



NeuroJava Core

Inteligência Artificial com Java

Inteligência Artificial com Java

Comece do Zero e Vá Longe

A Inteligência Artificial (IA) está por toda parte — nos sistemas de recomendação da Netflix, nos assistentes de voz, nos filtros de spam do e-mail. O que antes parecia ficção científica hoje é realidade, e programadores do mundo inteiro já estão colocando a IA em prática nos seus próprios projetos. Este e-book é para você que quer começar agora, mesmo sem experiência prévia com IA, mas já com alguma familiaridade com Java. Aqui, vamos direto ao ponto: sem enrolação, sem termos complicados. Cada conceito é explicado de forma simples e sempre acompanhado de exemplos reais de código em Java. Você vai aprender, passo a passo: Como tomar decisões com regras; Criar modelos com aprendizado de máquina usando a biblioteca Weka; Desenvolver redes neurais simples com Neuroph; E muito mais nos próximos capítulos. Se você sabe programar em Java, já tem o necessário para dar os primeiros passos com Inteligência Artificial. Vamos lá?



01

O que é IA e por que Java pode ser sua aliada

Descubra o que é Inteligência Artificial e por que Java é uma excelente linguagem para começar a aplicá-la em projetos reais.



O que é IA e por que Java pode ser sua aliada

Inteligência Artificial: Entenda em Minutos

A IA é a capacidade de uma máquina simular comportamentos inteligentes, como tomar decisões, identificar padrões ou aprender com experiências. Código exemplo: Sistema que recomenda uma ação com base na temperatura

```
public class TermometroInteligente {  
    public static void main(String[] args) {  
        int temperatura = 38;  
  
        if (temperatura >= 37) {  
            System.out.println("Temperatura alta. Recomenda-se repouso.");  
        } else {  
            System.out.println("Temperatura normal.");  
        }  
    }  
}
```

O que é IA e por que Java pode ser sua aliada

Java na Era da IA

Java é uma linguagem madura e confiável para aplicações sérias, e possui ferramentas poderosas para aplicar inteligência artificial, como Weka, Neuroph, Encog e Deeplearning4j.

Código exemplo: Simulação de uma escolha baseada em tempo de resposta

```
public class ChatbotBasico {  
    public static void main(String[] args) {  
        int tempoResposta = 5; // segundos  
  
        if (tempoResposta ≤ 3) {  
            System.out.println("Resposta rápida detectada. Usuário engajado.");  
        } else {  
            System.out.println("Resposta lenta. Envie um lembrete.");  
        }  
    }  
}
```

O que é IA e por que Java pode ser sua aliada

Seu Primeiro Programa com Lógica Inteligente

Um sistema baseado em regras pode simular decisões simples do dia a dia com if-else. Código exemplo: Sistema de controle de ambiente inteligente

```
public class ChatbotBasico {  
    public static void main(String[] args) {  
        int tempoResposta = 5; // segundos  
  
        if (tempoResposta ≤ 3) {  
            System.out.println("Resposta rápida detectada. Usuário engajado.");  
        } else {  
            System.out.println("Resposta lenta. Envie um lembrete.");  
        }  
    }  
}
```

02

Pensar em regras: o primeiro passo para IA

Aprenda a criar sistemas inteligentes com regras simples usando if-else, o primeiro passo antes de usar algoritmos mais avançados.



Pensar em regras: o primeiro passo para IA

IA com If-Else: A Lógica do Pensamento

A estrutura if-else permite que programas "pensem" com base em condições lógicas. Código exemplo: Verificação de idade para acesso a conteúdo

```
public class ControleDeAcesso {  
    public static void main(String[] args) {  
        int idade = 17;  
  
        if (idade ≥ 18) {  
            System.out.println("Acesso permitido.");  
        } else {  
            System.out.println("Acesso negado. Idade insuficiente.");  
        }  
    }  
}
```


Pensar em regras: o primeiro passo para IA

Casos do Mundo Real: Diagnóstico e Reação

Muitas decisões do cotidiano podem ser representadas por regras simples, como em sistemas médicos, automação de casas ou suporte técnico. Código exemplo: Diagnóstico simples com base em sintomas

```
public class DiagnosticoSimples {  
    public static void main(String[] args) {  
        boolean dorCabeca = true;  
        boolean febre = true;  
  
        if (dorCabeca && febre) {  
            System.out.println("Possível virose detectada.");  
        } else {  
            System.out.println("Sem sintomas preocupantes.");  
        }  
    }  
}
```

Pensar em regras: o primeiro passo para IA

Evoluindo para Árvores de Decisão

Quando há muitas decisões encadeadas, organizar essas escolhas em forma de árvore facilita a lógica. Código exemplo: IA que sugere ação baseada em horário e tarefa

```
public class PlanejadorIA {  
    public static void main(String[] args) {  
        String periodo = "manhã";  
        boolean temReuniao = true;  
  
        if (periodo.equals("manhã")) {  
            if (temReuniao) {  
                System.out.println("Preparar apresentação.");  
            } else {  
                System.out.println("Verificar e-mails.");  
            }  
        } else {  
            System.out.println("Organizar tarefas da tarde.");  
        }  
    }  
}
```

03

Seu primeiro modelo de aprendizado com Weka

Use a biblioteca Weka para treinar modelos de machine learning com dados reais e aprender a classificar informações automaticamente.

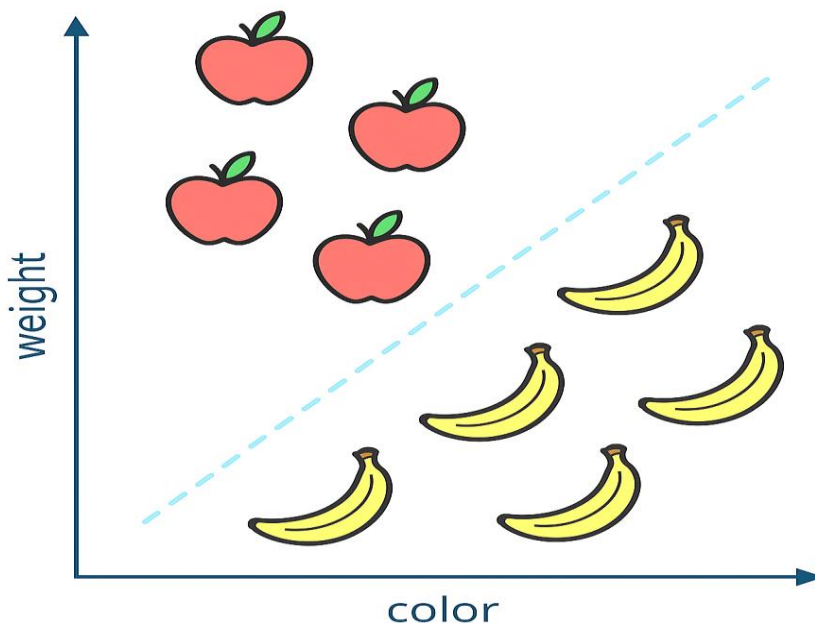


Seu primeiro modelo de aprendizado com Weka

O que é Machine Learning e como o Weka ajuda

O aprendizado de máquina (machine learning) é a área da IA que permite que programas aprendam com dados. Em vez de programar todas as regras, você alimenta o programa com exemplos — e ele aprende. O Weka é uma biblioteca Java poderosa que permite aplicar algoritmos de machine learning com poucas linhas de código. Ideal para quem está começando. Exemplo ilustrativo: Imagine que você tem um conjunto de dados com frutas. O Weka pode aprender a identificar se uma fruta é uma maçã ou uma banana com base em cor e peso.

Apple or banana?



Seu primeiro modelo de aprendizado com Weka

Preparando o Ambiente e os Dados

Para usar o Weka em Java:

1. Adicione o .jar ao projeto ou use Maven:

```
<dependency>
  <groupId>nz.ac.waikato.cms.weka</groupId>
  <artifactId>weka-stable</artifactId>
  <version>3.8.6</version>
</dependency>
```

2. Prepare um arquivo .arff com seus dados.
Exemplo:

```
@relation frutas

@attribute cor {vermelha, amarela}
@attribute peso numeric
@attribute tipo {maca, banana}

@data
vermelha,150,maca
amarela,120,banana
vermelha,130,maca
amarela,140,banana
```

Seu primeiro modelo de aprendizado com Weka

Construindo um Classificador com Weka

Vamos criar um modelo que aprende a identificar frutas com base nos dados.

```
import weka.classifiers.trees.J48;
import weka.core.Instances;
import weka.core.converters.ConverterUtils.DataSource;

public class ClassificadorFrutas {
    public static void main(String[] args) throws Exception {
        DataSource source = new DataSource("frutas.arff");
        Instances dados = source.getDataSet();
        dados.setClassIndex(dados.numAttributes() - 1);

        J48 arvore = new J48(); // Algoritmo de árvore de decisão
        arvore.buildClassifier(dados);

        System.out.println(arvore);
    }
}
```

snappify.com

04

Redes neurais básicas com Neuroph

Crie redes neurais simples com Java e simule o funcionamento de um cérebro artificial com a biblioteca Neuroph.



Redes neurais básicas com Neuroph

Como Funciona uma Rede Neural? Uma Explicação Simples

Redes neurais simulam o cérebro humano. São formadas por "neurônios" artificiais que recebem entradas, processam e produzem uma saída. O tipo mais simples é o Perceptron, ideal para aprender regras lógicas simples como AND, OR ou XOR.

- Baixe a biblioteca Neuroph: <https://neuroph.sourceforge.net>
 - Adicione o .jar ao seu projeto Java.

Vamos treinar uma rede para resolver o problema lógico OR

```
import org.neuroph.core.data.DataSet;
import org.neuroph.nnet.Perceptron;

public class ORNeural {
    public static void main(String[] args) {
        DataSet treino = new DataSet(2, 1);
        treino.add(new double[]{0, 0}, new double[]{0});
        treino.add(new double[]{0, 1}, new double[]{1});
        treino.add(new double[]{1, 0}, new double[]{1});
        treino.add(new double[]{1, 1}, new double[]{1});

        Perceptron rede = new Perceptron(2, 1);
        rede.learn(treino);

        rede.setInput(1, 0);
        rede.calculate();

        System.out.println("Saída: " + rede.getOutput()[0]);
    }
}
```


Redes neurais básicas com Neuroph

Ajustando e Interpretando os Resultados

Após o treinamento, podemos testar novas entradas e ver a resposta da rede.

Exemplo: testando a rede com outras entradas

```
rede.setInput(0, 0);
rede.calculate();
System.out.println("0 OR 0 = " + rede.getOutput()[0]);

rede.setInput(1, 1);
rede.calculate();
System.out.println("1 OR 1 = " + rede.getOutput()[0]);
```

05

Criando Chatbots com Java

Desenvolva um chatbot funcional com respostas automáticas usando Java e aprenda a deixá-lo mais inteligente.



Criando Chatbots com Java

O que é um Chatbot e como funciona?

Um chatbot é um programa que conversa com o usuário, podendo responder perguntas, executar tarefas ou simular um atendimento. O mais básico funciona com palavras-chave e respostas predefinidas, enquanto os mais avançados usam IA.

Vamos criar um chatbot básico que responde perguntas sobre clima:

```
import java.util.Scanner;

public class ChatBot {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Olá! Como posso ajudar?");

        while (true) {
            String pergunta = scanner.nextLine().toLowerCase();

            if (pergunta.contains("clima")) {
                System.out.println("Hoje está ensolarado com 30°C.");
            } else if (pergunta.contains("hora")) {
                System.out.println("Agora são " + java.time.LocalDateTime.now());
            } else if (pergunta.contains("sair")) {
                System.out.println("Tchau! Volte sempre.");
                break;
            } else {
                System.out.println("Desculpe, não entendi sua pergunta.");
            }
        }

        scanner.close();
    }
}
```

06

Visão Computacional com JavaCV

Explore como Java pode interpretar imagens, abrir a webcam e reconhecer padrões com a biblioteca JavaCV.



Visão Computacional com JavaCV

O que é Visão Computacional?

É a área da IA que permite às máquinas “ver” imagens ou vídeos, reconhecer objetos, rostos, placas de carro, entre outros. Em Java, a biblioteca mais usada é o JavaCV, que é um wrapper do OpenCV.

Adicione as dependências no seu pom.xml (usando Maven):

```
<dependency>
  <groupId>org.bytedeco</groupId>
  <artifactId>javacv-platform</artifactId>
  <version>1.5.9</version>
</dependency>
```

Visão Computacional com JavaCV

Abrindo a Webcam com JavaCV

Vamos criar um programa que abre a câmera e mostra o vídeo em tempo real:

```
import org.bytedeco.javacv.*;

public class WebcamViewer {
    public static void main(String[] args) throws FrameGrabber.Exception {
        OpenCVFrameGrabber grabber = new OpenCVFrameGrabber(0);
        grabber.start();

        CanvasFrame frame = new CanvasFrame("Webcam");
        frame.setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);

        while (frame.isVisible()) {
            Frame videoFrame = grabber.grab();
            frame.showImage(videoFrame);
        }

        grabber.stop();
    }
}
```

07

Algoritmos Genéticos com Encog

Implemente algoritmos genéticos para resolver problemas complexos simulando evolução natural com Java e Encog.



Visão Computacional com JavaCV

O que são Algoritmos Genéticos?

São técnicas inspiradas na evolução natural: seleção, cruzamento e mutação. Usados para resolver problemas onde não há uma solução clara, como otimização de rotas, parâmetros ou estratégias.

Encog é uma biblioteca poderosa para redes neurais e algoritmos genéticos.

Adicione ao Maven:

```
<dependency>
  <groupId>org.encog</groupId>
  <artifactId>encog-core</artifactId>
  <version>3.4</version>
</dependency>
```


Visão Computacional com JavaCV

Resolvendo o problema do alvo com Algoritmo Genético

Vamos simular uma IA que tenta adivinhar uma string ("HELLO") por tentativas e erro evolutivo.

```
public class AdivinhaString {
    public static void main(String[] args) {
        String alvo = "HELLO";
        int tentativas = 0;
        String tentativa = "";

        while (!alvo.equals(tentativa)) {
            tentativa = "";
            for (int i = 0; i < alvo.length(); i++) {
                tentativa += (char) (65 + (int)(Math.random() * 26));
            }
            tentativas++;
            System.out.println("Tentativa " + tentativas + ": " + tentativa);
        }

        System.out.println("Acertou: " + tentativa + " em " + tentativas + " tentativas.");
    }
}
```

(Esse é um modelo simplificado de algoritmo genético — sem cruzamento ou mutação, mas já ilustra bem o conceito.)

Agradecimentos



OBRIGADO POR LER ATÉ AQUI

Esse Ebook foi gerado por IA, e diagramado por humano.
O passo a passo se encontra no meu Github

Esse conteúdo foi gerado com fins didáticos de construção, não
foi realizado uma validação cuidadosa humana no conteúdo e
pode conter erros gerados por uma IA.



<https://github.com/gzeved14/ebook-java/>