Computer Vision

Classification with Neural Networks part II

script commands and examples

27 August 2008

Copyright © 2001 – 2008 by
Noordelijke Hogeschool Leeuwarden and Van de Loosdrecht Machine Vision
All rights reserved

j.van.de.loosdrecht@tech.nhl.nl, jaap@vdlmv.nl

Classification with Neural Networks part II

Overview script commands and examples :

- **Script commands**
  - **Class Image Set (CIS)**
  - **Class Feature Set (CFS)**
  - **Back Propagation Network (BPN)**
- **Examples**
  - **Learning sinus function**
  - **Generate CIS**
  - **Create CIS and BPN, Train BPN with CIS, Classify image using BPN and Evaluate BPN using CIS**
  - **Create CFS and BPN, Train BPN with CFS, Classify image using BPN and Evaluate BPN using CFS**

Class Image Set commands

Introduction:

A CIS is a collection of images with their associated classes.
All images in a CIS must have the same image type and size.
The image type and size are defined when the CIS is created.
The info field gives a text description of the CIS. This info field is defined when the CIS is created and the text should be between quotes<"> and <"> if the text field contains spaces.

A CIS has a class table and for each class an image table.
Each image in an image table has its unique image index number.

28-Aug-08 Classification with Neural Networks part II 3

Class Image Set commands

Overview:

- **CIS_AddClass <cisName> <className>**
- **CIS_AddImage <cisName> <imageName> <className>**
- **CIS_ClassId <cisName> <className>**
- **CIS_ClassName <cisName> <classId>**
- **CIS_Create <cisName> <imageType> <info> <height> <width>**
- **CIS_Delete <cisName>**
- **CIS_GetClassTab <cisName>**
- **CIS_GetInfo <cisName>**
- **CIS_GetImage <cisName> <imageName> <className> <imageIndex>**
- **CIS_GetImageHeight <cisName>**
- **CIS_GetImageWidth <cisName>**

28-Aug-08 Classification with Neural Networks part II 4

Class Image Set commands

Overview:

- **CIS_GetMaxPixel <cisName>**
- **CIS_GetMinPixel <cisName>**
- **CIS_MaxClassId <cisName>**
- **CIS_NrOfImages <cisName> <className>**
- **CIS_ReadFromFile <cisName> <filename>**
- **CIS_RemoveClass <cisName> <className>**
- **CIS_RemoveImage <cisName> <className> <imageIndex>**
- **CIS_Rename <old cisName> <new cisName>**
- **CIS_WriteToFile <cisName> <filename>**

28-Aug-08 Classification with Neural Networks part II 5

Class Image Set commands

CIS_AddClass <cisName> <className>
Add class to cis.
Function result is string with classId.

CIS_AddImage <cisName> <imageName> <className>
Add image to cis.
Function result is string with imageId.

CIS_ClassId <cisName> <className>
Function result is string with classId.

CIS_ClassName <cisName> <classId>
Function result is string with class name.

CIS_Create <cisName> <imageType> <info> <height> <width>
Create an cis for specified image type with images of height x width.

28-Aug-08 Classification with Neural Networks part II 6

Class Image Set commands

CIS_Delete <cisName>
Delete specified cis.

CIS_GetClassTab <cisName>
Function result is string with class table.

CIS_GetInfo <cisName>
Function result is string with info field.

CIS_GetImage <cisName> <imageName> <className> <imageIndex>
Specified image in cis is copied to image with image name.

CIS_GetImageHeight <cisName>
Function result is string with height of images in cis.

CIS_GetImageWidth <cisName>
Function result is string with width of images in cis.

28-Aug-08 Classification with Neural Networks part II 7

Class Image Set commands

CIS_GetMaxPixel <cisName>
Function result is string with max pixel value of all images in cis.

CIS_GetMinPixel <cisName>
Function result is string with min pixel value of all images in cis.

CIS_MaxClassId <cisName>
Function result is a string the largest classId in cis.

CIS_NrOfImages <cisName> <className>
Function result is a string the nr of images in cis.

CIS_ReadFromFile <cisName> <filename>
Read cis from file.

CIS_RemoveClass <cisName> <className>
Remove class from cis.

28-Aug-08 Classification with Neural Networks part II 8

Class Image Set commands

CIS_RemoveImage <cisName> <className> <imageIndex>
Remove specified image from cis.

CIS_Rename <old cisName> <new cisName>
Rename cis from old name to new name.

CIS_WriteToFile <cisName> <filename>
Write cis to file.

Class Feature Set commands

Introduction:

A CFS is a collection of images with their associated classes.

A CFS has a class table and for each class an image table.
Each image in an image table has its unique image index number.

Class Feature Set commands

Overview:
- **CFS_AddClass <cfsName> <className>**
- **CFS_AddFeature <cfsName> <featureName>**
- **CFS_AddImage <cfsName> <imageName> <className>**
- **CFS_CalcFeatures <cfsName> <internalScriptName>**
- **CFS_ClassId <cfsName> <className>**
- **CFS_ClassName <cfsName> <classId>**
- **CFS_Create <cfsName> <imageType> <info>**
- **CFS_Delete <cfsName>**
- **CFS_GetFeatureNameTab <cfsName>**
- **CFS_GetInfo <cfsName>**
- **CFS_GetImage <cfsName> <imageName> <className> <imageIndex>**
- **CFS_GetMinMaxTab <cfsName>**
- **CFS_GetNrFeatures <cfsName>**
- **CFS_GetNrSelectedFeatures <cfsName>**
- **CFS_GetFeaturesImage <cfsName> <className> <imageIndex>**
- **CFS_GetScriptName <cfsName>**

28-Aug-08 Classification with Neural Networks part II 11

Class Feature Set commands

Overview:
- **CFS_GetSelectedFeaturesImage <cfsName> <className> <imageIndex>**
- **CFS_GetSelectTab <cfsName>**
- **CFS_GetClassTab <cfsName>**
- **CFS_MaxClassId <cfsName>**
- **CFS_NrOfImages <cfsName> <className>**
- **CFS_ReadFromFile <cfsName> <filename>**
- **CFS_RemoveClass <cfsName> <className>**
- **CFS_RemoveFeature <cfsName> <featureName>**
- **CFS_RemoveImage <cfsName> <className> <imageIndex>**
- **CFS_Rename <old cfsName> <new cfsName>**
- **CFS_SelectFeature <cfsName> <featureName> <0|1>**
- **CFS_SetMinMaxTab <cfsName> <minmaxinputtab>**
- **CFS_SetFeaturesImage <cfsName> <className> <imageIndex> <features>**
- **CFS_SetScriptName <cfsName> <scriptname>**
- **CFS_SetSelectTab <cfsName> <selected>**
- **CFS_WriteToFile <cfsName> <filename>**

28-Aug-08 Classification with Neural Networks part II 12

Class Feature Set commands

CFS_AddClass <cfsName> <className>
Add class to cfs.
Function result is string with classId.

CFS_AddFeature <cfsName> <featureName>
Add feature to cfs.
Function result is string with featureId.

CFS_AddImage <cfsName> <imageName> <className>
Add image with imageName to cfs for class className.

CFS_CalcFeatures <cfsName> <internalScriptName>
Execute internal script with internalScriptName to calculate all features for all images in cfs.

CFS_ClassId <cfsName> <className>
Function result is string with classId for class with className in cfs.

28-Aug-08 Classification with Neural Networks part II 13

Class Feature Set commands

CFS_ClassName <cfsName> <classId>
Function result is string with class name for classId in cfs.

CFS_Create <cfsName> <imageType> <info>
Create a cfs for specified image type and add text in info parameter to info field of cfs.

CFS_Delete <cfsName>
Delete specified cfs.

CFS_GetFeatureNameTab <cfsName>
Function result is string with all feature names.

CFS_GetInfo <cfsName>
Function result is info field of cfs.

CFS_GetImage <cfsName> <imageName> <className> <imageIndex>
Specified image in cfs is copied to image with image name.

28-Aug-08 Classification with Neural Networks part II 14

Class Feature Set commands

CFS_GetMinMaxTab <cfsName>
Function result is string with MinMaxTable of cfs.

CFS_GetNrFeatures <cfsName>
Function result is string with nr of features in cfs.

CFS_GetNrSelectedFeatures <cfsName>
Function result is string with nr of selected features of cfs.

CFS_GetFeaturesImage <cfsName> <className> <imageIndex>
Function result is string with all features of selected image in cfs.

CFS_GetScriptName <cfsName>
Function result is string with internal script name of cfs used to calculated the features.

CFS_GetSelectedFeaturesImage <cfsName> <className> <imageIndex>
Function result is string with the selected features of selected image in cfs.

28-Aug-08 Classification with Neural Networks part II 15

Class Feature Set commands

CFS_GetSelectTab <cfsName>
Function result is a string with represents the selected feautures.
The string consists of '0' and '1' seperated by spaces.
'0' means feature not selected, '1' means feature is selected.
The '0' and '1' are in the order of increasing classId.

CFS_GetClassTab <cfsName>
Function result is a string with the class table.

CFS_MaxClassId <cfsName>
Fuction result is a string the largest classId in cfs.

CFS_NrOfImages <cfsName> <className>
Function result is a string with the number of images of className in cfs.

CFS_ReadFromFile <cfsName> <filename>
Read cfs from file.

28-Aug-08 Classification with Neural Networks part II 16

Class Feature Set commands

CFS_RemoveClass <cfsName> <className>
Remove class className from cfs, all images for className are deleted from cfs.

CFS_RemoveFeature <cfsName> <featureName>
Remove feature with featureName from cfs.

CFS_RemoveImage <cfsName> <className> <imageIndex>
Remove specified image from cfs.

CFS_Rename <old cfsName> <new cfsName>
Rename cfs from old name to new name.

CFS_SelectFeature <cfsName> <featureName> <0|1>
Include ('1') or remove ('0') specified feature to selected set of features of cfs.

28-Aug-08 Classification with Neural Networks part II 17

Class Feature Set commands

CFS_SetMinMaxTab <cfsName> <minmaxinputtab>
Set minMaxTab of cfs.

CFS_SetFeaturesImage <cfsName> <className> <imageIndex> <features>
Set features of specified image in cfs.

CFS_SetScriptName <cfsName> <scriptname>
Set name of internal script used to calculate the features for all images in cfs.

CFS_SetSelectTab <cfsName> <selected>
Set selected features of cfs.
The selected string consists of '0' and '1' seperated by spaces.
'0' means feature not selected, '1' means feature is selected.
The '0' and '1' are in the order of increasing classId.

28-Aug-08 Classification with Neural Networks part II 18

Class Feature Set commands

CFS_WriteToFile <cfsName> <filename>
Write cfs to file.

Back Propagation Network commands

Overview:
- **BPN_Classify <bpnName> <inputs>**
- **BPN_ClassifyFeatures <bpnName> <image> <iscript>**
- **BPN_ClassifyImage <bpnName> <image>**
- **BPN_Create <bpnName> <func> <bias> <$minMaxInputTab> <minOutput> <maxOutput> <$layerdescrtab>**
- **BPN_CreateClassifier <bpnName> <hiddens1> <hiddens2> <nrClasses> <bias> <$minMaxInputTab>**
- **BPN_CreateFeatureClassifier <bpnName> <cfsName> <nrHiddens1> <nrHiddens2> <bias>**
- **BPN_CreateImageClassifier <bpnName> <imageType> <nrPixels> <hiddens1> <hiddens2> <nrClasses> <bias> <$minMaxInputTab>**
- **BPN_Delete "<bpnName>**
- **BPN_EvaluateClass <bpnName> <input> <classExp>**
- **BPN_EvaluateCIS <bpnName> <cisName> <minConfidence> <NoDetails | LowDetails | HighDetails>**
- **BPN_EvaluateImage <bpnName> <imageName> <classExp>**
- **BPN_EvaluateImageSet <bpnName> <minConfidence> <NoDetails | LowDetails | HighDetails> <$images> <$classes>**

Back Propagation Network commands

Overview:

- **BPN_EvaluateClassSet <bpnName> <minConfidence> <NoDetails | LowDetails | HighDetails> <inputs> <classes>**
- **BPN_EvaluateSet <bpnName> <$inputs> <$targets>**
- **BPN_GetLearnError <bpnName>**
- **BPN_GetMomentum <bpnName>**
- **BPN_GetNameLearnSet <bpnName>**
- **BPN_GetSizeLayer <bpnName> <layerNr>**
- **BPN_Produce <bpnName> <inputs>**
- **BPN_ReadFromFile <bpnName> <filename>**
- **BPN_Rename <old bpnName> <new bpnName>**
- **BPN_Reset <bpnName> <minRand> <maxRandW>**
- **BPN_SetNameLearnSet <bpnName> <name learn set>**
- **BPN_Train <bpnName> <learnRate> <momentum> <$input> <$target>**
- **BPN_TrainSet <bpnName> <nrOfEpochs> <learnRate> <momentum> <$inputs> <$targets>**
- **BPN_TrainClass <bpnName> <learnRate> <momentum> <$input> <class>**

Back Propagation Network commands

Overview:

- **BPN_TrainClassSet <bpnName> <nrOfEpochs> <learnRate> <momentum> <setsize> <$input class> <$classes>**
- **BPN_TrainImage <bpnName> <learnRate> <momentum> <image> <class>**
- **BPN_TrainImageSet <bpnName> <nrOfEpochs> <learnRate> <momentum> <setsize> <$images> <$classes>**
- **BPN_TrainCIS <bpnName> <cisName> <nrOfEpochs> <learnRate> <momentum>**
- **BPN_WriteToFile <bpnName> <filename>**

Back Propagation Network commands

BPN_Classify <bpnName> <inputs>
Classify with specified inputs, function result is string with classId and confidence seperated with a space.
bpnName: name of bpn variable.
inputs: string with all inputs separated by space.

BPN_ClassifyFeatures <bpnName> <image> <iscript>
Classify image with feature classifier.
Features are calculated with internal script, %p1 is image.
Function result is string with classId and confidence seperated with a space.
bpnName: name of bpn variable.
image: name of image to classify
iscript: name of internal script.

Back Propagation Network commands

BPN_ClassifyImage <bpnName> <image>
Classify image with image classifier.
Function result is string with classId and confidence seperated with a space.
bpnName: name of bpn variable.
image: name of image to classify

BPN_Create <bpnName> <func> <bias> <$minMaxInputTab> <minOutput> <maxOutput> <$layerdescrtab>
Create a bpn.
bpnName: name of bpn variable.
func: activation function, values: Hyperbolic, Linear or Sigmoid.
bias: use of BiasNodes, values: Bias NoBias
minMaxInputTab: string with min and max value for all inputs, separeted by spaces
minOutPut: minimum value for output neuron
maxOutput: maximum value for output neuron
layerdescrtab: string with: nrLayers followed by nr of neurons for each layer

Back Propagation Network commands

BPN_CreateClassifier <bpnName> <hiddens1> <hiddens2> <nrClasses> <bias> <$minMaxInputTab>
Create a classifier.
bpnName: name of bpn variable.
hiddens1: size of hidden layer 1
hiddens2: size of hidden layer 2
nrClasses: nr of classes
bias: use of BiasNodes, values: Bias NoBias
minMaxInputTab: string with min and max value for all inputs, separeted by spaces

BPN_CreateFeatureClassifier <bpnName> <cfsName> <nrHiddens1> <nrHiddens2> <bias>
Create a feature classifier.
bpnName: name of bpn variable.
cfsName: name of cfs variable.
hiddens1: size of hidden layer 1
hiddens2: size of hidden layer 2
bias: use of BiasNodes, values: Bias NoBias

Back Propagation Network commands

BPN_CreateImageClassifier <bpnName> <imageType> <nrPixels> <hiddens1> <hiddens2> <nrClasses> <bias> <$minMaxInputTab>
Create an image classifier.
bpnName: name of bpn variable.
imagetype: type of image, like ByteImage or RGB888Image.
nrPixels: nr of pixels in image
hiddens1: size of hidden layer 1
hiddens2: size of hidden layer 2
nrClasses: nr of classes.
bias: use of BiasNodes, values: Bias NoBias
minMaxInputTab: string with min and max value for all inputs, separeted by spaces

BPN_Delete "<bpnName>
Delete a BPN.
bpnName: name of bpn variable.

Back Propagation Network commands

BPN_Evaluate <bpnName> <input> <target>
Evaluate BPN by producing outputs with input and comparing with target.
Function result is the mean error in the output layer.
bpnName: name of bpn variable.
input: string with all input values separated with spaces.
target: string with all expected target values separated with spaces.

BPN_EvaluateClass <bpnName> <input> <classExp>
Evaluate feature classifier.
Function result is string with classId and confidence.
bpnName: name of bpn variable.
input: string with all input values separated with spaces.
classExp: expected classId.

28-Aug-08 Classification with Neural Networks part II 27

Back Propagation Network commands

BPN_EvaluateCIS <bpnName> <cisName> <minConfidence> <NoDetails | LowDetails | HighDetails>
Evaluate a CIS.
Function result is a string with:
maxError, meanError, nr of missclassifications and nr of low confidences
If not NoDetails is specified the result string is expanded for all miss classications by:
expected classId, result, classId, confidence, imageIndex and if high details all values for all output neurons.
If not NoDetails is specified the result string is expanded for all low confidences by:
expected classId, result, classId, confidence, imageIndex and if high details all values for all output neurons.
bpnName: name of bpn variable.
cisName: name of cis variable.
minConfidence: the minimum value for the confidence wanted.
details: NoDetails, LowDetails or HighDetails.

28-Aug-08 Classification with Neural Networks part II 28

Back Propagation Network commands

BPN_EvaluateImage <bpnName> <imageName> <classExp>
Evaluate an Image.
Function result is string with classId and confidence.
bpnName: name of bpn variable.
input: string with all input values separated with spaces.
classExp: expected classId.

Back Propagation Network commands

BPN_EvaluateImageSet <bpnName> <minConfidence> <NoDetails | LowDetails | HighDetails> <$images> <$classes>
Evaluate an image set.
Function result is a string with:
maxError, meanError, nr of missclassifications and nr of low confidences
If not NoDetails is specified the result string is expanded for all miss classications by:
expected classId, result, classId, confidence, imageIndex and if high details all values for all output neurons.
If not NoDetails is specified the result string is expanded for all low confidences by:
expected classId, result, classId, confidence, imageIndex and if high details all values for all output neurons.
bpnName: name of bpn variable.
minConfidence: the minimum value for the confidence wanted.
details: NoDetails, LowDetails or HighDetails.
images: string with all image names.
classes: string with all classIds.

Back Propagation Network commands

BPN_EvaluateClassSet <bpnName> <minConfidence> <NoDetails | LowDetails | HighDetails> <inputs> <classes>
Evaluate a class set.
Function result is a string with:
maxError, meanError, nr of missclassifications and nr of low confidences
If not NoDetails is specified the result string is expanded for all miss classications by:
expected classId, result, classId, confidence, imageIndex and if high details all values for all output neurons.
If not NoDetails is specified the result string is expanded for all low confidences by:
expected classId, result, classId, confidence, imageIndex and if high details all values for all output neurons.
bpnName: name of bpn variable.
minConfidence: the minimum value for the confidence wanted.
details: NoDetails, LowDetails or HighDetails.
inputs: string with setsize followed by all inputs.
classes: string with setsize followed by all classIds.

28-Aug-08 Classification with Neural Networks part II 31

Back Propagation Network commands

BPN_EvaluateSet <bpnName> <$inputs> <$targets>
Evaluate a set of inputs.
Function result is string with max error and mean error.
bpnName: name of bpn variable.
inputs: string with setsize followed by all inputs.
targets: string with setsize followed by all targets.

BPN_GetLearnError <bpnName>
Function result is learn error of specified bpn.

BPN_GetLearnRate <bpnName>
Function result is learn rate of specified bpn.

BPN_GetMomentum <bpnName>
Function result is momentum of specified bpn.

BPN_GetNameLearnSet <bpnName>
Function result is name of learnset of specified bpn.

28-Aug-08 Classification with Neural Networks part II 32

Back Propagation Network commands

BPN_GetSizeLayer <bpnName> <layerNr>
Function result is size of layer with layernr of specified bpn.

BPN_Produce <bpnName> <inputs>
Function result is the values of the output layers.
bpnName: name of bpn variable.
inputs: string with the input values.

BPN_ReadFromFile <bpnName> <filename>
Read bpn from file.
bpnName: name of bpn variable.
filename: file to read bpn from.

BPN_Rename <old bpnName> <new bpnName>
Rename bpn with old bpnName to new bpnName.

BPN_Reset <bpnName> <minRand> <maxRandW>
Reset all weights of bpn to random values.

Back Propagation Network commands

BPN_SetNameLearnSet <bpnName> <name learn set>
Set name of learn set of bpn to specified value.

BPN_Train <bpnName> <learnRate> <momentum> <$input> <$target>
Train bpn with one example.
Function result is learn error.
bpnName: name of bpn variable.
learnRate: learn rate.
momentum: momentum.
input: string with input values
target: string with target values

Back Propagation Network commands

BPN_TrainSet <bpnName> <nrOfEpochs> <learnRate> <momentum> <$inputs> <$targets>
Train bpn with a set of examples.
Function result is learn error.
bpnName: name of bpn variable.
nrOfEpochs: number of times training process is repeated.
learnRate: learn rate.
momentum: momentum.
inputs: string with input values.
targets: string with target values.

BPN_TrainClass <bpnName> <learnRate> <momentum> <$input> <class>
Train a classifier with one examples.
Function result is learn error.
bpnName: name of bpn variable.
learnRate: learn rate.
momentum: momentum.
input: string with input values.
class: string with targetId.

Back Propagation Network commands

BPN_TrainClassSet <bpnName> <nrOfEpochs> <learnRate> <momentum> <setsize> <$input class> <$classes>
Train classifier with a set of examples.
Function result is the max error and mean error.
bpnName: name of bpn variable.
nrOfEpochs: number of times training process is repeated.
learnRate: learn rate.
momentum: momentum.
setsize: nr of examples.
inputs: string with input values.
classes: string with classIds.

Back Propagation Network commands

BPN_TrainImage <bpnName> <learnRate> <momentum> <image> <class>
Train a classifier with one image.
Function result is learn error.
bpnName: name of bpn variable.
learnRate: learn rate.
momentum: momentum.
image: string with name of image.
class: string with targetId.

Back Propagation Network commands

BPN_TrainImageSet <bpnName> <nrOfEpochs> <learnRate> <momentum> <setsize> <$images> <$classes>
Train a classifier with set of images.
Function result is the max error and mean error.
bpnName: name of bpn variable.
nrOfEpochs: number of times training process is repeated.
learnRate: learn rate.
momentum: momentum.
setsize: nr of examples.
images: string with name of images.
classes: string with targetIds.

Back Propagation Network commands

BPN_TrainCIS <bpnName> <cisName> <nrOfEpochs> <learnRate> <momentum>
Train a classifier with set of images.
Function result is the max error and mean error.
bpnName: name of bpn variable.
cisName: name of the cis variable.
nrOfEpochs: number of times training process is repeated.
learnRate: learn rate.
momentum: momentum.

BPN_WriteToFile <bpnName> <filename>
Write bpn to file.
bpnName: name of bpn variable.
filename: name of file to write bpn.

Examples

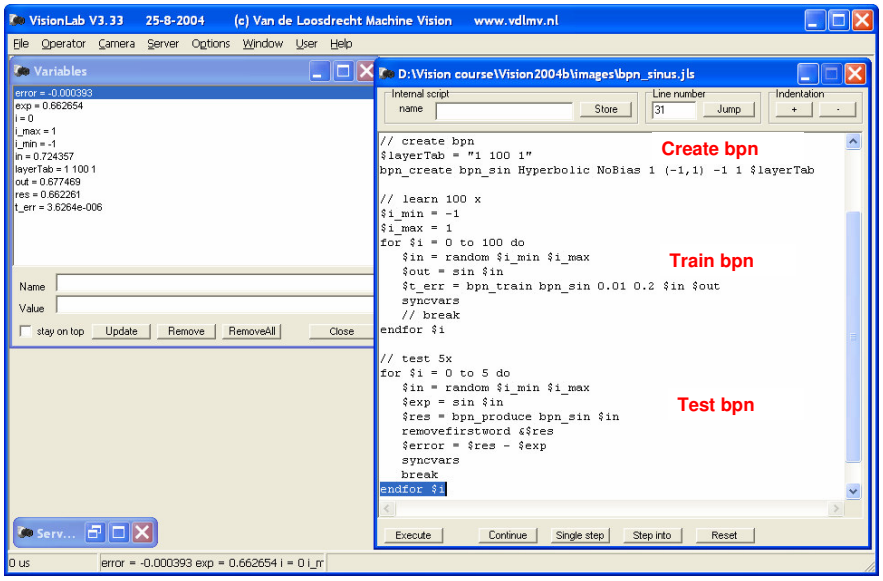
- **Learning sinus function**
- **Generate Class Image Set**
- **Create CIS and BPN, Train BPN with CIS, Classify image using BPN and Evaluate BPN using CIS**
- **Create CFS and BPN, Train BPN with CFS, Classify image using BPN and Evaluate BPN using CFS**

Example: learning sinus function

- Create BPN with
  - Input layer of 1 neuron
  - One hidden layer of 100 neurons
  - Output layer of 1 neuron
- Train the BPN with 100 random examples: x, sin(x)
  - $in = random –1 1
  - $out = sin $in
  - Note the behaviour of the training error ($t_err)
- Test the BPN with 5 random values
  - $in = random –1 1
  - $res = sinus of $in learned by BPN
  - $exp = sin $in
  - $error = error in result

- use script bpn_sinus.jls

Example: learning sinus function



Example: generate class image set

- Create cis
- Threshold image
- Remove big blobs
- Label image
- BlobAnalyse SortDown TopLeft UseX Height TopLeft Width
- for $group = 0 to 1 do
  - for $num = 0 to 9 do
    - for $i = 0 to 9 do
      - Roi $t $l $h $w
      - Contraststretch
      - Zoom NearestPixelInterpolation
      - Add normalized image to cis

script: gen_ocr1_cis.jls (1)

```
$path = lpwd
$oldpath = pwd
cwd $path

$imagename = ocr1.jl
$cisname = ocr1.cis
$h_norm = 23
$w_norm = 17
$size = $h_norm * $w_norm
cis_create cis ByteImage "big number test" $h_norm $w_norm
cis_addclass cis zero
cis_addclass cis one
cis_addclass cis two
cis_addclass cis three
cis_addclass cis four
cis_addclass cis five
cis_addclass cis six
cis_addclass cis seven
cis_addclass cis eight
cis_addclass cis nine
```

Save old file path server

File path server = file path client

Create CIS and add 10 classes

script: gen_ocr1_cis.jls (2)

```
lread org $imagename
display org
copy org bin
thresholdisodata bin DarkObject
removeblobs bin EightConnected Area 300 10000 UseX
labelblobs bin EightConnected
$ana = blobanalysis bin SortDown TopLeft UseX Height TopLeft Width
for $group = 0 to 1 do
  for $num = 0 to 9 do
    $classname = cis_classname cis
    for $i = 0 to 9 do
      removefirstword &$ana
      $h = removefirstword &$ana
      $tl = removefirstword &$ana
      $t = getnthfromvector 1 $tl
      $l = getnthfromvector 2 $tl
      $w = removefirstword &$ana
```

Remove 2 big horizontal lines

Find the image position of all digits

Note: blobs are sorted on x coordinate of top left

So there are first 10 zero's followed by 10 ones etc

There are two groups of 10 * 10 digits

script: gen_ocr1_cis.jls (3)

```
      $t = $t - 1
      $l = $l - 1
      $h = $h + 2
      $w = $w + 2
      roi org roi $t $l $h $w
      convert roi contrast ByteImage
      contraststretch contrast 0 255
      zoom contrast new $h_norm $w_norm NearestPixelInterpolation
      cis_addimage cis new $classname
      display new
      syncvars
    endfor $i
  endfor $num
endfor $group
cis_writetofile cis $cisname

cwd $oldpath
```

Cut out digits with a margin of 1 pixel

Convert to ByteImage

Normalize contrast and size

Add image with digit to CIS

Write CIS to file

Set file path server to original path

Example script bpn_cis_digits.jls

Steps:

- **Create CIS**
- **Create BPN**
- **Train BPN with CIS**
- **Classify image using BPN**
- **Evaluate BPN using CIS**

Notes

- **This example creates only a tiny CIS with only one image for each class. See example gen_ocr1_cis.jls for generating a bigger more realistic CIS.**
- **Set $debug to true to enable "debugging" script and examine the $variables with Examine Variables in Server menu**

28-Aug-08 Classification with Neural Networks part II 47

Script bpn_cis_digits.jls (1)

```
initrandomgen 0
$debug = true                                   Save old file path server
$path = lpwd                                    File path server = file path client
$oldpath = pwd
cwd $path

$cisName = small.cis                            Create CIS and add classes
$h_norm = 24
$w_norm = 16
$size = $h_norm * $w_norm
$base = number
cis_create $cisName ByteImage info $h_norm $w_norm
for $learn = 0 to 9 do
  $name = $base . $learn
  $classid = cis_addclass $cisName $name
endfor $learn

$classtab = cis_getclasstab $cisName            Get class table from cis
```

28-Aug-08 Classification with Neural Networks part II 48

Script bpn_cis_digits.jls (2)

```
lread org numbers.jl
copy org thres
threshold thres 0 250
removeblobs thres EightConnected Area 300 10000 UseX
labelblobs thres EightConnected
$ana = blobanalysis thres SortDown TopLeft UseX Height TopLeft Width
$base = number
```

Find the image position of the digits

Note: blobs are sorted on x coordinate of top left

Script bpn_cis_digits.jls (3)

```
for $label = 0 to 9 do
  removefirstword &$ana
  $h = removefirstword &$ana
  $tl = removefirstword &$ana
  $t = getnthfromvector 1 $tl
  $l = getnthfromvector 2 $tl
  $w = removefirstword &$ana
  roi org tmp $t $l $h $w
  contraststretch tmp 0 255
  $class = $label
  $name = $base . $class
  zoom tmp $name $h_norm $w_norm NearestPixelInterpolation
  if $debug then
    display $name
  endif
  cis_addimage $cisName $name $name
endfor $label
```

Cut out digits with a margin of 1 pixel

Convert to ByteImage

Normalize contrast and size

Add image with digit to CIS

Script bpn_cis_digits.jls (4)

```
cis_writetofile $cisName $cisName
cis_delete $cisName

cis_readfromfile cis_small small.cis

$h = cis_getimageheight cis_small
$w = cis_getimagewidth cis_small
$size = $h * $w

$hidden = 20
bpn_createimageclassifier bpn_small ByteImage $size $hidden 0 10 NoBias 0 255
$base = number
$nrOfEpochs = 10
for $epoch = 1 to 10 do
  $a_err = bpn_traincis bpn_small cis_small $nrOfEpochs 0.001 0.0001
  if $debug then
    syncvars
  endif
endfor $epoch
```

Save CIS to file

Delete CIS

Read CIS from file

Retrieve height/width info fromCIS

Create BPN

Train BPN using CIS

28-Aug-08 Classification with Neural Networks part II 51

Script bpn_cis_digits.jls (5)

```
for $t = 0 to 9 do
  $name = $base . $a_class_test
  $a_out = bpn_classifyimage bpn_small $name
  $a_class_res = getfirstword $a_out
  $a_class_id = cis_classname cis_small $a_class_res
  $a_class_confidence = getnthword 2 $a_out
  if $debug then
    syncvars
    break
  endif
endfor $t

$res = bpn_evaluatecis bpn_small cis_small 1.0 HighDetails
```

Classify image using BPN, result is classId and confidence

Use CIS to convert classId to className

Evaluate BPN using CIS
Note: in this example evaluation CIS and train CIS are the same

28-Aug-08 Classification with Neural Networks part II 52

Script bpn_cis_digits.jls (6)

```
bpn_writetofile bpn_small small.bpn
bpn_delete bpn_small
cis_delete cis_small

cwd $oldpath
```

Write BPN to file

Delete BPN and CIS from memory

Set file path server to original path

Example script test_feat_cfs.jls

Steps:

- **Create CFS**
- **Create BPN**
- **Train BPN with CFS**
- **Classify image using BPN**
- **Evaluate BPN using CFS**

Notes

- **Set $debug to true to enable "debugging" script and examine the $variables with Examine Variables in Server menu**

script test_feat_cfs.jls (1)

```
$display = true
$path = lpwd
$oldpath = pwd
cwd $path

$h_norm = 23
$w_norm = 17

cfs_create cfs Int16Image info
cfs_addclass cfs zero
cfs_addclass cfs one
cfs_addclass cfs two
cfs_addclass cfs three
cfs_addclass cfs four
cfs_addclass cfs five
cfs_addclass cfs six
cfs_addclass cfs seven
cfs_addclass cfs eight
cfs_addclass cfs nine
$classtab = cfs_getclasstab cfs
```

Save old file path server

File path server = file path client

Create CFS and add classes

Get class table from CFS

script test_feat_cfs.jls (2)

```
cfs_addfeature cfs Area
cfs_addfeature cfs AreaHoles
cfs_addfeature cfs Breadth
cfs_addfeature cfs CentreOfGravity_x
cfs_addfeature cfs CentreOfGravity_y
cfs_addfeature cfs MomentsScale_xy
cfs_addfeature cfs NrOfHoles
cfs_addfeature cfs Perimeter
$fname_base = SumRow
for $h = 1 to $h_norm do
  $fname = $fname_base . $h
  cfs_addfeature cfs $fname
endfor $h
$fname_base = sumCol
for $c = 1 to $w_norm do
  $fname = $fname_base . $c
  cfs_addfeature cfs $fname
endfor $h
$featurenametab = cfs_getfeaturenametab cfs
```

Add features to CFS

Get feature table from CFS

script test_feat_cfs.jls (3)

```
lread org ocr1.jl
if $display then
  display org
endif
copy org bin
thresholdisodata bin DarkObject
removeblobs bin EightConnected Area 300 10000 UseX
fillspecificholes bin FourConnected Area 0 10 UseX
labelblobs bin EightConnected
$ana = blobanalysis bin SortDown TopLeft UseX Height TopLeft Width
for $group = 0 to 1 do
  for $num = 0 to 9 do
    for $i = 0 to 9 do
      removefirstword &$ana
      $h = removefirstword &$ana
      $tl = removefirstword &$ana
      $t = getnthfromvector 1 $tl
      $l = getnthfromvector 2 $tl
      $w = removefirstword &$ana
```

Find the image position of the digits

Note: blobs are sorted on x coordinate of top left

script test_feat_cfs.jls (4)

```
      $t = $t - 1
      $l = $l - 1
      $h = $h + 2
      $w = $w + 2
      roi org roi $t $l $h $w
      $className = cfs_classname cfs $num
      contraststretch roi 0 255
      zoom roi new $h_norm $w_norm NearestPixelInterpolation
      cfs_addimage cfs new $className
      if $display then
        display roi
        syncvars
      endif
    endfor $i
  endfor $num
endfor $group
break
```

Cut out digits with a margin of 1 pixel

Convert to ByteImage

Normalize contrast and size

Add image with digit to CFS

script test_feat_cfs.jls (5)

```
cfs_setscriptname cfs test_calc_feat.jls     Set script name of CFS
laddscript ocr_feat test_calc_feat.jls       Calculate all features for all image in CFS
cfs_calcfeatures cfs ocr_feat
cfs_writetofile cfs test.cfs                 Save and read CFS
cfs_readfromfile cfs test.cfs
break
$nr_epochs = 40
$learnrate = 0.005
$momentum = 0.01
$nr_hidden = 200                             Create BPN
bpn_createfeatureclassifier bpn_feature_test cfs $nr_hidden 0 Bias
BPN_SetNameLearnSet bpn_feature_test test.cfs
for $epoch = 1 to $nr_epochs do              Train BPN using CFS
  $a_err = bpn_traincfs bpn_feature_test cfs 1 $learnrate $momentum
  if $display then
    syncvars
  endif
endfor $epoch                                Save BPN
bpn_writetofile bpn_feature_test feature_test.bpn
break
```


script test_feat_cfs.jls (6)

```
$a_eval = bpn_evaluatecfs bpn_feature_test cfs 0.7 LowDetails
break                                        Evaluate BPN using CFS

$s = ""
for $f = 1 to 48 do                          Create string with selected features
  $s = $s . 1
endfor $f
lread roi eight.jl                           Classify image with selected features
$a_class = bpn_classifyselectedfeatures bpn_feature_test roi ocr_feat $s
break
$a_class = bpn_classifyfeatures bpn_feature_test roi ocr_feat
break                                        Classify image

cfs_delete cfs                               Delete CFS from memory
delete org
delete bin
delete roi

                                             Set file path server to original path
cwd $oldpath
```