



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA  
GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO

---

**Análise de *bias* e de superposição em aprendizagem de  
máquina quântica**  
PROPOSTA DE TRABALHO DE GRADUAÇÃO

**Aluno:** Emmanuel Carreira Alves (eca2@cin.ufpe.br)  
**Orientador:** Adenilton Jose da Silva (ajsilva@cin.ufpe.br)  
**Área:** Computação Quântica

14 de junho de 2021

## Resumo

---

A computação quântica começa a mostrar seu potencial com os primeiros dispositivos quânticos físicos de baixo ruído disponibilizados para o público geral realizar experimentos. Com isso, surgem oportunidades de portar algoritmos e soluções da computação clássica, avaliar seu sucesso e melhorá-los nesse novo paradigma que é a computação quântica.

Dentre algumas das áreas com crescimento recente e ainda um quanto embrionária no paradigma da computação quântica está a de aprendizagem de máquina. Este trabalho visa replicar algumas das soluções existentes de aprendizagem de máquina no paradigma quântico e melhorá-las.

## Introdução

---

Com o recente crescimento da aprendizagem de máquina e da possibilidade de utilizar dispositivos quânticos físicos para realizar experimentos, surge o interesse de descobrir até onde o novo paradigma da computação quântica traz benefícios em relação à computação clássica. Também traz-se questionamentos sobre como utilizar a computação quântica para treinar modelos de aprendizagem de máquina. Dispositivos quânticos são mais eficazes que dispositivos clássicos no treinamento de um modelo? Os dispositivos quânticos atuais são robustos o suficiente para replicar o treinamento e classificação de um modelo sem que os ruídos inerentes desses dispositivos interfiram na acurácia? Quais os artifícios exclusivos do paradigma quântico utilizados para melhorar o desempenho e acurácia de tais modelos? Quais artifícios da aprendizagem de máquina da computação clássica são replicados no paradigma quântico.

Esses e outros questionamentos foram abordados por (GRANT; BENEDETTI; CAO; HALLAM; LOCKHART; STOJEVIC; GREEN; SEVERINI, 2018) em seu trabalho “Hierarchical Quantum Classifiers”[1]. Nele os pesquisadores replicaram o treinamento e classificação de modelos de aprendizagem de máquina em dispositivos quânticos, tanto com simulações quanto em um dispositivo quântico real. Foram um dos primeiros modelos de aprendizagem de máquina quântica, com a replicação dos modelos TTN e MERA clássicos para dispositivos quânticos.

Entretanto, algumas questões permanecem em aberto. Com a possibilidade de avaliar modificações na arquitetura dos modelos proposto, este Trabalho de Graduação baseia-se nos resultados obtidos por (GRANT; BENEDETTI; CAO; HALLAM; LOCKHART; STOJEVIC; GREEN; SEVERINI, 2018) para propor melhorias e experimentá-las.

## Objetivos

---

Este Trabalho de Graduação propõe-se a replicar os resultados obtidos por (GRANT; BENEDETTI; CAO; HALLAM; LOCKHART; STOJEVIC; GREEN; SEVERINI, 2018) como ponto de partida. Os modelos de classificadores replicados são o *tree tensor network* (TTN) e *multi-scale entanglement renormalization ansatz* (MERA). Cada um dos modelos é treinado sobre o conjunto de dados de variantes da planta íris[2]; o conjunto de dados de imagens de dígitos escritos à mão MNIST[3]; e um conjunto de estados quânticos sintéticos.

Para agregar ao trabalho de (GRANT; BENEDETTI; CAO; HALLAM; LOCKHART; STOJEVIC; GREEN; SEVERINI, 2018), busca-se replicar a técnica de aprendizagem de máquina de enviesamento (*bias*) nos modelos. Há de replicar tal técnica e experimentar a melhora ou piora da acurácia dos modelos em um dispositivo quântico simulado. Também propõe-se investigar se a superposição de *bits* quânticos, uma característica de dispositivos quânticos, intervém na acurácia e tempo de treinamento dos modelos.

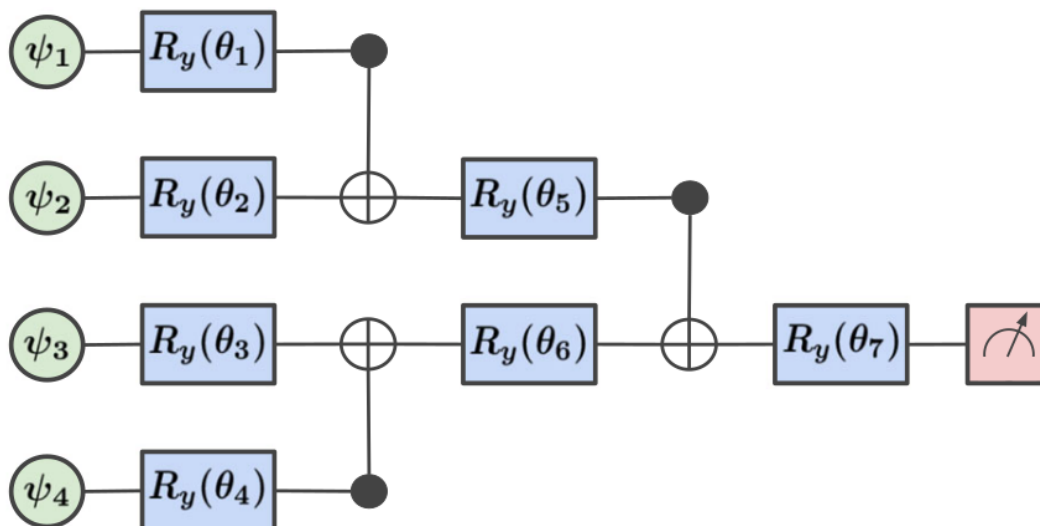
## Metodologia

---

A linguagem de programação utilizada nos experimentos é Python. Utiliza-se o *SDK* Qiskit para montar os circuitos quânticos dos modelos. Utiliza-se a biblioteca Pennylane para auxiliar no treinamento dos parâmetros  $\Theta$  do circuito quântico. Utiliza-se o Jupyter Notebook para melhor organização e disposição do código fonte. Para a criação de gráficos utiliza-se a biblioteca matplotlib.

No momento da escrita dessa proposta, o modelo TTN no conjunto de dados Iris foi replicado. O classificador TTN foi utilizado em uma classificação binária. Como o conjunto de dados Iris possui três classes, criou-se um classificador TTN para cada par de classes. O primeiro classificador para classificação binária entre as classes “Setosa” e “Versicolor”; o segundo classificador para as classes “Versicolor” e “Virginica”; o terceiro classificador para as classes “Setosa” e “Virginica”. O circuito quântico do classificador TTN está disposto na Figura 1. É composto de operadores de rotação sobre o eixo Y (operador  $R_y$ ) sobre um *qubit* e o operador CNOT. Ao final há uma medição para obter um número real a fim de estimar a classe resultante da classificação.

**Figura 1:** Circuito quântico do classificador TTN



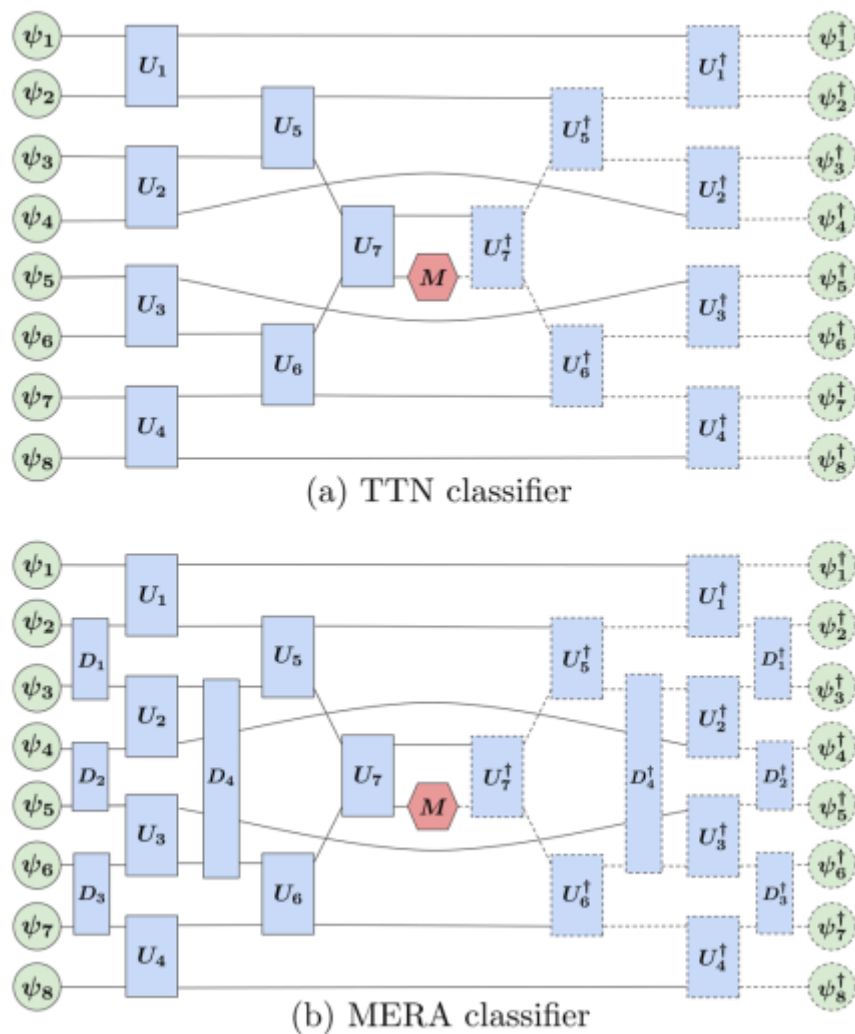
Fonte: Fig. 6 do artigo “Hierarchical quantum classifiers”[1]

Separou-se o conjunto de dados em dois *datasets*, um de treinamento e um de teste. Cada classe possui 50 amostras distintas. Separou-se 40 amostras para treinamento e 10 amostras de teste para cada classe. O *dataset* de teste corresponde então a 20% das

amostras totais. A inicialização dos parâmetros  $\Theta$  foi fixada através do uso de uma constante na semente randômica. O treinamento ocorre a fim de otimizar os parâmetros utilizados nas rotações. A técnica utilizada no treinamento foi a de gradiente descendente estocástico. A função de objetivo utilizada no treinamento foi a de erro médio quadrático, a fim de minimizar o erro entre a predição da classe da amostra e a real classe da amostra.

A replicação do classificador MERA será baseada no classificador TTN. Conforme visto na Figura 2, a topologia do classificador MERA é semelhante à do classificador TTN. A diferença entre as duas topologias é a presença de duas camadas a mais compostas de operadores sobre dois *qubits*  $D_i$ . Será utilizado o operador *QubitUnitary*[4] do PennyLane para implementação de tais operadores  $D_i$ .

**Figura 2:** Topologia dos classificadores TTN (a) e MERA (b)



Fonte: Fig. 1 do artigo "Hierarchical quantum classifiers"[1]

A replicação dos modelos no conjunto de dados MNIST será similar à replicação no conjunto de dados Iris. A diferença está na importação do conjunto de dados. Enquanto no Iris tem-se as características de cada exemplo de treinamento e teste como valores reais, no MNIST tem-se imagens 28x28 pixels. Cada um dos pixels está em escala de cinza. A importação de tais imagens para simulação no circuito quântico será análoga à de (GRANT; BENEDETTI; CAO; HALLAM; LOCKHART; STOJEVIC; GREEN; SEVERINI, 2018), com a análise dos componentes principais da imagem e extração dos oito componentes de maior variância.

A replicação dos modelos no conjunto de estados quânticos sintéticos será feita com o auxílio de um circuito quântico gerador de estados quânticos. Tal circuito quântico será composto por operadores de rotação sobre um *qubit* e o operador CNOT. Será sempre aplicado sobre o estado inicial “ket 0”. Ao aleatorizar as rotações dos operadores, é possível gerar um conjunto de estados quânticos. A saída do circuito quântico gerador alimentará os *qubits* de entrada do circuito quântico do classificador. Isso é possível pois a saída do circuito quântico gerador é um estado quântico. Assim, não há necessidade de codificação das características para um estado quântico.

Será possível investigar o efeito do *bias* nos classificadores com a replicação bem-sucedida dos classificadores TTN e MERA. O *bias* será feito com a adição de um deslocamento (*offset*) aleatório em cada um dos operadores de rotação sobre um *qubit* do circuito quântico. O efeito do *bias* na acurácia do classificador será mensurado ao aleatorizar o *offset* de cada operador entre sessões de treinamento.

A influência da superposição de *qubits* na acurácia do modelo será avaliada através da codificação dos *qubits*. Para isso será utilizada algumas formas de codificação[5] disponibilizadas pelo Pennylane. As formas de codificação a serem experimentadas nesse Trabalho de Graduação serão *AngleEmbedding*[6] e *AmplitudeEmbedding*[7]. *AngleEmbedding* permite codificar N características em N *qubits*. *AmplitudeEmbedding* permite codificar  $2^N$  características em N *qubits* ao fazer uso da superposição de *qubits*.

## Cronograma

---

Atividade	Período			
	Maio	Junho	Julho	Agosto
Estudo do artigo	X			
Replicação do TTN	X	X		
Replicação do MERA		X		
Implementação da superposição		X		
Implementação do bias			X	
Obtenção dos resultados		X	X	
Escrita do TG		X	X	X
Preparação da apresentação				X

OBS: o desenvolvimento do trabalho começou na disciplina de Computação Quântica no período de 2020.1



## Referências

---

- [1] GRANT, Edward; BENEDETTI, Marcello; CAO, Shuxiang; HALLAM, Andrew; LOCKHART, Joshua; STOJEVIC, Vid; GREEN, Andrew G.; SEVERINI, Simone. Hierarchical quantum classifiers. *Npj Quantum Information*, [S.L.], v. 4, n. 1, p. 65-65, 17 dez. 2018. Springer Science and Business Media LLC. <http://dx.doi.org/10.1038/s41534-018-0116-9>. Disponível em: <https://www.nature.com/articles/s41534-018-0116-9>. Acesso em: 07 jun. 2021.
- [2] FISHER, Ronald. Iris Data Set. 1936. Doado por Michael Marshall. Disponível em: <https://archive.ics.uci.edu/ml/datasets/iris>. Acesso em: 07 jun. 2021.
- [3] LECUN, Yann; CORTES, Corinna; BURGESS, Christopher J.C.. THE MNIST DATABASE: of handwritten digits. Disponível em: <http://yann.lecun.com/exdb/mnist/>. Acesso em: 07 jun. 2021.
- [4] XANADU. Qml.QubitUnitary. Disponível em: <https://pennylane.readthedocs.io/en/stable/code/api/pennylane.QubitUnitary.html>. Acesso em: 09 jun. 2021.
- [5] XANADU. Quantum embedding. Disponível em: [https://pennylane.ai/qml/glossary/quantum\\_embedding.html](https://pennylane.ai/qml/glossary/quantum_embedding.html). Acesso em: 09 jun. 2021.
- [6] XANADU. Qml.templates.embeddings.AngleEmbedding. Disponível em: <https://pennylane.readthedocs.io/en/stable/code/api/pennylane.templates.embeddings.AngleEmbedding.html>. Acesso em: 09 jun. 2021.
- [7] XANADU. Qml.templates.embeddings.AmplitudeEmbedding. Disponível em: <https://pennylane.readthedocs.io/en/stable/code/api/pennylane.templates.embeddings.AmplitudeEmbedding.html>. Acesso em: 09 jun. 2021.

## **Possíveis Avaliadores**

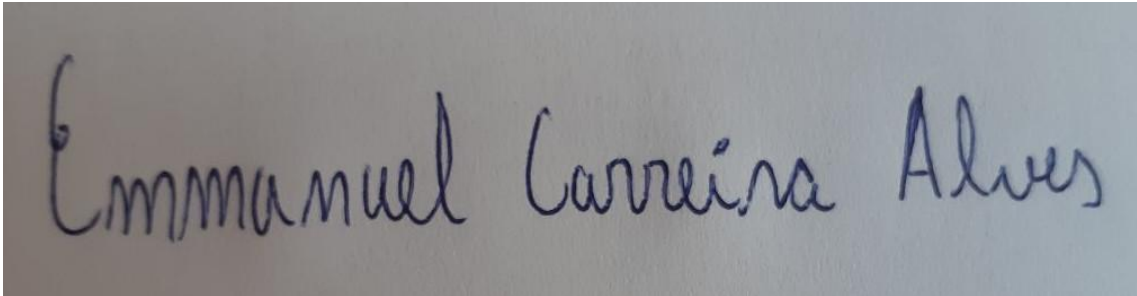
---

Professor Fernando Maciano de Paula Neto ([fmpn2@cin.ufpe.br](mailto:fmpn2@cin.ufpe.br))

## Assinaturas

---

Recife, 14 de junho de 2021

A photograph of a handwritten signature in blue ink on a light-colored background. The signature reads "Emmanuel Carreira Alves".

Emmanuel Carreira Alves

(Aluno)

A photograph of a handwritten signature in black ink. The signature reads "Adenilton Jose da Silva".

Adenilton Jose da Silva

(Orientador)