**Languages Evolve to Support AI Development**

**As linguagens evoluem para apoiar o desenvolvimento de IA**

On novembro 18, 2020 in All, Computing, EIT 2020: The Intelligent Revolution by M. Tim Jones

Disponível em: https://br.mouser.com/blog/languages-evolve-to-support-ai-development

Nos primeiros dias, a IA era alimentada predominantemente pela linguagem de programação Lisp (LISt Processor) em hardware dedicado que executava operações Lisp primitivas. Lisp foi uma das primeiras linguagens e foi eficiente no processamento de listas de itens. As máquinas de uso geral se tornaram moda e os modelos de programação seguiram o exemplo. Mas com o ressurgimento do aprendizado de máquina e, em particular, do aprendizado profundo, novas abordagens e kits de ferramentas otimizam esses fluxos de dados. Aqui, exploraremos a confluência de aprendizado de máquina e plataformas de software.

**Histórico de IA inicial**

A inteligência artificial e o Lisp estavam inexoravelmente entrelaçados porque o conceito e a linguagem se originaram da mesma pessoa, John McCarthy (1927-2011). Em sua forma inicial, a IA estava focada em busca e processamento simbólico mais do que as abordagens numéricas que dominam hoje. Lisp, com sua capacidade de representar dados complexos de forma simples e natural e seu uso de recursão (que é usado para iteração e pesquisa) o tornou ideal para muitos problemas da época. E com seu interpretador interativo – chamado REPL ou Read Evaluate Print Loop – o Lisp facilitou a programação exploratória, o que era ideal para resolver problemas que não eram totalmente compreendidos.

Mas o poder de Lisp também foi seu maior detrator; seu estilo funcional de programação era difícil e abriu as portas para novos paradigmas de linguagem de programação. E enquanto a programação funcional continua a ser usada hoje, as linguagens imperativas, orientadas a objetos e multiparadigmáticas são mais comuns hoje em dia.

**Linguagens modernas de IA**

Embora seja possível desenvolver aplicativos para IA em qualquer linguagem de programação, alguns são melhores que outros. Seja a linguagem em si ou o suporte em torno da linguagem, certas linguagens simplificam muito o desenvolvimento da IA.

**Programação lógica**

A linguagem Prolog foi introduzida em 1972 e tem suas raízes na lógica de primeira ordem onde os programas são definidos por fatos e regras. O programa pode ser consultado para aplicar as regras sobre os fatos e produzir um resultado. Prolog continua sendo amplamente utilizado hoje para aplicações como sistemas especialistas e sistemas de planejamento automatizados. O Prolog foi originalmente projetado para processamento de linguagem natural e continua a encontrar aplicações lá.

**Linguagens de uso geral**

Vinte anos após a introdução do Prolog, surgiu uma linguagem de uso geral chamada Python , projetada em torno da legibilidade do código. Embora o Python tenha despertado interesse precoce como uma linguagem educacional para ensinar programação, ele explodiu em uma linguagem amplamente usada em vários domínios, incluindo inteligência artificial e aprendizado de máquina. Uma das principais vantagens do Python é seu enorme conjunto de bibliotecas e toolkits (kits de ferramentas) que simplificam a criação de aplicativos. Por exemplo, o Python pode ser usado com o kit de ferramentas de código aberto TensorFlow para criar aplicativos de aprendizado profundo. Isso é benéfico quando você deseja implantar o aprendizado profundo sem desenvolver as estruturas detalhadas de rede neural profunda que seriam necessárias.

**Computação Estatística**

Um modelo semelhante foi usado na linguagem R , que é uma linguagem e um ambiente para computação estatística com apresentação gráfica. R é uma linguagem altamente extensível que é expandida através da integração de pacotes. Os pacotes coletam funções e dados juntos para algum aplicativo específico que pode ser usado em programas R, como funções estatísticas ou kits de ferramentas de aprendizado profundo completos. A partir de 2020, mais de 15.000 pacotes estão disponíveis para a linguagem R.

**Novas abordagens funcionais**

Embora o Lisp seja predominantemente uma nota de rodapé no aprendizado de máquina hoje, suas raízes funcionais brotaram novas linguagens que seguem esse paradigma. A linguagem Haskell é uma linguagem puramente funcional com um sistema de tipos forte que resulta em um código mais seguro; uma característica útil ao considerar o aprendizado de máquina e a explosão de dispositivos da Internet das Coisas. Embora não tenha o amplo conjunto de bibliotecas disponíveis para Python e R, Haskell inclui vinculação para kits de ferramentas de aprendizado de máquina, simplificando a criação de aplicativos de aprendizado de máquina com Haskell.

**Toolkits (kits de ferramentas)**

Juntamente com as linguagens, os toolkits e as bibliotecas também evoluíram na busca de aplicativos de aprendizado de máquina. Esses toolkits, como o TensorFlow , fornecem recursos para linguagens para criar aplicativos complexos de aprendizado de máquina sem criar esses recursos desde o início. O TensorFlow fornece interfaces para várias linguagens, como Python, Haskell e R, e simplifica a criação e a implantação de aplicativos de aprendizado profundo.

**Resumo**

O conceito de IA e a formação de sua progênie numérica de aprendizado de máquina criaram uma coevolução de linguagens e toolkits. As linguagens fornecem os recursos de uso geral para criar diversos aplicativos, enquanto os toolkits expandem esses idiomas com recursos específicos de aprendizado de máquina.

Após a leitura do artigo acima, responda as seguintes questões:

1. Qual foi a primeira linguagem aplicada à Inteligência Artificial?
2. Qual a principal vantagem da linguagem Lisp?
3. Que característica da liguagem Lisp pode ser apontado como uma desvantagem, fator que impediu sua ampla adoção?
4. Quando surgiu a linguagem Prolog e qual sua principal aplicação?
5. Qual é uma das principais vantagens da linguagem Python?
6. Por que é interessante usar Python com o kit de ferramentas TensorFlow?
7. Quais são as principais características da linguagem R?
8. Apesar de não ter o amplo conjunto de bibliotecas disponíveis para Python e R que característica importante a linguagem Haskell apresenta?
9. O TensorFlow apresenta interfaces para quais linguagens e porque isso é importante?