

MLP Modellierung

Gruppe 3

26. Mai 2008

Inhaltsverzeichnis

1	Core	3
1.1	Attribute	3
1.1.1	trainingErrors	3
1.1.2	testErrors	3
1.2	Funktionen	3
1.2.1	classify	3
1.2.2	openIdxPicture	3
1.2.3	openAiffSound	3
1.2.4	newMLP	3
1.2.5	loadMLP	3
1.2.6	saveMLP	4
1.2.7	trainOnline	4
1.2.8	trainBatch	4
1.2.9	preprocess	4
1.2.10	getAllActivationFunctions	4
2	Interface NeuralNetwork	4
2.1	Funktionen	4
2.1.1	getNumLayers	4
2.1.2	getLayerSize	4
2.1.3	getWeights	5
2.1.4	getActivationFunction	5
2.1.5	getBias	5
3	Klasse: MLP	5
3.1	Attribute	5
3.2	Funktionen	5
3.2.1	MLP	5
3.2.2	runOnline	6
3.2.3	runBatch	6
3.2.4	runTest	6
4	Klasse: Layer	6
4.1	Attribute	6
4.2	Funktionen	7
4.2.1	Layer	7
4.2.2	Layer	7
4.2.3	getOutput	7
4.2.4	backPropagate	7
4.2.5	update	7
4.2.6	getError	7
4.2.7	getActivationFunction	8

5	Interface ActivationFunction	8
5.1	Funktionen	8
5.1.1	compute	8
5.1.2	derivation	8
5.1.3	getMinimumValue	8
6	DVV	8
6.1	Attribute	8
6.2	Funktionen	8
6.2.1	DVV	8
6.2.2	DVV	8
6.2.3	getDataSet	9
6.2.4	getTrainingData	9
6.2.5	getTestData	9
6.2.6	selectTrainingData	9
7	Data (abstract)	9
7.1	Attribute	9
7.2	Funktionen	9
7.2.1	getData	9
7.2.2	getName	9
7.2.3	getLabel	9
7.2.4	getTarget	9
7.2.5	getFilename	9
7.2.6	getLabelFromTarget	10
7.2.7	subsample	10
7.2.8	setTraining	10
7.2.9	setTest	10
7.2.10	setNotUsed	10
7.2.11	isTraining	10
7.2.12	isTest	10
7.2.13	isNotUsed	10
8	IdxPicture	10
8.1	Attribute	10
8.2	Funktionen	11
8.2.1	IdxPicture	11
8.2.2	readFromFile	11
8.2.3	previewRawData	11
8.2.4	previewSubsampledData	11
8.2.5	subsample	11

9	AiffSound	12
9.1	Attribute	12
9.2	Funktionen	12
9.2.1	AiffSound	12
9.2.2	readFromFile	12
9.2.3	subsample	12
10	GUI	12
10.1	Funktionen	12
10.1.1	setTrainingError	12
10.1.2	setTestError	13
10.1.3	setDataSet	13

1 Core

1.1 Attribute

1.1.1 trainingErrors

Trainingsfehler von den Epochen für die GUI.

1.1.2 testErrors

Testfehler von den Epochen für die GUI.

1.2 Funktionen

1.2.1 classify

Klassifiziert Daten mit dem MLP.

data Zu klassifizierendes Datum.

1.2.2 openIdxPicture

Öffnet Bilddaten.

DataFilename Dateiname der Rohdatendatei.

LabelFilename Dateiname der Targetwert-Datei.

1.2.3 openAiffSound

Öffnet Audiodaten.

Filenames Dateinamen der Rohdatendateien.

1.2.4 newMLP

Erzeugt ein neues MLP.

layer Anzahl und Größe der einzelnen Layer.

activationFunctions Aktivierungsfunktionen für die einzelnen Layer.

autoEncoder Training als Autoencoder?

1.2.5 loadMLP

Lädt ein vorhandenes MLP.

Filename Dateiname des MLP-Saves.

1.2.6 saveMLP

Speichert das aktuell geladene MLP.

Filename Dateiname des MLP-Saves.

1.2.7 trainOnline

Trainiert ein MLP mit der Online-Lernmethode und einer gegebenen Lernrate.

Eta Lernrate.

1.2.8 trainBatch

Trainiert ein MLP mit der Batch-Lernmethode und einer gegebenen Lernrate.

Eta Lernrate

1.2.9 preprocess

Bereitet geöffnete Daten anhand der Konfiguration auf.

resolution Anzahl an Bildpunkten im aufbereiteten Bild.

overlap Gewünschte Überlappung beim Aufbereiten.

ActivationFunction Aktivierungsfunktion für die Skalierung.

1.2.10 getAllActivationFunctions

Liefert Pointer auf alle existierenden Aktivierungsfunktionen (für die Anzeige durch die GUI).

ActivationFunctions Liste von Aktivierungsfunktionen.

2 Interface NeuralNetwork

Dieses Interface wird vom MLP implementiert. Es dient dazu, das MLP von der GUI zu isolieren.

2.1 Funktionen

2.1.1 getNumLayers

Liefert die Anzahl der Schichten eines NeuronalesNetzes.

2.1.2 getLayerSize

Liefert die Größe einer bestimmten Schicht.

layer Nummer der Schicht, deren Größe zurückgegeben werden soll.

2.1.3 getWeights

Liefert die Gewichtsmatrix zwischen zwei Schichten.

layer Nummer der hinteren Schicht der zurückgegebenen Gewichtsmatrix.

2.1.4 getActivationFunction

Liefert die Aktivierungsfunktion einer Schicht.

layer Nummer der Schicht, deren Aktivierungsfunktion zurückgegeben werden soll.

2.1.5 getBias

Liefert den Bias einer Schicht.

layer Nummer der Schicht, deren Aktivierungsfunktion zurückgegeben werden soll.

3 Klasse: MLP

3.1 Attribute

layer Ein Array, das alle Schichten des MLP's speichert.

3.2 Funktionen

3.2.1 MLP

Konstruktor; Erstellt das neuronale Netz und setzt die benötigten Variablen.

inputNeurons Die Anzahl der Eingabeneuronen.

outputNeurons Die Anzahl der Ausgabeneuronen

hiddenLayers Die Array, dessen Größe gleich der Anzahl der versteckten Schichten ist. Als jeweiligen Wert ist die Anzahl der Neuronen der betreffenden Schicht anzugeben.

functions Dieses Array muss die entsprechenden Aktivierungsfunktionen der versteckten Schichten und der Eingabe- sowie Ausgabeschicht enthalten und braucht deshalb die Größe des Array hiddenLayers + 2.

bias Ein Array dessen Wert den Bias jeder versteckten Schicht beschreibt. Braucht deshalb die Größe des Array hiddenLayers

autoencoder Gibt an ob das Netz als Autoencoder initialisiert werden soll.

3.2.2 runOnline

Diese Funktion führt eine Berechnung mit dem neuronalen Netz im Online-Lernmodus durch

in Eingabedaten

eta Die Lernrate

3.2.3 runBatch

Diese Funktion führt eine Berechnung mit dem neuronalen Netz im Batch-Lernmodus durch.

in Eingabedaten

eta Die Lernrate

3.2.4 runTest

Mit dieser Methode kann ein Testlauf gestartet werden.

in Eingabedaten

4 Klasse: Layer

4.1 Attribute

prevLayer Die vorhergehende Schicht.

weightMatrix Die Gewichtsmatrix zwischen dieser Schicht und der vorhergehenden.

gradientMatrix Mit Hilfe dieser Matrix sollen die Gradienten gespeichert und beim "Update"-Schritt verwendet werden.

function Die Aktivierungsfunktion der Schicht.

output Speichert die Ausgabe der Schicht

input Speichert die Eingabe der vorhergehenden Schicht.

layerInput Speichert die Eingabe dieser Schicht.

bias Der Bias dieser Schicht.

4.2 Funktionen

4.2.1 Layer

Konstruktor; wird mit der vorhergehenden Schicht initialisiert, sowie der Aktivierungsfunktion und des Bias

prevLayer Die vorhergehende Schicht; Null wenn diese Schicht die Eingabeschicht des Netzes ist.

neurons Die Neuronen dieser Schicht.

function Die Aktivierungsfunktion dieser Schicht

bias Der Bias. Ein Wert der zur Eingabe jedes Neurons eingerechnet wird.

4.2.2 Layer

Diese Funktion wird bei der Eingabeschicht verwendet und setzt die Eingabedaten.

input Ein Eingabevektor, der die gleiche Dimension wie die Neuronen braucht.

4.2.3 getOutput

Diese Funktion verwendet den Verweis auf die letzte Schicht um den Ausgabevektor dieser Schicht zu erzeugen. Es wird also Rekursiv bis zur Eingabeschicht zurückgegriffen und mit den entsprechenden Gewichten multipliziert und ein Ausgabevektor erzeugt.

4.2.4 backPropagate

Diese Funktion wird bei der Eingabeschicht verwendet und setzt die Eingabedaten.

error Der Fehler der vorherigen Schicht auf diese Schicht bezogen. Wenn diese Schicht die Augabeschicht ist, wird der Ausgabefehler gebraucht. Nach berechnung der nötigen Gewichtsveränderungen und Speicherung dieser, wird der Fehler der vorherigen Schicht errechnet und dieser dann an die nächste Schicht weitergegeben.

4.2.5 update

Passt die Gewichte des Netztes an rekursiv an.

iterations Die Anzahl der Durchläufe des Netztes seit dem letzten Update.

eta Die Lernrate, die verwendet werden soll.

4.2.6 getError

Berechnet den Fehlerwert proportional zum Sollwert und berechnet den Gesamtfehler.

target Der Sollwertvektor

4.2.7 **getActivationFunction**

Gibt die Aktivierungsfunktion zurück.

5 **Interface ActivationFunction**

5.1 **Funktionen**

5.1.1 **compute**

Berechnet den Funktionswert. $f(x) = ?$

layer Eingabewert der Funktion.

5.1.2 **derivation**

Berechnet den Funktionswert der Ableitung. $f'(x) = ?$

layer Eingabewert der Funktion.

5.1.3 **getMinimumValue**

Liefert den Minimalwert der Funktion - sofern nicht festlegbar den Wert `Double.MIN_VALUE`

6 **DVV**

6.1 **Attribute**

allData Der gesamte Datensatz.

testData Der Testdatensatz.

trainingData Der Trainingsdatensatz.

6.2 **Funktionen**

6.2.1 **DVV**

Erstellt die DVV für die Verarbeitung von Audiodaten.

filenames Die Dateinamen der AIFF-Files.

6.2.2 **DVV**

Erstellt die DVV für die Verarbeitung von Videodaten.

dataFilename Die Datei, welche die Rohdaten enthält.

labelFilename Die Datei, welche die Target-Werte enthält.

6.2.3 `getDataSet`

Liefert die geladenen Rohdaten, damit die GUI sie anzeigen kann.

6.2.4 `getTrainingData`

Liefert den Trainings-Datensatz.

6.2.5 `getTestData`

Liefert den Test-Datensatz.

6.2.6 `selectTrainingData`

Sortiert den Datensatz nach Test- und Trainingsdaten.

7 Data (abstract)

7.1 Attribute

use Gibt an, ob das Datum zum Test- oder Trainingsdatensatz gehört oder nicht verwendet wird.

7.2 Funktionen

7.2.1 `getData`

Liefert die fertig verarbeiteten und skalierten Daten.

7.2.2 `getName`

Liefert den von der GUI anzuzeigenden Namen des Datums.

7.2.3 `getLabel`

Liefert die String-Repräsentation des Zielwerts.

7.2.4 `getTarget`

Liefert die Integer-Repräsentation des Zielwerts.

7.2.5 `getFilename`

Liefert den Dateinamen, aus dem das Datum gelesen wurde.

7.2.6 getLabelFromTarget

Bildet ein Label auf den zugehörigen Integer-Wert ab.

target Das zu mappende Target.

7.2.7 subsample

Startet das Subsampling und skaliert die Daten.

resolution Die Zielauflösung.

overlap Die Überlapp des Fensters.

activationFunction Die Funktion, die zur Skalierung der Daten verwendet wird.

7.2.8 setTraining

Markiert das Datum als zum Trainingsdatensatz gehörend.

7.2.9 setTest

Markiert das Datum als zum Testdatensatz gehörend.

7.2.10 setNotUsed

Markiert das Datum als nicht benutzt.

7.2.11 isTraining

Gibt an, ob das Datum zum Trainingsdatensatz gehört.

7.2.12 isTest

Gibt an, ob das Datum zum Testdatensatz gehört.

7.2.13 isNotUsed

Gibt an, ob das Datum nicht benutzt wird.

8 IdxPicture

8.1 Attribute

data Die verarbeiteten Daten.

rawData Die Rohdaten.

label Der Zielwert in String-Form.

target Der Zielwert in Integer-Form.

fileIndex Die Nummer des Datums in der Quelldatei.

filename Die Quelldatei des Datums.

8.2 Funktionen

8.2.1 IdxPicture

Erstellt ein Neues IdxPicture-Objekt.

rawData Die Rohdaten.

label Der Zielwert.

filename Der Name der Quelldatei.

fileIndex Die Nummer des Datums in der Quelldatei.

8.2.2 readFromFile

Liest eine Quelldatei und erstellt daraus eine Collection von IdxPicture-Objekten.

dataFilename Der Name der Rohdaten-Datei.

labelFilename Der Name der Zielwert-Datei.

8.2.3 previewRawData

Liefert die Rohdaten, um sie dem Benutzer anzuzeigen.

8.2.4 previewSubsampledData

Liefert die vorverarbeiteten Daten, um sie dem Benutzer anzuzeigen.

resolution Die Zielauflösung.

overlap Der Überlapp.

8.2.5 subsample

Startet das Subsampling.

resolution Die Zielauflösung.

overlap Der Überlapp.

activationFunction Die für die Skalierung verwendete Aktivierungsfunktion.

9 AiffSound

9.1 Attribute

data Die verarbeiteten Daten.

rawData Die Rohdaten.

label Der Zielwert in String-Form.

target Der Zielwert in Integer-Form.

filename Die Quelldatei des Datums.

9.2 Funktionen

9.2.1 AiffSound

Erstellt ein neues AiffSound-Objekt.

rawData Die Rohdaten.

label Der Zielwert.

filename Der Name der Quelldatei.

9.2.2 readFromFile

Liest eine Quelldatei und erstellt daraus eine Collection von AiffSound-Objekten.

filename Der Name der Quelldatei.

9.2.3 subsample

Startet das Subsampling.

resolution Die Zielauflösung.

overlap Der Überlapp.

activationFunction Die für die Skalierung verwendete Aktivierungsfunktion.

10 GUI

10.1 Funktionen

10.1.1 setTrainingError

Setzt die Trainingsfehler in der GUI.

errorData Die Liste der Trainingsfehler.

10.1.2 setTestError

Setzt die Testfehler in der GUI.

errorData Die Liste der Testfehler.

10.1.3 setDataSet

Setzt die Liste der geladenen Daten in der GUI.

dataSet Die Liste der geladenen Daten.