PROJETO DA UNIDADE 3
IMPLEMENTANDO UMA REDE NEURAL EMBARCADA

DISCENTES:
EFRAIN MARCELO PULGAR PANTALEON
MATHEUS GOMES DINIZ ANDRADE

DOCENTE:
IGNACIO SANCHEZ GENDRIZ

DOCÊNCIA ASSISTIDA:
MATEUS ARNAUD SANTOS DE SOUSA GOLDBARG

NATAL/RN
2022

1. INTRODUÇÃO

Um sistema digital é um sistema em que a representação dos sinais é feita através de um número finito de valores. Diversas aplicações são dadas aos sistemas digitais como a implementação de sistemas embarcados. Sistemas embarcados possuem capacidade de processar dados e desempenhar uma função específica.

O presente trabalho tem como objetivo desenvolver um sistema de uma rede neural que nos permita decodificar os níveis de tensões a serem aplicados num display 7 segmentos e que apresenta como saída o código BCD correspondente à entrada. A rede neural desenvolvida será aplicada dentro de um sistema embarcado, que no nosso caso contamos com o Arduino Uno R3 e seu microcontrolador ATMega328p.

2. METODOLOGIA

Para realização do projeto foi necessário o auxílio de ferramentas como a linguagem de programação Python e algumas bibliotecas implementadas nessa linguagem, bem como o software de simulação online Tinkercad.

O Tinkercad é responsável por simular o comportamento do Arduino Uno R3 e seu microcontrolador ATMega328p. Para isso foram escritos códigos na linguagem de programação C. O circuito simulado e o código podem ser acessados através do link https://www.tinkercad.com/things/7uR7SraE344-copy-of-atividadecontador1efrain/editel?sharecode=QB02mOrlSdc7cIx6QD3qwLUXfEhD5EOfpAH7KBa2wiM

A exploração dos dados bem como o treinamento da rede foram feitos com auxílio do Google Collaboratory que usa uma interface para o Jupyter Notebook. Os códigos em python podem ser acessados através do seguinte link https://colab.research.google.com/drive/1ZboXRMG3OjJL-oBTlfFR6j_UFN2xBlrh?usp=sharing

3. TREINAMENTO

Iniciando pela parte do Aprendizado de Máquina, utilizamos de bibliotecas como Keras do Tensor Flow, SKLearn para montarmos nossa Rede Neural Artificial. As plotagens dos gráficos foram feitas com auxílio da biblioteca MatPlotLib.

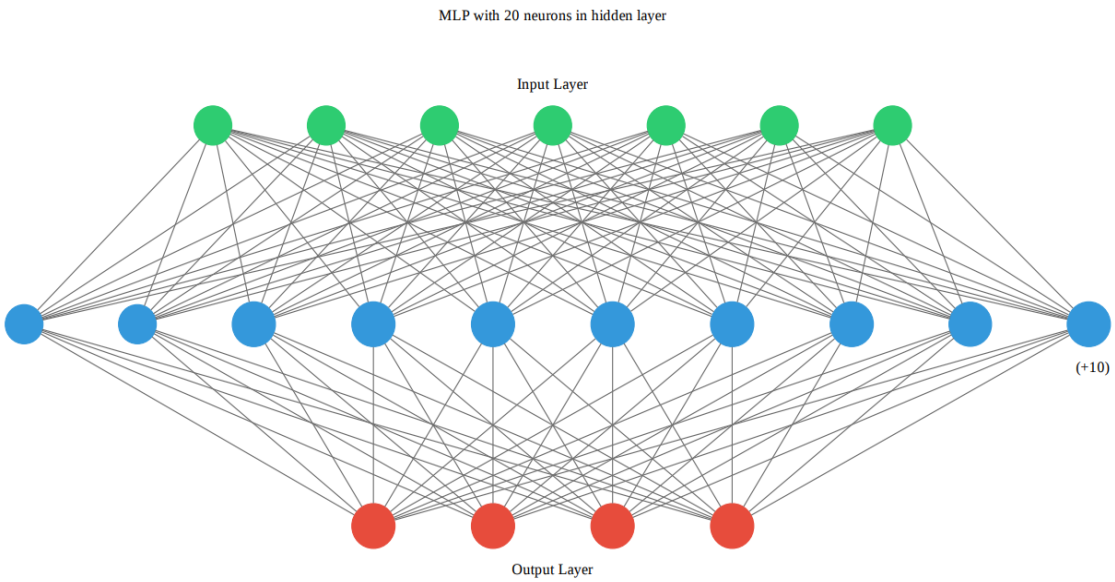
Os dados para usarmos de treino e de teste da nossa rede, foram disponibilizados pelo Docente através dos seguintes links:

- Dados de entrada : https://drive.google.com/file/d/193byUZkQFEhfeI8UJbTXZfICdbREaw

hk/view

- Dados de Saída: https://drive.google.com/file/d/11uoXGKni-fioF748Y6iDVG29VmXz_cxW/view

Esses dados foram particionados em uma razão de 80% para treinamento e 20% para realizar o teste da eficiência da nossa rede. Para a criação do nosso modelo de rede utilizamos 7 neurônios para dados de entrada combinadas de uma camada escondida contendo 20 neurônios ativadas pela função da tangente hiperbólica.Para a camada de saída temos 4 neurônios ativados pela função de sigmoid. A representação gráfica da rede é descrita abaixo:

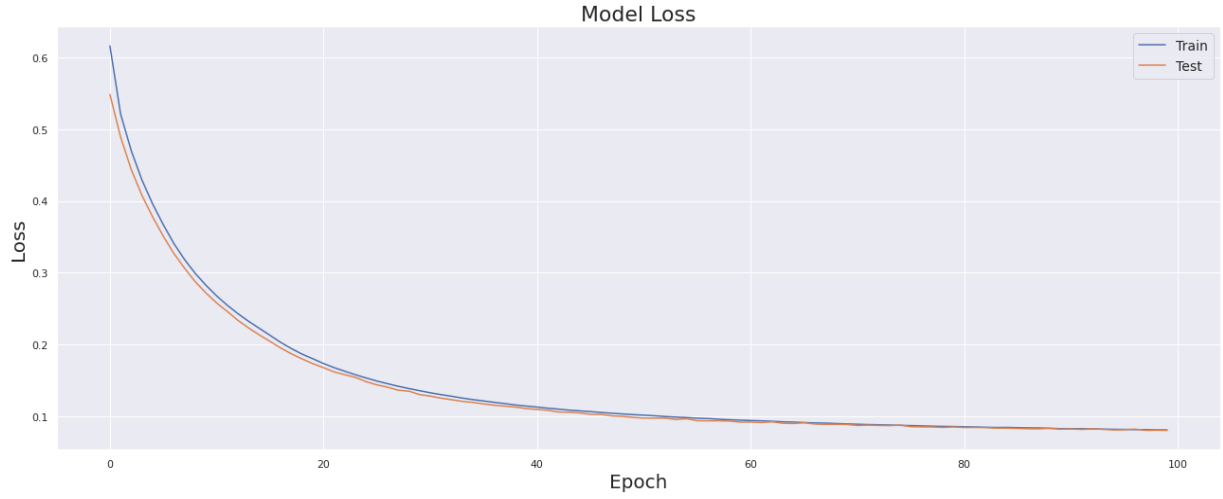


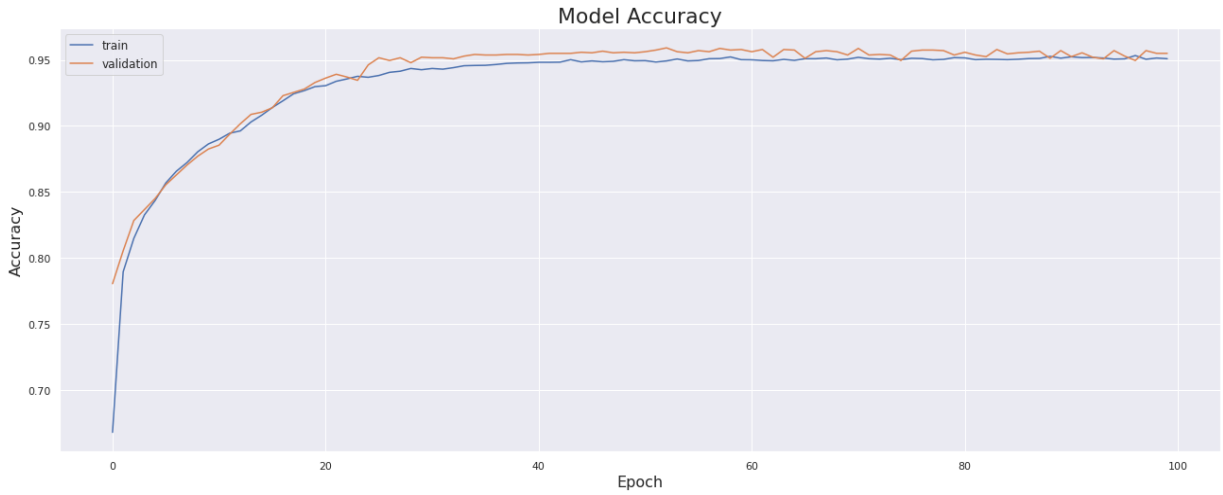
A montagem do modelo pode ser feita dentro do python seguindo a seguinte estrutura abaixo:

```
model = keras.Sequential([

keras.layers.InputLayer(input_shape=(7,)),

keras.layers.Dense(20, activation = 'tanh'),

keras.layers.Dense(4, activation = 'sigmoid')])
```

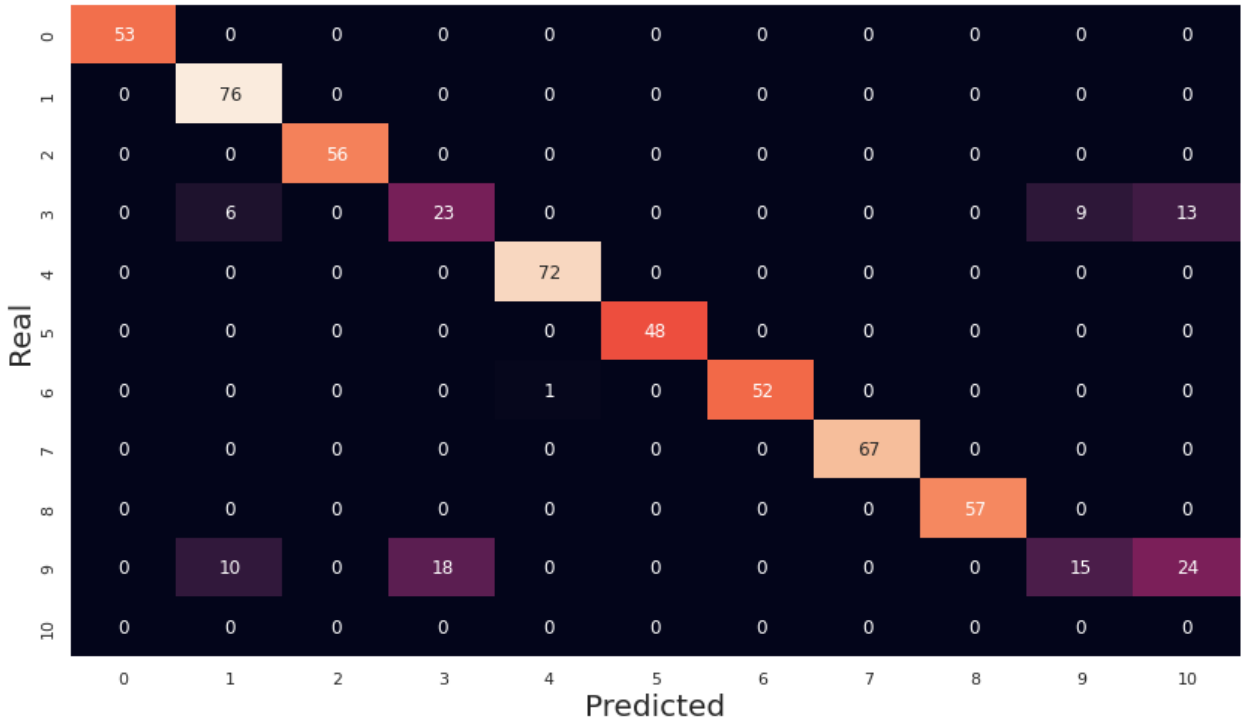
Foi feito o treinamento da rede utilizando como função de perda a Binary Crossentropy disponível na biblioteca Keras e utilizado como métrica a função Binary Accuracy também disponível na biblioteca Keras.

Abaixo é possível observar o desempenho do modelo ao longo de 100 épocas de treinamento.





Como foi possível observar, o modelo obteve um bom desempenho com cerca de 95% de acurácia. Usando a base de testes, podemos analisar o desempenho de outra maneira por meio de matriz de confusão.



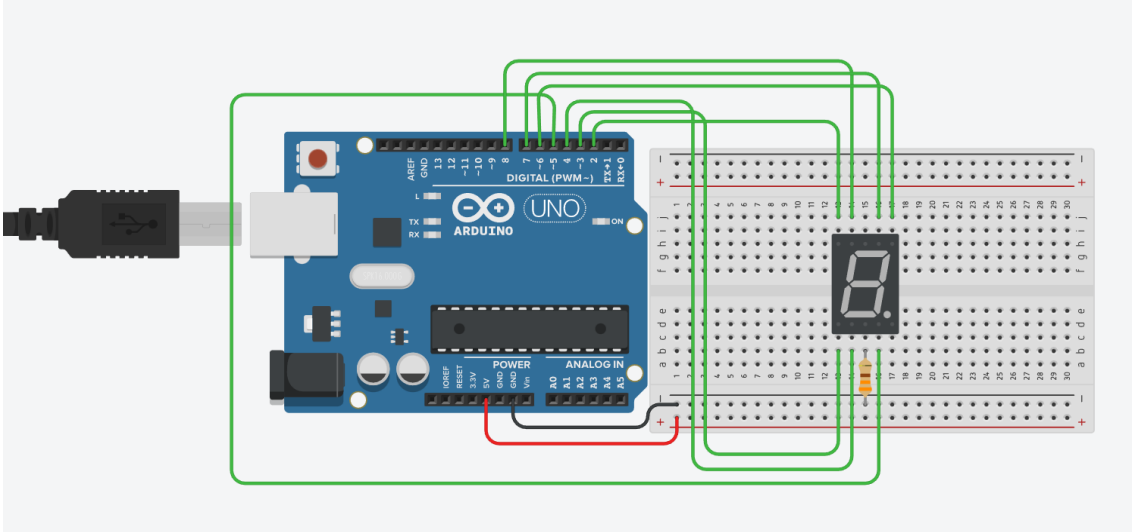
É relevante comentar que na base de dados não possuímos valores maiores que 9. No entanto com 4 bits é possível representar os valores de 0 a 15 e por isso o modelo previu o valor 10.

4. MONTAGEM

Para a montagem do circuito proposto no projeto, foi utilizado:

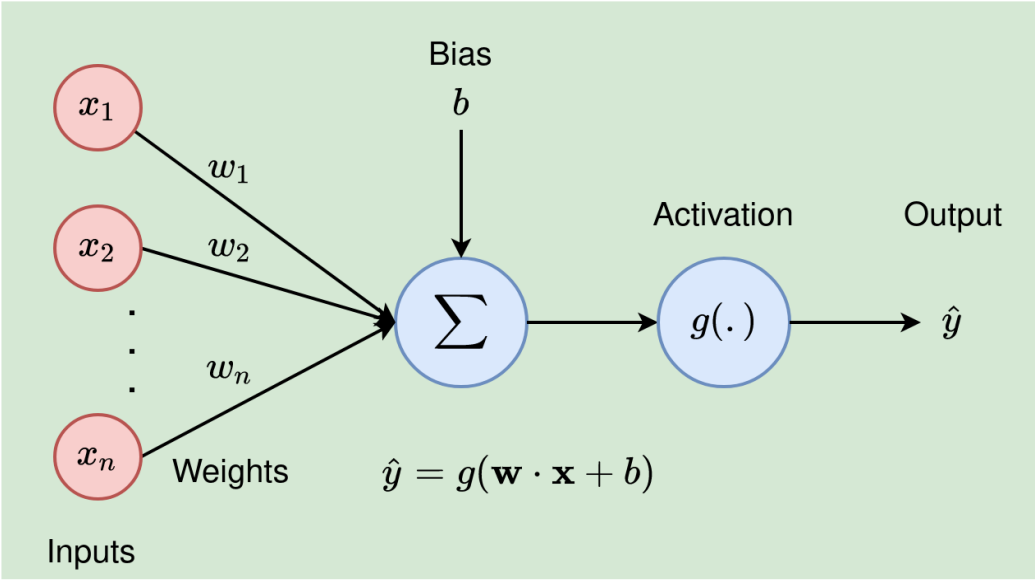
- 1 Arduino Uno R3
- 1 Visor de Sete Segmentos
- 9 Jumpers
- 1 Resistor de 330 Ohm

A montagem do sistema pode ser visualizada na seguinte imagem:



5. IMPLEMENTAÇÃO

Para que fosse possível embarcar a rede no Tinkercad foi necessário extrair os pesos gerados pelo aprendizado da rede durante o treinamento no Google Collaboratory. Com isso, foram implementadas operações de multiplicação e soma de matrizes bom como aplicação das funções de ativação Tanh e Sigmoid presentes na biblioteca do Keras



Fonte:https://towardsdatascience.com/the-basics-of-neural-networks-neural-network-series-part-1-4419e343b2b

6. CONCLUSÃO

Diante do exposto, é possível concluir que pode-se implementar um modelo embarcado. No decorrer do trabalho não foram discutidos tópicos que quando tratamos de microcontroladores são de alta relevância como o espaço de armazenamento. Isso se deu devido estarmos tratando de um modelo simples e relativamente pequeno.

7. REFERÊNCIAS

- *Subir Maity - Train and Test a Neural Network in Arduino Uno*
- *https://www.python.org/*
- *https://www.tensorflow.org/*
- *https://keras.io/*
- *https://matplotlib.org/*
- *https://seaborn.pydata.org/*
- *https://scikit-learn.org/stable/*
- *https://pypi.org/project/ann_visualizer/*
- *https://numpy.org/*
- *https://pandas.pydata.org/*