Assignment10

Zhicun Chen 518030910173

1. Machine learning method work well because of human-designed representation and input features. And ML just optimizes weights to best make a final prediction.

   Deep learning is a learning subfield of learning representations of data, which is effective at learning patterns. DL algorithms learn representation by using a hierarchy or multiple layers. It doesn't need people to design a method to extract features.

   DL is useful because of the flow properties:
   (a) Manually designed features are often over-specified, incomplete and take a long time to design and validate.
   (b) Learned Features are easy to adapt, fast to learn.
   (c) Deep learning provides an almost very flexible, universal, learnable framework for representing world, visual and linguistic information.
   (d) Can learn both unsupervised and supervised.
   (e) Effective end-to-end joint system learning.
   (f) Utilize large amounts of training data。

2.

   (a) $y=\sigma(w_1x_1+w_2x_2+w_3x_3+b)$, in which σ is sigmoid function, therefore,

$$y = \frac{1}{1 + e^{-(w_1x_1+w_2x_2+w_3x_3+\mathrm{b})}}$$

   (b)$x_{11}=\sigma(2*(-1)+1*1+2)= \sigma(1)=\frac{1}{1+e^{-1}}=0.73$

   $x_{12}=\sigma(2*1+(-1)*(-1)-4)= \sigma(-1)=\frac{1}{1+e^{1}}=0.27$

   $y=\sigma(1*0.73+1*0.27+1)= \sigma(2)= \frac{1}{1+e^{-2}}=0.88$

3.

   (a)During training:

   $x_{11}=\mathrm{ReLU}((-1)*1+0*2+0*1)= \sigma(-1)=0$
   $x_{12}=\mathrm{ReLU}(2*1+1*2+0*1)=\mathrm{ReLU}(4)=4$
   $x_{14}=\mathrm{ReLU}(0*1+0*2=1*1)=\mathrm{ReLU}(1)=1$

   $y_1=\mathrm{ReLU}((-1)*0+2*4+(-4)*1)=\mathrm{ReLU}(4)=4$
   $y_2=\mathrm{ReLU}(1*0+0*4+(-2)*1)=0$

(b) During testing:

x_{11}=ReLU(0.75*((-1)*1+2.5*2+0*2+0*1))=ReLU(3)=3

x_{12}=ReLU(0.75*(2*1+0*2+1*2+0*1))=ReLU(3)=3

x_{13}=ReLU(0.75*(3*1+(-1)*2+0*2+(-2)*1))=ReLU(-0.75)=0

x_{14}=ReLU(0.75*(0*1+0*2+0*2+1*1))=ReLU(0.75)=0.75

y_1=ReLU(0.75*((-1)*3+2*3+0*0+(-4)*0.75))=ReLU(0)=0

y_2=ReLU(0.75*(1*3+0*3+(-1)*0+(-2)*0.75))=ReLU(1.125)=1.125

4.

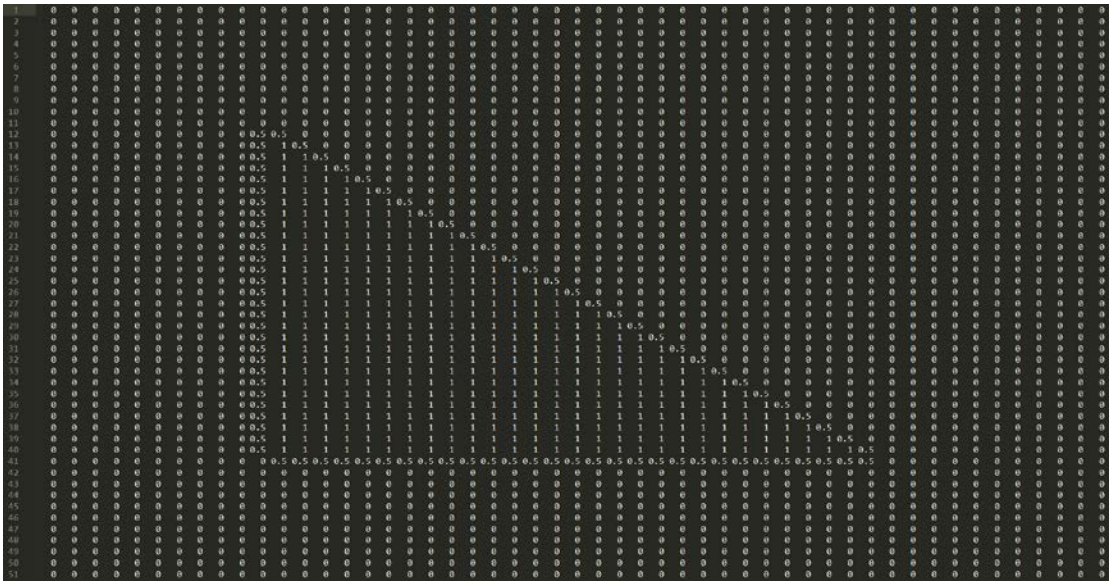
(a) I calculate the values by a simple program, which is as following:

```cpp
#include <iostream>
#include <cmath>
#include <fstream>
using namespace std;

const double a[]={10.,0.,-10.};
const double b[]={0.,10.,-10.};
const double e=2.7182818;
const double c[]={40.,40.,40.};

double sigmoid(double input)
{
    double output=1./(1+pow(e,-input));
    if(output<0.001) return 0;
    if(output>0.999) return 1;
    return output;
}


int main() {
    fstream result;
    result.open("result.txt");
    if(!result.is_open())
    {
        cerr<<"something wrong";
        exit(0);
    }
    for(int i=0;i<3;i++)
    {
        double x1,x2;
        cin>>x1>>x2;
        double y1,y2,y3;
        y1=a[0]*x1+b[0]*x2;
        y2=a[1]*x1+b[1]*x2;
        y3=a[2]*x1+b[2]*x2+300;
        y1=sigmoid(y1);
        y2=sigmoid(y2);
        y3=sigmoid(y3);
        double output=sigmoid(c[0]*y1+c[1]*y2+c[2]*y3-100);
        result<<x1<<' '<<x2<<':'<<output<<endl;
    }
    result.close();
    return 0;
}
```

And the results is:

output(A)=1; output(B)=0; output(C)=0.

(b) Just as (a) below, I use a simple program to calculate the values of output that input is [-10,40]X[-10,40]. The result is as following:



So the decision boundary can be displayed as:

$$\begin{cases} x_1 = 0 \\ x_2 = 0 \\ x_1 + x_2 = 30 \end{cases}$$