encog / **encog-java-core**

http://www.heatonresearch.com/encog

3,626 commits | **3** branches | **3** releases | **31** contributors

Branch: **master** ▾ | **New pull request** | **Create new file** | **Upload files** | **Find file** | **Clone or download** ▾

jeffheaton Fixed typo from last commit — Latest commit `e82ec66` on 4 Sep 2017

classes	Fix SGD and add more update rules, adam, adagrad, rmsprop, etc.	2 years ago
dependencies	forget aparapi while I think about something better.	4 years ago
gradle/wrapper	Added checks for rewrite algebra to make sure required operators are …	2 years ago
src	Fixed typo from last commit	7 months ago
.gitattributes	Add java to attributes	4 years ago
.gitignore	Updates to gitignore	2 years ago
.travis.yml	more work on travis	7 months ago
LICENSE.txt	Normalize line endings	4 years ago
NOTICE.txt	Normalize line endings	4 years ago
README.md	update readme	9 months ago
build.gradle	Rolling version to 3.4.1	7 months ago
gradle.properties	Rolling version to 3.4.1	7 months ago
gradlew	Update gradle	3 years ago
gradlew.bat	Normalize line endings	4 years ago

README.md

build passing

Encog Machine Learning Framework

Encog is an advanced machine learning framework that supports a variety of advanced algorithms, as well as support classes to normalize and process data. Machine learning algorithms such as Support Vector Machines, Artificial Neural Networks, Bayesian Networks, Hidden Markov Models, Genetic Programming and Genetic Algorithms are supported. Most Encog training algorithms are multi-threaded and scale well to multicore hardware. A GUI based workbench is also provided to help model and train machine learning algorithms. Encog has been in active development since 2008.

For more information: Encog Website

Simple Java XOR Example in Encog

```
import org.encog.Encog;
import org.encog.engine.network.activation.ActivationReLU;
import org.encog.engine.network.activation.ActivationSigmoid;
import org.encog.ml.data.MLData;
import org.encog.ml.data.MLDataPair;
import org.encog.ml.data.MLDataSet;
import org.encog.ml.data.basic.BasicMLDataSet;
import org.encog.neural.networks.BasicNetwork;
import org.encog.neural.networks.layers.BasicLayer;
import org.encog.neural.networks.training.propagation.resilient.ResilientPropagation;

public class XORHelloWorld {

        /**
         * The input necessary for XOR.
         */
        public static double XOR_INPUT[][] = { { 0.0, 0.0 }, { 1.0, 0.0 },
                        { 0.0, 1.0 }, { 1.0, 1.0 } };

        /**
         * The ideal data necessary for XOR.
         */
        public static double XOR_IDEAL[][] = { { 0.0 }, { 1.0 }, { 1.0 }, { 0.0 } };

        /**
         * The main method.
         * @param args No arguments are used.
         */
        public static void main(final String args[]) {

                // create a neural network, without using a factory
                BasicNetwork network = new BasicNetwork();
                network.addLayer(new BasicLayer(null,true,2));
                network.addLayer(new BasicLayer(new ActivationReLU(),true,5));
                network.addLayer(new BasicLayer(new ActivationSigmoid(),false,1));
                network.getStructure().finalizeStructure();
                network.reset();

                // create training data
                MLDataSet trainingSet = new BasicMLDataSet(XOR_INPUT, XOR_IDEAL);

                // train the neural network
                final ResilientPropagation train = new ResilientPropagation(network, trainingSet);

                int epoch = 1;

                do {
                        train.iteration();
                        System.out.println("Epoch #" + epoch + " Error:" + train.getError());
                        epoch++;
                } while(train.getError() > 0.01);
                train.finishTraining();

                // test the neural network
                System.out.println("Neural Network Results:");
                for(MLDataPair pair: trainingSet ) {
                        final MLData output = network.compute(pair.getInput());
                        System.out.println(pair.getInput().getData(0) + "," + pair.getInput().getData(1)
                                        + ", actual=" + output.getData(0) + ",ideal=" + pair.getIdeal().get
                }

                Encog.getInstance().shutdown();
        }
}
```