matrix-common.h

Go to the documentation of this file.

```
1  // matrix/matrix-common.h
2
3  // Copyright 2009-2011  Microsoft Corporation
4
5  // See ../../COPYING for clarification regarding multiple authors
6  //
7  // Licensed under the Apache License, Version 2.0 (the "License");
8  // you may not use this file except in compliance with the License.
9  // You may obtain a copy of the License at
10
11 //  http://www.apache.org/licenses/LICENSE-2.0
12
13 // THIS CODE IS PROVIDED *AS IS* BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
14 // KIND, EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED
15 // WARRANTIES OR CONDITIONS OF TITLE, FITNESS FOR A PARTICULAR PURPOSE,
16 // MERCHANTABLITY OR NON-INFRINGEMENT.
17 // See the Apache 2 License for the specific language governing permissions and
18 // limitations under the License.
19 #ifndef KALDI_MATRIX_MATRIX_COMMON_H_
20 #define KALDI_MATRIX_MATRIX_COMMON_H_
21
22 // This file contains some #includes, forward declarations
23 // and typedefs that are needed by all the main header
24 // files in this directory.
25
26 #include "base/kaldi-common.h"
27 #include "matrix/kaldi-blas.h"
28
29 namespace kaldi {
30 typedef enum {
31   kTrans    = CblasTrans,
32   kNoTrans = CblasNoTrans
33 } MatrixTransposeType;
34
35 typedef enum {
36   kSetZero,
37   kUndefined,
38   kCopyData
39 } MatrixResizeType;
40
41 typedef enum {
42   kTakeLower,
43   kTakeUpper,
44   kTakeMean,
45   kTakeMeanAndCheck
46 } SpCopyType;
47
48 template<typename Real> class VectorBase;
49 template<typename Real> class Vector;
50 template<typename Real> class SubVector;
51 template<typename Real> class MatrixBase;
52 template<typename Real> class SubMatrix;
53 template<typename Real> class Matrix;
54 template<typename Real> class SpMatrix;
55 template<typename Real> class TpMatrix;
56 template<typename Real> class PackedMatrix;
57
58 // these are classes that won't be defined in this
59 // directory; they're mostly needed for friend declarations.
60 template<typename Real> class CuMatrixBase;
61 template<typename Real> class CuSubMatrix;
62 template<typename Real> class CuMatrix;
63 template<typename Real> class CuVectorBase;
64 template<typename Real> class CuSubVector;
65 template<typename Real> class CuVector;
66 template<typename Real> class CuPackedMatrix;
67 template<typename Real> class CuSpMatrix;
68 template<typename Real> class CuTpMatrix;
```

```
 69
 70 class CompressedMatrix;
 71
 73 template<typename T> class OtherReal { };  // useful in reading+writing routines
 74                                            // to switch double and float.
 76 template<> class OtherReal<float> {
 77  public:
 78   typedef double Real;
 79 };
 81 template<> class OtherReal<double> {
 82  public:
 83   typedef float Real;
 84 };
 85
 86
 87 typedef int32 MatrixIndexT;
 88 typedef int32 SignedMatrixIndexT;
 89 typedef uint32 UnsignedMatrixIndexT;
 90
 91 // If you want to use size_t for the index type, do as follows instead:
 92 //typedef size_t MatrixIndexT;
 93 //typedef ssize_t SignedMatrixIndexT;
 94 //typedef size_t UnsignedMatrixIndexT;
 95
 96 }
 97
 98
 99
100 #endif  // KALDI_MATRIX_MATRIX_COMMON_H_
```