

Universidade Federal de Minas Gerais
Escola de Engenharia
Curso de Graduação em Engenharia de Controle e Automação



Aplicação de Redes Neurais em Microcontrolador Embarcado

Relatório 2 de Atividades PFC1

Orientador: Prof. Antônio de Pádua Braga, Dr.
Aluno: João Pedro Miranda Marques
Matricula: 2017050495

10 de junho de 2022

Sumário

1	Resumo	2
2	TensorFlow Lite no ESP32	2
3	GGClassification	3
4	Próximos Passos	3

1 Resumo

Esse relatório tem como objetivo expor as pesquisas e atividades realizadas nessa segunda etapa do PFC1. Nesse projeto iremos implementar o grafo de Gabriel em um sistema embarcado e analisar sua performance. Caso essa etapa se concretize um próximo passo será explorar uma aplicação desse algoritmo no embarcado.

O grafo de Gabriel[3] é um método de reconhecimento de padrões que separa regiões de dados de forma geométrica. O grafo foi desenvolvido pela comunidade da UFMG e demonstrou boa acurácia e baixo uso de recursos computacionais. Por esse motivo foi escolhido como modelo de reconhecimento de padrões a ser utilizado.

A ideia da aplicação a ser implementada é de construir um protótipo de um dispositivo wireless com bateria que possa ser um objeto paupável e faça reconhecimento de padrões de movimento do objeto. Para isso será utilizado um microcontrolador ESP32 e um sensor acelerômetro MPU6050.

Essa aplicação pode ser realizável com a utilização do grafo de Gabriel se mapearmos geometricamente os dados do sensor acelerometro por meio de uma série temporal da aquisição. Reconhecendo assim o movimento realizado em um certo espaço de tempo pré-definido.

2 TensorFlow Lite no ESP32

Primeiramente foi realizado uma validação simples da implementação de uma rede neural no ESP32 utilizando a biblioteca TensorFlow Lite.

Para isso foi validado um exemplo de aplicação encontrado no github onde o autor utilizou a biblioteca em conjunto com a plataforma de desenvolvimento PlatformIO para a classificar se entre duas variáveis float aleatórias de input o segundo valor é maior. Essa simples aplicação tem sua acurácia facilmente verificada pois basta uma operação de comparação.

Quanto ao funcionamento da biblioteca, a rede neural a ser utilizado é modelada e treinada na plataforma do TensorFlow e um arquivo de configuração hexadecimal é gerado. Esse arquivo é adicionado ao firmware e interpretado pela biblioteca TensorFlow Lite de forma a implementar essa rede e gerar as inferências desejadas.

Na Figura 1 vemos o resultado do teste implementado disponível na saída serial do microcontrolador. Nele vemos os dois valores aleatórios de input seguidos do resultado da inferência e o resultado esperado.

```

Used bytes 592
0.44 0.53 - result 0.84 - Expected True, Predicted True
0.63 0.41 - result 0.02 - Expected False, Predicted False
0.98 0.13 - result 0.00 - Expected False, Predicted False
0.24 0.17 - result 0.20 - Expected False, Predicted False
0.10 0.88 - result 1.00 - Expected True, Predicted True
0.01 0.11 - result 0.87 - Expected True, Predicted True
0.00 0.78 - result 1.00 - Expected True, Predicted True
0.10 0.57 - result 1.00 - Expected True, Predicted True
0.67 0.20 - result 0.00 - Expected False, Predicted False
0.94 0.31 - result 0.00 - Expected False, Predicted False
0.91 0.14 - result 0.00 - Expected False, Predicted False
0.25 0.81 - result 1.00 - Expected True, Predicted True
0.87 0.70 - result 0.04 - Expected False, Predicted False
0.90 0.64 - result 0.01 - Expected False, Predicted False

```

Figura 1: Resultados do teste realizado.

3 GGClassification

A equipe do laboratório de Inteligência Artificial da escola de engenharia desenvolveu uma biblioteca em R[2] disponível na plataforma CRAN[1] para a implementação do grafo de Gabriel chamada GGClassification¹.

A biblioteca foi implementada na linguagem C++ e utiliza a biblioteca Rcpp² para criar uma interface entre R e C++. O código fonte da biblioteca pode ser encontrado no git:

https://github.com/cran/GGClassification/blob/master/src/GGClassification_model.cpp

O código em C++ utiliza funções de álgebra linear Eigen.h³ e para exportar dados para o R utiliza a biblioteca Rcpp.

4 Próximos Passos

No momento estou me dedicando em criar maior familiaridade com a biblioteca GGClassification em R. Estou tentando rodar exemplos da aplicações com a biblioteca. Ainda estou me adaptando com a sintaxe da linguagem R, visto que estive estudando em python.

Em seguida, devo partir das funções em C++ da biblioteca GGClassification para implementar em firmware para o ESP32. Testes serão feitos para avaliar a viabilidade do algoritmo nesse microcontrolador.

Após validado exemplos do grafo de Gabriel no ESP32 podemos começar a integrar o sensor acelerômetro e fazer o algoritmo para padrões de movimentos. Por fim, caso tudo funcione finalizamos com a montagem do protótipo.

¹<https://CRAN.R-project.org/package=GGClassification>.

²<https://github.com/RcppCore/Rcpp>.

³https://eigen.tuxfamily.org/index.php?title=Main_Page.

Referências

- [1] Jan Philipp Dietrich e Waldir Leoncio. *citation: Software Citation Tools*. R package version 0.6.2. 2022.
- [2] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2021. URL: <https://www.R-project.org/>.
- [3] Wikipedia contributors. *Gabriel graph* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 12-May-2022]. 2022. URL: https://en.wikipedia.org/w/index.php?title=Gabriel_graph&oldid=1087461346.