**Project 4**

**劉波洲 0540153**

把每個圖片（28 x 28）作爲輸入，其中像素值換到0到1之間（即像素值除以255），這樣計算相對簡單。共有10個輸出nodes，分別表示0~9的數字。

**training numbers: 6000**

**test numbers:1000**

**number of hidden layers:2**

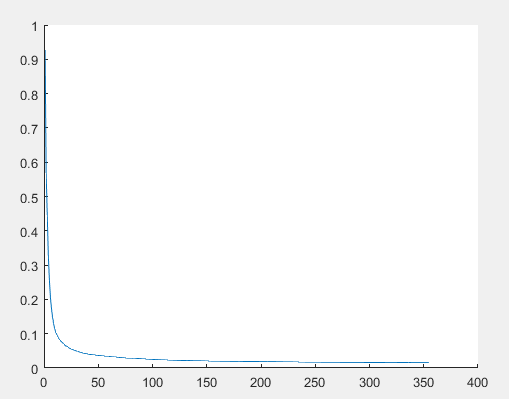
**number of hidden nodes:400 X 50**

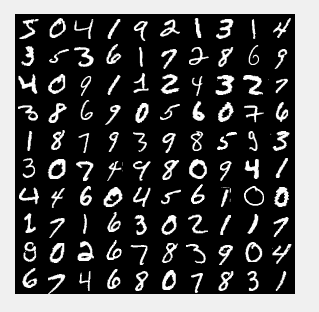
**learning rate parameter (η):0.25**

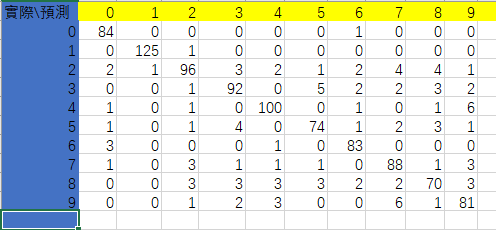
**stop criterion:0.01**

**result：Recognition rate：87.2% CPU time：about 24h**

**最後iteration355次，平均誤差為0.0155。**







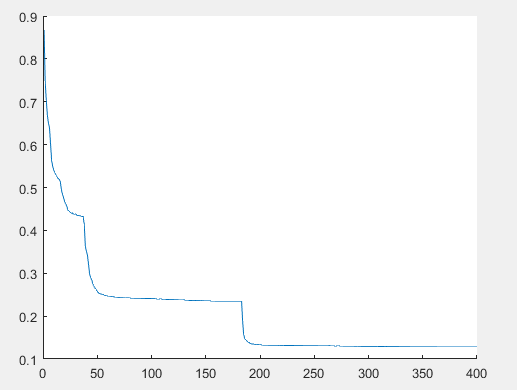
**flowchart：**



Discussion：

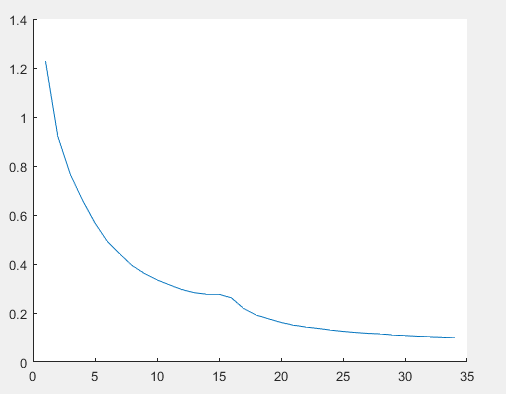
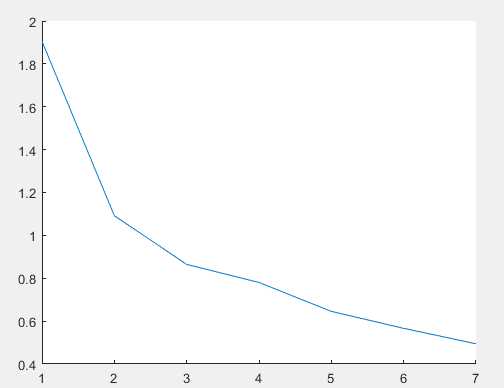
以下是一個hidden layer的實驗情況

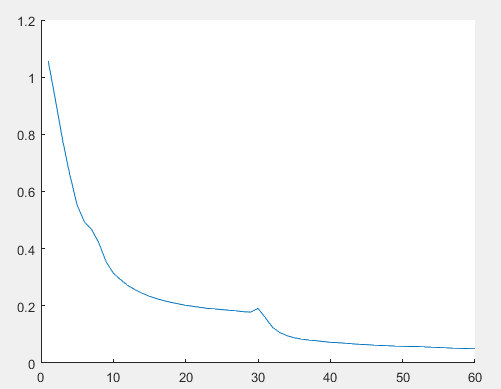
用600個hidden nodes, limit為0.05，從圖中看出200次之後錯誤率基本不變在0.12左右，説明遇到了local minimum。最後的識別成功率為76.6%。pattern為6000個。



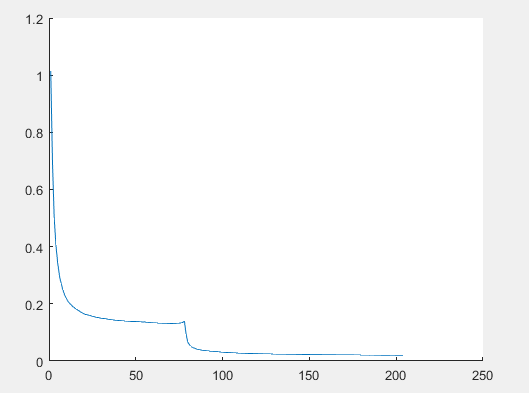
兩個hidden layer的實驗情況：

在400 x 50,learning rate為0.2的情況下，隨著limit值得變小，這是1000個patterns的情況下的成功率66%，73%，77%。





patterns調整為6000個后，成功率達到85%。Limit 值為0.05.



從實驗情況來看兩層 hidden layer效果比一層 hidden layer的效果要好，識別成功率高。

對於node個數，用35x25的情況下識別成功率低於20%，換成400x50后成功率達到85%以上，在最後分類比較多的情況下，node個數也需要比較大。對於其它的個數，沒有GPU加速，每次train時間比較長，沒有更多的實驗。

在開始的時候，直接把pixel的值作爲輸入，training效果較差，後來把pixel的值除以255轉換成0~1之間后效果有了明顯提升。

Code：

clear

%read training image

trainNum=6000;

testNum=1000;

imagefile=fopen('train-images.idx3-ubyte','r','b');

header=fread(imagefile,1,'int32');

if header ~= 2051

error('image file error');

end

numsofimage=fread(imagefile,1,'int32');

rows=fread(imagefile,1,'int32');

cols=fread(imagefile,1,'int32');

if trainNum>numsofimage

error('too much train samples')

end

imgs=zeros([rows cols trainNum]);

for t=1:trainNum

for r=1:rows

imgs(r,:,t)=fread(imagefile,cols,'uint8');

end

end

imgs=double(imgs)/255;

fclose(imagefile);

%read training label

labelfile=fopen('train-labels.idx1-ubyte','r','b');

header=fread(labelfile,1,'int32');

if header ~= 2049

error('image file error');

end

numsoflabel=fread(labelfile,1,'int32');

if trainNum>numsoflabel

error('too much train samples')

end

labels=fread(labelfile,trainNum,'uint8');

fclose(labelfile);

%training

nb=cols\*rows;

ni=400;

nj=50;

nk=10;

wkj=randn(nk,nj+1);

wkj\_tmp=wkj;

wji=randn(nj,ni+1);

wji\_tmp=wji;

wib=randn(ni,nb+1);

ob=zeros(nb+1,1);

si=zeros(ni,1);

oi=zeros(ni+1,1);

sj=zeros(nj,1);

oj=zeros(nj+1,1);

sk=zeros(nk,1);

ok=sk;

dk=sk;

lowerlimit=0.01;

itermax=400;

eta=0.25;

iter=0;

error\_avg=10;

deltak=zeros(1,nk);

deltaj=zeros(1,nj);

deltai=zeros(1,ni);

sumbackkj =zeros(1,nj);

sumbackji =zeros(1,ni);

while (error\_avg >lowerlimit ) && (iter<itermax)

errorn=0;

iter=iter+1;

for n=1:trainNum

%forward computation

for r=1:rows

for c=1:cols

ob((r-1)\*cols+c)=imgs(r,c,n);

end

end

ob(nb+1)=1;

dk=zeros(nk,1);

dk(labels(n)+1)=1;

for i=1:ni

si(i)=wib(i,:)\*ob;

oi(i)=1/(1+exp(-si(i)));

end

oi(ni+1)=1;

for j=1:nj

sj(j)=wji(j,:)\*oi;

oj(j)=1/(1+exp(-sj(j)));

end

oj(nj+1)=1;

for k=1:nk

sk(k)=wkj(k,:)\*oj;

ok(k)=1/(1+exp(-sk(k)));

end

errorn=errorn+sum(abs(dk-ok));

%error=error+(dk-ok)'\*(dk-ok)/2;

%backward learning

for k=1:nk

deltak(k)=(dk(k)-ok(k))\*ok(k)\*(1-ok(k));

end

for j=1:nj+1

for k=1:nk

wkj\_tmp(k,j)=wkj(k,j)+eta\*deltak(k)\*oj(j);

end

end

for j=1:nj

sumbackkj(j)=0.0;

for k=1:nk

sumbackkj(j)=sumbackkj(j)+deltak(k)\*wkj(k,j);

end

deltaj(j)=sumbackkj(j)\*oj(j)\*(1-oj(j));

end

for i=1:ni+1

for j=1:nj

wji\_tmp(j,i)=wji(j,i)+eta\*deltaj(j)\*oi(i);

end

end

for i=1:ni

sumbackji(i)=0.0;

for j=1:nj

sumbackji(i)=sumbackji(i)+deltaj(j)\*wji(j,i);

end

deltai(i)=sumbackji(i)\*oi(i)\*(1-oi(i));

end

for b=1:nb+1

for i=1:ni

wib(i,b)= wib(i,b)+eta\*deltai(i)\*ob(b);

end

end

wkj=wkj\_tmp;

wji=wji\_tmp;

end

ite(iter)=iter;

error\_avg=errorn/trainNum;

error\_r(iter)=error\_avg;

end

figure(2)

hold on

plot(ite,error\_r);

hold off

%read test data

imagefile=fopen('t10k-images.idx3-ubyte','r','b');

header=fread(imagefile,1,'int32');

if header ~= 2051

error('image file error');

end

numsofimage=fread(imagefile,1,'int32');

rows=fread(imagefile,1,'int32');

cols=fread(imagefile,1,'int32');

if testNum>numsofimage

error('too much train samples');

end

testimgs=zeros([rows cols testNum]);

for t=1:testNum

for r=1:rows

testimgs(r,:,t)=fread(imagefile,cols,'uint8');

end

end

for y=1:10

for x=1:10

temp((x-1)\*rows+1:x\*rows,(y-1)\*cols+1:y\*cols)=imgs(:,:,(x-1)\*10+y);

end

end

figure(3)

imshow(temp);

testimgs=double(testimgs)/255;

fclose(imagefile);

%read test label

labelfile=fopen('t10k-labels.idx1-ubyte','r','b');

header=fread(labelfile,1,'int32');

if header ~= 2049

error('image file error');

end

numsoflabel=fread(labelfile,1,'int32');

if testNum>numsoflabel

error('too much train samples')

end

testlabels=fread(labelfile,testNum,'uint8');

fclose(labelfile);

rightNums=0;

results=zeros([testNum,1]);

cm=zeros([10 10]);

for n=1:testNum

%forward computation

for r=1:rows

for c=1:cols

ob((r-1)\*cols+c)=testimgs(r,c,n);

end

end

ob(nb+1)=1;

for i=1:ni

si(i)=wib(i,:)\*ob;

oi(i)=1/(1+exp(-si(i)));

end

oi(ni+1)=1;

for j=1:nj

sj(j)=wji(j,:)\*oi;

oj(j)=1/(1+exp(-sj(j)));

end

oj(nj+1)=1;

for k=1:nk

sk(k)=wkj(k,:)\*oj;

ok(k)=1/(1+exp(-sk(k)));

end

if ok(testlabels(n)+1) > 0.5

rightNums=rightNums+1;

end

label=0;

min=-100;

for index=1:10

if(ok(index)>min)

min=ok(index);

label=index-1;

end

end

results(n)=label;

end

rate=rightNums/testNum;

for n=1:testNum

cm(testlabels(n)+1,results(n)+1)=cm(testlabels(n)+1,results(n)+1)+1;

end