Popular Methods

- Builder ( )
- fromJson ( )
- clone ( )
- getInputPreProcess ( )
- toJson ( )
- getConfs ( )
- toYaml ( )
- fromYaml ( )

Related Classes

- java.io.File
- java.util.Collections
- java.util.Iterator
- java.util.Random
- java.lang.reflect.Field
- java.nio.file.Files
- org.apache.spark.api.java.JavaSparkContext
- org.apache.spark.api.java.JavaRDD
- org.nd4j.linalg.api.ndarray.INDArray
- org.nd4j.linalg.factory.Nd4j
- org.deeplearning4j.nn.weights.WeightInit
- org.deeplearning4j.nn.conf.NeuralNetConfiguration
- org.nd4j.linalg.dataset.DataSet
- org.deeplearning4j.nn.multilayer.MultiLayerNetwork
- org.deeplearning4j.nn.api.OptimizationAlgorithm
- org.nd4j.linalg.lossfunctions.LossFunctions
- org.nd4j.linalg.dataset.api.iterator.DataSetIterator
- org.deeplearning4j.nn.conf.Updater

- org.deeplearning4j.eval.Evaluation
- org.deeplearning4j.nn.conf.layers.OutputLayer
- org.deeplearning4j.optimize.listeners.ScoreIterationListener
- org.deeplearning4j.nn.conf.layers.DenseLayer
- org.nd4j.linalg.activations.Activation
- org.deeplearning4j.util.ModelSerializer
- org.deeplearning4j.nn.conf.layers.ConvolutionLayer

Java Code Examples for org.deeplearning4j.nn.conf.MultiLayerConfiguration

The following are top voted examples for showing how to use org.deeplearning4j.nn.conf.MultiLayerConfiguration. These examples are extracted from open source projects. You can vote up the examples you like and your votes will be used in our system to generate more good examples.

+ Save this class

Example 1

Project: *Machine-Learning-End-to-Endguide-for-Java-developers* File: *NeuralNetworks.java* *View source code* 6 votes

```
private static MultiLayerNetwork softMaxRegression(int seed,
                int iterations, int numRows, int numColumns, int outputNum) {
        MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
                        .seed(seed)
                        .gradientNormalization(
                                        GradientNormalization.ClipElementWiseAbsoluteValue)
                        .gradientNormalizationThreshold(1.0)
                        .iterations(iterations)
                        .momentum(0.5)
                        .momentumAfter(Collections.singletonMap(3, 0.9))
                        .optimizationAlgo(OptimizationAlgorithm.CONJUGATE_GRADIENT)
                        .list(1)
                        .layer(0,
                                        new OutputLayer.Builder(
                                                        LossFunction.NEGATIVELOGLIKELIHOOD)
                                                        .activation("softmax")
                                                        .nIn(numColumns * numRows).nOut(outputNum)
                                                        .build()).pretrain(true).backprop(false)
                        .build();

        MultiLayerNetwork model = new MultiLayerNetwork(conf);

        return model;
}
```

Example 2

Project: *NeuralNetworksLite* File: *RegressionMathFunctions.java* *View source code*

6 votes

```
/** Returns the network configuration, 2 hidden DenseLayers of size 50.
 */
private static MultiLayerConfiguration getDeepDenseLayerNetworkConfiguration() {
    final int numHiddenNodes = 50;
    return new NeuralNetConfiguration.Builder()
            .seed(seed)
            .iterations(iterations)
            .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
            .learningRate(learningRate)
            .weightInit(WeightInit.XAVIER)
            .updater(Updater.NESTEROVS).momentum(0.9)
            .list()
            .layer(0, new DenseLayer.Builder().nIn(numInputs).nOut(numHiddenNodes)
                    .activation(Activation.TANH).build())
            .layer(1, new DenseLayer.Builder().nIn(numHiddenNodes).nOut(numHiddenNodes)
                    .activation(Activation.TANH).build())
            .layer(2, new OutputLayer.Builder(LossFunctions.LossFunction.MSE)
                    .activation(Activation.IDENTITY)
                    .nIn(numHiddenNodes).nOut(numOutputs).build())
            .pretrain(false).backprop(true).build();
}
```

Example 3

Project: *DL4J* File: *DeepBeliefNetworkModel.java* *View source code*

6 votes

```
protected MultiLayerConfiguration getConfiguration()
   {
int hiddenLayerNodes = parameters.getHiddeLayerNodes()[0];
final RBM hiddenLayer = new RBM.Builder(RBM.HiddenUnit.RECTIFIED, RBM.VisibleUnit.GAUSSIAN)
        .nIn(parameters.getInputSize()).nOut(hiddenLayerNodes).weightInit(WeightInit.XAVIER).k(1)
        .activation("relu").lossFunction(LossFunctions.LossFunction.RMSE_XENT).updater(Updater.ADAGRAD)
        .dropOut(0.5).build();

final OutputLayer outputLayer = new OutputLayer.Builder(LossFunctions.LossFunction.MCXENT).nIn(hiddenLayerNodes)
        .nOut(parameters.getOutputSize()).activation("softmax").build();

return new NeuralNetConfiguration.Builder().seed(parameters.getSeed()).iterations(parameters.getIterations())
        .learningRate(parameters.getLearningRate()).optimizationAlgo(OptimizationAlgorithm.CONJUGATE_GRADIENT)
        .l2(2e-4).regularization(true).momentum(0.9).useDropConnect(true).list(2).layer(0, hiddenLayer)
        .layer(1, outputLayer).build();
   }
```

Example 4

Project: *DL4J* File: *AnomalyDetectionModel.java* *View source code*

6 votes

```
@Override
   protected MultiLayerConfiguration getConfiguration()
   {
return new NeuralNetConfiguration.Builder().seed(parameters.getSeed()).iterations(parameters.getIterations())
        .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT).learningRate(parameters.getLearningRate()).l2(0.001)
        .list(4)
        .layer(0,
                new DenseLayer.Builder().nIn(parameters.getInputSize()).nOut(250).weightInit(WeightInit.XAVIER)
                        .updater(Updater.ADAGRAD).activation("relu").build())
        .layer(1,
                new DenseLayer.Builder().nIn(250).nOut(10).weightInit(WeightInit.XAVIER)
                        .updater(Updater.ADAGRAD).activation("relu").build())
        .layer(2,
                new DenseLayer.Builder().nIn(10).nOut(250).weightInit(WeightInit.XAVIER)
                        .updater(Updater.ADAGRAD).activation("relu").build())
        .layer(3,
                new OutputLayer.Builder().nIn(250).nOut(parameters.getInputSize()).weightInit(WeightInit.XAVIER)
                        .updater(Updater.ADAGRAD).activation("relu")
                        .lossFunction(LossFunctions.LossFunction.MSE).build())
                .pretrain(false).backprop(true).build();
    }
```

Example 5

Project: *DL4J* File: *StackedAutoEncoderModel.java* *View source code* 6 votes 👍 👎

```
@Override
    protected MultiLayerConfiguration getConfiguration()
    {
return new NeuralNetConfiguration.Builder().seed(parameters.getSeed())
        .gradientNormalization(GradientNormalization.ClipElementWiseAbsoluteValue)
        .gradientNormalizationThreshold(1.0).iterations(parameters.getIterations()).momentum(0.5)
        .momentumAfter(Collections.singletonMap(3, 0.9))
        .optimizationAlgo(OptimizationAlgorithm.CONJUGATE_GRADIENT).list(4)
        .layer(0,
                new AutoEncoder.Builder().nIn(parameters.getInputSize()).nOut(500).weightInit(WeightInit.XAVIER)
                        .lossFunction(LossFunction.RMSE_XENT).corruptionLevel(0.3).build())
        .layer(1, new AutoEncoder.Builder().nIn(500).nOut(250).weightInit(WeightInit.XAVIER)
                .lossFunction(LossFunction.RMSE_XENT).corruptionLevel(0.3)

                .build())
        .layer(2,
                new AutoEncoder.Builder().nIn(250).nOut(200).weightInit(WeightInit.XAVIER)
                        .lossFunction(LossFunction.RMSE_XENT).corruptionLevel(0.3).build())
        .layer(3, new OutputLayer.Builder(LossFunction.NEGATIVELOGLIKELIHOOD).activation("softmax").nIn(200)
                .nOut(parameters.getOutputSize()).build())
        .pretrain(true).backprop(false).build();
    }
```

Example 6

Project: *DL4J* File: *ConvolutionalNetModel.java* *View source code* 6 votes 👍 👎

```
@Override
    protected MultiLayerConfiguration getConfiguration()
    {
final ConvulationalNetParameters parameters = (ConvulationalNetParameters) this.parameters;
final MultiLayerConfiguration.Builder builder = new NeuralNetConfiguration.Builder().seed(parameters.getSeed())
        .iterations(parameters.getIterations())
        .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT).list(2)
        .layer(0,
                new ConvolutionLayer.Builder(new int[] { 1, 1 }).nIn(parameters.getInputSize()).nOut(1000)
                        .activation("relu").weightInit(WeightInit.RELU).build())
        .layer(1,
                new OutputLayer.Builder(LossFunctions.LossFunction.MCXENT).nOut(parameters.getOutputSize())
                        .weightInit(WeightInit.XAVIER).activation("softmax").build())
        .backprop(true).pretrain(false);

new ConvolutionLayerSetup(builder, parameters.getRows(), parameters.getColumns(), parameters.getChannels());

return builder.build();
    }
```

Example 7

Project: *DL4J* File: *ConvolutionalNetModel.java* *View source code* 6 votes 👍 👎

```
@Override
   protected MultiLayerConfiguration getConfiguration()
   {
final ConvulationalNetParameters parameters = (ConvulationalNetParameters) this.parameters;
final MultiLayerConfiguration.Builder builder = new NeuralNetConfiguration.Builder().seed(parameters.getSeed())
        .iterations(parameters.getIterations())
        .gradientNormalization(GradientNormalization.RenormalizeL2PerLayer)
        .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT).list(3)
        .layer(0,
                new ConvolutionLayer.Builder(10, 10).stride(2, 2).nIn(parameters.getChannels()).nOut(6)
                        .weightInit(WeightInit.XAVIER).activation("relu").build())
        .layer(1, new SubsamplingLayer.Builder(SubsamplingLayer.PoolingType.MAX, new int[] { 2, 2 }).build())
        .layer(2, new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
                .nOut(parameters.getOutputSize()).weightInit(WeightInit.XAVIER).activation("softmax").build())
        .backprop(true).pretrain(false);

new ConvolutionLayerSetup(builder, parameters.getRows(), parameters.getColumns(), parameters.getChannels());

return builder.build();
   }
```

Example 8

Project: *dl4j-trainer-archetype* File: *Train.java* View source code 6 votes 👍 👎

```
private static MultiLayerConfiguration net(int nIn, int nOut) {
    return new NeuralNetConfiguration.Builder()
            .seed(42)
            .iterations(1)
            .activation(Activation.RELU)
            .weightInit(WeightInit.XAVIER)
            .learningRate(0.1)
            .regularization(true).l2(1e-4)
            .list(
                    new DenseLayer.Builder().nIn(nIn).nOut(3).build(),
                    new DenseLayer.Builder().nIn(3).nOut(3).build(),
                    new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
                            .activation(Activation.SOFTMAX)
                            .nIn(3)
                            .nOut(nOut)
                            .build()
            )
            .build();
}
```

Example 9

Project: *dl4j-spark-ml-examples* File: *JavaLfwClassification.java* View source code 6 votes 👍 👎

```
private static MultiLayerConfiguration getConfiguration(DataFrame dataset) {

        MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
                .seed(seed)
                .constrainGradientToUnitNorm(true)
                .optimizationAlgo(OptimizationAlgorithm.CONJUGATE_GRADIENT)
                .list(4)
                .layer(0, new RBM.Builder(RBM.HiddenUnit.BINARY, RBM.VisibleUnit.BINARY)
                        .weightInit(WeightInit.XAVIER)
                        .nIn(rows * columns).nOut(600).build())
                .layer(1, new RBM.Builder(RBM.HiddenUnit.BINARY, RBM.VisibleUnit.BINARY)
                        .weightInit(WeightInit.XAVIER)
                        .nIn(600).nOut(250).build())
                .layer(2, new RBM.Builder(RBM.HiddenUnit.BINARY, RBM.VisibleUnit.BINARY)
                        .weightInit(WeightInit.XAVIER)
                        .nIn(250).nOut(200).build())
                .layer(3, new OutputLayer.Builder(LossFunctions.LossFunction.RMSE_XENT)
                        .weightInit(WeightInit.XAVIER)
                        .activation("softmax")
                        .nIn(200).nOut(AUTOMATIC).build())
                .pretrain(true).backprop(false)
                .build();

        return conf;
    }
```

Example 10

Project: *deeplearning4j* File: *LayerConfigTest.java* View source code 6 votes 👍 👎

```
@Test
public void testLayerName() {
    String name1 = "genisys";
    String name2 = "bill";

    MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder().list()
                    .layer(0, new DenseLayer.Builder().nIn(2).nOut(2).name(name1).build())
                    .layer(1, new DenseLayer.Builder().nIn(2).nOut(2).name(name2).build()).build();
    MultiLayerNetwork net = new MultiLayerNetwork(conf);
    net.init();

    assertEquals(name1, conf.getConf(0).getLayer().getLayerName());
    assertEquals(name2, conf.getConf(1).getLayer().getLayerName());

}
```

Example 11

Project: *deeplearning4j* File: *LayerConfigTest.java* View source code 6 votes

```
@Test
public void testUpdaterAdamParamsLayerwiseOverride() {
    MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
            .updater(new Adam(1.0, 0.5, 0.5, 1e-8))
            .list()
            .layer(0, new DenseLayer.Builder().nIn(2).nOut(2).build())
                    .layer(1, new DenseLayer.Builder().nIn(2).nOut(2).updater(new Adam(1.0, 0.6, 0.7, 1e-8)).build())
                    .build();
    MultiLayerNetwork net = new MultiLayerNetwork(conf);
    net.init();

    assertEquals(0.5, ((Adam) ((BaseLayer) conf.getConf(0).getLayer()).getIUpdater()).getBeta1(), 0.0);
    assertEquals(0.6, ((Adam) ((BaseLayer) conf.getConf(1).getLayer()).getIUpdater()).getBeta1(), 0.0);
    assertEquals(0.5, ((Adam) ((BaseLayer) conf.getConf(0).getLayer()).getIUpdater()).getBeta2(), 0.0);
    assertEquals(0.7, ((Adam) ((BaseLayer) conf.getConf(1).getLayer()).getIUpdater()).getBeta2(), 0.0);
}
```

Example 12

Project: *deeplearning4j* File: *ModelGuesserTest.java* View source code 6 votes

```
private MultiLayerNetwork getNetwork() {
    int nIn = 5;
    int nOut = 6;

    MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder().seed(12345).l1(0.01).l2(0.01)
            .updater(new Sgd(0.1)).activation(Activation.TANH).weightInit(WeightInit.XAVIER).list()
            .layer(0, new DenseLayer.Builder().nIn(nIn).nOut(20).build())
            .layer(1, new DenseLayer.Builder().nIn(20).nOut(30).build()).layer(2, new OutputLayer.Builder()
                    .lossFunction(LossFunctions.LossFunction.MSE).nIn(30).nOut(nOut).build())
            .build();

    MultiLayerNetwork net = new MultiLayerNetwork(conf);
    net.init();

    return net;
}
```

Example 13

Project: *deeplearning4j* File: *ConvolutionLayerSetupTest.java* View source code 6 votes

```
@Test
public void testSeparableConv2D() {

    MultiLayerConfiguration.Builder builder = new NeuralNetConfiguration.Builder().list()
            .layer( new SeparableConvolution2D.Builder(2, 2)
                    .depthMultiplier(2)
                    .padding(0, 0)
                    .stride(2, 2).nIn(1).nOut(3).build()) //(28-2+0)/2+1 = 14
            .layer( new SubsamplingLayer.Builder().kernelSize(2, 2).padding(1, 1).stride(2, 2).build()) //(14-2+2)/2+1 = 8 -> 8x8x3
            .layer(2, new OutputLayer.Builder().nOut(3).build())
            .setInputType(InputType.convolutional(28, 28, 1));

    MultiLayerConfiguration conf = builder.build();

    assertNotNull(conf.getInputPreProcess(2));
    assertTrue(conf.getInputPreProcess(2) instanceof CnnToFeedForwardPreProcessor);
    CnnToFeedForwardPreProcessor proc = (CnnToFeedForwardPreProcessor) conf.getInputPreProcess(2);
    assertEquals(8, proc.getInputHeight());
    assertEquals(8, proc.getInputWidth());
    assertEquals(3, proc.getNumChannels());

    assertEquals(8 * 8 * 3, ((FeedForwardLayer) conf.getConf(2).getLayer()).getNIn());
}
```

Example 14

Project: *deeplearning4j* File: *DenseTest.java* *View source code* 6 votes 👍 👎

```
private static MultiLayerNetwork getDenseMLNConfig(boolean backprop, boolean pretrain) {
    int numInputs = 4;
    int outputNum = 3;
    long seed = 6;

    MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder().seed(seed)
                    .updater(new Sgd(1e-3)).l1(0.3).l2(1e-3).list()
                    .layer(0, new org.deeplearning4j.nn.conf.layers.DenseLayer.Builder().nIn(numInputs).nOut(3)
                                    .activation(Activation.TANH).weightInit(WeightInit.XAVIER).build())
                    .layer(1, new org.deeplearning4j.nn.conf.layers.DenseLayer.Builder().nIn(3).nOut(2)
                                    .activation(Activation.TANH).weightInit(WeightInit.XAVIER).build())
                    .layer(2, new OutputLayer.Builder(LossFunctions.LossFunction.MCXENT)
                                    .weightInit(WeightInit.XAVIER).nIn(2).nOut(outputNum).build())
                    .backprop(backprop).pretrain(pretrain).build();

    MultiLayerNetwork model = new MultiLayerNetwork(conf);
    model.init();
    return model;

}
```

Example 15

Project: *deeplearning4j* File: *ConvolutionLayerSetupTest.java* *View source code* 6 votes 👍 👎

```
public MultiLayerConfiguration.Builder incompleteLRN() {
    MultiLayerConfiguration.Builder builder =
                    new NeuralNetConfiguration.Builder().seed(3)
                                    .optimizationAlgo(OptimizationAlgorithm.CONJUGATE_GRADIENT).list()
                                    .layer(0, new org.deeplearning4j.nn.conf.layers.ConvolutionLayer.Builder(
                                                    new int[] {5, 5}).nOut(6).build())
                                    .layer(1, new org.deeplearning4j.nn.conf.layers.SubsamplingLayer.Builder(
                                                    new int[] {2, 2}).build())
                                    .layer(2, new LocalResponseNormalization.Builder().build())
                                    .layer(3, new org.deeplearning4j.nn.conf.layers.ConvolutionLayer.Builder(
                                                    new int[] {5, 5}).nOut(6).build())
                                    .layer(4, new org.deeplearning4j.nn.conf.layers.SubsamplingLayer.Builder(
                                                    new int[] {2, 2}).build())
                                    .layer(5, new org.deeplearning4j.nn.conf.layers.OutputLayer.Builder(
                                                    LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD).nOut(2)
                                                                    .build());
    return builder;
}
```

Example 16

Project: *deeplearning4j* File: *ConvolutionLayerSetupTest.java* *View source code* 6 votes 👍 👎

```
@Test
public void testUpsampling() {

    MultiLayerConfiguration.Builder builder = new NeuralNetConfiguration.Builder().list()
            .layer(new ConvolutionLayer.Builder(2, 2).padding(0, 0).stride(2, 2).nIn(1).nOut(3).build()) //(28-2+0)/2+1 = 14
            .layer(new Upsampling2D.Builder().size(3).build()) // 14 * 3 = 42!
            .layer(new OutputLayer.Builder().nOut(3).build())
            .setInputType(InputType.convolutional(28, 28, 1));

    MultiLayerConfiguration conf = builder.build();

    assertNotNull(conf.getInputPreProcess(2));
    assertTrue(conf.getInputPreProcess(2) instanceof CnnToFeedForwardPreProcessor);
    CnnToFeedForwardPreProcessor proc = (CnnToFeedForwardPreProcessor) conf.getInputPreProcess(2);
    assertEquals(42, proc.getInputHeight());
    assertEquals(42, proc.getInputWidth());
    assertEquals(3, proc.getNumChannels());

    assertEquals(42 * 42 * 3, ((FeedForwardLayer) conf.getConf(2).getLayer()).getNIn());
}
```

Example 17

Project: *deeplearning4j* File: *BatchNormalizationTest.java* *View source code* 6 votes 👍 👎

```
@Test
```

```
public void testCNNBNActivationCombo() throws Exception {
    DataSetIterator iter = new MnistDataSetIterator(2, 2);
    DataSet next = iter.next();

    MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
                    .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT).seed(123)
                    .list()
                    .layer(0, new ConvolutionLayer.Builder().nIn(1).nOut(6).weightInit(WeightInit.XAVIER)
                                    .activation(Activation.IDENTITY).build())
                    .layer(1, new BatchNormalization.Builder().build())
                    .layer(2, new ActivationLayer.Builder().activation(Activation.RELU).build())
                    .layer(3, new OutputLayer.Builder(LossFunctions.LossFunction.MCXENT)
                                    .weightInit(WeightInit.XAVIER).activation(Activation.SOFTMAX).nOut(10).build())
                    .backprop(true).pretrain(false).setInputType(InputType.convolutionalFlat(28, 28, 1)).build();

    MultiLayerNetwork network = new MultiLayerNetwork(conf);
    network.init();
    network.fit(next);

    assertNotEquals(null, network.getLayer(0).getParam("W"));
    assertNotEquals(null, network.getLayer(0).getParam("b"));
}
```

Example 18

Project: *deeplearning4j* File: *TestTransferStatsCollection.java* *View source code* 6 votes 👍 👎

```
@Test
public void test() throws IOException {

    MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder().list()
                    .layer(0, new DenseLayer.Builder().nIn(10).nOut(10).build())
                    .layer(1, new OutputLayer.Builder().nIn(10).nOut(10).build()).build();

    MultiLayerNetwork net = new MultiLayerNetwork(conf);
    net.init();


    MultiLayerNetwork net2 =
                    new TransferLearning.Builder(net)
                                    .fineTuneConfiguration(
                                                    new FineTuneConfiguration.Builder().updater(new Sgd(0.01)).build())
                                    .setFeatureExtractor(0).build();

    File f = Files.createTempFile("dl4jTestTransferStatsCollection", "bin").toFile();
    f.delete();
    net2.setListeners(new StatsListener(new FileStatsStorage(f)));

    //Previosuly: failed on frozen layers
    net2.fit(new DataSet(Nd4j.rand(8, 10), Nd4j.rand(8, 10)));

    f.deleteOnExit();
}
```

Example 19

Project: *deeplearning4j* File: *LocalResponseTest.java* *View source code* 6 votes 👍 👎

```
@Test
public void testMultiCNNLayer() throws Exception {
    MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
                    .optimizationAlgo(OptimizationAlgorithm.LINE_GRADIENT_DESCENT).seed(123).list()
                    .layer(0, new ConvolutionLayer.Builder().nIn(1).nOut(6).weightInit(WeightInit.XAVIER)
                                    .activation(Activation.RELU).build())
                    .layer(1, new LocalResponseNormalization.Builder().build()).layer(2,
                                    new DenseLayer.Builder()
                                                    .nOut(2).build())
                    .layer(3, new OutputLayer.Builder(LossFunctions.LossFunction.MCXENT)
                                    .weightInit(WeightInit.XAVIER).activation(Activation.SOFTMAX).nIn(2).nOut(10)
                                    .build())
                    .backprop(true).pretrain(false).setInputType(InputType.convolutionalFlat(28, 28, 1)).build();

    MultiLayerNetwork network = new MultiLayerNetwork(conf);
    network.init();
    DataSetIterator iter = new MnistDataSetIterator(2, 2);
    DataSet next = iter.next();

    network.fit(next);
}
```

Example 20

Project: *deeplearning4j* File: *ConvolutionLayerSetupTest.java* View source code 6 votes

```
public MultiLayerConfiguration.Builder inComplete() {
    int nChannels = 1;
    int outputNum = 10;
    int seed = 123;

    MultiLayerConfiguration.Builder builder = new NeuralNetConfiguration.Builder().seed(seed)
                    .optimizationAlgo(OptimizationAlgorithm.LINE_GRADIENT_DESCENT).list()
                    .layer(0, new org.deeplearning4j.nn.conf.layers.ConvolutionLayer.Builder(new int[] {10, 10},
                                    new int[] {2, 2}).nIn(nChannels).nOut(6).build())
                    .layer(1, new SubsamplingLayer.Builder(SubsamplingLayer.PoolingType.MAX, new int[] {2, 2})
                                    .build())
                    .layer(2, new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
                                    .nOut(outputNum).weightInit(WeightInit.XAVIER).activation(Activation.SOFTMAX)
                                    .build())
                    .backprop(true).pretrain(false);

    return builder;
}
```

Example 21

Project: *deeplearning4j* File: *ModelSerializerTest.java* View source code 6 votes

```
@Test
public void testWriteMLNModel() throws Exception {
    int nIn = 5;
    int nOut = 6;

    MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder().seed(12345).l1(0.01)
                    .l2(0.01).updater(new Sgd(0.1)).activation(Activation.TANH).weightInit(WeightInit.XAVIER).list()
                    .layer(0, new DenseLayer.Builder().nIn(nIn).nOut(20).build())
                    .layer(1, new DenseLayer.Builder().nIn(20).nOut(30).build()).layer(2, new OutputLayer.Builder()
                                    .lossFunction(LossFunctions.LossFunction.MSE).nIn(30).nOut(nOut).build())
                    .build();

    MultiLayerNetwork net = new MultiLayerNetwork(conf);
    net.init();

    File tempFile = File.createTempFile("tsfs", "fdfsdf");
    tempFile.deleteOnExit();

    ModelSerializer.writeModel(net, tempFile, true);

    MultiLayerNetwork network = ModelSerializer.restoreMultiLayerNetwork(tempFile);

    assertEquals(network.getLayerWiseConfigurations().toJson(), net.getLayerWiseConfigurations().toJson());
    assertEquals(net.params(), network.params());
    assertEquals(net.getUpdater().getStateViewArray(), network.getUpdater().getStateViewArray());
}
```

Example 22

Project: *deeplearning4j* File: *TestSparkMultiLayerParameterAveraging.java* View source code 6 votes

```
@Test
public void testSmallAmountOfData() {
    //Idea: Test spark training where some executors don't get any data
    //in this case: by having fewer examples (2 DataSets) than executors (local[*])

    MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder().updater(new RmsProp())
                    .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT).list()
                    .layer(0, new org.deeplearning4j.nn.conf.layers.DenseLayer.Builder().nIn(nIn).nOut(3)
                                    .activation(Activation.TANH).build())
                    .layer(1, new org.deeplearning4j.nn.conf.layers.OutputLayer.Builder(
                                    LossFunctions.LossFunction.MSE).nIn(3).nOut(nOut).activation(Activation.SOFTMAX)
                                                    .build())
                    .build();

    SparkDl4jMultiLayer sparkNet = new SparkDl4jMultiLayer(sc, conf,
                    new ParameterAveragingTrainingMaster(true, numExecutors(), 1, 10, 1, 0));

    Nd4j.getRandom().setSeed(12345);
    DataSet d1 = new DataSet(Nd4j.rand(1, nIn), Nd4j.rand(1, nOut));
    DataSet d2 = new DataSet(Nd4j.rand(1, nIn), Nd4j.rand(1, nOut));

    JavaRDD<DataSet> rddData = sc.parallelize(Arrays.asList(d1, d2));

    sparkNet.fit(rddData);

}
```

Example 23

Project: *deeplearning4j* File: *TestFrozenLayers.java* View source code 6 votes

```
public static MultiLayerNetwork getOriginalNet(int seed){
    MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
            .seed(seed)
            .weightInit(WeightInit.XAVIER)
            .activation(Activation.TANH)
            .convolutionMode(ConvolutionMode.Same)
            .updater(new Sgd(0.3))
            .list()
            .layer(new ConvolutionLayer.Builder().nOut(3).kernelSize(2,2).stride(1,1).build())
            .layer(new SubsamplingLayer.Builder().kernelSize(2,2).stride(1,1).build())
            .layer(new ConvolutionLayer.Builder().nIn(3).nOut(3).kernelSize(2,2).stride(1,1).build())
            .layer(new DenseLayer.Builder().nOut(64).build())
            .layer(new DenseLayer.Builder().nIn(64).nOut(64).build())
            .layer(new OutputLayer.Builder().nIn(64).nOut(10).lossFunction(LossFunctions.LossFunction.MSE).build())
            .setInputType(InputType.convolutionalFlat(28,28,1))
            .build();


    MultiLayerNetwork net = new MultiLayerNetwork(conf);
    net.init();
    return net;
}
```

Example 24

Project: *greycat* File: *NeuralNetAttribute.java* View source code 5 votes

```
public void reconf() {

        int seed = 123;
        double learningRate = 0.01;
        int numInputs = 2;
        int numOutputs = 2;
        int numHiddenNodes = 5;

        MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
                .seed(seed)
                .iterations(1)
                .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
                .learningRate(learningRate)
                .updater(Updater.NESTEROVS).momentum(0.9)
                .list()
                .layer(0, new DenseLayer.Builder().nIn(numInputs).nOut(numHiddenNodes)
                        .weightInit(WeightInit.XAVIER)
                        .activation("relu")
                        .build())
                .layer(1, new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
                        .weightInit(WeightInit.XAVIER)
                        .activation("softmax").weightInit(WeightInit.XAVIER)
                        .nIn(numHiddenNodes).nOut(numOutputs).build())
                .pretrain(false).backprop(true).build();

        model = new MultiLayerNetwork(conf);

        System.out.println("Ready :-)");

        if (dirty != null) {
            dirty.run();
        }

    }
```

Example 25

Project: *anagnostes* File: *ConfigurationFactory.java* View source code 5 votes

```
public static MultiLayerConfiguration configuration() {
        /*
         * Regarding the .setInputType(InputType.convolutionalFlat(28,28,1)) line: This does a few things. (a) It adds
         * preprocessors, which handle things like the transition between the convolutional/subsampling layers and the dense
         * layer (b) Does some additional configuration validation (c) Where necessary, sets the nIn (number of input
         * neurons, or input depth in the case of CNNs) values for each layer based on the size of the previous layer (but
         * it won't override values manually set by the user) InputTypes can be used with other layer types too (RNNs, MLPs
         * etc) not just CNNs. For normal images (when using ImageRecordReader) use
         * InputType.convolutional(height,width,depth). MNIST record reader is a special case, that outputs 28x28 pixel
         * grayscale (nChannels=1) images, in a "flattened" row vector format (i.e., 1x784 vectors), hence the
         * "convolutionalFlat" input type used here.
         */
        return new NeuralNetConfiguration.Builder().seed(SEED).iterations(NUM_ITERATIONS).regularization(true).l2(0.0005).
        /*
         * Uncomment the following for learning decay and bias
         */
                        learningRate(.01).// biasLearningRate(0.02).
                        // learningRateDecayPolicy(LearningRatePolicy.Inverse).lrPolicyDecayRate(0.001).lrPolicyPower(0.75).
                        weightInit(WeightInit.XAVIER).optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
                        .updater(Updater.NESTEROVS).momentum(0.9).list()
                        .layer(0,
                                        new ConvolutionLayer.Builder(5, 5).nIn(NUM_CHANNELS).stride(1, 1).nOut(20).activation(Activation.IDENTITY)
                                                        .build())
                        .layer(1, new SubsamplingLayer.Builder(SubsamplingLayer.PoolingType.MAX).kernelSize(2, 2).stride(2, 2).build())
                        .layer(2, new ConvolutionLayer.Builder(5, 5).stride(1, 1).nOut(50).activation(Activation.IDENTITY).build())
                        .layer(3, new SubsamplingLayer.Builder(SubsamplingLayer.PoolingType.MAX).kernelSize(2, 2).stride(2, 2).build())
                        .layer(4, new DenseLayer.Builder().activation(Activation.RELU).nOut(500).build())
                        .layer(5,
                                        new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD).nOut(NUM_OUTPUTS)
                                                        .activation(Activation.SOFTMAX).build())
                        .setInputType(InputType.convolutionalFlat(28, 28, 1)).backprop(true).pretrain(false).build();
}
```

Example 26

Project: *ceidg-captcha* File: *MultiLayerConfigurationFactoryImpl.java* *View source code* 5 votes

```
@Override
public MultiLayerConfiguration create() {
    int width = imageTransformConfigurationResource.getScaledWidth();
    int height = imageTransformConfigurationResource.getScaledHeight();
    int channels = imageTransformConfigurationResource.getChannels();
    int outputs = networkConfigurationResource.getOutputs();
    return new NeuralNetConfiguration.Builder()
            .seed(seed)
            .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
            .iterations(1)
            .learningRate(0.0001)
            .activation(Activation.RELU)
            .weightInit(WeightInit.XAVIER)
            .updater(Updater.NESTEROVS).momentum(0.9)
            .regularization(true).l2(1e-3)
            .list()
            .layer(0, new DenseLayer.Builder()
                    .nIn(width * height * channels)
                    .nOut(1200)
                    .build())
            .layer(1, new DenseLayer.Builder()
                    .nIn(1200)
                    .nOut(600)
                    .build())
            .layer(2, new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
                    .nIn(600)
                    .activation(Activation.SOFTMAX)
                    .nOut(outputs)
                    .build())
            .pretrain(false).backprop(true)
            .setInputType(InputType.convolutional(height, width, channels))
            .build();
}
```

Example 27

Project: *Deep-Learning-with-Hadoop* File: *DeepAutoEncoder.java* *View source code* 5 votes

```
public static void main(String[] args) throws Exception {
    final int numRows = 28;
    final int numColumns = 28;
    int seed = 123;
    int numSamples = MnistDataFetcher.NUM_EXAMPLES;
    int batchSize = 1000;
    int iterations = 1;
    int listenerFreq = iterations/5;

    log.info("Load data....");
    DataSetIterator iter = new MnistDataSetIterator(batchSize,numSamples,true);

    log.info("Build model....");
    MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
            .seed(seed)
            .iterations(iterations)
            .optimizationAlgo(OptimizationAlgorithm.LINE_GRADIENT_DESCENT)
            .list(8)
            .layer(0, new RBM.Builder().nIn(numRows * numColumns).nOut(2000).lossFunction(LossFunctions.LossFunction.RMSE_XENT).build())
            .layer(1, new RBM.Builder().nIn(2000).nOut(1000).lossFunction(LossFunctions.LossFunction.RMSE_XENT).build())
            .layer(2, new RBM.Builder().nIn(1000).nOut(500).lossFunction(LossFunctions.LossFunction.RMSE_XENT).build())
            .layer(3, new RBM.Builder().nIn(500).nOut(30).lossFunction(LossFunctions.LossFunction.RMSE_XENT).build())
            .layer(4, new RBM.Builder().nIn(30).nOut(500).lossFunction(LossFunctions.LossFunction.RMSE_XENT).build())
            .layer(5, new RBM.Builder().nIn(500).nOut(1000).lossFunction(LossFunctions.LossFunction.RMSE_XENT).build())
            .layer(6, new RBM.Builder().nIn(1000).nOut(2000).lossFunction(LossFunctions.LossFunction.RMSE_XENT).build())
            .layer(7, new OutputLayer.Builder(LossFunctions.LossFunction.MSE).activation(Activation.SIGMOID).nIn(2000).nOut(numRows*numColumns).build())
            .pretrain(true).backprop(true)
            .build();

    MultiLayerNetwork model = new MultiLayerNetwork(conf);
    model.init();

    model.setListeners(new ScoreIterationListener(listenerFreq));

    log.info("Train model....");
    while(iter.hasNext()) {
        DataSet next = iter.next();
        model.fit(new DataSet(next.getFeatureMatrix(),next.getFeatureMatrix()));
    }
}
```

Example 28

Project: *Deep-Learning-with-Hadoop* File: *TestSparkMultiLayerParameterAveraging.java* *View source code* 5 votes

```
@Test
public void testSmallAmountOfData(){


    MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
            .updater(Updater.RMSPROP)
            .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT).iterations(1)
            .list()
            .layer(0, new org.deeplearning4j.nn.conf.layers.DenseLayer.Builder()
                    .nIn(nIn).nOut(3)
                    .activation("tanh").build())
            .layer(1, new org.deeplearning4j.nn.conf.layers.OutputLayer.Builder(LossFunctions.LossFunction.MSE)
                    .nIn(3).nOut(nOut)
                    .activation("softmax")
                    .build())
            .build();

    SparkDl4jMultiLayer sparkNet = new SparkDl4jMultiLayer(sc,conf,new ParameterAveragingTrainingMaster(true,numExecutors(),1,10,1,0));

    Nd4j.getRandom().setSeed(12345);
    DataSet d1 = new DataSet(Nd4j.rand(1,nIn),Nd4j.rand(1,nOut));
    DataSet d2 = new DataSet(Nd4j.rand(1,nIn),Nd4j.rand(1,nOut));

    JavaRDD<DataSet> rddData = sc.parallelize(Arrays.asList(d1,d2));

    sparkNet.fit(rddData);

}
```

Example 29

Project: *NeuralNetworksLite* File: *RegressionMathFunctions.java* *View source code* 5 votes

```
public static void main(final String[] args){

        //Switch these two options to do different functions with different networks
        final MathFunction fn = new SinXDivXMathFunction();
        final MultiLayerConfiguration conf = getDeepDenseLayerNetworkConfiguration();

        //Generate the training data
        final INDArray x = Nd4j.linspace(-10,10,nSamples).reshape(nSamples, 1);
        final DataSetIterator iterator = getTrainingData(x,fn,batchSize,rng);

        //Create the network
        final MultiLayerNetwork net = new MultiLayerNetwork(conf);
        net.init();
        net.setListeners(new ScoreIterationListener(1));


        //Train the network on the full data set, and evaluate in periodically
        final INDArray[] networkPredictions = new INDArray[nEpochs/ plotFrequency];
        for( int i=0; i<nEpochs; i++ ){
            iterator.reset();
            net.fit(iterator);
            if((i+1) % plotFrequency == 0) networkPredictions[i/ plotFrequency] = net.output(x, false);
        }

        //Plot the target data and the network predictions
        plot(fn,x,fn.getFunctionValues(x),networkPredictions);
    }
```

Example 30

Project: *blueweave* File: *TimeseriesClassifierNetwork.java* View source code 5 votes

```
private TimeseriesClassifierNetwork(Map<Integer, String> trainClasses, Map<String, Integer> trainClassifications,
                                   Table<Date, String, Double> trainTable, Table<Date, String, Double> testTable,
                                    Config configuration,
                                    MultiLayerConfiguration multiLayerConfig) {
    this();
    this.trainClasses = trainClasses;
    this.trainClassifications = trainClassifications;
    this.trainTable = trainTable;
    this.testTable = testTable;
    super.config = configuration;
    super.multiLayerConfiguration = multiLayerConfig;

}
```

Example 31

Project: *ml_demo* File: *MammographyAutoencoder.java* View source code 5 votes

```
/**
 * Set up network.
 * 6 in- and output dimensions (as mammography data has 6 feature dimensions).
 * 6 -> intermediate-dim -> core-dim -> intermediate-dim -> 6
 */
private static MultiLayerNetwork createNet(int intermediateDimensions, int coreDimensions) {
        int inputDimensions = 6;
        int outputDimensions = inputDimensions;

        MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
                        .seed(12345)
                        .iterations(1)
                        .weightInit(WeightInit.XAVIER)
                        .updater(Updater.ADAGRAD)
                        .activation(Activation.RELU)
                        .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
                        .learningRate(0.05)
                        .regularization(true).l2(0.0001)
                        .list()
                        .layer(0, new DenseLayer.Builder()
                                        .nIn(inputDimensions)
                                        .nOut(intermediateDimensions)
                                        .build())
                        .layer(1, new DenseLayer.Builder()
                                        .nIn(intermediateDimensions)
                                        .nOut(coreDimensions)
                                        .build())
                        .layer(2, new DenseLayer.Builder()
                                        .nIn(coreDimensions)
                                        .nOut(intermediateDimensions)
                                        .build())
                        .layer(3, new OutputLayer.Builder()
                                        .nIn(intermediateDimensions)
                                        .nOut(outputDimensions)
                                        .lossFunction(LossFunctions.LossFunction.MSE)
                                        .build())
                        .pretrain(false)
                        .backprop(true)
                        .build();

        return new MultiLayerNetwork(conf);
}
```

Example 32

Project: *DL4J* File: *LstmModel.java* View source code 5 votes

```
@Override
   protected MultiLayerConfiguration getConfiguration()
   {
final int[] hiddenLayerNodes = parameters.getHiddeLayerNodes();
final int nLayers = hiddenLayerNodes.length + 1;

final ListBuilder list = new NeuralNetConfiguration.Builder()
        .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
        .iterations(parameters.getIterations()).learningRate(parameters.getLearningRate()).rmsDecay(0.95)
        .seed(parameters.getSeed()).regularization(true).l2(0.001).list(nLayers).pretrain(false).backprop(true);

for (int i = 0; i < nLayers; i++)
{
    int nIn;
    if (i == 0)
    {
        nIn = parameters.getInputSize();
    }
    else
    {
        nIn = hiddenLayerNodes[i - 1];
    }
    if (i < nLayers - 1)
    {
        final GravesLSTM layer = new GravesLSTM.Builder().nIn(nIn).nOut(hiddenLayerNodes[i])
                .updater(Updater.RMSPROP).activation("tanh").weightInit(WeightInit.DISTRIBUTION)
                .dist(new UniformDistribution(-0.08, 0.08)).build();
        list.layer(i, layer);
    }
    else
    {
        final RnnOutputLayer outputLayer = new RnnOutputLayer.Builder(LossFunction.MCXENT).activation("softmax")
                .updater(Updater.RMSPROP).nIn(hiddenLayerNodes[1]).nOut(parameters.getOutputSize())
                .weightInit(WeightInit.DISTRIBUTION).dist(new UniformDistribution(-0.08, 0.08)).build();
        list.layer(i, outputLayer);
    }
}
return list.build();
   }
```

Example 33

Project: *DL4J* File: *DeepAutoEncoderModel.java* View source code   5 votes

```
@Override
   protected MultiLayerConfiguration getConfiguration()
   {
final int[] hiddenLayerNodes = parameters.getHiddeLayerNodes();
final int nLayers = hiddenLayerNodes.length;
final ListBuilder list = new NeuralNetConfiguration.Builder().seed(parameters.getSeed())
        .iterations(parameters.getIterations()).optimizationAlgo(OptimizationAlgorithm.LINE_GRADIENT_DESCENT)
        .list(nLayers);
for (int i = 0; i < nLayers; i++)
{
    int nIn;
    if (i == 0)
    {
        nIn = parameters.getInputSize();
    }
    else
    {
        nIn = hiddenLayerNodes[i - 1];
    }

    if (i < nLayers - 1)
    {
        final RBM hiddenLayer = new RBM.Builder().nIn(nIn).nOut(hiddenLayerNodes[i])
                .lossFunction(LossFunctions.LossFunction.RMSE_XENT).build();
        list.layer(i, hiddenLayer);
    }
    else
    {
        final OutputLayer outputLayer = new OutputLayer.Builder(LossFunctions.LossFunction.RMSE_XENT)
                .nIn(nIn).nOut(parameters.getOutputSize()).build();
        list.layer(nLayers - 1, outputLayer);
    }
}
return list.pretrain(true).backprop(true).build();
   }
```

Example 34

Project: *wekaDeeplearning4j* File: *ZooModel.java* View source code   5 votes

```
/**
 * Convert a MultiLayerConfiguration into a Computation graph
 *
 * @param mlc Layer-wise configuration
 * @param shape Inputshape
 * @return ComputationGraph based on the configuration in the MLC
 */
default ComputationGraph mlpToCG(MultiLayerConfiguration mlc, int[][] shape) {
  ComputationGraphConfiguration.GraphBuilder builder =
      new NeuralNetConfiguration.Builder()
          .trainingWorkspaceMode(WorkspaceMode.SEPARATE)
          .inferenceWorkspaceMode(WorkspaceMode.SEPARATE)
          .graphBuilder();
  List<NeuralNetConfiguration> confs = mlc.getConfs();
  // Start with input
  String currentInput = "input";
  builder.addInputs(currentInput);

  // Iterate MLN configurations layer-wise
  for (NeuralNetConfiguration conf : confs) {
    Layer l = conf.getLayer();
    String lName = l.getLayerName();

    // Connect current layer with last layer
    builder.addLayer(lName, l, currentInput);
    currentInput = lName;
  }
  builder.setOutputs(currentInput);

  // Configure inputs
  builder.setInputTypes(InputType.convolutional(shape[0][1], shape[0][2], shape[0][0]));

  // Build
  ComputationGraphConfiguration cgc = builder.build();
  return new ComputationGraph(cgc);
}
```

Example 35

Project: *dl4j-apr* File: *DBN.java* View source code 5 votes

```
public static MultiLayerNetwork getModel(int numInputs) {
    MultiLayerConfiguration conf =  new NeuralNetConfiguration.Builder()
            .seed(seed)
            .iterations(iterations)
            .gradientNormalization(GradientNormalization.ClipElementWiseAbsoluteValue)
            .gradientNormalizationThreshold(1.0)
            .regularization(true)
            .dropOut(Config.DROPOUT)
            .updater(Config.UPDATER)
            .adamMeanDecay(0.5)
            .adamVarDecay(0.5)
            .weightInit(WeightInit.XAVIER)
            .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
            .list()
            .layer(0, new RBM.Builder(RBM.HiddenUnit.BINARY, RBM.VisibleUnit.GAUSSIAN)
                    .nIn(numInputs).nOut(2750).dropOut(0.75)
                    .activation(Activation.RELU).build())
            .layer(1, new RBM.Builder(RBM.HiddenUnit.BINARY, RBM.VisibleUnit.BINARY)
                    .nIn(2750).nOut(2000)
                    .activation(Activation.RELU).build())
            .layer(2, new RBM.Builder(RBM.HiddenUnit.BINARY, RBM.VisibleUnit.BINARY)
                    .nIn(2000).nOut(1000)
                    .activation(Activation.RELU).build())
            .layer(3, new RBM.Builder(RBM.HiddenUnit.BINARY, RBM.VisibleUnit.BINARY)
                    .nIn(1000).nOut(200)
                    .activation(Activation.RELU).build())
            .layer(4, new OutputLayer.Builder(Config.LOSS_FUNCTION)
                    .nIn(200).nOut(Config.NUM_OUTPUTS).updater(Config.UPDATER)
                    .adamMeanDecay(0.6).adamVarDecay(0.7)
                    .build())
            .pretrain(true).backprop(true)
            .build();
    return new MultiLayerNetwork(conf);
}
```

Example 36

Project: *dl4j-apr* File: *DBN.java* View source code 5 votes

```
private MultiLayerNetwork getModelFromJson() throws IOException {
    String path = "./src/main/resources/models/d2v/" + language.getName() + "-model.json";
    byte[] encoded = Files.readAllBytes(Paths.get(path));
    String json =  new String(encoded, StandardCharsets.UTF_8);
    System.out.println(json);
    return new MultiLayerNetwork(MultiLayerConfiguration.fromJson(json));

}
```

Example 37

Project: *dl4j-spark-ml-examples* File: *JavaMnistClassification.java* View source code 5 votes

```
public static MultiLayerConfiguration getConfiguration() {
        final int numRows = 28;
        final int numColumns = 28;
        int nChannels = 1;
        int outputNum = 10;
        int numSamples = 2000;
        int batchSize = 500;
        int iterations = 10;
        int seed = 123;

        MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
                .seed(seed)
                .batchSize(batchSize)
                .iterations(iterations)
                .constrainGradientToUnitNorm(true)
                .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
                .list(3)
                .layer(0, new ConvolutionLayer.Builder(10, 10)
                        .nIn(nChannels)
                        .nOut(6)
                        .weightInit(WeightInit.XAVIER)
                        .activation("relu")
                        .build())
                .layer(1, new SubsamplingLayer.Builder(SubsamplingLayer.PoolingType.MAX, new int[] {2,2})
                        .build())
                .layer(2, new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
                        .nIn(150)
                        .nOut(outputNum)
                        .weightInit(WeightInit.XAVIER)
                        .activation("softmax")
                        .build())
                .inputPreProcessor(0, new FeedForwardToCnnPreProcessor(numRows, numColumns, 1))
                .inputPreProcessor(2, new CnnToFeedForwardPreProcessor())
                .backprop(true).pretrain(false)
                .build();

        return conf;
    }
```

Example 38

Project: *dl4j-spark-ml-examples* File: *JavaIrisClassification.java* View source code 5 votes

```
private static MultiLayerConfiguration getConfiguration() {

        MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
                .seed(11L) // Seed to lock in weight initialization for tuning
                .iterations(100) // # training iterations predict/classify & backprop
                .learningRate(1e-3f) // Optimization step size
                .optimizationAlgo(OptimizationAlgorithm.LINE_GRADIENT_DESCENT) // Backprop method (calculate the gradients)
                .momentum(0.9)
                .constrainGradientToUnitNorm(true)
                .useDropConnect(true)
                .list(2) // # NN layers (does not count input layer)
                .layer(0, new RBM.Builder(RBM.HiddenUnit.RECTIFIED, RBM.VisibleUnit.GAUSSIAN)
                                .nIn(4) // # input nodes
                                .nOut(3) // # fully connected hidden layer nodes. Add list if multiple layers.
                                .weightInit(WeightInit.XAVIER)
                                .activation("relu")
                                .lossFunction(LossFunctions.LossFunction.RMSE_XENT)
                                .updater(Updater.ADAGRAD)
                                .k(1) // # contrastive divergence iterations
                                .dropOut(0.5)
                                .build()
                ) // NN layer type
                .layer(1, new OutputLayer.Builder(LossFunctions.LossFunction.MCXENT)
                                .nIn(3) // # input nodes
                                .nOut(3) // # output nodes
                                .activation("softmax")
                                .weightInit(WeightInit.XAVIER)
                                .updater(Updater.ADAGRAD)
                                .dropOut(0.5)
                                .build()
                ) // NN layer type
                .build();

        return conf;
    }
```

Example 39

Project: *deeplearning4j-spark-ml-examples* File: *JavaMnistClassification.java* View source code 5 votes

```
public static MultiLayerConfiguration getConfiguration() {
        final int numRows = 28;
        final int numColumns = 28;
        int nChannels = 1;
        int outputNum = 10;
        int batchSize = 100;
        int iterations = 10;
        int seed = 123;

        MultiLayerConfiguration.Builder builder = new NeuralNetConfiguration.Builder()
                .seed(seed)
                .batchSize(batchSize)
                .iterations(iterations)
                .constrainGradientToUnitNorm(true)
                .optimizationAlgo(OptimizationAlgorithm.STOCHASTIC_GRADIENT_DESCENT)
                .list(3)
                .layer(0, new ConvolutionLayer.Builder(10, 10)
                        .nIn(nChannels)
                        .nOut(6)
                        .weightInit(WeightInit.XAVIER)
                        .activation("relu")
                        .build())
                .layer(1, new SubsamplingLayer.Builder(SubsamplingLayer.PoolingType.MAX, new int[]{2, 2})
                        .build())
                .layer(2, new OutputLayer.Builder(LossFunctions.LossFunction.NEGATIVELOGLIKELIHOOD)
                        .nIn(150)
                        .nOut(outputNum)
                        .weightInit(WeightInit.XAVIER)
                        .activation("softmax")
                        .build())
                .backprop(true).pretrain(false);
        new ConvolutionLayerSetup(builder, numRows, numColumns, nChannels);

        MultiLayerConfiguration conf = builder.build();
        return conf;
    }
```

Example 40

Project: *deeplearning4j-spark-ml-examples* File: *JavaIrisClassification.java* View source code 5 votes

```
private static MultiLayerConfiguration getConfiguration() {

        MultiLayerConfiguration conf = new NeuralNetConfiguration.Builder()
                .seed(11L) // Seed to lock in weight initialization for tuning
                .iterations(100) // # training iterations predict/classify & backprop
                .learningRate(1e-3f) // Optimization step size
                .optimizationAlgo(OptimizationAlgorithm.LINE_GRADIENT_DESCENT) // Backprop method (calculate the gradients)
                .momentum(0.9)
                .constrainGradientToUnitNorm(true)
                .useDropConnect(true)
                .list(2) // # NN layers (does not count input layer)
                .layer(0, new RBM.Builder(RBM.HiddenUnit.RECTIFIED, RBM.VisibleUnit.GAUSSIAN)
                                .nIn(4) // # input nodes
                                .nOut(3) // # fully connected hidden layer nodes. Add list if multiple layers.
                                .weightInit(WeightInit.XAVIER)
                                .activation("relu")
                                .lossFunction(LossFunctions.LossFunction.RMSE_XENT)
                                .updater(Updater.ADAGRAD)
                                .k(1) // # contrastive divergence iterations
                                .dropOut(0.5)
                                .build()
                ) // NN layer type
                .layer(1, new OutputLayer.Builder(LossFunctions.LossFunction.MCXENT)
                                .nIn(3) // # input nodes
                                .nOut(3) // # output nodes
                                .activation("softmax")
                                .weightInit(WeightInit.XAVIER)
                                .updater(Updater.ADAGRAD)
                                .dropOut(0.5)
                                .build()
                ) // NN layer type
                .build();

        return conf;
    }
```