

Aplicação de transformações lineares no auxílio ao diagnóstico médico - Análise e Otimização

Jefferson Uchoa Ponte
Gilmaicor Leandro Oliveira



Tempo de Execução ARM/LINUX

Para chegar a uma aproximação com o sistema real utilizamos uma simulação do raspbian no Virtual Box.

- **Encontrar tempo de execução:** utilizamos a biblioteca `time.h` para calcular o tempo de execução. Um laço de repetição aumentando em 10 vezes o número de execuções para obter melhor precisão nas medições. Executamos o algoritmo 5 vezes e tiramos uma média.

Tempo de Execução ARM/LINUX

```
double tempo;  
clock_t t; //variável para armazenar tempo  
int p;  
t = clock(); //armazena tempo  
  
for(p = 0; p < 10; p++){  
    //Aqui começa o treinamento de A.  
    transposta((float*)x, 34, 100,  
    multiplica((float*)x, 34, 100, 1
```

```
t = clock() - t; //tempo final - tempo inicial  
tempo = ((double)t)/((CLOCKS_PER_SEC/1000));  
  
//imprime o tempo na tela  
printf("Tempo de execucao: %lf\n\n", tempo); //c
```

Tempo de Execução no Arduino

Utilizamos o IDE do Arduino para executar uma versão do código otimizada para o arduino na própria placa do arduino, não executando o treinamento da matriz de inteligência, mas apenas fazendo os testes de diagnóstico.

- **Encontrar tempo de execução:** Contagem do tempo utilizando a função `millis()` e a execução do código feita dez vezes para obter maior precisão.

Tempo de Execução no Arduino

```
void main2() {  
    unsigned long meutempo;  
    int i;  
    meutempo = millis();  
    for(i = 0; i < 10; i++){  
        executar();  
    }  
    meutempo = millis() - meutempo;  
    Serial.println(meutempo);  
}
```

Resultados Tempo de Execução Arm/Linux e Arduino

Execução 10 vezes - Tempo em Milisegundos			
	Versão Inicial	Versão Otimizada	Versão SLIM ARDUINO
	28,138	11,896	29
	27,377	12,015	29
	28,788	10,773	29
	28,318	11,94	29
	27,636	11,85	28
Média	28,0514	11,6948	28,8

Medições de Memória ArmLinux e Arduino

Tentativas de medição de Memória no ArmLinux

- O virtual Box disponibiliza uma ferramenta para isso, mas não conseguimos utilizar.
- Tentamos utilizando software Valgrind, mas não chegamos a aprender como utilizar o software corretamente.

Medições de Memória ArmLinux e Arduino

Medições no Arduno

- **Encontrar Memória de dados e de Código:** Dados facilmente obtidos ao compilar a Sketch na IDE do Arduino.
 - ▷ Memória de Código: 3592 bytes(11%)
 - ▷ Memória de dados: 2048 bytes (Total) sendo:
 - ▷ 1142 bytes para variáveis globais(55%)
 - ▷ 906 bytes para variáveis locais(44%)

Otimização Arm/Linux

Otimização para o Arm/Linux

- Utilização de funções e reuso para as operações;
- Utilização de ponteiros para manipulação de matrizes;
- Diminuição de uso de variáveis;

Otimização para o Arduino

Otimização para o Arduino

- Não foi feito o treinamento da matriz de inteligência no arduino. Apenas o teste.
- Distribuiu-se os valores entre locais e globais para não estourar nenhuma das duas.



SOBRE ESTE TRABALHO

- Objetivos iniciais
- Dificuldades (técnicas) encontradas e como contornadas
- Resultados obtidos
- O que aprendemos



REFERÊNCIAS BIBLIOGRÁFICAS

- Exemplo de Aplicação de Transformações Lineares: Sistema de Auxílio ao Diagnóstico Médico, Guilherme de Alencar Barreto

É isso aí pessoal, obrigado!

