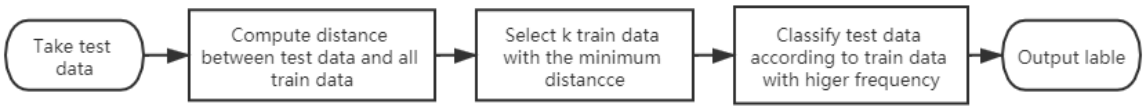
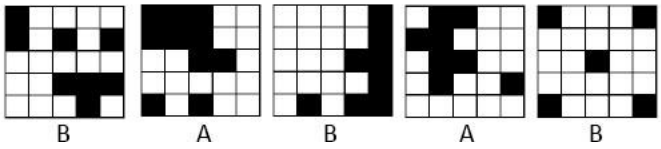
Classification

a) The first model that I choose is KNN, which is an algorithm widely use in classification tasks. The structure of my model is shown as following:



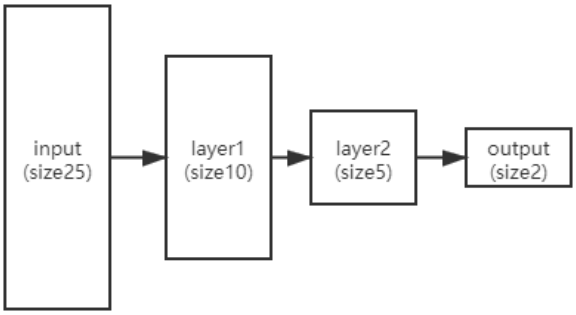
Every time we take a test data, we compare the distance between it and all the train data, after that we pick k train data that are most "similar" to the test data (I use Manhattan distance in this case). The label of the test data is then decided by the label appear in the k train data with higher frequency.

The following is the result given by KNN, which meet my expectation:



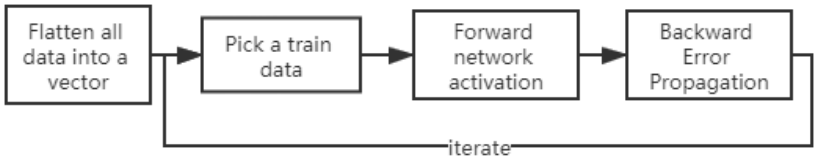
b) In order to avoid overfitting, there are strategy that might be helpful:
   1) As we can see, the image of class A and B have obvious different, so if we can generate new data ourselves by swapping one cell with its neighbor.
   2) Also, we can add some random noise in the origin data and get new train data. However, as the images is quite simple, the noise should be as simple as possible too.
   3) Choose model carefully, as some complex model such as CNN is known to have overfitting problems in small and simple datasets.

c) For my second model, I build a neural network:



Firstly, in order to fit the two-dimensional data into neural network, I flatten the matrix into a vector. After that the train data is pass into two hidden layer and an output layer and we get a vector of size two, representing the probability of data being class A and class B.

The data is trained in the following way:



The prediction of neural network is same as the one made by KNN.