التعرف على الأرقام العربية-الهندية باستخدام الشبكات العصبونية الالتفافية

مشروع مقدم لمادة التحكم الضبابي العصبوني للدكتور حسن البستاني

عدنان سعود

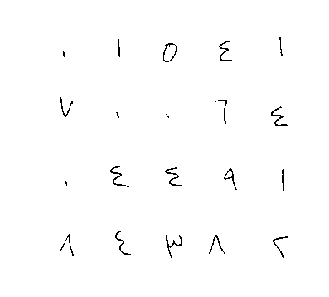
# ملخص

تعد الشبكات العصبونية الالتفافية أحد اهم إنجازات مخرجات تعلم الآلة الى عالم رؤية الحاسوب. يقدم هذا المستند مخرجات تطوير شبكة عصبونية التفافية للتعرف علي الأرقام الهندية المكتوبة بخط اليد. عشرون نموذجاً لكل رقم تم تجميعها وتركيبها في مجموعة تدريبية كدخل لشبكة التفافية نمطية. النتائج كانت مرضية بشكل عام نظراً لشح الدخل: دقة التدريب كانت 100% والتحقق 95%. أذكر هنا أن نسبة التحقق للتدريب كانت 1:1.

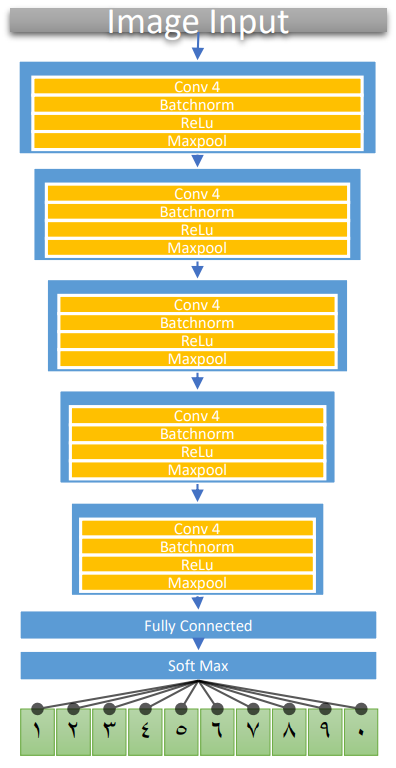
# الأدوات

تم تصميم وتدريب شبكة عصبونية عميقة مكونة من 14 طبقة مع 13000 متحول قابل للتدريب. دخل الشبكة عبارة عن صورة بوليانية مربعة بضلع 42 بكسل مع خرج شعاع مكون من 10 خانات تركز كل خانة فيه الى الرقم المطلوب من الشبكة. وأيضا بتايع خسارة crossentropyex

مثال عن دخل الشبكة موجود في الشكل 1، وبنية الشبكة موجودة في الشكل 2.



الشكل 1



الشكل 2

# التدريب

تم تدريب الشبكة باستخدام ADAM Stochastic Optimizer وذلك بسبب تقاربه السريع وحاجته المنخفضة للذاكرة العشوائية. معدل التدريب كان 0.005

الشكل 3 يبين مخطط التدريب ومنحني الخسارة.

منحني الدقة والخسارة

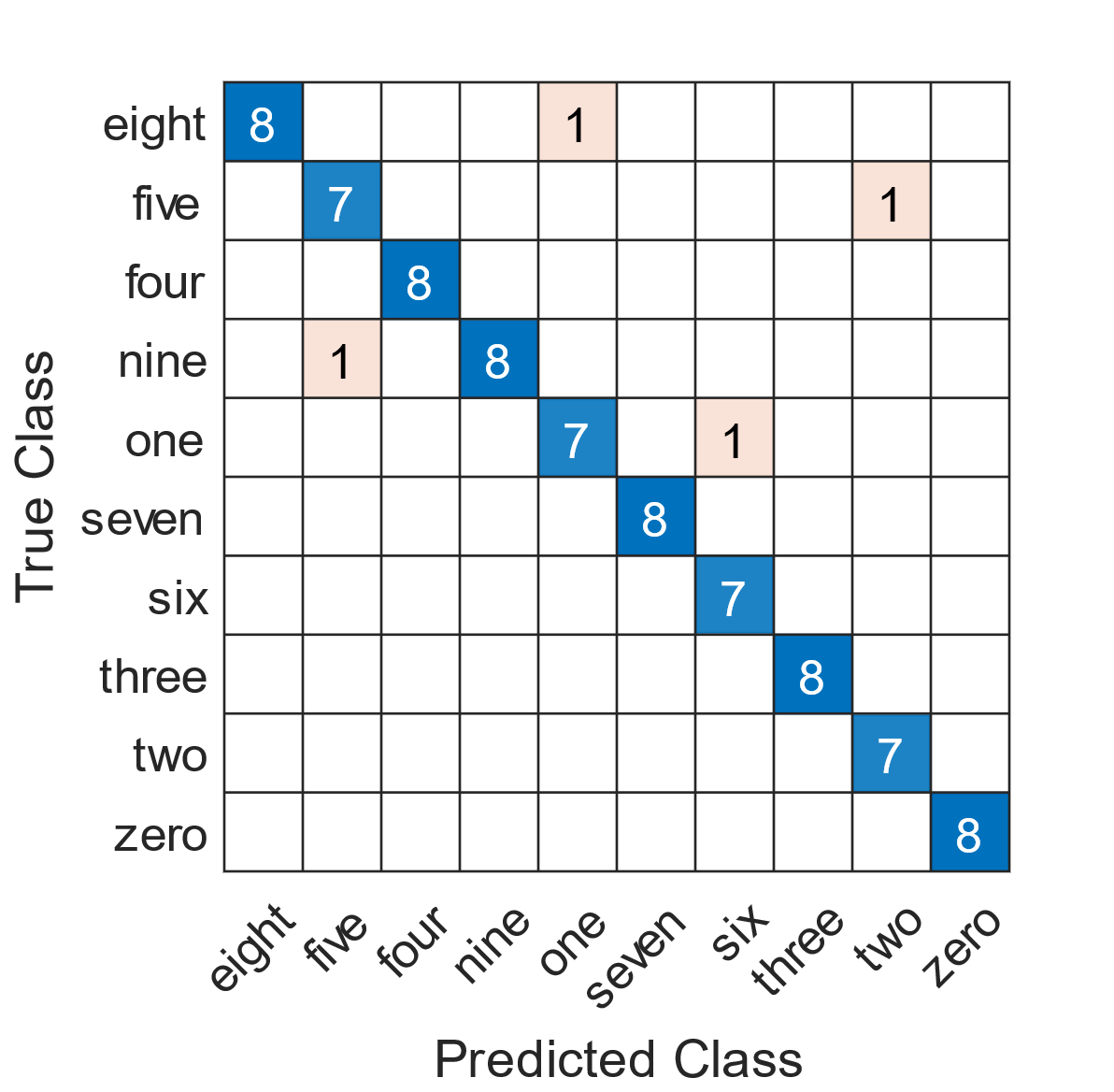
الدقة

الخسارة

الشكل 3

# النتائج

يبين الشكل رقم 4 منحني مصفوفة الـconfusion الخاصة بنتائج التدريب.



الشكل 4

# حدود المشروع

يعاني المشروع من المشاكل التالية:

* عدد عينات التدريب قليل نسبياً (الشبكات في الأبحاث الأخرى مدربة علة عشرات الآلاف)
* العينات مأخوذة من شخص واحد فقط (كاتب المشروع)، هذا يجعل اختلاف العينات عن بعضها قليل.
* أبعاد الصورة ليست من مضاعفات الـ2، أي وقت تدريب أكبر.

# تحليل النتائج

تضهر النتائج الموجودة في القسم السابق قدرة الشبكة الأكيدة على تمييز رموز الأرقام الهندية عن بعضها بيسر وسهولة.

# الخاتمة

قدم هذا المشروع طريقة مطروقة مسبقاً للتعرف على الأرقام العربية-الهندية على الحاسوب باستخدام الشبكات العصبونية الالتفافية.

# الملحق 1: عينات التدريب

# الملحق 2: برنامج تحصيل عينات التدريب

raw2 = imread('arabicocr.bmp');

raw = im2bw(raw2,0.5);

imshow(raw);

%%

offset\_x = 4+5;

offset\_y = 9+5;

w = 51 - 4/10;

l = 52.3 - 2/10;

W = 42;

L = 42;

%%

hold on;

for i = 0:9

for j = 0:19

y = round(i\*w + offset\_x: (i\*w + W + offset\_x-1));

x = round(j\*l + offset\_y: (j\*l + L + offset\_y-1));

temp(:,:,(i)\*20 + (j+1)) = raw(x ,y);

scatter(y , x , 'k.');

end

end

%%

for o = 1:size(temp,3)

imwrite(temp(:,:,o),[ 'train\tr' num2str(o) '.bmp']);

end

# الملحق 3: برنامج تدريب الشبكة

folder = 'train\_folder\';

imds = imageDatastore(folder, ...

'IncludeSubfolders',true,'LabelSource','foldernames');

rng(123);

figure;

perm = randperm(200,20);

for i = 1:20

subplot(4,5,i);

imshow(imds.Files{perm(i)});

end

labelCount = countEachLabel(imds)

img = readimage(imds,1);

size(img)

p = 0.6;

[imdsTrain,imdsValidation] = splitEachLabel(imds,p,'randomize');

%%

layers = [

imageInputLayer([42 42 1])

convolution2dLayer(3,4,'Padding','same')

batchNormalizationLayer

reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(3,8,'Padding','same')

batchNormalizationLayer

reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(3,16,'Padding','same')

batchNormalizationLayer

reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(3,32,'Padding','same')

batchNormalizationLayer

reluLayer

maxPooling2dLayer(2,'Stride',2)

convolution2dLayer(3,64,'Padding','same')

batchNormalizationLayer

reluLayer

fullyConnectedLayer(10)

softmaxLayer

classificationLayer];

%%

options = trainingOptions('adam', ...

'InitialLearnRate',0.005, ...

'MaxEpochs',60, ...

'Shuffle','every-epoch', ...

'ValidationData',imdsValidation, ...

'ValidationFrequency',4, ...

'Verbose',false, ...

'Plots','training-progress');

%%

[net,data] = trainNetwork(imdsTrain,layers,options);

%%

figure;

YPred = classify(net,imdsValidation);

YValidation = imdsValidation.Labels;

confusionchart(YPred,YValidation);