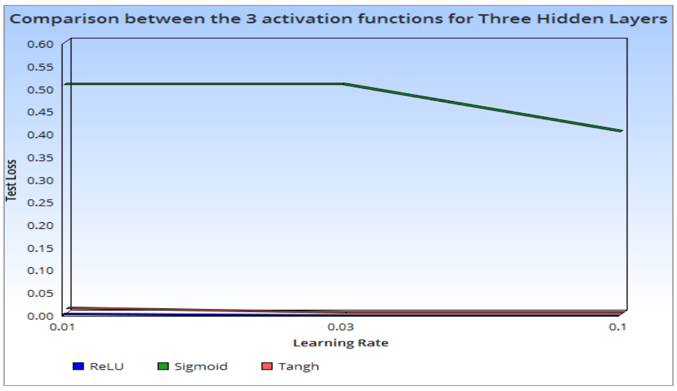

Fig. 8. Comparison between the 3 activation functions for three hidden layers

According to the results of Table 2,3 and 4, we find that the number of hidden layers play an im- portant role in the reduction of the error rate in terms of activation function that allows us to see that ReLU function shows suitable results compared to the other Sigmoid and Tangh activation functions.

Number of parameters:

Now, when talking about neural networks, it is nearly always the case that the network has far more parameters than training samples.

Theoretically, a simple two-layer neural network with 2n+d2n+d parameters is capable of perfectly fitting any dataset of N samples of dimension D.

The obvious benefit of having many parameters is that you can represent much more complicated functions than with fewer parameters. The relationships that neural networks model are often very com- plicated ones and using a small network (adapting the size of the network to the size of the training set, i.e. making your data look big just by using a small model) can lead to the problem when your network is too simple and unable to represent the desired mapping (high bias).

On the other hand, if you have many parameters, the network is flexible enough to represent the desired mapping and you can always employ stronger regularization to prevent over fitting, so this below table represents the parameter count per layer used:

Table 5. Increasing in number of parameters depending on the hidden layer

Layer	Filter size, stride	Output W×H×N	Parameter Count per Layer
Input	-	512 × 512 × 1	400
Conv	20 × 20, 2	247 × 247 × 1	
ReLU	-	247 × 247 × 1	
Max-pool	2 × 2, 1	246 × 246 × 1	
Conv	10 × 10, 2	119 × 119 × 1	500
ReLU	-	119 × 119 × 1	
Max-pool	4 × 4, 1	116 × 116 × 1	
Conv	5 × 5, 5	24 × 24 × 1	525
ReLU	-	24 × 24 × 1	
Max-pool	2 × 2, 2	12 × 12× 1	
Fully Connected FC	-	12 × 12× 1	21 549

According to this table 5, we notice an increase in the number of parameters in terms of the convolutional layers which layer allows a deeper network of neurons and returns better results.

The receptive field:

Another parameter and the most important concepts in Convolutional Neural Networks (CNNs) is the receptive field is defined as the region in the input space that a particular CNN's feature is looking at (i.e. be affected by). A receptive field of a feature can be described by its center location and its size. However, not all pixels in a receptive field is equally important to its corresponding CNN's feature. Within a receptive field, the closer a pixel to the center of the field, the more it contributes to the calculation of the output feature. Which means that a feature does not only look at a particular region (i.e. its receptive field) in the input image, but also focus exponentially more to the middle of that region?

We can compute the spatial size of the output volume as a function of the input volume size (W), the receptive field size of the Conv Layer neurons (F), the stride with which they are applied (S), and the amount of zero padding used (P) on the border. You can convince yourself that the correct formula for calculating how many neurons "fit" is given by:

$$(W-F+2P) / S+1$$

- Accepts a volume of size W1×H1×D1

- Requires four hyperparameters:

Number of filters K, Their spatial extent F, The stride S,

The amount of zero padding P.

For now, in this table below, we focus on calculating a particular receiver field based on the number of hidden layers:

Table 6. The performing activation function used

Layer #	Kernel size	Stride	Dilation	Padding	Input size	Output size	Receptive Field
1	20	2	1	0	512	247	20
2	10	2	1	0	247	119	38
3	5	5	1	0	119	23	54

As we can see from on - Table 6- The result is very clear with this graph :

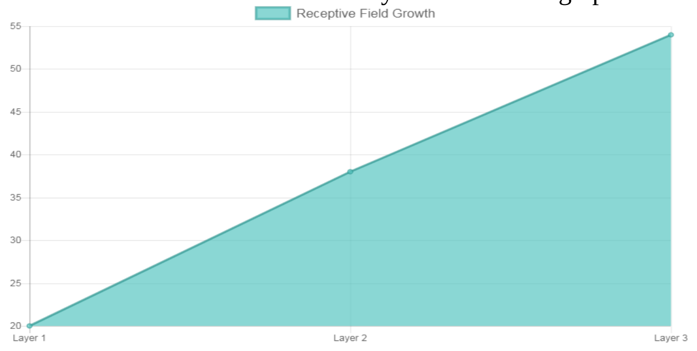

Fig. 9. Graph of Recetive Field Gtowth according to layers

Fig. 9, we can see the graph above this receptive field -in a convolutional neural network refers to the part of the image that is visible to one filter at a time- increases linearly as we stack more convolution- al layers or increases exponentially when we stack convolutions, When doing the forward pass, we can see that the central receptive field pixels can propagate their information to the output using many different paths, as they are part of multiple output unit's calculations, that help us to get an output more clearly to classify.

What is also very interesting is that the effective receptive field has a very close relationship with the foveal vision of the human eye, which produces the sharp central vision, effect of the high-density region of cone cells.

5 Conclusions

We presented in this article, the important parameters that we will need in our model of neural net- work, and we showed the results of each parameter so that we can then choose them and show the perfor- mance of our future model, and this results obtained show that all these parameters that we appreciated above, are important and robust parameters that allow us later to optimize our model with efficient results.

References

[1] A. Bhandare, M. Bhide, P. Gokhale, and R. Chandavarkar, "Applications of Convolutional Neural Networks," *Int. J. Comput. Sci. Inf. Technol. IJCSIT*, vol. 7, no. 5, pp. 2206–2215, 2016.

[2] B. H. Menze *et al.*, "The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS)," *IEEE Trans. Med. Imaging*, vol. 34, no. 10, pp. 1993–2024, Oct. 2015.

[3] M. Vardhana, N. Arunkumar, S. Lasrado, E. Abdulhay, and G. Ramirez-Gonzalez, "Convolutional neural net- work for bio-medical image segmentation with hardware acceleration," *Cogn. Syst. Res.*, vol. 50, pp. 10–14, Aug. 2018.

[4] W. Shen *et al.*, "Multi-crop Convolutional Neural Networks for lung nodule malignancy suspiciousness classifi- cation," *Pattern Recognit.*, vol. 61, pp. 663–673, Jan.2017.

[5] S. Mittal, "A survey of FPGA-based accelerators for convolutional neural networks," *Neural Comput. Appl.*, Oct. 2018.

[6] X. W. Gao, R. Hui, and Z. Tian, "Classification of CT brain images based on deep learning networks," *Comput. Methods Programs Biomed.*, vol. 138, pp. 49–56, Jan. 2017.

[7] S.-C. Pang, Z. Yu, and M. A. Orgun, "A novel end-to-end classifier using domain transferred deep convolutional neural networks for biomedical images," *Comput. Methods Programs Biomed.*, vol. 140, pp. 283–293, 2017.

[8] X. Gao, W. Li, M. Loomes, and L. Wang, "A Fused Deep Learning Architecture for Viewpoint Classification of Echocardiography," *Inf Fusion*, vol. 36, no. C, pp. 103–113, Jul. 2017.

[9] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, USA, 2010, pp. 807–814.

[10] D. Scherer, A. Müller, and S. Behnke, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," in *Artificial Neural Networks – ICANN 2010*, 2010, pp. 92–101.

[11] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Net- works," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, USA, 2012, pp. 1097–1105.

[13] O. Hicham, L. Mohamed, and T. Youness, "Improved self-organizing maps based on distance travelled by neu- rons," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 5, 2018.

[14] O. Hicham, M. Lazaar, et T. Youness, « Self-Organizing Maps and Principal Component Analysis to Improve Classification Accuracy », Res. J. Appl. Sci. Eng. Technol., vol. 15, no 5, p. 190-196, mai 2018.

[15] H. Omara, M. Lazaar, et Y. Tabii, « Effect of Feature Selection on Gene Expression Datasets Classification Accuracy », Int. J. Electr. Comput. Eng. IJECE, vol. 8, no 5, p. 3194-3203, oct. 2018.