

OVERVIEW OF DEEP LEARNING MODELS IN BIOMEDICAL DOMAIN WITH THE HELP OF R STATISTICAL SOFTWARE

Vinaitheerthan Renganathan

Skyline University College Sharjah, Sharjah United Arab Emirates

Received: 18.07.2018.

Accepted: 12.10.2018.

Corresponding author:

Vinaitheerthan Renganathan

Head of Institutional Research, Skyline University
College, P.O. Box 1797, University City,
Sharjah, United Arab Emirates

Phone: 971-6-5441155, Fax: 971-6-5441166

E-mail: vinairesearch@yahoo.com

ABSTRACT

With the increase in volume of data and presence of structured and unstructured data in the biomedical field, there is a need for building models which can handle complex & non-linear relations in the data and also predict and classify outcomes with higher accuracy. Deep learning models are one of such models which can handle complex and nonlinear data and are being increasingly used in the biomedical field in the recent years. Deep learning methodology evolved from artificial neural networks which process the input data through multiple hidden layers with higher level of abstraction. Deep Learning networks are used in various fields such as image processing, speech recognition, fraud deduction, classification and prediction. Objectives of this paper is to provide an overview of Deep Learning Models and its application in the biomedical domain using R Statistical software. Deep Learning concepts are illustrated by using the R statistical software package. X-ray Images from NIH datasets used to explain the prediction accuracy of the deep learning models. Deep Learning models helped to classify the outcomes under study with 91% accuracy. The paper provided an overview of Deep Learning Models, its types, its application in biomedical domain. - is paper has shown the effect of deep learning network in classifying images into normal and disease with 91% accuracy with help of the R statistical package.

Keywords: *Deep learning network, Convolution network, Classification, image processing, Artificial Neural Network.*



UDK: 61:004.8

Ser J Exp Clin Res 2022; 23(1): 3-11

DOI: 10.2478/sjecr-2018-0063



INTRODUCTION

With the increase in volume of data and presence of structured and unstructured data in the biomedical filed, there is a need for building models which can handle complex & non-linear relations in the data and also predict and classify outcomes with higher accuracy. Deep learning models are one of such models which can handle complex and nonlinear data and are being increasingly used in the biomedical filed in the recent years.

Deep learning (1, 2) networks or models evolved from the artificial neural networks which works on the basis of biological nervous system. Neuron is a centre point of nervous system which processes the information. It contains cell body, axon and dendrites. Neurons are connected to other neurons through synapses which include axon and dendrites structure at the two ends. Neuron receives the information from other neurons through its dendrites and sends the processed information through its axon. Neurons will be active only if the impulse they receive is above the threshold.

Artificial Neural Networks, like the biological nervous system, consists of input, hidden layer and output layer. Each layer consists of nodes which is equivalent to neurons in biological network. Hidden layer receives inputs from the input layer as weighted information and process the information through activation functions and sends it to the output layer. The weights are initially assigned randomly and updated based on the data and the network is trained several times before processing the final information.

Deep Learning networks contains more hidden layers which are capable of handling vast amount data such as pixels in images at higher level abstraction. Deep Learning Models are used in various fields such as image processing, speech recognition, fraud deduction, classification, segmentation and prediction.

DEEP LEARNING MODELS IN BIOMEDICAL DOMAIN

Deep Learning models are particularly useful in biomedical field (3) due to the nature, type and volume of data present in the field. It helps to create predictive and classification models from the heterogeneous datasets. The Deep Learning models fits well for nonlinear data which is not possible with the help of traditional statistical models and it is capable of handling both structured and non-

structured data. Deep learning models are especially useful for image processing (4-12), predicting the disease from the given patient characteristics (13-16), analyse the complex structure genomics datasets(17,18,19) and as a natural language processing tool to process clinical notes (20). Deep Learning models are becoming a one of the important tools in the field of Precision (20, 21) or personalized medicine which focuses on the providing treatment at the individual patient level.

TYPES OF DEEP LEARNING NETWORKS

Deep learning models use the following types of networks: Convolution Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) and Deep Belief Network layers.

Convolution Neural Networks (CNN)

Convolution Neural Networks (CNN) is especially useful in image processing (7, 23, 24). The convolution networks are widely used due to the availability of Graphical Processing Units (GPU). GPUs are capable of processing high volume of information at the same time compared to the normal Central Processing Units (CPU). The architecture of the CNN network is given in the figure-1 below and also explained in the sections below:

Input Layer and Receptive Fields

In CNN, unlike the fully connected networks, each neuron is not connected to the other as it involves too many weights & complexities and it is connected to only 'receptive field'. Receptive fields are two dimension regions in the input image which is used to connect the neurons from image to the convolution layer.

Convolution Layer - Filtering and Feature maps

The convolution process is carried out by the convolution layer (25) which consists of filters or kernel and feature map. If Input layer is an image with dimension $x*y*z$ where x is the height, y is the width and z is the color channels of the image then a filter for the image can be defined with dimension $r*s*t$ where r is the length ($< x$) and s is the width ($< y$) and t is the color channel ($<= z$). When the filter slides through the full size of the image, it produces

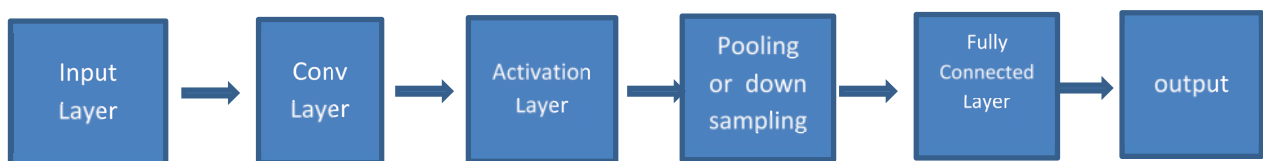


Figure-1: Convolution Neural Network Architecture



feature map of size $x-r+1$. The feature map with respect to an image can be edges and curves etc. If the CNN has 3 filters, then the filter x slides through the image and CNN convolve to produce 3 feature maps. The convolution process is also called activation. The feature maps are sampled using mean or maximum functions.

Let us take an example of $4 \times 4 \times 3$ image with pixel values and show how a $4 \times 4 \times 3$ image is filtered and feature map is produced through the CNN network.

- a. Original image ($4 \times 4 \times 3$) (length-4, width-4, colour channel -3)

1	1	0	0
0	1	0	1
1	0	0	0
0	1	0	1

- b. Filter ($2 \times 2 \times 3$) – (length-2, width-2, colour channel -3)

1	0
0	1

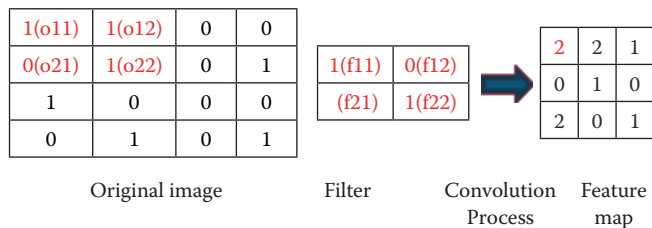


Figure-2: Convolution process – Creation of Feature map for an image by sliding image filters of a given size

Convolution process: The figure-2 shows how the filter of a given size will slide through the first four cells of the original image o11, o12, o21, o22 and each element in the original image will be multiplied by the corresponding cell elements in the filter f11, f12, f21, f22. For example in the first cell of feature map is derived from multiplying $1 \times 1 + 1 \times 0 + 0 \times 0 + 1 \times 1 = 2$. Similarly all the other cells of the feature map which is of size $4-2+1 = 3$ (3×3) is obtained.

Activation Layer

After the convolution layer in the CNN network, a non-linear functions such as tanh or sigmoid or ReLu function (26, 27) is used to introduce non linearity in the process as convolution layer uses linear functions during its operations. The process is carried out through the activation layer.

Pooling Layer

When the image sizes are large, it is desirable to introduce pooling layer or ‘downsampling’ layer which reduces

the special size of the image through max pooling or average or L2 norm (mean square norm) pooling (28, 29). The same values are considered from each dimension of the image during the pooling process which helps to maintain the dimension of the image with reduced size.

Fully Connected Layer

Fully connected layer (30, 31) is the final layer in the CNN which takes the input from the previous layer as vector with one dimension and outputs the data.

Drop out layer

Drop out layer (32) is used to reduce the overfitting problem in the network as the network will be normally tuned to the training data and when new data comes in, it might over fit. To avoid over fitting problem, some of the activation will be dropped from the layer. It helps the network function work properly even some of the activation is dropped and also reduces the over fitting problem.

Recurrent Neural Networks (RNN)

Recurrent Neural Networks (33, 34) takes into account the present input and as well as the previous input processed before it. RNN are capable of learning the dependencies between spatially correlated data such as image pixels. It adds memory into the network and creates a sequential process. The sequential information is processed by the hidden layers of RNN and correlation between the two periods of time are calculated and stored. The hidden layer in the network maintains a hidden state and updates it based on the input layer and previous hidden state. One of the problems with the RNN is it difficult to find the long-term dependencies due to vanishing gradients during back propagation process.

Long Short Term Memory Networks (LSTM)

Long Short Term Memory (LSTM) networks (35, 36) overcome the problem of vanishing gradient by preserving the errors developed during the past. LSTM networks are used in Natural Language Processing, Speech recognition and pattern recognition. LSTM networks contain memory cell, input, output gate (to input and output information), forget gate (to drop information) which allows the network to maintain its state during the information flow. The LSTM models learn the current state and predict the future state of the parameters under study.

Deep Belief Network

Deep Belief Network (DBN) (1) is multi-layer network which includes hidden layers which are connected to Restricted Boltzmann Machine (RBM). Restricted Boltzmann



Machines does not have connection between visible nodes but only among hidden nodes and enables us to get an unbiased sample from the posterior distribution. DBN captures higher level representation of input fed from the input layer and assigns weights initially and fine tunes through back-propagation. Deep Belief Network is used in image processing tasks.

DEEP LEARNING MODEL FOR IMAGE CLASSIFICATION & SEGMENTATION, CLINICAL DATA MINING

Deep Learning models are used in classifying and segmenting the images such as X-rays, Mammograms, Histopathological images, predicting diseases from the given patient characteristics and processing the text in the clinical documents such as clinical notes. The process of each model is explained in the following sections 4.1-4.3:

Deep learning models for image classification and segmentation

Deep Learning models are widely used for image classification and segmentation, as it has the advantage over the other traditional models due to its processing speed, use of less resources, higher accuracy of the classification and segmentation (37,38). The images are processed through the convolution layer to derive the feature maps for the image. The nonlinearity is introduced through activation layer and the image sizes are optimized through the pooling layer. In next stage the fully connected layers convert one dimension vector into output. The model uses training and test data set to classify or segment the image. The accuracy of classification is measured by Accuracy rate (Number classified correctly out of the total images) and through the Receiver Operating Characteristics Curve (True Positive rate vs False Positive rate)

Deep Learning models for clinical prediction

Deep Learning model uses the patient data from the sources such as Electronic Health Records (39, 40) and derive the patient characteristics using learning network for predicting the probability of having a disease condition (14,15,16). Each layer in the network encompasses a higher level of observed patient characteristics which are obtained from previous layers. The model uses training and test data set to predict the probability of a patient having a particular disease condition or not given the patient characteristics. The Receiver Operating Characteristic curve and the accuracy of prediction are calculated to decide the predictive accuracy of the model.

Deep Learning Models for Natural Language Processing

Deep Learning Models are helpful in analysing vast amount of data present in the clinical field in the form of

clinical notes (20, 41); Deep learning models are preferred over the other traditional models as data can be trained with a single end-to-end model (42-44). Deep Learning models learn the features from the language data set and there is no need to manually specify and extract the features from the language dataset. Deep learning models are useful in carrying out tasks such as Named Entity Recognition (patient name, drug name, city, etc.) or Part of Speech Tagging (marking of words according to the category).

METHODS

Compliance with Ethical Standards:

- * **Disclosure of potential conflicts of interest**
Conflict of Interest: Author declares that there is no conflict of interest.
- * **Research involving human participants and/or animals:**
Ethical approval: This article does not contain any studies with human participants performed by the author
- * **Informed consent**
Not Applicable. The study did not involve any human participants.
- * **Funding:**
This study was not funded by any organization.

SAMPLE DEEP LEARNING NETWORK USING R STATISTICAL PACKAGE

The following example uses R open source statistical computing software (45) which is useful for carrying out various statistical tests and methods, graphics, text and data mining procedures (29). The R software can be downloaded from the software website (45). The R software can be used in various integrated development environment (IDE) such as R-Studio, Eclipse and StatET. The R software works with the concepts called packages which is a compilation of user created codes and can be used to perform specific functions.

The paper uses mxnet (46) package. The data set used chest x-ray images from NIH website (47) which is classified as normal (no finding) or with some disease for the study purpose. Users can also try mammograms images database such as DDSM (48), mini-MIAS (49), B-Screen (50) and histopathological images from MITOSIS-ATYP-IA challenge (51) database. Results are shown for illustrative purpose only and are not meant to represent the true performance of the deep learning model.

The creation of sample deep learning model starts with the creation of working directory (**Table-1: Step-1**). The X ray images need to be stored in the respective directories for further processing. The images need to be reduced in size before converting and stored as Comma Separated



Table-1: Conversion of x-ray images into CSV files and creation of Train and Test Data sets

Steps	Description and R code	R Code
Step-1	Set working Directory	<code>wd = "C:// Normal" setwd(wd)</code>
Step-2	Call EBImage r –package	<code>library("EBImage")</code>
Step-3	Define Label and Data frame for normal x-ray images	<code>label <- 1 dataframe <- data.frame()</code>
Step-4	Convert the normal x-rays images into CSV files after reducing the image size into 36*36 dimension	<pre>for(i in 1:length(images)) { normalimg <- readImage(images(i)) normalimg <- resize(normalimg, 36, 36) normalimg1 <- normalimg@.Data normalimg2 <- as.vector(t(normalimg1)) nomralvector <- c(label, nomarlimg2) dataframe <- rbind(dataframe,normalvector) } wd = "C://datafiles" setwd(wd) names(dataframe) <- c("label", paste("pixel", c(1:1296))) write.csv(dataframe, "normal.csv", row.names = FALSE)</pre>
Step-5	Set working directory, define label and data frame for disease x-ray images	<code>wd = "C://disease" setwd(wd)</code> <code>label <- 2 df <- data.frame() images <- list.files()</code>
Step-6	Convert the disease x-rays images into CSV files after reducing the image size into 36*36 dimension	<pre>for(i in 1:length(images)) { normalimg <- readImage(images(i)) normalimg <- resize(normalimg, 36, 36) normalimg1 <- normalimg@.Data normalimg2 <- as.vector(t(normalimg1)) nomralvector <- c(label, nomarlimg2) dataframe <- rbind(dataframe,normalvector) } ke = "C://datafiles" setwd(ke) names(dataframe) <- c("label", paste("pixel", c(1:1296))) write.csv(dataframe, "disease.csv", row.names = FALSE)</pre>
Step-7	Calling the dataset	<code>normal <- read.csv("normal.csv") disease <- read.csv("disease.csv") overall <- rbind(normal, disease) reorded <- sample(overall(1:80),0)</code>
Step-8	Create the Train and test data sets	<code>train <- reorded(1:240, test <- reorded(241:400,)</code>

Value (CSV) file. The EBImage r package (52) is used to convert the normal and disease x ray images into respective CSV file after reducing the size of the x images which are stored in the respective working directories (**Table-1 :Step2 to Step6**).

Once the data is converted into disease and normal CSV file, a combined overall dataset is created and in order to create a sample test and train dataset (**Table-1: Step 7 to Step8**).

During the next step the mxnet r package is called to start the modelling process by creating the dependent and independent variables x and y for both test and train dataset (**Table-2: Step9 to Step10**). After creating the de-

pendent and independent variables, convolution layer, activation layer and pooling layers are created (**Table-2: Step-11**). The final layer in the model i.e fully connected layers is created before starting the model creation (**Table-2: Step12**) which provides the output to the model.

The model uses the output provided by the previous layers and independent and dependent variables to classify the images into disease or normal images (**Table-3: Step-13**). The final step in the whole process is to test and evaluate the accuracy of the model by using prediction function (**Table-3: Step-14**) which uses the test data which was created as a sample from the combined dataset (**Table-1:Step8**).



Table-2: Creation of independent and dependent Variables and layers

Steps	Description and R code	R Code
Step-9	Calling MXnet package	<code>require(mxnet)</code>
Step-10	Define Dependent and independent variable from the train and test dataset	<pre> train <- data.matrix(train) train.x <- t(train[,-1]) train.y <- train[,1] train_x1 <- train.x dim(train_x1) <- c(36, 36, 1, ncol(train.x)) test <- data.matrix(test) test.x <- t(test[,-1]) test.y <- test[,1] test.x1 <- test_x dim(test.x1) <- c(36, 36, 1, ncol(test_x)) </pre>
Step-11	Building 3 convolution, 3 activation and 3 pooling layers	<pre> data <- mx.symbol.Variable('data') convolution1 <- mx.symbol.Convolution(data= data, kernel = c(5,5), num_filter = 25) activation1 <- mx.symbol.Activation(data= convolution1, act_type = "relu") pooling1 <- mx.symbol.Pooling(data = activation1, pool_type = "max", kernel = c(2,2), stride = c(2,2)) convolution2 <- mx.symbol.Convolution(data= data, kernel = c(5,5), num_filter = 30) activation2 <- mx.symbol.Activation(data= convolution1, act_type = "relu") pooling2 <- mx.symbol.Pooling(data = activation1, pool_type = "max", kernel = c(2,2), stride = c(2,2)) convolution2 <- mx.symbol.Convolution(data= data, kernel = c(5,5), num_filter = 30) activation3 <- mx.symbol.Activation(data= convolution1, act_type = "relu") pooling3 <- mx.symbol.Pooling(data = activation1, pool_type = "max", kernel = c(2,2), stride = c(2,2)) </pre>
Step-12	Building 3 Fully Connected layers	<pre> flattendata <- mx.symbol.Flatten(data = pool_2) fully1 <- mx.symbol.FullyConnected(data = flattendata, num_hidden = 600) activation4 <- mx.symbol.Activation(data = fully1, act_type = "relu") fully2 <- mx.symbol.FullyConnected(data = activation4, num_hidden = 300) activation5 <- mx.symbol.Activation(data = fully2, act_type = "relu") fully3 <- mx.symbol.FullyConnected(data = activation5, num_hidden = 2) DL <- mx.symbol.SoftmaxOutput(data = fully3) mx.set.seed(200) device <- mx.cpu() </pre>

Steps	Description and R code	R Code
Step-13	Building Deep Convolution network model	<pre> model <- mx.model.FeedForward.create(DL, X = train.x1, y = train.y, ctx = device, num.round = 30, array.batch.size = 100, learning.rate = 0.05, momentum = 0.9, wd = 0.00001, eval.metric = mx.metric.accuracy, epoch.end.callback = mx.callback.log.train.metric(100)) </pre>
Step-14	Building Deep Convolution network model predictions	<pre> prediction <- predict(model, test.x1) predlabels <- max.col(t(prediction))-1 table(test[,1], predlabels) </pre>

Table-3: Building Deep Learning Convolution model and prediction model

RESULTS AND DISCUSSION

The initial output of the deep learning model is the conversion of the images into CSV file and stores them as train and test data set. The table-4 shows the sample test and train dataset content which are processed by EBimage package which are stored in the CSV file.

In Table 4a and Table4b above, the label indicates the type of image label 1-normal image and another is label2-disease image.

The training models iteration set for 200 in this example and after each iteration the training accuracy recorded and is shown in the below Table-5

To evaluate the accuracy and performance of the deep learning model, 60% dataset is used for training and 40% dataset is used for testing the model. The following Table-6 provides predicted values in comparison with the actual classification diseased vs. normal patient x-ray classification

From the above table-6, the classification accuracy is calculated as 91%.



Table 4a. Sample Train dataset - Conversion of images into CSV file and stored as pixels values

Label	pixel.1	pixel.2	pixel.3	pixel.4	pixel.5	pixel.1296
1	0.121917	0	0	0	0.003922		0
2	0.036765	0.047059	0.043137	0.035294	0.027451		0.156327
2	0.03927	0.054902	0.039216	0.035294	0.031373		0.178422
1	0.024837	0.031373	0.031373	0.031373	0.189052		0.003922

Table 4b. Sample Test dataset - Conversion of images into CSV file and stored as pixels values

Label	pixel.1	pixel.2	pixel.3	pixel.4	pixel.5	pixel.1296
2	0.195016	0.055882	0.047059	0.03893	0.03223		0.152982
2	0.039049	0.058824	0.048584	0.039216	0.039216		0.198641
1	0.121917	0	0	0	0.003922		0
1	0.400605	0.003922	0.002397	0.43435	0.456645		0

CONCLUSION

This paper has shown the effect of deep learning network in classifying images into normal and disease with 91% accuracy with help of the R statistical package. Deep learning models showed considerable improvement in classifying images over the other models in terms of accuracy and speed due to the availability of powerful GPU systems. Deep learning models will play an important role in classifying the images in the future which will be of significant help to radiologist in reducing the time and prepare the reports. More number of free and large databases is required to train the model and increase the accuracy in the biomedical domain.

REFERENCES

1. Goodfellow I, Bengio Y, Courville A, Bengio Y. 2016. Deep learning (Vol. 1). Cambridge: MA, USA, MIT press.
2. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015; 521(7553): 436.
3. Miotto R, Wang F, Wang S, Jiang X, Dudley JT. Deep learning for healthcare: review, opportunities and challenges. Briefings in bioinformatics. 2018; 19(6): 1236-46.
4. Urban G, Bache KM, Phan D, et al. Deep Learning for Drug Discovery and Cancer Research: Automated Analysis of Vascularization Images. IEEE/ ACM Transactions on Computational Biology and Bioinformatics. 2018; 16(3): 1029-35.
5. Lakhani P. Deep convolutional neural networks for endotracheal tube position and X-ray image classification: challenges and opportunities. Journal of digital imaging. 2017; 30(4): 460-8.
6. LeCun Y, Bengio Y. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks. 1995; 3361(10).
7. Gao M, Bagci U, Lu L, et al. Holistic classification of CT attenuation patterns for interstitial lung diseases via deep convolutional neural networks. Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization. 2018; 6(1): 1-6.
8. Mohamed AA, Berg WA, Peng H, Luo Y, Jankowitz RC, Wu S. A deep learning method for classifying mammographic breast density categories. Medical physics. 2018; 45(1): 314-21.
9. Esteva A, Kuprel B, Novoa RA, et al. Dermatologistlevel classification of skin cancer with deep neural networks. Nature. 2017; 542(7639), 115.
10. Saltz J, Gupta R, Hou L, et al. Spatial organization and molecular correlation of tumor-infiltrating lymphocytes using deep learning on pathology images. Cell reports. 2018; 23(1): 181-93.

Table 5. Training Accuracy

Iteration number	Train-accuracy
[1]	0.6705694
[2]	0.6507439
[3]	0.6633542
[4]	0.6866255
...	----
[21]	0.8738582
[22]	0.8920106
[23]	0.8885165
[24]	0.8518872
...	
[199]	0.9034987
[200]	0.9414315

Table-6: Actual vs Predicted values for 160 test images

	Normal	Disease
Normal	70	10
Disease	5	75



11. Kermany DS, Goldbaum M, Cai W, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*. 2018; 172(5): 1122-31.
12. Gerard SE, Patton TJ, Christensen GE, Bayouth JE, Reinhardt JM. FissureNet: A deep learning approach for pulmonary fissure detection in CT images. *IEEE transactions on medical imaging*. 2018; 38(1): 156-66.
13. Hinton, G. Deep learning—a technology with the potential to transform health care. *JAMA*. 2018; 320(11): 1101-2.
14. Chen M, Hao Y, Hwang K, Wang L, Wang L. Disease prediction by machine learning over big data from healthcare communities. *IEEE Access*. 2017; 5: 8869-79.
15. Miotto R, Li L, Kidd BA, Dudley JT. Deep patient: an unsupervised representation to predict the future of patients from the electronic healthy records. *Scientific reports*. 2016; 6: 26094.
16. Liu S, Liu S, Cai W, Pujol S, Kikinis R, Feng D. Early diagnosis of Alzheimer's disease with deep learning. In *Biomedical Imaging (ISBI)*. IEEE 11th International Symposium. 2014, p. 1015-1018.
17. Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA and RNA-binding proteins by deep learning. *Nature biotechnology*. 2015; 33(8): 831.
18. Park Y, Kellis M. Deep learning for regulatory genomics. *Nature biotechnology*. 2015; 33(8): 825.
19. Chen Y, Li Y, Narayan R, Subramanian A, Xie X. Gene expression inference with deep learning. *Bioinformatics*. 2016; 32(12): 1832-9.
20. Weng WH, Waghlikar KB, McCray AT, Szolovits P, Chueh HC. Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach. *BMC medical informatics and decision making*. 2017; 17(1): 155.
21. Collins FS, Varmus H. A new initiative on precision medicine. *New England Journal of Medicine*. 2015; 372(9): 793-5.
22. Nezhad MZ, Zhu D, Li X, Yang K, Levy P. Safs: A deep feature selection approach for precision medicine. In *Bioinformatics and Biomedicine (BIBM)*, IEEE International Conference, 15-18, Dec 2016, p. 501-06. Shenzhen, China, IEEE.
23. Lu L, Zheng Y, Carneiro G, Yang L. 2017. *Deep Learning and Convolutional Neural Networks for Medical Image Computing*. (1st Ed) MA, USA. Springer.
24. Lo, SCB, Chan HP, Lin JS, Li H, Freedman MT, Mun SK. Artificial convolution neural network for medical image pattern recognition. *Neural networks*. 1995; 8(7-8): 1201-14.
25. Ciresan DC, Meier U, Gambardella LM, Schmidhuber J. Convolutional neural network committees for handwritten character classification. In: *Document Analysis and Recognition (ICDAR)*, International Conference, 2011, p. 1135-9. Beijing, China, IEEE.
26. Mou L, Ghamisi P, Zhu XX. Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*. 2017; 55(7): 3639-55.
27. Tran SD, Manmatha R. U.S. Patent No. 9, 892,344. Washington, DC: U.S. Patent and Trademark.
28. Zhang Y, Shi B. Improving pooling method for regularization of convolutional networks based on the failure probability density. *Optik-International Journal for Light and Electron Optics*. 2017; 145, 258-65.
29. Del Fiol G, Michelson M, Iorio A, Cotoi C, Haynes RB. A Deep Learning Method to Automatically Identify Reports of Scientifically Rigorous Clinical Research from the Biomedical Literature: Comparative Analytic Study. *Journal of medical Internet research*. 2018; 20(6): e10281.
30. Choi K, Fazekas G, Sandler M, Cho K. (2017, March). Convolutional recurrent neural networks for music classification. In *Acoustics, Speech and Signal Processing (ICASSP)*. IEEE International Conference. 2017, p. 2392-6. New Orleans, USA, IEEE.
31. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille AL. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*. 2018; 40(4): 834-48.
32. Kimmel J, Brack A, Marshall WF. Deep convolution neural networks allow analysis of cell motility during stem cell differentiation and neoplastic transformation. *bioRxiv*. 2017; 159202.
33. Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp)*, IEEE international conference, 2013 p. 6645-9. Vancouver, Canada, IEEE.
34. Sun X, Li T, Li Y, Li Q, Huang Y, Liu J. Recurrent neural system with minimum complexity: A deep learning perspective. *Neurocomputing*. 2018; 275: 1333-49.
35. Tan JH, Hagiwara Y, Pang W, et al. Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals. *Computers in Biology and Medicine*. 2018; 94:19-26.
36. Lipton ZC, Kale DC, Elkan C, Wetzel R. Learning to diagnose with LSTM recurrent neural networks. *arXiv preprint arXiv*. 2015; 1511.03677.
37. Jnawali K, Arbabshirani MR, Rao N, Patel AA. (2018, February). Deep 3D convolution neural network for CT brain hemorrhage classification. In *Medical Imaging*



- 2018: Computer-Aided Diagnosis (Vol. 10575, p. 105751C).
38. Acharya UR, Oh SL, Hagiwara Y, Tan JH, Adeli H. Deep convolutional neural network for the automated detection and diagnosis of seizure using EEG signals. *Computers in biology and medicine*, 2018; 100: 270-8.
 39. Li W, Shi S, Gao Z, et al. Improved deep belief network model and its application in named entity recognition of Chinese electronic medical records. In *Big Data Analysis (ICBDA), IEEE 3rd International Conference*, 2018 p. 356-60. Shanghai, China, IEEE.
 40. Shickel B, Tighe PJ, Bihorac A, Rashidi P. Deep EHR: A survey of recent advances in deep learning techniques for electronic health record (EHR) analysis. *IEEE journal of biomedical and health informatics*. 2018; 22(5): 1589-1604.
 41. Zech J, Pain M, Titano J, et al. Natural Language-based Machine Learning Models for the Annotation of Clinical Radiology Reports. *Radiology*. 2018; 287(2), 570-80.
 42. Del Fiol G, Michelson M, Iorio A, et al. A Deep Learning Method to Automatically Identify Reports of Scientifically Rigorous Clinical Research from the Biomedical Literature: Comparative Analytic Study. *Journal of medical Internet research*, 2018; 20(6).
 43. Young T, Hazarika D, Poria S, Cambria E. Recent trends in deep learning based natural language processing. *IEEE Computational intelligence magazine*, 2018; 13(3), 55-75.
 44. Deng L, Liu Y. A joint introduction to natural language processing and to deep learning. In: *Deep learning in natural language processing*. 2018, p. 1-22. Springer, Singapore.
 45. R Core Team R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. Retrieved September 21, 2018 from <http://www.R-project.org/>.
 46. Chen T, Kou Q, He T. mxnet. MXNet [2015]. Retrieved September 21, 2018 from <https://github.com/dmlc/mxnet/R-package>.
 47. Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. *IEEE CVPR*, 21-26 July 2017(2097-2106). Honolulu, Hawaii, IEEE.
 48. Heath M, Bowyer K, Kopans D, et al. Current status of the digital database for screening mammography. In *Digital mammography* (pp. 457-460). Springer, Dordrecht.
 49. J Suckling et al. The Mammographic Image Analysis Society Digital Mammogram Database Exerpta Medica. *International Congress Series*. 1994; 1069: 375-8.
 50. Diagnostic image Analysis Group [2009] Retrieved September 21, 2018 from http://www.diagnijmegen.nl/index.php/NWO_Bayesian_Decision_Support_in_Medical_Screening_%28B-SCREEN%29
 51. MITOS-ATYPIA [2014] Retrieved September 21, 2018 from <https://mitos-atypia-14.grand-challenge.org/>
 52. Pau G, Fuchs F, Sklyar O, Boutros M, Huber W. EBImage—an R package for image processing with applications to cellular phenotypes. *Bioinformatics*. 2010; 26(7): 979-81.