Tópicos em Avanços Computacionais I

Rede Neural Simples – DL4J

Joinvile Batista Junior

Configurando via Maven

identificação e propriedades do projeto

```
<groupId>edu.comp.dl</groupId>
<artifactId>App1_DL4J</artifactId>
<version>1.0-SNAPSHOT</version>
<packaging>jar</packaging>
<properties>
    <project.build.sourceEncoding>UTF-8
    </project.build.sourceEncoding>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
     <dl4j.version>1.0.0-beta2</dl4j.version>
</properties>
```

Configurando via Maven
dependências

```
<dependencies>
    <dependency>
        <groupId>org.nd4j</groupId>
        <artifactId>nd4j-native-platform</artifactId>
        <version>${dl4j.version}</version>
    </dependency>
    <dependency>
        <groupId>org.deeplearning4j</groupId>
        <artifactId>deeplearning4j-core</artifactId>
        <version>${dl4j.version}</version>
    </dependency>
    <dependency>
        <groupId>org.nd4j</groupId>
        <artifactId>nd4j-api</artifactId>
        <version>${dl4j.version}</version>
    </dependency>
</dependencies>
```

Importações
io - datavec

java.io
- File
- IOException

org.datavec.api
- records.reader
  - RecordReader
  - impl.csv.CSVRecordReader
- split.FileSplit

Importações
nd4j

org.nd4j.linalg

- activations.Activation
- api.ndarray.INDArray
- learning.config.Sgd
- lossfunctions.LossFunctions.LossFunction
- dataset
  - api.iterator.DataSetIterator
  - DataSet

Importações
dl4j

org.deeplearning4j

- datasets.datavec.RecordReaderDataSetIterator
- eval.Evaluation
- nn
  - conf
    - MultiLayerConfiguration
    - NeuralNetConfiguration
    - layers
      - DenseLayer
      - OutputLayer
  - multilayer.MultiLayerNetwork
  - weights.WeightInit
- optimize.listeners.ScoreIterationListener

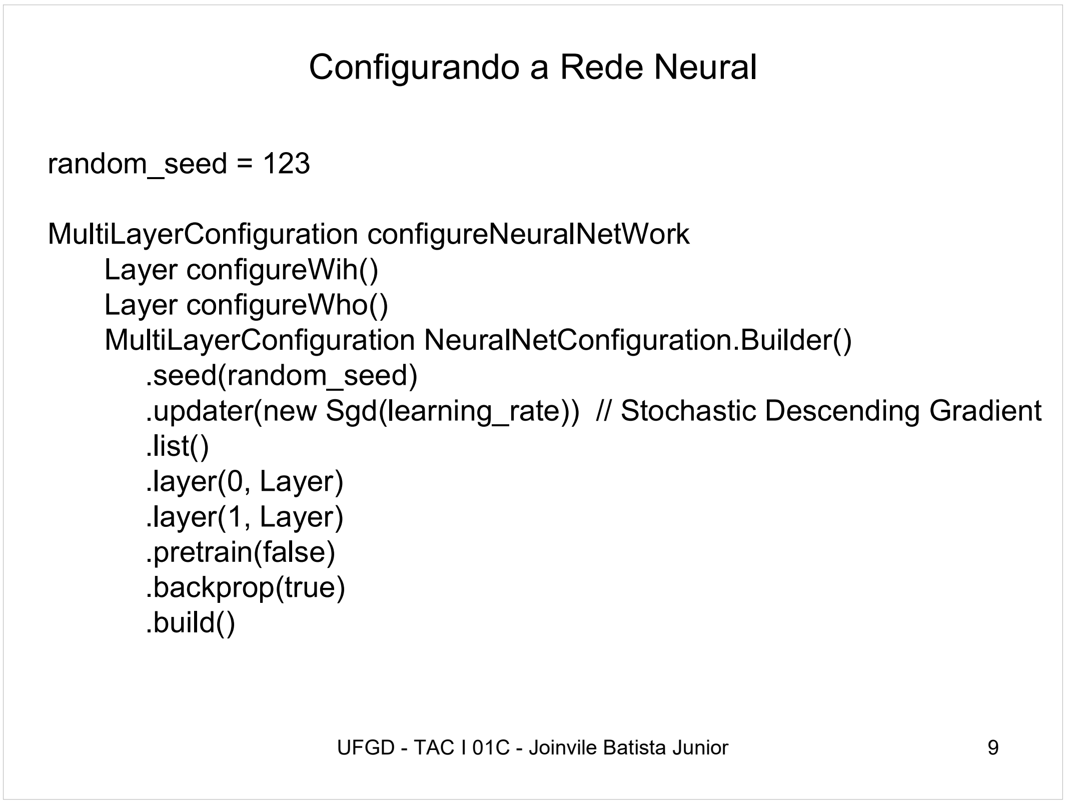
Rede Neural

configurando – criando – treinando – avaliando

```
main(String[ ]) throws IOException, InterruptedException
    MultiLayerConfiguration configureNeuralNetWork()
    MultiLayerNetwork createNeuralNetworkModel(MultiLayerConfiguration)
    DataSetIterator readCSV_File(String) // training
    trainNeuralNetwork(MultiLayerNetwork, DataSetIterator)
    DataSetIterator readCSV_File(String) // test
    evaluateNeuralNetwork(MultiLayerNetwork, DataSetIterator)
```

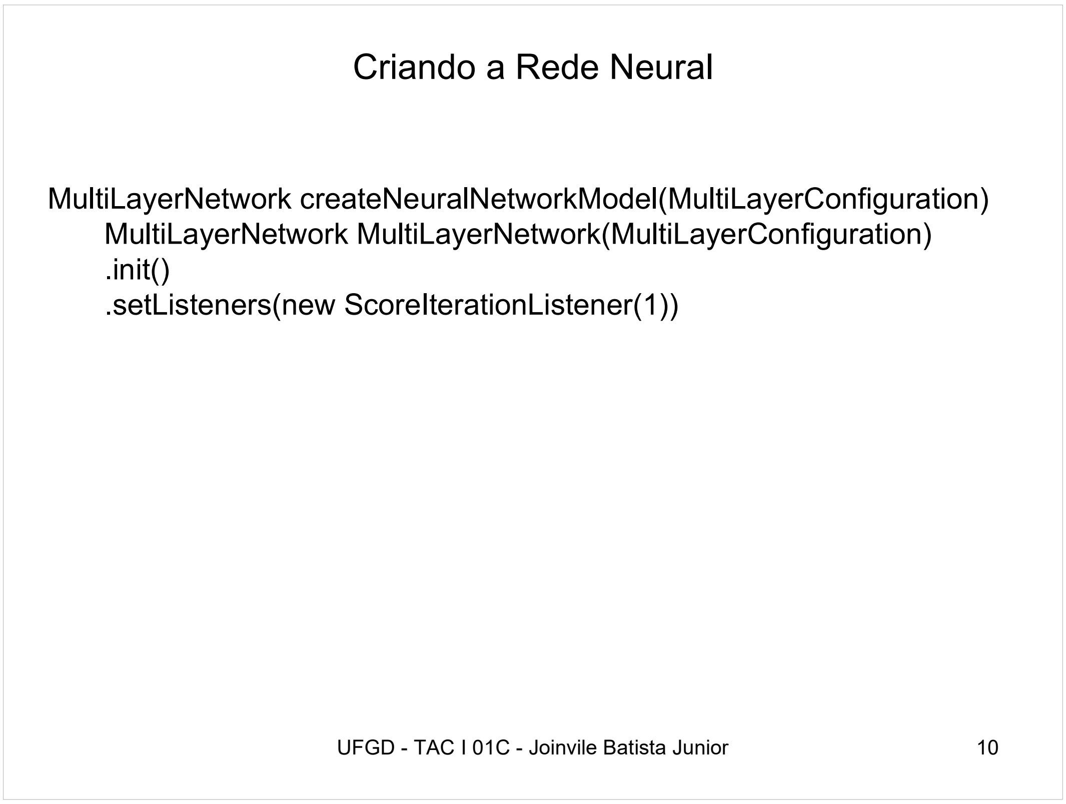
Configurando as Conexões entre as Camadas da Rede

```
Layer configureWih
    DenseLayer.Builder()
        .nIn(input_nodes)
        .nOut(output_nodes)
        .activation(Activation.SIGMOID)
        .weightInit(WeightInit.NORMAL)
        .build()

Layer configureWho
    OutputLayer.Builder(LossFunction.MSE)  // Mean Square Error
        idem
```



Configurando a Rede Neural

```
random_seed = 123

MultiLayerConfiguration configureNeuralNetWork
      Layer configureWih()
      Layer configureWho()
      MultiLayerConfiguration NeuralNetConfiguration.Builder()
          .seed(random_seed)
          .updater(new Sgd(learning_rate))  // Stochastic Descending Gradient
          .list()
          .layer(0, Layer)
          .layer(1, Layer)
          .pretrain(false)
          .backprop(true)
          .build()
```



Criando a Rede Neural

```
MultiLayerNetwork createNeuralNetworkModel(MultiLayerConfiguration)
      MultiLayerNetwork MultiLayerNetwork(MultiLayerConfiguration)
      .init()
      .setListeners(new ScoreIterationListener(1))
```

Lendo Dados de Treinamento e de Teste

```
batch_size = 32

DataSetIterator readCSV_File(String)
                throws IOException, InterruptedException
     RecordReader CSVRecordReader()
     .initialize(new FileSplit(new File(String)))
     DataSetIterator RecordReaderDataSetIterator
          (RecordReader, batch_size, correct_digit_index, output_classes)
```

Treinando a Rede Neural

```
void trainNeuralNetwork(MultiLayerNetwork, DataSetIterator)
     loop epochs
          .fit(DataSetIterator)
```

Avaliando a Rede Neural

```
void evaluateNeuralNetwork
            (MultiLayerNetwork model, DataSetIterator iterator)
     Evaluation score = new Evaluation(output_classes)
      while(iterator.hasNext()){
           DataSet next_data = iterator.next()
           INDArray model.output(next_data.getFeatures())
           score.eval(next_data.getLabels(), INDArray)
      }
      imprimir : score.stats())
```