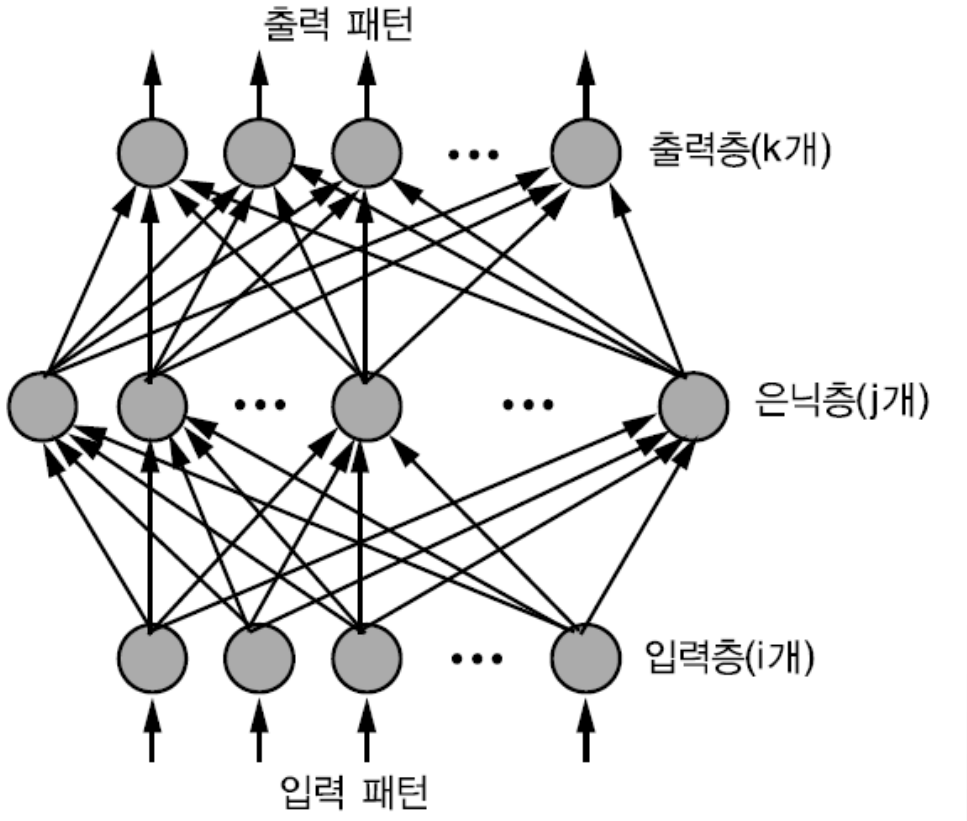
OpenCV

ANN

ANN

- ANN



ANN

- CvANN_MLP
  - create(cv::Mat layerSize, int activateFunc=CvANN_MLP::SIGMOID_SYM);
  - train(cv::Mat input, cv::Mat output, const cv::Mat& sampleWeights);
  - predic(cv::Mat input, cv::Mat out);
- CvANN_MLP_TrainParams

ANN

- CvTermCriteria
  - 반복적으로 사용되는 알고리즘 함수들은 CvTermCriteria 구조체 변수를 넘겨줌으로서 종료 조건을 설정할 수 있게 설계되어 있다.
  - Int type
    - 종료 타입
  - Int max_iter
    - 반복 횟수
  - Double epsilon
    - 정확도 오차

ANN

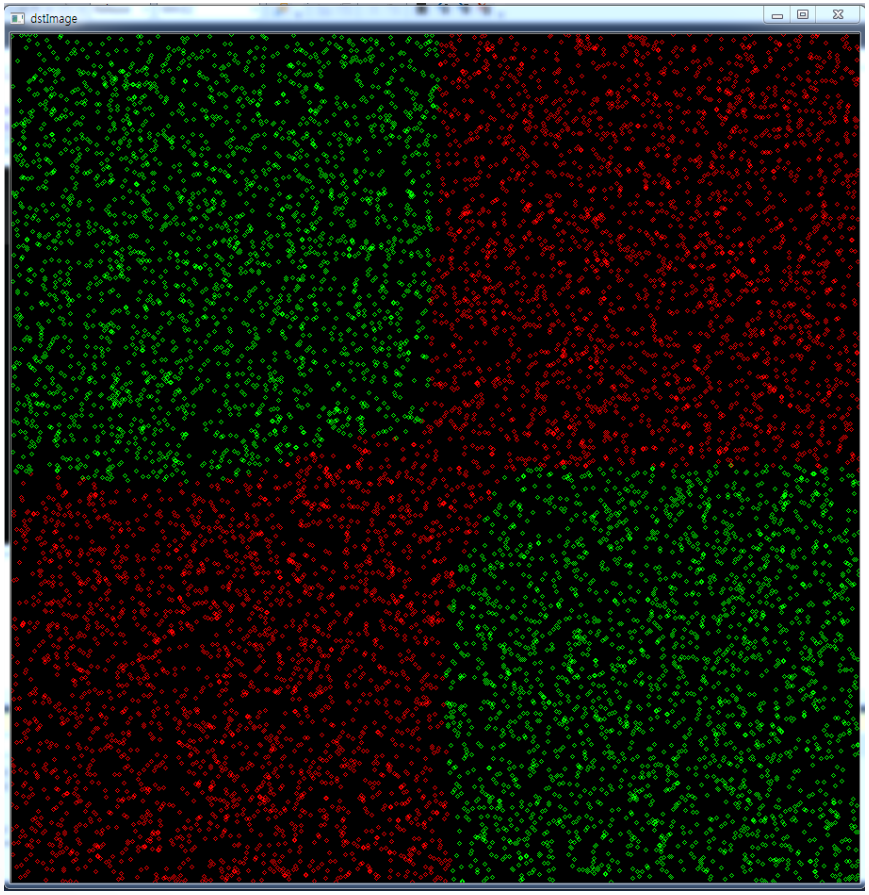
```
#include "opencv2₩opencv.hpp"

#define TRAININGDATA 500
#define TESTDATA 10000
#define IMAGESIZE 900

void main() {

    cv::Mat trainData = cv::Mat(TRAININGDATA, 2, CV_32FC1);
    cv::Mat testData = cv::Mat(TESTDATA, 2, CV_32FC1);

    cv::randu(trainData, 0, 1);
    cv::randu(testData, 0, 1);

    cv::Mat trainClasses = cv::Mat(trainData.rows, 1, CV_32FC1);
    cv::Mat testClasses = cv::Mat(testData.rows, 1, CV_32FC1);

    labelData(trainData, trainClasses);

    ANN(trainData, trainClasses, testData, testClasses);

}
```

ANN

```
void labelData(cv::Mat& data, cv::Mat classes) {

    for (int i = 0; i < data.rows; i++) {

        float x = data.at<float>(i, 0);
        float y = data.at<float>(i, 1);

        if ((x < 0.5 && y < 0.5) || (x > 0.5  && y > 0.5))
            classes.at<float>(i, 0) = -1;
        else
            classes.at<float>(i, 0) = 1;
    }

}
```

```
void ANN(cv::Mat& trainData, cv::Mat& trainClasses, cv::Mat& testData, cv::Mat& testClasses) {

    cv::Mat layer = cv::Mat(3, 1, CV_32SC1);

    layer.row(0) = cv::Scalar(2);
    layer.row(1) = cv::Scalar(10);
    layer.row(2) = cv::Scalar(1);

    CvANN_MLP ann;
    CvANN_MLP_TrainParams param;
    CvTermCriteria criteria;
    criteria.max_iter = 500;
    criteria.epsilon = 0.00001f;
    criteria.type = CV_TERMCRIT_ITER | CV_TERMCRIT_EPS;

    param.train_method = CvANN_MLP_TrainParams::BACKPROP;
    param.bp_dw_scale = 0.1f;
    param.bp_moment_scale = 0.1f;
    param.term_crit = criteria;
    ann.create(layer);

    ann.train(trainData, trainClasses, cv::Mat(), cv::Mat(), param);

    cv::Mat dstImage = cv::Mat::zeros(cv::Size(IMAGESIZE, IMAGESIZE), CV_8UC3);

    cv::Mat output = cv::Mat(1, 1, CV_32FC1);

    for (int i = 0; i < testData.rows; i++) {

        cv::Mat temp = testData.row(i);

        ann.predict(temp, output);

        int x = temp.at<float>(0, 0) * IMAGESIZE;
        int y = temp.at<float>(0, 1) * IMAGESIZE;

        if (output.at<float>(0, 0) < 0)
            cv::circle(dstImage, cv::Point(x, y), 2, cv::Scalar(0, 255, 0));
        else
            cv::circle(dstImage, cv::Point(x, y), 2, cv::Scalar(0, 0, 255));

    }

    cv::imshow("dstImage", dstImage);
    cv::waitKey(0);

}
```

ANN



ANN

- Exercise
  - 입력 층 2, 은닉 층 10, 출력 층 2의 신경망을 만들자!!
    (테스트 데이터를 활용하여 오류 율을 출력할 것!!)