JavaScript is disabled on your browser.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Tree](http://docs.google.com/package-tree.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* Prev Class
* [Next Class](http://docs.google.com/org/opencv/ml/ANN_MLP_ANNEAL.html)
* [Frames](http://docs.google.com/index.html?org/opencv/ml/ANN_MLP.html)
* [No Frames](http://docs.google.com/ANN_MLP.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#3znysh7) |
* Constr |
* [Method](#tyjcwt)
* Detail:
* [Field](#2s8eyo1) |
* Constr |
* [Method](#4i7ojhp)

org.opencv.ml

## Class ANN\_MLP

* java.lang.Object
  + [org.opencv.core.Algorithm](http://docs.google.com/org/opencv/core/Algorithm.html)
    - [org.opencv.ml.StatModel](http://docs.google.com/org/opencv/ml/StatModel.html)
      * org.opencv.ml.ANN\_MLP
* Direct Known Subclasses: [ANN\_MLP\_ANNEAL](http://docs.google.com/org/opencv/ml/ANN_MLP_ANNEAL.html)  
    
  public class ANN\_MLP  
  extends [StatModel](http://docs.google.com/org/opencv/ml/StatModel.html)  
  Artificial Neural Networks - Multi-Layer Perceptrons. Unlike many other models in ML that are constructed and trained at once, in the MLP model these steps are separated. First, a network with the specified topology is created using the non-default constructor or the method ANN\_MLP::create. All the weights are set to zeros. Then, the network is trained using a set of input and output vectors. The training procedure can be repeated more than once, that is, the weights can be adjusted based on the new training data. Additional flags for StatModel::train are available: ANN\_MLP::TrainFlags. SEE: REF: ml\_intro\_ann

### Field SummaryFields

| Modifier and Type | Field and Description |
| --- | --- |
| static int | [**ANNEAL**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#ANNEAL) |
| static int | [**BACKPROP**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#BACKPROP) |
| static int | [**GAUSSIAN**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#GAUSSIAN) |
| static int | [**IDENTITY**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#IDENTITY) |
| static int | [**LEAKYRELU**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#LEAKYRELU) |
| static int | [**NO\_INPUT\_SCALE**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#NO_INPUT_SCALE) |
| static int | [**NO\_OUTPUT\_SCALE**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#NO_OUTPUT_SCALE) |
| static int | [**RELU**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#RELU) |
| static int | [**RPROP**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#RPROP) |
| static int | [**SIGMOID\_SYM**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#SIGMOID_SYM) |
| static int | [**UPDATE\_WEIGHTS**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#UPDATE_WEIGHTS) |

### Fields inherited from class org.opencv.ml.[**StatModel**](http://docs.google.com/org/opencv/ml/StatModel.html)[COMPRESSED\_INPUT](http://docs.google.com/org/opencv/ml/StatModel.html#COMPRESSED_INPUT), [PREPROCESSED\_INPUT](http://docs.google.com/org/opencv/ml/StatModel.html#PREPROCESSED_INPUT), [RAW\_OUTPUT](http://docs.google.com/org/opencv/ml/StatModel.html#RAW_OUTPUT), [UPDATE\_MODEL](http://docs.google.com/org/opencv/ml/StatModel.html#UPDATE_MODEL)

### Method SummaryMethods

| Modifier and Type | Method and Description |
| --- | --- |
| static [ANN\_MLP](http://docs.google.com/org/opencv/ml/ANN_MLP.html) | [**\_\_fromPtr\_\_**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#__fromPtr__(long))(long addr) |
| static [ANN\_MLP](http://docs.google.com/org/opencv/ml/ANN_MLP.html) | [**create**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#create())() Creates empty model Use StatModel::train to train the model, Algorithm::load<ANN\_MLP>(filename) to load the pre-trained model. |
| double | [**getAnnealCoolingRatio**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getAnnealCoolingRatio())() SEE: setAnnealCoolingRatio |
| double | [**getAnnealFinalT**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getAnnealFinalT())() SEE: setAnnealFinalT |
| double | [**getAnnealInitialT**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getAnnealInitialT())() SEE: setAnnealInitialT |
| int | [**getAnnealItePerStep**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getAnnealItePerStep())() SEE: setAnnealItePerStep |
| double | [**getBackpropMomentumScale**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getBackpropMomentumScale())() SEE: setBackpropMomentumScale |
| double | [**getBackpropWeightScale**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getBackpropWeightScale())() SEE: setBackpropWeightScale |
| [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**getLayerSizes**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getLayerSizes())() Integer vector specifying the number of neurons in each layer including the input and output layers. |
| double | [**getRpropDW0**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getRpropDW0())() SEE: setRpropDW0 |
| double | [**getRpropDWMax**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getRpropDWMax())() SEE: setRpropDWMax |
| double | [**getRpropDWMin**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getRpropDWMin())() SEE: setRpropDWMin |
| double | [**getRpropDWMinus**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getRpropDWMinus())() SEE: setRpropDWMinus |
| double | [**getRpropDWPlus**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getRpropDWPlus())() SEE: setRpropDWPlus |
| [TermCriteria](http://docs.google.com/org/opencv/core/TermCriteria.html) | [**getTermCriteria**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getTermCriteria())() SEE: setTermCriteria |
| int | [**getTrainMethod**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getTrainMethod())() Returns current training method |
| [Mat](http://docs.google.com/org/opencv/core/Mat.html) | [**getWeights**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#getWeights(int))(int layerIdx) |
| static [ANN\_MLP](http://docs.google.com/org/opencv/ml/ANN_MLP.html) | [**load**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#load(java.lang.String))(java.lang.String filepath) Loads and creates a serialized ANN from a file Use ANN::save to serialize and store an ANN to disk. |
| void | [**setActivationFunction**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setActivationFunction(int))(int type) Initialize the activation function for each neuron. |
| void | [**setActivationFunction**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setActivationFunction(int,%20double))(int type, double param1) Initialize the activation function for each neuron. |
| void | [**setActivationFunction**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setActivationFunction(int,%20double,%20double))(int type, double param1, double param2) Initialize the activation function for each neuron. |
| void | [**setAnnealCoolingRatio**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setAnnealCoolingRatio(double))(double val) getAnnealCoolingRatio SEE: getAnnealCoolingRatio |
| void | [**setAnnealFinalT**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setAnnealFinalT(double))(double val) getAnnealFinalT SEE: getAnnealFinalT |
| void | [**setAnnealInitialT**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setAnnealInitialT(double))(double val) getAnnealInitialT SEE: getAnnealInitialT |
| void | [**setAnnealItePerStep**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setAnnealItePerStep(int))(int val) getAnnealItePerStep SEE: getAnnealItePerStep |
| void | [**setBackpropMomentumScale**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setBackpropMomentumScale(double))(double val) getBackpropMomentumScale SEE: getBackpropMomentumScale |
| void | [**setBackpropWeightScale**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setBackpropWeightScale(double))(double val) getBackpropWeightScale SEE: getBackpropWeightScale |
| void | [**setLayerSizes**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setLayerSizes(org.opencv.core.Mat))([Mat](http://docs.google.com/org/opencv/core/Mat.html) \_layer\_sizes) Integer vector specifying the number of neurons in each layer including the input and output layers. |
| void | [**setRpropDW0**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setRpropDW0(double))(double val) getRpropDW0 SEE: getRpropDW0 |
| void | [**setRpropDWMax**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setRpropDWMax(double))(double val) getRpropDWMax SEE: getRpropDWMax |
| void | [**setRpropDWMin**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setRpropDWMin(double))(double val) getRpropDWMin SEE: getRpropDWMin |
| void | [**setRpropDWMinus**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setRpropDWMinus(double))(double val) getRpropDWMinus SEE: getRpropDWMinus |
| void | [**setRpropDWPlus**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setRpropDWPlus(double))(double val) getRpropDWPlus SEE: getRpropDWPlus |
| void | [**setTermCriteria**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setTermCriteria(org.opencv.core.TermCriteria))([TermCriteria](http://docs.google.com/org/opencv/core/TermCriteria.html) val) getTermCriteria SEE: getTermCriteria |
| void | [**setTrainMethod**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setTrainMethod(int))(int method) Sets training method and common parameters. |
| void | [**setTrainMethod**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setTrainMethod(int,%20double))(int method, double param1) Sets training method and common parameters. |
| void | [**setTrainMethod**](http://docs.google.com/org/opencv/ml/ANN_MLP.html#setTrainMethod(int,%20double,%20double))(int method, double param1, double param2) Sets training method and common parameters. |

### Methods inherited from class org.opencv.ml.[**StatModel**](http://docs.google.com/org/opencv/ml/StatModel.html)[calcError](http://docs.google.com/org/opencv/ml/StatModel.html#calcError(org.opencv.ml.TrainData,%20boolean,%20org.opencv.core.Mat)), [empty](http://docs.google.com/org/opencv/ml/StatModel.html#empty()), [getVarCount](http://docs.google.com/org/opencv/ml/StatModel.html#getVarCount()), [isClassifier](http://docs.google.com/org/opencv/ml/StatModel.html#isClassifier()), [isTrained](http://docs.google.com/org/opencv/ml/StatModel.html#isTrained()), [predict](http://docs.google.com/org/opencv/ml/StatModel.html#predict(org.opencv.core.Mat)), [predict](http://docs.google.com/org/opencv/ml/StatModel.html#predict(org.opencv.core.Mat,%20org.opencv.core.Mat)), [predict](http://docs.google.com/org/opencv/ml/StatModel.html#predict(org.opencv.core.Mat,%20org.opencv.core.Mat,%20int)), [train](http://docs.google.com/org/opencv/ml/StatModel.html#train(org.opencv.core.Mat,%20int,%20org.opencv.core.Mat)), [train](http://docs.google.com/org/opencv/ml/StatModel.html#train(org.opencv.ml.TrainData)), [train](http://docs.google.com/org/opencv/ml/StatModel.html#train(org.opencv.ml.TrainData,%20int))

### Methods inherited from class org.opencv.core.[**Algorithm**](http://docs.google.com/org/opencv/core/Algorithm.html)[clear](http://docs.google.com/org/opencv/core/Algorithm.html#clear()), [getDefaultName](http://docs.google.com/org/opencv/core/Algorithm.html#getDefaultName()), [getNativeObjAddr](http://docs.google.com/org/opencv/core/Algorithm.html#getNativeObjAddr()), [save](http://docs.google.com/org/opencv/core/Algorithm.html#save(java.lang.String))

### Methods inherited from class java.lang.Objectequals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

#### ANNEAL public static final int ANNEALSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.ml.ANN_MLP.ANNEAL)

#### BACKPROP public static final int BACKPROPSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.ml.ANN_MLP.BACKPROP)

#### GAUSSIAN public static final int GAUSSIANSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.ml.ANN_MLP.GAUSSIAN)

#### IDENTITY public static final int IDENTITYSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.ml.ANN_MLP.IDENTITY)

#### LEAKYRELU public static final int LEAKYRELUSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.ml.ANN_MLP.LEAKYRELU)

#### NO\_INPUT\_SCALE public static final int NO\_INPUT\_SCALESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.ml.ANN_MLP.NO_INPUT_SCALE)

#### NO\_OUTPUT\_SCALE public static final int NO\_OUTPUT\_SCALESee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.ml.ANN_MLP.NO_OUTPUT_SCALE)

#### RELU public static final int RELUSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.ml.ANN_MLP.RELU)

#### RPROP public static final int RPROPSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.ml.ANN_MLP.RPROP)

#### SIGMOID\_SYM public static final int SIGMOID\_SYMSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.ml.ANN_MLP.SIGMOID_SYM)

#### UPDATE\_WEIGHTS public static final int UPDATE\_WEIGHTSSee Also:[Constant Field Values](http://docs.google.com/constant-values.html#org.opencv.ml.ANN_MLP.UPDATE_WEIGHTS)

### Method Detail

#### \_\_fromPtr\_\_ public static [ANN\_MLP](http://docs.google.com/org/opencv/ml/ANN_MLP.html) \_\_fromPtr\_\_(long addr)

#### create public static [ANN\_MLP](http://docs.google.com/org/opencv/ml/ANN_MLP.html) create() Creates empty model Use StatModel::train to train the model, Algorithm::load<ANN\_MLP>(filename) to load the pre-trained model. Note that the train method has optional flags: ANN\_MLP::TrainFlags.Returns:automatically generated

#### getAnnealCoolingRatio public double getAnnealCoolingRatio() SEE: setAnnealCoolingRatioReturns:automatically generated

#### getAnnealFinalT public double getAnnealFinalT() SEE: setAnnealFinalTReturns:automatically generated

#### getAnnealInitialT public double getAnnealInitialT() SEE: setAnnealInitialTReturns:automatically generated

#### getAnnealItePerStep public int getAnnealItePerStep() SEE: setAnnealItePerStepReturns:automatically generated

#### getBackpropMomentumScale public double getBackpropMomentumScale() SEE: setBackpropMomentumScaleReturns:automatically generated

#### getBackpropWeightScale public double getBackpropWeightScale() SEE: setBackpropWeightScaleReturns:automatically generated

#### getLayerSizes public [Mat](http://docs.google.com/org/opencv/core/Mat.html) getLayerSizes() Integer vector specifying the number of neurons in each layer including the input and output layers. The very first element specifies the number of elements in the input layer. The last element - number of elements in the output layer. SEE: setLayerSizesReturns:automatically generated

#### getRpropDW0 public double getRpropDW0() SEE: setRpropDW0Returns:automatically generated

#### getRpropDWMax public double getRpropDWMax() SEE: setRpropDWMaxReturns:automatically generated

#### getRpropDWMin public double getRpropDWMin() SEE: setRpropDWMinReturns:automatically generated

#### getRpropDWMinus public double getRpropDWMinus() SEE: setRpropDWMinusReturns:automatically generated

#### getRpropDWPlus public double getRpropDWPlus() SEE: setRpropDWPlusReturns:automatically generated

#### getTermCriteria public [TermCriteria](http://docs.google.com/org/opencv/core/TermCriteria.html) getTermCriteria() SEE: setTermCriteriaReturns:automatically generated

#### getTrainMethod public int getTrainMethod() Returns current training methodReturns:automatically generated

#### getWeights public [Mat](http://docs.google.com/org/opencv/core/Mat.html) getWeights(int layerIdx)

#### load public static [ANN\_MLP](http://docs.google.com/org/opencv/ml/ANN_MLP.html) load(java.lang.String filepath) Loads and creates a serialized ANN from a file Use ANN::save to serialize and store an ANN to disk. Load the ANN from this file again, by calling this function with the path to the file.Parameters:filepath - path to serialized ANN Returns:automatically generated

#### setActivationFunction public void setActivationFunction(int type) Initialize the activation function for each neuron. Currently the default and the only fully supported activation function is ANN\_MLP::SIGMOID\_SYM.Parameters:type - The type of activation function. See ANN\_MLP::ActivationFunctions.

#### setActivationFunction public void setActivationFunction(int type, double param1) Initialize the activation function for each neuron. Currently the default and the only fully supported activation function is ANN\_MLP::SIGMOID\_SYM.Parameters:type - The type of activation function. See ANN\_MLP::ActivationFunctions.param1 - The first parameter of the activation function, \(\alpha\). Default value is 0.

#### setActivationFunction public void setActivationFunction(int type, double param1, double param2) Initialize the activation function for each neuron. Currently the default and the only fully supported activation function is ANN\_MLP::SIGMOID\_SYM.Parameters:type - The type of activation function. See ANN\_MLP::ActivationFunctions.param1 - The first parameter of the activation function, \(\alpha\). Default value is 0.param2 - The second parameter of the activation function, \(\beta\). Default value is 0.

#### setAnnealCoolingRatio public void setAnnealCoolingRatio(double val) getAnnealCoolingRatio SEE: getAnnealCoolingRatioParameters:val - automatically generated

#### setAnnealFinalT public void setAnnealFinalT(double val) getAnnealFinalT SEE: getAnnealFinalTParameters:val - automatically generated

#### setAnnealInitialT public void setAnnealInitialT(double val) getAnnealInitialT SEE: getAnnealInitialTParameters:val - automatically generated

#### setAnnealItePerStep public void setAnnealItePerStep(int val) getAnnealItePerStep SEE: getAnnealItePerStepParameters:val - automatically generated

#### setBackpropMomentumScale public void setBackpropMomentumScale(double val) getBackpropMomentumScale SEE: getBackpropMomentumScaleParameters:val - automatically generated

#### setBackpropWeightScale public void setBackpropWeightScale(double val) getBackpropWeightScale SEE: getBackpropWeightScaleParameters:val - automatically generated

#### setLayerSizes public void setLayerSizes([Mat](http://docs.google.com/org/opencv/core/Mat.html) \_layer\_sizes) Integer vector specifying the number of neurons in each layer including the input and output layers. The very first element specifies the number of elements in the input layer. The last element - number of elements in the output layer. Default value is empty Mat. SEE: getLayerSizesParameters:\_layer\_sizes - automatically generated

#### setRpropDW0 public void setRpropDW0(double val) getRpropDW0 SEE: getRpropDW0Parameters:val - automatically generated

#### setRpropDWMax public void setRpropDWMax(double val) getRpropDWMax SEE: getRpropDWMaxParameters:val - automatically generated

#### setRpropDWMin public void setRpropDWMin(double val) getRpropDWMin SEE: getRpropDWMinParameters:val - automatically generated

#### setRpropDWMinus public void setRpropDWMinus(double val) getRpropDWMinus SEE: getRpropDWMinusParameters:val - automatically generated

#### setRpropDWPlus public void setRpropDWPlus(double val) getRpropDWPlus SEE: getRpropDWPlusParameters:val - automatically generated

#### setTermCriteria public void setTermCriteria([TermCriteria](http://docs.google.com/org/opencv/core/TermCriteria.html) val) getTermCriteria SEE: getTermCriteriaParameters:val - automatically generated

#### setTrainMethod public void setTrainMethod(int method) Sets training method and common parameters.Parameters:method - Default value is ANN\_MLP::RPROP. See ANN\_MLP::TrainingMethods.

#### setTrainMethod public void setTrainMethod(int method, double param1) Sets training method and common parameters.Parameters:method - Default value is ANN\_MLP::RPROP. See ANN\_MLP::TrainingMethods.param1 - passed to setRpropDW0 for ANN\_MLP::RPROP and to setBackpropWeightScale for ANN\_MLP::BACKPROP and to initialT for ANN\_MLP::ANNEAL.

#### setTrainMethod public void setTrainMethod(int method, double param1, double param2) Sets training method and common parameters.Parameters:method - Default value is ANN\_MLP::RPROP. See ANN\_MLP::TrainingMethods.param1 - passed to setRpropDW0 for ANN\_MLP::RPROP and to setBackpropWeightScale for ANN\_MLP::BACKPROP and to initialT for ANN\_MLP::ANNEAL.param2 - passed to setRpropDWMin for ANN\_MLP::RPROP and to setBackpropMomentumScale for ANN\_MLP::BACKPROP and to finalT for ANN\_MLP::ANNEAL.

* [Overview](http://docs.google.com/overview-summary.html)
* [Package](http://docs.google.com/package-summary.html)
* Class
* [Tree](http://docs.google.com/package-tree.html)
* [Index](http://docs.google.com/index-all.html)
* [Help](http://docs.google.com/help-doc.html)
* Prev Class
* [Next Class](http://docs.google.com/org/opencv/ml/ANN_MLP_ANNEAL.html)
* [Frames](http://docs.google.com/index.html?org/opencv/ml/ANN_MLP.html)
* [No Frames](http://docs.google.com/ANN_MLP.html)
* [All Classes](http://docs.google.com/allclasses-noframe.html)
* Summary:
* Nested |
* [Field](#3znysh7) |
* Constr |
* [Method](#tyjcwt)
* Detail:
* [Field](#2s8eyo1) |
* Constr |
* [Method](#4i7ojhp)

Generated on 2021-04-02 03:15:03 / OpenCV 3.4.14