Review: Statistical Analysis of Training Image & Label of CNN in Brain Tumor

S. Vaishali[(1)], *PG Student, SALITER, Opp. Science City, Ahmedabad,*

B.Jigar[(2)], *H.O.D. (EC/IC), SALITER, Opp. Science City, Ahmedabad,*

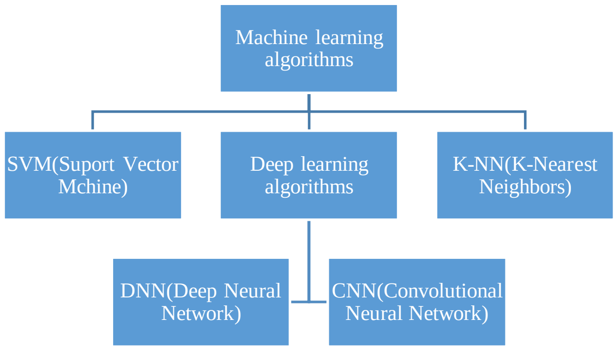
M.Vyoma[(3)], *Assistant Professor , SALITER, Opp. Science City, Ahmedabad*

***Abstract*—**Brain has chemical communication complex function that creates convolution in human body. While cells are in uncontrolled formation then as a result tumor comes. Identified techniques for tumor are DNN (Deep Neural Network), KNN (K-Nearest Neighbor), SVM (Support Vector Machine) and CNN (Convolutional Neural Network). DNNs are typically feed forward networks in which data flows from the input layer to the output layer without looping back. *K*-Nearest Neighbors algorithm (*k*-NN) is a non-parametric method used for classification and regression. SVM is a supervised learning method that looks at data and sorts it into one of two categories. An SVM outputs a map of the sorted data with the margins between the two as far apart as possible. SVMs are used in text categorization, image classification, handwriting recognition and in the sciences. CNNs are fundamental examples of deep learning, where a more sophisticated model pushes the evolution of artificial intelligence by offering systems that simulate different types of biological human brain activity. Above techniques are part of ML (Machine Learning) but CNN has effective application on MRI Images. Leading advantage of CNN is it identify the probability of the level of voltage in between two axioms. And maintain the probability balanced uncontrolled voltage if not then produced tumor. Level of voltage of 0.54mv to 0.7mv per micro second. The mathematical base of CNN is label and training images. In general, label is vector representation of CNN for finding probability and training image is separation of frames by that colors in input image. But disadvantage of CNN is, if noise is in first training image it will added up to last training image. That reduces the efficiency below 70% practically. In this paper, Author observed Techniques and mathematically try to improve its efficiency.

Index Terms*—CNN, DNN, KNN, SVM, DL, ML, Brain Tumor*

I. INTRODUCTION

TODAYS, imaging is a crucial part of the biomedical science and a mainstay of medical practice. No longer a scarce resource, imaging is often the first port of call for diagnosis, and now increasingly for treatment. Brain tumor is anomalous augmentation of tissues in the brain and that can disrupt proper brain function. Brain Magnetic Resonance Imaging (MRI) is one of the best imaging techniques that researchers relied on for detecting the brain tumors and modeling of the tumor progression in both the detection and the treatment phases. MRI images have a big impact in the automatic medical image analysis field for its ability to provide a lot of information about the brain structure and abnormalities within the brain tissues due to the high resolution of the images. The supervised machine learning algorithm is used to detect the brain MRI image[1].



In the last decade, Machine learning based algorithm have gained great attention. However, Support vector machine (SVM) method has the advantage of generalization and working in high dimensional feature space, it assumes that data are independently and identically distributed which is not appropriate for tasks such as segmenting medical images with inhomogeneity and noise and so it must be combined with other methods to consider spatial information and also have the advantages of such classifiers are that they are independent of the dimensionality of the feature space and that the results obtained are very accurate, although the training time is very high. In addition the problem of patient specific learning and storage must be added to the disadvantage of SVM-based methods. Also see that regular one class SVMs do not consider the negative information which cannot learn the feedback well[5][6][7].

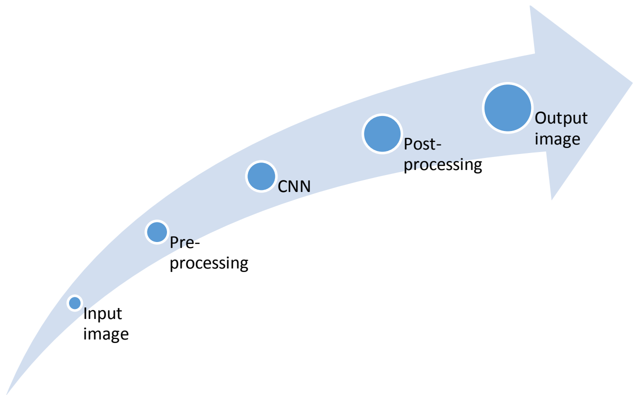
K-NN is easy to implement and debug, in situations where an clarification of the output of the classifier is useful, K-NN can be very effective if an analysis of the neighbors are useful as clarification and there are some noise reduction techniques that work only for K-NN that can be effective in improving the accuracy of the classifier. Disadvantages are K-NN can have poor run-time performance for larger training set so that all the work is done at run-time, K- NN is very sensitive to irrelevant or redundant features because all features contribute to the similarity and thus to the classification and this can be ameliorated by careful feature selection or feature weighting. On very difficult classification tasks, K-NN may be outperformed by more exotic techniques such as Support Vector Machines or Neural Networks. A simple classifier is the K-NN classifier, where each pixel or voxel is classified in the same class as the training data with the closest intensity. This method did not include any spatial regularization, so it is very sensitive to noise and in homogeneity of tumors. This system provides good result for small tumors but in the case of large deformations in the brain it will fail. This method also needs much calculation by repeating the classification and registration, therefore it is relatively slow. This algorithm fails in cases where the intensity distribution in the tumor is highly inhomogeneous and shows large spectral overlap with brain tissues. Other disadvantage of this K-NN algorithm include the dependence on the parameter K, large storage requirements (for training points),sensitivity to noise in the training data, and the undesirable behavior that can occur in cases where a class is under-represented in the training data, which make it unsuitable for brain tumor segmentation in MRI[11].

Deep Neural Network (DNN) is another DL architecture that is widely used for classification or regression with success in many areas. . A deep neural network (DNN) is an artificial neural network (ANN) with multiple hidden layers between the input and output layers. DNNs can model complex non-linear relationships. DNNs are typically feed forward networks in which data flows from the input layer to the output layer without looping back. DNNs are prone to over fitting because of the added layers of abstraction, which allows that models are rare dependencies in training data. Regularization methods can be applied during training to combat over fitting. Alternatively dropout regularization randomly omits units from the hidden layers during training. Data can be augmented via methods that smaller training sets can be increased in size to reduce the chances over fitting[1].

Other methods known as Deep Learning deal with representation learning by automatically learning an hierarchy of increasingly complex features directly from data. So, the focus is on designing architectures instead of developing homemade features, which may require specialized knowledge. Convolution is performed on the artificial neural networks (ANN) that gives us the Convolutional neural network. CNNs have been used to win several object recognition and biological image segmentation challenges. Since a CNN operates over patches using kernels, it has the advantages of taking context into account and being used with raw data. This paper; discussed the architecture of convolutional neural network. Neurons having weights and biases, which can be learned, forms a CNN. Three main layers used in building up the convolutional neural network architecture are : 1) Convolutional layer 2)Pooling layer and 3)Fully connected layer. A convolutional neural network (CNN) is so named as it contains one or more convolutional layers. In the input images, certain local features are detected through convolutional layers. There is a connection between every node of a convolutional layer and a subset of neurons which are connected spatially. This helps in detecting the local forms (structures) in the channels of input image. The weights on the connections are shared among the convolutional layer's nodes, to search for the similar local trait in the input channels. Every shared weight set is known as a kernel (convolution kernel). Across the input images, the local features (whose strength is visible in the feature map) to be detected are learned by convolutional layer having kernels[2]. A pooling layer is the next layer after convolution layer in CNN and main purpose of pooling layer decreasing the size of the representation spatially, which further helps reducing the parameter numbers and, network's computational complexity, and also helps in controlling the over fitting To reduce the feature maps' size, the max-pooling layer is used, which selects the maximum feature response among the local neighborhoods that can be overlapping or non-overlapping. This is done by disposing the precise location of maximum responses. A fully connected layer is the next layer after pooling layer. A CNN with fully connected layers is just as end-to-end learnable as a fully convolutional one. Even the decision-making layers at the end of the network are filters[2][3][9].

II. BRAIN TUMOR SEGMENTATION USING CONVOLUTIONAL NEURAL NETWORK

The method for segmentation of brain tumor in MRI images with the use of convolutional neural network. Pre-processing, classification using CNN and post-processing are the three main stages in the process of segmentation.



1) Pre-Processing: The bias field distortion can alter the MRI images which can vary the intensity of similar tissues in the complete image. Thus, the intensity normalization method is applied so as to form the similar ranges of intensity and contrast, across various patients. A set of intensity landmarks $I_L=\{pc_1, i_{p10}, i_{p20}, ..., i_{p90}, pc_2\}$ for every sequence are learned out of the training data set to perform normalization. In this, pc_1 and pc_2 are taken for each MRI sequence and at the 1^{st} percentile, intensity is represented as i_{p1}. After using this training data, next step in the intensity normalization method is transforming the original intensities linearly among two landmarks into the learned landmarks. Like this, every sequence can have the similar histogram across various subjects. After the normalization of the MRI images, mean value of intensity and standard deviation are calculated. These calculation over all the training patches which are taken for each image sequence. Then, to have mean to be zero and variance to be unit, the patches on each image sequence are normalized.

2) Convolutional Neural Network: The convolutional layers are applied to convolve an image (or signal) with kernels to result into the feature maps. Thus, the previous layer is connected though the kernel weights to an element in the feature map. During the training phase, the kernels' weights are adjusted to improve the input characteristics by using back propagation method. Since all the units of the similar feature maps share the kernels, the convolutional layers will be having lesser weights for training than the FC layers which are dense, and thus making CNN less susceptible to over fitting and easy to train. The similar feature can be observed irrespective of the location, making it translation invariance, as over all the image, the similar kernel is convolved. The information of the neighborhood is extracted by using kernels. On each neural unit's output, a non-linear activation function is implemented.

3) Post-Processing:

In this step, the clusters which are small are to be classified as tumor. Then, the Volumetric constrains are imposed by removing the clusters obtained in the segmentation by the CNN that are lesser than a preset threshold[2][4][8].

III. MATH

Method of Mathematical Analysis for detection of Brain Tumor is nothing but probability. And if any unbalancing or blunder convolution while process of communication in brain then result is Tumor. Back-propagation is a method used to calculate a slope that is needed in the calculation of the weights to be used in the network. Back-propagation is a automatic differentiation technique. Back-propagation is used with the slope descent optimization algorithm to regulate the weight of neurons by calculating the slope of the loss function. For the reason that the error is calculated at the output as well as distributed back through the network layers. To identify with the mathematical derivation of the Back-propagation algorithm, the relationship between the actual output of a neuron and the correct output for a particular training case. Examine a simple neural network with two input units, one output unit and no hidden units and every neuron uses a linear output that is the weighted sum of its input. Consider the neural network set of tipple(x_1,x_2,t), where x_1 and x_2 are the inputs to the network and *t* is the correct output. Difference between the expected output *t* and the actual output *y* is the squared error measure i.e.

$$E=(t-y)^2 \text{ ,where E is error.}$$

If input x_1 and x_2 are 1 and the correct output t is 0. Than the actual output *y* is plotted against the error E the result will be parabola. The minimum of the parabola corresponds to the output *y* which minimizes the error *E*. A single training case, the error will be zero and the network output *y* that exactly to the expected output *t*. For that reason, the problem of mapping inputs to outputs can be reduced to an optimization problem of finding a function that will produce the least error.

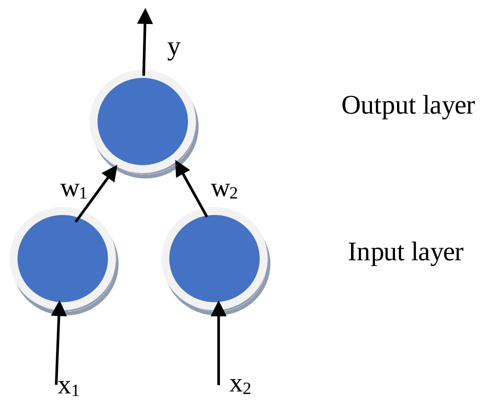


Fig.1 A neural network with two input units and one output unit

But, the output of a neuron depends on the weighted sum of all its inputs:

$$y=x_1w_1+x_2w_2$$

where w_1 and w_2 are the weights on the connection from the input units to the output unit. Therefore, the error also depends on the incoming weights to the neuron. In back-propagation the derivative of the squared error function with respect to the weights of the network. Presuming one output neuron, the squared error function is:

$$E=\frac{1}{2}(t-y)^2$$

where, E is the squared error, t is the target output for a training sample, y is the actual output of the neuron, $\frac{1}{2}$ is constant. For each neuron i its output O_i is defined as,

$$O_i=\varphi(N_i)=\varphi\left(\sum_{k=1}^{n}W_{ki}O_k\right)$$

The input N_i to a neuron is the weighted sum of outputs O_k of previous neurons. If the neuron is in the first layer after the input layer, the O_k of the input layer are simply the inputs x_k to the network. The number of input units to the neuron is n. The variable W_{ki} denotes the weight between neurons "k" and "i".The activation function φ is non-linear and differentiable.

i= O/P Neuron

i= Inner Neuron

$$\varphi(z) = \frac{1}{1+e^{-z}}$$

The derivation of $\varphi(z)$ is,

$$\frac{d\varphi}{dz}(z) = \varphi(z)\left(1-\varphi(z)\right)$$

Calculate the partial derivative of the error with respect to W_{ji}.

$$\frac{\partial E}{\partial W_{ji}} = \frac{\partial E}{\partial O_i}\frac{\partial O_i}{\partial N_i}\frac{\partial N_i}{\partial W_{ji}}$$

But the sum of N_i depends on W_{ji} so,

$$\frac{\partial N_i}{\partial W_{ji}} = \frac{\partial}{\partial W_{ji}}\left(\sum_{k=1}^{n} W_{ki}O_k\right) = \frac{\partial}{\partial W_{ji}} W_{ji}O_j = O_j$$

If the neuron is in the upper most layer after the input layer O_j is x_j The partial derivative of the output of neuron O_j with respect to partial derivative of N_i.

$$\frac{\partial O_i}{\partial N_i} = \frac{\partial}{\partial N_i}\varphi(N_i) = \varphi(N_i)\left(1-\varphi(N_i)\right)$$

For that purpose back-propagation requires the activation function to be differentiable. The first factor is uncomplicated to evaluate if the neuron is in the output layer, because then $O_j = y$ and

$$\frac{\partial E}{\partial O_i} = \frac{\partial E}{\partial y} = \frac{\partial}{\partial y}\frac{1}{2}(t-y)^2 = y-t$$

if "i" is in an arbitrary inner layer of the network, so the derivative E with respect to O_i is lesser. Taking into consideration E as a function of the inputs of all neurons $L = u, v, \ldots, w$ in receipt of input from neuron "i",

$$\frac{\partial E(O_i)}{\partial O_i} = \frac{\partial E\left(N_{u,} N_{v,\ldots,} N_w\right)}{\partial O_i}$$

The total derivative with respect to O_i,

$$\frac{\partial E}{\partial O_i} = \sum_{l\in L}\left(\frac{\partial E}{\partial N_l}\frac{\partial N_l}{\partial O_i}\right) = \sum_{l\in L}\left(\frac{\partial E}{\partial O_l}\frac{\partial O_l}{\partial N_l}W_{il}\right)$$

So, the derivative with respect to O_i can be calculated if all the derivatives with respect to the outputs O_l of the next layer the one is closer to the output neuron are known.

$$\frac{\partial E}{\partial W_{ji}} = \delta_i O_j$$

where,

$$\delta_i = \frac{\partial E}{\partial O_i}\frac{\partial O_i}{\partial N_i} = \begin{cases}(O_i - t_i)O_i(1-O_i) \\ \left(\sum_{l\in L} W_{il}\delta_l\right)O_i(1-O_i)\end{cases}$$

To update the weight W_{ji} using gradient descent, one must choose a learning rate, $\eta > 0$. The change in weight needs to reflect the impact on E of an increase or decrease in W_{ji}. If $\frac{\partial E}{\partial W_{ji}} > 0$ an increase W_{ji} in increases E; On the other hand, if $\frac{\partial E}{\partial W_{ji}} < 0$ an increase W_{ji} in decreases E. The new ΔW_{ji} is added to the old weight, and the product of the learning rate and the gradient, multiplied -1 by guarantees that W_{ji} changes in a way that always decreases E. In other words, in the equation immediately below, $-\eta\frac{\partial E}{\partial W_{ji}}$ always changes W_{ji} in such a way that is E decreased.

$$\Delta W_{ji} = -\eta\frac{\partial E}{\partial W_{ji}} = -\eta\delta_i O_j$$

Further, the following cost function is minimized which is done with respect to the weights W that are unknown.

$$L = -\frac{1}{|X|}\sum_{i}^{|X|}\ln\left(p\left(y^i \middle| X^i\right)\right) - \eta\delta_i O_j$$

... **By Author (1)**

Where $|X|$ is represented number of training images, the i[th] training image represents X^i and y^i is the label, and the probability by which X^i is correctly classified is given by

$p\left(y^i \middle| X^i\right)$. The weights in l[st] convolutional layer are W_l^t , at the iteration *t*, and the cost over a small batch of size *N* is represented by *L*.

IV. CONCLUSION

Brain tumor segmentation has a very important role of diagnostic procedures. With accurate segmentation, Machine learning algorithms are used. In this paper, a CNN architecture for the brain tumor are present. But if noise added in the first training image it will add up to last image, that can be reduce the efficiency of the image. For improve the efficiency, Gaussian filter is one of the parameter to stabilized the noise.

REFERENCES

1. H.Mohsen, El-Sayed A. El-Dahshan, El-Sayed M. El-Horbaty, Abdel-Badeeh M. Salem, *"Classification using deep learning neural networks for brain tumors",* Future computing and informatics journal , vol. 3, no. 1, pp. 68-71, Jun. 2018.
2. Anil Singh Parihar, *"A Study on Brain Tumor Segmentation Using Convolution Neural Network"*, In 2017 International Conference on Inventive Computing and Informatics (ICICI), Coimbatore, India, 23-24 Nov. 2017, ISBN: 978-1-5386-4032-6
3. Lina Chato, Shahram Latifi, *"Machine Learning and Deep Learning Techniques to Predict Overall Survival of Brain Tumor Patients using MRI Images"*, in IEEE 17th International Conference on Bioinformatics and Bioengineering, Washington, DC, USA., 23-25 Oct. 2017, ISSN: 2471-7819.
4. Sérgio Pereira, Adriano Pinto, Victor Alves, and Carlos A. Silva, *"Brain Tumor Segmentation Using Convolutional Neural Networks in MRI Images", IEEE TRANSACTIONS ON MEDICAL IMAGING*, VOL. 35, NO. 5, MAY 2016.
5. Zhe Xiao, Ruohan Huang, Yi Ding, Tian Lan, RongFeng Dong, Zhiguang Qin, Xinjie Zhang, Wei Wang, *"A deep learning-based segmentation method for brain tumor in MR images",* 2016 IEEE 6th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS), Atlanta, GA, USA., 13-15 Oct. 2016, ISSN: 2473-4659
6. Ayachi R., Ben Amor N., *"Brain Tumor Segmentation Using Support Vector Machines"*, European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty ,vol. 5590, pp 736-747, 2009. ISBN 978-3-642-02905-9
7. T. Sathies Kumar, K. Rashmi, Sreevidhya Ramadoss, L.K.Sandhya, T.J. Sangeetha, *"Brain Tumor Detection Using SVM Classifier",* 2017 IEEE 3rd International Conference on Sensing, Signal Processing and Security (ICSSS), Chennai, India, 4-5 May 2017, ISBN : 978-1-5090-4929-5.
8. Saddam Hussain , Syed Muhammad Anwar, Muhammad Majid, *" Brain Tumor Segmentation using Cascaded Deep Convolutional Neural Network",* 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Seogwipo , South Korea, 11-15 July 2017, ISBN :978-1-5090-2809-2.
9. Jose Bernal, Kaisar Kushibar, Daniel S. Asfaw, Sergi Valverde, Arnau Oliver, Robert Mart´, Xavier Llad´o, *"Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review",* (https://www.researchgate.net/publication/321744786)
10. Yan Xu, Zhipeng Jia, Yuqing Ai, Fang Zhang, Maode Lai, Eric I-Chao Chang, *"Deep convolutional activation features for large scale Brain Tumor histopathology image classification and segmentation",* 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, QLD, Australia, 19-24 April 2015, ISBN : 978-1-4673-6997-8.
11. Mohammad Havaei, Pierre-Marc Jodoin, Hugo Larochelle, *"Efficient interactive brain tumor segmentation as within-brain kNN classification",* 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24-28 Aug. 2014, ISBN : 978-1-4799-5209-0.