

## New Regressor Evaluator

```

1 %% !header!
2 NNRegressorEvaluator < NNEvaluator (nne,
3   evaluator for a neural network
4   regressor) evaluates the performance
5   of a neural network regressor with a
6   specific dataset.
7
8 %% !description!
9 This evaluator evaluates the performance
10  of a neural network regressor
11  on root mean square error (RMSE).
12
13 %% !props!
14
15 %% !prop!
16 RMSE (result, scalar) is the root mean
17  squared error between targets and
18  predictions for validation set.
19
20 %% !calculate!
21 if nne.get('GR_PREDICTION').get('SUB_DICT'
22   ').length() == 0
23   value = 0;
24 else
25   preds = cellfun(@(x) cell2mat(x.get(
26     'PREDICTION')), nne.memorize(
27     'GR_PREDICTION').get('SUB_DICT').
28     getItems(), 'UniformOutput', false
29   );
30   preds = cell2mat(preds);
31   targets = cellfun(@(x) cell2mat(x.get(
32     'TARGET')), nne.get(
33     'GR_PREDICTION').get('SUB_DICT').
34     getItems(), 'UniformOutput', false
35   );
36 end
37
38 %% !examples!
39 % Example 1: Evaluate a neural network regressor
40 % with a specific dataset.
41 %> nne = nntrain4();
42 %> nre = nnregress(nne, 'GR_PREDICTION');
43 %> rmse = nre(nne);
44 %> rmse
45 %> ans = 0.0000
46
47 %% !methods!
48
49 %% !methods!
50
51 %% !methods!
52
53 %% !methods!
54
55 %% !methods!
56
57 %% !methods!
58
59 %% !methods!
60
61 %% !methods!
62
63 %% !methods!
64
65 %% !methods!
66
67 %% !methods!
68
69 %% !methods!
70
71 %% !methods!
72
73 %% !methods!
74
75 %% !methods!
76
77 %% !methods!
78
79 %% !methods!
80
81 %% !methods!
82
83 %% !methods!
84
85 %% !methods!
86
87 %% !methods!
88
89 %% !methods!
90
91 %% !methods!
92
93 %% !methods!
94
95 %% !methods!
96
97 %% !methods!
98
99 %% !methods!
100
101 %% !methods!
102
103 %% !methods!
104
105 %% !methods!
106
107 %% !methods!
108
109 %% !methods!
110
111 %% !methods!
112
113 %% !methods!
114
115 %% !methods!
116
117 %% !methods!
118
119 %% !methods!
120
121 %% !methods!
122
123 %% !methods!
124
125 %% !methods!
126
127 %% !methods!
128
129 %% !methods!
130
131 %% !methods!
132
133 %% !methods!
134
135 %% !methods!
136
137 %% !methods!
138
139 %% !methods!
140
141 %% !methods!
142
143 %% !methods!
144
145 %% !methods!
146
147 %% !methods!
148
149 %% !methods!
150
151 %% !methods!
152
153 %% !methods!
154
155 %% !methods!
156
157 %% !methods!
158
159 %% !methods!
160
161 %% !methods!
162
163 %% !methods!
164
165 %% !methods!
166
167 %% !methods!
168
169 %% !methods!
170
171 %% !methods!
172
173 %% !methods!
174
175 %% !methods!
176
177 %% !methods!
178
179 %% !methods!
180
181 %% !methods!
182
183 %% !methods!
184
185 %% !methods!
186
187 %% !methods!
188
189 %% !methods!
190
191 %% !methods!
192
193 %% !methods!
194
195 %% !methods!
196
197 %% !methods!
198
199 %% !methods!
200
201 %% !methods!
202
203 %% !methods!
204
205 %% !methods!
206
207 %% !methods!
208
209 %% !methods!
210
211 %% !methods!
212
213 %% !methods!
214
215 %% !methods!
216
217 %% !methods!
218
219 %% !methods!
220
221 %% !methods!
222
223 %% !methods!
224
225 %% !methods!
226
227 %% !methods!
228
229 %% !methods!
230
231 %% !methods!
232
233 %% !methods!
234
235 %% !methods!
236
237 %% !methods!
238
239 %% !methods!
240
241 %% !methods!
242
243 %% !methods!
244
245 %% !methods!
246
247 %% !methods!
248
249 %% !methods!
250
251 %% !methods!
252
253 %% !methods!
254
255 %% !methods!
256
257 %% !methods!
258
259 %% !methods!
260
261 %% !methods!
262
263 %% !methods!
264
265 %% !methods!
266
267 %% !methods!
268
269 %% !methods!
270
271 %% !methods!
272
273 %% !methods!
274
275 %% !methods!
276
277 %% !methods!
278
279 %% !methods!
280
281 %% !methods!
282
283 %% !methods!
284
285 %% !methods!
286
287 %% !methods!
288
289 %% !methods!
290
291 %% !methods!
292
293 %% !methods!
294
295 %% !methods!
296
297 %% !methods!
298
299 %% !methods!
299
300 %% !methods!
301
302 %% !methods!
303
304 %% !methods!
305
306 %% !methods!
307
308 %% !methods!
309
310 %% !methods!
311
312 %% !methods!
313
314 %% !methods!
315
316 %% !methods!
317
318 %% !methods!
319
320 %% !methods!
321
322 %% !methods!
323
324 %% !methods!
325
326 %% !methods!
327
328 %% !methods!
329
330 %% !methods!
331
332 %% !methods!
333
334 %% !methods!
335
336 %% !methods!
337
338 %% !methods!
339
340 %% !methods!
341
342 %% !methods!
343
344 %% !methods!
345
346 %% !methods!
347
348 %% !methods!
349
350 %% !methods!
351
352 %% !methods!
353
354 %% !methods!
355
356 %% !methods!
357
358 %% !methods!
359
360 %% !methods!
361
362 %% !methods!
363
364 %% !methods!
365
366 %% !methods!
367
368 %% !methods!
369
370 %% !methods!
371
372 %% !methods!
373
374 %% !methods!
375
376 %% !methods!
377
378 %% !methods!
379
380 %% !methods!
381
382 %% !methods!
383
384 %% !methods!
385
386 %% !methods!
387
388 %% !methods!
389
390 %% !methods!
391
392 %% !methods!
393
394 %% !methods!
395
396 %% !methods!
397
398 %% !methods!
399
400 %% !methods!
401
402 %% !methods!
403
404 %% !methods!
405
406 %% !methods!
407
408 %% !methods!
409
410 %% !methods!
411
412 %% !methods!
413
414 %% !methods!
415
416 %% !methods!
417
418 %% !methods!
419
420 %% !methods!
421
422 %% !methods!
423
424 %% !methods!
425
426 %% !methods!
427
428 %% !methods!
429
430 %% !methods!
431
432 %% !methods!
433
434 %% !methods!
435
436 %% !methods!
437
438 %% !methods!
439
440 %% !methods!
441
442 %% !methods!
443
444 %% !methods!
445
446 %% !methods!
447
448 %% !methods!
449
450 %% !methods!
451
452 %% !methods!
453
454 %% !methods!
455
456 %% !methods!
457
458 %% !methods!
459
460 %% !methods!
461
462 %% !methods!
463
464 %% !methods!
465
466 %% !methods!
467
468 %% !methods!
469
470 %% !methods!
471
472 %% !methods!
473
474 %% !methods!
475
476 %% !methods!
477
478 %% !methods!
479
480 %% !methods!
481
482 %% !methods!
483
484 %% !methods!
485
486 %% !methods!
487
488 %% !methods!
489
490 %% !methods!
491
492 %% !methods!
493
494 %% !methods!
495
496 %% !methods!
497
498 %% !methods!
499
500 %% !methods!
501
502 %% !methods!
503
504 %% !methods!
505
506 %% !methods!
507
508 %% !methods!
509
510 %% !methods!
511
512 %% !methods!
513
514 %% !methods!
515
516 %% !methods!
517
518 %% !methods!
519
520 %% !methods!
521
522 %% !methods!
523
524 %% !methods!
525
526 %% !methods!
527
528 %% !methods!
529
530 %% !methods!
531
532 %% !methods!
533
534 %% !methods!
535
536 %% !methods!
537
538 %% !methods!
539
540 %% !methods!
541
542 %% !methods!
543
544 %% !methods!
545
546 %% !methods!
547
548 %% !methods!
549
550 %% !methods!
551
552 %% !methods!
553
554 %% !methods!
555
556 %% !methods!
557
558 %% !methods!
559
560 %% !methods!
561
562 %% !methods!
563
564 %% !methods!
565
566 %% !methods!
567
568 %% !methods!
569
570 %% !methods!
571
572 %% !methods!
573
574 %% !methods!
575
576 %% !methods!
577
578 %% !methods!
579
580 %% !methods!
581
582 %% !methods!
583
584 %% !methods!
585
586 %% !methods!
587
588 %% !methods!
589
590 %% !methods!
591
592 %% !methods!
593
594 %% !methods!
595
596 %% !methods!
597
598 %% !methods!
599
600 %% !methods!
601
602 %% !methods!
603
604 %% !methods!
605
606 %% !methods!
607
608 %% !methods!
609
610 %% !methods!
611
612 %% !methods!
613
614 %% !methods!
615
616 %% !methods!
617
618 %% !methods!
619
620 %% !methods!
621
622 %% !methods!
623
624 %% !methods!
625
626 %% !methods!
627
628 %% !methods!
629
630 %% !methods!
631
632 %% !methods!
633
634 %% !methods!
635
636 %% !methods!
637
638 %% !methods!
639
640 %% !methods!
641
642 %% !methods!
643
644 %% !methods!
645
646 %% !methods!
647
648 %% !methods!
649
650 %% !methods!
651
652 %% !methods!
653
654 %% !methods!
655
656 %% !methods!
657
658 %% !methods!
659
660 %% !methods!
661
662 %% !methods!
663
664 %% !methods!
665
666 %% !methods!
667
668 %% !methods!
669
670 %% !methods!
671
672 %% !methods!
673
674 %% !methods!
675
676 %% !methods!
677
678 %% !methods!
679
680 %% !methods!
681
682 %% !methods!
683
684 %% !methods!
685
686 %% !methods!
687
688 %% !methods!
689
690 %% !methods!
691
692 %% !methods!
693
694 %% !methods!
695
696 %% !methods!
697
698 %% !methods!
699
700 %% !methods!
701
702 %% !methods!
703
704 %% !methods!
705
706 %% !methods!
707
708 %% !methods!
709
710 %% !methods!
711
712 %% !methods!
713
714 %% !methods!
715
716 %% !methods!
717
718 %% !methods!
719
720 %% !methods!
721
722 %% !methods!
723
724 %% !methods!
725
726 %% !methods!
727
728 %% !methods!
729
730 %% !methods!
731
732 %% !methods!
733
734 %% !methods!
735
736 %% !methods!
737
738 %% !methods!
739
740 %% !methods!
741
742 %% !methods!
743
744 %% !methods!
745
746 %% !methods!
747
748 %% !methods!
749
750 %% !methods!
751
752 %% !methods!
753
754 %% !methods!
755
756 %% !methods!
757
758 %% !methods!
759
760 %% !methods!
761
762 %% !methods!
763
764 %% !methods!
765
766 %% !methods!
767
768 %% !methods!
769
770 %% !methods!
771
772 %% !methods!
773
774 %% !methods!
775
776 %% !methods!
777
778 %% !methods!
779
779
780 %% !methods!
781
782 %% !methods!
783
784 %% !methods!
785
786 %% !methods!
787
788 %% !methods!
789
789
790 %% !methods!
791
792 %% !methods!
793
794 %% !methods!
795
796 %% !methods!
797
798 %% !methods!
799
799
800 %% !methods!
801
802 %% !methods!
803
804 %% !methods!
805
806 %% !methods!
807
808 %% !methods!
809
809
810 %% !methods!
811
812 %% !methods!
813
814 %% !methods!
815
816 %% !methods!
817
818 %% !methods!
819
819
820 %% !methods!
821
822 %% !methods!
823
824 %% !methods!
825
826 %% !methods!
827
828 %% !methods!
829
829
830 %% !methods!
831
832 %% !methods!
833
834 %% !methods!
835
836 %% !methods!
837
838 %% !methods!
839
839
840 %% !methods!
841
842 %% !methods!
843
844 %% !methods!
845
846 %% !methods!
847
848 %% !methods!
849
849
850 %% !methods!
851
852 %% !methods!
853
854 %% !methods!
855
856 %% !methods!
857
858 %% !methods!
859
859
860 %% !methods!
861
862 %% !methods!
863
864 %% !methods!
865
866 %% !methods!
867
868 %% !methods!
869
869
870 %% !methods!
871
872 %% !methods!
873
874 %% !methods!
875
876 %% !methods!
877
878 %% !methods!
879
879
880 %% !methods!
881
882 %% !methods!
883
884 %% !methods!
885
886 %% !methods!
887
888 %% !methods!
889
889
890 %% !methods!
891
892 %% !methods!
893
894 %% !methods!
895
896 %% !methods!
897
898 %% !methods!
899
899
900 %% !methods!
901
902 %% !methods!
903
904 %% !methods!
905
906 %% !methods!
907
908 %% !methods!
909
909
910 %% !methods!
911
912 %% !methods!
913
914 %% !methods!
915
916 %% !methods!
917
918 %% !methods!
919
919
920 %% !methods!
921
922 %% !methods!
923
924 %% !methods!
925
926 %% !methods!
927
928 %% !methods!
929
929
930 %% !methods!
931
932 %% !methods!
933
934 %% !methods!
935
936 %% !methods!
937
938 %% !methods!
939
939
940 %% !methods!
941
942 %% !methods!
943
944 %% !methods!
945
946 %% !methods!
947
948 %% !methods!
949
949
950 %% !methods!
951
952 %% !methods!
953
954 %% !methods!
955
956 %% !methods!
957
958 %% !methods!
959
959
960 %% !methods!
961
962 %% !methods!
963
964 %% !methods!
965
966 %% !methods!
967
968 %% !methods!
969
969
970 %% !methods!
971
972 %% !methods!
973
974 %% !methods!
975
976 %% !methods!
977
978 %% !methods!
979
979
980 %% !methods!
981
982 %% !methods!
983
984 %% !methods!
985
986 %% !methods!
987
988 %% !methods!
989
989
990 %% !methods!
991
992 %% !methods!
993
994 %% !methods!
995
996 %% !methods!
997
998 %% !methods!
999
999
1000 %% !methods!
```

```

    );
19 targets = cell2mat(targets);
20 value = double(sqrt(mean((preds -
    targets).^2)));
21 end
22
23 %% prop!
24 SCATTER_CHART (result, matrix) creates a
    scatter chart with circular markers at
    the locations specified by
    predictions and targets.
25 %% calculate!
26 if nne.get('GR_PREDICTION').get('SUB_DICT'
    ').length() == 0
27     value = 0;
28 else
29     preds = cellfun(@(x) cell2mat(x.get(
        'PREDICTION')), nne.memorize('
        GR_PREDICTION').get('SUB_DICT').
        getItems(), 'UniformOutput', false
    );
30     preds = cell2mat(preds);
31     targets = cellfun(@(x) cell2mat(x.get(
        'TARGET')), nne.get(
        'GR_PREDICTION').get('SUB_DICT').
        getItems(), 'UniformOutput', false
    );
32     targets = cell2mat(targets);
33     value = double([preds' targets']);
34 end
35 %% gui!
36 pr = PanelPropMatrix('EL', nne, 'PROP',
    NNRegressorEvaluator.SCATTER_CHART,
    ...
37     'ROWNAME', char("cellfun(@(x) x.get('
        ID'), pr.get('EL').memorize('GR') .
        get('SUB_DICT').getItems(), '
        UniformOutput', false)'), ...
38     'COLUMNNAME', char("{'Prediction', '"
        Target'}"), ...
39     varargin{:});
```

```

40
41 %%% ;prop!
42 PFSP (gui, item) contains the panel
43   figure of the scatter plot.
44 %%% ;settings!
45 'PFScatterPlot'
46 %%% ;postprocessing!
47 if ~braph2_testing % to avoid problems
48   with isequal when the element is
49   recursive
50   nne.memorize('PFSP').set('NNE', nne)
51 end
52 %%% ;gui!
53 pr = PanelPropItem('EL', nne, 'PROP',
54   NNRegressorEvaluator.PFSP, ...
55   'GUICLASS', 'GUIFig', ...
56   varargin{:});
57
58 %% ;props_update!
59
60 %%% ;prop!
61 NN (data, item) is a neural network model
62   that needs to be evaluated.
63 %%% ;settings!
64 'NNRegressorDNN'
65 %%% ;default!
66 NNRegressorDNN()
67
68 %%% ;prop!
69 GR_PREDICTION (result, item) is a group
70   of NN subjects containing the
71   prediction from the neural network.
72 %%% ;settings!
73 'NNGroup'
74 %%% ;calculate!
75 if nne.get('GR').get('SUB_DICT').length()
76   == 0
77   value = NNGroup();
78 else
79   nn = nne.memorize('NN');
80   nn_gr = nne.get('GR');

```

```

73   inputs = nn.reconstruct_inputs(nn_gr)
74   ;
75   net = nn.get('MODEL');
76   if isa(net, 'NoValue') || ~BRAPH2.
77     installed('NN', 'msgbox')
78     predictions = zeros(nn_gr.get(
79       'SUB_DICT').length(), 1);
80   else
81     predictions = net.predict(inputs)
82   ;
83 end
84 nn_gr_pred = NNGroup( ...
85   'SUB_CLASS', nn_gr.get('SUB_CLASS'
86   '), ...
87   'SUB_DICT', IndexedDictionary('
88   IT_CLASS', 'Subject') ...
89 );
90 nn_gr_pred.set( ...
91   'ID', nn_gr.get('ID'), ...
92   'LABEL', nn_gr.get('LABEL'), ...
93   'NOTES', nn_gr.get('NOTES'), ...
94   'FEATURE_SELECTION_MASK', nn_gr.
95     get('FEATURE_SELECTION_MASK')
96   ...
97 );
98 % add subjects, it has to be created
99 % as new subjects
100 sub_dict = nn_gr_pred.get('SUB_DICT')
101 ;
102 subs = nn_gr.get('SUB_DICT').getItems()
103 ();
104 for i = 1:length(subs)
105   sub = NNSubject( ...
106     'ID', [subs{i}.get('ID') ' in
107       nn_gr.get('ID')], ...
108     'BA', subs{i}.get('BA'), ...
109     'age', subs{i}.get('age'),
110     ...
111     'sex', subs{i}.get('sex'),
112   );
113 end

```

```
    ...
101     'input', subs{i}.get('input')
102     , ...
103     'INPUT_TYPE', subs{i}.get(
104         'INPUT_TYPE'), ...
105     'INPUT_LABEL', subs{i}.get(
106         'INPUT_LABEL'), ...
107     'PREDICTION', {predictions(i,
108         :)}, ...
109     'TARGET', subs{i}.get('TARGET
110         '), ...
111     'TARGET_NAME', subs{i}.get(
112         'TARGET_NAME') ...
113     );
114     sub_dict.add(sub);
115 end
116 nn_gr_pred.set('SUB_DICT', sub_dict);
117
118 value = nn_gr_pred;
end

%% staticmethods!
function lbls = measure_types()
    lbls = {'Global', 'Nodal', 'Binodal'
        };
end
```