DNA Sequences Classification with Deep Learning: A Survey

Samia M. Abd –Alhalem
Department of Electronics and Electrical Communications Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf

El-Sayed M. El-Rabaie
Department of Electronics and Electrical Communications Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf

Naglaa. F. Soliman
Faculty of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia

Salah Eldin S. E. Abdulrahman
Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf

Nabil A. Ismail
Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf

Fathi E. Abd El-samie
Department of Electronics and Electrical Communications Engineering, Faculty of Electronic Engineering, Menoufia University, Menouf

Abstract-**Deep learning (DL) methods have been achieving amazing results in solving a variety of problems in many different fields especially in the area of big data. With the advances of the big data era in bioinformatics, applying DL techniques, the DNA sequences can be classified with accurate and scalable prediction. The strength of DL methods come from the development of software and hardware, such as processing abilities graphical processing units (GPU) for the hardware and new learning or inference algorithms for the software, which reducing the main primary difficulties that faced the training process. In This work, we start from the previous classification methods such as alignment methods pointing out the problems, which are face to use these methods.After that, we demonstrate deep learning, from artificial neural networks to hyper parameter tuning, and the most recent state-of-the-art DL architectures used in DNA classification. After that, the paper ended with limitations and suggestions.**

1. Introduction

Biochemical molecules such as Deoxyribonucleic acid DNA, Ribonucleic acid RNA, and protein are fundamental for cellular organization [1, 2], they are responsible for many biological processes of any given organism, by playing a regulated organism's role to control different pressures during an organism's life time. DNA can be viewed as the blue print for cell machinery. RNA help carry out this blueprint during organism life. Usually DNA consists of a long linear chains of subunits called nucleotides[3] that is a long but finite string over the 4 letter "nucleotide" alphabet {A; C; T; G}(e.g. of DNA sequence ….ATCGCTGA ...). The Central Dogma of molecular biology is shown in figure 1, and it was first proposed by Francis Crick [2].

Much of the biological information can be provided by molecular biology, such information is related to the analysis of biochemical molecules (DNA, RNA, and Protein), these information help molecular biologists to solve many problems. The task of extracting this kind of information is not an easy process. However, even if we can get this information, biologists cannot retrieve any useful knowledge due to the quantity and the complexities of this data. For this, the merging between biology, informatics and mathematics for analyzing these biological data happened in order to achieve the many goals of the new discipline called bioinformatics [4]. One of the main tasks of bioinformatics are the computational analysis of sequence databases. The biological sequences classification is extremely useful for this task, based on the principle that sequences having similar structures have also similar functions. Since when sequencing a new genome, perhaps its function and structure are among the most important questions. To determine them, a newly sequenced is compared to well-known databases via a similarity function. Then predict which group the new sequence belongs to [5]. There are many well-studied, more specific problems are examples of this one: for instance, the taxonomic problem where the phylogenetic group of an organism can be known using given some of its biological sequence data and that is the focus of this paper.

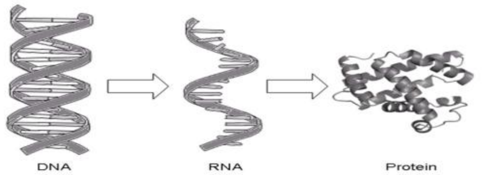

Figure 1: The Central Dogma of molecular biology [2].

In the 18th century Linnaeus, a Swedish naturalist, classified living things according to a hierarchy: Kingdom, Phylum, Class, Order, Family, Genus, Species, and his classification was based on an observed similarity and widely reflects biological ancestry (taxonomy) [6]. As a result, Taxonomy is defined as science of Classification of organisms, hopes to show relationships among organisms, a way to provide universal identification of an organism.

Currently, the issue of DNA sequence classification at all taxonomic levels are usually performed by alignment-based, alignment-free techniques and combination between machine learning and digital signal processing (ML-DSP) , as a common-goal to alignment-free method and software tool [7, 8, 9, 10, 11, 12,13]. This combination (ML-DSP) implemented to overcome the difficult which face the previously alignment methods. The main contribution of ML-DSP is the feature vector that used for the supervised learning algorithms. The main problem behind this method remains how to find the best feature selection process. Since, the letter / character-based description of a genome sequence is readable and understandable for humans [13] but is an issue for machines, and the commonly used representations introduce the main drawback of the high dimensionality. Recently, DL architectures models were proved to be able to extract automatically useful features from input data and the power of deep learning in area of big data has facilitated major advances in numerous bioinformatics applications such as genomic medicine, medical imaging research field and sequences classifications. Artificial neural networks (ANNs), one of the most widely used models in ML [14]. Classifiers based on ANNs are loosely based on the brain's neural network. These type of algorithms are behind most state of the art applications in artificial intelligence (AI) and are chosen when dealing with very large datasets. They are also the main architecture behind DL techniques.

In order to know DNA classification tools, a survey study is desired. The main target of such review study is to give a collective survey of DNA classification techniques applicability that could be helpful in medical, legal and social applications. Several studies have treated this issue from different perspectives. Some of them are mentioned. A survey study has been presented by Andrzej Zielezinskiet. al. [15]. The research provides a survey study for the benefit, applications, and tools of Alignment-free sequence and the challenges that appear in alignment based method and how alignment based method faced these challenges.

Jie Renet. al. also introduced another review study entitled by "Alignment-Free Sequence Analysis and Applications" in 2018 [16]. This survey study was provide an updated review of these applications and related developments of word count–based approaches for alignment-free sequence analysis.

Another review study of "Deep learning for computational biology" for Christof Angermueller, Tanel Pärnamaa, Leopold Parts & Oliver Stegle [10]. The study discuss applications of deep learning approaches in regulatory genomics and cellular imaging. Moreover in 2017 [12], Deep Learning Architectures for DNA Sequence Classification by Giosue Lo Bosco and Mattia Antonino Di Gangi. They show the main advantages for DNA Sequence Classification based on Deep Learning. Few numbers of researches have discussed the subject of DNA classification based on deep learning in different perspectives such as [17, 18, 19].

1.2 Work Objectives

The principle objective of this work is to presented DL based methods for classifying DNA sequences and how it works. This objective may be expressed in terms of a number of aims:

- Provide some basic biological definitions to build biological background for the reader.
- Review existing DNA classification methods such as alignment based method, alignment free methods and combination between DSP and machine learning and show the key issue that seriously limits these methods.
- Introduce the methods based on deep learning which used for DNA classification. Since deep neural network models represent the state of the art for pattern classification, this leads to a better classification of DNA sequences with respect to other classification schemes.
- We introduced two deep learning architectures, namely convolutional neural network (CNN) and recurrent neural networks (RNN). We chose these two algorithms because they are based on different computational models and usually used for DNA sequences classification.
- Introduce the efficacy and challenges of deep neural architectures regards to sequence classification in bioinformatics.

This paper is organized as follows: Section 1presents this introduction. Section 2 review the recently published classification methods of DNA sequences into phylum, class, order, family and genus level, which are broadly classified into three major groups: alignment –based methods , alignment free methods and DSP-ML. It also includes the comparison between different alignment methods in terms of their merits, demerits and their popular software. Moreover introduces the classification based on machine learning. It also includes how deep learning approaches work. Section 3 includes some limitations and suggestions. Finally, section 4 concludes the paper.

2. From Alignment Methods to deep learning

In this section, firstly different alignment methods have been introduced and comparison between them. After that, we demonstrate definition and common stages in DNA classification based on machine learning. In addition, some factors contributing for increasing popularity of are explained. Then introduce the components of the convolutional neural networks (CNNs) and recurrent neural networks (RNNs) that mainly used for DNA sequences classification. Finally, we provides example of applying CNNs to classification of bacterial species.

2.1 Alignment –based Methods

Alignment-based techniques are all techniques based on the search for base-to-base matches in two or more sequences [5, 8]. These techniques evaluate sequence similarity, by computing a score based on the amount of matches and mismatches between sequences. In this way, they may compute the class of a given query sequence by locating the most similar sequence in the known set. For example, let x=TCCTGCCTCTGCCAATC and y=TCGGTGCATCTGCAATC be two nucleotide sequences. One available alignment between the x and y sequences is:

```
TCCTGCCTCTGCCAATC----
||::::::|::|||::|||||||
TCGGTGCATCTGCAATC----
```

Here the vertical lines between the letters denote the match while colons represent substitutions. There have been many alignment-based tools developed, including single-sequence aligners such as BLAST [20] and FASTA [21] and multiple-sequence aligners such as ClustalW [22] and MUSCLE [23].

2.2 Alignment Free Methods

Next Generation Sequencing technologies (NGS) are producing vast quantities of genomic data. Unlike the older dideoxy-based sequencing methods, e.g. the Sanger method, NGS fragments are much shorter and more error-prone. Earlier methods for analyzing genomic sequence data are generally challenged by the smaller fragment sizes and error-rich data produced from NGS machines. Alignment-based approaches do not use location information for a sequence within the genome. This means, comparisons of sequences based only on alignments. Sequence location is irrelevant to the problem of substring alignment; the possibility of conserved regions shifting in the genome due to recombination or interactions over large distances in the genome is ignored when looking for substring matches.

Alignment-free methods can be divided into five categories: a) methods based on k-mer/word frequency, b) methods based on the length of common substrings, c) methods based on the number of (spaced) word matches, d) methods based on micro-alignments, e) methods based on information theory and f) methods based on graphical representation. Alignment-free approaches have been used in sequence similarity searches [25], clustering and classification of sequences [26] and more recently in phylogenetic [27, 28].There are alignment-free software applications, including COMET [29, 30], CASTOR [31] and FFP (Feature Frequency Profile) [32].

Table 1 summarizes Comparison between different alignment methods in terms of their merits and demerits. It indicates that the key issue that seriously limits the application of the alignment methods remains their time computational complexity. Some of most popular programs based on alignment method are presented in table 2.

Alignment-free sequencing processing techniques have arisen as a method to evaluate genomic data when re-assembly of the fragments is unfeasible or impossible [24].

Table 1: Merits and demerits of different alignment methods.

Method	Merits	Demerits
Alignment based methods	• Can report the exact regions of high similarity between a pair of sequences, their output is highly relevant to researchers and can be used to study functional, structural, or evolutionary relationships between sequences.	-The high time-memory computational cost for multiple alignment in multi-genome sequence data. -There is a need for continuous homologous sequences and the reliance on a priori assumptions , e.g., the gap penalty -Many brief reads from separate components of the genomes may not always be aligned
Alignment free methods	-It has arisen as a method to evaluate genomic data when re-assembly of the fragments is unfeasible or impossible. -overcome some of the Demerits faced by alignment based methods	- Lack to implement software: Most of the current alignment-free techniques continue to explore technical foundations and lack software execution needed to compare techniques on prevalent datasets. -Most current alignment free techniques are tested utilizing simulated sequences or very small real-world datasets. Due to this selecting one tool over the others difficult for experts. -Memory overlap: Scalability to multi genome information can create memory overhead, particularly when using word-based methods with long k-mers.

2.3 Machine Learning with Genomic Data

Because of its high classification precision, accelerate and scalability to big datasets, ML-DSP is extremely useful for freshly found species classification, identification of genomic signatures, and in the assessment of genome integrity [13]. In addition, to overcome alignment methods challenges they faced. The basic idea behind ML-DSP methods is used the methods of alignment free method as feature vectors then used machine learning as classifier.

There are distinct machine learning methods that are used for classification objectives that achieved high accuracy in binary and multiple classification of genomic sequences. The authors in [34] presented a deep review about deep learning in bioinformatics. Lots of evidence show that Recurrent Neural Networks (RNN) and CNN mainly used in the taxonomy of deep neural models [12, 34, 35]. Some of examples for ML-DSP are, in [7], the authors used Support Vector Machine (SVM) and CNN as classifier.

2.3.1 Classification based on Machine Learning

Machine learning is a part of artificial intelligence (AI) based on the concept that machines should learn through experience [36]. Computers could use pattern recognition through machine learning to discover hidden meanings without candid programming, In comparison to the traditional biological computational methods. The primary concept behind machine learning is to build a model with an acceptable quantity of information, regardless of the instructions to overcome the issue, which can produce valuable predictions of the solutions. In consequence, machines learning is about various models, which use various methods to learn by adapt and improve their result from experience [37, 38] and deep learning is considered the new trend used now especially with big data. Figure 2 visualizes the distinction between AI, ML and DL where DL is a subset of ML, which is also a subset of AI.

Table 2: Some popular Algorithms for searching sequence similarities based on alignment methods.

Method	Software
Alignment based methods	•FASTA[21]: performs a Pearson and Lipman search for similarity between a query amino acid sequence and any group of amino acid sequences that either reside in the user's computer or is a database. An extension to this program TFASTA does the search of the amino acid sequence on any group of nucleotide sequences. TFASTA translates the nucleotide sequences in all six frames before performing the comparison therefore searching also all "implied amino acid sequences." •BLAST [20] (Basic Local Alignment Search Tool): the heuristic search algorithm employed by the programs blastp (for comparison of an amino acid query sequence against a protein sequence database), blastn (for comparison of a nucleotide query sequence against a nucleotide sequence database), blastx (for comparison of the six-frame conceptual translation products of a nucleotide query sequence (both strands) against a protein sequence database), tblastn (for comparison of a protein query sequence against a nucleotide sequence database dynamically translated in all six reading frames (both strands), and tblastx (for comparison of the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database). •Multiple Sequence Alignment a collection of alignment programs such as CLUSTALW [22], for aligning nucleic acid or amino acid sequences. Maintained by the Department of Genetics at the University of Washington, Seattle.
Alignment free methods	•CoMet [30]: Universe server allows you to analyze the taxonomic and functional composition of your metagenomic sample and to compare it with a large collection of publicly available data from previous metagenome studies. •CASTOR [31] a virus classification platform, based on machine learning methods. It is inspired by a well-known technique in molecular biology: restriction fragment length polymorphism (RFLP). It simulates, in silico, the restriction digestion of genomic material by different enzymes into fragments.

Classification is the activity of examining the features of an object and assigning it to a predefined set of classes based on supervised learning [38]. The input data and output data are already known in supervised learning and a learning algorithm is used to learn how to map the input to the output. Classification task consist of four main stages, preprocessing, feature extraction, training and lastly classification. Figure 3 shows common classification stages based on machine learning.

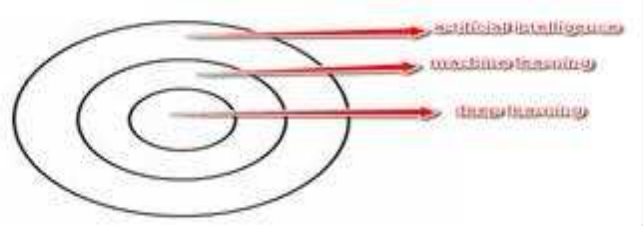

Figure2: The relationship between DL, ML and AI.

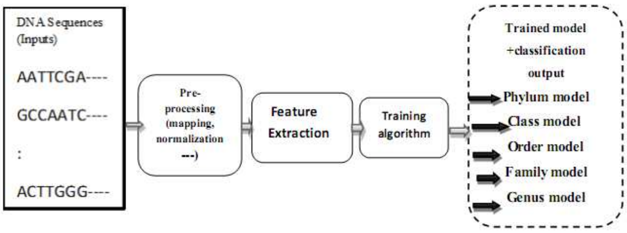

Figure 3: Common stages in DNA classification based on machine learning.

- Preprocessing: In preprocessing stage, raw data, which is gathered from source, handled before employed as an input.This stage is essential because the information acquired in most instances are noisy, incomplete or /and inconsistent. Preprocessing involves data-related tasks such as cleaning, transformation, reduction and so on.
- Feature Extraction: Next phase is feature extraction. Features are the domain-specific measurements, which have relative information to create the best possible representation of the input. For classification, most appropriate characteristics must be obtained from raw data.
- Training: In this phase as also explained before, model trains to give the most accurate Prediction as possible.
- Classification: In classification, the model produced assign input to one of the classes depending on the decision rules.

In every machine learning approaches, the training process is the most important phases. This training ultimately determines the utility of a model and how it works when properly tested. There are, however, frequent known issues that can arise when training a model from scratch that yielded poor performance such as Vanishing gradient, Over fitting, Computational load. These problems are the reasons for increasing popularity of deep learning [39, 40] than other machine learning approaches. Because of the development of software and hardware, such as processing abilities graphical processing units (GPU) for the hardware and new learning or inference algorithms for the software, which reducing the main primary difficulties in the training process. Table 3 introduced the difference between Ml and DL.

2.3 Deep Learning (DL)

Deep Learning (DL) can be briefly explained as a machine learning subfield that works with algorithms that structurally and functionally resemble a brain. DL is a very broad subject, having many distinct concepts and notions that need to be understood before being put to practice. In the next subsections, we will explain and give examples of deep neural networks and how they work[37]. In additions, why are deep neural networks garnering so much attention now?

2.3.1 Some Factors Contributing for Increasing Popularity of Deep Learning

- The availability of large training data sets with high-quality labels.
- Progress in parallel computing implementations.
- Niche software platforms such as PyTorch [41], Tensorflow [42], Caffe [43] , Chainer [44], Keras [45], BigDL [46] etc. that allow seamless integration of architectures into a GPU computing framework without the complexity of addressing low-level details such as derivatives and environment setup.
- Robust optimization algorithms that produce near-optimal solutions: Algorithms with adaptive learning rates (AdaGrad, RMSProp, Adam,Adaboost), Stochastic Gradient Descent (with standard momemtum or Nesterov momentum), Particle Swarm Optimization, etc.

2.3.2 Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANN), like the brain's neural network, have specific structures connected between each other. In ANNs, the nodes can be seen as the neurons. Each neuron connects and interacts with at least one another through links. The inputs are processed and passed along through the network, eventually reaching a final state where all the computed values are shown to the user [47]. Where each artificial neuron receives a set of inputs, with each of this inputs being subsequently multiplied by its corresponding weight. After all the inputs are processed, the values are combined together and added resulting in an intermediate value called sum or net. The sum is then passed through an activation function that produces the final output of that single node. This output serves as input for one or more connected nodes in the next layer, where they will use that value in a similar manner for its inner calculations.

Each artificial neuron receives a set of inputs, with each of this inputs being subsequently multiplied by its corresponding weight. After all the inputs are processed, the values are combined together and added resulting in an

Table 3: The difference between machine learning and deep learning.

Classifier	Machine learning	Deep learning
By definitions	Algorithm can parse, learn from the data and then apply the same to informed decision making. Where this algorithm needs to be told how to make an accurate prediction by providing it with more information, this means selection which features are suitable manually (feature extraction stage) to do accurate prediction.	Algorithm can parse, learn from the data and then apply the same to informed decision making. Where this algorithm can providing accurate prediction automatically. There is no feature extraction stage.
Management	The various algorithms are directed by the analysis to examine the different variables in the datasets.	The algorithms are generally self-directed for the appropriate data analysis once they are applied.
The output	Usually the output is a numerical value, such as a score or classification.	The output can be anything from a score, an element, free text, etc.
Number of data points	Usually, there are a few thousand data points used for analysis.	There are a few million thousand data points used for analysis.

intermediate value called sum or net. The sum is then passed through an activation function, which produces the final output of that single node. This output y can be represent as follow:

$$y = \emptyset(v) \qquad (1)$$

$$v = x_1 w_1 + x_2 w_2 + \cdots + x_m w_m + b \quad (2)$$

Where v is the weighted sum of the input signals and $\emptyset(.)$is the equation of the activation function.

This output serves as input for one or more connected nodes in the next layer, where they will use that value in a similar manner for its inner calculations. Figure 4 shows behavior inside each artificial neuron. The detail of ANN in [47].

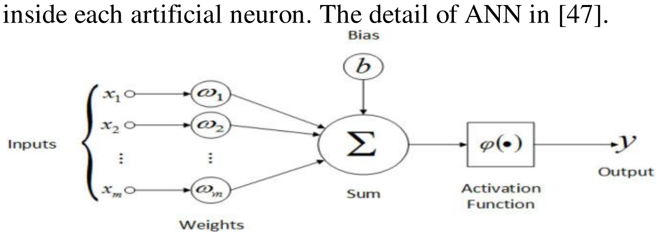

Figure 4: Behavior inside each artificial neuron.

2.3.3 Training process algorithm

Training an ANN is usually done with a method called back propagation. The goal of this technique is to optimize the weights of the nodes in the network by calculating a concept called gradient. The following steps can describe the algorithm:

1. Weight initialization at the input layer.
2. Forward propagation of the weights in the network with each node using its inputs and associated weights to calculate the activation values.
3. Calculation of the loss function at the output layer.
4. Backward propagation, where the gradients of the loss function are calculated and each layer
specific parameters are updated.
5. Repeat the steps 2, 3 and 4 until the stop criteria is met, usually until the loss function is minimized without achieving over fitting.

Deep neural network learns by minimizing the loss function by changing the parameters (weigts and biases) of the model in training. Stochastic gradient descent (SGD) is one of the most common used techniques to learning the parameters. The gradients of a loss function are calculated by using the back-propagation (BP) algorithm which explained previously, then the results fed to the SGD method to update the parameters (weights and biases) incrementally after each epoch. An epoch indicates the number of times all of the training input used to update the parameters. All training samples get through the leaning phase in one epoch before parameters are updated. SGD calculates the approximation of the true error gradient error based on a single training sample, instead of compute the gradient of the error based on the all training sample like in gradient descent (GD). Thus, DNN can train faster with SGD, since calculating the approximation is faster.

$w_j \leftarrow w_j + \Delta w_j (3)$

After each epoch, weights in previous Equation, updated as:
$\Delta w_j = \alpha(target - output)x_j$ (4)
Where α is learning rate, w_j is weight from the input node j, target and output indicate target label and the final output of that node j, x_j is the input for connected node j.

2.3.4 Backpropagation

As explained previous, to use SGD in multi-layer networks, gradient of the loss function is needed to be computed. Backpropagation is the most common method used to overcome this problem. In backpropagation, calculating the partial derivatives ∂L/∂w of the loss function L with respect to some weight w is enough to analyze the change in the loss with the change of weights. Using mean squared error (MSE) as cost function one output neuron over all n examples is:

$$l = \frac{1}{2}\sum_{j=1}^{n}(t_j - y_j)^2 \qquad (5)$$

Where: t is target label, y is output of ANN. L is scaled by $\frac{1}{2}$ for mathematical convenience. Error gradient is calculated in following equation 6 to use SGD:

$$\Delta w_{kj} = -\alpha \frac{\partial L}{\partial w_{kj}} \qquad (6)$$

Where a node in layer k is connected to a node in layer j. The result is taking negative. We take the negative because the change of the weights are in the direction of where error is decreasing. Because weight changes are in the direction where the error is decreasing. Using chain rule gets us:

$$\frac{\partial L}{\partial w_{kj}} = \frac{\partial L}{\partial y_j}\frac{\partial y_j}{\partial x_j}\frac{\partial x_j}{\partial w_{kj}} \qquad (7)$$

In Equation 7, x_j is the weighted sum of the inputs being passed to j^{th}node andy_j = f (x_j) is the output of the activation function. In the end, gradient of the calculated error feed to the SGD algorithm.

2.3.5 Some Deep Learning Architectures.

This subsection illustrate the main component of more practical DL architectures that mainly used for DNA sequences classification; Convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Table 4introduced someof related papers.

Convolutional neural networks (CNNs)

Convolutional neural networks are types of discriminative connectionist models. They are originally designed to operate directly on observed images without pre-processing [48]. Although they have also been successfully applied to text and sound classification [49]. These networks have applications in the most diverse of real world areas from traffic sign recognition to empowering vision in self-driving cars [50], a CNN architecture created for simple recognition operations such as reading digits and zip codes, was created around 1990 and was pioneer in deep ANNs, rendering this type of networks as one of the reasons behind DL's popularity growth over the years. Models based on a CNN architecture can have many different forms, but are usually based on the four main tasks seen on Figure 5. These tasks are:

- ➢ Convolution, consisting in the feature extraction from the input data. This extraction is done by filters or feature detectors that perceive particular conditions (such as edge recognitionin the case of images) by scrolling through the data and producing feature maps.
- ➢ Introduction of non-linearity functions such as Sigmoid or ReLU after each convolution.
- ➢ Sub-sampling or pooling is used to decreasing the feature maps. This leads to more manageable representation of information and a decrease in the amount of parameters, controlling potential overfitting.
- ➢ Fully connected layer, an ANN with a non-linear activation function that utilizes the sub-sampled results to show a higher-level classification of the learned data.

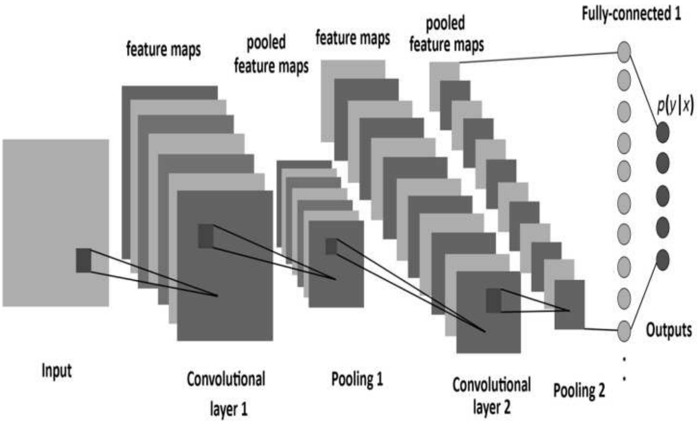

Figure 5: Architecture of a convolutional neural network.

Table 4: Some of Related Papers

Ref.	Summary
[10]	The study discusses applications of deep learning approaches in regulator genomics and cellular imaging.
[12]	It discusses DNA classification based on RNN and CNN, as well as show the main advantages for DNA Sequence Classification based on Deep Learning.
[7]	It focuses on the applications of CNN in DNA classification, as well as using FCGR as data representation.
[18]	It discusses the applications of CNN based on different subsampling layers into DNA classification
[11]	A general review of the CNN and RNN architectures, with the applications in omics, image processing and signal processing.

- Recurrent neural networks (RNNs)

Another type of architecture is RNN, and is particularly used in problems with sound or sequential data, such as speech recognition, natural language and sentiment analysis [51]. Figure 6 shows a basic RNN architecture. The layer disposition is very similar to FFNNs, with an input layer followed by a hidden layer and an output layer. RNNs are dynamic networks; with their state changing continuously until an equilibrium is reached. They mainly differ from FFNNs because they allow feedback between nodes, with each hidden node's computation being a combined calculation of the input value and the information produced in previous nodes.

There are many RNN variants. Examples include an independently recurrent neural network or IndRNN, a special type of RNN which solves the vanishing gradient issue that usually appears in more basic forms. A long short-term memory or LSTM is another form of RNN that solve the issue of vanishing gradients by incorporating components called "gates". Yet another variation called Hopfield network in which all connections between nodes are bidirectional, is used in the studying of many mathematical known problems such as the traveling salesman.

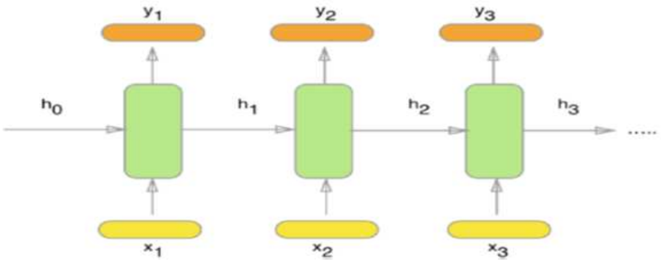

Figure 6: Architecture of a recurrent neural network.

2.4.6 Hyperparameters Tuning

Once a deep learning architecture is selected, many parameters are subsequently represented to set, including the number of epochs, the number of iterations, the number of mini-batch, optimizing parameters and the learning rate, all of which remarkably influence the results [52].Multiple epochs and iterations are needed for training DNNs. Learning Rate; this value determines how quickly a network updates its hyper parameters. High learning rates speed up the training process but can result in gradient convergence becoming more difficult. On the other hand, slower learning rates will ensure a smooth convergence while drastically increasing the time needed to learn. An epoch is defined as one total pass of the training data through the network. The minibatch size is the number of samples received by the network before the parameters are updated. However, multiple iterations are generally used when training small data sets. If one sets too few epochs, not enough time is given to train DNNs and the training results may not be reliable. If too many epochs are set, an over fitting problem might occur during the training process. So far, there is no automatic means to configure the number of epochs. The rule-of-thumb is to monitor the progress and use early stopping, which can stop the training at the early stage and prevent the neural network from overfitting. Finally, SGD is usually used method for update optimization. Although SGD can create instability problems for DNNs, it is widely used because of its simplicity. Layer-wise training for DNNs was created in order to solve the vanishing gradient problem caused by SGD. Alternatively, other optimization algorithms which is to combine the SGD with some optimizer algorithms, such as Momentum, RMSProp, and Adam [53, 54], etc. Using these optimizer algorithms to train DNN results in faster training than when using a traditional SGD.

2.4.6 DNA sequences Classification with CNN

Currently, there are few studies have been provided for the DNA sequence classification problem and shown the success of using deep learning [7, 10, 11, 12].In this example, CNN is used to classify bacterial into taxonomic levels. The dataset download from the Ribosomal Database Project, Release 11 [55].DNA sequence string have been encoded into 2D data (image) which can be fed to the model as input.FCGR are used for mapping with suitable k-mers as discussed in [18]. We implement CNN, as shown in Fig. 7 with specific number of hidden layers. We first trained these models for each taxa. Finally, we run the standard optimization procedure with best Hyper parameters as discussed in section 2.4.6, cross-entropy loss and Adam optimizer. In order to achieve high classification, we can use different subsampling layers such as random projection and wavelet pooling [18]. Also combination between CNNs and RNN. This example can be easily adopted for other classification applications [56], such as breast cancer classification [57]and CT image classification [58].

3. Limitations and Suggestions

- Data Representation: there are some problems in quantifying aspects of DNA sequences. No one knows which representation is the best for encoding numeric values in these nucleotides. However, we cannot avoid using the numeric representations of these biological units when applying learning machine to biological studies. Future directions may include more accurate identification of DNA signals using multidimensional DSP-based features, and combining signal processing based work with data-driven methods to advance the state of the art in DNA sequences classification algorithms.
- Reducing computational requirement: deep learning models are usually very complex and have lots of parameters to be trained, it is often computationally demanding and memory intensive to obtain well-trained models and even for the productive usage of the models [59]. Those requirements seriously limit the deployment of deep learning in machines with limited computational power, especially in the field of bioinformatics and healthcare, which is also data intensive. Several methods have been proposed to compress the deep learning model, which can reduce the computational requirement of those models from the beginning such as parameter

pruning, which reduces the redundant parameters that do not contribute to the model's performance significantly, including the famous Deep Compresion. In addition, we can use compact convolutional filters to save parameters [60].

- Hybrid methods: refer to combination between different deep learning models such as CNN used as feature extraction stage with RNN classifier to adding additional knowledge, which can indeed provide better results. Which may provide new different viewpoints and fine insights to help understand of the genome nature. This might be the coolest trend for Law enforcement agencies that are using DNA data classification to solve crimes at unprecedented speeds.
- Implementation of different optimization techniques to achieve the best performance and perfect DNA classification.
- The major improvement would have to be in our working environment. With achievements such as GPU usage for calculations or the integration of our machine in a cluster of other computers, we could improve our hyperparameters such as a reduced learning rate, train additional state of the art architectures or dramatically increase the size of our dataset. All of this could result in better outcomes than those attained.
- More attempts at extracting other features from DNA sequences before feeding them to the models could also be made. This could result in overall more accurate evaluations.

4. Conclusions

There are many studies associated to the analysis of biologic al data and solutions to these issues. Different methods such as alignment method and alignment free methodand combination between machine learning and digital signal processing have become more and more popular in this area. Deep learning stands out among rest, due to its remarkable performance and reduction of the challenges that other methods faced. One weakness of deep learning that should be considered is its need of data. It really shines where there is high amount of data available to train. In this paper, we first reviewed all the alignment methods needed to do the DNA classification. We also provided the most common instruments for classifying DNA and the main features behind a few state of the art frameworks in DL are exploring.Finally, the limitations and corresponding suggestions of the DNA classification based on deep learning technology are pointed.

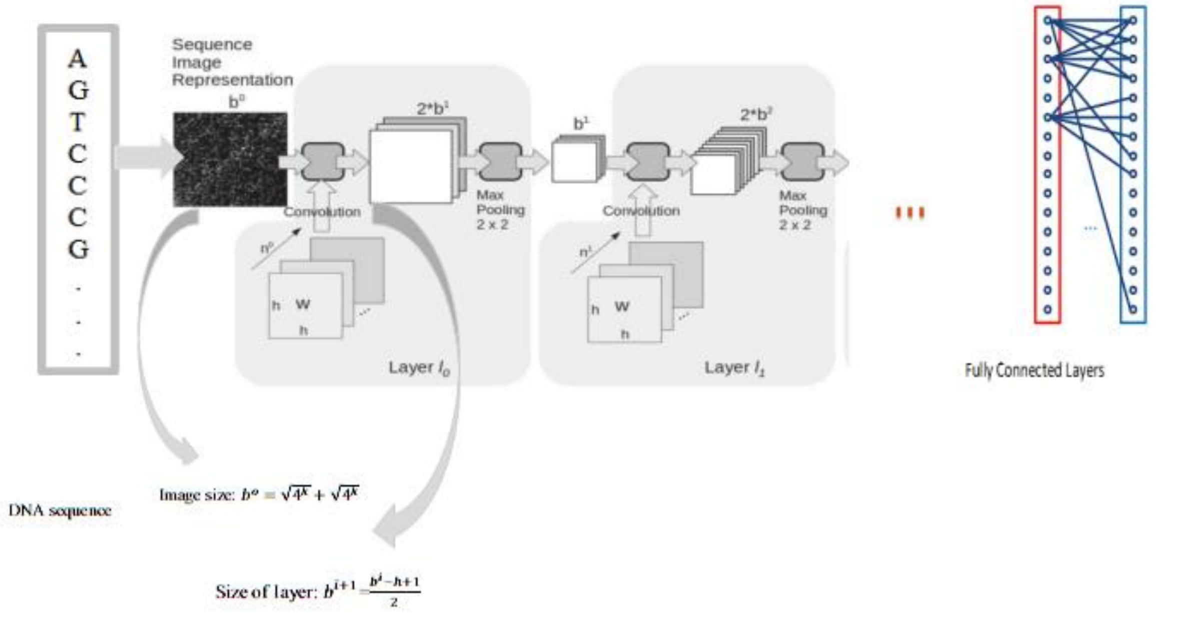

Figure 7: DNA Classification based on CNNs

References

[1] Anastassiou, D. "*Genomic Signal Processing*," IEEE Signal Process. Mag., pp: 8–20, Jun. 2001. DOI:10.1109/79.939833.

[2] Hartwell, L.; Hood, L.; Goldberg, M. L.; Reynolds, A.; Silver, L. M.; Veres, R. C. "*Genetics: From Genes to Genomes*," 2nd ed.; McGraw-Hill: New York, 2003.

[3] Aerssens, J.; Armstrong, M.; Gilissen, R.; Cohen, N. "*The Human Genome: An Introduction,*" Oncologist, pp: 100–109, June 2001.

[4] Xiong, J., "*Essential bioinformatics*", Cambridge University Press: pp: 318-362, 2006.

[5] Attila Kertész-Farkas "*Protein Classification in a Machine Learning Framework*" Ph.D. thesis, Research Group on Artificial Intelligence, University of Szeged, August 2008.

[6] URL: http://en.wikipedia.org/wiki/homology_(biology).(Acess date 11 July 2018).

[7] Riccardo Rizzo, Antonino Fiannaca, Massimo La Rosa, Alfonso Urso, "*Classification Experiments of DNA Sequences by Using a Deep Neural Network and Chaos Game Representation,*" International Conference on Computer Systems and Technologies - CompSysTech', Palermo, Italy, pp: 222-228, 23-24 June 2016.

[8] Susana Vinga, Jonas Almeida, "*Alignment-FreeSequenceComparison-Areview*," Bioinformatics, Vol: 19, Issue: 4, pp: 513–523, 1 Mar 2003.

[9] Genta Aoki Yasubumi Sakakibara, "*Convolutional Neural Networks for Classification of Alignments of Non-coding RNA Sequences,*" Bioinformatics, Volume 34, Issue 13, pp: i237–i244, 1 July 2018.

[10] Christof Angermueller1, Tanel Pärnamaa, Leopold Parts & Oliver Stegle1, "*Deep Learning for Computational Biology*" Molecular Systems Biology, Jul 29, 2016.

[11] Seonwoo, M., Byunghan, L., Sungroh, Y.: "*Deep learning in bioinformatics*," In: Briefings in Bioinformatics, 2016.

[12] Giosu´e Lo Bosco and Mattia Antonino Di Gangi, "*Deep Learning Architectures for DNA Sequence Classification,*" Fuzzy Logic and Soft Computing Applications, 11th International Workshop, Naples, Italy, pp. 162–171, 07 March 2017.

[13] GurjitS.Randhawa1, KathleenA.Hill andLilaKari "*ML-DSP:Machine Learning with Digital Signal Processing for Ultrafast, Accurate, and Scalable Genome Classification at all Taxonomic Levels*"Randhawaetal. BMCGenomics, 2019. Doi.org/10.1186/s12864-019-5571-y.

[14] Saptarshi Sengupta, Sanchita Basak, Pallabi Saikia, Sayak Paul, Vasilios Tsalavoutis, Frederick Ditliac Atiah,Vadlamani Ravi and Richard Alan Peter "A Review of Deep Learning with Special Emphasison Architectures, Applications and Recent Trends" IEEE Transactions, Mar. 2019.

[15] Andrzej Zielezinski, Susana Vinga, Jonas Almeida and Wojciech M. Karlowski," *Alignment-free sequence comparison: benefits, applications, and tools*" Zielezinski et al. Genome Biology (2017) 18:186. DOI 10.1186/s13059-017-1319-7.

[16] Jie Ren, Xin Bai, Yang Young Lu, Kujin Tang, "*Alignment-Free Sequence Analysis and Applications"* Annual Review of Biomedical Data Science, pp:13:23, 16 April 2018.

[17] Aoki Sakakibara, G. Y. "*Convolutional Neural Networks for Classification of Alignments of Non-Coding RNA Sequences*" Bioinformatics2018, 34, i237–i244.DOI:10.1093/bioinformatics/bty228.

[18] Samia M. Abd –Alhalem, Naglaa F. Solimanb, Salah Eldin S. E. Abd Elrahman, Nabil A. Ismail, El-Sayed M. El-Rabaie, and Fathi E. Abd El-Samie "Bacterial classification with convolutional neural networks based on different data reduction layers" Nucleosides, Nucleotides and Nucleic Acids, 16 Aug 2019. Doi.org/10.1080/15257770.2019.1645851.

[19] Angermueller C, Lee H, Reik W, Stegle, "*Accurate Prediction of Single Cell DNA Methylation States using Deep Learning*," Genome Biology, 2016.

[20] Stephen F. Altschul et al. "*Basic local alignment search tool*". In: Journal of Molecular Biology, pp. 403:410, Mar 1990.

[21] DJ Lipman and WR Pearson. "*Rapid and Sensitive Protein Similarity Searches*," In: Science, pp. 1435-1441, 227.4693 (1985).

[22] Julie D. Thompson, Desmond G. Higgins, and Toby J. Gibson. "*CLUSTALW: Improving the Sensitivity of Progressive Multiple Sequence Alignment Through Sequence Weighting, Position-Specific Gap Penalties and Weight Matrix Choice*". In: Nucleic Acids Research, pp. 4673-4680, 22.22 (1994).

[23] Robert C Edgar. "*MUSCLE: Multiple Sequence Alignment with High Accuracy and High Throughput*" In: Nucleic Acids Research, pp. 1792-1797, 32.5 (2004).

[24] C.-K. K. Chan, A. L. Hsu, S.-L. Tang, and S. K. Halgamuge, "*Using Growing Selforganising Maps to Improve the Binning Process in Environmental Whole-Genomeshotgun Sequencing*," BioMed Research International, vol. 2008, 2007.

[25] Miller, RT; Christoffels, AG; Gopalakrishnan, C; Burke, J; Ptitsyn, AA; Broveak, TR; Hide, WA. "*A comprehensive Approach to Clustering of Expressed Human Gene Sequence: the Sequence Tag Alignment and Consensus Knowledge Base*". Genome Research, 9 (11): 1143–55, 1999. Doi:10.1101/gr.9.11.1143. PMC 310831.

[26] Domazet-Lošo, M; Haubold, B "*Alignment-Free Detection of Local Similarity Among Viral and Bacterial Genomes*" Bioinformatics, 27 (11): 1466–72, 2011. Doi:10.1093/bioinformatics/btr176. PMID 21471011.

[27] Chan, CX; Ragan, MA. "Next-Generation Phylogenomics" Biology Direct. 8: 3, Jan 22, 2013. Doi:10.1186/1745-6150-8-3. PMC 3564786. PMID 23339707.

[28] Deschavanne PJ, Giron A, Vilain J, Fagot G, Fertil B. "*Genomic Signature: Characterization and Classification of Species Assessed by Chaos Game Representation of Sequences*" Mol Biol Evol. 1999;16:1391–9.

[29] Chenglong Yu et al. "*Real Time Classification of Viruses in 12 Dimensions*". In: PLoS One 8.5 (2013).

[30] Daniel Struck et al. "*COMET: Adaptive Context-Based Modeling for Ultrafast HIV-1 Subtype Identification*" In: Nucleic Acids Research 42.18, 2014.

[31] Mohamed Amine Remita et al. "*A machine learning Approach for Viral Genome Classification*" In: BMC Bioinformatics 18.208 (2017).

[32] G. E. Sims and S.-H. Kim, "*Whole-Genome Phylogeny of Escherichia Coli/Shigella Group by Feature Frequency Profiles (FFps),*" Proceedings of the National Academy of Sciences, 2011.

[33] Deschavanne PJ, Giron A, Vilain J, Fagot G, Fertil B. "*Genomic Signature: Characterization and Classification of Species Assessed by Chaos Game Representation of Sequences*," Mol Biol Evol.;16:1391–9, 1999 .

[34] Seonwoo Min, Byunghan Lee, and Sungroh Yoon, "*Deep Learning in Bioinformatics*" BriefBioinform., pp:851-869, Sep 2017.doi: 10.1093/bib/bbw068.

[35] Ngoc Giang Nguyen, Vu Anh Tran1, Duc Luu Ngo, Dau Phan1, Favorisen Rosyking Lumbanraja, Mohammad Reza Faisal, Bahriddin Abapihi, Mamoru Kubo, Kenji Satou "*DNA*

Sequence Classification by Convolutional Neural Network" J. Biomedical Science and Engineering, pp: 280-286,April 2016.

[36] Mitchell, T. M. "*Machine learning*," Burr Ridge, IL: McGraw Hill,45(37):870–877, 1997.

[37] Kim, P. "*MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence,*" Springer, Seoul, Soul-t'ukpyolsi, Korea (Republic of). DOI: 10.1007/978-1-48422845-6.

[38] Kotsiantis, S. B. "*Supervised machine learning, A review of Classification Techniques*," Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies, pp: 3–24. IOS Press, 2007.

[39] James Martens. "*Deep learning via hessian-free optimization*," In ICML, volume 27, pp: 735– 742, 2010.

[40] Douglas M Hawkins. "*The problem of Overfitting*" Journal of chemical information and computer sciences," 44(1):1–12, 2004.

[41] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch, in: NIPS-W, 2017.

[42] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S.Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kud-lur, J. Levenberg, D. Man´e, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker,V. Vanhoucke, V. Vasudevan, F. Vi´egas, O. Vinyals, P. Warden, M. Wat-tenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machinelearning on heterogeneous systems, software available from tensorflow.org(2015). URL http://tensorflow.org/.

[43] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22Nd ACM International Conference on Multimedia, MM '14, ACM, New York, NY, USA, 2014, pp.675–678. doi:10.1145/2647868.2654889.

[44] S. Tokui, K. Oono, S. Hido, J. Clayton, Chainer: a next-generation open source framework for deep learning, in: Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS), 2015.

[45] F. Chollet, et al., Keras, https://github.com/fchollet/keras (2015).

[46] J. Dai, Y. Wang, X. Qiu, D. Ding, Y. Zhang, Y. Wang, X. Jia, C. Zhang,Y. Wan, Z. Li, J. Wang, S. Huang, Z. Wu, Y. Wang, Y. Yang, B. She, D. Shi, Q. Lu, K. Huang, G. Song, Bigdl: A distributed deep learningframework for big data, CoRR abs/1804.05839.

[47] B Yegnanarayana. "Artificial Neural Networks," PHI Learning Pvt. Ltd., 2009.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "*Imagenet Classification with Deep Convolutional Neural Networks*," In Advances in neural information processing systems, pp: 1097–1105, 2012.

[49] Yann LeCun et al. "*Lenet-5, Convolutional Neural Networks*," URL: http://yann.lecun. com/exdb/lenet, 2015.

[50] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. "*Speech Recognition with Deep Recurrent Neural Networks,*" In Acoustics, speech and signal processing (icassp), IEEE international conference on, pp: 6645–6649, 2013.

[51] Stephen Grossberg. "*Recurrent Neural Networks*," Scholarpedia, 8(2):1888, 2013.

[52] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in Neural Networks:Tricks of the Trade. Heidelberg: Springer, 2012, pp. 437-478.

[53] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochasticoptimization," Journal of Machine Learning Research, vol. 12, pp. 2121-2159, 2011.

[54] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014 [Online]. Available: https://arxiv.org/pdf/1412.6980.pdf.

[55] https://rdp.cme.msu.edu.(accessed May 11, 2018).

[56] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud ArindraAdiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram Van Ginneken, Clara I. Sánchez, "A survey on deep learning in medical image analysis", Med. Image Anal.,pp: 60–88, 42 (2017) .

[57] Fabio A. Spanhol, Luiz S. Oliveira, Caroline Petitjean, Laurent Heutte, "A dataset for breast cancer histopathological image classification," IEEE Trans. Biomed. Eng., pp: 1455–1462, 63 (7) (2016).

[58] Devinder Kumar, Alexander Wong, David A Clausi, Lung nodule "Classification using deep features in CT images", Computer and Robot Vision (CRV), 2015 12th Conference on, IEEE, pp: 133–138, 2015.

[59] Yu Cheng, Duo Wang, Pan Zhou, Tao Zhang, "A survey of model compression and acceleration for deep neural networks", 2017. arXiv:1710.09282.

[60] Taco Cohen, Max Welling, Group equivariant convolutional networks, International Conference on Machine Learning, 2016, pp. 2990–2999