


NHL  
STENDEN  
computer vision  
& data science



## Computer Vision

### Classification with Neural Networks part II (\*) script commands and examples

10 April 2018

Copyright © 2001 – 2018 by  
NHL Stenden Hogeschool and Van de Loosdrecht Machine Vision BV  
All rights reserved

j.van.de.loosdrecht@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**

27-Aug-18 Classification with Neural Networks part II 2

### 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.

27-Aug-18

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>

27-Aug-18

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>

27-Aug-18

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.

27-Aug-18

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.

27-Aug-18

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.

27-Aug-18

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.

27-Aug-18

Classification with Neural  
Networks part II

9

### **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.**

27-Aug-18

Classification with Neural  
Networks part II

10

### 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>

27-Aug-18

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>

27-Aug-18

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.

27-Aug-18

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.

27-Aug-18

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.

27-Aug-18

Classification with Neural  
Networks part II

15

### Class Feature Set commands

**CFS\_GetSelectTab <cfsName>**

Function result is a string with represents the selected features.

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.

27-Aug-18

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.

27-Aug-18

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.

27-Aug-18

Classification with Neural  
Networks part II

18

### Class Feature Set commands

**CFS\_WriteToFile** <cfsName> <filename>  
Write cfs to file.

27-Aug-18

Classification with Neural  
Networks part II

19

### Back Propagation Network commands

#### Overview:

- **BPN\_Classify** <bpnName> <inputs>
- **BPN\_ClassifyFeatures** <bpnName> <image> <iscript>
- **BPN\_ClassifySelectedFeatures** <bpnName> <image> <iscript> <selected>
- **BPN\_ClassifyFeaturesOutputTab** <bpnName> <image> <iscript> <&\$array>
- **BPN\_ClassifySelectedFeaturesOutputTab** <bpnName> <image> <iscript> <selected> <&\$array>
- **BPN\_ClassifyImage** <bpnName> <image>
- **BPN\_ClassifyImageOutputTab** <bpnName> <image> <&\$array>
- **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>

27-Aug-18

Classification with Neural  
Networks part II

20

### Back Propagation Network commands

#### Overview:

- BPN\_EvaluateCIS <bpnName> <cisName> <minConfidence> <NoDetails | LowDetails | HighDetails>
- BPN\_EvaluateImage <bpnName> <imageName> <classExp>
- BPN\_EvaluateImageSet <bpnName> <minConfidence> <NoDetails | LowDetails | HighDetails> <\$images> <\$classes>
- 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>

27-Aug-18

Classification with Neural  
Networks part II

21

### Back Propagation Network commands

#### Overview:

- 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>
- 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>

27-Aug-18

Classification with Neural  
Networks part II

22

### Back Propagation Network commands

**BPN\_Classify <bpnName> <inputs>**

Classify with specified inputs, function result is string with classId and confidence separated 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 separated with a space.

**bpnName:** name of bpn variable.

**image:** name of image to classify

**iscript:** name of internal script.

27-Aug-18

Classification with Neural  
Networks part II

23

### Back Propagation Network commands

**BPN\_ClassifySelectedFeatures <bpnName> <image> <iscript> <selected>**

Classify image with feature classifier.

Features are calculated with internal script, %p1 is image.

Function result is string with classId and confidence separated with a space.

**bpnName:** name of bpn variable.

**image:** name of image to classify

**iscript:** name of internal script

**selected:** selection of features with "01" string

27-Aug-18

Classification with Neural  
Networks part II

24

### Back Propagation Network commands

**BPN\_ClassifyFeaturesOutputTab** <bpnName> <image> <iscript>  
<&\$array>

**Classify image with feature classifier.**

**Features are calculated with internal script, %p1 is image.**

**Function result is the confidence.**

**bpnName:** name of bpn variable.

**image:** name of image to classify

**iscript:** name of internal script.

**&\$array:** an array with tuples (<classId>,<output>) sorted on highest output values

27-Aug-18

Classification with Neural  
Networks part II

25

### Back Propagation Network commands

**BPN\_ClassifySelectedFeaturesOutputTab** <bpnName> <image> <iscript>  
<selected> <&\$array>

**Classify image with feature classifier.**

**Features are calculated with internal script, %p1 is image.**

**Function result is the confidence.**

**bpnName:** name of bpn variable.

**image:** name of image to classify

**iscript:** name of internal script

**selected:** selection of features with "01" string

**&\$array:** an array with tuples (<classId>,<output>) sorted on highest output values

27-Aug-18

Classification with Neural  
Networks part II

26

### Back Propagation Network commands

**BPN\_ClassifyImage <bpnName> <image>**

Classify image with image classifier.

Function result is string with classId and confidence separated with a space.

**bpnName:** name of bpn variable.

**image:** name of image to classify

**BPN\_ClassifyImageOutputTab <bpnName> <image> <&\$array>**

Classify image with image classifier.

Function result is string with classId and confidence separated with a space.

**bpnName:** name of bpn variable.

**image:** name of image to classify

**&\$array:** an array with tuples (<classId>,<output>) sorted on highest output values

27-Aug-18

Classification with Neural  
Networks part II

27

### Back Propagation Network commands

**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, separated 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

27-Aug-18

Classification with Neural  
Networks part II

28

### 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, separated 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

27-Aug-18

Classification with Neural  
Networks part II

29

### 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, separated by spaces

27-Aug-18

Classification with Neural  
Networks part II

30

### Back Propagation Network commands

**BPN\_Delete "<bpnName>**

Delete a BPN.

**bpnName:** name of bpn variable.

**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.

27-Aug-18

Classification with Neural  
Networks part II

31

### 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 missclassifications 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.**

27-Aug-18

Classification with Neural  
Networks part II

32



### 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.

27-Aug-18

Classification with Neural  
Networks part II

33

### 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 missclassifications 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.

27-Aug-18

Classification with Neural  
Networks part II

34

### 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 missclassifications 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.

27-Aug-18

Classification with Neural  
Networks part II

35

### 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.

27-Aug-18

Classification with Neural  
Networks part II

36

### 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.

27-Aug-18

Classification with Neural  
Networks part II

37

### 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

27-Aug-18

Classification with Neural  
Networks part II

38

### 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.

27-Aug-18

Classification with Neural  
Networks part II

39

### 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.

27-Aug-18

Classification with Neural  
Networks part II

40

### 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.

27-Aug-18

Classification with Neural  
Networks part II

41

### 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.

27-Aug-18

Classification with Neural  
Networks part II

42

### 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.

27-Aug-18

Classification with Neural  
Networks part II

43

### Back Propagation Network commands

**BPN\_CreateImageOptimizer** <popSize> <nrEpochs> <lowConf> <trainCIS>  
<evalCIS> <hidden1Low> <hidden1High> <hidden2Low> <hidden2High>  
<learnRateLow> <learnRateHigh> <momentumLow> <momentumHigh>

Create a BPN Image Optimizer (genetic algorithm)

**popSize**: the population size for the GA algorithm

**nrEpochs**: the number of epochs the BPN is trained before being evaluated

**lowConf**: the low confidency factor used in the evaluation

**trainCIS**: the name of the CIS to be used for training

**evalCIS**: the name of the CIS to be used for evaluation

**hidden1Low**: lowest value for size first hidden layer

**hidden1High**: highest value for size first hidden layer

**hidden2Low**: lowest value for size second hidden layer

**hidden2High**: highest value for size second hidden layer

**learnRateLow**: lowest value for learn rate

**learnRateHigh**: highest value for learn rate

**momentumLow**: lowest value for momentum

**momentumHigh**: highest value for momentum

27-Aug-18

Classification with Neural  
Networks part II

44

### Back Propagation Network commands

**BPN\_ImageOptimize** <nrGenerations> <minError> <deltaError> <microP>  
 Optimize for nrGenerations generations  
 Function result: <error> <hidden1> <hidden2> <learnRate> <momentum>  
 MinError is the minimal target error  
 DeltaError is the minimal error for start of hillclimbing  
 MicroP is the propability during self mate that parameter is a micro mutation

**BPN\_DeletelImageOptimizer**  
 BPN Image Optimizer is deleted from memory

27-Aug-18

Classification with Neural  
Networks part II

45

### Back Propagation Network commands

**BPN\_CreateFeatureOptimizer** <popSize> <nrEpochs> <lowConf> <trainCIS>  
 <evalCIS> <hidden1Low> <hidden1High> <hidden2Low> <hidden2High>  
 <learnRateLow> <learnRateHigh> <momentumLow> <momentumHigh>  
 <selectedTab>

Create a BPN Feature Optimizer (genetic algorithm)  
 popSize: the population size for the GA algorithm  
 nrEpochs: the number of epochs the BPN is trained before being evaluated  
 lowConf: the low confidency factor used in the evaluation  
 trainCIS: the name of the CIS to be used for training  
 evalCIS: the name of the CIS to be used for evaluation  
 hidden1Low: lowest value for size first hidden layer  
 hidden1High: highest value for size first hidden layer  
 hidden2Low: lowest value for size second hidden layer  
 hidden2High: highest value for size second hidden layer  
 learnRateLow: lowest value for learn rate  
 learnRateHigh: highest value for learn rate  
 momentumLow: lowest value for momentum  
 momentumHigh: highest value for momentum  
 selectedTab: selection of features with "01" string.

27-Aug-18

Classification with Neural  
Networks part II

46

### Back Propagation Network commands

**BPN\_FeatureOptimize** <nrGenerations> <minError> <deltaError> <microP>

Optimize for nrGenerations generations

Function result: <error> <hidden1> <hidden2> <learnRate> <momentum>

MinError is the minimal target error

DeltaError is the minimal error for start of hillclimbing

MicroP is the propability during self mate that parameter is a micro mutation

**BPN\_DeleteFeatureOptimizer**

BPN Feature Optimizer is deleted from memory

27-Aug-18

Classification with Neural  
Networks part II

47

### 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**

27-Aug-18

Classification with Neural  
Networks part II

48



### 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:  $x, \sin(x)$ 
  - $\$in = \text{random } -1 \ 1$
  - $\$out = \sin \$in$
  - Note the behaviour of the training error ( $\$t\_err$ )
- Test the BPN with 5 random values
  - $\$in = \text{random } -1 \ 1$
  - $\$res = \text{sinus of } \$in \text{ learned by BPN}$
  - $\$exp = \sin \$in$
  - $\$error = \text{error in result}$
- use script `bpn_sinus.jls`

27-Aug-18

Classification with Neural  
Networks part II

49

### Example: learning sinus function

The screenshot shows the VisionLab V3.33 interface. On the left, the 'Variables' window displays the following values:

```

error = -0.000393
exp = 0.662654
i = 0
l_max = 1
l_min = -1
n = 0.724357
layerTab = 1 100 1
out = 0.877469
res = 0.662261
t_err = 3.6264e-006
  
```

On the right, the 'Internal script' window shows the code being executed, with red annotations:

```

// create bpn
layerTab = "1 100 1"
bpn_create bpn_sin Hyperbolic NoBias 1 (-1,1) -1 1 layerTab
// learn 100 x
i_min = -1
i_max = 1
for $i = 0 to 100 do
  $in = random $i_min $i_max
  $out = sin $in
  $t_err = bpn_train bpn_sin 0.01 0.2 $in $out
  syncvars
  // break
endfor $i
// test 5x
for $i = 0 to 5 do
  $in = random $i_min $i_max
  $exp = sin $in
  $res = bpn_produce bpn_sin $in
  removefirstword $res
  $error = $res - $exp
  syncvars
  break
endfor $i
  
```

Red annotations in the script window indicate: 'Create bpn' (next to `bpn_create`), 'Train bpn' (next to the `bpn_train` loop), and 'Test bpn' (next to the `bpn_produce` loop).

27-Aug-18

Classification with Neural  
Networks part II

50

**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

27-Aug-18

Classification with Neural  
Networks part II

51

**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

27-Aug-18

Classification with Neural  
Networks part II

52

**script: gen\_ocr1\_cis.jls (2)**

```

lread org $imagenname
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**

27-Aug-18

Classification with Neural  
Networks part II

53

**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 Bytelmage
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 Bytelmage**

**Normalize contrast and size**

**Add image with digit to CIS**

**Write CIS to file**

**Set file path server to original path**

27-Aug-18

Classification with Neural  
Networks part II

54

### 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

27-Aug-18

Classification with Neural  
Networks part II

55

### Script bpn\_cis\_digits.jls (1)

```

initrandomgen 0
$debug = true
$path = lpwd
$oldpath = pwd
cwd $path

$ CisName = small.cis
$h_norm = 24
$w_norm = 16
$size = $h_norm * $w_norm
$base = number
cis_create $CisName Bytelmage 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

```

Save old file path server

File path server = file path client

Create CIS and add classes

Get class table from cis

27-Aug-18

Classification with Neural  
Networks part II

56

### Script bpn\_cis\_digits.jls (2)

```

lread org numbers.jl          Find the image position of the digits
copy org thres                Note: blobs are sorted on x coordinate
threshold thres 0 250         of top left
removeblobs thres EightConnected Area 300 10000 UseX
labelblobs thres EightConnected
$ana = blobanalysis thres SortDown TopLeft UseX Height TopLeft Width
$base = number

```

27-Aug-18

Classification with Neural  
Networks part II

57

### Script bpn\_cis\_digits.jls (3)

```

for $label = 0 to 9 do
  removefirstword &$ana
  $h = removefirstword &$ana
  $tl = removefirstword &$ana          Cut out digits with a margin of 1 pixel
  $t = getnthfromvector 1 $tl          Convert to ByteImage
  $l = getnthfromvector 2 $tl          Normalize contrast and size
  $w = removefirstword &$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                      Add image with digit to CIS
  endif
  cis_addimage $cisName $name $name
endfor $label

```

27-Aug-18

Classification with Neural  
Networks part II

58

### Script bpn\_cis\_digits.jls (4)

```

cis_writetofile $cisName $cisName Save CIS to file
cis_delete $cisName Delete CIS

cis_readfromfile cis_small small.cis Read CIS from file

$h = cis_getimageheight cis_small Retrieve height/width info fromCIS
$w = cis_getimagewidth cis_small
$size = $h * $w

Create BPN
$hidden = 20
bpn_createimageclassifier bpn_small Bytelmage $size $hidden 0 10 NoBias 0 255
$base = number
$nrOfEpochs = 10
for $epoch = 1 to 10 do Train BPN using CIS
  $a_err = bpn_traincis bpn_small cis_small $nrOfEpochs 0.001 0.0001
  if $debug then
    syncvars
  endif
endfor $epoch

```

27-Aug-18

Classification with Neural  
Networks part II

59

### 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 Classify image using BPN, result is classId and confidence
    syncvars
    break Use CIS to convert classId to className
  endif
endfor $t

$res = bpn_evaluatecis bpn_small cis_small 1.0 HighDetails

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

```

27-Aug-18

Classification with Neural  
Networks part II

60

### Script bpn\_cis\_digits.jls (6)

```
bpn_writetofile bpn_small small.bpn Write BPN to file  
bpn_delete bpn_small Delete BPN and CIS from memory  
cis_delete cis_small  
  
cwd $oldpath Set file path server to original path
```

27-Aug-18

Classification with Neural  
Networks part II

61

### 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**

27-Aug-18

Classification with Neural  
Networks part II

62

## 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

27-Aug-18

Classification with Neural  
Networks part II

63

## 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
$name_base = SumRow
for $h = 1 to $h_norm do
  $fname = $name_base . $h
  cfs_addfeature cfs $fname
endfor $h
$name_base = sumCol
for $c = 1 to $w_norm do
  $fname = $name_base . $c
  cfs_addfeature cfs $fname
endfor $h
$featurenametab = cfs_getfeaturenametab cfs

```

Add features to CFS

Get feature table from CFS

27-Aug-18

Classification with Neural  
Networks part II

64



**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

27-Aug-18

Classification with Neural  
Networks part II

65

**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
$class_name = cfs_classname cfs $num
contraststretch roi 0 255
zoom roi new $h_norm $w_norm NearestPixelInterpolation
cfs_addimage cfs new $class_name
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

27-Aug-18

Classification with Neural  
Networks part II

66

**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 Save and read CFS
cfs_writetofile cfs test.cfs Save and read CFS
cfs_readfromfile cfs test.cfs
break
$nr_epochs = 40
$learnrate = 0.005
$momentum = 0.01 Create BPN
$nr_hidden = 200
bpn_createfeatureclassifier bpn_feature_test cfs $nr_hidden 0 Bias
BPN_SetNameLearnSet bpn_feature_test test.cfs
Train BPN using CFS
for $epoch = 1 to $nr_epochs do
  $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

```

27-Aug-18

Classification with Neural  
Networks part II

67

**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
Classify image with selected features
lread roi eight.jl

$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

cwd $oldpath Set file path server to original path

```

27-Aug-18

Classification with Neural  
Networks part II

68