UNIVERSIDADE FEDERAL FLUMINENSE

DEPARTAMENTO DE FÍSICA

A MACHINE LEARNING APPROACH TO QUANTUM SPIN CHAINS

LUIZ FELIPE COSTA DE MORAES

ORIENTADOR : PROF. DR. MARCELO SILVA SARANDY

Niterói

4 de outubro de 2019

UNIVERSIDADE FEDERAL FLUMINENSE

DEPARTAMENTO DE FÍSICA

A MACHINE LEARNING APPROACH TO QUANTUM SPIN CHAINS

LUIZ FELIPE COSTA DE MORAES

Dissertação apresentada ao Programa de Pós Graduação de Física da Universidade Federal Fluminense, como parte dos requisitos para a obtenção do título de Mestre em Ciências
Orientador: Prof. Dr. Marcelo Silva Sarandy

Niterói

4 de outubro de 2019

Ficha catalográfica automática - SDC/BIF
Gerada com informações fornecidas pelo autor

C837m Costa de moraes, Luiz Felipe
A MACHINE LEARNING APPROACH TO QUANTUM SPIN CHAINS / Luiz Felipe Costa de moraes ; Marcelo Silva Sarandy, orientador. Niterói, 2019.
71 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense, Niterói, 2019.

DOI: http://dx.doi.org/10.22409/PPGF.2019.m.40097380814

1. Cadeias quânticas de spin. 2. Aprendizado de máquinas. 3. Mecânica quântica. 4. Informação quântica. 5. Produção intelectual. I. Silva Sarandy, Marcelo, orientador. II. Universidade Federal Fluminense. Instituto de Física. III. Título.

CDD -

Bibliotecário responsável: Mario Henrique de Oliveira Castro - CRB7/6155

UNIVERSIDADE FEDERAL FLUMINENSE

DEPARTAMENTO DE FÍSICA

A MACHINE LEARNING APPROACH TO QUANTUM SPIN CHAINS

LUIZ FELIPE COSTA DE MORAES

Dissertação apresentada ao Programa de Pós Graduação de Física da Universidade Federal Fluminense, como parte dos requisitos para a obtenção do título de Mestre em Ciências
Orientador: Prof. Dr. Marcelo Silva Sarandy

Aprovado em 05 de setembro de 2019.

Membros da banca:

Prof. Dr. Marcelo Silva Sarandy

(Orientador– IF-UFF)

Prof. Dr. Alexandre Grezzi de Miranda Schmidt

(Membro interno – ICEx-UFF)

Prof. Dr. Alexandre Martins de Souza

(Membro externo – CBPF)

Niterói

4 de outubro de 2019

A meus pais.

AGRADECIMENTOS

Agradeço a meus pais por todo carinho e apoio incondicional mesmo nos momentos que não poderam estar presentes. Sem a presença deles não estaria onde estou hoje.

Agradeço ao meu orientador, Marcelo Sarandy, por toda a paciência e dedicação que teve comigo ao longo desses dois anos que trabalhamos juntos. Que mesmo com a pressão de três defesas de seus alunos de doutorado encontrava tempo para discutir o meu trabalho.

Agradeço também a professora Andreia pelos dados que foram muito importantes para a conclusão desse trabalho e pelas conversas que fortificaram meu conhecimento sobre o assunto.

Agradeço muito ao professor Clécio por todas as dúvidas esclarecidas e sugestões de caminhos a serem escolhidos para resolver o nosso problema. Devo muito desse trabalho a ele.

Se você não consegue explicar algo de forma simples, voçê não entendeu suficientemente bem.

Albert Einstein

RESUMO

O objetivo deste trabalho é apresentar um método computacional para determinar propriedades físicas de sistemas quânticos de muitos corpos. A abordagem proposta foi um método de Aprendizado de Máquina para lidar com imagens, onde a entrada é o Hamiltoniano do sistema desejado. Nós treinamos uma Rede Neural Convolucional VGG-16 para prever a energia do estado fundamental e o emaranhamento entre pares de partículas, tal como medido pela negatividade, para o Hamiltonianano de um sistema quântico de spins-1/2. Os resultados preveem com boa precisão os valores da energia e o emaranhamento entre pares. Particularmente, podemos reduzir a dimensão do Hamiltoniano antes que ele passe pela rede. O método pode ser generalizado para propriedades físicas arbitrárias.

Palavras-chave: Cadeias quânticas de spin, Aprendizado de máquina, Mecânica quântica, Regressão, Informação quântica

ABSTRACT

The aim of this work is to present a computational method to solve many-body quantum system problems. The proposed approach is a Machine Learning method to deal with images, where the input is the Hamiltonian of the desired system. We trained a VGG-16 Convolutional Neural Network to predict the ground state energy of the XXZ Hamiltonian, as well as the negativity of the pairwise entangled mixed states. The results predict with good precision the values of the energy and the pairwise entanglement. Particularly, we can reduce the dimension of the Hamiltonian before it passes through the network. The method can be generalized to arbitrary physical properties.

Keywords: Quantum spin chains, Machine learning, Quantum mechanics, Regression, Quantum information

LIST OF FIGURES

LIST OF TABLES

CONTENTS

Chapter 1

INTRODUCTION

Quantum mechanics is the theory to describe the microscopic world. It was developed in the beginning of the 20th century by physicists such as M.Plank, A.Einstein, N.Bohr, P.A.M. Dirac, W. Heisenberg, E. Schrödinger (PLANCK, 1901), (EINSTEIN, 1905), (BOHR, 1915),(DIRAC, 1925), (DIRAC, 1926a), (HEISENBERG, 1925), (SCHRÖDINGER, 1930), (EINSTEIN; PODOLSKY; ROSEN, 1935). Quantum mechanics accomplished to solve exactly problems like the hydrogen atom, (DIRAC, 1926b), (PAULING, 1928). However, to solve a more complex system, a model of the system is necessary to approach the different complexity. In (ANDERSON, 1972) it is discussed the scale problem, how making the system bigger influences in the way to solve the problem, and how dealing with it in a different way can change the strategy to approach the problem. For instance solving a molecular problem directly using microscopic nuclear interactions is not a usual clever procedure. To solve different scale processes it is required to devise distinct methods.

Machine learning (ML) in quantum problems appears as a method to provide quantum physical properties through a learning procedure implemented in a computing device. ML has appeared in the middle '50s and had a second boom in the later '80s and early '90s. It then became clear how it could be a powerful way to approach a challenging problem at that time. The methods started to develop in computer science approaches but physicists at that time had already seen the potential and started to apply them to statistical physics, non-linear dynamics, and thermodynamics. The following

references are excellent examples of the work done by physicists in this period of time. (CARNEVALI; PATARNELLO,1987), (GUYON, 1991), (HANSEL; MATO; MEUNIER, 1992), (BöS; KINZEL; OPPER, 1993), (BIEHL, 1994), (BIEHL; SCHWARZE, 1995), (BIEHL; RIEGLER; WöHLER, 1996), (BIEHL; SCHLöSSER; AHR, 1998), (AHR; BIEHL; SCHL öSSER, 1999), (AHR; BIEHL; URBANCZIK, 1999), (APOLLONI; BATTISTINI; FALCO, 1999).

At the end of 2010, the neural networks were well established and largely used even as a commercial product. The computational problems in the early era of the method were overcome with powerful GPU (Graphics Processing Units) and problems with larger dimensions were solvable. Due to the well established method and the computational improvements it became a way to solve quantum problems. (FISCHER; IGEL, 2012), (SNYDER et al., 2012), (LI et al., 2016), (CARRASQUILLA; MELKO, 2017), (LIU et al., 2017), (NIEUWENBURG; LIU; HUBER, 2017), (HIGHAM et al., 2018), (TENG, 2018), (CARRASQUILLA et al., 2019).

Another contribution to the ML approach is that it provides a perspective on how to take advantage of quantum mechanics as a learning tool, which originated the research field called quantum machine learning. It is basically a machine learning quantum algorithm performed on a quantum computer. The quantum computer has quantum bits (qubits) instead of classical bits. The rules of quantum mechanics govern the computational process, providing resources beyond classical computation. With these quantum mechanical properties, quantum computers can efficiently solve classes of problems that are not efficiently solvable in a classical computer. ML in quantum computers is still in early development, but here is some work in this way. (NEVEN et al., 2008), (NEVEN et al., 2009), (LLOYD; MOHSENI; REBENTROST, 2013), (PUDENZ; LIDAR, 2013), (ADCOCK et al., 2015), (SCHULD; SINAYSKIY; PETRUCCIONE, 2015a), (SCHULD; SINAYSKIY; PETRUCCIONE, 2015b), (WECKER; HASTINGS; TROYER, 2016), (ZENG; COECKE, 2016), (BIAMONTE et al., 2017), (NEUKART; DOLLEN; SEIDEL, 2018), (SARMA; DENG; DUAN, 2019).

This work purpose is to present a classical ML approach to solving quantum phys-

ical problems. We apply a neural network that was designed to find patterns in images to find properties of quantum spin chains. The difference for what has been done in the physics community is the input of the network. The difference from the works cited above is that over here we treat the Hamiltonian as an image, which is contrast with the usual procedure of taking the quantum state as the input of the network (CARLEO; TROYER, 2017).

The structure of this dissertation is divided into the following chapters. In chapter 2, we briefly introduce the fundaments of quantum mechanics, writing its postulates for pure and mixed states and showing a quantity to measure entanglement of mixed states with an explicit example. Chapter 3 discusses Machine Learning (ML) methods, starting with a regular Neural Network (NN) and its components, giving a brief explanation on the algorithm steps and how it works. Still in this chapter, we introduce the Convolutional Neural Network (CNN) and present the elements of it. In chapter 4 are the results obtained in this dissertation, the result of the application of ML methods in one-dimensional quantum spin systems. Conclusions are presented in chapter 5, where the results are discussed and shown how it can be useful as practical tools. Moreover, future perspectives in how the problem can be approached in future works are discussed.

Chapter 2

FUNDAMENTS OF QUANTUM MECHANICS

Quantum mechanics is the theory that accomplished to precisely describe the rules that influence the microscopic world. These rules are grounded into four postulates as proposed by (NIELSEN; CHUANG, 2002)

2.1 Pure states

2.1.1 State space

The first quantum mechanics postulate is the one that sets the space where a quantum system lives and how vectors in this space must be.

Postulate 1 *Associated to any isolated physical system is a complete complex vector space with inner product (that is, a Hilbert space) known as the state space of the system. The system is completely described by its state vector, which is a unit vector in the system's state space.*

The simplest state that can be described in this complex vector space, ket space, is a quantum bit (qubit). It is a two-dimensional vector with the orthogonal bases being $|0i$ and $|1i$. So an arbitrary vector is:

$$|\psi i = a|0i + b|1i, \tag{2.1}$$

where a and b are complex numbers.

For each vector there is a one-to-one correspondence to another vector space, the bra space. The vector bra associated with the arbitrary qubit in Eq. (2.1) are:

$$\langle\psi| = a^*\langle 0| + b^*\langle 1|, \tag{2.2}$$

where a^* and b^* are the complex conjugate of a and b respectively.

With both spaces defined the inner product are expressed as:

$$\langle\psi|\psi\rangle = |a|^2 + |b|^2. \tag{2.3}$$

The normalization condition, $\langle\psi|\psi\rangle = 1$, implies that:

$$|a|^2 + |b|^2 = 1. \tag{2.4}$$

2.1.2 Time Evolution

The second postulate tells how the vector of the system changes with time.

Postulate 2 *Schrödinger equation describes the time evolution of the state of a closed quantum system,*

$$i\hbar\frac{d|\psi\rangle}{dt} = H|\psi\rangle. \tag{2.5}$$

In this equation, $\hbar$ is a physical constant known as Planck's constant whose value must be experimentally determined. In practice, it is common to absorb the factor $\hbar$ into H, effectively setting $\hbar = 1$. H is a Hermitian operator (it is equal to the transpose conjugate, $H = H^\dagger$) known as the Hamiltonian of the closed system.

From this postulate, the evolution can be associated with a unitary operator acting

on the system, writing:

$$|\psi(t)i = U(t)|\psi(0)i, \tag{2.6}$$

with $U^\dagger(t)U(t) = \mathbb{I}$, which comes from normalization of $|\psi(t)i$. Indeed, for an arbitrary initial state:

$$h\psi(0)|U^\dagger(t)U(t)|\psi(0)i = 1, \tag{2.7}$$

$$\Rightarrow U^\dagger(t)U(t) = \mathbb{I}. \tag{2.8}$$

For a time independent Hamiltonian, Eq. (2.5) becomes:

$$\begin{aligned} H|\psi(t)i &= i\hbar\frac{d}{dt}|\psi(t)i \\ HU(t)|\psi(0)i &= i\hbar \quad \frac{d}{dt}U(t) \quad |\psi(0)i \\ \Rightarrow HU(t) &= i\hbar \quad \frac{d}{dt}U(t) \quad . \end{aligned} \tag{2.9}$$

The solution for this differential equation for a time-independent Hamiltonian is:

$$U(t) = e^{\frac{-i}{\hbar}Ht}, \tag{2.10}$$

where $e^{At} = \sum_n \frac{(At)^n}{n!}$.

2.1.3 Measurement

The second postulate tells how an quantum system evolves, but it does not express what happens if something interacts through a measurement with it. To describe how an observer interacts with the quantum system, the following postulate is needed.

Postulate 3 *Quantum measurements are described by a collection* $\{M_m\}$ *of measure-*

ment operators. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement then the probability that results m occurs is given by:

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle. \tag{2.11}$$

Also, the state system after the measurement is:

$$|\psi'\rangle = \frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}. \tag{2.12}$$

As $p(m)$ is a probability, the sum of all m must follow:

$$\sum_m p(m) = \sum_m \langle\psi|M_m^\dagger M_m|\psi\rangle = 1. \tag{2.13}$$

So the measurement operators must satisfy the completeness relation:

$$\begin{aligned} \sum_m \langle\psi|M_m^\dagger M_m|\psi\rangle &= 1 \\ \langle\psi| \sum_m M_m^\dagger M_m |\psi\rangle &= 1 \Leftrightarrow \\ \Leftrightarrow \sum_m M_m^\dagger M_m &= I \end{aligned} \tag{2.14}$$

2.1.4 Composite Systems

The last and following postulate describes hows to deal with composite systems and how the state space of the system is described in terms of the isolated ones.

Postulate 4 *The state space of a composite physical system is the tensor product of the*

state spaces of the component physical systems. Therefore, if we have systems numbered 1 *through n, and system number i is prepared in the state* $|\psi_i\rangle$ *, then the joint state of the total system is* $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots |\psi_n\rangle$.

2.2 Mixed states

The necessity of dealing with mixed states comes from the lack of knowledge on the quantum state of the system. In this case there is an ensemble of states $|\psi_i\rangle$ with probability p_i, where the ensemble is the set of all states and probabilities, $\{|\psi_i\rangle, p_i\}$.

A simple example that makes it clear to understand is the famous Stern-Gerlach experiment (GERLACH; STERN, 1922). The experiment consists of a neutral beam of silver particles passing in a non-uniform magnetic field region and observing how the beam deflects by the force of the magnetic fields acting in the dipole moments of the atoms. If a pure state describes the particle beam that leaves the oven, it cannot guarantee that there will be half probability of finding the initial state in $|1/2\rangle$ or $|-1/2\rangle$ for all linear combinations of the bases options, X , Y and Z.

For example, if it is described as a superposition of the Z eigenvectors:

$$|\psi\rangle = \frac{|+\rangle_z + |-\rangle_z}{\sqrt{2}}, \tag{2.15}$$

it will be precisely the state $1/2$ in the X basis

$$|+\rangle_x = \frac{|+\rangle_z + |-\rangle_z}{\sqrt{2}}. \tag{2.16}$$

2.2.1 State space

The first quantum mechanics postulate for mixed states sets the space and the properties of it.

Postulate 1 *Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the state space of the system. The system is completely described by its density operator, which is a positive operator ρ with trace one, acting on the state space of the system. If a quantum system is in the state ρ_i (an ensemble of pure states $\{|\psi_i\rangle\}$) with probability p_i, then the density operator for the system is:*

$$\begin{aligned}\rho &= \sum_i p_i \rho_i \\ &= \sum_i p_i |\psi_i\rangle\langle\psi_i|.\end{aligned} \tag{2.17}$$

2.2.2 Evolution

Analogous to the pure case, the evolution is described by a unitary transformation.

Postulate 2 *The evolution of a closed quantum system is described by a unitary transformation. That is, the state ρ of the system at time t_1 is related to the state ρ' of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2,*

$$\rho' = U\rho U^\dagger, \tag{2.18}$$

or

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i| \xrightarrow{U} \sum_i p_i U|\psi_i\rangle\langle\psi_i|U^\dagger. \tag{2.19}$$

2.2.3 Measurement

Postulate 3 *Quantum measurements are described by a collection $\{M_m\}$ of measurement operators. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is ρ immediately before the measurement*

then the probability that results m occurs is given by

$$p(m) = Tr\left[M_m^\dagger M_m \rho\right], \tag{2.20}$$

also, the system after the measurement is:

$$\rho_m = \frac{M_m \rho M_m^\dagger}{Tr\left[M_m^\dagger M_m \rho\right]}. \tag{2.21}$$

Postulate 3 can be understood from the conditional probability theory applied to the ensemble $\{|\psi_i\rangle, p_i\}$. A conditional probability describes the probability of measure a state at $|\psi_i\rangle$ with an operator M_m and getting the result m:

$$p(m|i) = \langle\psi_i|M_m^\dagger M_m|\psi_i\rangle = Tr\left[M_m^\dagger M_m |\psi_i\rangle\langle\psi_i|\right]. \tag{2.22}$$

Using the law of total probability and the linearity of the trace operation, the probability is:

$$\begin{aligned} p(m) &= \sum_i p(m|i)p_i \\ &= \sum_i Tr\left[M_m^\dagger M_m |\psi_i\rangle\langle\psi_i|\right] p_i \\ &= Tr\left[M_m^\dagger M_m \rho\right], \end{aligned} \tag{2.23}$$

which is in agreement with Eq. (2.20).

After the measurement, a state initially at $|\psi_i\rangle$ after getting the result m will be at:

$$|\psi_i^m\rangle = \frac{M_m|\psi_i\rangle}{\sqrt{\langle\psi_i|M_m^\dagger M_m|\psi_i\rangle}}. \tag{2.24}$$

Given the ensemble of states $|\psi_i^m\rangle$ with conditional probability $p(i|m)$, which is the probability of finding the state after the measurement in m provided its initial was i, we have that the system is described by a density matrix ρ_m, given by

$$\rho_m = \sum_i p(i|m)|\psi_i^m\rangle\langle\psi_i^m| = \sum_i p(i|m)\frac{M_m|\psi_i\rangle\langle\psi_i|M_m^\dagger}{\langle\psi_i|M_m^\dagger M_m|\psi_i\rangle}. \tag{2.25}$$

From Bayes product rule, $p(i|m) = p(m,i)/p(m) = p(m|i)p_i/p(m)$, the density matrix become:

$$\begin{aligned}\rho_m &= \sum_i \frac{p(m|i)p_i}{p(m)}\frac{M_m|\psi_i\rangle\langle\psi_i|M_m^\dagger}{\langle\psi_i|M_m^\dagger M_m|\psi_i\rangle} \\ &= \frac{M_m\rho M_m^\dagger}{Tr\left(M_m^\dagger M_m\rho\right)}.\end{aligned} \tag{2.26}$$

2.2.4 Composite Systems

Postulate 4 *The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Therefore, if we have systems numbered* 1 *through n, and system number i is prepared in the state* ρ_i*, then the joint state of the total system is* $\rho_1 \otimes \rho_2 \otimes \cdots \rho_n$.

2.3 Entanglement

Entanglement arises in quantum mechanics theory as a consequence of the superposition principle applied to two or more qubits. It was firstly pointed out by (EINSTEIN; PODOLSKY; ROSEN, 1935) and (SCHRÖDINGER, 1935), evidencing a unique feature of quantum mechanics. More specifically, entanglement may reveal non-local aspects of the quantum world. A simple way to see it is looking at the singlet state

$$|\psi\rangle = \frac{|01\rangle - |10\rangle}{\sqrt{2}}. \tag{2.27}$$

This state implies that a measurement in one qubit guarantees the knowledge of the

other, independently of the distance that separates those qubits.

2.3.1 Pure states

Entanglement is a synonym of a non-separable state. A separable pure state is a state the can be written as a tensor product of the systems that compose it, for an N part system,

$$|\psi_{SEP}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_N\rangle. \tag{2.28}$$

So an entangled pure state is the one that cannot be written in a separable form, namely,

$$|\psi\rangle \neq |\psi_{SEP}\rangle. \tag{2.29}$$

For bipartite pure states, we can obtain a criterion for entanglement through the Schmidt decomposition. (NIELSEN; CHUANG, 2002)

Theorem 1 *(Schmidt decomposition) Suppose $|\psi\rangle$ is a pure state of a composite system, AB. Then there exist orthonormal states $|i_A\rangle$ for system A, and orthonormal states $|i_B\rangle$ of system B such that*

$$|\psi\rangle = \sum_i \lambda_i |i_A\rangle |i_B\rangle, \tag{2.30}$$

where λ_i are non-negative real numbers satisfying $\sum_i \lambda_i^2 = 1$ known as Schmidt coefficients.

For two systems A and B, with dimensions m and n, respectively, and basis vectors $\{|j\rangle\}$ and $\{|k\rangle\}$, respectively, the quantum composite system for AB can be written as,

$$|\psi\rangle = \sum_{jk} a_{jk} |j\rangle|k\rangle, \tag{2.31}$$

where $a_{j,k}$ are the complex elements of an $m \times n$ matrix. Using Singular Value Decomposition (SVD) this matrix can be written as $a = U(m \times m)D(m \times n)V^{\dagger}(n \times n)$, two unitary matrices, U and $V^{\dagger}$ and one real matrix, D. The state can be rewritten as:

$$|\psi\rangle = \sum_{ijk} u_{ji} d_{ii} v_{ik} |j\rangle|k\rangle. \tag{2.32}$$

Defining a new base vectors for A and B,

$$\begin{aligned} |i_A\rangle &\equiv \sum_j u_{ji} |j\rangle \\ |i_B\rangle &\equiv \sum_k v_{ik} |k\rangle. \end{aligned} \tag{2.33}$$

By defining $d_{ii} \equiv \lambda_i$ the state becomes:

$$|\psi\rangle = \sum_{i}^{\min(m,n)} \lambda_i |i_A\rangle|i_B\rangle. \tag{2.34}$$

From the theorem, any bipartite pure state can be written in Schmidt decomposition form. A state that has more than than one term in the sum is said a non-separable state or an entangled state.

2.3.2 Mixed states

The mixed state definition for a separable state is written in the following form:

$$\rho = \sum_{r=1}^{d} p_r \rho_r^1 \otimes \cdots \otimes \rho_r^N, \tag{2.35}$$

where it can be thought as a system composed of N parts. Each part can be constructed with local operations through the local density matrix ρ_r^i, where r label the possible operator in the subsystem $i \in [1, N]$. To construct it, the parameter $r \in [1, d]$ is generated with probability p_r and distributed classically to all the N parts, as shown in Table (2.1). For an ensemble, the system is the one in Eq. (2.35)

Initial state:	$\|0\rangle\langle 0\|$	$\otimes$	$\|0\rangle\langle 0\|$	$\otimes \cdots \otimes$	$\|0\rangle\langle 0\|$
Local operation:	ρ_r^1	$\otimes$	ρ_r^2	$\otimes \cdots \otimes$	ρ_r^N
classical communication:		Random number generator(r)			

Table 2.1: Scheme to generate a separable mixed state

2.3.3 Negativity

Negativity is a measurement of entanglement for arbitrary 2×2 or 2×3 bipartite systems, proposed by (PERES, 1996)(HORODECKI, 1997). It is a monotonic function for the eigenvalues in local operation and quantifies the degrees of entanglement of the composite system. It was proposed by (VIDAL; WERNER, 2002) inspired by the idea of (PERES, 1996) to investigate positive maps under partial transposition, and rewritten in the following form by (VERSTRAETE; AUDENAERT; MOOR, 2001),(WU; SARANDY; LIDAR, 2004)

$$\mathcal{N}(\rho) = 2\max\left[0, -\min_m(\lambda_m)\right], \tag{2.36}$$

where the λ_m are the eigenvalues of ρ^{T_A}, that is the partial transpose in one of the subsystems. The partial transpose adopted here is the transposition in the first subsystem, i.e.

$$\langle\alpha\beta|\rho^{T_A}|\gamma\delta\rangle = \langle\gamma\beta|\rho|\alpha\delta\rangle, \tag{2.37}$$

The values of the negativity vary continuously in the range $[0, 1]$, where 0 denotes a separable state (non-entangled mixed state) and 1 denotes a non-separable state (entangled state). The reason is that it measures when the ρ^{T_A} is negative, and for a separable state, the positivity is a necessity.

Notice that the partial transpose density matrix is a Hermitian operator

$$\begin{aligned} \langle \alpha\beta | \rho^{T_A} | \gamma\delta \rangle^* &= (\langle \eta\beta | \rho | \alpha\delta \rangle)^* \\ &= \langle \alpha\delta | \rho^\dagger | \gamma\beta \rangle \\ &= \langle \alpha\delta | \rho | \gamma\beta \rangle \\ &= \langle \eta\delta | \rho^{T_A} | \alpha\beta \rangle \\ \Rightarrow \rho^{T_A} &= \left(\rho^{T_A}\right)^\dagger, \end{aligned} \tag{2.38}$$

If the composite bipartite system (A and B) is separable, the density matrix can be written as Eq. (2.35),

$$\begin{aligned} \rho_{AB}^{T_A}{}_{\alpha\beta;\gamma\delta} &= \sum_i p_i (\rho_A)_{\gamma\alpha} \otimes (\rho_B)_{\beta\delta} \\ \Rightarrow \rho_{AB}^{T_A} &= \sum_i p_i \left(\rho_A^{T_A}\right) \otimes (\rho_B), \end{aligned} \tag{2.39}$$

where $\rho_A^{T_A}$ is a non-negative matrix with trace equal to 1. The condition to be a separable state is non-negative eigenvalues, so to be non-separable it must have negative eigenvalues.

A good way to exemplify how to calculate the negativity is using the Werner state (WERNER, 1989):

$$\rho = \frac{(1-\lambda)}{4} \mathbb{I}_{2\times 2} + \lambda |\psi\rangle\langle\psi|, \tag{2.40}$$

where the state $|\psi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$ and $0 \leq \lambda \leq 1$.

The matrix representation of the Werner state is:

$$\rho = \begin{bmatrix} \frac{(1-\lambda)}{4}+\frac{\lambda}{2} & 0 & 0 & \frac{\lambda}{2} \\ 0 & \frac{(1-\lambda)}{4} & 0 & 0 \\ 0 & 0 & \frac{(1-\lambda)}{4} & 0 \\ \frac{\lambda}{2} & 0 & 0 & \frac{(1-\lambda)}{4}+\frac{\lambda}{2} \end{bmatrix} = \begin{bmatrix} \frac{(1+\lambda)}{4} & 0 & 0 & \frac{\lambda}{2} \\ 0 & \frac{(1-\lambda)}{4} & 0 & 0 \\ 0 & 0 & \frac{(1-\lambda)}{4} & 0 \\ \frac{\lambda}{2} & 0 & 0 & \frac{(1+\lambda)}{4} \end{bmatrix}. \tag{2.41}$$

To calculate the negativity, we use the eigenvalues of the partial transpose density matrix ,

$$\rho^{T_A} = \begin{bmatrix} \frac{(1+\lambda)}{4} & 0 & 0 & 0 \\ 0 & \frac{(1-\lambda)}{4} & \frac{\lambda}{2} & 0 \\ 0 & \frac{\lambda}{2} & \frac{(1-\lambda)}{4} & 0 \\ 0 & 0 & 0 & \frac{(1+\lambda)}{4} \end{bmatrix}. \tag{2.42}$$

To calculate the eigenvalues of it only the middle block needs to be diagonalized, so it becomes:

$$\begin{vmatrix} \frac{(1-\lambda)}{4}-\gamma & \frac{\lambda}{2} \\ \frac{\lambda}{2} & \frac{(1-\lambda)}{4}-\gamma \end{vmatrix} = 0, \tag{2.43}$$

the determinant implies in the following equation of λ,

$$\Rightarrow \frac{(1-\lambda)^2}{16}+\gamma^2-\frac{(1-\lambda)}{2}\gamma-\frac{\lambda^2}{4}=0. \tag{2.44}$$

Solving for γ the four eigenvalues of it are:

$$\begin{aligned}\gamma_1 &= \frac{1+\lambda}{4} \\ \gamma_2 &= \frac{1-3\lambda}{4} \\ \gamma_3 &= \frac{1+\lambda}{4} \\ \gamma_4 &= \frac{1+\lambda}{4}.\end{aligned} \tag{2.45}$$

For the values of λ the minimum eigenvalue is γ_3 and the negativity is given by:

$$\boldsymbol{N}(\rho) = 2\max\left[0, \frac{3\lambda-1}{4}\right]. \tag{2.46}$$

or,

$$\begin{cases} \boldsymbol{N} = 0, & 0 \leq \lambda \leq 1/3 \\ \boldsymbol{N} = \frac{3\lambda-1}{2}, & \lambda \geq 1/3. \end{cases} \tag{2.47}$$

So, Werner states with λ values greater than $1/3$ are entangled, as shown in Fig. 2.1

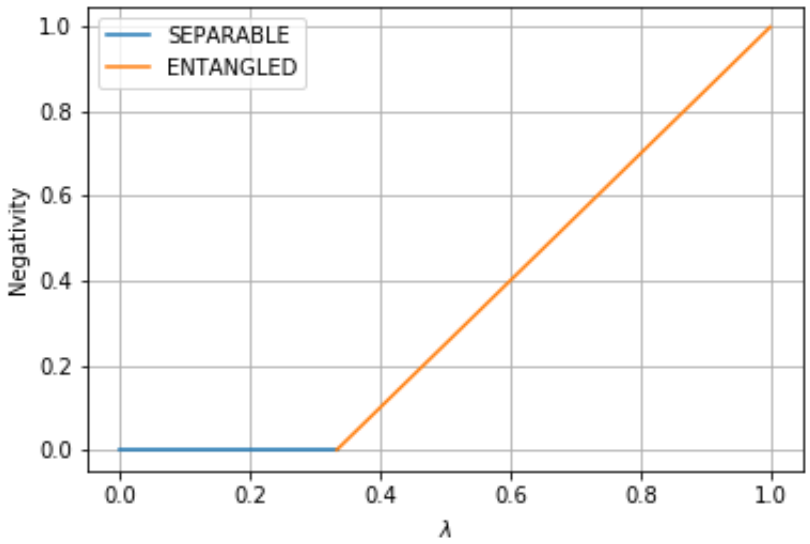


Figure 2.1: Negativity for a Werner state

Chapter 3

MACHINE LEARNING

The aim of this chapter is to introduce the computational tools used in the approach to solve the physical problem. ML is a kind of a new topic among physicists, so this is a significant part of this dissertation. In this chapter, the main calculations were based on (GOODFELLOW; BENGIO; COURVILLE, 2016; NIELSEN; BENGIO; COUVILLE, 2017).

ML is a subset of artificial intelligence and, as its name implies, the objective of it is to, in a sense, be able to learn. Artificial intelligence is a set of algorithms that adapt to the general input function and change the output based on the input. Usual algorithms are done to overcome one kind of problem explicitly so it can only be used to solve that task. ML methods approach the problem differently, it uses large samples as input and from them generalize the abstract pattern of the data. With this pattern, the algorithm can predict the output of a new input that belongs to the data set of the problem but has not been seen by the generalizing code (also known as training process).

There are two different ways to deal with a problem using ML, either supervised or unsupervised learning. In the supervised approach, the input is already labeled, and we know the values of the output for each input. The work in this process is to label the data as previously done by humans. If the input-output set is the pair (x_i, y_i), where x_i is the i-th input and y_i the i-th output, the human part is to label each x_i with the respective y_i. In unsupervised learning we do not know any output a priori and want the algorithm to predict it without human influence. As the problem in this dissertation is based on a

supervised method, I have chosen to focus in this approach.

Many problems can be solved using ML, but we have chosen to discuss two kinds: classification and regression. In the first one, the program is asked to set each input into one of k finite categories. So its task is to produce a function $f : R^n \rightarrow \{1, \cdots k\}$, where n is the input dimension. A problem that can be solved using this is the handwritten digits discussed in (LECUN et al., 1989). In this problem, each input is an image of handwritten digits, and the output can be categorized in ten classes, each one representing a number between 0 and 9. Concerning the regression method, the aim is to predict a value for each input, which defines a function $f : R^n \rightarrow R$. The difference from the classification problem is the output dimension; in classification, the output dimension is a discrete number of classes and in regression is a continuous set. An example of a regression problem is to predict a house price using the area of it as input.

3.1 Artificial Neural Networks

Artificial Neural Networks are an abstract structural way to organize a set of elements that holds a number, that structure can learn from data. It are, also known as Neural Networks (NN), are called that way because they were inspired by brain neural networks in the early beginning of the field. Frank Rosenblatt (ROSENBLATT, 1958), inspired by the work of (MCCULLOCH; PITTS, 1943), constructed a well known type of NN, Perceptrons, which are shown in Fig. (3.1). Perceptron is a binary neuron, its output is either 0 or 1. The lines in the figure are the input of the neuron and the neuron are the circles, the rectangular box represents the Step function that makes it binary.

3.1.1 Components of an NN

In Fig (3.2), we show the essential elements of an NN and how they interact with each other. The left circles are the input values. They are organized as a vector with n components. Their arrangement in the figure are a vector, but they can be more general

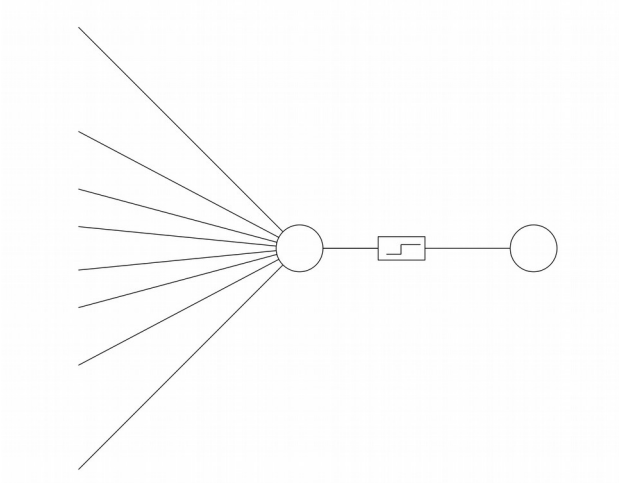


Figure 3.1: Perceptron

structures. If the input x is a matrix, we still can use that vector arrangement, by rearranging it into a vector. If the matrix has m rows and n columns, the vector will have $n \times m$ rows. It can be a tensor as well, like a three channel image (Red-Green-Blue, RGB, colored images), with m lines, n columns and k channels. The input vector will have $m \times n \times k$ lines.

The lines connecting the circles represent the weights, which are free parameters adjusted along the layers to predict the right answer in the output. Each of it can be thought as the importance of the previous input (left side of the line) when calculating the next layer element (right side of the line). To calculate the element in the next layer, we combine all the previous inputs with its respective weight and bias in the following function:

$$z_1^{(1)} = \sum_{i=1}^{n} w_{1n}^{(1)} a_n^{(0)} + b_1, \tag{3.1}$$

where the upper label refers to the layer of the network and the lower labels to which neuron of the layer. In Eq. (3.1), $z_1^{(1)}$ is calculated based on all the inputs $a_n^{(0)}$ and each respective weight $w_{1n}^{(1)}$. The term b_1 is called bias, and as it happens for the weights, it corresponds to a free parameter to be changed as the training process occurs. Eq. (3.1)

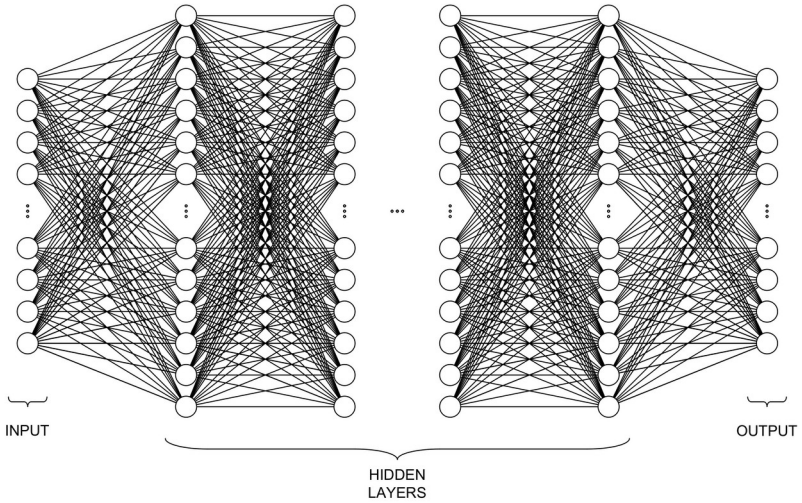
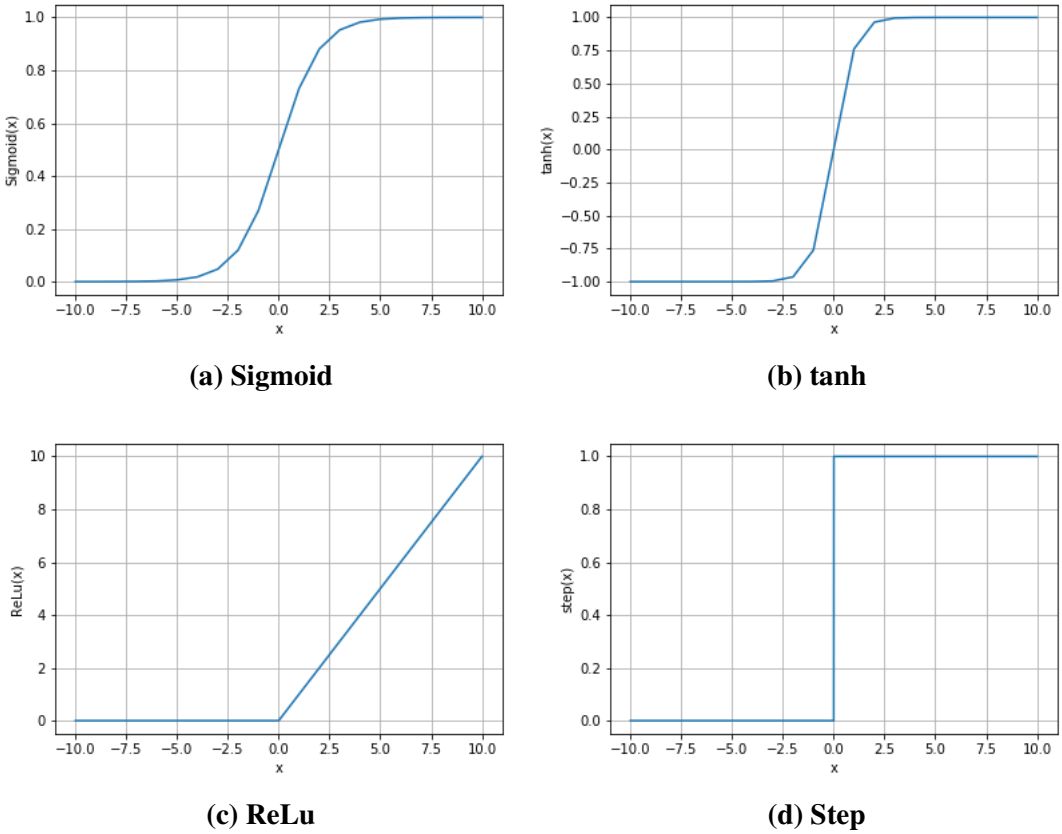


Figure 3.2: General NN

does not represent the value of the next neuron, if it were, the network would be limited to be built as a linear combination of the input. So to deal if that, and to be able to solve more complex problems, a nonlinear function f is applied in the expression. That function is called activation function, which is then used to define the input values of the next layer as,

$$\begin{aligned} a_1^{(1)} &= f\left(z_1^{(1)}\right) \\ &= f\left(\sum_{i=1}^{n} w_{1n}^{(1)} a_n^{(0)} + b_1^{(1)}\right). \end{aligned} \tag{3.2}$$

Examples of activation functions typically used are expressed in Fig. 3.3



(a) Sigmoid **(b) tanh**

(c) ReLu **(d) Step**

Figure 3.3: Usual Activation Functions

All the layers between the input and the output are the hidden layers, hidden because we do not have precise control over them. To calculate a neuron value at layer l, $a_j^{(l)}$, using the values of the previous layer, it is then used,

$$a_j^{(l)} = f\left(\sum_{i=1}^{n} w_{jn}^{(l)} a_n^{(l-1)} + b_j^{(l)}\right). \tag{3.3}$$

Eq. (3.4) can be rearranged in to a vector form, reading,

$$a^{(l)} = f\left(W^{(l)} a^{(l-1)} + b^{(l)}\right), \tag{3.4}$$

where $a^{(l)}$ and $b^{(l)}$ are vector whose components are $a_j^{(l)}$ and $b_j^{(l)}$, respectively, and $W^{(l)}$

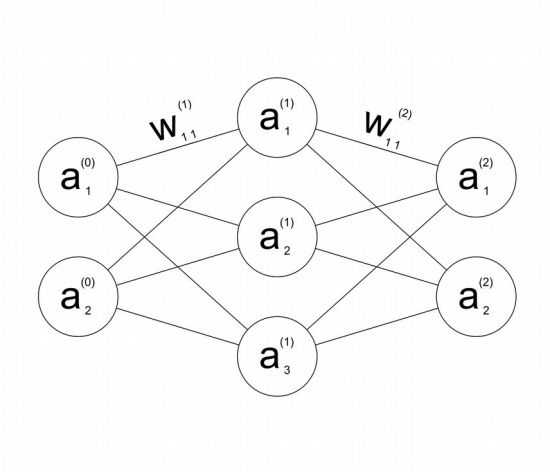


Figure 3.4: NN with 2 input values

is a weights matrix with components $w_{jn}^{(l)}$.

The last part is the output, the prediction values of the NN. After calculating all the neurons values using Eq. (3.4), the last layer will be the output of the network. The array will be compared with the real values, y, and then can be evaluated.

3.1.2 Feedforward example

Feedforward is the process of passing through the input x_i (where i is the i-th training example) in the neural net and update each neuron value to obtain the final output, $a^{(L)}$ (L represents the last layer). It is the first part of the training process, which corresponds to pass the data to generate the output. In the first try, the predicted answer will not be usually correct, since the weights and biases start with random values. In this stage, the output for each training example is generated, the part of changing the parameters to generate the correct answer is the backpropagation that will be the next section.

In order to describe feedforward let us see a simple example where the input and the output have two values, and there is a single hidden layer with three neurons in it, as shown in Fig. (3.4). By properly labelling the input values

$$\begin{aligned} x_1 &= a_1^{(0)} \\ x_2 &= a_2^{(0)}, \end{aligned} \tag{3.5}$$

and calculating the next layer function we have,

$$\begin{aligned} z_1^{(1)} &= w_{1,1}^{(1)} a_1^{(0)} + w_{1,2}^{(1)} a_2^{(0)} + b_1^{(1)}, \\ z_2^{(1)} &= w_{2,1}^{(1)} a_1^{(0)} + w_{2,2}^{(1)} a_2^{(0)} + b_2^{(1)}, \\ z_3^{(1)} &= w_{3,1}^{(1)} a_1^{(0)} + w_{3,2}^{(1)} a_2^{(0)} + b_3^{(1)}. \end{aligned} \tag{3.6}$$

Applying a nonlinear function in Eq. (3.6) we get the output of which neuron:

$$\begin{aligned} a_1^{(1)} &= f\left(z_1^{(1)}\right), \\ a_2^{(1)} &= f\left(z_2^{(1)}\right), \\ a_3^{(1)} &= f\left(z_3^{(1)}\right). \end{aligned} \tag{3.7}$$

Then by using the results of the first layer, the second and final layer is:

$$\begin{aligned} z_1^{(2)} &= w_{1,1}^{(2)} a_1^{(1)} + w_{1,2}^{(2)} a_2^{(1)} + w_{1,3}^{(2)} a_3^{(1)} + b_1^{(2)}, \\ z_2^{(2)} &= w_{2,1}^{(2)} a_1^{(1)} + w_{2,2}^{(2)} a_2^{(1)} + w_{2,3}^{(2)} a_3^{(1)} + b_2^{(2)}. \end{aligned} \tag{3.8}$$

Therefore, the output values in the final layer are given by,

$$\begin{aligned} a_1^{(2)} &= f\left(z_1^{(2)}\right), \\ a_2^{(2)} &= f\left(z_2^{(2)}\right). \end{aligned} \tag{3.9}$$

From the output of the final layer, we can evaluate the network performance. This can be accomplished by defining a loss function ($\boldsymbol{L}$) that compares each output with the correct answer y_i. To evaluate the performance of the NN, we compose theses loss function to a new function, denoted as the cost function C_1 given by:

$$C_1 = \boldsymbol{L}\left(a_1^{(2)}, y\right) + \boldsymbol{L}\left(a_2^{(2)}, y\right). \tag{3.10}$$

3.1.3 Backpropagation

Backpropagation was popularised in this area by the article (RUMELHART; HINTON; WILLIAMS, 1985). It is an algorithmic way to change the weights and bias to make the computer ”learn” how to predict the correct output. In the Feedforward, we end up with a cost function that tells how good the network was to predict a simple example. To properly correct it, we average it over all the training examples, namely,

$$C = \frac{1}{m}\sum_{i=1}^{m} C_i, \tag{3.11}$$

C_i is the individual cost, as provided by an equation such as in Eq. 3.10 and m is the number of training examples. This cost is a function of all the parameters ($w_{k,j}^{(l)}$, $b_j^{(l)}$; $C = C(\mathbf{w}, \mathbf{b})$). In order to minimize it, we must change it in the direction that minimizes the cost. This can be achieved by evaluating $-\nabla C$, where ∇ denotes the gradient operation. This will tell in which direction we need to take a step in the parameters hyperspace. The basic idea of it is to take a step in the direction that minimizes the parameter, which is the gradient descent:

$$\begin{aligned} w &= w - \eta \nabla C, \\ b &= b - \eta \nabla C. \end{aligned} \tag{3.12}$$

where η is the constant that tells the step of the next weight value in the convergence direction, also known as learning rate. w and b over here represent the weights and biases of the layer that are being calculated.

The backpropagation computes each layer error starting in the last one and ending in the first, to calculating the cost function derivatives to respect to the parameters. The name came because the calculation of the error is done by going back layer by layer. The derivation of the set of equations ruling can be found in (NIELSEN; BENGIO; COUVILLE, 2017). They are:

$$
\begin{aligned}
\delta^L &= \nabla_a C \quad f'(z^L), \\
\delta^l &= \left(w^{l+1}\right)^T \delta^{l+1} \quad f'(z^l), \\
\frac{\partial C}{\partial b^l_j} &= \delta^l_j, \\
\frac{\partial C}{\partial w^l_{jk}} &= a^{l-1}_k \delta^l_j.
\end{aligned}
\tag{3.13}
$$

In the first equation in Eq. (3.13) the operation $\odot$ is the Hadamard product, a vector operation that multiplies corresponding elements of the vector (or, more generally, of matrices) without changing the dimension of it.

$$
\begin{aligned}
\mathcal{A} &\equiv (a_1, a_2) \\
\mathcal{B} &\equiv (b_1, b_2)
\end{aligned}
\qquad \Longrightarrow \qquad \mathcal{A} \odot \mathcal{B} \equiv (a_1 b_1, a_2 b_2). \tag{3.14}
$$

∇_a represents derivatives with respect to each neuron in the layer that are being calculated, each component is the derivative with respect of one neuron a. f' is the derivative to the activation function.

3.2 Convolutional Neural Networks

Convolution Neural Networks (CNN) appeared as a way to deal with old problems in NN. To solve a complex issue using conventional NN is easy to get a large number of parameters to minimize and, with it, a requirement of good hardware to computationally minimize it . In (LECUN et al., 1989), Lecun applied CNN on the handwritten digits problems. The basic difference is that the input in a CNN is not a vector any more, the matrix structure is maintained. The convolutional name came from the operation that acts on the input that is similar to a mathematical convolution. The function is called cross-correlation and has the form:

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i+m, j+n) K(m,n), \tag{3.15}$$

where I is the image, K the kernel (or filters) that acts in the image, and S the operation result. The kernel here is analogous to the weights in the NN, it corresponds to the free parameters to adjust an predicts the correct answer. The image corresponds to the input of the CNN, where here it maintains its matrix structure in the convolutional stage. The image has dimension $(i+m, j+n)$, and the kernel (m,n).

The weights in a regular NN are independent of each other, and they act only in one input value. It does not occur in CNN because the kernel passes through the entire image, it slides in the data input multiplying all of it as Eq. (3.15) shows, and each element of it sees and is influenced by the whole input. Because the way the convolution occurs, it is invariant by small changes in the input, with each kernel working as a feature detector in the image. To detect more than one parameter the convolution is done with more than one kernel, so that the result of the operation is a tensor with one of the dimensions being the number of filters applied.

In Fig. (3.5), it is shown the general structure of a CNN, after the convolution stage, the next operation is called pooling. This stage, the dimension of the data is changed, reducing it. This sequence of convolution and pooling is done repeatedly, the hidden part, and in the end, the data are transformed into a vector and pass through a regular NN.

3.2.1 Components of a CNN

As said before, the input of this kind of network is a matrix, which does not need to be converted into a vector as regular NN. The first stage is the convolution, where a set of filters (kernels) is applied in the input ruled by Eq. (3.15). The filter's elements are the weights that multiply the input values. An example is shown in Fig. 3.6. As more than one filter is applied the dimension of the input changes, each one will generate a new matrix, so the numbers of filters will tell the dimension of the convolution results.

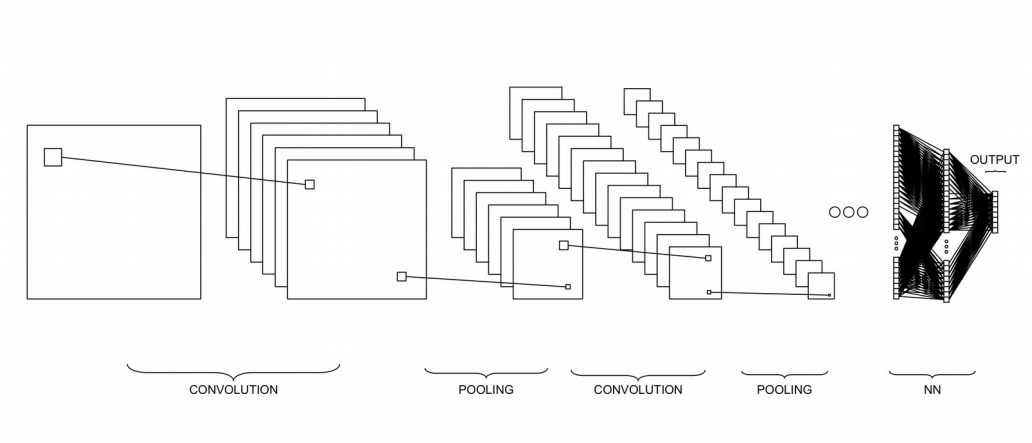


Figure 3.5: General CNN

The size of the matrix changes as well, the way it will change depending on the filter size and how it will ”slide” into the input. After the convolution operation, a bias is added to each convolution, and a nonlinear activation function is applied

The second element of CNN is the pooling stage. This stage makes the structure invariant over small local changes in the input data. The pooling progressively reduces the size of the previous operation, reducing the numbers of parameter in the network. In Fig. (3.7) is an example of a max pool operation, with a 2×2 filter that selects over the elements by the maximum value as realized. Operations other than maximum are also possible, such as average or minimum.

This sequence of convolution and pooling is the feature detection stage. After doing it, the data pass through a regular NN, a fully connected layer, usually with a small number of layers before the output. The output depends on the kind of problem the CNN is solving as the NN case.

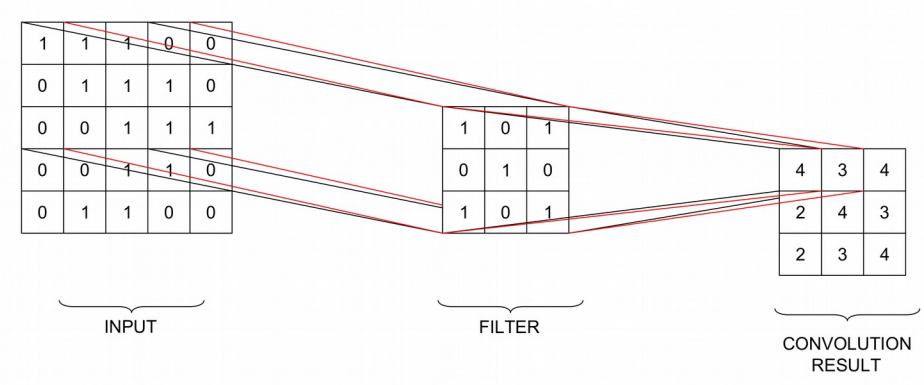


Figure 3.6: Convolutional layer

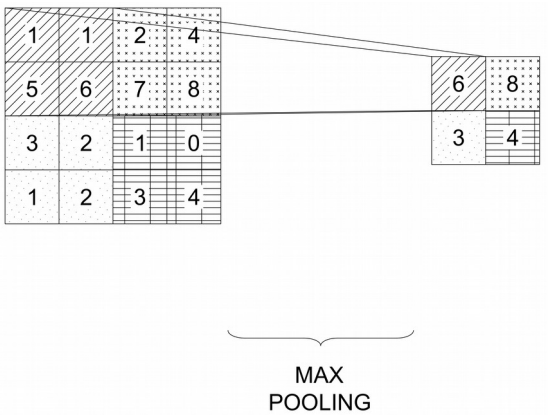


Figure 3.7: Pooling layer

Chapter 4

MACHINE LEARNING IN QUANTUM SPIN CHAINS

This chapter joins the two previous ones. It provides applications of ML to describe physical properties of quantum systems. The underlying systems to be deal with using the ML approach are quantum spin chains, which will be introduced in the following section.

4.1 Quantum Spin Chains

Quantum spin chains arise to successfully describe magnetic properties of materials with quantum mechanics fundaments. It associates the interaction of angular momenta of the electron on the incomplete shell with macroscopic phenomena like ferromagnetism, first proposed by (HEISENBERG, 1927) (DIRAC, 1929). The spin $1/2$ Heisenberg model has been an important model to describe the magnetic properties of materials. In the early '30s Bethe found an analytical solution to the one dimensional case with nearest neighbours coupling (BETHE, 1931), this starts the study of analytical quantum many-body systems.

The one dimensional Heisenberg spin chain Hamiltonian with anisotropy in Z direction, XXZ model, is expressed in function of Pauli operators,

$$H = \sum_{i}^{L} J \left(\hat{\sigma}_i^x \hat{\sigma}_{i+1}^x + \hat{\sigma}_i^y \hat{\sigma}_{i+1}^y \right) + J_z \hat{\sigma}_i^z \hat{\sigma}_{i+1}^z \ , \tag{4.1}$$

where J and J_z are coupling constants and the Pauli operators are,

$$\begin{aligned} \hat{\sigma}_x &\equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \\ \hat{\sigma}_y &\equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \\ \hat{\sigma}_z &\equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \end{aligned} \tag{4.2}$$

4.2 The proposed method

There are innumerable known ways to solve a problem using ML, but the first important thing is to define the input and output values of the problem. In quantum mechanics, the Hamiltonians are mathematically represented by matrices. Therefore it can be a valid input of a CNN, since a matrix is computationally identical to a single-channel image, as shown in Fig. (4.1). A famous and well know CNN architecture to deal with images is then chosen to approach the problem, the VGG16 (SIMONYAN; ZISSERMAN, 2014). It was originally applied in a classification problem, where it could classify a large set of classes, but the approach here is to use it to resolve regression problems. The problem will be to associate with each matrix in the sample set a single number. In particular, it can be the energy of the ground state of the Hamiltonian (the eigenvalue with the lowest value) for a different size and magnetic field. To change from classification to regression, we just change the output dimension, and the function to compute the difference from the real value. The output of classification is the number of classes, and regression is just a single value so that the last layer will have just one neuron now. A activation function frequently used in classification is softmax, a multidimensional sigmoid function, and it is substituted by a mean squared error (MSE)

function, a usual loss to resolve regression problems.

The scheme of the used VGG16 architecture is represented in the Table 4.1. This architecture is used to solve the four distinct Hamiltonian described in the sections below, and all will have the same input dimension. The inputs has dimensions $(256 \times 256 \times 1)$ and in the first layer 64 convolutions of (3×3) filters are applied in it with a non linear activation function ReLu. This exact process is repeated two more times. The next operation is the Maxpooling (2×2), where the data size is compressed by a factor 2. This block of operations, some number of convolutions (3×3) with some filters and activation function ReLu followed by Maxpooling (2×2), is repeated four more times. The first are two convolution layers with 128 filters each, than Maxpooling. The second process is composed by three convolution layers with 256 filters, then Maxpoling. The two lasts are equal, three convolution layers with 512 filters, then the last Maxpooling. After this set of convolutions-max pool, the data is converted into a vector, expressed in Table 4.1 as a Flatten operation, and pass through a regular NN, in this case, two layers of 4096 neurons in each. In the end, a single output is compared to the real value using mean squared error.

4.2.1 Homogeneous Magnetic Field

The first Hamiltonian to be solved is the 8-spins XXZ chain in the presence an external magnetic field in the Z direction. Because the Hilbert space of the system grows exponentially with the number of spins, it will be a $(256 \times 256) = (2^8 \times 2^8)$ matrix. It is written in terms of spin operators as

$$H(h) = -\frac{1}{2}\sum_{j}^{8} \left[\sigma_j^x \sigma_{j+1}^x + \sigma_j^y \sigma_{j+1}^y - \sigma_j^z \sigma_{j+1}^z + h\sigma_j^z \right] . \tag{4.3}$$

The output of it is the energy of the ground state. It can be calculated by diagonalizing the Hamiltonian, and it will be the lowest eigenvalue. The Hamiltonian is a function of h, so for different values of it, there will be a single Hamiltonian and with it

Input ($256 \times 256 \times 1$)
64 Convolution (3×3), activation ReLu
64 Convolution (3×3), activation ReLu
64 Convolution (3×3), activation ReLu
Maxpooling (2×2)
128 Convolution (3×3), activation ReLu
128 Convolution (3×3), activation ReLu
Maxpooling (2×2)
256 Convolution (3×3), activation ReLu
256 Convolution (3×3), activation ReLu
256 Convolution (3×3), activation ReLu
Maxpooling (2×2)
512 Convolution (3×3), activation ReLu
512 Convolution (3×3), activation ReLu
512 Convolution (3×3), activation ReLu
Maxpooling (2×2)
512 Convolution (3×3), activation ReLu
512 Convolution (3×3), activation ReLu
512 Convolution (3×3), activation ReLu
Maxpooling (2×2)
Flatten
Full connected Layer with 4096 neurons
Full connected Layer with 4096 neurons
1 output MSE

Table 4.1: Architecture of the Convolutional Neural Network

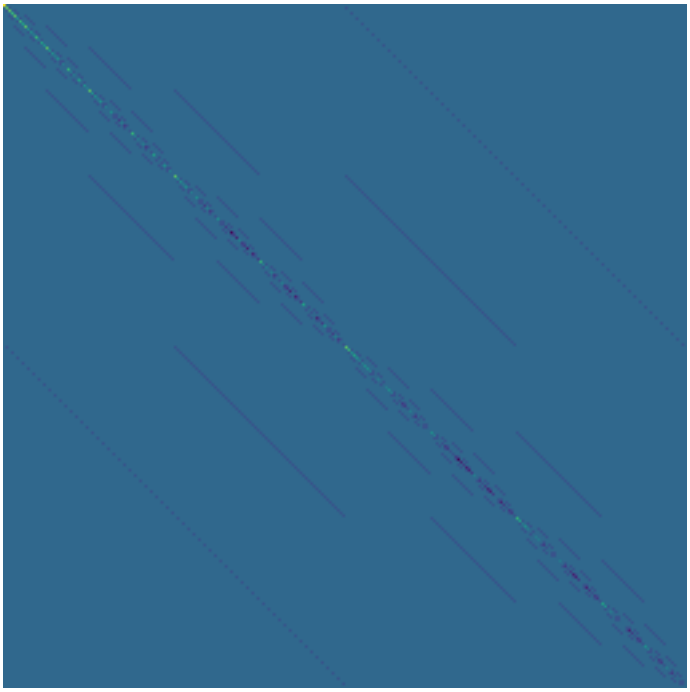


Figure 4.1: Hamiltonian of a 8 spin system represented as an $(256 \times 256 \times 1)$ **image.**

a different ground state energy.

Data sets were generated and divided into three different examples to evaluate how the input size of the training set affects the predictive power of the method. To generate the different Hamiltonians, the domain, h, was set to be $[-8, 8]$, so all the magnetization zones were accessible. In Fig. 4.2 are the data set representation where each point on it representing a matrix.

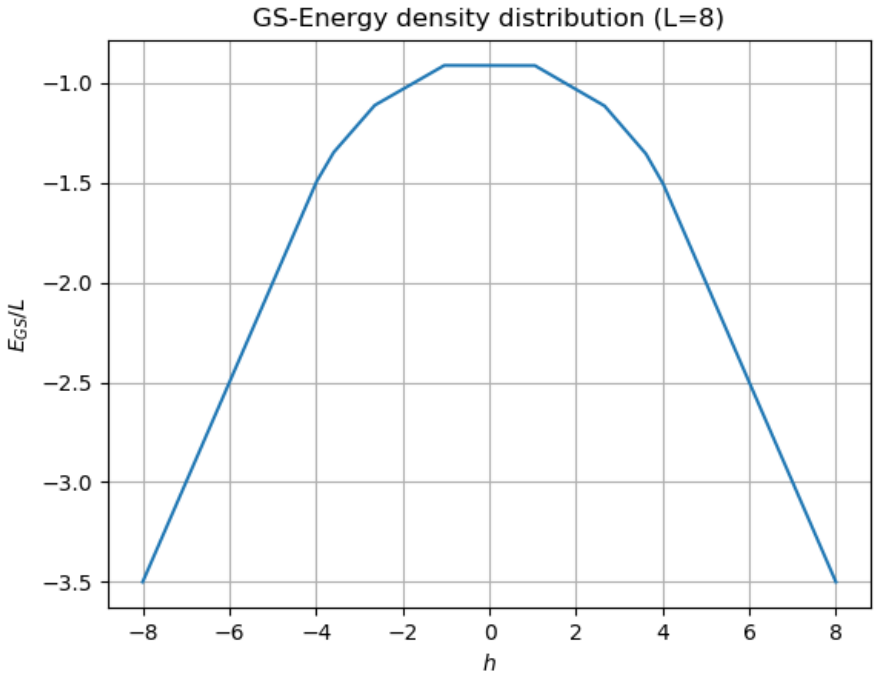


Figure 4.2: Ground state energy density in function of the external magnetic field h for the data set with length L=8 spins

The data were separated into three different samples to see how the number of training matrices influence the output of the neural network. In the Table 4.2 are the h's interval $[-8, 8]$ and how were they divided into Training, Validation, and Test sets.

Data	Training	Validation	Test
5000	4900	50	50
10000	9800	100	100
20000	19600	200	200

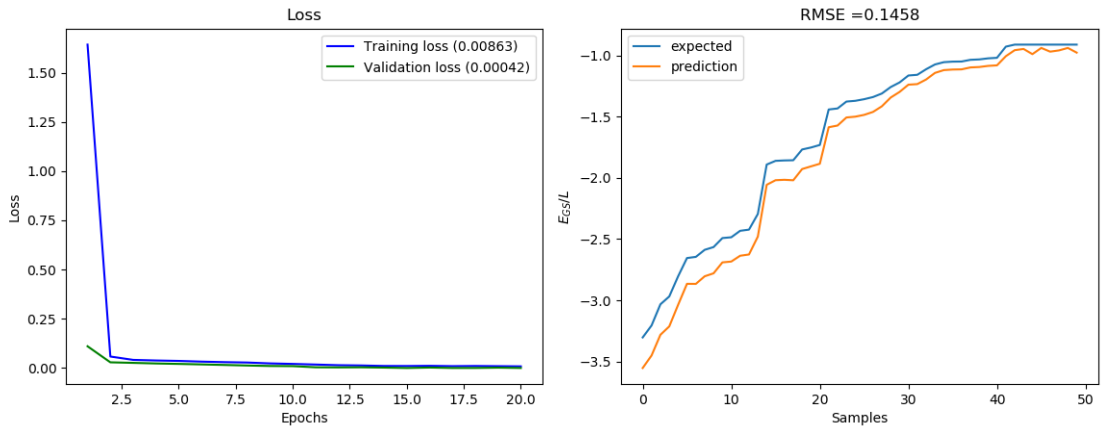
Table 4.2: Data distribution among training, validation, and test sets for a chain with length L=8 spins in a homogeneous magnetic field

Fig. 4.3 represents the results of the 5000 data passing through the neural network represented in Table 4.1. The left one Fig. 4.3a represents the evolution of the output

of the network and how it changes in each epoch in the training process, where epoch is when all the training set passes in the network one time. The training loss, result of the MSE function, indicates the quality prediction of the neural network on the training data. The validation loss is a parameter to avoid overfitting, to guarantee the best weights choice, and to be a criterion on when to stop the training process, if it does not decrease in four epochs the process stops. The other, Fig. 4.3b, is the result of the neural network in the test sample. The number in the title represents the root mean squared error (RMSE),

$$RMSE = \sqrt{\frac{1}{M}\sum_{i}^{M} y_{real_i} - y_{predict_i}^{\,2}}. \tag{4.4}$$

It is a measure to see how well the CNN can predict non-trained data.

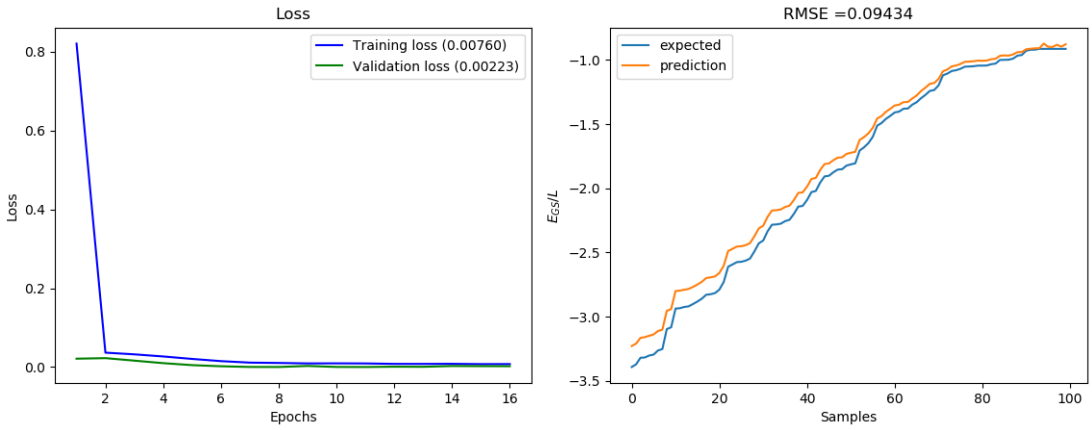


(a) Loss of the training process **(b) Prediction of the CNN**

Figure 4.3: Loss evolution for the training and validation sets and result of the test set for the 5000 matrices L=8 XXZ Hamiltonian with homogeneous external magnetic field data set

The images in Fig. 4.4 represent the same process described above in the 5000 data set sample but now with 10000 matrices. As shown in Table 4.2, 9800 matrices were used to train the CNN, 100 to validation, and 100 to test. Fig. 4.4a is the loss evolution for training and validation per epoch and Fig. 4.4b is the evaluation of the CNN in

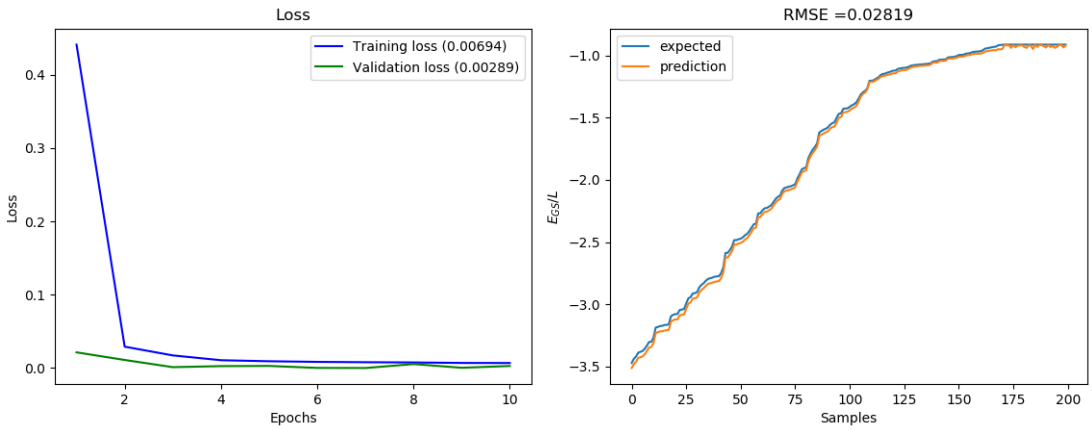
the unseen test sample. Here the increase of precision on the CNN by giving more training examples first appears. The RMSE value decreases significantly, one order of magnitude.



(a) Loss of the training process **(b) Prediction of the CNN**

Figure 4.4: Loss evolution for the training and validation sets and result of the test set for the 10000 matrices L=8 XXZ Hamiltonian with homogeneous external magnetic field data set

The last example in this data set is using all the 20000 matrices generated. As shown in Table 4.2, the training sample were 19600 matrices, the validation were 200 matrices, and the test were 200 matrices. As in the two previous examples, the process to generate Fig. 4.5 was the same. Fig. 4.5a is the loss of training and Fig. 4.5b is the result of the test sample, in it the RMSE is even smaller than the last case.



(a) Loss of the training process **(b) Prediction of the CNN**

Figure 4.5: Loss evolution for the training and validation sets and result of the test set for the 20000 matrices L=8 XXZ Hamiltonian with homogeneous external magnetic field data set

The next data set was proposed to deal with the dimension problem of quantum spin chains, smoothening the exponential growth of it with the number of particles. The proposed way to deal with it was to get a larger dimension Hamiltonian,

$$H(h) = -\frac{1}{2}\sum_{j=1}^{16}\left[\sigma_j^x\sigma_{j+1}^x + \sigma_j^y\sigma_{j+1}^y - \sigma_j^z\sigma_{j+1}^z + h\sigma_j^z\right]. \tag{4.5}$$

We then perform the same construction as before but now with a larger dimension, 16 instead of 8. In this case the matrix has dimension $2^{16}\times 2^{16}$ (or 65536×65536). The main idea is to reduce the dimension of this matrices and see if CNN can still predict with good precision. This data set was created differently from the last one, the $2^{16}\times 2^{16}$ matrices are diagonalized numerically and stored with the dimensions reduced, $2^8\times 2^8$ as the previous example. The reduction can be made by transforming a block of the matrix into a number, with the block averaged similarly to the pooling method. It is then tested if CNN can predict the ground state energy even with the input reduced. In Fig. 4.6 are the energy distribution in function of the external magnetic field, h. The values of h are in domain $[-12, 12]$ to get all the magnetization zones.

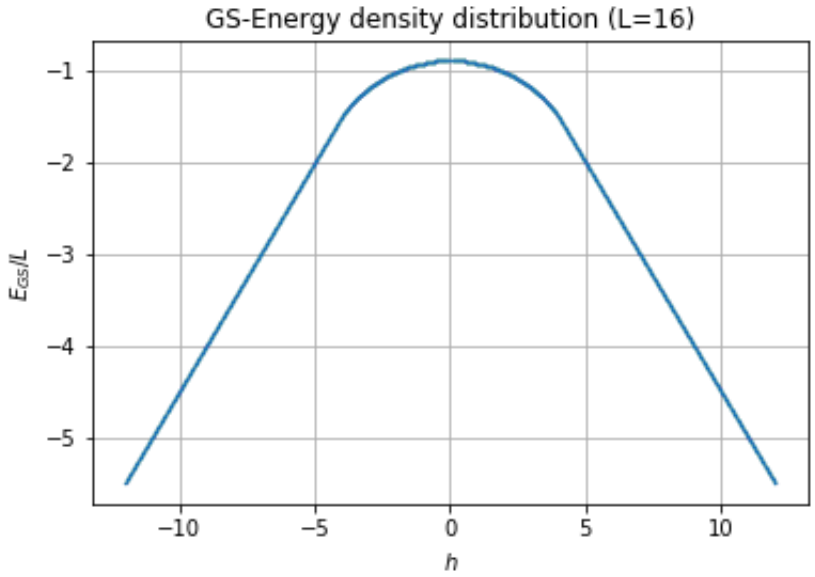


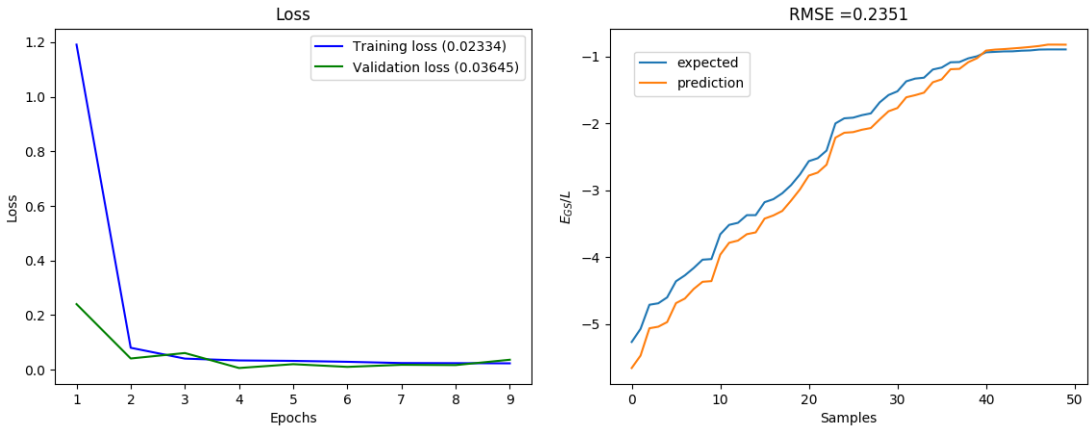
Figure 4.6: Ground state energy density in function of the external magnetic field h for the data set with length L=16 spins

As the previous homogeneous case, the data set distributed in three different sets to see how the number of inputs influences the prediction. In Table 4.3 are how they were divided into training, validation, and test sets.

	Training	Validation	Test
5000	4900	50	50
10000	9800	100	100
20000	19600	200	200

Table 4.3: Data distribution among training, validation, and test sets for a chain with length L=16 spins in a homogeneous magnetic field

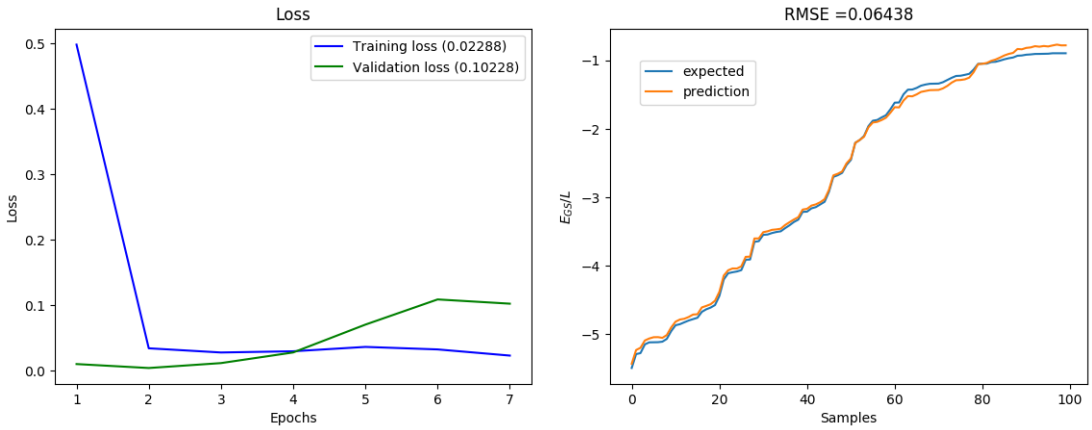
Fig. 4.7 are the data with 5000 matrices in the CNN , how the dimension was changed the input is the same, $256 \times 256 \times 1$ matrices. Fig. 4.7a shows the evolution of the loss in the training processes for the training and validation samples. Fig. 4.7b is the prediction on the CNN on the test sample where the *RMSE* is evaluated as in Eq. (4.4)



(a) Loss of the training process **(b) Prediction of the CNN**

Figure 4.7: Loss evolution for the training and validation sets and result of the test set for the 5000 matrices L=16 XXZ Hamiltonian with homogeneous external magnetic field data set

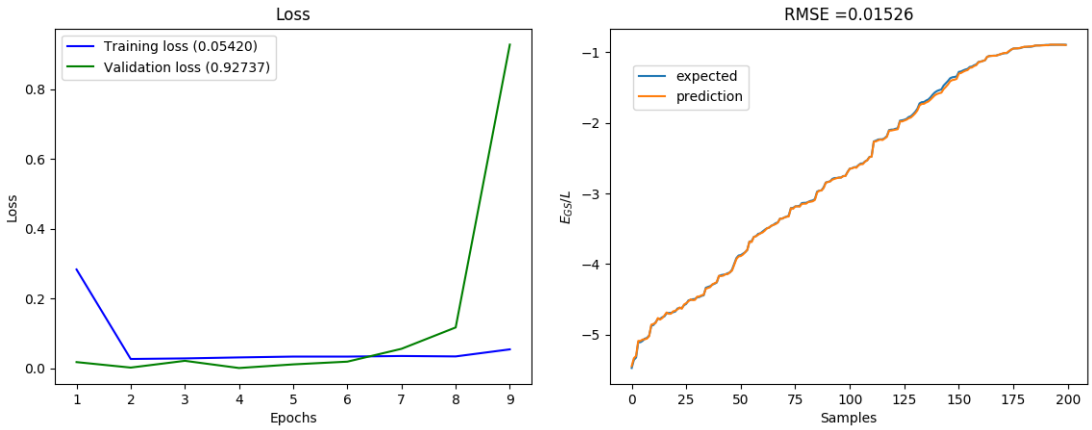
Fig. 4.8 represents the 10000 matrices data set in the CNN in Table 4.1. Fig. 4.8a represents the loss of the training and validation samples. Fig. 4.8b exhibits the results of the CNN on the test samples, and as the 8 spins case, an improvement from the 5000 is clear, as shown in the value of the RMSE.



(a) Loss of the training process **(b) Prediction of the CNN**

Figure 4.8: Loss evolution for the training and validation sets and result of the test set for the 10000 matrices L=16 XXZ Hamiltonian with homogeneous external magnetic field data set

The last result from this example, the 20000 matrices case is displayed in Fig. 4.9. The evolution of the loss in training and validation sets is in Fig. 4.9a. In this case, the validation loss diverge showing that if the training process continues the results will be overfitted. To deal with that the weights of all the epochs are stored and the best ones are used as parameters to test the CNN. Fig. 4.9b is the result of the test samples passing through the CNN with the best parameters, the improvement of the *RMSE* is clear as the case with 8 spins. This shows that the CNN was able to predict with good precision the energy of the ground state of a Hamiltonian even if it is compressed into a smaller dimension.



(a) Loss of the training process **(b) Prediction of the CNN**

Figure 4.9: Loss evolution for the training and validation sets and result of the test set for the 20000 matrices L=16 XXZ Hamiltonian with homogeneous external magnetic field data set

4.2.2 Random Local Magnetic Field

The other approached problem is the XXZ Hamiltonian with a local magnetic field, now the external magnetic term depends on the spin position. The expression is:

$$H(\mathbf{h}) = -\frac{1}{2}\sum_{j=1}^{8}\left[\sigma_j^x\sigma_{j+1}^x + \sigma_j^y\sigma_{j+1}^y - \sigma_j^z\sigma_{j+1}^z + h_i\sigma_j^z\right], \tag{4.6}$$

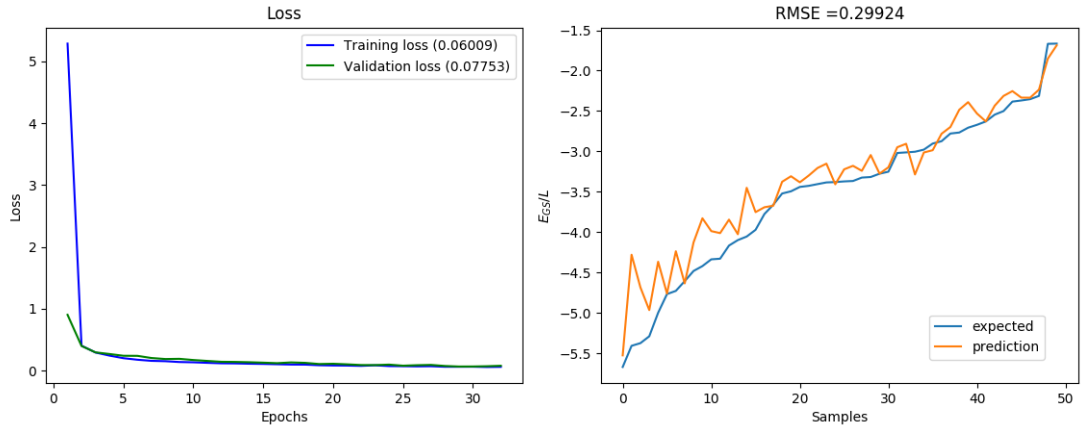
where $\mathbf{h}$ is a vector and h_i are its components.

To generate a data set, 20000 vectors with dimension 8, where each component is a random number between $[-8, 8]$, were created (each one represents $\mathbf{h}$). For each vector, we have a new Hamiltonian and a new ground state energy value. As the two previous cases, the data have been divided into three different cases to evaluate the influence on the number of inputs in the result. It was distributed as in the Table 4.4.

	Training	Validation	Test
5000	4900	50	50
10000	9800	100	100
20000	19600	200	200

Table 4.4: Data distribution L=8 among training, validation, and test sets for a chain with length L=8 spins in a inhomogeneous magnetic field

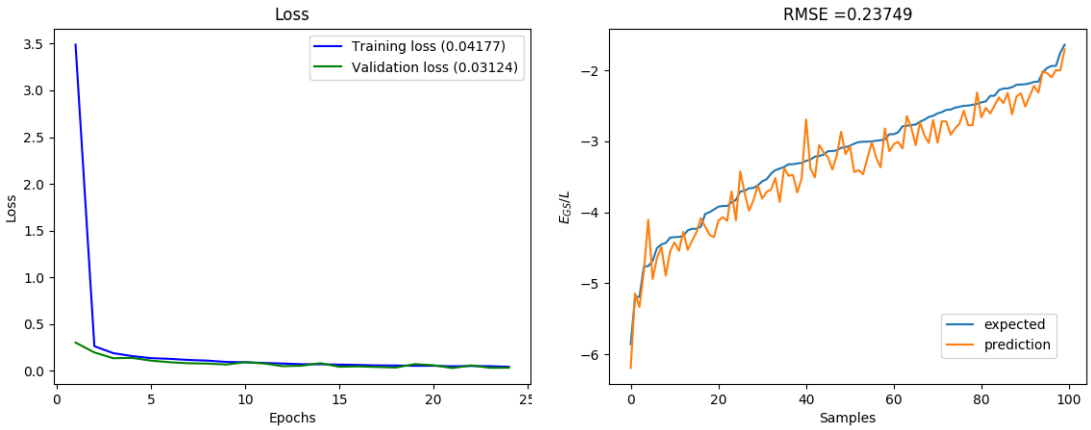
Fig. 4.10 is the 5000 samples passing through the CNN in Table 4.1. Fig. 4.10a shows the loss of the validation and training process. In Fig. 4.10b are the result of the CNN in the test samples, the value of the RMSE is in the same magnitude order as the problems showed before in the 5000 case.



(a) Loss of the training process **(b) Prediction of the CNN**

Figure 4.10: Loss evolution for the training and validation sets and result of the test set for the 5000 matrices L=8 XXZ Hamiltonian with inhomogeneous site dependent external magnetic field data set

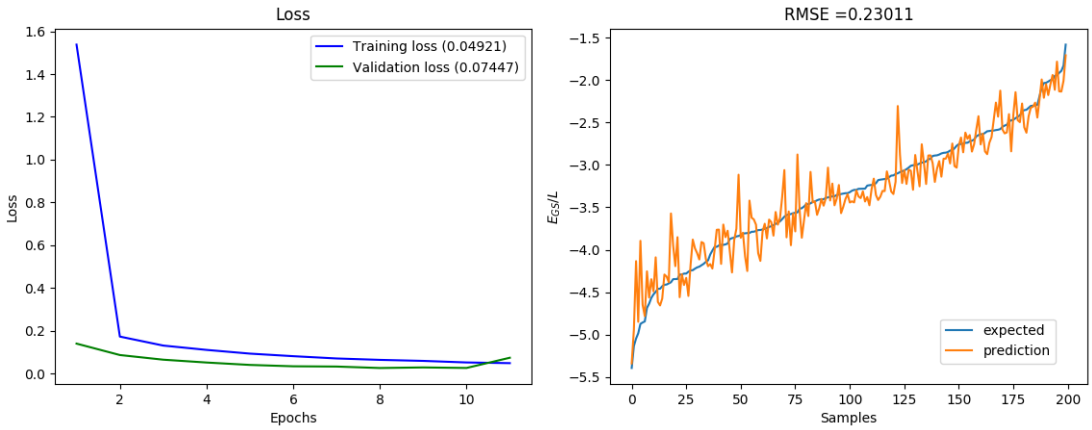
Fig. 4.11 is the result of the 10000 matrices in the CNN. Fig. 4.11a is the loss of the training and validation process. In Fig. 4.11b is the result of the test samples in the CNN, in it the improvement on the test result on the data increase. The value of the RMSE makes it explicit.



(a) Loss of the training process **(b) Prediction of the CNN**

Figure 4.11: Loss evolution for the training and validation sets and result of the test set for the 10000 matrices L=8 XXZ Hamiltonian with inhomogeneous site dependent external magnetic field data set

The last example of a local magnetic field, now with all 20000 matrices, is in Fig. 4.12. The loss of the training and the validation are in Fig. 4.12a. The result of the test in Fig. 4.12b shows that in this case the increase in the numbers of input does not significantly improve the result.



(a) Loss of the training process **(b) Prediction of the CNN**

Figure 4.12: Loss evolution for the training and validation sets and result of the test set for the 20000 matrices L=8 XXZ Hamiltonian with inhomogeneous site dependent external magnetic field data set

4.2.3 Negativity

The final problem approached is the negativity as output. The input is given by an 8-spin XXZ Hamiltonian with a variable anisotropy in Z:

$$H(\Delta) = -\frac{1}{2}\sum_{j}^{8}\left[\sigma_j^x\sigma_{j+1}^x + \sigma_j^y\sigma_{j+1}^y + \Delta\sigma_j^z\sigma_{j+1}^z\right]. \tag{4.7}$$

To generate the data set the values of Δ are set in the interval $[-1, 1]$. The data set is expressed in Fig. 4.13. For each value of Δ there is a different Hamiltonian, and for each Hamiltonian there are different eigenstates that are used to calculate the density matrix and with it the negativity.

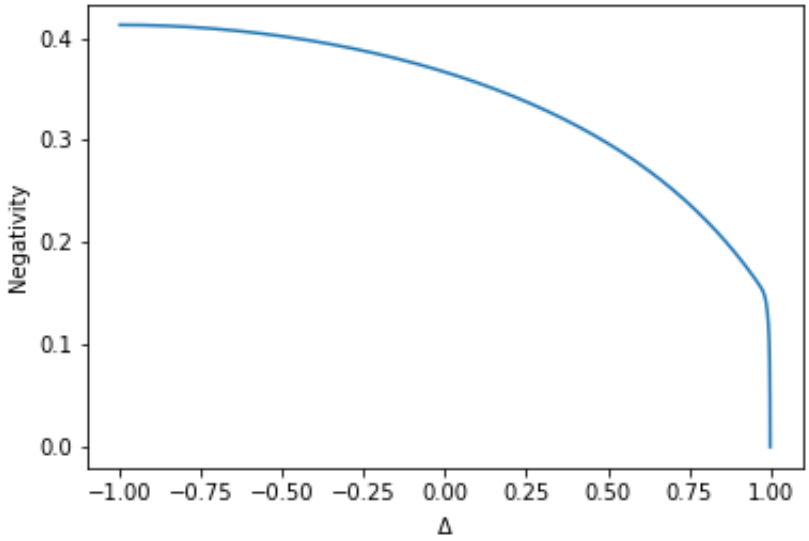


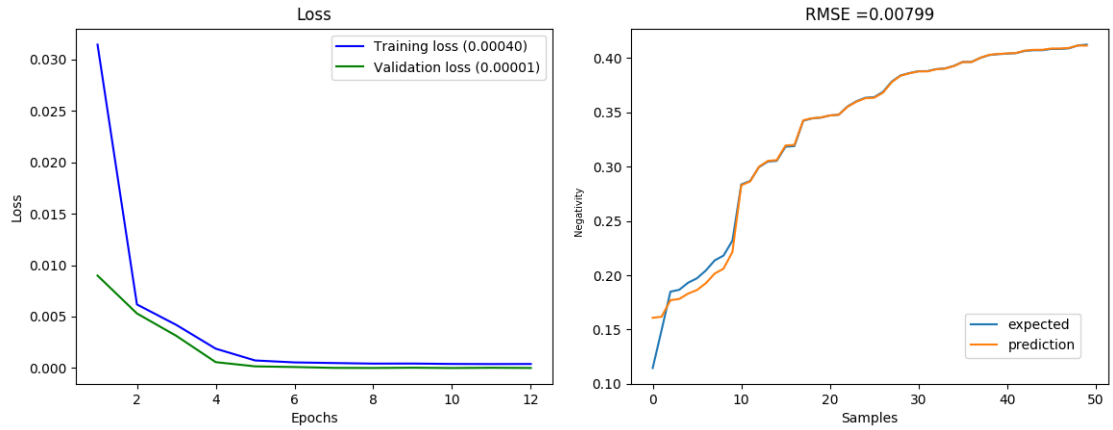
Figure 4.13: Negativity in function of the anisotropy constant in Z direction

The data set was distributed in three different sets to see how the number of inputs influences the prediction. The way they were distributed is in Table 4.5.

	Training	Validation	Test
5000	4900	50	50
10000	9800	100	100
20000	19600	200	200

Table 4.5: Data distribution among training, validation, and test sets for a chain with length L=8 spins with an varying anisotropy in Z direction

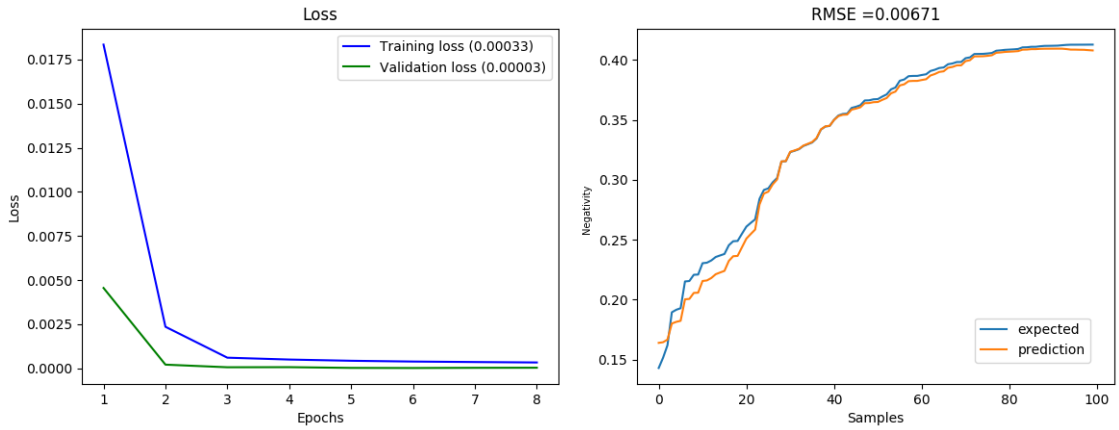
Fig. 4.14 is the result of 5000 matrices in the CNN described in Table 4.1. Fig. 4.14a is the evolution of the loss on the training and validation process. Fig. 4.14b is the result of the test samples.



(a) Loss of the training process **(b) Prediction of the CNN**

Figure 4.14: Loss evolution for the training and validation sets and result of the test set for the 5000 matrices L=8 XXZ Hamiltonian with a variable anisotropy in Z direction data set

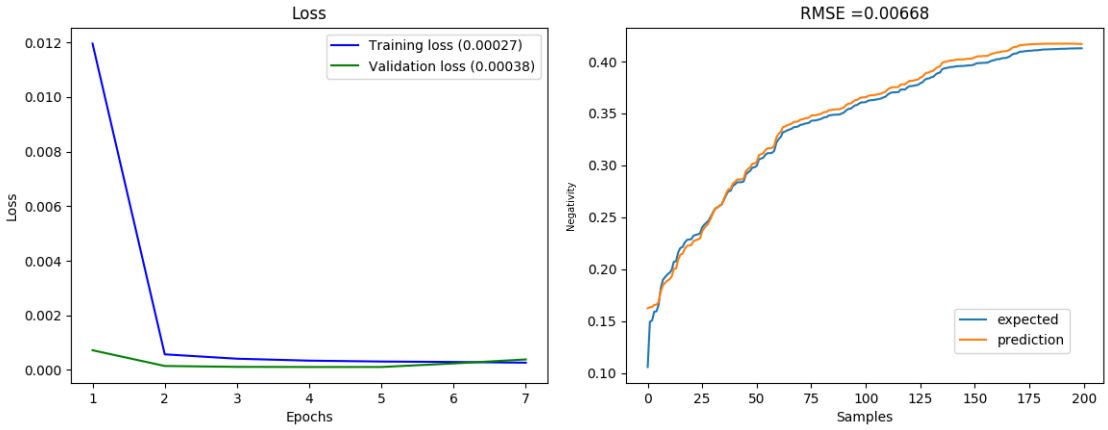
In Fig. 4.15 is the result of 10000 matrices in the CNN described in Table 4.1. Fig. 4.15a is the evolution of the loss on the training and validation process. Fig. 4.15b is the result of the test samples. The value of RMSE improves in this case.



(a) Loss of the training process **(b) Prediction of the CNN**

Figure 4.15: Loss evolution for the training and validation sets and result of the test set for the 10000 matrices L=8 XXZ Hamiltonian with a variable anisotropy in Z direction data set

In Fig. 4.16 is the result of 20000 matrices in the CNN described in Table 4.1. Fig. 4.16a is the evolution of the loss on the training and validation process. Fig. 4.16b is the result of the test samples. The value of RMSE improves in this case.



(a) Loss of the training process **(b) Prediction of the CNN**

Figure 4.16: Loss evolution for the training and validation sets and result of the test set for the 20000 matrices L=8 XXZ Hamiltonian with a variable anisotropy in Z direction data set

Chapter 5

CONCLUSIONS

This dissertation aimed to propose a general method to provide physical properties of quantum systems through a ML approach. Two physical properties have been considered, namely, the ground state energy and the pairwise entanglement (as measured by the negativity) in quantum spin chains.

In order to tackle this problem, we have treated the Hamiltonian as a single channel image and applied a largely used ML method. The inputs for addressing both properties were similar, i.e., XXZ Hamiltonians with an external magnetic field, and XXZ Hamiltonians with a varying anisotropy in the Z direction. The expressions are explicitly expressed in the last chapter in Eq. (4.3) and (4.7). The output for the first Hamiltonian was the ground state energy of the system, while for the second was the negativity. The ML architecture was the same for both kinds of problems.

The ground state energy problem was divided into three different research lines. The first was to take an 8-spin XXZ chain in the presence of an external magnetic field in the Z direction. The idea of it was to get the first contact with ML in quantum systems and see if it were solvable using that method. The results of the last chapter have shown that a CNN can be used to correctly predict the ground state of a Hamiltonian never seen before by the network if it was trained using Hamiltonians of the same kind of problem. The number of trained samples influences considerably in the precision of the final output, and it is crucial for CNN to calibrate its weights with a representative data sample.

The second research line was set to deal with the problem of scaling, since the dimension of the Hilbert space of the system grows exponentially with the number of particles. As the XXZ Hamiltonian is sparse, most elements are zeros. Then, a reduction of the dimension was proposed to investigate whether or not the CNN was still able to predict with good precision the value of the ground state. Over here is clear the importance of the large and representative data set too, the increase of training examples improves the precision of the result. This problem illustrates the high potential of this method, which deals with reduced matrices, while still getting a good approximation.

The third and last research line on the ground state prediction has been to consider a random magnetic field. Instead of a uniform magnetic field for all spins, we used here a random local magnetic field for each site. Based on the results of the last chapter for this problem, it is explicit that the size of the training sample does not influence the results, i.e., if we increase it the value of RMSE does not decrease. In this problem, the local magnetic field influences a lot in the Hamiltonian, as it changes the diagonal elements randomly. It is then possible that the number of parameters of the chosen CNN is not enough. To approach this kind of Hamiltonian using this ML method is vital to change the architecture of the network.

Moving forward to the pairwise entanglement analysis, we investigated whether the chosen CNN can predict patterns that are intrinsically connected to the structure of the ground state of the system. The output here, in this case, is the negativity, a measure of pairwise entanglement for mixed states. As shown in chapter 2, the negativity is calculated using the eigenvalues of the partially transposed density matrix. Even with no direct correlation with the Hamiltonian, the chosen CNN was able to recognize the negativity using the Hamiltonian as a single channel image.

In summary, this work proposes a proof of principle approach to deal with the quantum spin problem using ML. As a way to improve the scalability of the method, by making the system grows even larger, we speculate, as a perspective, its mixing with other approximative approaches, such as Density Matrix Renormalization Group (DMRG) (WHITE; HUSE, 1993), which is a widely used method for dealing with many-body-

systems that can approximate physical quantities with excellent precision. In this case the density matrix of the larger system would be treated as the input of the neural network, and so solve problems of larger dimensions. Furthermore, as another future perspective, quantum machine learning has been proved to have potential to solve physical problems as cited in the Introduction. It could be another way to approach this method in the future, a quantum algorithm that uses Hamiltonians as input.

REFERENCES

ADCOCK, J. et al. Advances in quantum machine learning. *arXiv:1512.02900 [quant-ph]*, dez. 2015. ArXiv: 1512.02900. Disponível em: http://arxiv.org/abs/1512.02900.

AHR, M.; BIEHL, M.; SCHLöSSER, E. Weight-decay induced phase transitions in multilayer neural networks. *Journal of Physics A: Mathematical and General*, v. 32, n. 27, p. 5003–5008, jul. 1999. ISSN 0305-4470, 1361-6447. Disponível em: http://stacks.iop.org/0305-4470/32/i=27/a=301?key=crossref.98a470dbac58a44fc4ee7029fa39f1d6.

AHR, M.; BIEHL, M.; URBANCZIK, R. Statistical physics and practical training of soft-committee machines. *The European Physical Journal B*, v. 10, n. 3, p. 583–588, ago. 1999. ISSN 1434-6028. Disponível em: http://link.springer.com/10.1007/s100510050889.

ANDERSON, P. W. More Is Different. *Science*, v. 177, n. 4047, p. 393–396, ago. 1972. ISSN 0036-8075, 1095-9203. Disponível em: https://science.sciencemag.org/content/177/4047/393.

APOLLONI, B.; BATTISTINI, E.; FALCO, D. d. Higher-order Boltzmann machines and entropy bounds. *Journal of Physics A: Mathematical and General*, v. 32, n. 30, p. 5529–5538, jul. 1999. ISSN 0305-4470, 1361-6447. Disponível em: http://stacks.iop.org/0305-4470/32/i=30/a=301?key=crossref.020833bb5e299f4fa1b13ce81eda9398.

BETHE, H. Zur theorie der metalle. *Zeitschrift für Physik*, Springer, v. 71, n. 3-4, p. 205–226, 1931.

BIAMONTE, J. et al. Quantum machine learning. *Nature*, v. 549, n. 7671, p. 195–202, set. 2017. ISSN 0028-0836, 1476-4687. Disponível em: http://www.nature.com/articles/nature23474.

BIEHL, M. An exactly solvable model of unsupervised learning. *EPL (Europhysics Letters)*, IOP Publishing, v. 25, n. 5, p. 391, 1994.

BIEHL, M.; RIEGLER, P.; W öHLER, C. Transient dynamics of on-line learning in two-layered neural networks. *Journal of Physics A: Mathematical and General*, v. 29, n. 16, p. 4769–4780, ago. 1996. ISSN 0305-4470, 1361-6447.

Disponível em: http://stacks.iop.org/0305-4470/29/i=16/a=005?key=crossref.b42a1b7b647c070fb7d8a84d8677b66c.

BIEHL, M.; SCHLöSSER, E.; AHR, M. Phase transitions in soft-committee machines. *Europhysics Letters (EPL)*, v. 44, n. 2, p. 261–267, out. 1998. ISSN 0295-5075, 1286-4854. Disponível em: http://stacks.iop.org/0295-5075/44/i=2/a=261?key=crossref.4178de39bf4c9eb2a1aa9c1fd2510f4e.

BIEHL, M.; SCHWARZE, H. Learning by on-line gradient descent. *Journal of Physics A: Mathematical and General*, v. 28, n. 3, p. 643–656, fev. 1995. ISSN 0305-4470, 1361-6447. Disponível em: http://stacks.iop.org/0305-4470/28/i=3/a=018?key=crossref.406043237f4a8be9cd243c7d2d8db195.

BOHR, N. Xlii. on the quantum theory of radiation and the structure of the atom. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Taylor & Francis, v. 30, n. 177, p. 394–415, 1915.

BöS, S.; KINZEL, W.; OPPER, M. Generalization ability of perceptrons with continuous outputs. *Physical Review E*, v. 47, n. 2, p. 1384–1391, fev. 1993. ISSN 1063-651X, 1095-3787. Disponível em: https://link.aps.org/doi/10.1103/PhysRevE.47.1384.

CARLEO, G.; TROYER, M. Solving the quantum many-body problem with artificial neural networks. *Science*, American Association for the Advancement of Science, v. 355, n. 6325, p. 602–606, 2017.

CARNEVALI, P.; PATARNELLO, S. Exhaustive thermodynamical analysis of boolean learning networks. *EPL (Europhysics Letters)*, IOP Publishing, v. 4, n. 10, p. 1199, 1987.

CARRASQUILLA, J.; MELKO, R. G. Machine learning phases of matter. *Nature Physics*, v. 13, n. 5, p. 431–434, maio 2017. ISSN 1745-2473, 1745-2481. Disponível em: http://www.nature.com/articles/nphys4035.

CARRASQUILLA, J. et al. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, v. 1, n. 3, p. 155–161, mar. 2019. ISSN 2522-5839. Disponível em: http://www.nature.com/articles/s42256-019-0028-1.

DIRAC, P. A. M. The fundamental equations of quantum mechanics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, The Royal Society London, v. 109, n. 752, p. 642–653, 1925.

DIRAC, P. A. M. On the theory of quantum mechanics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, The Royal Society London, v. 112, n. 762, p. 661–677, 1926.

DIRAC, P. A. M. Quantum mechanics and a preliminary investigation of the hydrogen atom. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, The Royal Society London, v. 110, n. 755, p. 561–579, 1926.

DIRAC, P. A. M. Quantum mechanics of many-electron systems. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, The Royal Society London, v. 123, n. 792, p. 714–733, 1929.

EINSTEIN, A. Ist die tr ägheit eines k örpers von seinem energieinhalt abh ängig? *Annalen der Physik*, Wiley Online Library, v. 323, n. 13, p. 639–641, 1905.

EINSTEIN, A.; PODOLSKY, B.; ROSEN, N. Can quantum-mechanical description of physical reality be considered complete? *Physical review*, APS, v. 47, n. 10, p. 777, 1935.

FISCHER, A.; IGEL, C. An Introduction to Restricted Boltzmann Machines. In: HUTCHISON, D. et al. (Ed.). *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. v. 7441, p. 14–36. ISBN 978-3-642-33274-6 978-3-642-33275-3. Disponível em: hhttp://link.springer.com/10.1007/978-3-642-33275-3_2i.

GERLACH, W.; STERN, O. Der experimentelle nachweis der richtungsquantelung im magnetfeld. *Zeitschrift für Physik A Hadrons and Nuclei*, Springer, v. 9, n. 1, p. 349–352, 1922.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. hhttp://www.deeplearningbook.orgi.

GUYON, I. Neural networks and applications tutorial. *Physics Reports*, v. 207, n. 3-5, p. 215–259, set. 1991. ISSN 03701573. Disponível em: hhttps://linkinghub.elsevier.com/retrieve/pii/037015739190146Di.

HANSEL, D.; MATO, G.; MEUNIER, C. Memorization Without Generalization in a Multilayered Neural Network. *Europhysics Letters (EPL)*, v. 20, n. 5, p. 471–476, nov. 1992. ISSN 0295-5075, 1286-4854. Disponível em: hhttp://stacks.iop.org/0295-5075/20/i=5/a=015?key=crossref.3e4bc5efaf9352f9eee154d3074a75bci.

HEISENBERG, W. Quantum-theoretical re-interpretation of kinematic and mechanical relations. *Z. Phys*, v. 33, p. 879–893, 1925.

HEISENBERG, W. Multi-body problem and resonance in quantum mechanics ii. *Z Phys*, v. 41, p. 239, 1927.

HIGHAM, C. F. et al. Deep learning for real-time single-pixel video. *Scientific Reports*, v. 8, n. 1, p. 2369, dez. 2018. ISSN 2045-2322. Disponível em: http://www.nature.com/articles/s41598-018-20521-y.

HORODECKI, P. Separability criterion and inseparable mixed states with positive partial transposition. *Physics Letters A*, Elsevier, v. 232, n. 5, p. 333–339, 1997.

LECUN, Y. et al. Backpropagation applied to handwritten zip code recognition. *Neural computation*, MIT Press, v. 1, n. 4, p. 541–551, 1989.

LI, L. et al. Understanding machine-learned density functionals: Understanding Machine-Learned Density Functionals. *International Journal of Quantum Chemistry*, v. 116, n. 11, p. 819–833, jun. 2016. ISSN 00207608. Disponível em: http://doi.wiley.com/10.1002/qua.25040.

LIU, D. et al. Machine Learning by Unitary Tensor Network of Hierarchical Tree Structure. *arXiv:1710.04833 [cond-mat, physics:physics, physics:quant-ph, stat]*, out. 2017. ArXiv: 1710.04833. Disponível em: http://arxiv.org/abs/1710.04833.

LLOYD, S.; MOHSENI, M.; REBENTROST, P. Quantum algorithms for supervised and unsupervised machine learning. *arXiv:1307.0411 [quant-ph]*, jul. 2013. ArXiv: 1307.0411. Disponível em: http://arxiv.org/abs/1307.0411.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, Springer, v. 5, n. 4, p. 115–133, 1943.

NEUKART, F.; DOLLEN, D. V.; SEIDEL, C. Quantum-assisted cluster analysis. *arXiv:1803.02886 [quant-ph]*, mar. 2018. ArXiv: 1803.02886. Disponível em: http://arxiv.org/abs/1803.02886.

NEVEN, H. et al. Training a Binary Classifier with the Quantum Adiabatic Algorithm. *arXiv:0811.0416 [quant-ph]*, nov. 2008. ArXiv: 0811.0416. Disponível em: http://arxiv.org/abs/0811.0416.

NEVEN, H. et al. Training a Large Scale Classifier with the Quantum Adiabatic Algorithm. *arXiv:0912.0779 [quant-ph]*, dez. 2009. ArXiv: 0912.0779. Disponível em: http://arxiv.org/abs/0912.0779.

NIELSEN, M.; BENGIO, Y.; COUVILLE, A. Deep learning. *Retrievedfrom http://neuralnetworksanddeeplearning*, 2017.

NIELSEN, M. A.; CHUANG, I. *Quantum computation and quantum information*. [S.l.]: AAPT, 2002.

NIEUWENBURG, E. van; LIU, Y.-H.; HUBER, S. Learning phase transitions by confusion. *Nature Physics*, v. 13, n. 5, p. 435–439, maio 2017. ISSN 1745-2473, 1745-2481. Disponível em: http://www.nature.com/articles/nphys4037.

PAULING, L. The application of the quantum mechanics to the structure of the hydrogen molecule and hydrogen molecule-ion and to related problems. *Chemical Reviews*, ACS Publications, v. 5, n. 2, p. 173–213, 1928.

PERES, A. Separability criterion for density matrices. *Physical Review Letters*, APS, v. 77, n. 8, p. 1413, 1996.

PLANCK, M. On the law of distribution of energy in the normal spectrum. *Annalen der physik*, v. 4, n. 553, p. 1, 1901.

PUDENZ, K. L.; LIDAR, D. A. Quantum adiabatic machine learning. *Quantum Information Processing*, v. 12, n. 5, p. 2027–2070, maio 2013. ISSN 1570-0755, 1573-1332. Disponível em: http://link.springer.com/10.1007/s11128-012-0506-4.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, American Psychological Association, v. 65, n. 6, p. 386, 1958.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. *Learning internal representations by error propagation*. [S.l.], 1985.

SARMA, S. D.; DENG, D.-L.; DUAN, L.-M. Machine learning meets quantum physics. *Physics Today*, v. 72, n. 3, p. 48–54, mar. 2019. ISSN 0031-9228, 1945-0699. Disponível em: http://physicstoday.scitation.org/doi/10.1063/PT.3.4164.

SCHRÖDINGER, E. About heisenberg uncertainty relation. *Proc. Prussian Acad. Sci. Phys. Math. XIX*, Citeseer, v. 293, 1930.

SCHRÖDINGER, E. Discussion of probability relations between separated systems. In: CAMBRIDGE UNIVERSITY PRESS. *Mathematical Proceedings of the Cambridge Philosophical Society*. [S.l.], 1935. v. 31, n. 4, p. 555–563.

SCHULD, M.; SINAYSKIY, I.; PETRUCCIONE, F. An introduction to quantum machine learning. *Contemporary Physics*, v. 56, n. 2, p. 172–185, abr. 2015. ISSN 0010-7514, 1366-5812. Disponível em: http://www.tandfonline.com/doi/full/10.1080/00107514.2014.964942.

SCHULD, M.; SINAYSKIY, I.; PETRUCCIONE, F. Simulating a perceptron on a quantum computer. *Physics Letters A*, v. 379, n. 7, p. 660–663, mar. 2015. ISSN 03759601. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S037596011401278X.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

SNYDER, J. C. et al. Finding Density Functionals with Machine Learning. *Physical Review Letters*, v. 108, n. 25, p. 253002, jun. 2012. ISSN 0031-9007, 1079-7114. Disponível em: https://link.aps.org/doi/10.1103/PhysRevLett.108.253002.

TENG, P. Machine-learning quantum mechanics: Solving quantum mechanics problems using radial basis function networks. *Physical Review E*, v. 98, n. 3, p. 033305, set. 2018. ISSN 2470-0045, 2470-0053. Disponível em: https://link.aps.org/doi/10.1103/PhysRevE.98.033305.

VERSTRAETE, F.; AUDENAERT, K.; MOOR, B. D. Maximally entangled mixed states of two qubits. *Physical Review A*, APS, v. 64, n. 1, p. 012316, 2001.

VIDAL, G.; WERNER, R. F. Computable measure of entanglement. *Physical Review A*, APS, v. 65, n. 3, p. 032314, 2002.

WECKER, D.; HASTINGS, M. B.; TROYER, M. Training a quantum optimizer. *Physical Review A*, v. 94, n. 2, p. 022309, ago. 2016. ISSN 2469-9926, 2469-9934. Disponível em: https://link.aps.org/doi/10.1103/PhysRevA.94.022309.

WERNER, R. F. Quantum states with einstein-podolsky-rosen correlations admitting a hidden-variable model. *Physical Review A*, APS, v. 40, n. 8, p. 4277, 1989.

WHITE, S. R.; HUSE, D. A. Numerical renormalization-group study of low-lying eigenstates of the antiferromagnetic s= 1 heisenberg chain. *Physical Review B*, APS, v. 48, n. 6, p. 3844, 1993.

WU, L.-A.; SARANDY, M. S.; LIDAR, D. A. Quantum phase transitions and bipartite entanglement. *Physical review letters*, APS, v. 93, n. 25, p. 250404, 2004.

ZENG, W.; COECKE, B. Quantum Algorithms for Compositional Natural Language Processing. *Electronic Proceedings in Theoretical Computer Science*, v. 221, p. 67–75, ago. 2016. ISSN 2075-2180. Disponível em: http://arxiv.org/abs/1608.01406.