International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 121-128 | Cosmos Impact Factor (Germany): 5.195
Received: 23.03.2018 Published : 22.04.2018

Classification and Modelling Images Using Deep Learning Methods

Y. Pavan Kumar Reddy[1] and Dr.K.Fayaz[2]
Research Scholar, RayalaSeemaUniversity[1], SriKrishnadevarayaUniversity[2], India.

Abstract: Deep learning is a branch of machine learning technique, which contains multiple hidden layers in between input and output layers of neural network. There are levels of abstraction and representation in deep architecture, which allows modelling high dimensional, no-linear data such as images, audio and text. Deep learning methods are useful to solve various real world classification, regression and prediction problems. Each hidden layers represent an abstract feature of the model trained on the network. Hidden layers are trained using de-noising auto encoder or sparse auto encoder for better representation of futures. Both supervised and unsupervised methods are used to train the network. The data set is divided into three parts: The training, the validation and test set. The training set is used to train the methods with different hyper parameter settings. The validation set is used to compare the hyper-parameter settings in terms of predictive performance of every method. Finally accuracy of model is tested using test data set.

Keywords – Deep learning, hyper parameter, Machine Learning, Neural network, Encoders

I. INTRODUCTION

In machine learning, the typical goal is to find a mapping from input patterns to an output value [13]. For instance, we have images of objects as input data (represented by pixel intensity values) and correct labels (one for every type of object) as corresponding output values. Then the aim of the algorithm is to learn this mapping (from the samples to the output value), to be able to predict the correct output of a new input sample. There are different machine learning settings [13], such as supervised learning, unsupervised learning and Semi-Supervised learning.

1.1 Hyper-Parameters

Bengio defines the problem in the following way: ”We define a hyper-parameter for a learning algorithm A as a value to be selected prior to the actual application of A to the data, a value that is not directly selected by the learning algorithm itself.” [9, p.7]. Choosing hyper-parameters is therefore formally equivalent to model-selection, i.e. choosing the most appropriate value/algorithm in the given set of values/algorithms [9]. Hyper-parameters can be continuous (e.g. learning rate) or discrete (e.g. the number of neurons in one layer) and can be seen as an outside control button [9]. To point out the difference between hyper-parameters and parameters, we consider an example where we would like to train a polynomial function to fit a specific function. The polynomial function takes the form 1 Where M is the order of the

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 121-128 | Cosmos Impact Factor (Germany): 5.195
Received: 23.03.2018 Published : 22.04.2018

polynomial and wi are the polynomial coefficients (w = w0, w1… wm). M has to be chosen 'by hand' and before the training of the polynomial is started. In contrast, the polynomial coefficients are adapted within the training procedure itself. As a result M can be seen as a hyper-parameter, whereas the polynomial coefficients are parameters. For all learning algorithms which are examined in this work, hyper-parameters have to be adjusted.

1.2 The Role of Pre-processing

As stated in Coates [21], pre-processing is a common step in machine learning. There are several different methods, like PCA and whitening, which can be used, depending on the type of input data. This section points out the basic ideas of pre-processing and helps to understand why pre-processing can help to improve the performance of machine learning approaches.
Normalization of Data: A simple pre-processing step in machine learning consists of computing the mean value (of all dimensions) of an example and subtracting this mean value from every dimension of that example [21]. For images, this can be seen as normalization of the brightness.

2. RELATED WORK

2.1 K-Nearest Neighbour Approach for Classification

The K-nearest neighbour approach is one of the approaches for classification, which is very intuitive and simple to understand, but works very well in practice [26]. In the classification

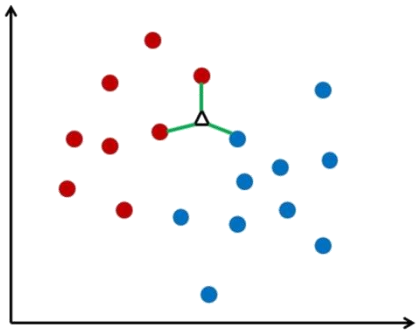

Figure 2: Example for a K-nearest neighbour classifier, where K = 3. The black triangle is the new data point, which is classified according to the majority class membership of the 3 closest training data points. In this example, the back triangle would be classified as red.

The classification of a new point can now be divided into two steps: First, the k nearest neighbours have to be determined. There are different techniques to compute the distance between two examples (e.g. Euclidean distance, Manhattan distance) and the performance of this algorithm depends on the used distance metric to identify the nearest neighbours [2, 8]. Secondly, the class of the new data point has to be determined, corresponding to the labels of the k nearest neighbours, which have been identified in the previous step [13, 6, 3]. There are various ways to calculate the final class from the k nearest neighbours, beside the simple

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 121-128 | Cosmos Impact Factor (Germany): 5.195
Received: 23.03.2018 Published : 22.04.2018

majority vote [2]. For example, Cunningham et al. [2] introduce a technique, where the votes of nearer neighbours have more influence on the final classification than neighbours which are further away. In contrast to SVMs or Random Forests, the K-nearest neighbour algorithm has no, or only a minimal, explicit training phase. Instead, the whole training set is used for the prediction of a new data sample (in contrast to the 'sparse representation' in SVM, where only the support vectors are used for prediction) [13, 26, 83].

2.2 Support Vector Machines for Classification

Support Vector Machines (SVMs) were first introduced in 1992 by Boser et al. [14] and became popular for solving problems in classification, regression and novelty detection [11]. An important property of SVMs is that the learning of the model parameters involves the optimization of a convex function. First, we consider the two-class classification problem, where we assume that the classes are linearly separable. The training data set comprises the input vectors {x1,.., xn} with the corresponding target values {y,…; yn}, where yi Є {- 1, 1} and new data points are classified according to the sign of sign [wT x + b], where wT x + b = 0 denotes the decision hyperplane (w determines the orientation of the plane, and b the offset of the plane from the origin) [11]. If we look at Figure 3(a), we can see that there are many possible solutions for the decision boundary. The first key concept of support vector machines is to choose the decision hyperplane with the maximum margin, where the margin is the smallest distance between the plane and any of the samples, as illustrated in Figure 3(b) [13]. In this example, the solution depends only on two points, which are marked in Figure 3(b). These two points (a subset of the training data set), which determine the location of the boundary, are called support vectors. Mathematically, the maximum margin solution can be formulated as a constrained optimization problem:

$$\textbf{minimize} \quad \frac{1}{2}||\mathbf{w}||^2 \qquad (2)$$

$$\textbf{subject to} \quad y_i(\mathbf{w}^t \mathbf{x}_i + \mathbf{b}) \geq 1 \quad \forall_i \qquad (3)$$

Where minimizing $\frac{1}{2}||\mathbf{w}||^2$ is equal to maximizing the margin (given by $\frac{2}{||\mathbf{w}||}$), and the constraint ensure all points are classified correctly(one constraint for each point [13].

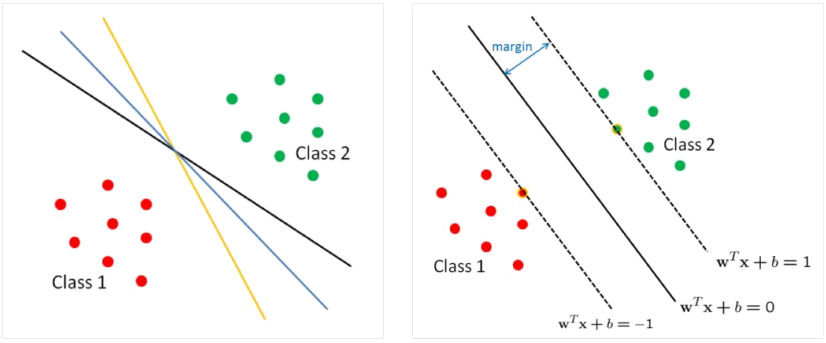

Figure 3: (a) Examples for possible solutions of the two-class classification problem (linearly separable). (b) Illustration of the maximum margin

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 121-128 | Cosmos Impact Factor (Germany): 5.195
Received: 23.03.2018 Published : 22.04.2018

3. PROPOSED DEEP NETWORKS

3.1Auto-Encoders

In this approach each layer of neural network is pre-trained using an unsupervised learning algorithm, instead of using random initialization. The idea of this approach is to train each layer unsupervised one after another, where the output of a trained layer is used as in-put of the subsequent layer in the training procedure [7]. This idea of pre-training is known as greedy layer-wise unsupervised learning. After this unsupervised learning procedure, the network is trained using the conventional supervised learning algorithm, where this supervised training step is also known as fine-tuning.

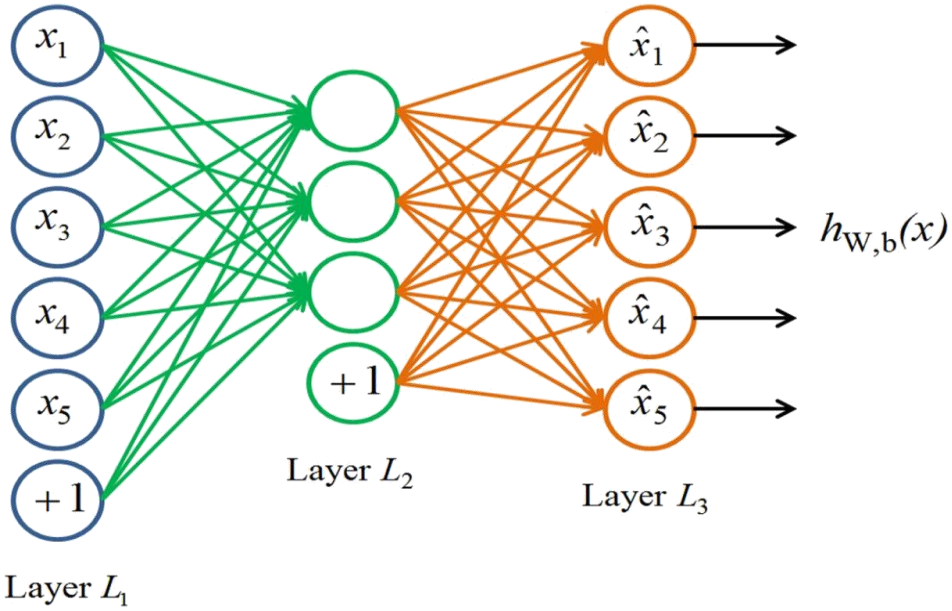

Figure 4: Basic structure of an auto-encoder. In the hidden layer L2 the activation function f( ) and in the output layer L3 the activation function g( ) is used, as described in the text. The first part of the network is called encoder (drawn in green) and the second part of the auto-encoder is known as decoder (highlighted in orange).

3.2 Stacked Auto encoders

In this work, stacked auto-encoders are used to implement the idea of greedy layer-wise unsupervised learning, also known as pre-training. The main idea of this pre-training stage is to shift the weights of the network in the" right direction", before applying the conventional super-vised learning algorithm, to find a better local optimum. At first, this auto-encoder is trained using error back propagation and gradient descent to optimize the sparse error function Esparse( ), without the need for labels. Afterwards, the last layer of this trained network (the decoder) is removed.

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 121-128 | Cosmos Impact Factor (Germany): 5.195
Received: 23.03.2018 Published : 22.04.2018

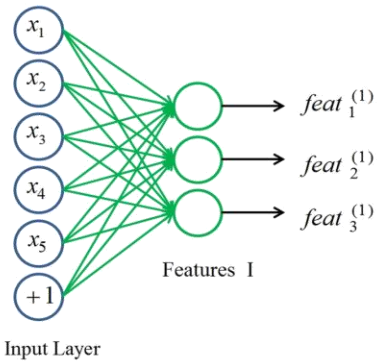

Figure 5: First hidden layer: This network is obtained if the last layer of the (trained) auto-encoder shown in Figure 4 is removed. , the outputs f eat(1)(i), also known as features, should give a better representation of the inputs xi.

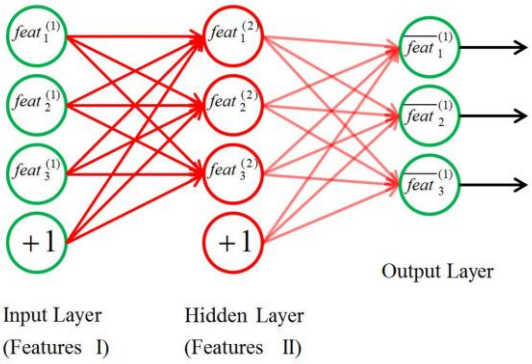

Figure 6: Second hidden layer training: The second auto-encoder, trained with the features obtained from the first auto-encoder. The parameters of the first auto-encode are not changed throughout the training procedure of this second one. This way, the layers are trained greedily, layer-wise.

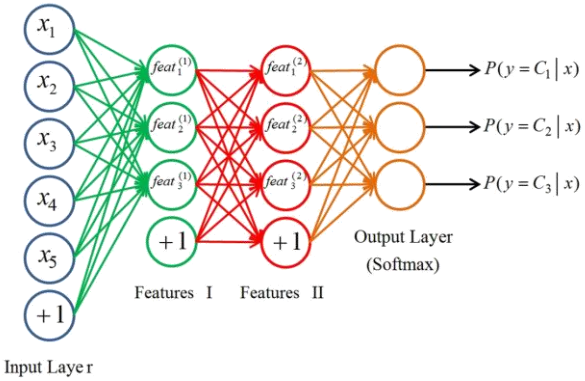

Figure 7: The final network formed by stacking the pre-trained hidden layers together and adding a final output layer (softmax-layer). Finally, fine-tuning of the whole network is performed.

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 121-128 | Cosmos Impact Factor (Germany): 5.195
Received: 23.03.2018 Published : 22.04.2018

The original features are fed to the network shown in Figure 5, to obtain the primary features f eat(1)(i) for the input xi. These features are new representations of the original input-samples and they are used as inputs for the next sparse auto-encoder, which is shown in Figure 6 [39]. This auto-encoder is trained the same way as the first one, where the resulting secondary features f eat(2)(i) can be seen as an even higher-level representation [7]. The last layer of the second auto-encoder is removed again and the primary features are fed into this second sparse auto-encoder to obtain the secondary feature representations. In this way, the hidden layers of a deep neural network can be pre-trained one after another, where only the network parameters of the currently trained layer are updated and the output of the currently trained layer is used as input for the subsequent auto-encoder [3]. To form the final network, at first the pre-trained hidden layers are stacked together. Afterwards, the final layer is added (e.g. a softmax-layer in case of a multi-class classification problem) and the whole network is trained in a supervised fashion (with labelled data) [9]. This second training stage, where error back-propagation and gradient descent are used to update the parameters of all layers (and not only the weights of the final layer), is called fine-tuning [9]. In the concrete example considered in this section, the two pre-trained hidden layers are stacked together and a softmax-layer is added to form the final network, as can be seen in Figure 7.

4. EXPERIMENT AND RESULT

The deep learning methods are evaluated on the DICOM (Digital imaging and communication in medicine) dataset [5]. The DICOM dataset is a database of 28-by-28 pixel medical images and is a subset of the MRI database [50]. The DICOM training set comprises 30000 images from the MRI s Special Database 1 (SD-1), 30000 images. To be exact, stacked sparse auto-encoders and stacked denoising auto-encoders use unsupervised data in the pre-training stage. The DICOM database was normalized to fit into a 20-by-20 pixel image, where the aspect ratio was kept for all examples during this resizing process [50]. Since an anti-aliasing technique was used within this normalization procedure, the resulting images also contain grey levels. The size-normalized examples were centred in the final 28-by-28 image by computing the centre of mass of the pixels (for the 20-by-20 image) and translating the image so the centre of mass is located at the centre of the final 28-by-28 image [50]. Randomly selected images from the DICOM dataset are shown in Figure 8.

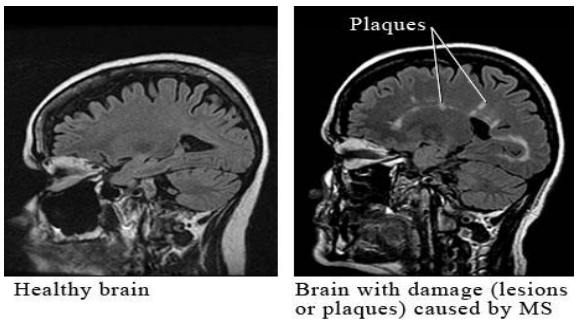

Figure 8: Randomly selected examples from the DIOCOM database of MRI of brain. (a) Examples taken from the training set. (b) Test set samples

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 121-128 | Cosmos Impact Factor (Germany): 5.195
Received: 23.03.2018 Published : 22.04.2018

5. CONCLUSION

With respect to the general performance of the methods, it can be stated that the stacked sparse auto-encoder, the stacked denoising auto-encoder and the SVM(Support vector Machines) achieved the highest accuracy among all evaluated approaches on DICOM datasets. At the same time, these three methods exhibit the highest training time among all evaluated approaches on DICOM datasets. Therefore these methods are preferable if the available computational resources allow to use them. In contrast, the K-nearest neighbour approach exhibit the shortest training time on DICOM datasets, but achieve a poorer accuracy than the aforementioned approaches. The K-nearest neighbour algorithm fails to solve more complicated classification tasks, but the stacked sparse auto-encoder and the stacked denoising auto-encoder are all successful in learning problem-specific features.

REFERENCES

1. Aparna M. Nichat, S. A. Ladhake," Brain Tumor Segmentation and Classification Using Modified FCM and SVM Classifier", International Journal of Advanced Research in a. Computer and Communication Engineering Vol. 5, Issue 4, April 2016.
2. Jurgen Schmidhuber. Deep learning in neural networks: An overview. Neural Networks, 61:85–117, 2015.
3. Rupesh K Srivastava, Klaus Greff, and Jurgen Schmidhuber. Training very deep networks. In Advances in Neural Information Processing Systems, pages 2368–2376, 2015
4. Lakshmiprabha, S. A New Method of Image Denoising Based on Fuzzy Logic. Med well, International Journal of Software Computing vol. 3, no. 1, pp. 74-77,2008.
5. Muhammad Javed Iqbal, Ibrahima Faye and Brahim Belhaouari Samir, "Efficient Feature Selection and Classification of Protein Sequence Data in Bioinformatics," The Scientific World Journal, vol. 2014, Article ID 173869, 12 pages, June 2014.
6. Y.Murali Mohan Babu, Dr.M.V. subramanyam, Dr.M.N.Giri Prasad," PCA based image de-noising", Published in SIPIJ,Vol 2 ,April 2012.
7. Dr.N.NandhaGopal ," Automatic Detection of Brain Tumor through Magnetic Resonance Image.International Journal of Advanced Research in Computer and Communication Engineering, Vol:2, Issue 4. 2013.
8. Parveen and Amritpal singh, "Detection of Brain Tumor in MRI Images, using Combination of Fuzzy C-Means and SVM" 2015 2nd International Conference on Signal Processing and Integrated Networks (SPIN) ©2015 IEEE.
9. Dr.G.Padmavathi, Dr.P.Subashini, M. MuthuKumar and Suresh Kumar Thakur "Comparison of the filters used for underwater Image-Preprocessing",IJCSNS International. Journal of Computer Science and the Network Security, VOL.10 No.1, January 2010.
10. Pandey, R. An Improved Switching Median filter for the Uniformly Distributed Impulse Noise Removal. WASET, vol.28, pp. 349-351.2008.
11. C. Platero and M. C. Tobar, "A multiatlas segmentation using graph cuts with applications to liver segmentation in CT scans,"Computational and Mathematical Methods in Medicine, vol.2014, Article ID182909, 16 pages, 2014.

International Research Journal in Global Engineering and Sciences. (IRJGES)
ISSN : 2456-172X | Vol. 3, No. 1, March - May, 2018
Pages 121-128 | Cosmos Impact Factor (Germany): 5.195
Received: 23.03.2018 Published : 22.04.2018

12. Y. Pavan Kumar Reddy and G.Kesavan," Tumor identification using Self Organizing Map and BFR Algorithm", Middle-East Journal of Scientific Research 24 (6): 2110-2115, 2016. IDOSI Publications, 2016
13. Pawan Patidar ,Manoj Gupta, Sumit Srivastava ,Ashok Kumar Nagawat,"Image de-noising by various filters for different types of noises ",Published in International Journal of computer applications,Vol-9,November 2010.
14. Prakash Mahindrakar and Dr. M. Hanumanthappa, "Data Mining In Healthcare: A Survey of Techniques and Algorithms with Its Limitations and Challenges", Int. Journal of a Engineering Research and Applications, ISSN: 2248-9622, Vol. 3, Issue 6, Nov-Dec 2013, pp.937-941.