UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO
CURSO DE ENGENHARIA DE COMPUTAÇÃO

RELATÓRIO DA AVALIAÇÃO - REPOSIÇÃO

MODELO DE MACHINE LEARNING PARA EMBARCADOS

Aylton Correia Gomes Dias: 202000000572

Natal-RN
2022

Aylton Correia Gomes Dias: 202000000572

Relatório apresentado à disciplina de Sistemas Digitais, correspondente à avaliação da 3º unidade do semestre 2022.1 do 7º período do curso de Engenharia de Computação e Automação da Universidade Federal do Rio Grande do Norte, sob orientação do **Prof. Ignacio Sanchez Gendriz.**

Professor: Ignacio Sanchez Gendriz.

Natal-RN
2022

RESUMO

O presente relatório tem como objetivo apresentar a metodologia, desenvolvimento e resultados obtidos a partir da resposta solicitada como forma de avaliação para a disciplina de Sistemas Digitais. A proposta solicitada é o desenvolvimento de um modelo de Machine Learning que possa ser utilizado em sistema embarcado, como o Arduino UNO.

Palavras-chave: Modelo computacional, Arduino UNO, Machine Learning

Sumário

1 INTRODUÇÃO

O presente relatório descreve brevemente o processo de desenvolvimento de um modelo de machine learning para sistemas embarcados. O modelo de Machine Learning foi desenvolvido com o auxílio da plataforma Colab e linguagem de programação Python e o projeto embarcado foi adaptado para uso dentro da plataforma KinterCad e construído usando componentes básicos e linguagem de programação AVR.

2 PROBLEMA

Treine uma rede neural que permite decodificar os níveis de voltagem a serem aplicados a um display de 7 segmentos e que apresenta como saída o código BCD correspondente a entrada.

A entrada da rede proposta deverá ser um vetor de dimensão 7, onde cada elemento do vetor de entrada representa um nível de voltagem (valores reais a faixa de 0 - 5 volts), os valores de 0 - 0,8v representam 0 lógico, já os valores de 2 - 5v representam 1 lógico.

Tabela 1 - Relação entre valores de entrada real e nível lógico

Valores	Nível lógico
0 - 0,8V	0 lógico
2 - 5V	1 lógico

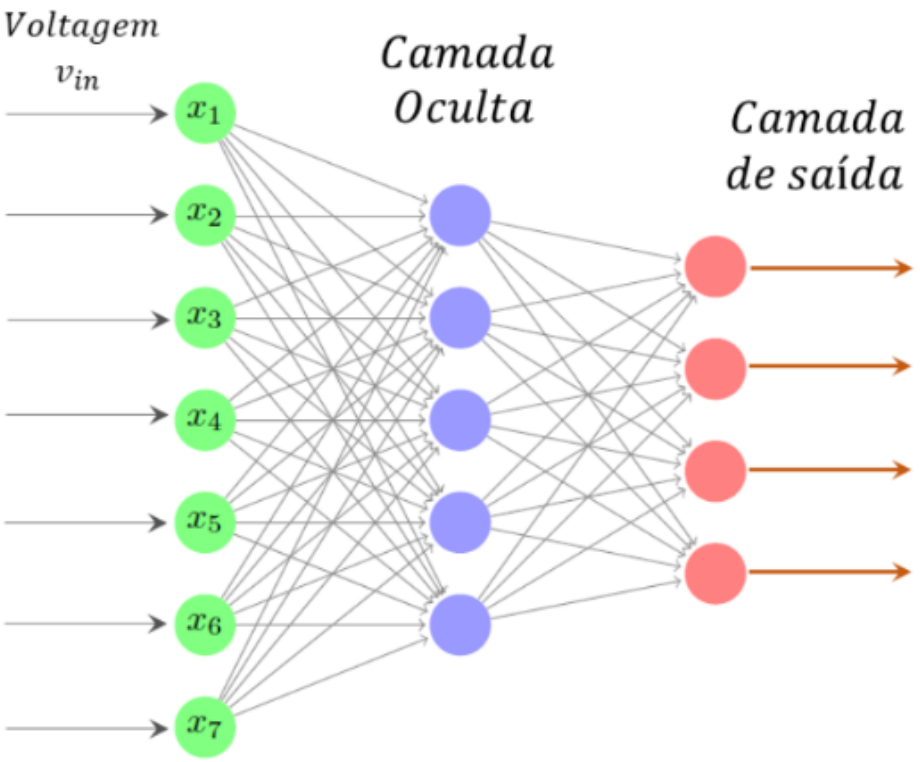
A saída da rede neural deverá ser um vetor de 4 bits em código BCD.

Figura 1 - Saída da rede em código BCD

Decimal	Binay (BCD) 8 4 2 1
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Para a rede neural em específico, recomenda-se o uso de 5 - 20 neurônios na camada oculta da rede neural implementada, como ilustrado na figura abaixo.

Figura 2 - Ilustração da arquitetura de uma rede neural.



Exemplo de relação relação esperada entre entrada - saída para 4 casos diferentes.

Figura 3: Exemplo de Entrada e Saída

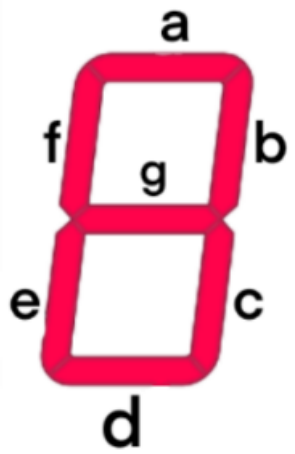
a) Seleção de X_In

	G	F	E	D	C	B	A
0	4.231242	0.780309	4.730696	2.156199	0.645430	3.787909	2.969870
1	0.636853	2.024109	2.090387	2.159703	3.502169	2.557124	3.135354
2	2.986594	0.456140	0.779718	3.521068	2.736177	3.317250	2.819513
3	0.038345	4.963844	3.928005	2.248092	3.118439	2.997237	4.477473
4	0.161174	0.377934	0.603821	0.664421	3.294831	4.219109	0.200306

b) Seleção de Y_Out

	D8	D4	D2	D1
0	0	0	1	0
1	0	0	0	0
2	1	0	0	1
3	0	0	0	0
4	0	0	0	1

c)



ESPECIFICAÇÕES GERAIS PARA DESENVOLVIMENTO DO MODELO DE ML:

- A rede deverá ser treinada usando o software de sua preferência (por exemplo Python);
- Para treinamento da rede, use a base de dados disponibilizada em dois arquivos '.csv';
- O arquivo ***X_In.csv*** contém as entradas a serem usadas na rede.
- O arquivo ***Y_Out.csv*** contém as saídas esperadas para cada instância de entrada;
- Separe uma parte dos dados para treino (exemplo 80%), e o restante dos dados deverá ser usado para testar o modelo.

ESPECIFICAÇÕES GERAIS PARA DESENVOLVIMENTO DO MODELO EM HARDWARE:

- Após treinamento da rede, a mesma deverá ser implementada no Arduino Uno;
- Use para essa finalidade a plataforma online Tinkercad;
- Após implementação da rede treinada no Arduino Uno, o modelo deverá ser testado usando um subconjunto dos dados de teste.

3 METODOLOGIA

Inicialmente, selecionamos as ferramentas que seriam usadas para desenvolver o modelo de machine learning. Por questões de praticidade, foi decidido usar a plataforma Colab e sua linguagem base, o Python.

Logo em seguida, foi necessário revisar alguns conceitos básicos, da linguagem e bibliotecas específicas que poderiam ser usadas para implementação dos primeiros modelos.

ETAPA DE DESENVOLVIMENTO DOS MODELOS EM PYTHON

Dentro do Colab, seguimos os seguintes passos:

Modelo 1:

- Importação das bibliotecas necessarias;

```
# importação das bibliotecas necessarias
import pandas as pd
import numpy as np
import tensorflow as tf

from sklearn.model_selection import train_test_split
from tensorflow import keras
```

- Comunicação com o Google Drive;

```
from google.colab import drive
drive.mount('/content/drive')
```

- Importação dos arquivos .csv para dentro do ambiente do Colab;

```
x_in = pd.read_csv('/content/drive/MyDrive/dados/X_In.csv')
y_out = pd.read_csv('/content/drive/MyDrive/dados/Y_Out.csv')
```

- Divisão das amostras entre dados de treino e dados de teste;

```
from sklearn.model_selection import train_test_split
x_treino, x_teste, y_treino, y_teste = train_test_split(x_in, y_out, test_size = 0.30)
```

- Importação do algoritmo de machine learning com auxílio da biblioteca Keras.

```
#Cria o modelo usando a biblioteca Keras

hidden_size = 5

model = keras.Sequential([
  keras.layers.InputLayer(input_shape=(7,)),
  keras.layers.Dense(hidden_size, activation = 'tanh'),
  keras.layers.Dense(4, activation = 'sigmoid'),
])

model.summary()
```

- Compilação do modelo

```
#Compila o modelo
model.compile(optimizer='adam', loss=tf.keras.losses.BinaryCrossentropy(from_logits=True) , metrics=['binary_accuracy'])
```

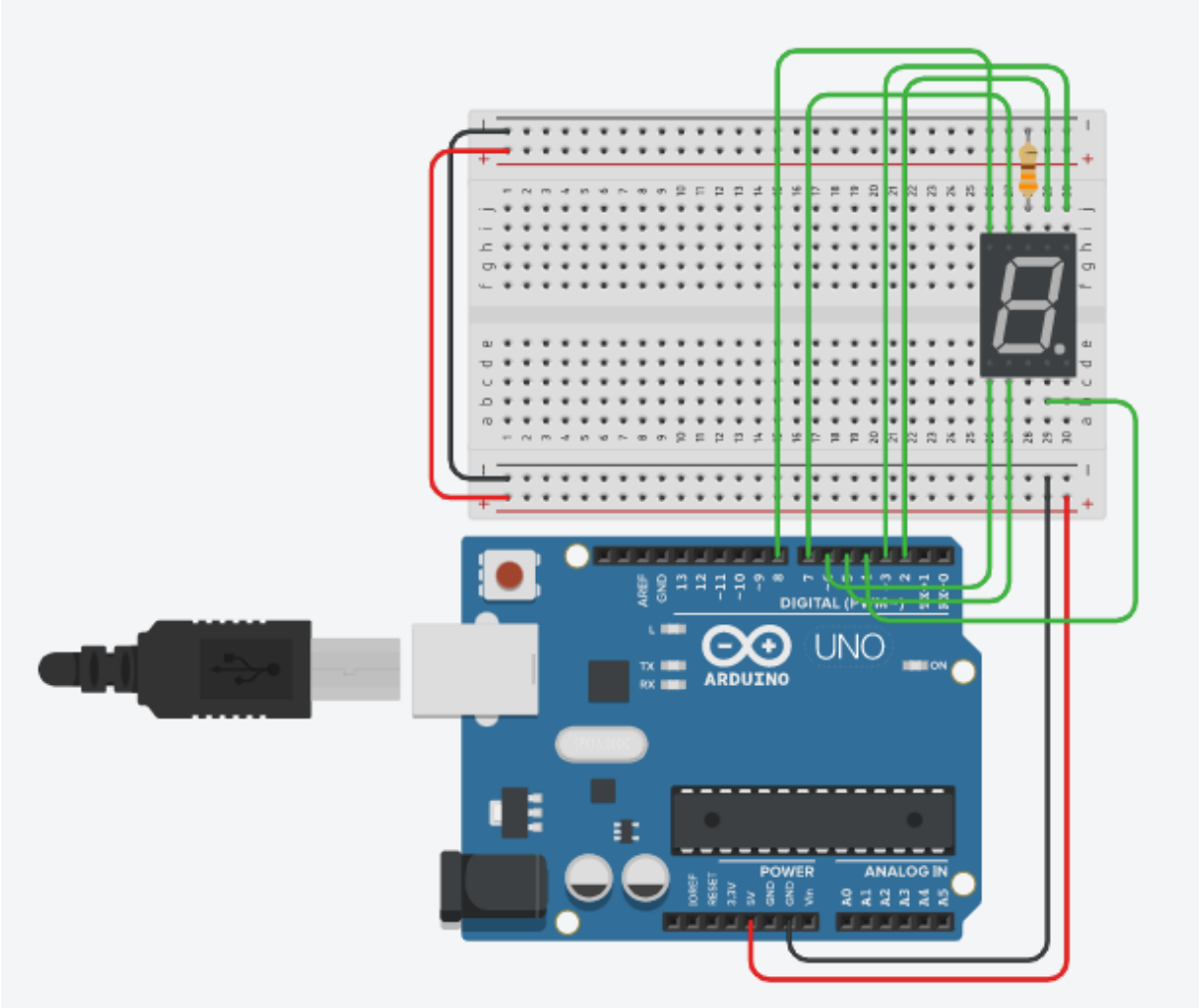
- Treino e teste do modelo.

```
num_epochs = 200
#Realiza um treinamento e teste do modelo com num_epochs definido
result = model.fit(x_treino, y_treino, epochs=num_epochs, batch_size = 32, validation_data=(x_teste, y_teste))
```

ETAPA DE DESENVOLVIMENTO DOS MODELOS EM EMBARCADO

Com o auxílio da plataforma TinkerCad, elaboramos um modelo de machine learning para teste de entradas singulares.

Com as informações obtidas na etapa anterior, construímos matrizes de pesos e bias para nossa estrutura neural. Esse processo foi necessário pela limitação de processamento e armazenamento do TinkerCad. Ao importar esses dados para a plataforma, foi possível elaborar uma rede neural similar, em um sistema embarcado.



Link para a representação do projeto em Hardware:
https://www.tinkercad.com/things/6CJpyCqTBAV-ingenious-densor/editel?sharecode=sb8N3iHaaGGDNCWzFMWGJ90Tkm-YWMpwSDfTHqJYqPs

Link para a representação do projeto no Colab:
https://colab.research.google.com/drive/1wOSj806OXcsAl5dqcz9DV6oKgQpmH22n?usp=sharing

4 CONCLUSÃO

Esses experimentos, foram de suma importância para a compreensão dos conceitos teóricos e práticos, trabalhados até o momento, abordados na disciplina de Sistemas Digitais. Ajudou a compreender melhor a dinâmica da rotina de um profissional da área, com um casos práticos que se assemelha às aplicações do cotidiano. Também ajudou a entender o princípio de funcionamento de modelos de Machine Learning.

Em relação aos resultados específicos, foi possível concluir que pode-se implementar um modelo embarcado. Entretanto, existem limitações relacionadas ao processamento e armazenamento de determinados dispositivos. O tempo de execução também teve uma variação expressiva de uma plataforma para outra e esse fato ocorreu mediante as limitações da plataforma TinkerCad e facilidades/bibliotecas específicas da plataforma Colab e linguagem Python.

Referências

Hobbizine. **A neural network for arduino**. Disponível em:<http://robotics.hobbizine.com/arduinoann.html/>. Acesso em: 17 de dez de 2022