

[Document title]

[Document subtitle]



[Date]

[Company name]

[Company address]

Contents

[Qno1 Take a 5\*5 matrix 2](#_Toc123957854)

[Qno2 Neural Networks 3](#_Toc123957855)

[Introduction to MLP neural networks 3](#_Toc123957856)

[Literature review 3](#_Toc123957857)

[Explanation of Dataset 4](#_Toc123957858)

[Impact of different activation functions (minimum 2) on the prediction 5](#_Toc123957859)

[Part iv 5](#_Toc123957860)

[Part v 6](#_Toc123957861)

[Qno3 6](#_Toc123957862)

[Part i 6](#_Toc123957863)

[Part ii 7](#_Toc123957864)

[Opportunities 7](#_Toc123957865)

[References 8](#_Toc123957866)

# Qno1 Take a 5\*5 matrix

First, let's define the input matrix and convolutional kernel that we will be using:

Input matrix:

[ab11, ab12, ab13, ab14, ab15]

[ab21, ab22, ab23, ab24, ab25]

[ab31, ab32, ab33, ab34, ab35]

[ab41, ab42, ab43, ab44, ab45]

[ab51, ab52, ab53, ab54, ab55]

Convolution kernel:

[kz11, kz12, kz13]

[kz21, kz22, kz23]

[kz31, kz32, kz33]

1. Convolutional Layer

The first operation in a CNN is the convolutional layer. In this layer, the input matrix is convolved with the kernel to produce a feature map.

The convolution operation is performed by sliding the kernel over the input matrix and element-wise multiplying the entries of the kernel with the input matrix and summing them up. This is done for each position of the kernel on the input matrix, resulting in a new matrix called the feature map.

For example, let's say we want to calculate the first element of the feature map (f11). We can do this by sliding the kernel over the input matrix and element-wise multiplying and summing the entries as follows:

f11 = (ab11 \* kz11) + (ab12 \* kz12) + (ab13 \* k13) + (ab21 \* kz21) + (ab22 \* kz22) + (ab23 \* kz23) + (ab31 \* kz31) + (ab32 \* kz32) + (ab33 \* kz33)

Certainly! To begin, let's consider the input matrix as **X** and the convolutional kernel as **K**.

For the convolutional layer, we perform the following operation:

**Y = X \* K + b**

Where **\*** represents the element-wise matrix multiplication and **b** is a bias term. This operation is performed for each element in the output matrix **Y**, which will be of size

**(5 - 3 + 1) x (5 - 3 + 1) = 3 x 3**.

For the pooling layer, we can use either max pooling or average pooling. For max pooling, we take the maximum value in each window of the input matrix and place it in the corresponding element in the output matrix. For example, if we use a 2 x 2 pooling window and stride of 2, the output matrix **Y** will be of size **(5 - 2) / 2 + 1 x (5 - 2) / 2 + 1 = 2 x 2**.

For average pooling, we take the average of all the values in each window of the input matrix and place it in the corresponding element in the output matrix. The output matrix **Y** will be the same size as for max pooling.

For the flattening layer, we take the output matrix from the previous layer and flatten it into a single vector. For example, if the output matrix from the pooling layer is **2 x 2**, the flattening layer will produce a vector of length **2 \* 2 = 4**.

I hope this helps! Let me know if you have any questions.

# Qno2 Neural Networks

## Introduction to MLP neural networks

Emulating the brain, the most flawless and well-organized processing system, remains one of the most intriguing scientific issues. We don't wish to downplay contemporary computers' true revolution, but it is commonly established that human brain can accomplish tasks like pattern recognition and restoration, object categorization, and abstract scheme learning in real time, but traditional computers cannot. The brain's structure explains why biological processing rates are many orders of magnitude faster than the simplest conventional computer equipment. This hypothesis has inspired scientists in the past 50 years to study the brain's processing philosophy and attempt to imitate its structure. The brain acts like a huge series of asynchronous, chemically driven, pulse-generating switches. Massive collaboration between many basic processing pieces (neurons) produces several intriguing props.

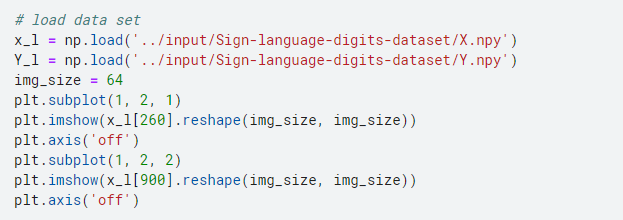
## Literature review

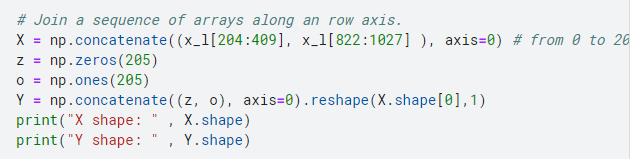
Multilayer perceptron’s are nonlinear mapping models with simple connected neurons, or nodes. The sum of the node's inputs and a basic nonlinear transfer, or activation, function determine weights and output signals. Multilayer perceptron superposes fundamental nonlinear transfer functions to represent extremely nonlinear processes. The multilayer perceptron only described linear functions with a linear transfer function. Logistic functions are popular transfer functions because derivative calculation is straightforward. The linking weight scales the next layer's nodes' output. The multilayer perceptron is feed-forward. Multilayer perceptron’s have numerous neuron layers. The input layer sends the vector to the network without computation. Multilayer perceptron has hidden and output layers. Multilayer perceptron links all nodes in the next and previous layers.

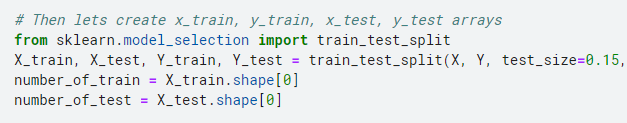
ANNs can adapt to changing system features, making them desirable. Many studies have optimised training techniques, learn parameters, and network topology to increase ANN performance in recent decades, but few have used activation functions. RBF neural networks are popular [1]. The conventional sigmoid approximates better than splines and polynomials [2]. Jordan's logistic function naturally represents the posterior probability in a binary classification issue [3].

## Explanation of Dataset

The selection of the dataset is the important part of the model building. We are going to choose a dataset with the sign language digits data set. The dataset is compromised of the 2062 sign language digit numbers. There are the 10 unique symbols because there are the 10 numbers from 0 to 9. The total number of available 0 signs is 205. They are all starting from the 204 and 408. And the one number lies between 822 and 827. The figure of 205 is very small for the deep learning but we have to proceed it with.





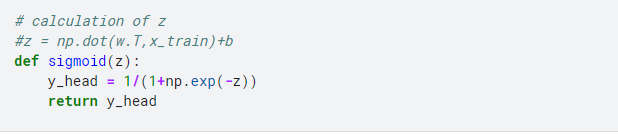


## Part iii Impact of different activation functions (minimum 2) on the prediction

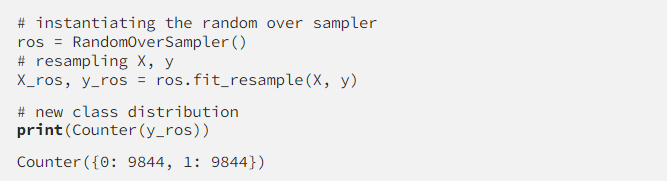
Sigmoid — this yields a number between zero and one, which may be interpreted as the degree to which the model is certain that the model belongs to the class.

**Functional Loss**

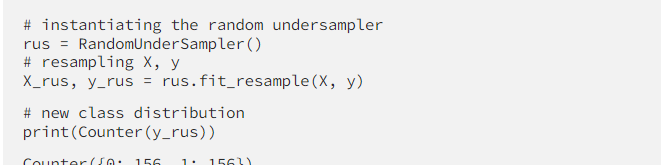
The dissimilarity between two probability distributions may be measured using the binary cross entropy. Each class in our model is expected to have a model distribution of p, 1-p (binary distribution). Binomial cross-entropy is used to compare these hypothetical distributions to the actual ones (y, 1-y) for each category.



## Part iv

Random oversampling, which includes selecting random instances from the minority class using substitution in order to augment the training data with numerous copies of this instance, may choose the same instance more than once. Overfitting may become more likely due to the fact that random oversampling produces duplicates of the minority class's instances. So, for instance, a symbolic classifier may construct rules that seem correct but only cover a single reproduced case. This method is very useful for improving the performance of Machine Learning algorithms like artificial neural networks and SVMs that are sensitive to skewed distribution. Tuning the target class distribution, however, is often recommended, since it might stop the algorithm against doing the overfitting the minority class, which would raise our generalization error if we desired a balanced distribution for a highly unbalanced dataset. In addition, we should be cognizant of the larger computing cost. Because the model is being exposed to the same examples multiple times, Building the model with more minority class instances may increase calculation time, particularly for highly skewed data.

When compared to Random Oversampling, Random Under sampling is the inverse. This technique aims to reduce the number of instances belonging to the majority class in the converted data by randomly selecting and removing samples from the majority class. Random under-sampling (may) lead to the loss of a huge amount of information. This is important because missing data may make it more difficult to understand where the minority cases begin and the majority instances end, which in turn can reduce classification accuracy. The under sampling method produces a new data set in which the majority class has fewer occurrences; this may be continued until the number of occurrences in each class is equal. This strategy works well when the underrepresented group still has enough representative examples to work with. Unfortunately, there is no method to identify or save the information-rich cases in the majority class, thus it is necessary to consider the possibility that significant information would be lost when we randomly eliminate them from our data collection.



# Qno3

## Part i

* • Analysis of brain MRI images for the diagnosis of Alzheimer's disease and related dementias by image processing was one of the earliest clinical data applications of deep learning, which had already proven successful in computer vision. In other areas of medicine Automatic cartilage segmentation and osteoarthritis risk assessment from low-field knee MRI images has been achieved by using CNNs . We found that our technique, which uses automatically selected 3D multi-scale characteristics, outperformed the state-of-the-art approach.while only employing 2D photos. Separating MS lesions in multi-channel 3D MRI and distinguishing benign from malignant breast lumps in ultrasound pictures are two other examples of how deep learning has been put to use in the field of medicine. Biopsy-proven clinical pictures of skin cancer (keratinocyte carcinomas vs. benign seborrheic keratoses and malignant melanomas vs. benign nevi) were classified by CNNs, and the results were comparable to those of 21 board-certified dermatologists working with a dataset of 130 000 images (1942 biopsy-labeled test images)
* Structured (diagnosis, medications, lab results) and unstructured data (medical history, patient demographics) found their way into the hands of deep learning systems in recent years, allowing for the processing of aggregated EHRs. Most of these studies used deep architecture to handle electronic health records inside a healthcare system in order to do some kind of supervised prediction clinical job. Specifically, one typical strategy is to demonstrate that deep learning outperforms traditional machine learning models on a variety of measures, including AUC, accuracy, and F-score. Most published research uses entire supervised networks, although some use unsupervised models to create latent patient representations and test those using shallow classifiers.
* Capturing the underlying structure of ever-growing, high-dimensional data sets is the goal of deep learning applications in high-throughput biology (e.g. DNA sequencing, RNA measurements). When compared to more conventional models, deep models perform better, are easier to comprehend, and provide more insight into the structure of the biological data. In the literature, a variety of works have been offered. Here, we provide a high-level overview of the concepts, while directing interested parties to for more reading. The initial neural network applications in genomics used deep architectures in lieu of traditional machine learning without modifying the input characteristics. A fully connected feed-forward neural network was utilised by Xiong et al to forecast the splicing activity of specific exons. More than a thousand characteristics were used to train the model, all taken from the candidate exon and neighbouring introns. This methodology was able to predict splicing activity with better accuracy than simpler methods and to pinpoint uncommon mutations that contributed to splicing misregulation.

## Part ii

### Opportunities

In healthcare, the only reason deep learning systems are valuable is if they help improve accuracy and/or efficiency. The future of healthcare may lie in a symbiotic or even cyborg interaction between humans and machines. Even yet, we're still at a point where robots and people are doing jobs that neither is particularly good at. As deep learning systems continue to advance and improve, they will become more useful in helping people with activities at which they struggle. As an example, humans excel at interpreting data obtained by their senses, especially vision. They have a remarkable capacity for understanding what other people are feeling. Humans, however, have limitations in areas such as short-term memory, data retrieval and organisation, and logical deduction based on collected facts. Consequently, I expect DL systems will make doctors and other providers more efficient and accurate in their diagnoses and choices, saving money and lives while cutting down on risks and delays.

# References

* Bishop, C. M. (2006) [Pattern Recognition and Machine Learning](http://research.microsoft.com/en-us/um/people/cmbishop/prml/). Chapter 5: Neural Networks.
* Schmidhuber, J. (2015). [Deep Learning in Neural Networks: An Overview](http://www.sciencedirect.com/science/article/pii/S0893608014002135). Neural Networks 61: 85-117.
* Bengio, Y., LeCun, Y., Hinton, G. (2015). [Deep Learning](http://www.nature.com/nature/journal/v521/n7553/full/nature14539.html). Nature 521: 436-44.
* Goodfellow, I., Bengio, Y. and Courville, A. (2016) [Deep Learning](https://github.com/HFTrader/DeepLearningBook). MIT Press.
* LeCun, Y., Bengio, Y. and Hinton, G., 2015. Deep learning. *nature*, *521*(7553), pp.436-444.
* Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning* (Vol. 1). Cambridge, MA, USA: MIT press.
* Yan, L. C., Yoshua, B., & Geoffrey, H. (2015). Deep learning. *nature*, *521*(7553), 436-444.
* Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends® in signal processing*, *7*(3–4), 197-387.
* Rusk, N. (2016). Deep learning. *Nature Methods*, *13*(1), 35-35.
* Hao, X., Zhang, G., & Ma, S. (2016). Deep learning. *International Journal of Semantic Computing*, *10*(03), 417-439.