DNA/Gene classification using RNN Sequential analysis

Problem Statement

Gene classification using RNN is the problem of identifying the functionality of genes using only the sequence information (ATGTGT…..) automatically. This problem can be addressed by using RNN (Recurrent Neural Network) which will monitor the sequence and provide meaningful information. They can incorporate contextual information from past inputs, with the advantage to be robust to localized distortions of the input sequence along the time. In this research a RNN based network is used for the DNA/gene classification. RNN is a type of recurrent neural networks with a more complex computational unit that leads to better performance. All RNN model in this research are build using tensorflow python package. Recurrent Neural Networks are generally used for processing sequences of data which evolves along the time axis. But the main challenge behind the problem remains the feature selection process. Sequences do not have explicit features, and the commonly used representations introduce the main drawback of the high dimensionality. For sure, machine learning method devoted to supervised classification tasks are strongly dependent on the feature extraction step, and to build a good representation it is necessary to recognize and measure meaningful details of the items. The concept of multi-task learning affects the models, both in terms of performance and training time.

Background Work: One of the primary goal in biology is the understanding of the relationships between protein structure and function. To understand the structure-function paradigm, particularly useful structural information comes from the primary amino acid sequences. DNA sequence classification is extremely useful for this task, following the principle that sequences having similar structures have also similar functions. Sequence similarity is traditionally established by using sequence alignment methods, such as BLAST. Despite of the challenges given in the problem statement is still an active area of research. Numerous approaches have been proposed over the years.

In traditional approach for DNA/gene classification. DeepSEA a tool performs prediction based solely on a fixed-length genomic sequence. It used a deep convolution neural network over an input sequence, alternating between convolution and pooling layers to extract sequence features. The use of sigmoid output layer to compute the probability of seeing a particular genomic feature is established. DeepBind also used a deep CNN with 4 stages of convolution, rectification, pooling, and some nonlinearity. Unlike for DeepSEA, DeepBind accommodated varying-length inputs and also incorporated extra information in addition to just the input sequence. This extra information came from known facts about a particular sequence gathered from other experiments Other experiments also used a deep CNN architecture to learn the functional activity of genomics sequences. There is also a bidirectional long-short-term-memory (LSTM) RNN on top of the outputs of a max pooling layer after convolution on the input sequence. Each of these tools fed a one-hot encoding of an input sequence into a convolution layer.

Methodology

1. **Data Collection and Dataset Preparation**: The genomic sequences in the dataset is from 16S dataset. Images in the dataset were grouped into five different classes.

2. **Character Embedding**: Character embedding is done by character level one hot encoding. This representation considers each character *i* of the alphabet by a vector of length equal to the alphabet size, having all zero entries except for a single one in position *i*. This method leads to a sparse representation of the input, which is tackled in the NLP literature by means of an embedding layer.
3. **Training:** Training the deep convolution neural network for making an image classification model is done. CaffeNet architecture is used and adjusted to support our 15 categories (classes). Rectified Linear Units (ReLU) are used as substitute for saturating nonlinearities. This activation function adaptively learns the parameters of rectifiers and improves accuracy at negligible extra computational cost.
4. **Testing:** In this phase the author used the test set for prediction of gene sequence category.

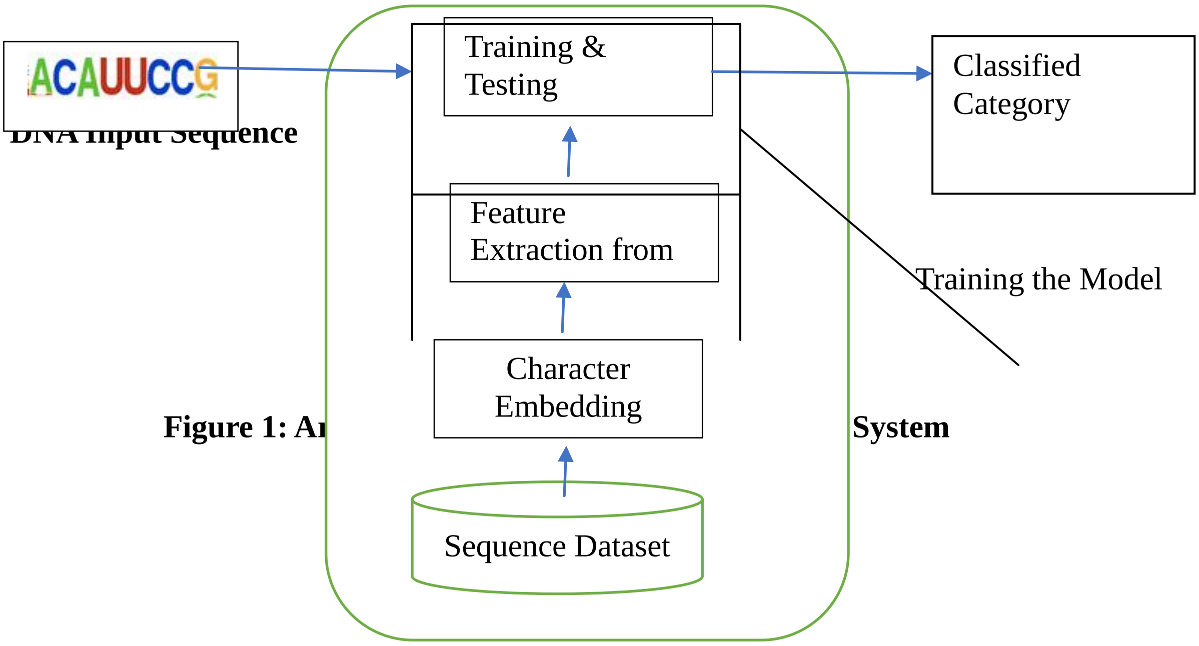

Figure 1: Ar **System**

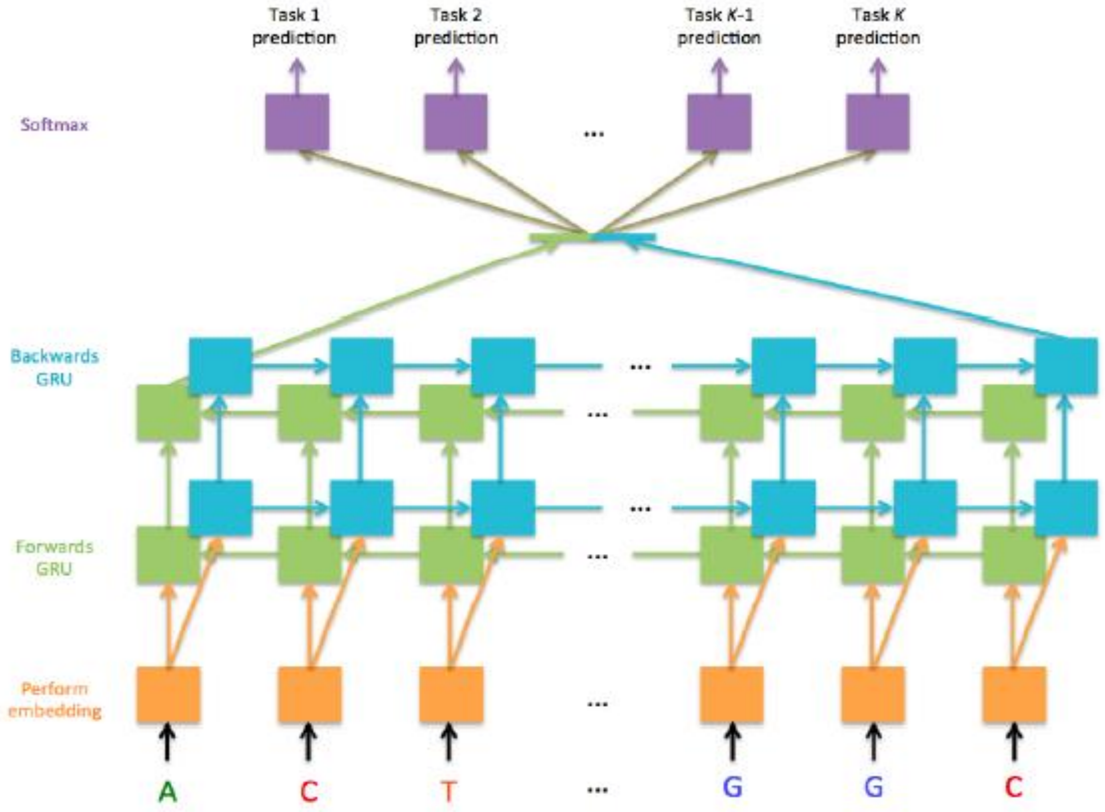

Figure 2: Architecture of proposed network [Jesse M. Zhang and Govinda M. Kamath Stanford University Learning the Language of the Genome using RNNs]

Table 1: Table showing the accuracy results[Jesse M. Zhang and Govinda M. Kamath Stanford Universiy Learning the Language of the Genome using RNNs]

	Phylum		Class		Order		Family		Genus	
	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
CNN	0,995	0,003	0,993	0,006	0,937	0,012	0,893	0,019	0,676	0,065
CNN-MT	0,981	0,007	0,978	0,008	0,908	0,021	0,851	0,024	0,692	0,024
LSTM	0,982	0,028	0,977	0,022	0,902	0,028	0,857	0,034	0,728	0,030
LSTM-MT	0,992	0,007	0,990	0,008	0,941	0,029	0,897	0,023	0,733	0,030

Experimental Design

Dataset: This research used 16S dataset. It was downloaded from the RDP Ribosomal Database Project II (RDP-II) by NCBI. By a subsequent filtering phase, a total amount of 3000 sequences have been selected, and can be grouped into 5 ordered taxonomic ranks, named *Phylum*, *Class*, *Order*, *Family* and *Genus*.

Evaluation Measures: Measures such as mean& standard deviation is computed on 10 validation test folds.

Software and Hardware Requirements: Python based Computer Vision and Deep Learning libraries will be exploited for the development and experimentation of the project. Training will be conducted on GPUs.