UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

EMMANUEL CARREIRA ALVES

ANÁLISE DA PREPARAÇÃO DE ESTADO QUÂNTICO POR CODIFICAÇÃO POR AMPLITUDE EM APRENDIZAGEM DE MÁQUINA QUÂNTICA

RECIFE

2021
UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

EMMANUEL CARREIRA ALVES

ANÁLISE DA PREPARAÇÃO DE ESTADO QUÂNTICO POR CODIFICAÇÃO POR AMPLITUDE EM APRENDIZAGEM DE MÁQUINA QUÂNTICA

Monografia apresentada ao Centro de Informática (CIN) da Universidade Federal de Pernambuco (UFPE), como requisito parcial para conclusão do Curso de Engenharia da Computação, orientada pelo professor Adenilton Jose da Silva.

RECIFE

2021

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

EMMANUEL CARREIRA ALVES

ANÁLISE DA PREPARAÇÃO DE ESTADO QUÂNTICO POR CODIFICAÇÃO POR AMPLITUDE EM APRENDIZAGEM DE MÁQUINA QUÂNTICA

Monografia submetida ao corpo docente da Universidade Federal de Pernambuco, defendida e aprovada em 20 de Agosto de 2021.

Banca Examinadora:

_____ Orientador

Adenilton Jose da Silva

Doutor

_____ Examinador

Fernando Maciano de Paula Neto

Doutor

À minha mãe, que me incentivou e me amparou nos momentos difíceis.

AGRADECIMENTOS

Agradeço à minha família pelo apoio durante minha jornada no curso. Agradeço ao Prof. Dr. Adenilton Jose por me acompanhar neste trabalho e pelo suporte fornecido. E expresso minha gratidão por todos os professores e professoras do Centro de Informática que compartilharam comigo um pouco de seus conhecimentos.

“Existem muitas hipóteses em ciência que estão erradas. Isso é perfeitamente aceitável. Elas são a abertura para achar as que estão certas.”
Carl Sagan

RESUMO

A computação quântica se mostra um interessante campo de estudo conforme se aproxima da era de dispositivos quânticos ruidosos de escala intermediária. Surge então a possibilidade de utilizar dispositivos quânticos físicos para experimentação e descobrir benefícios de se utilizar a computação quântica face à computação clássica. Um dos campos de recente interesse é a aprendizagem de máquina em dispositivos quânticos para investigar as vantagens e desvantagens de utilizar tais dispositivos. Dito isso, o presente trabalho visa replicar resultados anteriores obtidos com os modelos hierárquicos TTN e MERA em dispositivos quânticos. Também investiga a influência na acurácia de modelos hierárquicos ao utilizar preparações de estado quântico distintas, a *Angle Embedding* e a *Amplitude Embedding*. Variações dos modelos TTN e MERA, com diferentes configurações dos operadores quânticos, foram experimentados em dois conjuntos de dados clássicos, o conjunto iris e o conjunto MNIST. Utilizou-se da simulação de dispositivos quânticos para treinar e testar os modelos hierárquicos, através do uso da biblioteca Pennylane. Os resultados dos experimentos mostraram que é possível chegar às conclusões obtidas por GRANT et al. (2018). Também comprovou a conclusão de SIERRA-SOSA et al. (2020) de que modelos com *Amplitude Embedding* obtêm acurácias maiores em comparação com modelos com inicialização *Angle Embedding*.

Palavras-chave: aprendizagem de máquina; computação quântica; aprendizagem de máquina quântica;

ABSTRACT

Quantum computing reveals itself as an interesting study field as the NISQ era approaches. The possibility of using quantum physical devices for experimentation and discovering benefits of using the quantum computing paradigm instead of classical Turing machines arises. One of the recent interesting study fields is quantum machine learning to investigate the pros and cons of using such quantum devices. Therefore, this work aims to replicate previous results obtained with hierarchical quantum classifiers TTN and MERA in quantum devices. It also investigates the influence of quantum state preparation, Angle Embedding and Amplitude Embedding, in the accuracy of hierarchical classifiers. Variations of TTN and MERA classifiers, with different quantum gates configurations, were experimented in two classical datasets, the iris dataset and the MNIST dataset. Simulation of quantum devices was used to train and test the hierarchical quantum classifiers, with Pennylane as the main library. The experiments's results show that it is possible to confirm conclusions obtained by GRANT et al. (2018). It also confirms SIERRA-SOSA et al. (2020) conclusion that classifiers with Amplitude Embedding as state preparation achieves higher accuracies in comparison with classifiers with Angle Embedding preparation.

Keywords: machine learning; quantum computing; quantum machine learning;

SUMÁRIO

TABELA DE SIGLAS

Sigla	Significado	Página
TTN	*Tree Tensor Network*	20
MERA	*Multi-scale Entanglement Renormalization Ansatz*	20
MNIST	*Modified National Institute of Standards and Technology*	20
qubit	*Quantum bit*	22
CNOT	*Controlled NOT*	24
MLP	*Multi-layer Perceptron*	29
MSE	*Mean Square Error*	29
PCA	*Principal Component Analysis*	25
NISQ	*Noisy intermediate-scale quantum*	44

1. Introdução

A aprendizagem de máquina é uma área da computação focada em resolver problemas através do reconhecimento de padrões e extração de informação a partir de dados previamente coletados (RIPLEY, 1996). E que melhora sua precisão e tomada de decisão com mais dados disponíveis e com o passar do tempo. Nos anos recentes, vimos a capacidade de utilizar aprendizagem de máquina para soluções dos mais diversos problemas, como melhores recomendações de consumo baseados nos gostos e desgostos do usuário (MOONEY; ROY, 2000); processamento de linguagem natural (COLLOBERT; WESTON, 2008), permitindo assistentes virtuais; carros autônomos com pouca ou nenhuma interferência do motorista (CHEN et al., 2015); processamento de imagens para identificação e rastreio de objetos (WANG et al., 2020) ou pessoas (CASTANHEIRA et al., 2020). Assim como soluções que superam o ser humano em algumas tarefas, como jogar o milenar jogo de tabuleiro Go (SILVER et al., 2017); analisar uma imagem e classificá-la corretamente (KRIZHEVSKY; SUTSKEVER; HINTON, 2012); realizar diagnósticos médicos (SHEN; WU; SUK, 2017) e identificar doenças em estágios iniciais (LIU et al., 2014).

Já a computação quântica (NIELSEN; CHUANG, 2000) é um paradigma de computação diferente da computação clássica das máquinas de Turing. Baseia-se nos fenômenos da física quântica para resolver problemas, de forma computacional, com uma melhor perícia ou desempenho de tempo/memória em comparação com algoritmos da computação clássica. Alguns exemplos de soluções de tais problemas computacionais são o Algoritmo de Grover para a busca em uma coleção de elementos (GROVER, 1996); e o Algoritmo de Shor, para fatoração de números não-primos (SHOR, 1997).

Ambas as áreas, aprendizagem de máquina e computação quântica, não são novas. Aprendizagem de máquina data de seu início em 1943 com o modelo de neurônio de McCulloch-Pitts (MCCULLOCH; PITTS, 1943). Computação quântica tem início em 1980 com a concepção da ideia de dispositivos quânticos por Yuri Manin e Richard Feynman (FEYNMAN, 1982). Apesar do desenvolvimento teórico em ambas as áreas, somente na última década tiveram-se mais experimentações. Na aprendizagem de máquina isso foi possível pelo avanço do poder de processamento dos computadores recentes, capazes de treinar modelos com milhares de parâmetros. E na computação quântica com o desenvolvimento dos primeiros dispositivos quânticos experimentais de baixo ruído e quantidade de bits quânticos pequena.

Com este 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?

Questionamentos semelhantes foram abordados por GRANT et al. (2018) no trabalho "*Hierarchical Quantum Classifiers*". Nele os pesquisadores replicaram o treinamento e classificação de modelos de aprendizagem de máquina quânticos, tanto com simulações quanto em um dispositivo quântico físico. Foram um dos primeiros modelos de aprendizagem de máquina quântica, com a replicação dos modelos *tree tensor network* (TTN) e *multi-scale entanglement renormalization ansatz* (MERA) clássicos para dispositivos quânticos.

1.1. Motivação

Ainda que alguns questionamentos pertinentes à aprendizagem de máquina quântica tenham sido respondidos por GRANT et al. (2018), algumas questões permanecem em aberto. Com a possibilidade de avaliar modificações na arquitetura dos modelos propostos, este trabalho baseia-se nos resultados obtidos por GRANT et al. (2018) para propor melhorias e experimentá-las.

1.2. Objetivos

Este trabalho propõe-se a replicar os resultados obtidos por GRANT et al. (2018) como ponto de partida. Os modelos de classificadores replicados são o TTN e MERA. Cada um dos modelos é treinado e avaliado sobre o conjunto de dados de variantes da planta iris (FISHER, 1936) e sobre o conjunto de dados de imagens de dígitos escritos à mão MNIST (LECUN; CORTES; BURGES, 1998).

Ademais, propõe-se a investigar o efeito da codificação por amplitude na inicialização do estado quântico dos modelos TTN e MERA. Avalia-se o efeito tanto na acurácia dos modelos quanto da quantidade de lotes de treinamento necessários para a convergência do modelo.

1.3. Estrutura

Este trabalho está dividido em sete capítulos. Além deste primeiro capítulo introdutório, no segundo capítulo são apresentados alguns dos conceitos básicos utilizados ao longo deste trabalho. O terceiro capítulo discute dois trabalhos relacionados em que este trabalho se baseou, suas contribuições e no que o presente trabalho difere. O capítulo quatro descreve a metodologia utilizada nos experimentos. No quinto capítulo é apresentado e discutido os resultados obtidos na experimentação. O sexto capítulo conclui os principais pontos deste trabalho e introduz possíveis trabalhos futuros. O sétimo capítulo é reservado para as referências deste trabalho.

2. Conceitos Básicos

Alguns termos e conceitos utilizados ao longo deste trabalho são introduzidos neste capítulo. São detalhados alguns conceitos e definições da computação quântica, como estado quântico, superposição e preparação de estado quântico.

2.1. Estado quântico

A unidade computacional básica da computação quântica é o bit quântico (ou *qubit*, do inglês *quantum bit*). O *qubit* é análogo ao dígito binário da computação clássica. Na computação quântica, o estado quântico representa os possíveis estados de um ou mais *qubits*. A notação de Dirac de *bra* e *ket* é utilizada para representar um estado quântico $|\psi>$ (ZWIEBACH, 2013). Leia-se tal notação como estado psi ou *ket* psi.

Segundo Barenco et al. (1995), um estado quântico pode ser descrito com uma superposição linear complexa de todos os estados binários dos *bits* $x_m \in \{0, 1\}$:

$$|\psi> = \sum_{x \in \{0,1\}^m} \alpha_x |x_1, \ldots, x_m> \tag{1}$$

$$\sum_x |\alpha_x|^2 = 1 \tag{2}$$

Em (1), α_x representa a amplitude de probabilidade associada ao *qubit*. $|x_1,\ldots, x_m>$ representa o estado da base computacional. As amplitudes de probabilidade são restritas pela condição explicitada em (2).

2.2. Superposição

Uma das características da computação quântica é a superposição de vários estados quânticos. Um *qubit* pode se encontrar em um estado representado por mais de um estado da base computacional ao mesmo tempo, através de uma combinação complexa dos estados da base computacional.

Um *qubit* em superposição possui um estado definido. A partir do momento da medição, o *qubit* perde a característica da superposição e passa a ser representado pelo estado da base computacional ao qual ele colapsou, já que a medição possui efeito computacional.

O estado da base computacional ao qual o *qubit* colapsa depende da amplitude de probabilidade associada a esse estado enquanto o *qubit* estava em superposição. Quanto maior o módulo da amplitude de probabilidade α_x do estado x, maior a probabilidade do *qubit*

colapsar para o estado x ao realizar-se uma medição. Tal probabilidade é dada por (3), onde p(x) representa a probabilidade do *qubit* assumir o estado x.

$$p(x) = |\alpha_x|^2 \tag{3}$$

2.3. Preparação de estado quântico

A preparação (ou inicialização) de estado quântico refere-se a codificar alguma informação em um estado quântico (SIERRA-SOSA; TELAHUN; ELMAGHRABY, 2020). Neste trabalho, usa-se o nome de preparação de estado quântico para indicar o carregamento das características de um exemplo de treinamento em um dispositivo quântico. Assim, exemplos de um conjunto de dados clássico podem ser utilizados em modelos de aprendizagem de máquina em dispositivos quânticos.

Há diferentes formas de inicializar um estado quântico. A primeira, e mais direta, abordagem é inicializar o estado quântico no mesmo estado da base computacional representado pela cadeia binária clássica. Dessa forma, um dado codificado em uma cadeia de *bits* $y \equiv b_1 b_2 \ldots b_n$ de tamanho n é carregado no dispositivo quântico ao inicializá-lo no estado quântico $|\psi> \equiv |b_1 b_2 \ldots b_n>$. Tal forma de preparação é conhecida como *Basis Embedding ou Basis Encoding* (WEIGOLD; BARZEN; LEYMANN; SALM, 2020).

Uma segunda abordagem é a chamada *Angle Embedding* por XANADU (2020) ou *Qubit Encoding* por GRANT et al. (2018). Nesta inicialização o estado quântico é inicializado usando-se n *qubits* para codificar N características, onde $N \leq n$. Assim, as N características são carregadas nas amplitudes dos n *qubits* separadamente, através de operadores de rotação. Os operadores de rotação podem ser os operadores sobre um *qubit* RX, RY ou RZ.

Outra possível abordagem é a inicialização de um estado quântico pela codificação do dado nas amplitudes da superposição do estado. Tal abordagem é conhecida como *Amplitude Encoding* por WEIGOLD et al. (2020) ou *Amplitude Embedding* por XANADU (2020). Esta inicialização permite compressão exponencial da memória, já que um estado de n *qubits* pode representar um dado de dimensão até 2^n. A implementação dessa forma de preparação depende do algoritmo utilizado. Deve-se então ponderar o custo computacional dessa preparação, pois o ganho pode ser negado durante a fase de classificação segundo GRANT et

al. (2018), onde o número de medições necessárias para uma boa estimativa aumenta com o número de amplitudes, conforme exposto por WEIGOLD et al. (2020).

Há ainda outras formas de preparação de um estado quântico fora as citadas anteriormente. Para saber mais, referir-se a WEIGOLD et al. (2020).

2.4. Decomposição de operadores

Operador quântico é um operador linear unitário que transforma o estado quântico. Os operadores quânticos que agem sobre mais de dois *qubits* dificilmente são implementados nos dispositivos quânticos atuais. Isso ocorre porque implementações diretas de tais operadores são demasiadamente custosas (SHENDE; BULLOCK; MARKOV, 2006). Então muitas vezes é necessário decompor um operador quântico, que age em três ou mais *qubits*, em operadores sobre um ou dois *qubits*.

Segundo BARENCO et al. (1995), um operador que age sobre n *qubits* pode ser decomposto nos operadores de ou exclusivo — também conhecido como *controlled not* (CNOT) ou Pauli-X controlado — e do operador geral sobre um *qubit*. Já SHENDE et al. (2004) demonstram como decompor um operador arbitrário sobre dois *qubits* utilizando CNOT e operadores sobre um *qubit*, notavelmente os operadores R_y e R_z.

Basta então combinar operadores sobre um *qubit* e CNOT para implementar qualquer operador quântico nos dispositivos quânticos atuais. Tais resultados foram importantíssimos para a implementação prática deste trabalho. As decomposições de operadores utilizadas serão descritas e discutidas em detalhes no capítulo 4.

2.5. Conjuntos de treinamento, validação e teste

É comum segmentar o conjunto de dados na aprendizagem de máquina. Divide-se o conjunto de dados com todos os exemplos disponíveis em subconjuntos de exemplos: um conjunto para treinamento, outro para validação e outro para teste.

O conjunto de treinamento contém os exemplos utilizados para ajustar os parâmetros do modelo durante a etapa de treinamento. O conjunto de testes contém exemplos utilizados para mensurar as métricas do modelo. É importante que o conjunto de testes contenha exemplos distintos do conjunto de treinamento para que sejam exemplos com os quais o modelo nunca adaptou seus parâmetros de acordo.

O conjunto de validação por sua vez contém exemplos do conjunto de treinamento, sendo um subconjunto dos exemplos de treinamento. É utilizado para mensurar as métricas de classificação do modelo durante a etapa de treinamento e se ter noção da convergência do modelo.

2.6. Parada antecipada

Uma técnica comum em aprendizagem de máquina é a parada antecipada (ou *early stopping*). Idealmente, quanto mais dados para treinamento, melhor. Entretanto, o modelo pode ficar enviesado demais com os exemplos de treinamento e perder a capacidade de generalização (*overfitting*), conforme aprende cada vez mais as características dos exemplos de treinamento. Para evitar essa perda na generalização do modelo, utiliza-se da parada antecipada para cessar o treinamento quando o modelo converge, a fim de evitar o *overfitting*.

Uma possível técnica de parada antecipada, e a utilizada neste trabalho na replicação dos modelos do conjunto MNIST, é a de acompanhar a acurácia em um conjunto de validação. Tal técnica baseia-se em ter um conjunto de exemplos de validação e utilizá-lo como um conjunto de testes durante o treinamento. Acompanha-se então a acurácia obtida nesse conjunto conforme o treinamento avança, e pára-se o treinamento caso a acurácia no conjunto de validação não aumente após um número pré-determinado de épocas ou lotes de treinamento.

2.7. Análise de componentes principais

A Análise de Componentes Principais, ou *Principal Component Analysis* (PCA) (WOLD; ESBENSEN; GELADI, 1987), é uma técnica de redução de dimensionalidade. É utilizada para reduzir a dimensionalidade de conjuntos de dados com pouca perda de informação. Pode ser utilizada para acelerar o treinamento de modelos de aprendizagem de máquina, já que com menos características de entrada pode-se utilizar um modelo mais simples.

É realizada em cinco passos. O primeiro é a padronização das características iniciais a fim de que contribuam igualmente para a análise. O segundo passo é a criação de uma matriz de covariância para indicar como as características influenciam umas às outras e se são correlatas ou inversamente correlatas. O terceiro passo consiste em identificar os componentes principais ao calcular os autovetores e autovalores da matriz de covariância.

No quarto passo são escolhidos os componentes principais. Usualmente descarta-se os de menor ordem de significância, a fim de reduzir a dimensionalidade sem tantas perdas de informação. Monta-se uma matriz de características cujas colunas são os autovetores dos componentes principais escolhidos. Por fim, com os componentes principais selecionados, o quinto passo consiste em transformar os dados do conjunto original, orientados nos eixos originais para os eixos representados pelos componentes principais.

3. Trabalhos Relacionados

Este capítulo trata dos trabalhos anteriores aos quais este trabalho se baseia. O primeiro trabalho abordado é o "*Hierarchical Quantum Classifiers*" de GRANT et al. (2018), na seção 3.1. O segundo trabalho abordado é "*TensorFlow Quantum: Impacts of Quantum*" de SIERRA-SOSA et al. (2020). São discutidos os principais pontos e conclusões de cada trabalho, assim como a diferença entre cada um dos trabalhos e este.

3.1. Hierarchical Quantum Classifiers

O trabalho de GRANT et al. (2018) caracteriza-se por ser um dos primeiros trabalhos a treinar modelos de aprendizagem de máquina quântica em dispositivos quânticos físicos. Nele, os pesquisadores treinaram classificadores com base em circuitos de duas topologias hierárquicas, a TTN e a MERA. Tais topologias são descritas em mais detalhes na seção 4.1.

Os modelos foram treinados sobre três conjuntos de dados, o conjunto de variantes da flor iris (FISHER, 1936), o conjunto de dígitos escritos a mão MNIST (LECUN; CORTES; BURGES, 1998) e um conjunto de estados quânticos sintéticos (GRANT; BENEDETTI; CAO; HALLAM; LOCKHART; STOJEVIC; GREEN; SEVERINI, 2018). Nos dois primeiros conjuntos de dados, a preparação do estado quântico foi feita por *Angle Embedding*. E por *Amplitude Embedding* no conjunto de dados quânticos sintéticos.

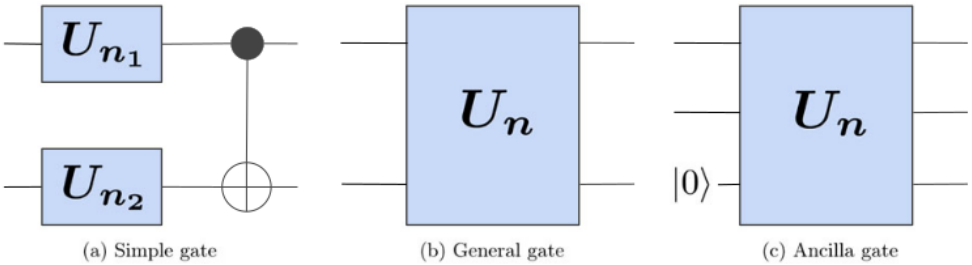
Os pesquisadores também investigaram a influência na acurácia por diferentes estruturas dos operadores quânticos, os *unitaries*. A primeira estrutura consistia de dois operadores sobre um *qubit* e o CNOT, chamada de *simple gate*, visualizada na Figura 1 (a).

A segunda estrutura consistia de um operador sobre dois *qubits*, chamada de *general gate*, visualizada na Figura 1 (b). E a terceira de um operador sobre três *qubits*, sendo o terceiro *qubit* auxiliar, inicializado no estado $|0>$. A terceira estrutura é visualizada na Figura 1 (c).

Analisaram também a diferença na acurácia ao utilizar operadores com rotações reais ou complexas. Os operadores U_{n_1} e U_{n_2} visualizados na Figura 1 (a) são de rotação. Para os modelos que permitem rotações complexas, tais operadores são o operador geral sobre um *qubit*. Já nos modelos de rotações reais, ambos os operadores são o operador de rotação sobre o eixo y, o RY.

Em cada conjunto de dados, os modelos foram divididos em tarefas de classificação binária. As tarefas de classificação sobre o conjunto de dados iris consistiram da classificação entre classes dois-a-dois das três classes do conjunto. O modelo de TNN com operadores de estrutura *simple gate* alcançou acurácia de 100% e desvio padrão 0 nas tarefas de classificação entre a classe 1 ou 2 e entre a classe 1 ou 3. Tal modelo alcançou acurácia de 96,77% e desvio padrão 0 na tarefa de classificação entre a classe 2 ou 3.

Figura 1 - Estruturas dos operadores investigadas



Fonte: Figura 2 de GRANT et al. (2018)

No conjunto de dados MNIST, as tarefas de classificação binária foram quatro: classificar números maiores do que 4; classificar números pares ou ímpares; classificar algarismo 0 ou 1; classificar algarismo 2 ou 7. Os pesquisadores investigaram a acurácia para os modelos TTN e MERA, com operadores *simple gate* ou *general gate*, e ao utilizar rotações reais ou complexas. Também investigaram a acurácia de um modelo híbrido que consistiu de uma topologia MERA pré-treinada de uma topologia TTN. Referir à Tabela 3 de GRANT et al. (2018) para os resultados completos sobre o conjunto MNIST.

Segundo GRANT et al. (2018), modelos com operadores *general gate* tiveram acurácia maior do que modelos com operadores *simple gate*; rotações complexas melhoraram a acurácia em relação a rotações reais; modelos da topologia MERA superaram modelos da topologia TTN em todas as tarefas de classificação.

No conjunto de dados sintéticos, os pesquisadores treinaram um modelo TTN com operadores *general gate* e compararam a acurácia com a obtida num modelo TTN com operadores *ancilla gate*. Conforme GRANT et al. (2018), os resultados da acurácia da TTN com operadores *general gate* ficaram próximos à 50%, onde há a necessidade de um modelo mais expressivo. Enquanto que os resultados da TTN com operadores *ancilla gate* chegaram até 64%.

Outra contribuição do trabalho foi a investigação do impacto de ruído despolarizador no desempenho de classificação do modelo TTN com operadores *simple gate*. Basearam-se na tarefa de classificação entre as classes 1 e 2 do conjunto iris. Chegaram à conclusão de que a

acurácia permaneceu acima de 95% para um ruído de até 0.07 e de que o desvio padrão da acurácia aumentou conforme a intensidade do ruído.

Por fim, os pesquisadores implementaram o modelo TTN com operadores *simple gate* no dispositivo quântico ibmqx4, disponível no IBM Quantum Experience. O modelo foi treinado para classificar entre a classe 1 e 2 do conjunto iris. Alcançou uma acurácia de 100%, conforme a obtida anteriormente durante as simulações.

Apesar de importantes contribuições na época, o trabalho não investigou como diferentes inicializações impactam na acurácia e no treinamento necessário para os modelos. A quantidade de operadores cresce exponencialmente com a quantidade de *qubits* no circuito quântico da TTN ou MERA, conforme observado por GRANT et al. (2018). Ao substituir a inicialização *Angle Embedding* por *Amplitude Embedding* é possível utilizar mais características no treinamento sem aumentar a quantidade de *qubits*, dado que *Amplitude Embedding* pode representar um dado de dimensão até 2^n com n *qubits*. Assim poderia-se utilizar 256 características com 8 *qubits*, ao invés das 8 utilizadas pelo trabalho. Ou até mesmo reduzir a quantidade de *qubits* dos modelos para 3 para utilizar as mesmas 8 características.

Outro ponto não abordado pelo trabalho é o efeito do *bias* nos operadores dos modelos. A adição de *bias* é um recurso da aprendizagem de máquina clássica para ajudar o modelo a escapar de mínimos/máximos locais. É interessante investigar se o mesmo comportamento se mantém para a aprendizagem de máquina quântica.

3.2. TensorFlow Quantum: Impacts of Quantum State Preparation on Quantum Machine Learning Performance

No trabalho de SIERRA-SOSA et al. (2020), os pesquisadores comparam duas formas de preparação de estados ao treinar modelos híbridos de classificação. Os modelos consistiram de camadas de convolução quântica e de redução da dimensionalidade (*pooling layers*), seguidas de uma medição para alimentar uma *Multi-layer Perceptron* (MLP) clássica, para enfim calcular o erro médio quadrático (do inglês *mean square error* ou MSE) e a função de custo.

As formas de preparação de estado comparadas foram *Angle Embedding* e *Amplitude Embedding*. Os pesquisadores mantiveram os mesmos hiperparâmetros entre os dois modelos, variando somente a forma de preparação do estado quântico.

Os conjuntos de dados sobre os quais os modelos foram treinados e utilizados para classificação foram conjuntos de dados sintéticos da biblioteca Scikit-Learn. Os conjuntos de

dados possuíam duas classes. Os pesquisadores variaram os centróides das classes para variar a sobreposição entre os exemplos de cada classe (*cf.* Figura 5 do trabalho).

Para obtenção dos resultados, compararam os modelos sobre conjuntos de dados com diferentes centróides e utilizando diferentes épocas de treinamento. As épocas de treinamento utilizadas na comparação foram 8 épocas e 50 épocas. As métricas utilizadas para comparação foram a acurácia, o *recall* (*cf.* Equação 10 do trabalho) e o *F1-score* (*cf.* Equação 11 do trabalho).

Os pesquisadores chegaram às conclusões de que o modelo com *Amplitude Embedding* obteve métricas melhores com 50 épocas ao invés de 8 épocas; que o modelo com *Amplitude Embedding* supera o modelo com *Angle Embedding* em todos os conjuntos de dados; que o modelo com *Amplitude Embedding* não melhora significativamente após 12 épocas, e então o número de épocas ideais para treinamento seria entre 10 e 20, segundo SIERRA-SOSA et al. (2020).

O presente trabalho se diferencia do de SIERRA-SOSA et al. (2020) por todo o processo da aprendizagem de máquina quântica, desde a preparação do estado quântico até o treinamento dos modelos e a classificação, ser em dispositivo quântico — mesmo que simulado. No trabalho comparado, dispositivo quântico é utilizado para inicialização do estado e *pooling*, enquanto que o treinamento e classificação é realizado em uma MLP clássica.

Outra diferença notável são os modelos e conjuntos de dados utilizados. É utilizado uma MLP clássica, um modelo de Rede Neural Artificial, e um conjunto de dados sintéticos da biblioteca Scikit-Learn no trabalho comparado. Neste trabalho utilizou-se modelos hierárquicos, como a TTN e MERA em conjuntos de dados clássicos, o conjunto iris e o MNIST.

4. Metodologia dos experimentos

Este capítulo descreve os experimentos realizados, explica sua metodologia e o porquê de algumas decisões de implementação. Os experimentos replicados foram: o modelo TTN, com operadores simples e rotações reais, no conjunto de dados iris; o modelo TTN e o modelo MERA no conjunto de dados MNIST com preparação de estado *Angle Embedding*; e o modelo TTN e o modelo MERA no conjunto de dados MNIST com preparação de estado *Amplitude Embedding*.

4.1 Visão Geral

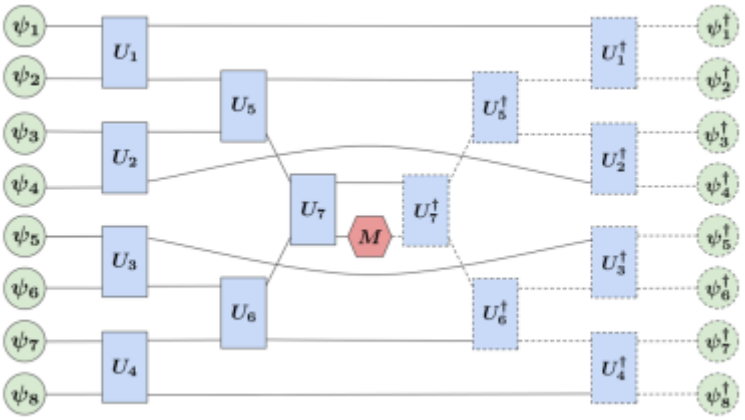
A linguagem de programação utilizada nos experimentos foi Python em sua versão 3.8.8. Utilizou-se o Jupyter Notebook na versão 6.3.0 para melhor organização e disposição do código fonte. Utilizou-se a biblioteca Pennylane (XANADU, 2020) na versão 0.16.0 para preparar o estado quântico, criar o circuito quântico e realizar o treinamento com técnicas de gradiente descendente. Para importar os conjuntos de dados e manipulá-los utilizou-se da biblioteca pandas na versão 1.2.4, da biblioteca numpy na versão 1.19.5. e da biblioteca scikit-learn (DEVELOPERS, 2020) na versão 0.24.1.

A máquina utilizada para o treinamento e tarefas de classificação continha:

- Processador i5-1135G7, 8 MB de *cache*, 4 núcleos, 8 threads, 2,4 GHz a 4,2 GHz
- Memória RAM de 16 GB, DDR4, 2400 MHz
- Armazenamento SSD M.2 2230, 256 GB, PCIe x4 NVMe de 3ª geração
- Placa gráfica integrada Intel Iris Xe

As topologias hierárquicas utilizadas nos circuitos quânticos dos modelos foram a TTN, visível na Figura 2, e a MERA, visível na Figura 3.

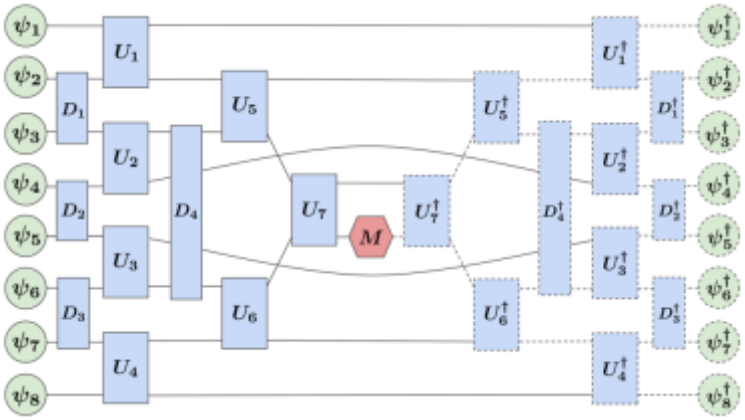
Figura 2 - Circuito quântico do classificador TTN



Fonte: Figura 1 (a) de GRANT et al. (2018)

Os operadores U_i, vistos nas Figuras 2 e 3, e D_j, vistos na Figura 3, foram parametrizados em quatro configurações diferentes. Para a primeira configuração, a fim de replicar operadores *simple gate* com rotações reais, utilizou-se operador R_y sobre um *qubit* e o operador CNOT. Na segunda configuração, a fim de replicar operadores *simple gate* com rotações complexas, utilizou-se operador $U3$ sobre um *qubit* (XANADU, 2020) e o operador CNOT. Na terceira configuração, para replicar operadores *general gate* com rotações reais, utilizou-se operador R_y sobre um *qubit* e o operador CNOT para decompor um operador sobre dois *qubits*, conforme trabalho de SHENDE et al. (2004). Na quarta configuração, para replicar operadores *general gate* com rotações complexas, utilizou-se um *template* da Pennylane, *Arbitrary Unitary* (XANADU, 2020), para construir um operador qualquer sobre dois *qubits*.

Figura 3 - Circuito quântico do classificador MERA



Fonte: Figura 1 (b) de GRANT et al. (2018)

Para obter os resultados e compará-los com os obtidos por GRANT et al. (2018), seguiu-se o mesmo procedimento do trabalho referido. Os modelos das tarefas de classificação, tanto do iris quanto do MNIST, foram treinados e testados com cinco vetores de parâmetros iniciais diferentes. A acurácia e um desvio padrão, para cada modelo em cada tarefa de classificação, foi determinada a partir dos testes de classificação em cada uma das cinco inicializações diferentes. Utilizou-se a biblioteca Numpy, com a semente no valor 42, para gerar os vetores de inicialização aleatoriamente.

4.2 Conjunto de treinamento e de teste

Os classificadores foram treinados e executaram tarefas de classificação binária em dois conjuntos de dados. O primeiro deles foi o conjunto de variantes da flor iris (FISHER, 1936), que contém exemplos de três variantes da flor, sendo elas a Setosa, Versicolor e

Virginica. Nesse conjunto de dados há cinquenta exemplos de cada uma das variantes, totalizando cento e cinquenta exemplos no conjunto. As tarefas de classificação foram binárias. Assim, treinaram um modelo para classificar as variantes Setosa e Versicolor; um modelo para classificar as variantes Versicolor e Virnigica; e um modelo para classificar as variantes Setosa e Virginica.

O segundo conjunto de dados foi o de dígitos escritos à mão, o MNIST (LECUN; CORTES; BURGES, 1998). O conjunto é composto por imagens de 28x28 *pixels*, em escala de cinza, contendo exemplos de todos os algarismos arábicos. No total há 70000 exemplos considerando todos os algarismos. As tarefas de classificação foram as seguintes: algarismo é maior do que quatro ou é menor ou igual a quatro; algarismo é par ou ímpar; algarismo é o zero ou algarismo é o um; algarismo é o dois ou algarismo é o sete.

4.3 Pré-processamento dos dados

Ambos os conjuntos de dados iris e MNIST não continham exemplos com dados faltantes. Não foi necessário utilizar técnicas de pré-processamento para limpar os dados. Entretanto, dois tratamentos foram feitos para carregar os dados nos modelos durante a fase de treinamento e a de teste.

O primeiro tratamento, realizado tanto no conjunto do iris quanto do MNIST, foi a normalização dos valores das características. Isso foi feito para colocar os valores numa escala em comum, que variou entre os valores -1 e 1.

O segundo tratamento foi o PCA (WOLD; ESBENSEN; GELADI, 1987), realizado somente no conjunto MNIST. Cada exemplo do conjunto MNIST é uma imagem 28x28 *pixels*. Assim, cada exemplo possui 784 características de entrada, que são os valores em escala de cinza de cada um dos *pixels* da imagem. Não seria possível carregar todas as 784 características de entrada nos 8 *qubits* do circuito quântico. Por causa disso, e para replicar o tratamento de GRANT et al. (2018), aplicou-se a PCA nas 784 características e obteve-se as 8 características com maior variância. Para implementar tal tratamento, utilizou-se a PCA da biblioteca scikit-learn (DEVELOPERS, 2020) com o número de componentes principais iguais a 8.

Algumas tarefas de classificação não necessitaram de todo o conjunto de dados. No conjunto iris, as tarefas de classificação consideraram as classes dois-a-dois. Então, para classificar entre as classes Setosa e Versicolor, os exemplos da classe Virginica foram desconsiderados dos conjuntos de treinamento e de teste, antes da etapa de treinamento. Seguiu-se a mesma lógica ao classificar entre as classes Setosa e Virginica, ao excluir os

exemplos da classe Versicolor, e ao classificar entre as classes Virginica e Versicolor, ao excluir os exemplos da classe Setosa.

No conjunto MNIST, as tarefas de classificação “algarismo é o zero ou algarismo é o um” e “algarismo é o dois ou algarismo é o sete” não utilizaram todo o conjunto de dados. Retirou-se do conjunto de dados todos os exemplos de algarismos que não participaram da tarefa de classificação. Então o modelo treinado para a tarefa “algarismo é o zero ou algarismo é o um” foi treinado e testado somente com exemplos dos algarismos zero e um do conjunto de dados. O mesmo se fez para treinar e testar o modelo para a tarefa “algarismo é o dois ou algarismo é o sete”.

Um último tratamento feito foi a normalização dos rótulos (*labels*) das classes. Definida a tarefa de classificação, uma das classes assumia o valor do rótulo de -1 e a outra assumia o valor de 1. Tal tratamento foi executado pois a medição ao final do circuito quântico era feita com um operador Pauli-Z.

4.4 Inicialização do estado quântico

A inicialização do estado quântico foi feita com o auxílio de *templates* da Pennylane (XANADU, 2020). Utilizou-se o *template Angle Embedding* para os modelos que replicaram os resultados de GRANT et al. (2018). Tal *template* foi utilizado com operadores R_y sobre um *qubit* para inicializar cada *qubit*. A rotação utilizada na inicialização *Qubit Encoding* por GRANT et al. (2018) não foi especificada. Por conta disso, e para abranger tantos os modelos com operadores de rotação real quanto os modelos com operadores de rotação complexa, escolheu-se o operador R_y ao invés de inicializar com operador R_x ou R_z.

Utilizou-se o *template Amplitude Embedding* para os modelos com a preparação de estado *Amplitude Encoding*. As configurações utilizadas neste *template* foram verdadeiro para normalização e o valor 0.0 para o preencher as 2^n características de entrada suportadas.

Utilizou-se oito *qubits* no circuito quântico ao utilizar *Angle Embedding*, pois foram utilizados os 8 componentes principais do conjunto MNIST para o treinamento e classificação. Ao utilizar *Amplitude Embedding*, reduziu-se o circuito quântico para utilizar 3 *qubits*, que é a quantidade necessária para treinar com as características de entrada dos oito componentes principais.

4.5 Fase de treinamento

Os modelos do conjunto iris foram treinados sobre 66% dos exemplos disponíveis. Os outros 34% foram utilizados na fase de teste para medir a acurácia do modelo. Adotou-se tal divisão para replicar a utilizada por GRANT et al. (2018).

A técnica utilizada para otimizar os parâmetros do modelo foi a de gradiente descendente. Utilizou-se o otimizador *GradientDescentOptimizer* da Pennylane (XANADU, 2020) para implementar tal técnica. O hiperparâmetro da taxa de aprendizagem η utilizado foi o padrão da Pennylane de 0.01. O modelo foi treinado com base em 100 épocas de treinamento. E utilizou-se lotes de treinamento de tamanho único. Tanto a quantidade de épocas quanto o tamanho do lote de treinamento e o hiperparâmetro η foram determinados empiricamente, pois não são explicitados nos experimentos de GRANT et al. (2018).

Os modelos do conjunto MNIST foram treinados seguindo a divisão de conjuntos de treinamento, de validação e de teste conforme os experimentos de GRANT et al. (2018). 55000 dos 70000 exemplos do conjunto foram utilizados para treinamento, 5000 exemplos foram utilizados para validação e 10000 foram utilizados para teste. Nas tarefas que não utilizaram todo o conjunto, como as tarefas "algarismo é o zero ou algarismo é o um" e "algarismo é o dois ou algarismo é o sete", manteve-se a mesma proporção para os três diferentes conjuntos. Assim, utilizou-se 78% dos exemplos para treinamento, 7% para validação e os 15% restantes para teste.

Utilizou-se a técnica *Adaptive Moment Estimation* (ADAM) (KINGMA; BA, 2015) para otimizar os parâmetros dos modelos nas tarefas de classificação do MNIST. Para sua implementação, utilizou-se o otimizador *AdamOptimizer* da Pennylane (XANADU, 2020). Os hiperparâmetros da taxa de aprendizagem η , do primeiro momento β_1 e segundo momento β_2 , não foram explicitados por GRANT et al. (2018). Por conta disso utilizou-se os valores padrões do Pennylane, que são $\eta = 0.01$, $\beta_1 = 0.9$ e $\beta_2 = 0.99$.

Replicou-se também a condição de parada nos modelos treinados sobre o conjunto MNIST. Os exemplos de treinamento foram divididos em lotes de 20 exemplos. A cada 10 lotes de treinamento foi avaliada a acurácia nos conjuntos de validação e de teste. O treinamento do modelo era interrompido quando a acurácia no conjunto de validação não incrementava durante 30 avaliações consecutivas. Os parâmetros ótimos encontrados pelo modelo eram determinados como os parâmetros com os quais o modelo atingiu a acurácia máxima no conjunto de validação.

A função de custo utilizada tanto no *GradientDescentOptimizer* quanto no *AdamOptimizer* foi a do erro médio quadrático. Os otimizadores agiram a fim de minimizar o erro médio quadrático entre o valor do rótulo da classe e o valor predito pelo modelo ao realizar a medição..

4.6 Fase de testes

Ao final da etapa de treinamento, obtinha-se o vetor de parâmetros ótimo encontrado pelo modelo durante o treinamento. Para mensurar a acurácia do modelo, cada um dos exemplos do conjunto de teste foi carregado no circuito quântico do modelo. O circuito era executado com os operadores com as rotações do vetor de parâmetro ótimo. Ao final do circuito, realizava-se a medição.

Determinou-se o limiar no valor 0 pois a classificação era binária em todas as tarefas e uma classe assumiu o rótulo no valor de -1 e a outra no valor de 1. Caso o valor da medição estivesse entre -1 e 0 (incluso), considerava-se que o modelo classificou o exemplo de teste como pertencente à classe cujo rótulo tinha valor -1. Caso contrário, considerava-se que o modelo classificou o exemplo de teste como pertencente à classe com o rótulo de valor 1.

A acurácia de um modelo em uma dada tarefa de classificação é dada por (4). A quantidade de classificações erradas foi o número de vezes que o modelo previu o exemplo de teste como pertencente à classe de rótulo -1 quando deveria ser da classe de rótulo 1 e vice-versa.

$$Acurácia = \left(1 - \frac{quantidade\ de\ classificações\ erradas}{número\ total\ de\ exemplos\ do\ conjunto\ de\ teste}\right) \quad (4)$$

5. Análise de resultados

Este capítulo apresenta os resultados obtidos nos experimentos de replicação dos de GRANT et al. (2018), tanto no conjunto de dados iris quanto no conjunto de dados MNIST. Também apresenta um comparativo entre os resultados experimentais com os modelos com inicialização *Angle Embedding* e *Amplitude Embedding*.

5.1 Conjunto de dados Iris

Os resultados dos experimentos sobre o conjunto de dados iris estão dispostos na Tabela 1. É exibida tanto a acurácia quanto um desvio padrão para cada uma das três tarefas de classificação binária. A coluna Trabalho identifica de qual trabalho são os resultados dos experimentos. (i) identifica os resultados obtidos por GRANT et al. (2018) (*cf.* Tabela 2 do referido trabalho), enquanto que (ii) identifica os resultados obtidos na replicação dos experimentos.

Tabela 1 - Acurácias de classificação no conjunto iris

Classificador	Operador	Rotações	Trabalho	1 ou 2	2 ou 3	1 ou 3
TTN	Simples	Real	(i)	100.00±0.00	96.77±0.00	100.00±0.00
TTN	Simples	Real	(ii)	100.00±0.00	95.76±4.60	100.00±0.00

Fonte: Elaborada pelo autor com base na Tabela 2 de GRANT et al. (2018)

Conforme visto na Tabela 1, o modelo TTN com operadores *simple gate* de rotações reais obteve um bom desempenho nas três tarefas de classificação. Para as tarefas de classificação entre as classes 1 (Setosa) ou 2 (Versicolor) e entre as classes 1 (Setosa) ou 3 (Virginica), os resultados da replicação foram idênticos ao obter acurácia de 100% com desvio padrão nulo.

Por outro lado, o resultado da replicação da tarefa de classificação entre as classes 2 (Versicolor) ou 3 (Virginica) não foi idêntico. Os resultados obtidos nas replicações da tarefa ocasionaram uma acurácia média de 95.76%, cerca de um ponto percentual abaixo do obtido no trabalho a ser comparado. Outro ponto a se notar é o desvio padrão não nulo. Segundo GRANT et al. (2018), o desvio padrão obtido pela TTN na tarefa de classificação entre as classes 2 (Versicolor) ou 3 (Virginica) foi nulo. Isso implica que o modelo TTN obteve a mesma acurácia nos cinco vetores de parâmetros iniciais distintos. O mesmo resultado não foi obtido na experimentação deste trabalho, onde o desvio padrão ficou em 4.60%.

Nota-se então que é possível replicar os resultados obtidos por GRANT et al. (2018). A acurácia e um desvio padrão para a tarefa de classificação entre a classe 2 (Versicolor) ou 3

(Virginica) poderia ser melhor replicada caso utilizado o mesmo número de épocas de treinamento, taxa de aprendizagem e vetores de parâmetros iniciais.

5.2 Conjunto de dados MNIST

Os resultados dos experimentos no conjunto de dados MNIST são expostos na Tabela 2. Observa-se tanto a acurácia quanto um desvio padrão para cada uma das quatro tarefas de classificação binária sobre o conjunto.

Novamente a coluna Trabalho expressa a qual trabalho a linha de resultados se refere. (i) identifica como sendo os resultados de GRANT et al. (2018) das tarefas de classificação para determinado modelo. A linha com (ii) identifica os resultados obtidos pelo presente trabalho ao replicar os experimentos.

Percebe-se que a tarefa melhor replicada foi se o algarismo é maior do que quatro ao analisar a Tabela 2. A maior diferença na acurácia foi de cerca de 5.99% nesta tarefa de classificação, ocorrida no modelo TTN com operadores *general gate* com rotações reais.

Nas outras tarefas de classificação não obteve-se um resultado tão expressivo quanto da tarefa de classificação “algarismo é maior do que quatro”. A diferença máxima entre a acurácia obtida por GRANT et al. (2018) e a acurácia da replicação foi de 15.32% para a tarefa de classificação “algarismo é par”, que aconteceu no modelo TTN com operadores *simple gate* com rotações complexas. Para a tarefa “algarismo é o zero ou algarismo é o um” foi de 16.92%, no modelo TTN com operadores *simple gate* com rotações complexas. E para “algarismo é o dois ou algarismo é o sete” foi de 24.17% e também ocorreu no modelo TTN com operadores *simple gate* com rotações complexas.

As constatações feitas por GRANT et al. (2018) podem ser observadas na replicação. Os modelos com operadores *general gate* obtêm acurácia maior do que modelos com operadores *simple gate*. Conforme a Tabela 2, isso pôde ser confirmado pelas replicações dos modelos.

Tabela 2 - Acurácias de classificação no conjunto MNIST

Classificador	Operador	Rotações	Trabalho	É > 4	É par	0 ou 1	2 ou 7
TTN	Simples	Real	(i)	65.59±0.57	72.17± 0.89	92.12± 2.17	68.07± 2.42
TTN	Simples	Real	(ii)	65.21±0.84	62.48± 0.40	79.67± 0.87	66.98± 2.08
TTN	Geral	Real	(i)	74.89±0.95	83.13± 1.08	99.79± 0.02	97.64± 1.60
TTN	Geral	Real	(ii)	68.9±3.87	71.77± 1.32	88.05± 3.25	78.72± 1.32
MERA	Geral	Real	(i)	75.20±1.51	82.83± 1.19	99.84± 0.06	98.02± 1.40
MERA	Geral	Real	(ii)	75.62±0.31	73.41± 0.47	94.35± 0.67	81.92± 0.76
TTN	Simples	Complexa	(i)	70.90±0.73	80.12± 0.64	99.37± 0.12	94.09± 3.37
TTN	Simples	Complexa	(ii)	69.91±1.71	64.80± 0.92	82.45± 0.48	69.92± 1.09
TTN	Geral	Complexa	(i)	77.56±0.45	83.53± 0.69	99.77± 0.02	97.63± 1.48
TTN	Geral	Complexa	(ii)	75.62±0.98	74.64± 0.46	91.39± 1.60	81.01± 0.19
MERA	Geral	Complexa	(i)	79.10±0.90	84.85± 0.20	99.74± 0.02	98.86± 0.07
MERA	Geral	Complexa	(ii)	78.05±1.92	76.45± 0.84	93.25± 1.65	82.51± 0.49

Fonte: Elaborada pelo autor com base na Tabela 3 de GRANT et al. (2018)

Além disso, operadores que permitem rotação complexa obtêm melhores acurácias do que operadores com rotação real, com exceção da tarefa de classificação “algarismo é o zero ou algarismo é o um” onde a acurácia já ultrapassava 99.5%. E isso ocorre apesar das características dos dados do conjunto MNIST serem todas reais. Como visto na Tabela 2, o modelo MERA com rotações reais superou o modelo MERA com rotações complexas nas replicações realizadas.

Uma outra constatação, confirmada pelas replicações e que pode ser observada na Tabela 2, é de que modelos com a mesma configuração mas com topologia MERA superaram modelos com topologia TTN em todas as tarefas de classificação.

No mais, constata-se que a replicação nas tarefas de classificação “algarismo é par”, “algarismo é o zero ou algarismo é o um” e “algarismo é o dois ou algarismo é o sete” não obteve resultados idênticos aos de GRANT et al. (2018). Possivelmente os resultados em tais tarefas de classificação poderiam ser replicados fidedignamente caso utilizados os mesmos parâmetros do ADAM, os mesmos vetores de parâmetros iniciais e a mesma semente de variável aleatória na divisão dos conjuntos de treinamento, validação e teste. Entretanto, tais informações não foram descritas no trabalho de GRANT et al. (2018).

5.3 Conjunto de dados MNIST com *Amplitude Embedding*

Os resultados dos experimentos no conjunto de dados MNIST, tanto com inicialização *Angle Embedding* quanto com inicialização *Amplitude Embedding*, são expostos nas Tabela 3 e Tabela 4. Na Tabela 3 observa-se tanto a acurácia quanto um desvio padrão para cada uma das quatro tarefas de classificação binária sobre o conjunto. Na Tabela 4 exibe-se um comparativo entre a quantidade de lotes de treinamento necessários para a convergência de cada modelo.

Conforme visto na Tabela 3, o modelo com preparação de estado *Amplitude Embedding* foi superior ao mesmo modelo com preparação de estado *Angle Embedding* em todas as tarefas de classificação. Para a tarefa de classificação “algarismo é maior do que quatro”, a maior diferença da média de acurácia ocorreu no modelo TTN com operadores *general gate* e rotações reais, onde a diferença foi de 5.51%. A menor diferença entre as médias de acurácia, para esta mesma tarefa, ocorreu no modelo MERA com operadores *general gate* e rotações complexas, onde a melhora ao utilizar *Amplitude Embedding* foi de 0.94%.

Para a tarefa de classificação “algarismo é par”, o maior ganho de acurácia ocorreu no modelo MERA com operadores *general gate* e rotações complexas com um ganho de 3.14% na acurácia. Na tarefa de classificação “algarismo é o zero ou algarismo é o um” o maior ganho de acurácia foi de 10.32% e ocorreu no modelo TTN com operadores *simple gate* e rotações complexas. Apesar deste modelo ter tido o maior desvio padrão dentre todos os modelos em todas as tarefas de classificação, o ganho foi expressivo.

Tabela 3 - Acurácias de classificação no conjunto MNIST com duas inicializações de estado distintas

Classificador	Operador	Rotações	Inicialização	É > 4	É par	0 ou 1	2 ou 7
TTN	Simples	Real	*Angle*	65.21± 0.84	62.48± 0.40	79.67± 0.87	66.98± 2.08
TTN	Simples	Real	*Amplitude*	68.08± 0.46	64.51± 0.17	83.01± 0.88	68.01± 1.38
TTN	Geral	Real	*Angle*	68.9±3.87	71.77± 1.32	88.05± 3.25	78.72± 1.32
TTN	Geral	Real	*Amplitude*	74.41± 0.30	73.89± 0.39	89.84± 0.98	80.26± 0.6
MERA	Geral	Real	*Angle*	75.62± 0.31	73.41± 0.47	94.35± 0.67	81.92± 0.76
MERA	Geral	Real	*Amplitude*	77.28± 0.87	74.61± 0.81	95.94± 1.74	82.78± 3.32
TTN	Simples	Complexa	*Angle*	69.91± 1.71	64.80± 0.92	82.45± 0.48	69.92± 1.09
TTN	Simples	Complexa	*Amplitude*	74.85± 0.73	67.67± 0.47	92.77± 5.90	77.40± 2.13
TTN	Geral	Complexa	*Angle*	75.62± 0.98	74.64± 0.46	91.39± 1.60	81.01± 0.19
TTN	Geral	Complexa	*Amplitude*	78.67± 0.13	76.44± 0.29	93.01± 2.97	86.32± 3.31
MERA	Geral	Complexa	*Angle*	78.05± 1.92	76.45± 0.84	93.25± 1.65	82.51± 0.49
MERA	Geral	Complexa	*Amplitude*	78.99± 1.05	79.59± 0.69	94.24± 2.95	86.96± 1.28

Fonte: Elaborada pelo autor com base nos resultados das experimentações

Já na tarefa “algarismo é o dois ou algarismo é o sete” o maior ganho ocorreu novamente no TTN com operadores *simple gate* e rotações complexas, sendo o ganho de 7.48% na acurácia. A menor melhora de acurácia ocorreu no modelo MERA com operadores *general gate* e rotações reais na tarefa de classificação “algarismo é o dois ou algarismo é o sete”, onde a melhora foi de 0.86%. Essa melhora é menor do que um desvio padrão entre as acurácias, o que leva a crer que a melhora não foi expressiva neste caso.

Na Tabela 4 é possível visualizar a média e um desvio padrão dos lotes de treinamento necessários para a convergência dos modelos replicados neste trabalho. Nota-se então que a convergência ocorreu mais cedo ao utilizar *Amplitude Embedding* do que *Angle Embedding*

em todas as tarefas de classificação de todas as configurações do modelo TTN, exceto no modelo TTN com rotações simples e reais..

Tabela 4 - Lotes de treinamento necessários para a convergência do modelo no conjunto MNIST

Classificador	Operador	Rotações	Inicialização	É > 4	É par	0 ou 1	2 ou 7
TTN	Simples	Real	*Angle*	715±77.78	590± 226.27	460± 127.28	601± 57.09
TTN	Simples	Real	*Amplitude*	816±99.55	530± 55.67	450± 121.24	520± 38.56
TTN	Geral	Real	*Angle*	710± 325.27	940± 84.85	624± 50.66	585± 70.07
TTN	Geral	Real	*Amplitude*	617± 251.46	867± 96.55	520± 81.33	565± 57.14
MERA	Geral	Real	*Angle*	750±28.28	745± 77.78	614± 19.80	533± 103.23
MERA	Geral	Real	*Amplitude*	946±66.53	883± 89.40	522± 93.38	574± 49.65
TTN	Simples	Complexa	*Angle*	945± 106.03	1100± 168.68	550± 28.29	538± 96.16
TTN	Simples	Complexa	*Amplitude*	776±68.08	516± 15.27	472± 80.76	495± 100.42
TTN	Geral	Complexa	*Angle*	895± 134.35	685± 49.50	619± 12.73	553± 74.95
TTN	Geral	Complexa	*Amplitude*	854±36.71	490± 43.58	555± 25.86	495± 08.09
MERA	Geral	Complexa	*Angle*	485±21.21	570± 141.42	415± 63.64	606± 53.68
MERA	Geral	Complexa	*Amplitude*	706±25.01	680± 15.50	496± 39.52	574± 56.47

Fonte: Elaborada pelo autor com base nos resultados das experimentações

O mesmo não ocorreu para o modelo MERA, onde foram necessários mais lotes de treinamento para treinar os modelos com *Amplitude Embedding* nas tarefas de classificação. A única exceção foi a tarefa de classificação “algarismo é o zero ou algarismo é o um” para o modelo MERA com *general gate* e rotações reais.

6. Conclusões e Trabalhos Futuros

Este trabalho buscou replicar os resultados obtidos por GRANT et al. (2018) nos conjuntos de dados iris e MNIST. Para isso utilizou-se da biblioteca Pennylane para auxiliar na criação e simulação de circuitos quânticos.

Replicou-se o modelo TTN com operadores *simple gate* e rotações reais para as tarefas binárias de classificação do conjunto de dados iris. Conforme visto na Tabela 1, os resultados obtidos na replicação foram similares aos obtidos por GRANT et al. (2018). Pode-se confirmar as conclusões dos pesquisadores de que o modelo TTN com operadores *simple gate* e rotações reais desempenhou bem em todas as três tarefas.

Replicou-se os modelos TTN e MERA no conjunto de dados MNIST. Replicou-se tais modelos com os operadores *simple* e *general gate* e com rotações reais e complexas. Nas quatro tarefas de classificação, os resultados da tarefa "algarismo é maior do que quatro" foram similares aos obtidos por GRANT et al. (2018) em todos os modelos.

As replicações dos modelos nas tarefas "algarismo é par", "algarismo é o zero ou o um" e "algarismo é o dois ou o sete" poderiam ser mais fidedigna aos resultados de GRANT et al. (2018). Suspeita-se que a utilização de hiperparâmetros e vetores de inicialização diferentes influenciaram na replicação dos modelos.

Apesar disso, as conclusões de GRANT et al. (2018) sobre os experimentos no conjunto MNIST foram confirmadas: os modelos com topologia MERA foram superiores aos modelos com topologia TTN em todas as tarefas de classificação; modelos com operadores *general gate* obtiveram acurácia maior que os modelos homólogos com operadores *simple gate* em todas as tarefas de classificação; modelos com operadores com rotação complexa classificaram os exemplos de teste com uma acurácia maior do que os modelos homólogos com rotação real em todas as tarefas de classificação.

Quanto à utilização de *Amplitude Embedding* na preparação de estados quânticos pôde-se confirmar a conclusão de SIERRA-SOSA et al. (2020) de que modelos com *Amplitude Embedding* obtêm uma acurácia maior na classificação do que modelos com *Angle Embedding*. A melhora na acurácia ao utilizar *Amplitude Embedding* ocorreu em todas as tarefas de classificação do conjunto MNIST replicadas neste trabalho. Entretanto, a conclusão de que modelo com *Amplitude Embedding* converge mais cedo do que modelo com *Angle Embedding*, como preparação de estado, foi parcialmente confirmado. Para modelos de topologia TTN, realmente a convergência ocorre com menos lotes de treinamento. O mesmo não pôde ser confirmado para modelos de topologia MERA, onde somente na tarefa

“algarismo é o zero ou algarismo é o um” do modelo MERA com operadores *simple gate* e rotações reais que o modelo convergiu mais cedo ao utilizar *Amplitude Embedding*.

6.1 Contribuições

Este trabalho contribuiu para confirmar as conclusões de GRANT et al. (2018) acerca das topologias TTN e MERA nos conjuntos de dados iris e MNIST.

Também investigou a diferença entre as preparações de estado quântico *Angle Embedding* e *Amplitude Embedding* em modelos hierárquicos. Pode comprovar que houve ganho na acurácia em todos os modelos simulados em todas as tarefas de classificação. E obteve o resultado de que o modelo hierárquico TTN atinge acurácias maiores ao mesmo tempo que converge mais cedo ao utilizar *Amplitude Embedding*.

Por fim, todo o código fonte utilizado desenvolvido neste trabalho encontra-se sobre a licença MIT e disponível publicamente na plataforma Github no endereço github.com/emmanuel-carreira/quantum-computing.

6.2 Trabalhos Futuros

Os seguintes tópicos são sugeridos para um maior aprofundamento:

- Treinar e classificar, em um dispositivo quântico, o modelo TTN com operadores *simple gate* e rotações reais na tarefa de classificação entre as classes 1 (Setosa) ou 2 (Versicolor) do conjunto iris. Essa experimentação visa replicar outro resultado importante obtido por GRANT et al. (2018);
- Replicar os modelos híbridos, de topologia MERA baseada em um modelo da topologia TTN com operadores pré-treinados. Essa replicação de um dos modelos propostos por GRANT et al. (2018) visa comparar a eficácia do modelo MERA pré-treinado com os outros modelos replicados;
- Replicar os modelos com inicialização *Amplitude Embedding* em um dispositivo quântico. Visto que aproxima-se da era *Noisy Intermediate-Scale Quantum* (NISQ) dos dispositivos quânticos, a condensação de características em uma quantidade de *qubits* menor traz uma vantagem e é interessante investigar seu comportamento em um dispositivo quântico físico, além do simulado realizado neste trabalho;
- Replicar os modelos com inicialização *Amplitude Embedding* com 256 componentes principais. Como o circuito quântico de 8 *qubits* permite utilizar até 256 características de entrada, é interessante simular o modelo com *Amplitude Embedding* com 256 componentes principais e analisar o ganho na acurácia em relação ao modelo com *Amplitude Embedding* que utilizou 8 componentes principais;

- Investigar a adição de *bias* nos operadores quânticos. É interessante a investigação desta técnica na aprendizagem de máquina quântica, visto que é uma utilizada na aprendizagem de máquina clássica para evitar que o modelo fique preso em máximos ou mínimos locais;

7. Bibliografia

BARENCO, Adriano; BENNETT, Charles H.; CLEVE, Richard; DIVINCENZO, David P.; MARGOLUS, Norman; SHOR, Peter; SLEATOR, Tycho; SMOLIN, John A.; WEINFURTER, Harald. Elementary gates for quantum computation. **Physical Review A**, [S.L.], v. 52, n. 5, p. 3457-3467, 1 nov. 1995. American Physical Society (APS). http://dx.doi.org/10.1103/physreva.52.3457. Disponível em: https://arxiv.org/abs/quant-ph/9503016. Acesso em: 12 jul. 2021.

CASTANHEIRA, J. et al. **Machine Learning Methods for Radar-Based People Detection and Tracking by Mobile Robots**. Advances in Intelligent Systems and Computing. Anais...2020.

CHEN, C. et al. **DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving**. 2015 IEEE International Conference on Computer Vision (ICCV). Anais...IEEE, dez. 2015.

COLLOBERT, R.; WESTON, J. **A unified architecture for natural language processing: Deep neural networks with multitask learning**. Proceedings of the 25th International Conference on Machine Learning. Anais...2008.

DEVELOPERS, Scikit-Learn. **Scikit-learn**. 2020. Disponível em: https://scikit-learn.org/stable/. Acesso em: 23 jul. 2021.

FEYNMAN, R. P. Simulating physics with computers. **International Journal of Theoretical Physics**, v. 21, n. 6–7, jun. 1982.

FISHER, Ronald. **Iris Data Set**. 1936. Doado por Michael Marshall. Disponível em: https://archive.ics.uci.edu/ml/datasets/iris. Acesso em: 11 jul. 2021.

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, dez.

2018. Springer Science and Business Media LLC. http://dx.doi.org/10.1038/s41534-018-0116-9.

GROVER, L. K. **A fast quantum mechanical algorithm for database search**. Proceedings of the Annual ACM Symposium on Theory of Computing. Anais...1996

KINGMA, D. P.; BA, J. L. **Adam: A method for stochastic optimization**. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. Anais...2015.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. **ImageNet classification with deep convolutional neural networks**. Advances in Neural Information Processing Systems. Anais...2012.

LECUN, Yann; CORTES, Corinna; BURGES, Christopher J.C.. **THE MNIST DATABASE**: of handwritten digits. of handwritten digits. 1998. Disponível em: http://yann.lecun.com/exdb/mnist/. Acesso em: 11 jul. 2021.

LIU, S. et al. **Early diagnosis of Alzheimer's disease with deep learning**. 2014 IEEE 11th International Symposium on Biomedical Imaging, ISBI 2014. Anais...2014.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The Bulletin of Mathematical Biophysics**, v. 5, n. 4, dez. 1943.

MOONEY, R. J.; ROY, L. Content-based book recommending using learning for text categorization. **Proceedings of the ACM International Conference on Digital Libraries**, 2000.

NIELSEN, M.; CHUANG, I. **Quantum Computation and Quantum Information**. 1. ed. Cambridge: Cambridge University Press, 2000.

RIPLEY, B. D. **Pattern recognition and neural networks**. 1. ed. Cambridge: Cambridge University Press, 1996.

SHEN, D.; WU, G.; SUK, H. IL. Deep Learning in Medical Image Analysis. **Annual Review of Biomedical Engineering**, v. 19, 2017.

SHENDE, V.V.; BULLOCK, S.S.; MARKOV, I.L.. Synthesis of quantum-logic circuits. **IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems**, [S.L.], v. 25, n. 6, p. 1000-1010, jun. 2006. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tcad.2005.855930. Disponível em: https://arxiv.org/abs/quant-ph/0406176. Acesso em: 17 jul. 2021.

SHENDE, V.V.; MARKOV, I.L.; BULLOCK, S.S.. Smaller two-qubit circuits for quantum communication and computation. In: DESIGN, AUTOMATION AND TEST IN EUROPE CONFERENCE AND EXHIBITION, 7., 2004, Paris. **Proceedings Design, Automation and Test in Europe Conference and Exhibition**. [S.L.]: IEEE Comput. Soc, 2004. p. 980-985. Disponível em: https://ieeexplore.ieee.org/document/1269020. Acesso em: 17 jul. 2021.

SHOR, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. **SIAM Journal on Computing**, v. 26, n. 5, 1997.

SIERRA-SOSA, Daniel; TELAHUN, Michael; ELMAGHRABY, Adel. TensorFlow Quantum: Impacts of Quantum State Preparation on Quantum Machine Learning Performance. **IEEE Access**. [S.L.], p. 215246-215255. nov. 2020. Disponível em: https://ieeexplore.ieee.org/document/9272350. Acesso em: 18 jul. 2021.

SILVER, D. et al. Mastering the game of Go without human knowledge. **Nature**, v. 550, n. 7676, p. 354-359, 19 out. 2017.

WANG, S. et al. PointTrackNet: An End-to-End Network for 3-D Object Detection and Tracking from Point Clouds. **IEEE Robotics and Automation Letters**, v. 5, n. 2, 2020.

WEIGOLD, Manuela; BARZEN, Johanna; LEYMANN, Frank; SALM, Marie. **Data Encoding Patterns for Quantum Computing**. 2020. Disponível em: https://hillside.net/plop/2020/papers/weigold.pdf. Acesso em: 12 jul. 2021.

WOLD, Svante; ESBENSEN, Kim; GELADI, Paul. Principal component analysis. **Chemometrics And Intelligent Laboratory Systems**, [S.L.], v. 2, n. 1-3, p. 37-52, ago. 1987. Elsevier BV. http://dx.doi.org/10.1016/0169-7439(87)80084-9. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/0169743987800849. Acesso em: 26 jul. 2021.

XANADU. **PennyLane**. 2020. Disponível em: https://pennylane.ai/. Acesso em: 13 jul. 2021.

ZWIEBACH, B.. **DIRAC's BRA AND KET NOTATION**. 2013. Disponível em: https://ocw.mit.edu/courses/physics/8-05-quantum-physics-ii-fall-2013/lecture-notes/MIT8_05F13_Chap_04.pdf. Acesso em: 12 jul. 2021.