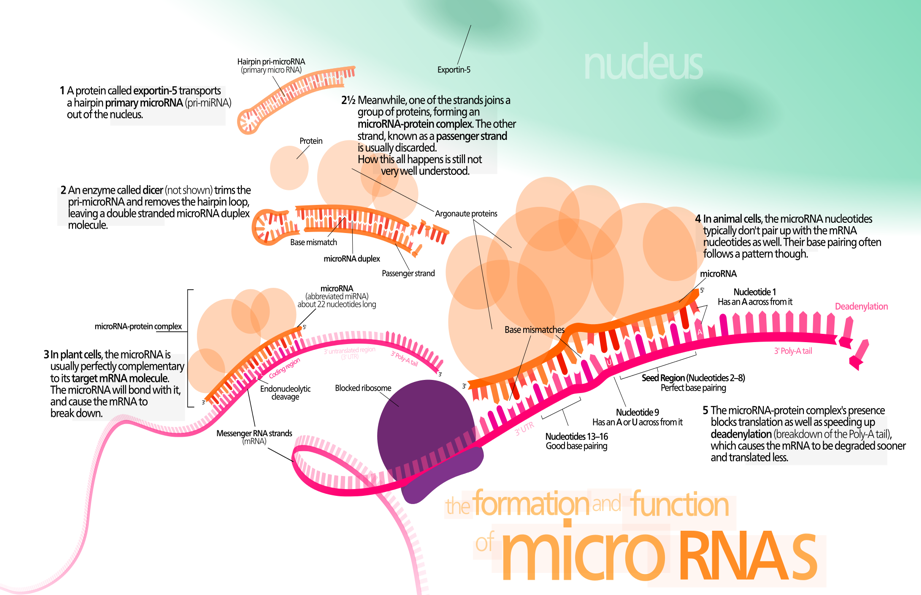
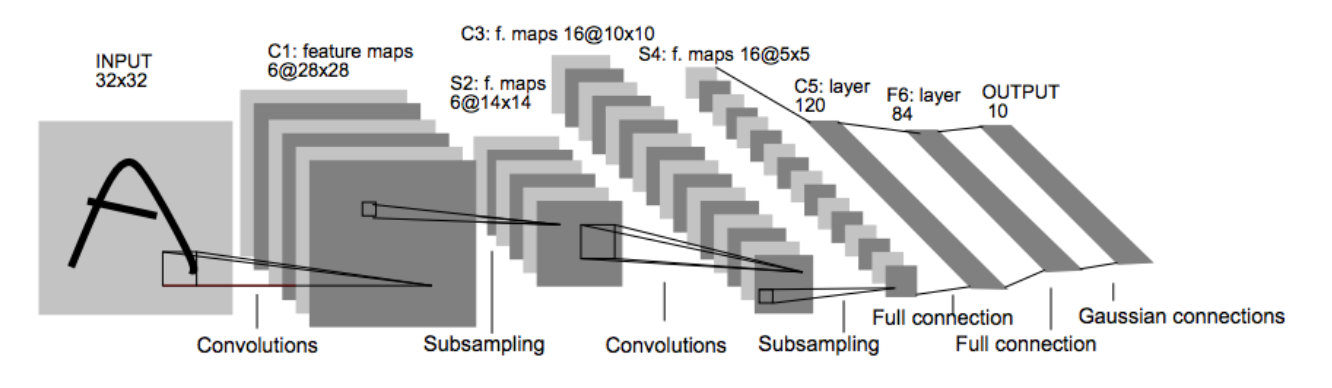
In recent years, deep learning methods, as a branch of machine learning, have become a powerful tool in many scientific fields. One of deep learning methods, convolutional neural network(CNN), which has an ability of extracting features of high-level abstraction from minimum preprocessing data, is the most widely used method in classification. In this project, we choose CNN to do microRNA classification and distinguish microRNA from non-microRNA. We use one-hot vector model to convert our input sequence into numeric matrices and use these matrices as input to CNN. At last we train as test our model using miRNA sequences from open-source database and the result shows that our model is efficient in doing miRNA classification.

MicroRNA (abbreviated miRNA) is a small non-coding RNA molecule (containing about 22 nucleotides) found in plants, animals and some viruses, that functions in RNA silencing and post-transcriptional regulation of gene expression. While the majority of miRNAs are located within the cell, some miRNAs, commonly known as circulating miRNAs or extracellular miRNAs, have also been found in the extracellular environment, including various biological fluids and cell culture media. So it's of great value to distinguish miRNAs from the whole genome and make accurate classifications. [1]



In machine learning, a convolutional neural network (CNN, or ConvNet) is a type of feed-forward artificial neural network in which the connectivity pattern between its neurons is inspired by the organization of the animal visual cortex. Individual cortical neurons respond to stimuli in a restricted region of space known as the receptive field. The receptive fields of different neurons partially overlap such that they tile the visual field. The response of an individual neuron to stimuli within its receptive field can be approximated mathematically by a convolution operation. Convolutional networks were inspired by biological processes and are variations of multilayer perceptrons designed to use minimal amounts of preprocessing. They have wide applications in image and video recognition, recommender systems and natural language processing.[2][3]

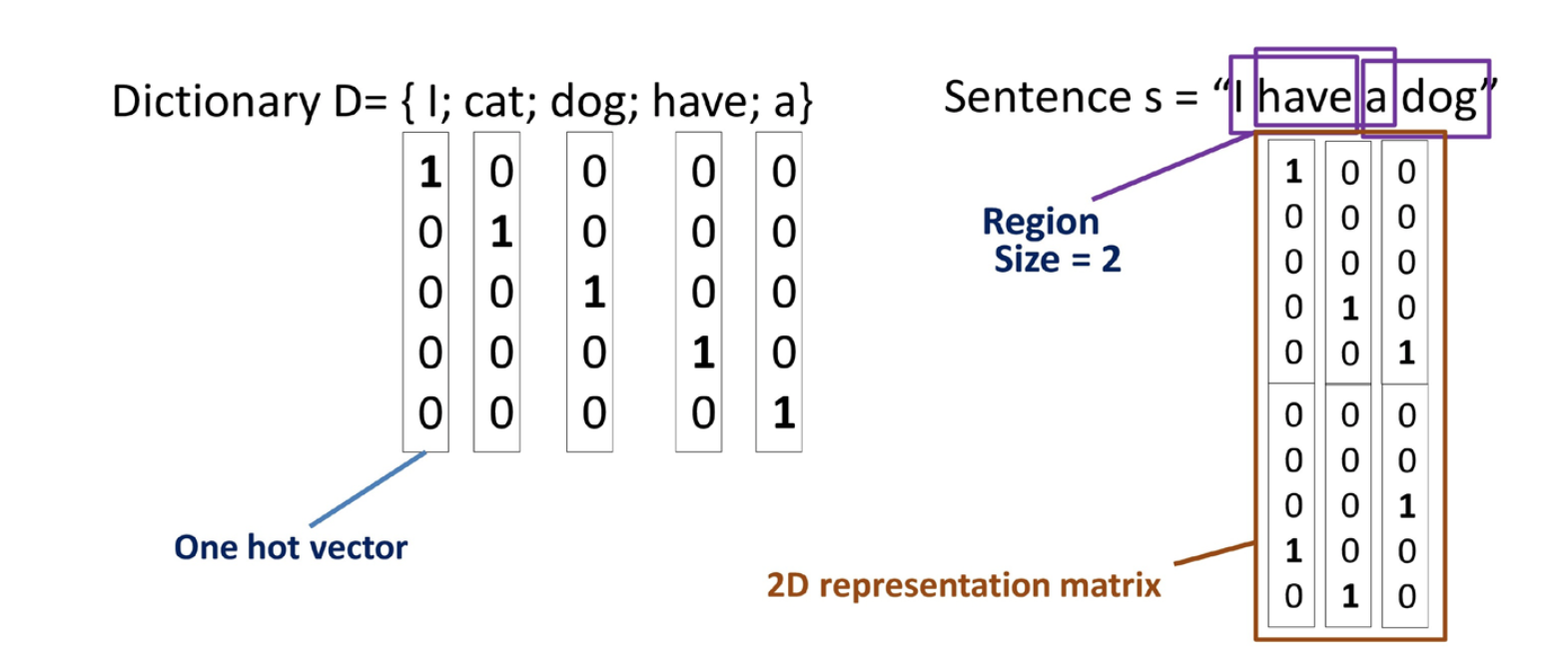


The data we use to train and test our model is from miRBase. [4] MiRBase is the central online repository of miRNA nomenclature, sequence data, annotation and target pre- diction, which first appeared in Oct. 2002 Release 15 contains 14197 miRNA loci from 66 species. From version 5.0, miRBase began to classify miRNAs into different families. [5] We use miRNA sequences from this data base as our positive input, meaning that this sequence belongs to the miRNA family.

Methods.

1. TO construct numeric matrices:

To convert data into vectors, we use a method called one-hot vector. In Fig3 below, we show an example of using one-hot vector to represent text into two-dimensional numerical matrix. Assume that we have a small dictionary D which contains 5 words: “I”, “cat”, “dog”, “have”, and “a”. Each word in the dictionary will be represented by a one-hot vector. Then to represent the sentence “I have a dog”, for each region of successive words (in this case is 2 words) we concatenate one-hot vector of each word in the region to generate a vector representing this region. After representing all regions by this mechanism, we will have a 2D numerical matrix representing the sentence. [6]



In our experiment, to represent each sequence, we use 3-mers, each can be A, T,C,G, to construct the whole dictionary. So in total our dictionary has 64 different words. And we use the same method with in this picture to convert our sequence into numerical matrix. The reason why we choose 3-mers is that every 3 nucleotides can be a codon for an amino acid. And the reason why we choose concatenate 2 words into a row vector is that we think a word can be influenced by other words, and the one with greatest influence is the one next to it.

1. Deep-learning Network:

The structure of our CNN is one input layer, one convolution layer, one inner product layer, one ReLu layer, one loss layer. We split data into 10-fold and take 9 fold to train and the rest to test. We use 10\*10 size of convolution matrix to both consider influence from nearby and avoid noise from distant regions.

1. https://en.wikipedia.org/wiki/MicroRNA
2. Lee H, Grosse R, Ranganath R, et al. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations[C]Proceedings of the 26th annual international conference on machine learning. ACM, 2009: 609-616.
3. <https://en.wikipedia.org/wiki/Convolutional_neural_network>
4. http://www.mirbase.org/
5. Ding J, Zhou S, Guan J. miRFam: an effective automatic miRNA classification method based on n-grams and a multiclass SVM[J]. BMC bioinformatics, 2011, 12(1): 216.
6. Nguyen N G, Tran V A, Ngo D L, et al. DNA Sequence Classification by Convolutional Neural Network[J]. Journal of Biomedical Science and Engineering, 2016, 9(05): 280.