Example: AutoencoderDeep Part 2: Custom Hyperparameter Setting

Author: Anika Terbuch

History: \change{2.0}{20-Apr-2022}{Original}

\change{2.0}{04-Nov-2022}

Source: Chair of Automation, University of Leoben, Austria

email: automation@unileoben.ac.at *url:* automation.unileoben.ac.at

This script shows the use of the framework "AutoencoderDeep".

It is the second part of the series of examples showing how the hyperparameters of the autoencoder can be customized.

It is recommended to perform some kind of hyperparameter optimization also known as hyperparameter tuning to archieve the best possible performance. The hyperparameter optimization adjusts the neural network structure to the problem on hand.

Table of Contents

Workspace

Clear the workspace.

```
clear all
close
```

Path settings

Set the current directory to the path of this file.

```
filePath = matlab.desktop.editor.getActiveFilename;
[pathstr,name,ext]  = fileparts(filePath);
parentDir=fileparts(pathstr);
```

set the current directory in matlab to the root folder of this toolbox

```
cd(parentDir);
```

Add all folders of the toolbox to the matlab path

```
addpath(genpath(parentDir));
```

Data ingestion

Comment: this data set is available from Matlab 2022a on.

This implementation was created using Matlab 2022b.

The data set consits of multi-variate time series (3 channels) with varying number of time-steps per sample (varying time-series length).

```
load WaveformData
```

Devide the data into sets for training and validation

Get the number of samples included in the data set `WaveformData`.

```
numObservations = numel(data);
```

Define the percentage of data that should be used for training.

```
splitTrain=0.1;
```

Assign the data samples based on the `splitTrain` to the training and validation set.

```
XTrain = data(1:floor(splitTrain*numObservations));
XValidation = data(floor(splitTrain*numObservations)+1:end);
```

clear the variable data

```
clear data
```

Create an autoencoder with custom hyperparameters

The structure of the toolbox is based on two object types: the hyperparameters (settings) of the neural networks and the autoencoder (neural networks) itself.

It is recommended to perform some kind of hyperparameter optimization before setting the hyperparameters and training the AutoencoderDeep.

Set customized hyperparameters

To set customized hyperparameters an object of the class `HyperparametersAED` is required. This object can be created by calling the constructor of the class.

```
hpCustom=HyperparametersAED();
```

At creation the hyperparameters are initialized with default values and can be displayed as follows:

```
hpCustom.Hyperparameters
```

```
ans = struct with fields:
           AutoencoderType: 'VAE'
             LayersEncoder: {'FC'  'LSTM'}
             LayersDecoder: {'LSTM'}
            NeuronsEncoder: [50 20]
            NeuronsDecoder: 30
                 LatentDim: 2
               NumberEpoch: 10
             NumberFeature: 1
              LearningRate: 0.0500
             MiniBatchSize: 15
      ExecutionEnvironment: 'auto'
               WeightingKL: 1
    OutputTransferFunction: 'none'
```

The hyperparameters can be changed by calling the function `setHyperparametersAED` with name-value pairs.

Lets assume we want to create an autoencoder with one Bi-LSTM layer in the encoder with 20 neurons and one Bi-LSTM layer in the decoder with 20 neurons, the funtion call would look as follows:

```
hpCustom.setHyperparametersAED('AutoencoderType','AE','LayersEncoder',{'Bi-LSTM'},'NeuronsEncoc
```

Display the changed hyperparameter settings.

```
hpCustom.Hyperparameters
```

```
ans = struct with fields:
           AutoencoderType: 'AE'
             LayersEncoder: {'Bi-LSTM'}
             LayersDecoder: {'Bi-LSTM'}
            NeuronsEncoder: 20
            NeuronsDecoder: 20
                 LatentDim: 2
               NumberEpoch: 10
             NumberFeature: 1
              LearningRate: 0.0500
             MiniBatchSize: 15
      ExecutionEnvironment: 'auto'
    OutputTransferFunction: 'none'
```

Train autoencoder with customized hyperparameters

Training an autoencoder: Call the function `trainAutoenocoderDeep` and pass the training data `XTrain` as input parameter to the function as well as the object of the class hyperparameterAED (in our case: `hpCustom`). The autoencoder is created and trained based on the values specified in the hyperparameter-object.

```
aeCustomHP=trainAutoencoderDeep(XTrain,hpCustom)
```

```
ans = struct with fields:
```

```
           AutoencoderType: 'AE'
             LayersEncoder: {'Bi-LSTM'}
             LayersDecoder: {'Bi-LSTM'}
            NeuronsEncoder: 20
            NeuronsDecoder: 20
                 LatentDim: 2
               NumberEpoch: 10
             NumberFeature: 1
              LearningRate: 0.0500
             MiniBatchSize: 15
      ExecutionEnvironment: 'auto'
    OutputTransferFunction: 'none'
Starting parallel pool (parpool) using the 'local' profile ...
Connected to the parallel pool (number of workers: 1).
aeCustomHP = 
  AutoencoderDeep with properties:

            Version: '3.0.0'
        Description: {'This object was created with the class AutoencoderDeep. This is an object-oriented implementa
            Trained: 1
    Hyperparameters: [1×1 HyperparametersAED]
```

Reconstruct / predict the data of the validation-set

Use the just trained Autoencoder to make a prediction / produce a reconstruction of the data

```
[reconstructedOutput,latentRepresentation,reconstructionErrorPerSampleNormalized,reconstructio
```

reconstructedOutput = 900×1 cell

	1
1	3×200 single
2	3×144 single
3	3×170 single
4	3×188 single
5	3×147 single
6	3×130 single
7	3×146 single
8	3×105 single
9	3×159 single
10	3×153 single
11	3×109 single
12	3×136 single
13	3×121 single
14	3×175 single
15	3×183 single
16	3×187 single
17	3×108 single

	1
18	3×106 single
19	3×194 single
20	3×118 single
21	3×133 single
22	3×170 single
23	3×197 single
24	3×179 single
25	3×181 single
26	3×167 single
27	3×120 single
28	3×184 single
29	3×111 single
30	3×166 single
31	3×147 single
32	3×181 single
33	3×141 single
34	3×154 single
35	3×111 single
36	3×192 single
37	3×121 single
38	3×160 single
39	3×110 single
40	3×175 single
41	3×132 single
42	3×184 single
43	3×103 single
44	3×174 single
45	3×117 single
46	3×198 single
47	3×196 single
48	3×176 single
49	3×111 single
50	3×119 single

	1
51	3×183 single
52	3×111 single
53	3×181 single
54	3×149 single
55	3×173 single
56	3×186 single
57	3×171 single
58	3×148 single
59	3×158 single
60	3×188 single
61	3×192 single
62	3×162 single
63	3×169 single
64	3×150 single
65	3×175 single
66	3×198 single
67	3×160 single
68	3×104 single
69	3×199 single
70	3×185 single
71	3×132 single
72	3×198 single
73	3×168 single
74	3×110 single
75	3×182 single
76	3×123 single
77	3×180 single
78	3×159 single
79	3×122 single
80	3×117 single
81	3×161 single
82	3×188 single
83	3×134 single

	1
84	3×156 single
85	3×180 single
86	3×132 single
87	3×190 single
88	3×170 single
89	3×160 single
90	3×143 single
91	3×145 single
92	3×156 single
93	3×120 single
94	3×126 single
95	3×177 single
96	3×120 single
97	3×191 single
98	3×127 single
99	3×172 single
100	3×171 single

⋮

```
latentRepresentation = 900×1 cell
```

	1
1	2×200 single
2	2×144 single
3	2×170 single
4	2×188 single
5	2×147 single
6	2×130 single
7	2×146 single
8	2×105 single
9	2×159 single
10	2×153 single
11	2×109 single
12	2×136 single
13	2×121 single
14	2×175 single

	1
15	2×183 single
16	2×187 single
17	2×108 single
18	2×106 single
19	2×194 single
20	2×118 single
21	2×133 single
22	2×170 single
23	2×197 single
24	2×179 single
25	2×181 single
26	2×167 single
27	2×120 single
28	2×184 single
29	2×111 single
30	2×166 single
31	2×147 single
32	2×181 single
33	2×141 single
34	2×154 single
35	2×111 single
36	2×192 single
37	2×121 single
38	2×160 single
39	2×110 single
40	2×175 single
41	2×132 single
42	2×184 single
43	2×103 single
44	2×174 single
45	2×117 single
46	2×198 single
47	2×196 single

	1
48	2×176 single
49	2×111 single
50	2×119 single
51	2×183 single
52	2×111 single
53	2×181 single
54	2×149 single
55	2×173 single
56	2×186 single
57	2×171 single
58	2×148 single
59	2×158 single
60	2×188 single
61	2×192 single
62	2×162 single
63	2×169 single
64	2×150 single
65	2×175 single
66	2×198 single
67	2×160 single
68	2×104 single
69	2×199 single
70	2×185 single
71	2×132 single
72	2×198 single
73	2×168 single
74	2×110 single
75	2×182 single
76	2×123 single
77	2×180 single
78	2×159 single
79	2×122 single
80	2×117 single

	1
81	2×161 single
82	2×188 single
83	2×134 single
84	2×156 single
85	2×180 single
86	2×132 single
87	2×190 single
88	2×170 single
89	2×160 single
90	2×143 single
91	2×145 single
92	2×156 single
93	2×120 single
94	2×126 single
95	2×177 single
96	2×120 single
97	2×191 single
98	2×127 single
99	2×172 single
100	2×171 single

⋮

```
reconstructionErrorPerSampleNormalized = 900×1
    0.3422
    0.3474
    0.2941
    0.3942
    0.2974
    0.3962
    0.3275
    0.3500
    0.3134
    0.3498
      ⋮
reconstructionErrorPerChannelNormalized = 900×3
    0.1514    0.0987    0.0921
    0.1330    0.1508    0.0637
    0.1506    0.0883    0.0552
    0.1472    0.1554    0.0916
    0.1427    0.0950    0.0597
    0.1602    0.1548    0.0811
    0.1530    0.1024    0.0721
    0.1626    0.1234    0.0641
    0.1691    0.0764    0.0679
```

```
    0.1472    0.0931    0.1094
      ⋮
failedIndex =

     []
originalInput = 900×1 cell
```

	1
1	3×200 double
2	3×144 double
3	3×170 double
4	3×188 double
5	3×147 double
6	3×130 double
7	3×146 double
8	3×105 double
9	3×159 double
10	3×153 double
11	3×109 double
12	3×136 double
13	3×121 double
14	3×175 double
15	3×183 double
16	3×187 double
17	3×108 double
18	3×106 double
19	3×194 double
20	3×118 double
21	3×133 double
22	3×170 double
23	3×197 double
24	3×179 double
25	3×181 double
26	3×167 double
27	3×120 double
28	3×184 double
29	3×111 double

	1
30	3×166 double
31	3×147 double
32	3×181 double
33	3×141 double
34	3×154 double
35	3×111 double
36	3×192 double
37	3×121 double
38	3×160 double
39	3×110 double
40	3×175 double
41	3×132 double
42	3×184 double
43	3×103 double
44	3×174 double
45	3×117 double
46	3×198 double
47	3×196 double
48	3×176 double
49	3×111 double
50	3×119 double
51	3×183 double
52	3×111 double
53	3×181 double
54	3×149 double
55	3×173 double
56	3×186 double
57	3×171 double
58	3×148 double
59	3×158 double
60	3×188 double
61	3×192 double
62	3×162 double

	1
63	3×169 double
64	3×150 double
65	3×175 double
66	3×198 double
67	3×160 double
68	3×104 double
69	3×199 double
70	3×185 double
71	3×132 double
72	3×198 double
73	3×168 double
74	3×110 double
75	3×182 double
76	3×123 double
77	3×180 double
78	3×159 double
79	3×122 double
80	3×117 double
81	3×161 double
82	3×188 double
83	3×134 double
84	3×156 double
85	3×180 double
86	3×132 double
87	3×190 double
88	3×170 double
89	3×160 double
90	3×143 double
91	3×145 double
92	3×156 double
93	3×120 double
94	3×126 double
95	3×177 double

	1
96	3×120 double
97	3×191 double
98	3×127 double
99	3×172 double
100	3×171 double

⋮