Radboud University

MSc Physics and Astronomy

Master Thesis

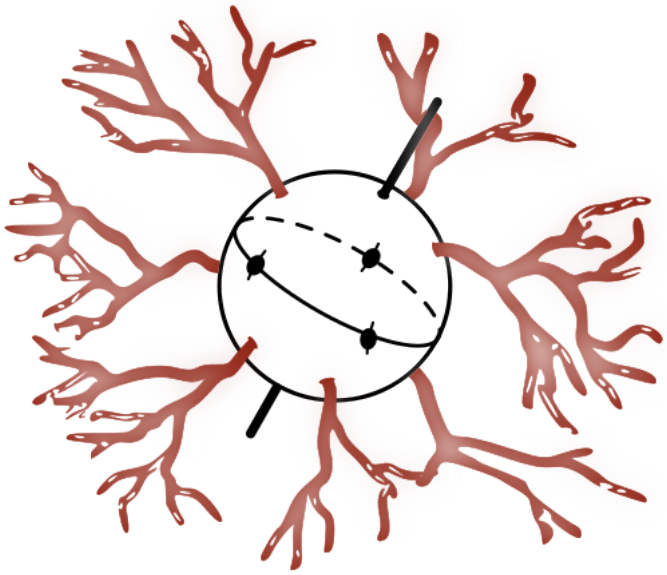
Quantum Perceptron Learning

Author:
R.C. Wiersema

Supervisor:
Prof. Dr. H.J. Kappen

Examination date:
October 2, 2019

Second Supervisor:
Prof. Dr. J. Mentink



Department of Biophysics

RADBOUD UNIVERSITY

Abstract

Faculty of Science
Department of Biophysics

Master of Science

Quantum Perceptron Learning

by Roeland Wiersema

We propose a method for learning a quantum probabilistic model of a perceptron. By considering a cross entropy between two density matrices, we can learn a model that takes noisy output labels into account while learning. A multitude of proposals already exist that aim to utilize the curious properties of quantum systems to build a quantum perceptron, but these proposals rely on a classical cost function for the optimization procedure. We demonstrate the usage of a quantum equivalent of the classical log-likelihood, which allows for a quantum model and training procedure. We show that this allows us to better capture noisiness in data compared to a classical perceptron. By considering entangled qubits we can learn nonlinear separation boundaries, such as XOR. We outline possible extensions of this model and consider the constraints of physical systems for learning. The work in this thesis is the culmination of one and a half year of research in the Biophysics group at Radboud University. The most important results are summarized in a *Physical Review A* paper [1].

Acknowledgements

There are a couple of people who have been essential to the development of this thesis. First of all, Professor Kappen's guidance has been inspirational. He has offered me a perfect introduction to the wonderful physics community and life as an academic. Next, I want to thank the Biophysics group for their input and the inspiring lunch breaks, specifically: Giel van Bergen, Manu Compen, Eduardo Dominiquez en Onno Huygen. I also want to thank Professor Mentink for interesting discussions and taking me along to Eindhoven. Finally, my thanks go to my friends and family for keeping me sane and believing in me.

Contents

Chapter 1

Quantum Perceptron

Intelligent machines are as old as Greek mythology. In *The Argonautica*, the bronze automaton Talos wards the island of Crete to protect it from the troublesome Argonauts, who seek to steal the Golden Fleece [2]. Although we are still far away from animating metal statues, substantial progress has been made with regards to simulating intelligence that does not require a divine touch. One of the striking properties of human cognition is the ability to process complex sensory data, such as vision or sound. This ability stems from the fact that the brain is able to filter redundant information, and extract useful details that are required for human functioning. This mechanism of information extraction is what we try to replicate in the field of artificial intelligence (AI). The problem is that nature has refined this mechanism through billions of years of trial-and-error, with a myriad individuals trying to adapt to their surroundings, optimizing the cost function that is survival. Even with modern computers we cannot hope to simulate this whole process. What we can learn from nature is that a computational model that generalizes from experience is powerful. The question becomes: what does this model look like?

The field of AI originated from computing science, with Alan Turing first speculating about the possibility of building intelligent machines using digital computers in a 1950 paper. In this seminal work, he explored early ideas of machine learning, genetic algorithms and reinforcement learning [3]. Soon thereafter, two MIT scientists embraced Hebb's neuroscientific theory that "Cells that fire together, wire together" [4], and built a neural net consisting of 40 Hebbian synapses from analogue electronics. A few years later, Rosenblatt's perceptron received a lot of attention for its ability to learn any problem that could be separated linearly [5]. In general, there was a lot of optimism in the 1950s and 1960s about the future of AI, but problems that plagued the field from the beginning were becoming more troublesome. In those early days, most AI approaches were symbolic and rule-based in nature, which did not scale well to larger of more complex systems. Limited generalization abilities, a combinatoric explosion of the state space and intractable calculations stumped progress, and led governments to cut funding for most of the research throughout the 1970s. Additionally, the infant industry around machine learning never bloomed, as companies failed to fulfill exuberant promises. The field entered a period which has received the dramatic name "AI winter", where progress was slow. Parallel to the symbolic approach to AI, the field of machine learning focused on more statistical approach to intelligence. The language of machine learning is one of linear algebra, optimization and statistics; fields of study that developed rapidly after the axiomization of mathematics in the early 1900s. When the shortcomings of the symbolic AI became apparent, it allowed machine learning to rise to prominence. In the mid 1980s several groups rediscovered the method of automatic differentiation which allowed for easy calculation of gradients by backpropagation. This paved the way for training

multilayer networks that could replace hand engineered features with more abstract representations. In 1984 Valiant proposed to study what computers can learn instead of what they can calculate, laying the foundation for the field of computational learning theory [6]. Along with this development, connections between statistical physics and machine learning were made, which finally shifted the field from arule-based approach to a more statistical paradigm [7]. A thaw had set in, and the 1990s and 2000s saw the development of many different algorithms: support vector machines, Recurrent Neural Networks, Random Forests and Hidden Markov models to name a few [8, 9]. Due to the successes of these techniques machine learning permeated into scientific fields as bioinformatics [10], medical imaging [11], astrophysics [12] and particle physics [13]. In the 2010s, catalyzed by the availability of large data sets and cheap parallel processing power in the form of GPU's, neural networks with a large amount of layers (deep learning) could be trained successfully. This has led to numerous successes, from beating the best Go player [14], real-time language translation from video [15] and improved autonomous driving [16, 17]. Although simulating intelligent life is probably still far away, machine learning has been applied very successfully to specific tasks for which data was abundant and its impact on society has been significant.

1.1 Quantum Machine Learning

A model agnostic approach to understanding nature probably does not sit well with most physicists, but the potential of machine learning for physics cannot be ignored. The fusion of machine learning with quantum mechanics is called quantum machine learning and shows great promise [18]. The usage of the term quantum machine learning is a bit ambiguous, since it is used to describe different types of research. The relations between these types of research are depicted schematically in figure 1.1.

Inspired by the success of fusing physics with machine learning we focus on a relatively unexplored area of this research: learning machine learning models for classical data using quantum cost functions. The perceptron is of particular interest, since it serves as the building block for neural networks and deep learning [19]. Some work has been done to develop quantum equivalents of neural networks, but these proposals only use quantum effects in the nodes or synapses of the network. They are still trained by minimizing a classical cost function [20–27]. Constructing quantum probabilistic models from density matrices is a new direction of quantum machine learning research [28, 29], where one exploits quantum effects in both the model and training procedure, by constructing a differentiable cost function in terms of density matrices. In this work, we use this approach to construct a quantum perceptron that uses a generalization of the classical likelihood function for learning, replacing the classical perceptron bit with a qubit. Others have attempted to generalize probability theory to density matrices, but the equivalent of conditional probabilities, conditional density matrices, do not preserve positive definiteness so states can be assigned a negative probability [30, 31]. Our approach bypasses this difficulty because we construct a data density matrix from the probability amplitude of the empirical data distribution, which is always positive semidefinite.

The proposed quantum-inspired perceptron allows us to better capture noisiness in data sets than a classical perceptron. Additionally, we show that by considering entangled qubits we can learn non-trivial separation boundaries, such as XOR. The model can easily be expanded to learn different types of data and provides a solid foundation to investigate other physically-inspired models. The most important results of this thesis are summarized in a 2019 paper [1]. In chapter 2, we present a full analysis of the proposed model. Appendix A contains background information for quantum physics, and will be referred to on occasion. The additional appendices contain lengthy calculations that were required for the analysis. For a detailed overview of quantum computing, which is briefly touched upon in the final section, the reader is referred to [32]. For an excellent reference on machine learning, consider [33].

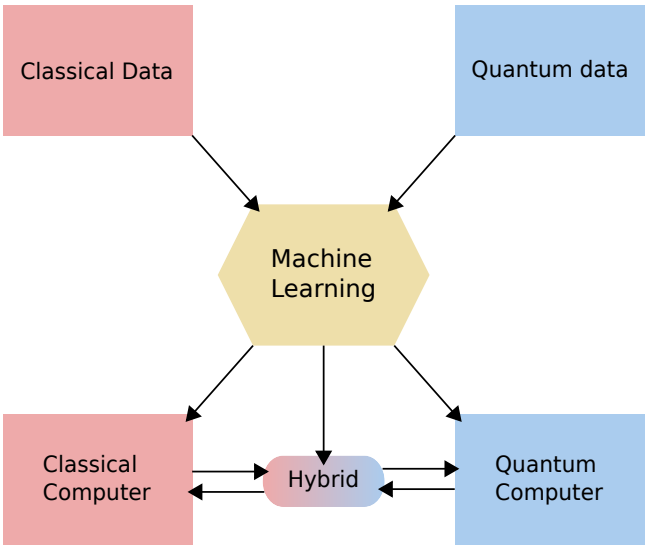


FIGURE 1.1: Overview of the field of quantum machine learning. A classical computer is defined as a von Neumann architecture computer. A quantum computer is either a Noisy Intermediate-Scale Quantum Computer or a quantum annealer. The difference between classical data and quantum data is that the source of quantum data is either some complex model of a quantum system, or experimental measures of a physical system. **Classical data → Classical computer:** The traditional field of machine learning, as previously discussed. **Quantum data → Classical computer:** This field is for instance concerned with understanding quantum many body interactions [34, 35] through neural network states, providing speedups for Density Field Theory approaches [36, 37] or learning phase transitions in condensed matter systems [38]. **Classical data → Quantum computer:** Here, classical data is encoded onto a quantum computer, and a quantum computing equivalent of a classical machine learning algorithm is used for learning [39, 40]. Another approach is to provide speedups for the underlying linear algebra routines [41–43]. However, most of these proposals remain unfeasible due to the current limitations of modern quantum computers, which still lack long qubit coherence times and high gate fidelity [44]. **Quantum data → Quantum computer:** In quantum state tomography the goal is to reconstruct the density matrix of an unknown quantum state, through experimental measurement. The amount of measurements required can be large and machine learning can assist in reducing this cost [45]. **Quantum kernel hybrid:** With quantum kernels one computes a classically intractable kernel function on a quantum computer, which is fed back to a classical computer and incoporated into a classical machine learning model [40]. **Quantum-inspired hybrid:** Quantum-inspired algorithms are a hybrid, where classical data is used, but a quantum model is learned [18, 28, 46–48]. This is without involvement of a quantum computer and it is where the work of this thesis is located. The work in this thesis falls in the category **Hybrid**, since we use a quantum-inspired model but execute it on a classical computer.

1.2 Code

The code used for this research initially contained only a model with the exact gradient update rules written in Python. When the models expanded to continuous, multiclass and entangled learning all the code was rewritten in a single big TensorFlow class. The reason for this is that for larger density matrices we have to rely on the numerical evaluation of the eigenvectors, which can be sped up significantly if performed in parallel on the GPU.

TensorFlow is a high level graphical computing framework for Python that is optimized for GPU usage. It is maintained by Google and used by many authors in the machine learning community. Complicated models can easily be trained through the process of automatic differentiation, where the entire optimization schedule is decomposed in terms of simple mathematical operations so that gradients can be computed efficiently with the chain rule. The code with the TensorFlow model and ways to reproduce the most important figures in this thesis can be found on GitHub [49].

1.3 Machine Learning

Machine learning can be defined as a set of methods that can automatically detect patterns in data, and then uses these uncovered patterns to predict future data. Supervised learning is a subset of machine learning where we try to find a mapping $y = f(\mathbf{x})$ for data $\mathcal{D}\{(y, \mathbf{x})\}^N$. The input vector $\mathbf{x}$ is often called the feature vector and the output y is called the label or response variable. In the case that y is some real-valued scalar $y \in \mathbb{R}$ we talk about regression. If the output y is from some finite set $C \in \{1, ..., p\}$ we talk about classification, which is what we will consider from now on. Because data is often noisy, we make use of probability theory as an underlying framework for machine learning. The function $f(\mathbf{x})$ is then replaced by a conditional probability distribution $p(y|\mathbf{x}, \mathcal{D})$. Let us establish a couple of basic definitions from probability theory that will allow us to proceed with describing machine learning models.

In machine learning we often deal with probability distributions over multiple random variables. A joint probability over random variables X, Y is defined as

$$p(X, Y) = p(X|Y)p(Y),$$

where $p(X|Y)$ is the conditional probability that X is true given that Y is true. This also defines a conditional probability as

$$p(X|Y) = \frac{p(X, Y)}{p(Y)}.$$

The marginal is obtained by summing out the random variables that are not of interest,

$$p(X) = \sum_y p(X|Y = y)p(Y = y).$$

Combining these definitions gives us Bayes' Rule,

$$p(X|Y) = \frac{p(Y|X)p(X)}{p(Y)}.$$

To not obfuscate everything with excessive notation we use the convention that for some data set $\mathcal{D} = \{(\mathbf{x}, y)\}^N$ the sum $\sum_{\mathbf{x}}$ is the sum over all the samples and $\sum_y$ is the sum over the possible labels for that sample. The sum over the tuple $\sum_{(\mathbf{x},y)}$ indicates the sum over the sample, with its corresponding label in the data set $\mathcal{D}$.

1.4 Classical Perceptron

For supervised learning we are interested in inferring a function $f(\mathbf{x})$ that correctly predicts the output data y for all samples $\mathbf{x} \in \mathbb{R}^d$ in the data. The classical perceptron is such a supervised learning model. It is a classifier that takes a set of linear predictors $h(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$ and produces an output $y \in \{1, -1\}$. Perceptrons are important, because they form the building blocks of Multilayer Neural Networks (MLP) which are the driving force behind deep learning [19].

The most basic perceptron is Rosenblatt's perceptron, which was briefly mentioned in the introduction. This model generates an output by applying a nonlinear activation function $f(x) = \text{sgn}(x)$ to a linear predictor of the input data [5]. The sign function $\text{sgn}(x)$ is defined as

$$\text{sgn(x)} = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0 \end{cases}.$$

In order to find the weights $\mathbf{w}$ that provide the optimal split of the data we use a simple learning rule. For the classical perceptron this learning rule is given by

$$\Delta\mathbf{w} = \epsilon(y - \text{sgn}(\mathbf{w} \cdot \mathbf{x}))\mathbf{x}, \tag{1.1}$$

where ϵ is the learning rate. Geometrically the perceptron can be interpreted as an algorithm that tries to split the input space in two parts for $y = \{+1, -1\}$. When possible, the learning algorithm converges to a solution that separates the data perfectly. We then call the data linearly separable. We can add biases w_0 to the input to shift the activation function,

$$\mathbf{w} \cdot \mathbf{x} = w_0 + \mathbf{w} \cdot \mathbf{x} = w_0 + \sum_i w_i x_i^\mu,$$

where p is the number of inputs. Adding a bias provides an extra degree of freedom that allows the algorithm to deal with data that can not be separated with a line through the origin. This is shown schematically in figure 1.2.
Also, a translation $\mathbf{x} \to \mathbf{x} - \mathbf{x}'$ of the data can be absorbed in the bias,

$$\begin{aligned} h^k &= \mathbf{w}^k \cdot (\mathbf{x} - \mathbf{x}') = \mathbf{w}^k \cdot \mathbf{x} - \overbrace{\mathbf{w}^k \cdot \mathbf{x}' + w_0}^{w_0'} \\ &= \mathbf{w} \cdot \mathbf{x} - w_0'. \end{aligned} \tag{1.2}$$

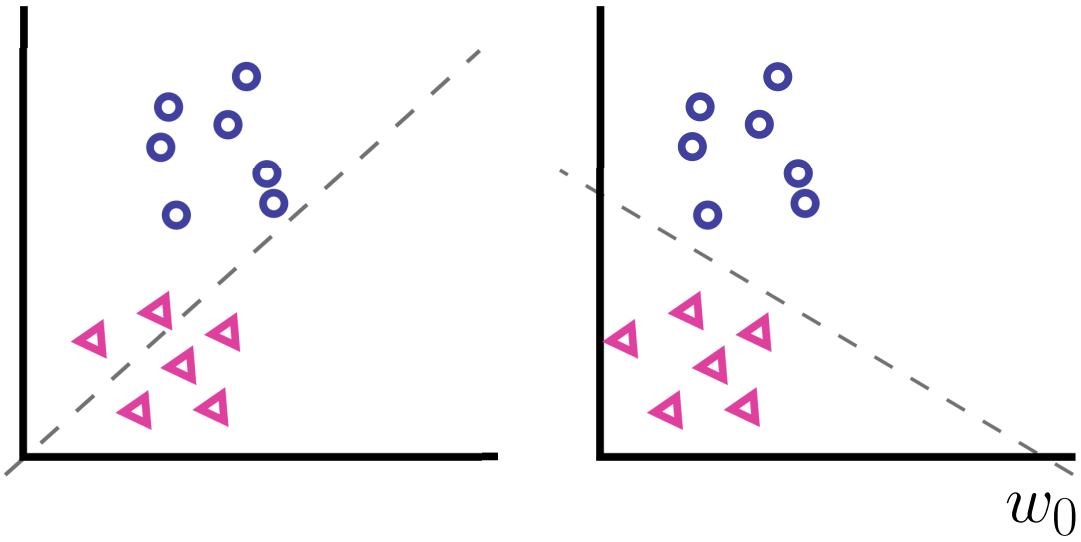


FIGURE 1.2: Schematic representation of separation boundaries through the input space. Adding a bias allows for easier learning because we are not restricted to slices through the feature space which go through the origin.

It has been shown that stacks of connected perceptrons, also known as neural networks, can in principle approximate any possible function [50]. This implies that any mapping from data to label can be learned. The difficulty is in coming up with a learning procedure and network architecture that finds this universal approximator, and does not get stuck in suboptimal local minima.

The activation function is a key part of the perceptron model. In Rosenblatt's perceptron the step function is proposed *ad hoc* to mimic behaviour the behaviour of biological neurons. In the last two decades, research into neural networks has spawned a number of activation functions that attempt to either improve generalization properties of a network or hope to increase training speed [51–53]. Another method for improving performance is choosing a suitable a cost function $E(x)$. This function attributes a cost to some event. For instance, by calculating the euclidean distance between the predicted label and the true label. If $E(x)$ is continuous and differentiable we can use gradient descent to find a local (under convexity global) minimum of that function. By making some assumptions about how the data is distributed, we can come up with an optimization criterion that is derived from a probabilistic model. An example of a model that uses such a criterion is logistic regression.

1.5 Logistic Regression

In order to learn a statistical model we need come up with a model distribution $p(\mathcal{D}|\boldsymbol{\theta})$ of the data $\mathcal{D}$ given some parameters $\boldsymbol{\theta}$. This distribution is called the likelihood, it describes how likely the model is given the data and the parameters that we have chosen. In order to estimate the parameters of a statistical model we can use the technique of Maximum Likelihood Estimation.

Consider a probability density $p(\mathbf{x}, y; \boldsymbol{\theta})$. For a fixed data set $\mathcal{D} = \{(\mathbf{x}, y)\}^N$ the likelihood function describes the probability of the data given the model and some set of parameters $\boldsymbol{\theta}$, written as

$$\mathcal{L} = p(\mathcal{D}; \boldsymbol{\theta}). \tag{1.3}$$

If the data independent and identically distributed (i.i.d.) then we can write the likelihood as the product of individual probabilities,

$$\mathcal{L} = \prod_{(\mathbf{x},y)} p(\mathbf{x}, y|\boldsymbol{\theta}). \tag{1.4}$$

By maximizing this likelihood, we can find the optimal set of parameters $\boldsymbol{\theta}$ that assigns the highest total probability of the data under the model probability density $p(\mathbf{x}, y; \boldsymbol{\theta})$ [33]. To ensure that we can properly calculate the total probability, we often minimize the negative log-likelihood instead, since the log-transformed function preserves the maximum but replaces the product by a sum. We will continue with the negative log-likelihood,

$$\mathcal{L} = -\sum_{(\mathbf{x},y)} \log(p(\mathbf{x}, y; \boldsymbol{\theta})). \tag{1.5}$$

Using the law of large numbers we can write

$$\begin{aligned}
&\lim_{N\to\infty} -\sum_{(\mathbf{x},y)} \log(p(\mathbf{x}, y; \boldsymbol{\theta})) \to -\mathbb{E}\left[\log(p(y|\mathbf{x}, \boldsymbol{\theta}))\right]_{q(\mathbf{x},y)} \\
&= -\sum_{\mathbf{x},y} q(\mathbf{x}, y) \log(p(\mathbf{x}, y; \boldsymbol{\theta})) \\
&= -\sum_{\mathbf{x},y} q(\mathbf{x}) q(y|\mathbf{x}) \log(p(y|\mathbf{x}; \boldsymbol{\theta})) - q(\mathbf{x}) \underbrace{q(y|\mathbf{x})}_{=1} \log(p(\mathbf{x}; \boldsymbol{\theta})) \\
&= -\sum_{\mathbf{x}} q(\mathbf{x}) \sum_{y} q(y|\mathbf{x}) \log(p(y|\mathbf{x}; \boldsymbol{\theta})) - q(\mathbf{x}) \log(p(\mathbf{x}; \boldsymbol{\theta})) \\
&= \sum_{\mathbf{x}} q(\mathbf{x}) H(q(y|\mathbf{x}; \boldsymbol{\theta}), p(y|\mathbf{x}; \boldsymbol{\theta})) - q(\mathbf{x}) \log(p(\mathbf{x}; \boldsymbol{\theta})),
\end{aligned}$$

where $q(\mathbf{x}, y)$ is the empirical distribution given by the data. The term

$$H(q(y|\mathbf{x}; \boldsymbol{\theta}), p(y|\mathbf{x}; \boldsymbol{\theta})), \tag{1.6}$$

is known as the cross entropy and is a measure of "distance" between two distributions. Notice that if we assume $p(\mathbf{x}; \boldsymbol{\theta}) = p(\mathbf{x})$ independent of $\boldsymbol{\theta}$ then when only have to calculate the cross entropy term, since the final term is constant with respect to the parameters of the model.

We can now construct a perceptron model, a binary classifier, analogous to Rosenblatt's perceptron as a probabilistic model with a binary response variable $y \in \{0, 1\}$. For a random process with a binary random variable we can use a binomial regression model, which has probability density function is given by

$$\text{Bin}(k|n, \mu) = \binom{n}{k} \mu^k (1-\mu)^{n-k},$$

where we observe k times label $y = 1$ out of n tries and $\mu \in [0, 1]$ is the probability of finding $y = 1$. If $n = 1$, then

$$p(y|\boldsymbol{\theta}) = \text{Bin}(k|1, \boldsymbol{\theta}) \to \text{Ber}(y|\boldsymbol{\theta}) = \mu^{\mathbb{I}(y=1)} (1-\mu)^{\mathbb{I}(y=0)}.$$

In other words, for a single trial the variable is Bernoulli distributed. Assume that the distribution of labels also depends on the input $\mathbf{x}$. Then we can write the negative

log-likelihood as

$$\begin{aligned}\mathcal{L} &= -\sum_{\mathbf{x}} \log\Big(\mu(\mathbf{x};\theta)^{\mathbb{I}(y=1)}(1-\mu(\mathbf{x};\theta))^{\mathbb{I}(y=0)}\Big) \\ &= -\sum_{\mathbf{x}} y\log(\mu(\mathbf{x};\theta)) + (1-y)\log(1-\mu(\mathbf{x};\theta)).\end{aligned}$$

We further require that $\mu(\mathbf{x};\theta)$ is a function of the linear predictor $h(\mathbf{x}) = \mathbf{w}\cdot\mathbf{x}$, omitting the argument of h for now. To ensure that $\mu(\mathbf{x};\mathbf{w})$ is contained in $[0,1]$ we can assume that the errors are distributed according to the logistic distribution, given by the sigmoid function,

$$\begin{aligned}\mu(\mathbf{x};\mathbf{w}) \equiv S(h(\mathbf{x})) &= \frac{1}{1+e^{-h(\mathbf{x})}} = \frac{e^{h(\mathbf{x})/2}}{e^{h(\mathbf{x})/2}+e^{-h(\mathbf{x})/2}} = \frac{2e^{h(\mathbf{x})/2}}{\cosh(h(\mathbf{x})/2)} \\ 1-S(h(\mathbf{x})) &= \frac{2e^{-h(\mathbf{x})/2}}{\cosh(h(\mathbf{x})/2)}.\end{aligned} \tag{1.7}$$

which is the continuous extension of the step function with a codomain $(0,1)$, as can be seen in figure 1.3. This model is called logistic regression. Now, if we take the perceptron of section 1.4 with a sigmoid activation as nonlinear function, we end up with logistic regression, but without the probabilistic motivation. Logistic regression is also equivalent to the probability distribution of an Ising spin at finite temperature under the Boltzmann distribution. If we rescale the output y, from $\{0,1\} \to \{-1,1\}$, then we can write the negative log-likelihood as

$$\mathcal{L} = -\sum_{\mathbf{x},y} \log\left(\frac{2e^{yh(\mathbf{x})/2}}{\cosh(h(\mathbf{x})/2)}\right),$$

which under the distribution $q(\mathbf{x},y)$ of the data becomes

$$= -\sum_{\mathbf{x}} q(\mathbf{x}) \sum_{y} q(y|\mathbf{x}) \log\left(\frac{e^{yh(\mathbf{x})/2}}{2\cosh(h(\mathbf{x})/2)}\right). \tag{1.8}$$

The nice thing about this formulation is that it gives us an optimization criterion motivated by probability theory, the negative log-likelihood. If we find the global minimum of this function we will have found the optimal parameters of the model under the distribution of the data.

1.5.1 Convexity of the Log-Likelihood

Finding a global minimum of a cost function is a hard problem in general. However, if we are dealing with a convex function, then we are assured to find the global minimum with gradient descent. A scalar function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if for any two elements $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^n$ and $0 \le \lambda \le 1$ we have

$$f(\lambda\mathbf{x} + (1-\lambda)\mathbf{x}') \le \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{x}'). \tag{1.9}$$

For a schematic explanation see figure 1.4. We will show that this inequality holds for the negative log-likelihood in equation 1.8. First off, a weighted sum of two convex functions is again convex. This follows immediately from the definition in

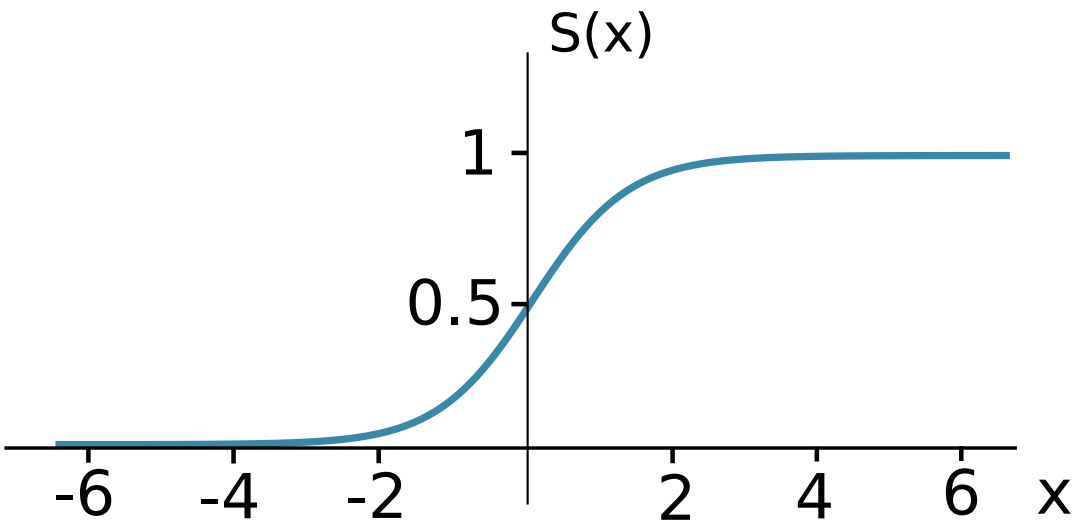


FIGURE 1.3: The sigmoid function. $S(x)$ has codomain (0,1). For $x \approx 4.5$ this gives $S(4.5) = 0.99$ which is already close to 1. Adding an inverse temperature $S(\beta x)$ allows for tuning of the smoothness of the function. In the limit $\beta \to \infty$ the sigmoid become a step function.

equation 1.9. Assume that f and g are convex, then

$$f(\lambda \mathbf{x} + (1-\lambda)\mathbf{x}') + g(\lambda \mathbf{x} + (1-\lambda)\mathbf{x}') \leq \lambda f(\mathbf{x}) + (1-\lambda) f(\mathbf{x}') + \lambda g(\mathbf{x}) + (1-\lambda) g(\mathbf{x}'),$$

so $f + g$ is also convex. This means that we only have to prove that $-\log(S(x))$ is convex to show that the full negative log-likelihood is convex. It can be shown that a function is convex if the Hessian is positive semidefinite [54]. For this we only need to look at the second order derivative,

$$\begin{aligned} -\frac{d^2}{dx^2}\log\left(\frac{1}{1+e^{-ax}}\right) &= -\frac{d}{dx}(1+e^{-ax})\frac{ae^{-ax}}{(1+e^{-ax})^2} = -\frac{d}{dx}\frac{ae^{ax}}{1+e^{ax}} \\ &= \frac{a^2 e^{-ax}}{(1+e^{-ax})^2}, \end{aligned}$$

which is clearly always larger than zero, so the negative log-likelihood is convex.

When designing a quantum equivalent of the perceptron we would like to retain this convexity, since it makes optimization easy. For the quantum perceptron we will try to stay close to the method of learning probabilities distributions, except that the nature of those probability distributions will no longer be classical, but quantum instead.

1.6 The Quantum Perceptron

In this section we will consider a qubit perceptron at finite temperature that uses a generalization of the classical likelihood function for learning. For this description we use density matrices as a generalized probability framework. Density matrices are used in quantum mechanics to describe statistical ensembles of quantum states (See Appendix A.3). We will have to formulate the classification problem in terms of these density matrices and come up with a learning rule that preserves Hermiticity,

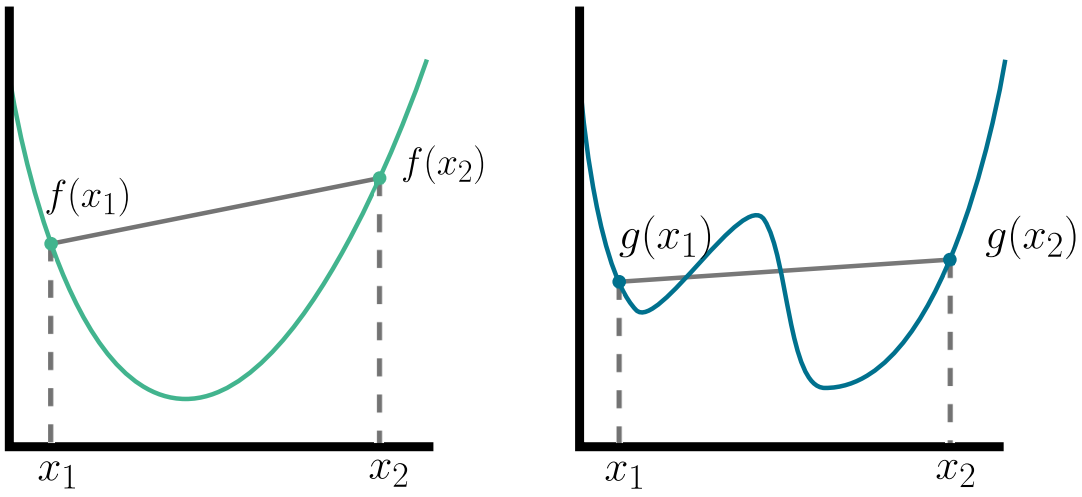


FIGURE 1.4: Schematic representation of the idea of complexity. The function f is always below the chord between the points $(x_1, f(x_1))$ and $(x_2, f(x_2))$. For g this is not the case. The figure on the left has a global minimum that can be found by following the gradient of the function f. The figure on the right has a local minima that represents a suboptimal set of parameter values.

positive semidefiniteness and trace.

Consider again a classification problem where we have a data set consisting of input vectors $\mathbf{x} \in \mathbb{R}^d$ of length d with corresponding labels $y \in \{1, -1\}$. For this problem the classical negative log-likelihood is given by

$$\mathcal{L}_{cl} = -\sum_{\mathbf{x}} q(\mathbf{x}) \sum_{y} q(y|\mathbf{x}) \log(p(y|\mathbf{x}; \mathbf{w})). \tag{1.10}$$

Here, $q(\mathbf{x})$ is the probability of observing $\mathbf{x}$, $q(y|\mathbf{x})$ is the conditional probability of observing label y for data $\mathbf{x}$ and $p(y|\mathbf{x}, \mathbf{w})$ is the proposed model conditional probability distribution of the data.

1.6.1 Quantum Data

To extend the classical likelihood in equation 1.10 to the realm of quantum mechanics we will describe the model and the conditional probability $q(y|\mathbf{x})$ in terms of density matrices. Consider a classical distribution over two variables, $q(y = 1|\mathbf{x})$ and $q(y = -1|\mathbf{x})$. For each $\mathbf{x}$, $q(y|\mathbf{x})$ is a distribution over a single bit y, and is fully determined by its conditional expectation value of y given $\mathbf{x}$ written as $b(\mathbf{x})$,

$$\begin{aligned} q(y|\mathbf{x}) &= \frac{1}{2}(1 + b(\mathbf{x})y), \\ \text{with } b(\mathbf{x}) &= \frac{1}{M}\left(\sum_{\mathbf{x}'} y' \mathbb{I}(\mathbf{x}' = \mathbf{x})\right), \\ \text{and } M &= \sum_{\mathbf{x}'} \mathbb{I}(\mathbf{x}' = \mathbf{x}). \end{aligned} \tag{1.11}$$

We count how many times label y occurs for some sample $\mathbf{x}$ and divide it by M, the total number of times the sample appears in the data. We define the empirical

probability

$$q(\mathbf{x}) = \frac{M}{N},$$

for M occurrences of $\mathbf{x}$, and N the total number of samples. We now note that the diagonal of a density matrix defines a classical probability distribution, since its entries must be positive and sum to one due to the trace constraint. In other words, we can write the classical distribution $q(y|\mathbf{x})$ as a mixed state density matrix,

$$\eta_{cl} = q(y=1|\mathbf{x})\,|1\rangle\langle 1| + q(y=-1|\mathbf{x})\,|0\rangle\langle 0| = \begin{pmatrix} q(y=1|\mathbf{x}) & 0 \\ 0 & q(y=-1|\mathbf{x}) \end{pmatrix}.$$

Instead of representing the data with a statistical ensemble $q(y|\mathbf{x})$, we represent the data with the following wave function:

$$|\psi\rangle = \sqrt{q(y=1|\mathbf{x})}\,|1\rangle + \sqrt{q(y=-1|\mathbf{x})}\,|0\rangle\,.$$

The density matrix belonging to this pure state is given by

$$\eta_{\mathbf{x}} = |\psi\rangle\langle\psi| = \begin{pmatrix} q(y=1|\mathbf{x}) & \sqrt{q(y=1|\mathbf{x})}\sqrt{q(y=-1|\mathbf{x})} \\ \sqrt{q(y=1|\mathbf{x})}\sqrt{q(y=-1|\mathbf{x})} & q(y=-1|\mathbf{x}) \end{pmatrix}.$$

Which reduces to a classical probability distribution if $q(y=1|\mathbf{x}) \in \{0,1\}$, so that the matrix becomes diagonal.

All density matrices can be represented by a vector in the Bloch sphere (see Appendix A.4). If we look at the Bloch sphere in figure 1.5, we see that we can only represent the data distribution along the z-axis, $\mathbf{r}_{cl} = (0,0,q(y=1|\mathbf{x}) - q(y=-1|\mathbf{x})) = (0,0,2q(y=1|\mathbf{x})-1)$. However, the quantum representation of the data makes use of an extra degree of freedom on the Bloch sphere as can be seen in figure 1.6.
We now propose a quantum model parametrized by a density matrix $\rho(\mathbf{x},\mathbf{w};y,y') \equiv \rho_{\mathbf{x}}$,

$$\rho_{\mathbf{x}} = \frac{1}{Z}e^{-\beta H},$$

where $H = \sum_k h^k\sigma^k$, with $h^k \in \mathbb{R}$ and σ^k the Pauli matrices with $k = (x,y,z)$ from equation A.7. This is a finite temperature description of a qubit, where we will set $\beta = -1$ for now. If $\rho_{\mathbf{x}}$ is diagonal, this model reduces to the Boltzmann distribution of equation 1.7. We can absorb the inverse temperature $-\beta$ in the field $-\beta h^k \to h^k$ by rescaling the weights $\mathbf{w}^k$. Using that

$$e^{a\hat{n}\cdot\sigma} = \cosh a + \sinh a \sum_k \sigma^k,$$

and writing $\sum_k h^k\sigma^k = h\sum_k \frac{h^k}{h}\sigma^k =$ with $h = \sqrt{\sum_k (h^k)^2}$ we find

$$\rho_{\mathbf{x}} = \frac{1}{Z}\left(\cosh h + \sinh h \sum_k \frac{h^k\sigma^k}{h}\right).$$

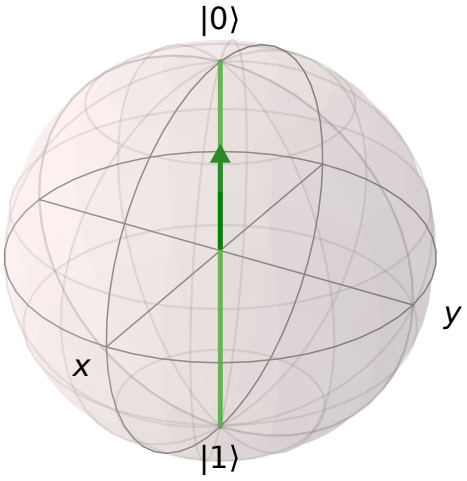


FIGURE 1.5: Classical data density matrix. The Bloch vector is confined to z-axis (light green) and has length $|\mathbf{r}| \leq 1$. Since the data density matrix is always real, the vector must have $r_y = 0$

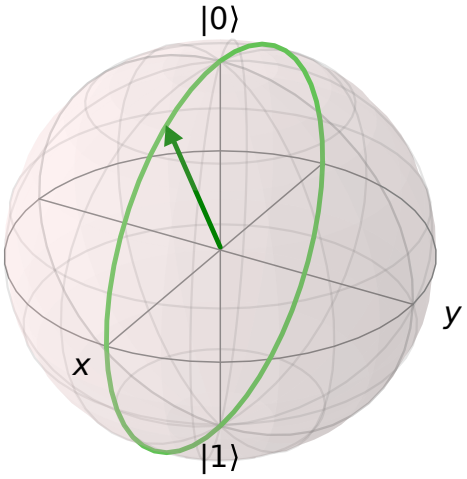


FIGURE 1.6: Quantum data density matrix. The Bloch vector is confined to the surface(light green) of the sphere and has length $|\mathbf{r}| = 1$.

Solving $\mathrm{Tr}\{\rho\} = 1$ gives $Z = 2\cosh h$. If we plug this in we get

$$\begin{aligned} \rho_{\mathbf{x}} &= \frac{1}{2}\mathbb{1} + \frac{1}{2}\tanh h \sum_k \frac{h^k \sigma^k}{h} \\ &= \frac{1}{2}\left(\mathbb{1} + \sum_k m^k \sigma^k\right), \end{aligned} \tag{1.12}$$

where $\mathbb{1}$ is a 2×2 identity matrix and $m^k = \tanh h\frac{h^k}{h}$. Equation 1.12 gives us the general description of qubit that we established in section A.4. This definition spans the full space of 2×2 Hermitian matrices, for all $h^k \in \mathbb{R}$. From the definition of m^k it is clear that $m^k \in (-1, 1)$. This means that $\rho_{\mathbf{x}}$ is positive semidefinite because the eigenvalues of $\rho_{\mathbf{x}}$ are $\lambda_{\pm} = \frac{1}{2}(1 \pm \sqrt{\sum_k (m^k)^2}) \geq 0$. From the eigenvalues we also see that $\rho_{\mathbf{x}}$ describes a mixed state, since it is only rank one if $\sum_k (m^k)^2 = 1$. We parametrize the field $h^k \to h^k(\mathbf{x})$ by setting $h^k(\mathbf{x}) = \mathbf{w}^k \cdot \mathbf{x}$ with $\mathbf{w}^k \in \mathbb{R}^d$, so that the qubit state is dependent on classical input data. To clean up the notation we omit the argument of h^k.

To summarize, the model density matrix can learn a mixed state density matrix that corresponds best to the quantum representation of the data. Since it is mixed, we can move in the interior of the circle of the Bloch sphere, as can be seen in figure 1.7. The classical perceptron tries to minimize the likelihood with respect to a classical distribution, that is located on a single axis with interval $[0, 1]$. The quantum description of the data is more expressive and contains additional information about the underlying distribution. We will now construct a cost function that minimizes the distance between density matrices, analogous to the cross entropy. This model will make use of the additional degree of freedom on the Bloch sphere to provide a more expressive model.

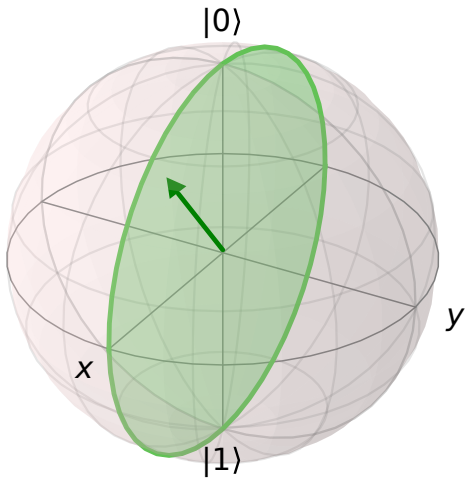


FIGURE 1.7: Quantum perceptron model distribution. The Bloch vector can now also lie in the interior of the surface (light green) of the sphere and has length $|\mathbf{r}| \leq 1$. We will show later on that the r_y component will always go to zero, so the vector is confined to the xz-plane

1.6.2 Quantum Likelihood

We define the quantum conditional likelihood as a cross entropy between a conditional data density matrix $\eta_{\mathbf{x}}$ and a model conditional density matrix $\rho_{\mathbf{x}}$, analogous to equation 1.10,

$$\mathcal{L}_q = -\sum_{\mathbf{x}} q(\mathbf{x}) \operatorname{Tr}\{\eta_{\mathbf{x}} \log(\rho_{\mathbf{x}})\}. \tag{1.13}$$

This is the quantum mechanical equivalent of the classical log-likelihood which minimizes the "distance" between the density matrix representations of the data and the model. Next, we rewrite this with the parametrized $\rho_{\mathbf{x}}$,

$$\begin{aligned}\mathcal{L}_q &= -\sum_{\mathbf{x}} q(\mathbf{x}) \operatorname{Tr}\{\eta_{\mathbf{x}} \log(\rho_{\mathbf{x}})\} \\ &= -\sum_{\mathbf{x}} q(\mathbf{x}) \sum_{y,y'} \langle y' | \sqrt{q(y|\mathbf{x})}\sqrt{q(y'|\mathbf{x})} \log(\rho_{\mathbf{x}}) |y\rangle ,\end{aligned}$$

with $\{|y\rangle\}$ a set of orthonormal vectors in the σ^z basis. Then,

$$= -\sum_{\mathbf{x}} q(\mathbf{x}) \sum_{y,y'} \sqrt{q(y|\mathbf{x})}\sqrt{q(y'|\mathbf{x})} \langle y' | \left(\sum_k h^k \sigma^k - \log(2\cosh h) \right) |y\rangle . \tag{1.14}$$

Calculating the statistics for the Pauli matrices gives

$$\sum_{y,y'} \langle y' | \sum_k h^k \sigma^k |y\rangle = \sum_{y,y'} \sum_k \langle y' | h^k \sigma^k |y\rangle ,$$

which gives three delta functions that we can plug into equation 1.14 together with the definition of $q(y|\mathbf{x})$ from equation 1.11. Then,

$$\begin{aligned}&\sum_{y,y'} \sqrt{q(y|\mathbf{x})}\sqrt{q(y'|\mathbf{x})} \left(h^x \delta_{y',-y} + iyh^y \delta_{y',-y} + yh^z \delta_{y',y} \right) \\ &= \sum_y h^x \frac{1}{2}\sqrt{1+b(\mathbf{x})y}\sqrt{1-b(\mathbf{x})y} + iyh^y \frac{1}{2}\sqrt{1+b(\mathbf{x})y}\sqrt{1-b(\mathbf{x})y} + yh^z \frac{1}{2}\sqrt{1+b(\mathbf{x})y}\sqrt{1+b(\mathbf{x})y} \\ &= h^x \sqrt{1-b(\mathbf{x})^2} + h^z b(\mathbf{x}).\end{aligned}$$

The h^x term quantifies how often a sample occurs with a flipped output label and is the distinguishing factor from the classical perceptron. The source of this term is the σ^x matrix in the likelihood which flips the state $|y\rangle$ and scales h^x with the off-diagonal elements of $\eta_{\mathbf{x}}$. As a final likelihood we get

$$\mathcal{L}_q = -\sum_{\mathbf{x}} q(\mathbf{x}) \left(h^x \sqrt{1-b(\mathbf{x})^2} + h^z b(\mathbf{x}) - \log(2\cosh h) \right). \tag{1.15}$$

In order to perform learning we have to find update rules that minimize the function in equation 1.15. To find the minimum we perform gradient descent to update the parameters $\mathbf{w}^k$. Derive with respect to $\mathbf{w}^k$ to obtain

$$\begin{aligned}\frac{\partial \mathcal{L}_q}{\partial \mathbf{w}^x} &= -\sum_{\mathbf{x}} q(\mathbf{x}) \left(\sqrt{1-b(\mathbf{x})^2} - \tanh h \frac{h^x}{h} \right) \mathbf{x}, \\ \frac{\partial \mathcal{L}_q}{\partial \mathbf{w}^y} &= \sum_{\mathbf{x}} q(\mathbf{x}) \tanh h \frac{h^y}{h} \mathbf{x}, \\ \frac{\partial \mathcal{L}_q}{\partial \mathbf{w}^z} &= -\sum_{\mathbf{x}} q(\mathbf{x}) \left(b(\mathbf{x}) - \tanh h \frac{h^z}{h} \right) \mathbf{x},\end{aligned} \tag{1.16}$$

and update the weights at iteration t with

$$\mathbf{w}^k(t+1) = \mathbf{w}^k(t) - \epsilon \left(\frac{\partial \mathcal{L}}{\partial \mathbf{w}^k(t)} \right). \tag{1.17}$$

These are the learning rules for the quantum perceptron, with learning parameter ϵ for each gradient. Since the gradient step of $\mathbf{w}^y$ is proportional to $\mathbf{w}^y$, the fixed point solution is $\mathbf{w}^y \to \mathbf{0}$ in the limit of many iterations. In the case that there exists a function $f(\mathbf{x}) = y$ (no noise in the data) for all data points, the statistics $b(\mathbf{x})$ become either 1 or -1, which gives a fixed point solution $\mathbf{w}^x \to \mathbf{0}$. The z field then corresponds to the single field of a classical perceptron (see section 1.4) and the quantum perceptron approaches the classical case. However, in the case that there are samples which have both 1 and -1 labels, the weight $\mathbf{w}^x$ becomes finite and the solution of the quantum perceptron will diverge from the classical perceptron. This change in behaviour is reflected in the probability boundaries, which differ from the classical case as we will see in section 1.7.1.

1.6.3 Convexity of the Quantum Log-Likelihood

To prove the convexity of the likelihood function, we only look at the expression $\eta_{\mathbf{x}} \log(\rho_{\mathbf{x}})$. It is known that the relative entropy

$$S(\rho||\sigma) = \mathrm{Tr}\{\rho \log(\rho)\} - \mathrm{Tr}\{\rho \log(\sigma)\},$$

is jointly convex [55]. This means that for any $\rho_0, \rho_1, \sigma_0, \sigma_1 \geq 0$

$$S((1-t)\rho_0 + t\rho_1||(1-t)\sigma_0 + t\sigma_1) \leq (1-t)S(\rho_0||\sigma_0) + tS(\rho_1||\sigma_1),$$

which implies that $S(\rho||\sigma)$ is strictly convex in ρ and that thus $\eta_{\mathbf{x}} \log(\rho_{\mathbf{x}})$ is convex in $\rho_{\mathbf{x}}$. Similar to the classical case, the cost function is convex, so we will find a global minimum with gradient descent.

1.7 Predicting Classes

Once the model is trained, we can construct a state $\rho_{\mathbf{x}}$ for the qubit based on some input $\mathbf{x}$. From the definition in equation 1.12 we get

$$\rho_{\mathbf{x}} = \frac{1}{2}\begin{pmatrix} 1+m^z & m^x - im^y \\ m^x + im^y & 1-m^z \end{pmatrix}.$$

The eigenvectors of the learned density matrix correspond to the physical states of the model qubit system. By decomposing the density matrix into rank one density matrices composed of the eigenvectors $\{|\psi\rangle\}$ we get

$$\rho = \sum_i p_i \rho_i = \sum_i p_i |\psi_i\rangle \langle\psi_i| = \sum_i \lambda_i^2 |\psi_i\rangle \langle\psi_i|. \tag{1.18}$$

where the probabilities p_i correspond to the squares of the respective eigenvalues λ_i for each eigenvector $|\psi_i\rangle$ of the density matrix. The learned model thus contains two levels of probabilities:

a) λ_i^2: the classical probability of finding the system in eigenstate i

b) $|\psi_i\rangle$: the quantum probability of finding either a $|0\rangle$ or $|1\rangle$ state.

So how do we obtain a class prediction from this model? As we will see, there are several methods to do this. Each method has different qualitative properties that we will explore in the next sections.

1.7.1 Eigenvectors

To start off, we will determine the eigenvalues and eigenvectors of the model,

$$\lambda_\pm = \frac{1}{2}\left(1 \pm \sqrt{\sum_k m^k}\right) = \frac{1}{2}(1 \pm \tilde{m}). \tag{1.19}$$

The orthonormal eigenvectors corresponding to each eigenvalue are

$$\begin{aligned} \mathbf{v}_+ &= \frac{1}{Z}\begin{pmatrix} m^z + \tilde{m} \\ m^x + i m_y \end{pmatrix}, \\ \mathbf{v}_- &= \frac{1}{Z}\begin{pmatrix} -m^x + i m^y \\ m^z + \tilde{m} \end{pmatrix}. \end{aligned} \tag{1.20}$$

with the normalization factor given by $Z = \sqrt{\mathbf{v}_\pm \cdot \mathbf{v}_\pm^*}$,

$$\begin{aligned} \mathbf{v}_\pm \cdot \mathbf{v}_\pm^* &= \left((m^z)^2 + \tilde{m}^2 + 2m^z\tilde{m} + (m^x)^2 + (m^y)^2\right) \\ &= 2\tilde{m}(\tilde{m} + m^z), \end{aligned}$$

which gives the eigenvectors the desired properties $\langle \mathbf{v}_+, \mathbf{v}_- \rangle = \delta_{+-}$. There appear to be singularities for

$$\begin{aligned} m^z = \tilde{m} &= \sqrt{\sum_k (m^k)^2} \\ (m^z)^2 &= \sum_k (m^k)^2 \\ (m^x)^2 &= -(m^y)^2 \end{aligned}$$

But for $m^k \in \mathbb{R}$ this is only singular for $m^x = m^y = 0$, which is the limiting case for no noise in the data. This gives $\tilde{m} = 1$, so the system become rank one. This could lead to some numerical instability during learning, but no problems have been encountered while obtaining the results in this thesis. The system can only become degenerate in the case that $\tilde{m} \to 0$. This solution is rather unlikely, since this would imply that the fixed point of $\mathbf{w}^z = 0$ which is only possible if $b(\mathbf{x}) = 0$ for all $\mathbf{x}$. In the following sections we will discuss different methods to construct a probability from the eigenvectors.

1.7.2 Proposal 1: Rank One Approximation

The first proposal looks at the largest eigenvector $\mathbf{v}_+$. If $\tilde{m}$ is close to one this eigenvector dominates the system. We can project the eigenstate onto the basis states belonging to the classes $y = \{-1, 1\}$, which are the given by the state vectors $|0\rangle = \mathbf{e}_0$ and $|1\rangle = \mathbf{e}_1$. By taking the square of these projections we get properly defined class probabilities,

$$p(y = 1|\mathbf{x}; \mathbf{w}) = |\mathbf{e}_0 \cdot \mathbf{v}|^2 = |v_1|^2, \tag{1.21}$$

$$p(y = -1|\mathbf{x}; \mathbf{w}) = |\mathbf{e}_1 \cdot \mathbf{v}|^2 = |v_2|^2, \tag{1.22}$$

where $\langle \mathbf{v}, \mathbf{v} \rangle = 1$ by construction, so $|v_1|^2 + |v_2|^2 = 1$. Taking $m^y \to 0$ allows us to write the largest eigenvector from equation 1.20 as

$$p(y|\mathbf{x};\mathbf{w}) = \begin{pmatrix} v_1^2 \\ v_2^2 \end{pmatrix} = \frac{1}{2\tilde{m}(m^z + \tilde{m})} \begin{pmatrix} (m^z + \tilde{m})^2 \\ (m^x)^2 \end{pmatrix}.$$

Using that $(m^x)^2 = \tilde{m}^2 - (m^z)^2 = (\tilde{m} + m^z)(\tilde{m} - m^z)$ we can write

$$p(y|\mathbf{x};\mathbf{w}) = \frac{1}{2\tilde{m}(m^z + \tilde{m})} \begin{pmatrix} m^z + \tilde{m} \\ \tilde{m} - m^z \end{pmatrix} = \frac{1}{2\tilde{m}} \begin{pmatrix} \tilde{m} + m^z \\ \tilde{m} - m^z \end{pmatrix}.$$

This probability can be expressed in terms of the output y.

$$\boxed{\textbf{Proposal 1:} \qquad p(y|\mathbf{x};\mathbf{w}) = \frac{1}{2}(1 + y\frac{m^z}{\tilde{m}})} \tag{1.23}$$

So what kind of model does this give us? We can analytically determine the shape of the boundary separating the two classes, by setting $p(y = 1|\mathbf{x};\mathbf{w}) = p(y = -1|\mathbf{x};\mathbf{w})$.

$$\begin{aligned} \frac{1}{2}(1 + \frac{m^z}{\tilde{m}}) &= \frac{1}{2}(1 - \frac{m^z}{\tilde{m}}) \\ \frac{m^z}{\tilde{m}} &= \frac{h^z}{h} = 0. \end{aligned} \tag{1.24}$$

which is solved for h^z=0. The field

$$h^z = \mathbf{w} \cdot \mathbf{x} = w_0^z + \sum_i w_i^z x_i^\mu,$$

is equal to the equation of a hyperplane, so a linear separation boundary. Let us compare this to the separation boundary for logistic regression. For logistic regression the probability was given by the sigmoid. Setting the probabilities from equation 1.7 for $y = 1$ and $y = -1$ equal gives

$$\frac{1}{1 + e^{-h}} = \frac{1}{1 + e^{h}},$$

which is solved for $h = \mathbf{w} \cdot \mathbf{x} = 0$. This corresponds to a hyperplane, just as for the quantum perceptron. However, this does not imply that $\mathbf{w}^z = \mathbf{w}$, since both weight vectors belong to the minimum of a different cost function, the quantum log-likelihood and the classical log-likelihood, respectively. Only in the case that there is no noise in the data do the weights coincide. However, both algorithms have a probability boundary that is linear. Significant differences between the two algorithms start appearing when we start looking at the probability curves. These curves describe regions of equal probability through the input space. Algebraically this corresponds to

$$p(y = 1|\mathbf{x};\mathbf{w}) = p(y = -1|\mathbf{x};\mathbf{w}) + \epsilon, \tag{1.25}$$

with $\epsilon \in [0, 1]$. Using that $p(y = 1|\mathbf{x};\mathbf{w}) + p(y = -1|\mathbf{x};\mathbf{w}) = 1$ we get

$$\begin{aligned} 2p(y = 1|\mathbf{x};\mathbf{w}) &= p(y = -1|\mathbf{x};\mathbf{w}) + p(y = 1|\mathbf{x};\mathbf{w})\epsilon \\ p(y = 1|\mathbf{x};\mathbf{w}) &= \frac{1}{2}(1 + \epsilon). \end{aligned} \tag{1.26}$$

Solving this equation means that we find the boundary separating the space into two parts. One part of the space gets assigned a probability $p(y = 1|\mathbf{x}; \mathbf{w}) > \frac{1}{2}(1 + \epsilon)$ and $p(y = 1|\mathbf{x}; \mathbf{w}) < \frac{1}{2}(1 + \epsilon)$. Combining equations 1.23 and 1.26 in the limit that $h^y \to 0$ gives

$$\begin{aligned}
\frac{1}{2}\left(1 + \frac{h^z}{\sqrt{(h^x)^2 + (h^z)^2}}\right) &= \frac{1}{2}(1 + \epsilon) \\
\frac{h^z}{\sqrt{(h^x)^2 + (h^z)^2}} &= \epsilon \\
(h^z)^2 &= ((h^x)^2 + (h^z)^2)\epsilon^2 \\
(h^z)^2(1 - \epsilon^2) &= (h^x)^2\epsilon^2.
\end{aligned}$$

We need to keep careful track of the minus signs. In the first line, h^x always gives a positive contribution, so we take take the absolute value. But h^z does have a sign, so its sign not impacted by squaring and taking the roots. This gives

$$\begin{aligned}
h^z &= \pm\frac{\epsilon^2}{\sqrt{1 - \epsilon^2}}|h^x| \\
h^z &= \pm\delta|h^x|.
\end{aligned}$$

This equation defines a hyperplane,

$$\mathbf{w}^z \cdot \mathbf{x} \mp \delta|\mathbf{w}^x \cdot \mathbf{x}| = 0.$$

For $\delta \neq 0$ we also require that

$$\begin{aligned}
\mathbf{w}^x \cdot \mathbf{x} + w_0^x &= 0 \\
\mathbf{w}^z \cdot \mathbf{x} + w_0^z &= 0.
\end{aligned}$$

Both these equations do not depend on δ, so all δ-hypersurfaces intersect in the same subspace. Assuming that $\mathbf{w}^x$ and $\mathbf{w}^z$ are linearly independent, we can solve for 2 of the n variables in $\mathbf{x}$. This means that the hyperplane intersect in a linear subspace spanned by the $n - 2$ free variables of the system of equations. For $n = 2$ this means that they intersect in a single point, for $n = 3$ a line, $n = 4$ a plane and $n = m$ in a m-dimensional hypersurface. We can plot these curves for a simple two-dimensional example with bias. See figure 1.8 for an example of the curves of the quantum perceptron. We can also find the boundary given by logistic regression, for small h we approximate $S(h) \approx h$ to get

$$\begin{aligned}
h &= -h + \epsilon \\
2h - \epsilon &= 2\mathbf{w} \cdot \mathbf{x} - \epsilon = 0,
\end{aligned}$$

which is a plane equation with the origin shifted along in the direction $\epsilon \sum_i \mathbf{e}_i$, so planes parallel to the original separation boundary. Of course the approximation $S(h) \approx h$ completely falls flat for $h > 2$, since we do not capture the asymptotic behaviour. Without the approximation this becomes

$$\frac{1}{1 + e^{-h}} - \frac{1}{1 + e^h} = \epsilon,$$

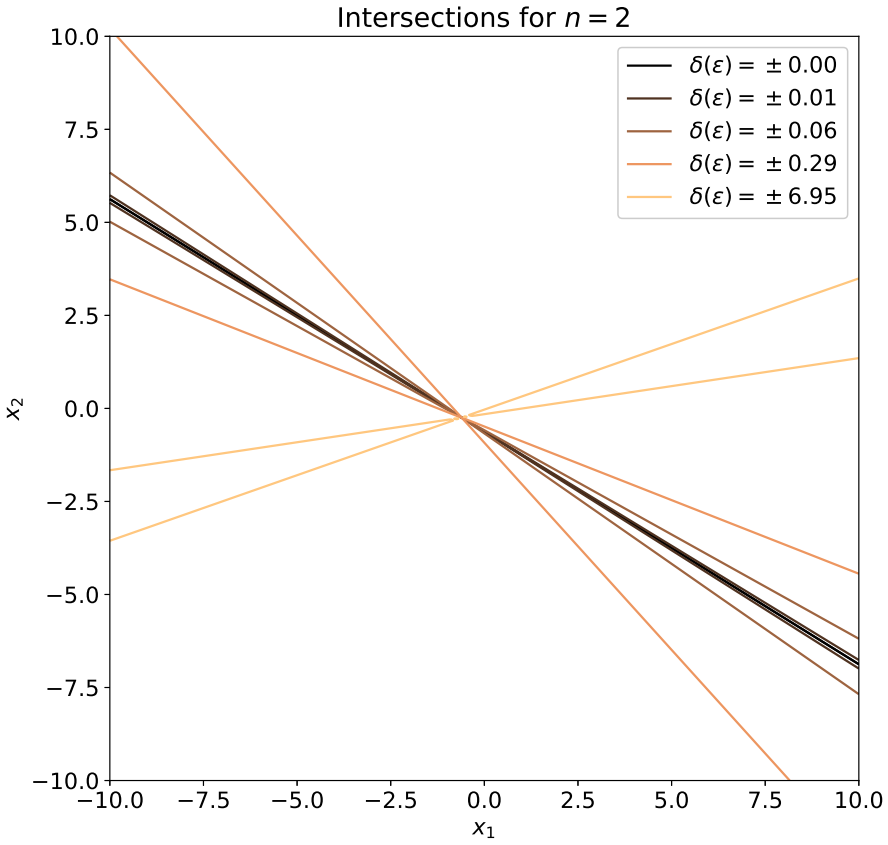


FIGURE 1.8: Separation boundaries for the quantum perceptron based on the rank one approximation. The model is given by $\mathbf{w}^x = (-0.24, 1)$, $w_0^x = 0.1$ and $\mathbf{w}^z = (0.5, 0.8)$, $w_0^z = 0.5$. Since $\mathbf{w}^x$ is non-zero we have a noisy problem. Adding the negative δ contribution provides us with the symmetrical boundaries image. The region in the top right half enclosed by the yellow lines is the region where $p(y = 1|\mathbf{x}; \mathbf{w}) \approx 1$, per equation 1.26. Conversely, for the yellow lines in the bottom left half we have $p(y = -1|\mathbf{x}; \mathbf{w}) \approx 1$

which does not give a nice plane equation, but can be plotted nonetheless. As expected we get linear separation boundaries parallel to the boundary of equal probability. See figure 1.9. The difference between the two models is striking: the classical perceptron can only assign probability boundaries parallel to the separation boundary where the quantum perceptron can assign different titled boundaries to separate areas of probability. In the case of $m^x = 0$, we get

$$p(y|\mathbf{x}; \mathbf{w}) = \frac{1}{2}\left(1 + y\text{sgn}(x)\right),$$

so a hard step function.

1.7.3 Proposal 2: Quantum Statistics as Boundary

A more physically-inspired idea is to measure observables that translate naturally to class probabilities. For this system the easiest observables that we can measure are

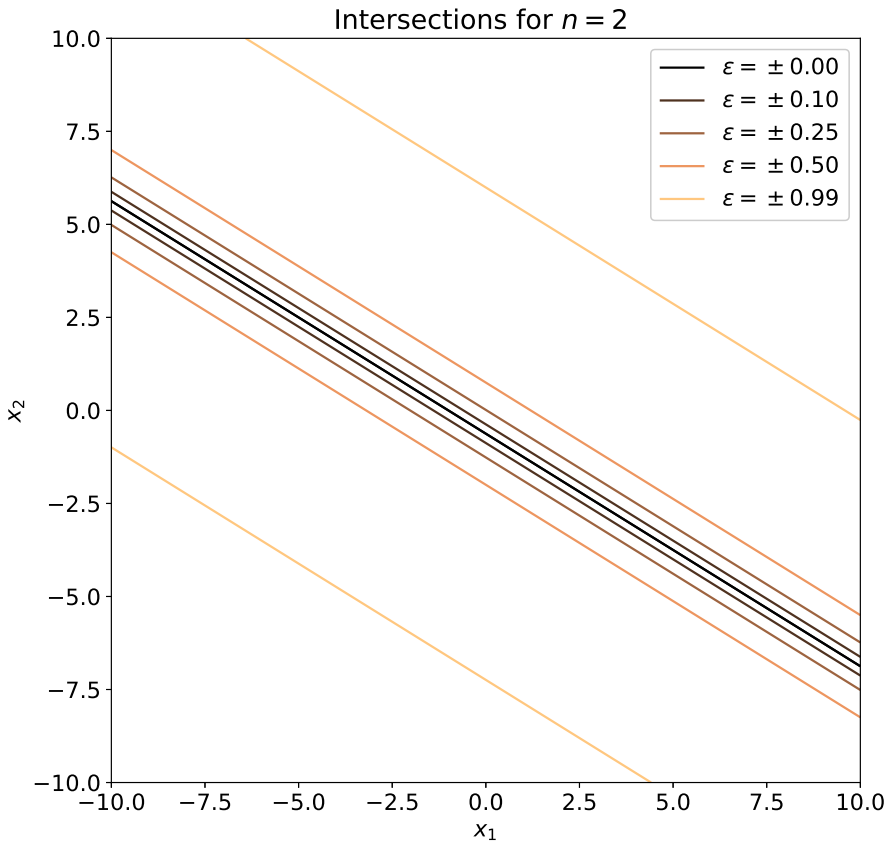


FIGURE 1.9: Separation boundaries for logistic regression. For logistic regression we take the $\mathbf{w}^z$ weights of the quantum perceptron, since in the limit of no noise these weights will be equal. The distance between the ϵ-curves is determined by the sigmoid. The closer ϵ gets to 1, the larger the distance between the lines, which is clear from the shape of the sigmoid and its asymptotic behaviour, as we saw earlier in figure 1.3.

$\langle \sigma^k \rangle$,

$$\langle \sigma^k \rangle = \mathrm{Tr}\Big\{ \sigma^k \rho_{\mathbf{x}} \Big\}.$$

With the above definition of $\rho_{\mathbf{x}}$ we then have

$$\mathrm{Tr}\left\{ \sigma^k \frac{1}{2}(1 + \sum_{k'} m^{k'} \sigma^{k'}) \right\} = \frac{1}{2} \mathrm{Tr}\left\{ \sigma^k + \sum_{k'} m^{k'} \sigma^z \sigma^{k'}) \right\} = \frac{1}{2} \left(\underbrace{\mathrm{Tr}\{\sigma^z\}}_{=0} + \sum_{k'} m^{k'} \underbrace{\mathrm{Tr}\Big\{ \sigma^z \sigma^{k'}) \Big\}}_{2\delta_{kk'}} \right) = m^k.$$

In other words, we can determine the values of m^z by measuring the spin of the qubit in the z direction. These statistics can then be used to construct a probability of finding the class $p(y) = \frac{1}{2}(1 + y m^z)$. Note that these probabilities are equal to the

probabilities of the classical logistic regression in the limit that m^x, $m^y = 0$,

$$\begin{aligned}\frac{1}{2}(1+ym^z) &= \frac{1}{2}\left(1+y\tanh h\frac{h^z}{h}\right) = \frac{1}{2}(1+\tanh yh^z)\\ &= \frac{1}{2}\left(1+\frac{e^{yh^z}-e^{-yh^z}}{e^{yh^z}+e^{-yh^z}}\right) = \frac{1}{2}\left(\frac{e^{yh^z}+e^{-yh^z}+e^{yh^z}-e^{-yh^z}}{e^{yh^z}+e^{-yh^z}}\right) = \frac{e^{yh^z}}{e^{yh^z}+e^{-yh^z}}\\ &= \frac{1}{1+e^{-2yh^z}} = S(2yh^z),\end{aligned}$$

which serves as a motivation for using this description: In the limiting case we get back the logistic regression.

Proposal 2: $$p(y|\mathbf{x};\mathbf{w}) = \frac{1}{2}(1+ym^z)$$

We now note something interesting: we can retrieve this proposal with a simple argument by looking at the eigenvectors and eigenvalues from proposal 1. The smallest eigenvector v_- assigns the same probabilities $p(y)$ as v_+, but switches the classes,

$$\begin{aligned}p(y|\mathbf{x};\mathbf{w}) &= \frac{1}{2\tilde{m}(m^z+\tilde{m})}\begin{pmatrix}(m^x)^2\\(m^z+\tilde{m})^2\end{pmatrix} = \frac{1}{2\tilde{m}}\begin{pmatrix}\tilde{m}-m^z\\\tilde{m}+m^z\end{pmatrix}\\ p(y|\mathbf{x};\mathbf{w}) &= \frac{1}{2}(1-y\frac{m^z}{\tilde{m}}).\end{aligned}$$

From equation 1.18 we can see that the $\lambda_\pm$'s serve as a quantity that tells us how likely one of the eigenstates of the system is. If we take these extra probabilities into account, we get that

$$\begin{aligned}p(y|\mathbf{x};\mathbf{w}) &= \frac{1}{2}(1+y\frac{m^z}{\tilde{m}})\lambda_+ + \frac{1}{2}(1-y\frac{m^z}{\tilde{m}})\lambda_-\\ &= \frac{1}{4}\left((1+y\frac{m^z}{\tilde{m}})(1+\tilde{m})+(1-y\frac{m^z}{\tilde{m}}))(1-\tilde{m})\right)\\ &= \frac{1}{4}\left(1+\tilde{m}+y\frac{m^z}{\tilde{m}}+ym^z+1-\tilde{m}-y\frac{m^z}{\tilde{m}}+ym^z\right)\\ &= \frac{1}{2}(1+ym^z).\end{aligned}$$

We can thus conclude that the weighted eigenvector description coincides with measuring $\langle\sigma^z\rangle$.

As before, we look the separation boundary and probability boundaries that this model gives. From setting $m^z = 0$ it is again clear that the separation boundary is a hyperplane. For the probability boundaries we get

$$\begin{aligned}m^z &= \epsilon\\ \tanh h\frac{h^z}{h} &= \epsilon.\end{aligned}$$

Notice that the hyperbolic tangents and norms do not fall out of the fraction as they did before in equation 1.24. This means that the boundaries will not be linear, but curved instead since they are scaled by a nonlinear term. This can be seen in figure 1.10. The probability boundaries from the quantum perceptron differ significantly

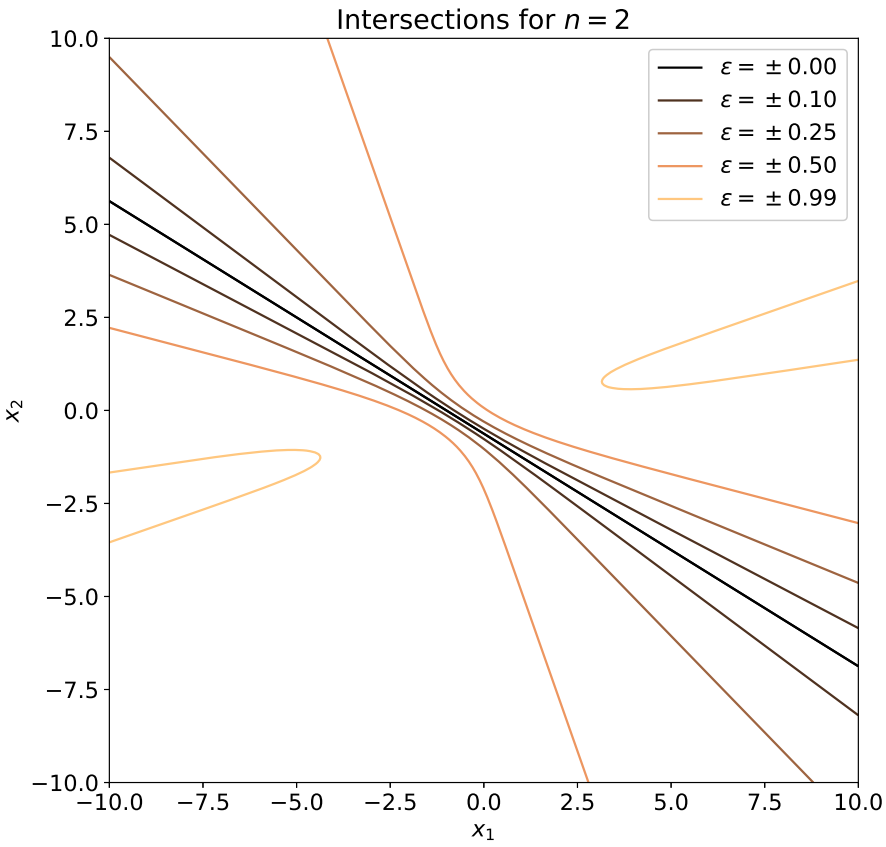


FIGURE 1.10: Separation boundary for quantum perceptron based on m^z statistics. The previously found boundaries of figure 1.8 are now smoothed by the hyperbolic tangent.

from the logistic regression boundaries in the case of noise. However, the $p(y = 1|\mathbf{x};\mathbf{w}) = p(y = -1|\mathbf{x};\mathbf{w})$ boundary is still straight. Since the probability curves are only obtained after learning, we cannot really use them to increase classification performance. However, in the case of noise it is possible that the tilted boundaries of the perceptron are better at assigning a more representative probability to a class.

1.7.4 Proposal 3: Eigenvector Ellipse

For completeness we will state this final method to obtain a probability from the eigensystem, even though it is incredibly convoluted. We can describe an ellipse through the equation $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ with a, b the semi-major and semi-minor axes, respectively. The eigenvectors of a positive definite matrix span such an ellipse, through the quadratic form

$$\mathbf{x}^T A\mathbf{x} = \mathbf{x}^T QDQ^{-1}\mathbf{x} = \mathbf{x}^T \left((\mathbf{v}_+, \mathbf{v}_-) \begin{pmatrix} \lambda_+ & 0 \\ 0 & \lambda_- \end{pmatrix} \begin{pmatrix} \mathbf{v}_+ \\ \mathbf{v}_- \end{pmatrix} \right) \mathbf{x} = 1,$$

with Q the matrix corresponding to the basis transformation to the eigensystem and $Q^{-1} = Q^T$, because the eigenvectors are orthonormal. We see that this description corresponds to an eigenvector rotated to the coordinate system $\mathbf{x}' = \mathbf{x}^T Q$. For the

ellipse we have $a = \frac{1}{\sqrt{\lambda_+}}$ and $b = \frac{1}{\sqrt{\lambda_-}}$. Such an ellipse has an area given by $A = ab\pi$, as was already proved by Archimedes in 250 B.C. [56].

We now want to consider what area of the ellipse is located in a single quadrant of the output space. This could function as a measure of uncertainty, since the smallest eigenvector can be seen to point in the direction of mislabeled sample in the data density matrix space. If this vector is large, then the area of the eclipse increases proportional to the uncertainty in this direction. The quadrants are given by the lines $y = \pm x$, since this determines the threshold probability when looking at the largest eigenvector.

We have to determine the intersection of the ellipse with these quadrants and find the corresponding area. Examples of the eigenvector ellipses in the output space are given in figure 1.11 and 1.12.

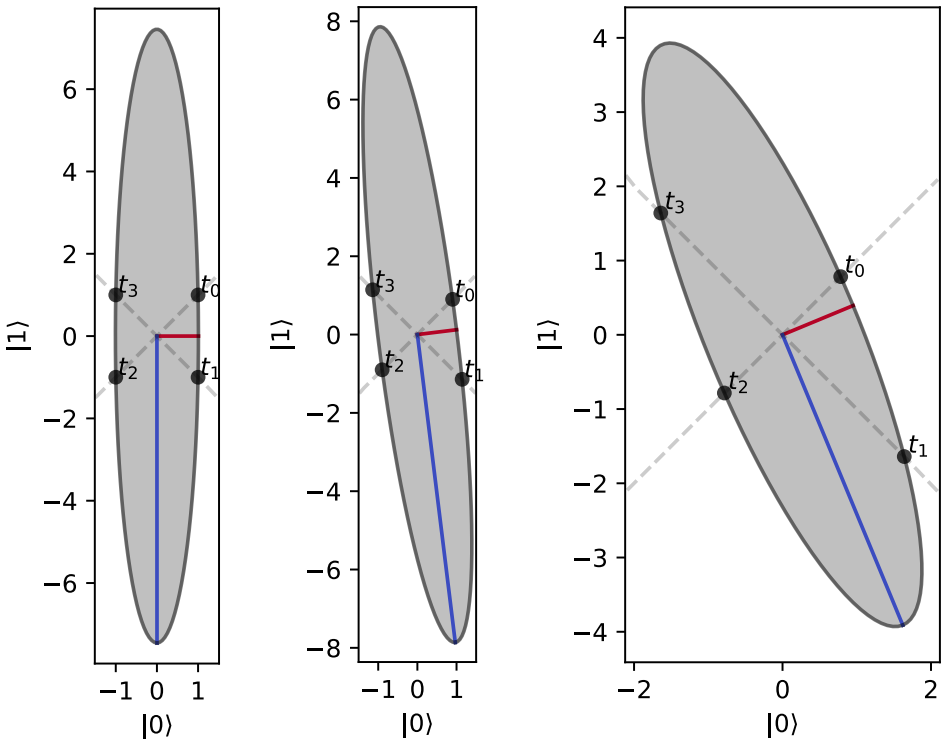


FIGURE 1.11: Ellipses spanned by the eigenvectors. The red and blue vectors are the vectors $\mathbf{v}_+$ and $\mathbf{v}_-$ respectively and the x-axis and y-axis correspond to the directions of $|0\rangle, |1\rangle$ respectively. The vectors are scaled with their corresponding eigenvalues $\lambda_\pm$. The gray dotted lines indicate the quadrants $y = \pm x$ where the intersections are indicated with the dots. The smallest eigenvector increases in length as the amount of uncertainty about the state changes. The ratio of the areas that lie between the lines $y = x\pm$ and the ellipse gives a class label that takes uncertainty into account, since this ratio is affected by the width and rotation of the ellipse. In the left panel we see the eigenvector system for $\mathbf{m} = (10^{-4}, 0, 0.96)$. Since $m^x \approx 0$, the directions of the eigenvectors align with the directions of the canonical basis $\{|0\rangle, |1\rangle\}$. Even thought there is no uncertainty concerning flipped labels, there is still uncertainty about the correct label, which is indicated by the small area of the ellipse between quadrants t_0, t_1 and t_2, t_3 ($|0\rangle$-quadrant). The system is close to being pure since $\|\mathbf{m}\| = 0.96 < 1$. In the middle panel, $\mathbf{m} = (0.44, 0, 0.87)$. Here, we have $m^x \neq 0$ so there are flipped labels which cause a rotation of the ellipse. As a result, more of the mass of the ellipse enters the $|0\rangle$-quadrant. The norm increases to $\|\mathbf{m}\| = 0.97$ so the system becomes purer. Finally, in the left panel we have $\mathbf{m} = (0.63, 0, 0.63)$

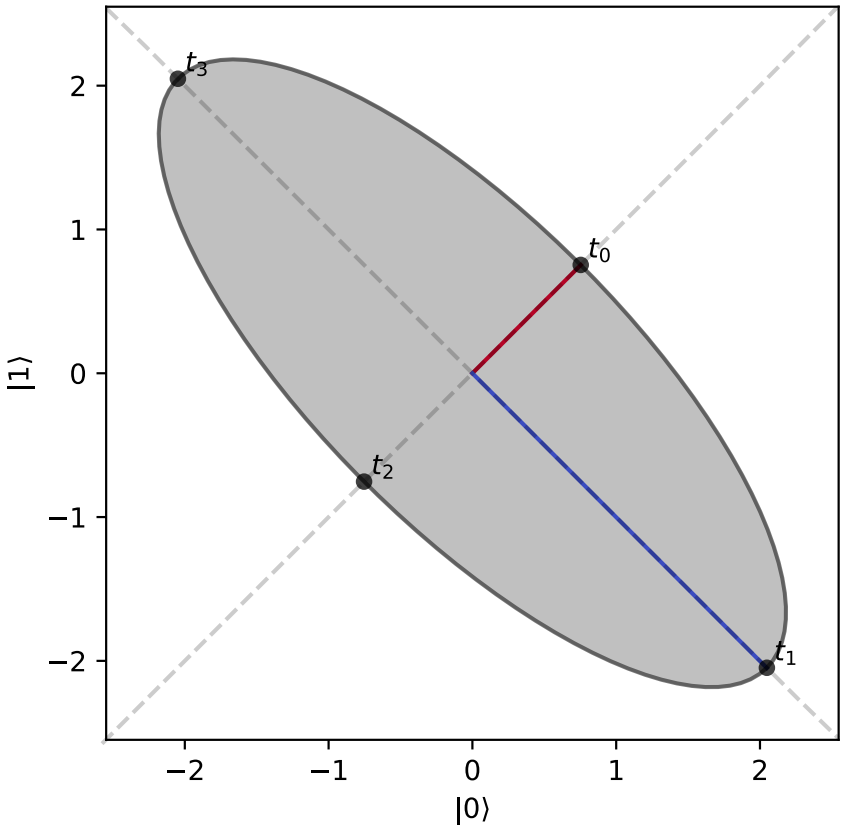


FIGURE 1.12: Ellipses spanned by the eigenvectors. In this final example we have $\mathbf{m} = (0.76,\ 0,\ 10^{-4})$. Since $m^z \approx 0$ we have maximum uncertainty about the class, which is why the eigenvectors align with the quadrants.

We use the parametric description of the ellipse to calculate the area in a quadrant,

$$\mathbf{x}_{\text{ellipse}}(t) = \frac{1}{\sqrt{\lambda_+}}\mathbf{v}_+ \cos t + \frac{1}{\sqrt{\lambda_-}}\mathbf{v}_- \sin t \tag{1.27}$$

which when separated in x,y components looks as follows

$$x(t) = \overbrace{\frac{1}{\sqrt{\lambda_+}}\mathbf{v}_+^x}^{a} \cos t + \overbrace{\frac{1}{\sqrt{\lambda_-}}\mathbf{v}_-^x}^{b} \sin t = a\cos t + b\sin t,$$
$$y(t) = \overbrace{\frac{1}{\sqrt{\lambda_+}}\mathbf{v}_-^y}^{c} \cos t + \overbrace{\frac{1}{\sqrt{\lambda_-}}\mathbf{v}_+^y}^{d} \sin t = c\cos t + d\sin t.$$

We solve the equation $y = x$ in Mathematica to get

$$t_{0,1} = 2 \arctan \frac{b - d \pm \sqrt{a^2 + b^2 + c^2 + d^2 - 2(ac - bd)}}{a - c}$$

Solving $y = -x$ gives

$$t_2 = -\arctan \frac{a + c}{b + d}$$
$$t_3 = \pi - \arctan \frac{a + c}{b + d}.$$

This gives us the 4 intersections of the ellipse with the lines separating the quadrants. We can now conveniently integrate this ellipse between two intersections at times t_i, t_j where $i \neq j$ to find the area in a specific quadrant. For this we use Green's theorem,

$$\oint_C (Ldx + Mdy) = \iint_D \left(\frac{\partial M}{dx} - \frac{\partial L}{\partial y} \right) dA,$$

which allows us to calculate the area A through the line integral if we choose $\left(\frac{\partial M}{dx} - \frac{\partial L}{\partial y} \right) = 1$, which can be done by setting $L = \frac{1}{2}y$ and $M = -\frac{1}{2}x$,

$$\frac{1}{2} \oint_C (ydx - xdy) = \iint_D dA = A.$$

Clearly we can use this method to calculate the area spanned by the curve between two points. Plugging in the parametric equations $x(t)$ and $y(t)$ gives

$$\begin{aligned} A(t_i, t_f) &= \frac{1}{2} \int_{x(t_i)}^{x(t_j)} (y(t)dx(t) - x(t)dy(t)) = \frac{1}{2} \int_{t_i}^{t_j} \left(y(t) \frac{x(t)}{dt} - x(t) \frac{dy(t)}{dt} \right) dt \\ &= \frac{1}{2} \int_{t_i}^{t_j} (bc - ad)(\sin^2(t) + \cos^2(t)) = \frac{1}{2}(bc - ad)(t_f - t_j), \end{aligned}$$

which gives back the Archimedes' result for the total area for the interval $[0, 2\pi]$:

$$2\pi(bc - ad) = \frac{2\pi}{\sqrt{\lambda_+ \lambda_-}} (\mathbf{v}_-^x \mathbf{v}_-^y - \mathbf{v}_+^x \mathbf{v}_+^y) = \frac{\pi}{\sqrt{\lambda_+ \lambda_-}}.$$

where $(\mathbf{v}_-^x \mathbf{v}_-^y - \mathbf{v}_+^x \mathbf{v}_+^y) = 1$ since $\mathbf{v}_-$ and $\mathbf{v}_+$ are orthonormal.

To calculate the amount of probability we determine the ratio of the surfaces in the quadrant of a certain state.

Proposal 3: $$p(y = 1|\mathbf{x}; \mathbf{w}) = \left| \frac{A(t_0, t_3)}{\frac{1}{2} A_{tot}} \right| = \frac{(t_3 - t_0)}{(t_3 - t_1)}$$

Similar as before we plot the resulting probability curves in figure 1.13.

1.7.5 Choosing a class probability

Looking at these different proposals, using the m^z statistic is the most natural approach that makes the least assumptions. It can directly be connected to the physical

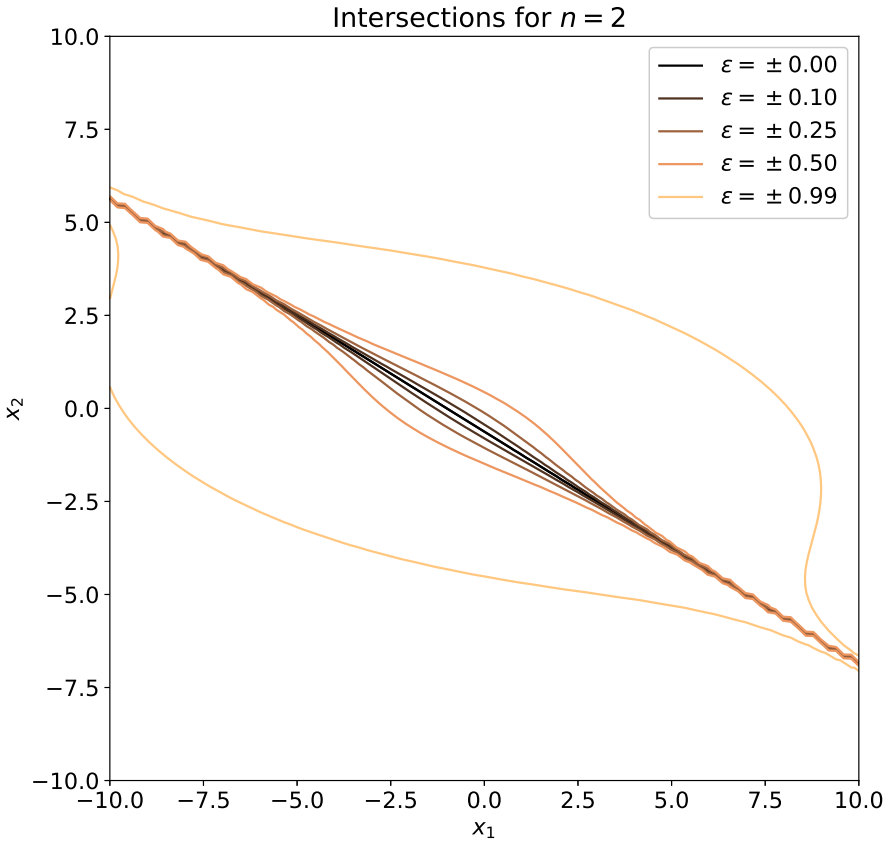


FIGURE 1.13: Separation boundaries for probabilities constructed from the eigenvector ellipse. The line of equal probability is straight as before. It is hard to say something concrete about the shape of the probability boundaries, due to the complexity of the expressions. It seems that the asymptotic behaviour for $\|\mathbf{x}\| \gg 1$ is that the probability boundaries coincide with the separation boundary.

world by measuring $\langle \sigma^z \rangle$. Also, these class probabilities simply correspond to the diagonal values of $\rho_{\mathbf{x}}$, a fact that will be useful when we extend the model to multiple classes. This approach can be seen as a direct generalization of logistic regression, whereas Proposal 1 is a generalization of the hard sign function. Proposal is more a mathematical exercise, then a well motivated physical measure of probability. For these reasons, we will use Proposal 2 throughout the rest of this thesis as the class probability.

1.8 Results

In this section we apply the quantum perceptron to some toy data sets and compare with the classical perceptron with a sigmoid activation function, i.e., logistic regression. For both the classical and quantum perceptron we look at the mean squared

error (MSE) to evaluate the performance of both methods,

$$\text{MSE} = \frac{1}{N} \sum_{(\mathbf{x},y)} \left(y - p(y|\mathbf{x};\boldsymbol{\theta})\right)^2 .$$

For each problem we worked with a test set of 20% of the data. We always reach the global minimum through batch gradient descent because the cost functions are convex for both models. The algorithm is considered converged if the difference of the quantum or classical likelihood $\Delta\mathcal{L} < 10^{-7}$. The learning parameter is set to $\epsilon = 0.01$ for both algorithms.

1.8.1 Simple Two-Dimensional Binary Problem

In order to demonstrate the difference between the classical and quantum perceptron we consider a two-dimensional binary classification problem. If the problem is linearly separable the classical perceptron converges to a solution where the two classes are perfectly separated. In the case where some samples are mislabeled the quantum perceptron should behave differently, because we account for noise in the learning rule. Consider the data

$$\mathbf{x} = \{(1,1),(1,-1),(-1,1),(-1,-1)\},$$

with labels

$$y = \{-1,1,-1,-1\}.$$

This problem is trivial since it is linearly separable and all algorithms converge to the same solution ($\mathbf{w}^{x,y} = 0$ and $\mathbf{w}_z \approx \mathbf{w}_{cl}$). However, if we flip some of the output labels to simulate mislabeled samples or errors in the data, we suspect that the quantum perceptron will perform better. We make 40 copies of the 4 data points in the binary feature space and for $\mathbf{x} \in \{(1,-1),(-1,-1)\}$ we flip 30% of the outputs from -1 to 1. The probability boundaries of the perceptrons differ significantly, as can be seen in figure 1.14, which leads to a better assignment of probability the correct states.

1.8.2 Binary Teacher-Student Problem

A more complex, higher dimensional problem is the Teacher-Student problem where we take a random weight vector $\mathbf{w}_{teacher} \sim \mathcal{N}(0,1)$ and determine labels $y = \text{sgn}(\mathbf{x} \cdot \mathbf{w}_{teacher})$. The input data $\mathbf{x} \in \mathbb{R}^d$ consists of 600 random binary vectors of length $d = 8$, where $\mathbf{x} \in \{-1,1\}^d$. We then create 5 duplicates of each input vector to ensure that there are multiple copies of each sample and attempt to learn 100 different problems where in each run we flip some percentage of the labels. This asserts whether the algorithms can still find the original separation of the data even if noise is introduced. The performance of the quantum perceptron and classical perceptron is compared in figure 1.15.

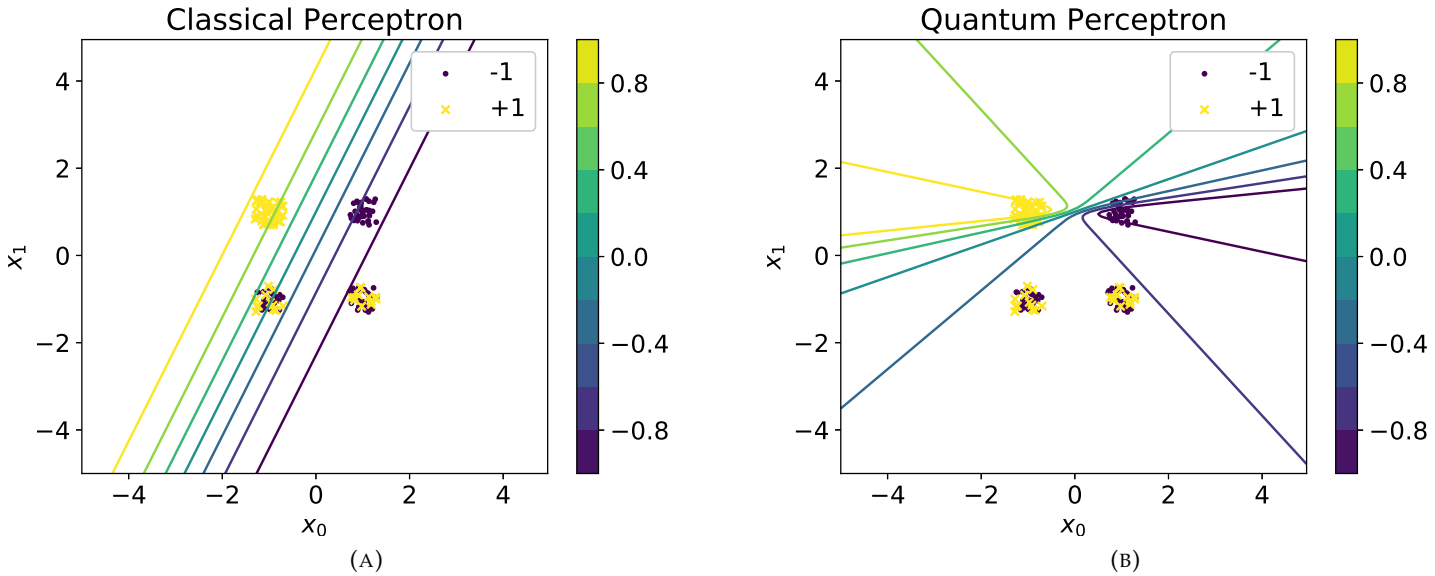


FIGURE 1.14: Separation boundaries in the input space for a two-dimensional problem with $\mathbf{x} = (x_0, x_1)$. The contour lines indicate the expectation value $\mathbb{E}[y|\mathbf{x};\mathbf{w}]_p \in (-1, 1)$. The 0.0-line indicates the separation boundary where $p(y = 1|\mathbf{x};\mathbf{w}) = p(y = -1|\mathbf{x};\mathbf{w}) = \frac{1}{2}$. Jitter is added to the plot to clarify which samples are noisy. **(a)** The classical perceptron assigns linear boundaries through the input space, where the distance between the boundaries is scaled with the sigmoid. **(b)** The quantum perceptron assigns curved boundaries through the input space. Samples with mislabelings get assigned a lower expectation value which results in a lower MSE of $\text{MSE}(quantum) \approx 0.106$ for the quantum perceptron versus $\text{MSE}(classical) \approx 0.154$ for the classical perceptron. Note that if we threshold the quantum perceptron boundary at $p(y = 1|\mathbf{x};\boldsymbol{\theta}) = 0.5$, we get a linear boundary that would assign similar classes as in figure (a), even though the boundary is tilted with respect to the classical boundary. However, the quantum perceptron assigns high probabilities to classes about which it is certain ($\mathbf{x} \in \{(-1, 1), (1, 1)\}$) and lower probabilities to classes about which it is uncertain ($\mathbf{x} \in \{(-1, -1), (1, -1)\}$). The classical perceptron does this significantly worse, which is reflected in the difference in MSE.

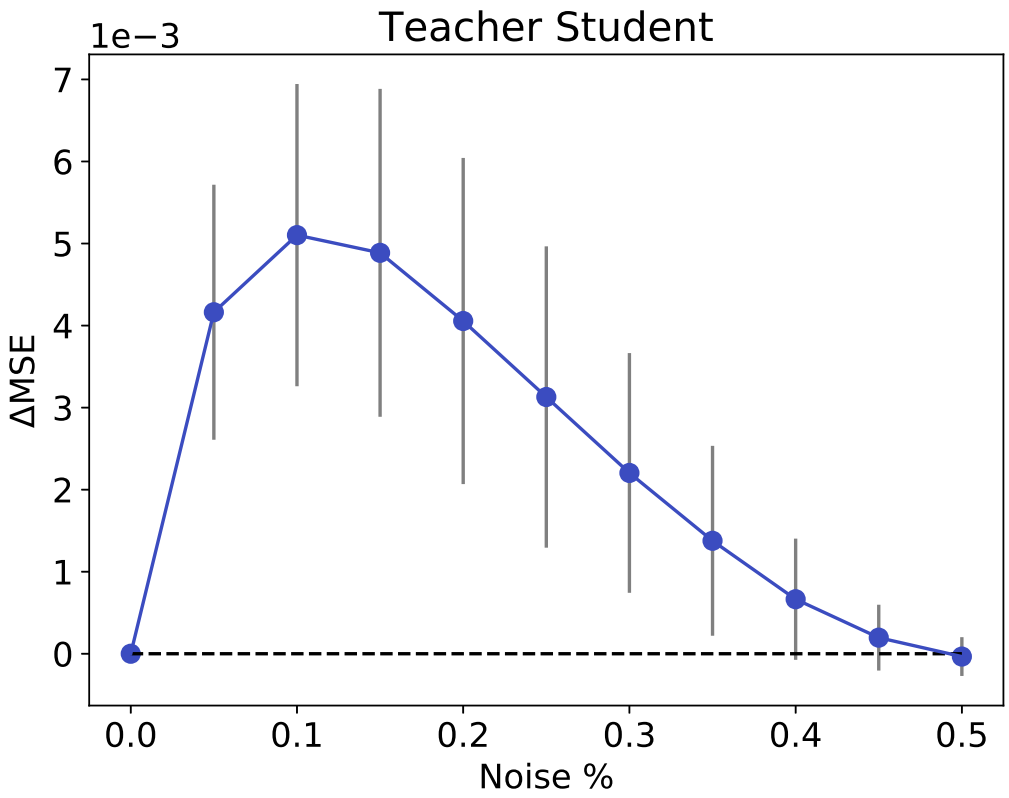


FIGURE 1.15: $\Delta\text{MSE} = \text{MSE}(classical) - \text{MSE}(quantum)$ versus the percentage of labels flipped in the training data. Error bars indicate the standard deviation over the 100 different runs. If the amount of noise is 0%, the classical and quantum perceptron will converge to the same solution. If the amount of noise is 50% then both models cannot learn anything. Between these two points lies an area where the quantum perceptron has a lower MSE.

1.9 Extensions

While the results for simple data sets are promising, the quantum perceptron appears to have some limitations. We will discuss and resolve these problems in the next few sections by extending the initial model. A quantitative analysis will be left out for these extensions, but we will consider some toy problems to investigate the qualitative properties.

1.9.1 Continuous Data

The definition of $b(\mathbf{x})$ in equation 1.28 is problematic for continuous data, since $b(\mathbf{x})$ will always be ± 1 due to no overlapping samples occurring. This means that we always end up with the limiting case $m^x \to 0$, logistic regression. To still be able to use m^x we define a different $b(\mathbf{x})$ statistic for continuous data, that is dependent on the distance between samples.

To start of, we rewrite the first term in the gradient rule for $\mathbf{w}^z$ as

$$\begin{aligned}
&\sum_{\mathbf{x}} q(\mathbf{x}) b(\mathbf{x}) \mathbf{x} = \sum_{\mathbf{x}} q(2q(y=1|\mathbf{x}) - 1)\mathbf{x}, \\
\text{Bayes' Rule } &\rightarrow \sum_{\mathbf{x}} \frac{q(y=1|\mathbf{x})q(\mathbf{x})}{q(y=1)}\mathbf{x} = \sum_{\mathbf{x}} q(\mathbf{x}|y=1)\mathbf{x} \\
&= \left(\sum_{\mathbf{x}} 2q(\mathbf{x}|y=1)q(y=1)\mathbf{x} \right) - \mathbb{E}[\mathbf{x}]_q \\
&= 2q(y=1)\mathbb{E}[\mathbf{x}|y=1]_q - \mathbb{E}[\mathbf{x}]_q.
\end{aligned}$$

For the $\mathbf{w}^x$ gradient the we have to deal with the square root:

$$\begin{aligned}
&= \sum_{\mathbf{x}} q(\mathbf{x})\mathbf{x} \sum_{y} \sqrt{1 - b(\mathbf{x})^2} \\
&= \sum_{\mathbf{x}} q(\mathbf{x})\mathbf{x} \sum_{y} \frac{1}{2} \sqrt{q(y=1|\mathbf{x})q(-y=1|\mathbf{x})}.
\end{aligned}$$

By again using Bayes' rule and noticing the symmetry $y \Leftrightarrow -y$ we get

$$\begin{aligned}
&= \sum_{\mathbf{x}} q(\mathbf{x}) \sqrt{1 - b(\mathbf{x})^2}\mathbf{x}, \\
&= \sum_{\mathbf{x}} q(\mathbf{x}) 2 \sqrt{q(y=1|\mathbf{x})q(y=-1|\mathbf{x})}\mathbf{x} \\
\text{Bayes Rule } &\rightarrow \sum_{\mathbf{x}} 2q(\mathbf{x}) \sqrt{\frac{q(y=1)q(\mathbf{x}|y=1)}{q(\mathbf{x})}} \times \sqrt{\frac{q(y=-1)q(\mathbf{x}|y=-1)}{q(\mathbf{x})}}\mathbf{x} \\
&= \sqrt{q(y=1)(1-q(y=1))} \times \sum_{\mathbf{x}} 2\sqrt{q(\mathbf{x}|y=1)q(\mathbf{x}|y=-1)}\mathbf{x} \\
&= 2\sqrt{q(y=1)(1-q(y=1))}\mathbb{E}[\mathbf{x}]_Q, \qquad (1.28)
\end{aligned}$$

with $Q(\mathbf{x}) = \sqrt{q(\mathbf{x}|y=1)q(\mathbf{x}|y=-1)}$. The new update rules are then given by

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathbf{w}^x} &= 2\sqrt{q(y=1)(1-q(y=1))}\mathbb{E}[\mathbf{x}]_Q - \sum_{\mathbf{x}} q(\mathbf{x}) \tanh h \frac{h^x}{h}\mathbf{x}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{w}^y} &= -\sum_{\mathbf{x}} q(\mathbf{x}) \tanh h \frac{h^y}{h}\mathbf{x}, \\
\frac{\partial \mathcal{L}}{\partial \mathbf{w}^z} &= 2q(y=1)\mathbb{E}[\mathbf{x}|y=1]_q - \mathbb{E}[\mathbf{x}]_q - \sum_{\mathbf{x}} q(\mathbf{x}) \tanh h \frac{h^z}{h}\mathbf{x}.
\end{aligned}$$

Both gradients are now dependent on the expectation value of $\mathbf{x}$ given y, instead of the other way around. This allows us to introduce a kernel for $q(\mathbf{x}|y)$ in the input space that assigns a probability to $\mathbf{x}$ based on nearby samples with the same labels. Assume that we have data $\mathbf{x} \in \mathbb{R}^d$ and $y \in \{1, -1\}$. We place a multivariate Gaussian of fixed variance over each data point and then we calculate the conditional

probability for a sample $\mathbf{x}$ of length d,

$$q(\mathbf{x}|y) = \frac{1}{Z}\sum_{\mathbf{x}'}\mathbb{I}(y'=y)f(\mathbf{x},\mathbf{x}';\Sigma)$$
$$= \frac{1}{Z}\sum_{\mathbf{x}'}\frac{\mathbb{I}(y'=y)}{\sqrt{(2\pi)^d|\Sigma|}}\exp\left\{-\frac{(\mathbf{x}'-\mathbf{x})^T\Sigma^{-1}(\mathbf{x}'-\mathbf{x})}{2}\right\},$$

where Z ensures that $\sum_{\mathbf{x}} q(\mathbf{x}|y) = 1$. If we set all covariances zero and the variances to σ^2, we get $\Sigma = \sigma^2\mathbb{1}$ which simplifies the above expression to

$$= \frac{1}{Z}\sum_{\mathbf{x}'}\frac{\mathbb{I}(y'=y)}{\sqrt{(2\pi\sigma^2)^d}}\exp\left\{-\frac{(\mathbf{x}'-\mathbf{x})^T(\mathbf{x}'-\mathbf{x})}{2\sigma^2}\right\}.$$

Because we place a Gaussian over each point $\mathbf{x}$, we have to correct the distribution $q(\mathbf{x})$ accordingly,

$$q(\mathbf{x}) = \sum_y q(y)q(\mathbf{x}|y) = \sum_y q(y)\frac{1}{Z}\sum_{\mathbf{x}'}\mathbb{I}(y'=y)f(\mathbf{x};\mathbf{x}',\sigma^2\mathbb{1}).$$

The resulting probability landscapes can be seen in figure 1.16 If the distributions

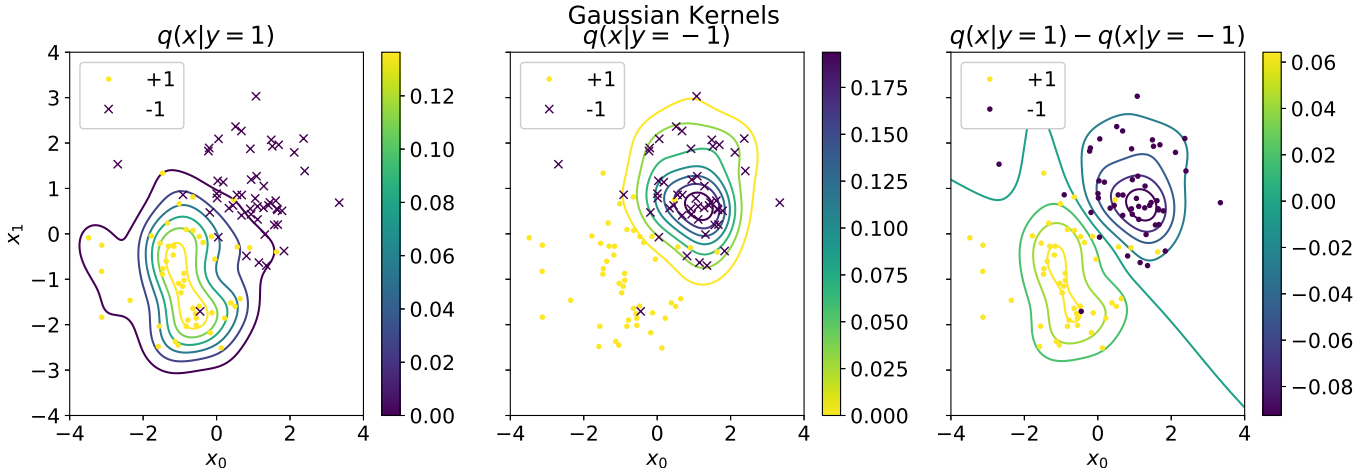


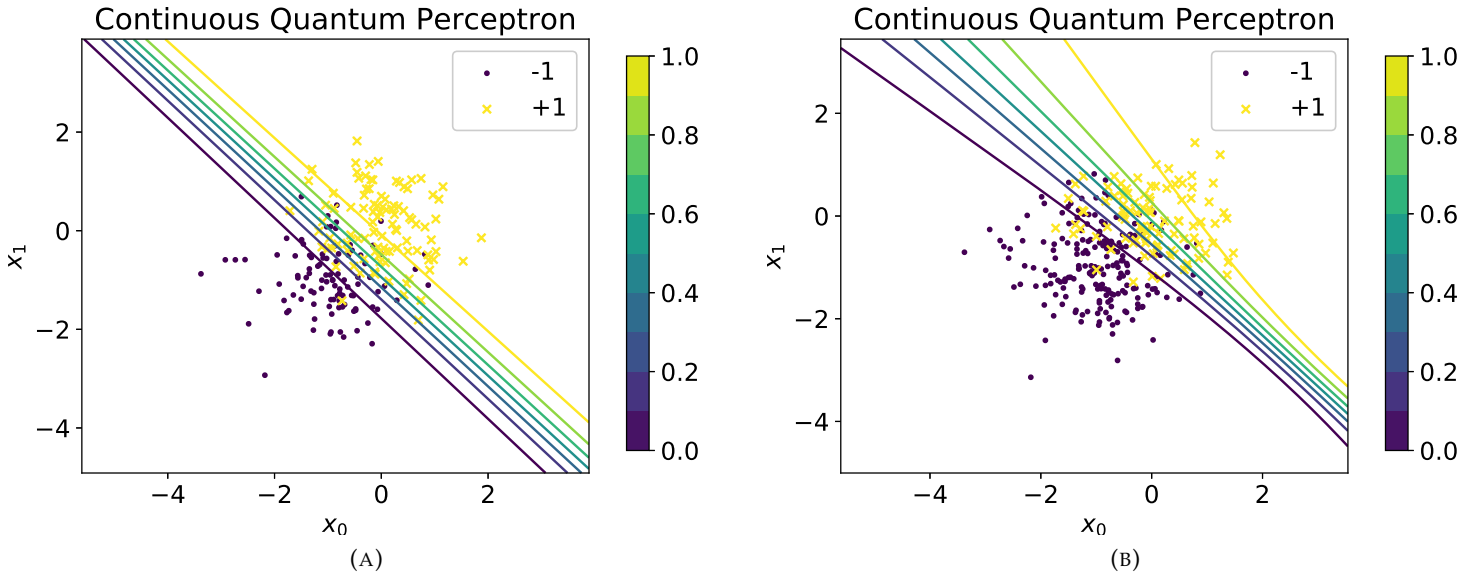
FIGURE 1.16: Gaussian landscapes for $\sigma'^2 = 0.25$ for two random data sets. In the left panel we see the probability of finding a sample with label $y = 1$ in the input space. In the middle we have the same figure but now for $y - 1$. The expectation value for finding the sample with a certain label is plotted in the right panel.

over the samples overlap, then $\mathbb{E}[\mathbf{x}]_Q \neq 0$ and as a result $\mathbf{w}^x$ becomes non-zero. We thus measure the impact of neighbouring samples on the likelihood of observing a sample within a certain class. If the samples are far apart, then we require broad distributions to create overlap between samples.

To test this model we create N samples of length $d = 2$ where $X_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \sigma^2\mathbb{1})$ with label $y = 1$ and N samples $X_2 \sim \mathcal{N}(\boldsymbol{\mu}_2, \sigma^2\mathbb{1})$ with label $y = -1$. $\sigma^2 = 0.5$. $\boldsymbol{\mu}_1 = (-1,-1)$ and $\boldsymbol{\mu}_2 = (0,0)$. We generate 100 samples for each label and set $\epsilon = 0.01$. We set the variance of the sample Gaussians in $q(\mathbf{x}|y)$ to $\sigma'^2 = 0.05$. The shape of the separation boundary is very sensitive to asymmetries in the distributions. If the data for both labels is distributed equally, and there are an equal

number of samples with label $y = 1$ as there are with $y = -1$, then we get a straight boundary through the data, since the distance based kernel cancels everywhere. If $q(\mathbf{x}|y) \in \{0, 1\}$, so only sharp peaks around the samples, then $\mathbf{w}^x \to 0$ and we get a thin straight line as boundary, just as in the classical case. However, if there are asymmetries in the data we observe the wavering effect of section 1.8, as can be seen in figure 1.17.

The cost function is changed as a result of the continuous kernel and this affects the placement of the boundaries. However, the probability boundaries retain the same shapes as in section 1.7.1, since the only thing affected is the data density matrix. The model $\rho_{\mathbf{x}}$ still has the same degrees of freedom as before and thus retains the same properties as before. The introduction of the Gaussian kernels introduces an extra parameter, namely the variance σ'^2. This introduces a bias in the algorithm. We either need to set it before training or adapt it during learning for optimal performance. This was not further explored here, since we only wanted to show the qualitative properties of the continuous model. In the end, the quantum perceptron retains its functionality in dealing with better noisy data, at the cost of introducing an additional free parameter.



(A) (B)

FIGURE 1.17: Quantum perceptron for continuous data. On the left side the number of samples between the two classes is equal ($n = 100$). The kernels functions average out and the separation boundary is symmetric with respect to the two distributions. If we double the number of samples for the -1 class the asymmetry in the data causes the star like separation boundaries to appear.

1.9.2 Multiple Classes

The classical perceptron is inherently a binary classifier, but can be extended to allow for multiclass regression. The model probability distribution for a multinomial logistic regression or maximum entropy classifier is given by [33],

$$p(y = c|\mathbf{x}, (\mathbf{w}_1, \ldots, \mathbf{w}_C)) = p(y = c|\mathbf{x}, \mathbf{W}) = \frac{\exp(\mathbf{w}_c \cdot \mathbf{x})}{\sum_{c'}^{C} \exp(\mathbf{w}_{c'} \cdot \mathbf{x})} = \mathcal{S}_c^{\mu}. \tag{1.29}$$

For C classes $c \in \{0, 1, \ldots, C-1\}$ where we adopt Murphy's capital letter notation for matrices, where $\mathbf{W}$ is a matrix with columns equal to $\mathbf{w}_i$. The output y is one-hot encoded by a vector $\mathbf{y}^{\mu}$ where the c-th bit is turned on if and only if $y = c$. By using Lagrange Multipliers we can show that this distribution maximizes the classical entropy (See appendix A.6) [57]. Plugging equation 1.29 into the classical likelihood gives

$$\begin{aligned} f(\mathbf{W}) &= -\sum_{\mathbf{x}}^{N} \sum_{c}^{C-1} y_c \log p(y = c|\mathbf{x}, \mathbf{W}) \\ &= -\sum_{\mathbf{x}}^{N} \left(\left(\sum_{c}^{C-1} y_c \mathbf{w}_c \cdot \mathbf{x} \right) - \log \left(\sum_{c'}^{C-1} \exp(\mathbf{w}_{c'} \cdot \mathbf{x}) \right) \right). \end{aligned}$$

Clearly, a gradient with respect to a set of weights $\mathbf{w}_c$ gives

$$\nabla_{\mathbf{w}_c} f(\mathbf{W}) = \sum_{\mathbf{x}} (\mathcal{S}_c^{\mu} - y_c)\mathbf{x}.$$

If we look at the softmax function we see that it is actually fully determined by $C-1$ sets of weights, due to the fact the probability must sum to 1. By shifting all weights with a constant vector $\mathbf{d}$ we get

$$\frac{\exp((\mathbf{w}_c - \mathbf{d}) \cdot \mathbf{x})}{\sum_{c'}^{C-1} \exp((\mathbf{w}_{c'} - \mathbf{d}) \cdot \mathbf{x})} = \frac{\exp(-\mathbf{d} \cdot \mathbf{x}) \exp(\mathbf{w}_c \cdot \mathbf{x})}{\exp(-\mathbf{d} \cdot \mathbf{x}) \sum_{c'}^{C-1} \exp(\mathbf{w}_{c'} \cdot \mathbf{x})} = \frac{\exp(\mathbf{w}_c \cdot \mathbf{x})}{\sum_{c'}^{C-1} \exp(\mathbf{w}_{c'} \cdot \mathbf{x})}.$$

If we set $\mathbf{d} = \mathbf{w}_C$, we eliminate one set of parameters. If $C = 2$ we retrieve the original description of logistic regression,

$$\mathcal{S}^{\mu} = \frac{\exp(\mathbf{w} \cdot \mathbf{x})}{1 + \exp(\mathbf{w} \cdot \mathbf{x})} = \frac{1}{1 + \exp(-\mathbf{w} \cdot \mathbf{x})}.$$

In order to extend this multinomial regression to the quantum likelihood description, we need to find a density matrix model that is analogous to the maximum entropy ansatz for a classical system. As with the quantum perceptron, we use the quantum mechanical canonical ensemble ansatz for the density matrix,

$$\rho_{\mathbf{x}} = \frac{\exp(H)}{\mathrm{Tr}\{\exp(H)\}}, \tag{1.30}$$

which reduces to the Boltzmann distribution if H contains no off-diagonal elements and diagonal elements corresponding to the Boltzmann probabilities. Up to this point, we considered models where the output state was a qubit, so $\dim(H) = 2$ and a general basis for H was given by the Pauli basis $\sum_k h^k \sigma^k$. From group theory we know that this is the fundamental representation of the SU(2) Lie Algebra, and a basis for 2×2 traceless Hermitian matrices. For multinomial regression we can

look at a description of the data in terms of the representation of the Lie algebra of $SU(N)$, where $N = C$ is the number of different classes in the data. Let us start with $C = 3$ classes, which describes a 3 level system called a qutrit [58]. We have to note that this is not a description of a boson, because bosons are described by the adjoint representation **3** of $SU(2)$, which does not span the full space of density matrices. The group $SU(3)$ is used in quantum chromodynamics to describe the symmetry of quark colors [1].

The fundamental representation of $SU(3)$ is given by the so called Gell-Mann matrices, a set of 8 linearly independent 3×3 matrices that span the full space of traceless Hermitian 3×3 matrices. They are given by

$$\lambda^1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \lambda^2 = \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \lambda^3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$
$$\lambda^4 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \lambda^5 = \begin{pmatrix} 0 & 0 & -i \\ 0 & 0 & 0 \\ i & 0 & 0 \end{pmatrix},$$
$$\lambda^6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \lambda^7 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix}, \quad \lambda^8 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}.$$

Using these matrix we can write a general three state Hamiltonian H as

$$H = \sum_k h^k \lambda^k$$

with $h^k = \mathbf{x} \cdot \mathbf{w}^k$ as before. Remember that for the Pauli matrices we had the following convenient expression:

$$\exp\{ia(\hat{n} \cdot \boldsymbol{\sigma})\} = \mathbb{1} \cos a + i(\hat{n} \cdot \boldsymbol{\sigma}) \sin a,$$

with $|\hat{n}| = 1$ and $a \in \mathbb{R}$. By proving this expression we will discover a problem related to the $SU(3)$ (and more generally $SU(N)$) descriptions of the density matrix model. The proof relies on the fact that $(\hat{n} \cdot \boldsymbol{\sigma})^{2p} = \mathbb{1}$, for $p = 1$ we have

$$\begin{aligned}
(\hat{n} \cdot \boldsymbol{\sigma})^2 &= (n^x \sigma^x + n^y \sigma^y + n^z \sigma^z)^2 \\
&\quad + n^y n^z \sigma^y \sigma^z + n^z n^x \sigma^z \sigma^x + n^z n^y \sigma^z \sigma^y + n^z n^z \sigma^z \sigma^z \\
&= (n^x)^2 (\sigma^x)^2 + (n^y)^2 (\sigma^y)^2 + (n^z)^2 (\sigma^z)^2 \\
&\quad + n^x n^y (\sigma^x \sigma^y + \sigma^y \sigma^x) + n^x n^z (\sigma^x \sigma^z + \sigma^z \sigma^x) + n^y n^z (\sigma^y \sigma^z + \sigma^z \sigma^y) \\
&= \mathbb{1} \left((n^x)^2 + (n^y)^2 + (n^z)^2 \right) + n^x n^y \{\sigma^x, \sigma^y\} + n^x n^z \{\sigma^x, \sigma^z\} + n^y n^z \{\sigma^y, \sigma^z\} \\
&= \mathbb{1},
\end{aligned}$$

[1] Maybe one day we will have quantum chromo computers, capable of manipulating quark colors.

where we used that $\{\sigma_i, \sigma_i\} = 2I$ and $\{\sigma_i, \sigma_j\} = 0$ for $i \neq j$. If we write out the exponential operator we get

$$\begin{aligned}
\exp\{ia(\hat{n}\cdot\boldsymbol{\sigma})\} &= \sum_{k=0}^{\infty} \frac{i^k((a\hat{n}\cdot\boldsymbol{\sigma})^k)}{k!} = \sum_{p=0}^{\infty} \frac{(-1)^p(a\hat{n}\cdot\boldsymbol{\sigma})^{2p})}{2p!} + i\sum_{q=0}^{\infty} \frac{(-1)^q(a\hat{n}\cdot\boldsymbol{\sigma})^{2q+1})}{(2q+1)!} \\
&= \sum_{p=0}^{\infty} \frac{(-1)^p(a\hat{n}\cdot\boldsymbol{\sigma})^{2p}}{2p!} + i(\hat{n}\cdot\boldsymbol{\sigma})\sum_{q=0}^{\infty} \frac{(-1)^q a^{2q+1}(\hat{n}\cdot\boldsymbol{\sigma})^{2q}}{(2q+1)!} \\
&= \mathbb{1}\sum_{p=0}^{\infty} \frac{(-1)^p a^{2p}}{2p!} + i(\hat{n}\cdot\boldsymbol{\sigma})\sum_{q=0}^{\infty} \frac{(-1)^q a^{2q+1})}{(2q+1)!} \\
&= \mathbb{1}\cos a + i(\hat{n}\cdot\boldsymbol{\sigma})\sin a. \qquad (1.31)
\end{aligned}$$

For $\exp\{a(\hat{n}\cdot\boldsymbol{\sigma})\}$ we replace the sine and cosine with their respective hyperbolic functions. For the quantum perceptron this relation meant that we could easily find the derivatives of the likelihood function, but this seems to be a property of SU(2) alone. The commutation relations of SU(3) are

$$\begin{aligned}
\left(\frac{\lambda^a}{2}, \frac{\lambda^b}{2}\right) &= i\sum_{c=1}^{8} f^{abc}\frac{\lambda^c}{2}, \\
\left\{\frac{\lambda^a}{2}, \frac{\lambda^b}{2}\right\} &\neq 0,
\end{aligned}$$

with f^{abc} the structure constant of SU(3). This means that $(\hat{n}\cdot\boldsymbol{\lambda})^{2p} \neq 1$ and that there is no easy way to represent the exponent in a similar fashion as in equation 1.31. Some work has been done to come up with such a formula nonetheless, but it is not as elegant as for the 2×2 case, and hard to generalize to SU(N) [59]. This means that we will have to calculate the gradient numerically. The quantum likelihood is given by

$$g(\mathbf{W}) = \sum_{\mathbf{x}} q(\mathbf{x})\,\mathrm{Tr}\{\eta_{\mathbf{x}}\log\rho_{\mathbf{x}}\},$$

and the model is given by equation 1.30. In order to calculate the log of the trace we have to determine the eigenvalues of H:

$$\log\left(\frac{\exp(H)}{\mathrm{Tr}\{\exp(H)\}}\right) = \log(\exp(H)) - \log(\mathrm{Tr}\{\exp(H)\}) = H - \log\left(\sum_i \exp(\lambda_i)\right).$$

So numerically determining the eigenvalues of H will give us a value for quantum log-likelihood. Coming up with an analytic expression for the eigenvalues will be quite cumbersome, so we will have to resort to a numerical method to determine them. The data density matrix can be constructed similarly as for the qubit by writing the data as a wave function,

$$\begin{aligned}
|\psi\rangle &= \sqrt{q(y=0|\mathbf{x})}\,|0\rangle + \sqrt{q(y=1|\mathbf{x})}\,|1\rangle + \sqrt{q(y=2|\mathbf{x})}\,|2\rangle \\
\eta_{\mathbf{x}} &= |\psi\rangle\langle\psi|\,.
\end{aligned}$$

We determine the gradient with respect to each weight w_i numerically by calculating

$$\nabla_{\mathbf{w}_i} g(\mathbf{W}) = \frac{g(\mathbf{w}_i + \epsilon|\mathbf{w}_j) - g(\mathbf{w}_1, \ldots, \mathbf{w}_8)}{\epsilon},$$

for $j \in \{1, \ldots, 8\}$ and $j \neq i$. Update the weights as

$$\mathbf{w}_i^{t+1} = \mathbf{w}_i^t - \epsilon \nabla_{\mathbf{w}_i} g(\mathbf{W}^t).$$

Again, we are faced with the question of how to determine a probability. For the binary problem, the expectation value $\langle \sigma^z \rangle = m^z$ gave us a value between -1 and 1, allowing for an easy mapping to a probability $p(y|\mathbf{x}; \mathbf{w}) \in [0, 1]$. Measuring one of the λ^k operators would give a scalar field that we would have to map to a probability, but there is no clear way of how to do this. A method that also generalizes to C class perceptrons is taking the diagonal elements of $\rho_{\mathbf{x}}$. If the model density matrix contains no off-diagonal elements, these would correspond to the class probabilities of the classical multinomial logistic regression. Taking m^z for constructing a probability for the qubit quantum perceptron is equivalent to taking the diagonal elements as probabilities. With this in mind, we choose the diagonal elements of $\rho_{\mathbf{x}}$ as the class probabilities,

$$p(y = c|\mathbf{x}; \mathbf{w}) = \rho_{c,c}.$$

As before, we observe that the off-diagonal elements go to zero if there is no noise in the data, since $\eta_{\mathbf{x}}$ will contain no off diagonal elements as well. From figure 1.18 we observe this behaviour and retrieve the spread out probability boundaries for a 3 class problem.

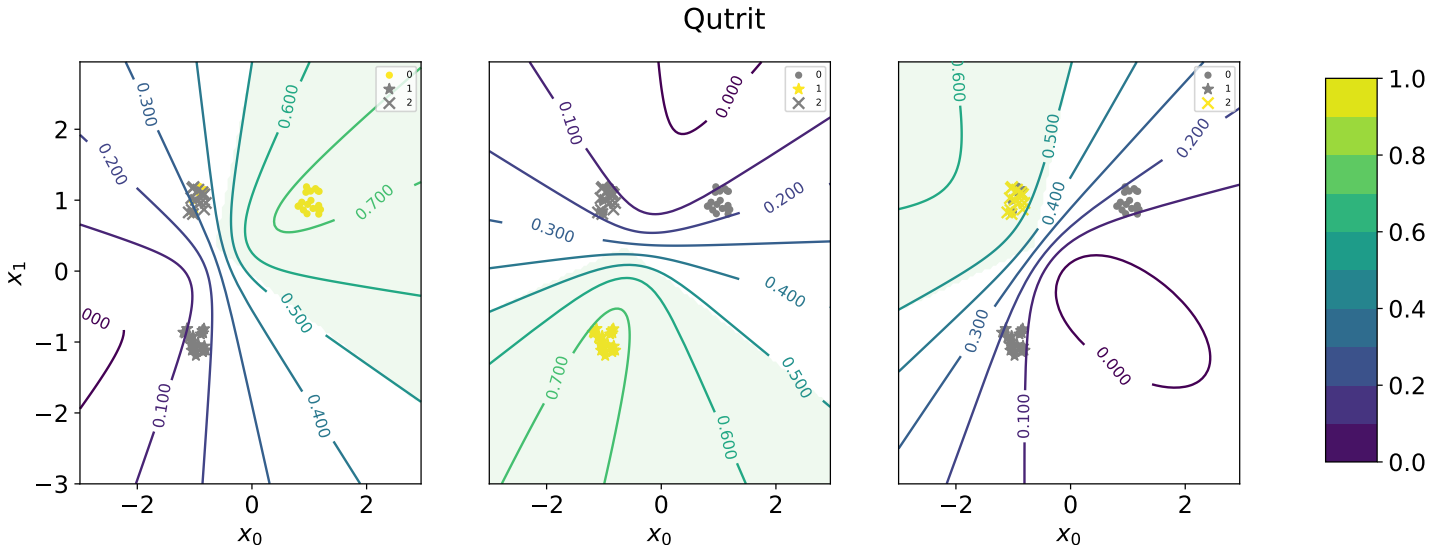


FIGURE 1.18: Separation boundaries of qutrit perceptron for noisy data. The contours show the probabilities $p(y = c|\mathbf{x}; \mathbf{w})$. The yellow icon in the legend indicates the class c. The light green background indicates the area in the input space where $p(y = c|\mathbf{x}; \mathbf{w})$, so where we would predict class c . For 30% of the outputs of the samples $\mathbf{x} = (-1, 1)$ we assign label 0. The wavering effect of the probability boundaries that we saw for the binary case returns.

The idea of qutrit regression is easily extended to a qudits perceptron. A qudit is the generalization of a two-level system to an d-level system. For SU(C), the number of matrices that are required to span the full space is equal to $C^2 - 1$, which can be constructed with the following rules [60]:

a) Symmetric matrices

$$\Gamma^{jk}_{sym} = |j\rangle\langle k| + |k\rangle\langle j|\,, \quad 1 \leq j < k \leq C$$

b) Antisymmetric matrices

$$\Gamma^{jk}_{asym} = -i\,|j\rangle\langle k| + i\,|k\rangle\langle j|\,, \quad 1 \leq j < k \leq C$$

c) Diagonal matrices

$$\Gamma^{l}_{diag} = \sqrt{\frac{2}{l(l+1)}}\left(\sum_{j=1}^{l} |j\rangle\langle j| - l\,|l+1\rangle\langle l+1|\right), \quad 1 \leq j \leq C-1$$

With these simple rules we can extend out quantum perceptron model to any number of classes. An example solution for $C = 6$ is shown in figure 1.19. The quantum perceptron can be extended to learn multiclass classification problems, but this comes at the cost of a significant performance decrease. Although learning $C^2 - 1$ vector weights in not a problem, we have to find the eigenvalues of each $C \times C$ density matrix for each sample $\mathbf{x}$ at each training step in order to calculate $\log \sum_i \exp(\lambda_i)$. Consider the MNIST data set, a collection of 60000 handwritten digits from $0 - 9$ of size 28×28. In order for the perceptron to learn this data we would have to diagonalize 60000 10×10 matrices at each training step and update $(10^2 - 1) = 99$ vectors. Compare this to the classical perceptron, which only has to calculate $\sum_{\mathbf{x}}(\mathcal{S}^{\mu}_{c} - y_c)\mathbf{x}$ for 9 vectors weights and update them. However, the extra utility of the quantum perceptron might outweigh the increase in computational cost.

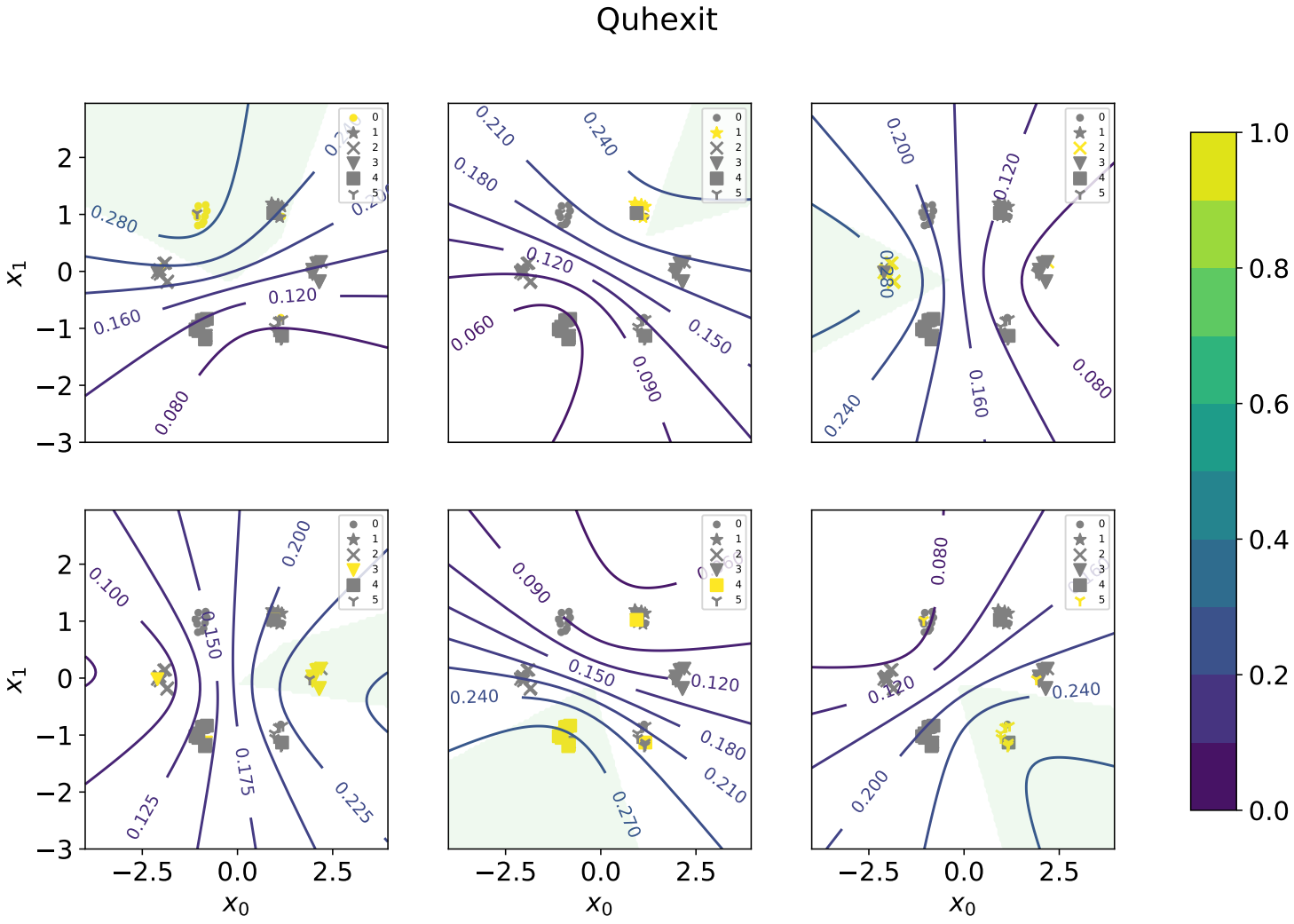


FIGURE 1.19: Separation boundaries of qudit perceptron for noisy data arranged in a hexagon. The contours show the probabilities $p(y = c|\mathbf{x}; \mathbf{w})$. The yellow icon in the legend indicates the class c. The light green background indicates the area in the input space where $p(y = c|\mathbf{x}; \mathbf{w})$, so where we would predict class c. Additioally, 30% of the labels are randomly exchanged between the samples. The flipped labels perturb the boundaries as expected.

1.9.3 Classical Limit

Some of the work in this thesis was presented and discussed at the "Bits and brains" meeting, an interdisciplinary seminar with different groups working on materials science and their possible applications to neuromorphic computing. With regards to the quantum perceptron, the following point was put forward: Since we are not using non-classical correlations or entanglement, we cannot talk experimentally of a quantum system. We are using a finite temperature description, and measuring this ensemble means the quantum properties of the system such as the superposition of states, might not be distinguishable from a non-quantum system due to thermal effects. We are thus not using quantum mechanics in any way and the qualitative properties can be achieved by considering a classical system. More specifically, simply measuring m^x and m^z is not enough to call it a quantum perceptron. It was suggested that a similar model might be obtained from a classical magnetic material.

To analyze this argument we will attempt to replace m^k with a classical magnetization function. The perfect candidate for this is the Langevin function, which links the applied magnetic field to the total magnetization of a paramagnetic material. See figure 1.20. A paramagnetic material has permanent magnetic dipoles, even in

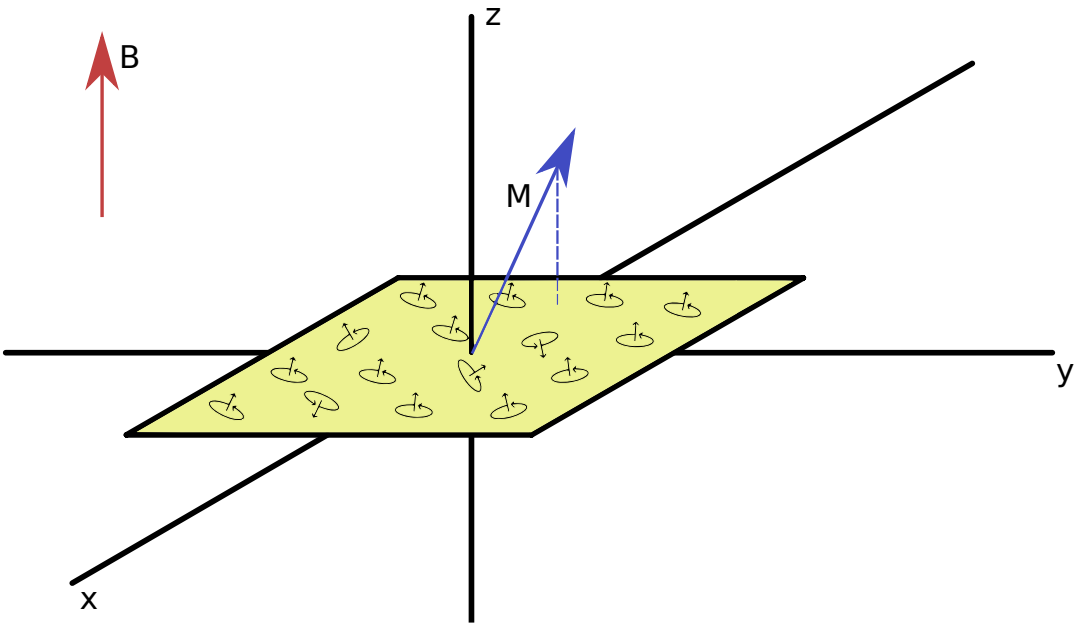


FIGURE 1.20: Magnetization in a paramagnetic material. Individual magnetic dipoles all contribute to the total magnetization in the material.

the absence of an external magnetic field. In a uniform field these dipoles have a potential energy of

$$E = -\mathbf{m} \cdot \mathbf{B}.$$

If we consider the coordinate system where $\mathbf{m}$ lies in the xz-plane, we get $U = -mB\cos\theta$ with θ the angle between the two vectors in the xz-plane. The change in energy is given by $dE = mB\sin\theta d\theta$. If we assume that quantum effects are negligible

then the number of states N can be written with the Boltzmann distribution,

$$N = \int n(E)dE,$$
$$\frac{dN}{dE} = n(E) = \frac{1}{Z}\exp(-\beta E).$$

Since this can be rewritten in terms of θ, this is also the number of dipoles with a orientation between θ and $\theta + d\theta$.

The total magnetization can be written as an integral over all N dipoles projected onto the direction of the applied magnetic field,

$$M = \int_0^N m\cos\theta dn = N\langle m\rangle,$$
$$\frac{\langle m\rangle}{m} = \frac{1}{N}\int_0^N \cos\theta dn,$$
$$N \equiv \int_0^N dn,$$

where $\langle m\rangle$ is the average magnetization in the direction of $\mathbf{B}$. Substituting

$$dn = \frac{1}{Z}mB\sin\theta\exp[\beta mB\cos\theta]d\theta,$$
$$x = \beta mB\cos\theta = a\cos\theta,$$
$$d\theta = -1/a\sin\theta dx,$$

gives

$$\begin{aligned}\frac{\langle m\rangle}{m} &= \frac{\frac{1}{a}\int_{-a}^{a} xe^x dx}{\int_{-a}^{a} e^x dx}\\ &= \frac{e^a + e^{-a}}{e^a - e^{-a}} - \frac{1}{a} = \frac{1}{\tanh a} - \frac{1}{a}\\ &= L(a) = L(m).\end{aligned}$$

So $M = NmL(a)$ [61]. This is the length of the magnetization vector, $\mathbf{M}$, which points in the same direction as the magnetic moment $\mathbf{m}$ so $\mathbf{M} = NL(a)m\hat{\mathbf{m}} = NL(a)\mathbf{m}$, since $\mathbf{m} = \mathbf{M}V$ if the magnetization is constant across the magnet. Parametrizing $\mathbf{m} = \frac{\mathbf{h}}{h}$ with $h^k = \mathbf{w}^k \cdot \mathbf{x}$ and absorbing the constants in the weights gives a relation of the form

$$M^k = L(h)\frac{h^k}{h},$$

which is similar to the parametrization in equation 1.12, except the tanh is replaced by the Langevin function. Since $M^z \in [-1, 1]$, we can define a probability $p(y|\mathbf{x};\mathbf{w}) = \frac{1}{2}(1 + yM^z)$ that assigns a proper probability to a class y. For a physical realization we have to bring a material into a certain state and measure M^x and M^z. If $M^z > 0$, so the magnetization vector points in the positive z direction, we assign class 1. If $M^z < 0$, we assign class 0. Depending on the norm, which is still $\sqrt{(h^x)^2 + (h^z)^2}$, we introduce an uncertainty about this classification. Unlike with the quantum perceptron, the only purpose of M^x is an additional degree of freedom to control the norm. M^x is not inherently coupled to flipped labels. Another more obvious difference is that we do not have a hyperbolic tangent but a Langevin function to link

the fields with the magnetization. Both functions have a codomain of $(-1, 1)$ and have a similar shape, as can be seen in figure 1.21. Also, we cannot just plug in a

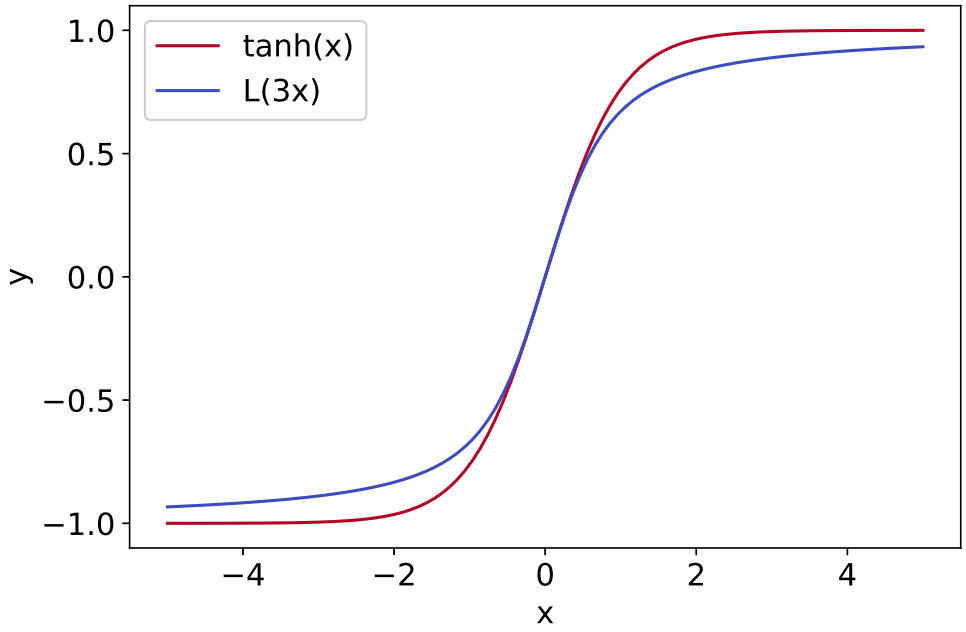


FIGURE 1.21: Difference between Langevin function and tanh. Multiplying the argument with 3 gives a function similar to $\tanh(x)$.

different nonlinear function in equation 1.16. These gradients are derived from an optimization criterion, the negative quantum log-likelihood. We have to rederive the gradient update rules from a new optimization criterion that incorporates this redefinition of the fields. To this end, define the likelihood

$$\mathcal{L} = \sum_{\mu} q(\mathbf{x}) \sum_{y} q(y|\mathbf{x}) \log[\frac{1}{2}(1 + yM^z)].$$

Deriving this likelihood with respect to the weights $\mathbf{w}^k$, with $h^k = \mathbf{w}^k \cdot \mathbf{x}$ gives

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}^k} = \sum_{\mu} q(\mathbf{x}) \sum_{y} q(y|\mathbf{x}) \frac{\partial}{\partial \mathbf{w}^k} \log[\frac{1}{2}(1 + yM^z)].$$

The derivative of the log then becomes

$$\frac{\partial}{\partial \mathbf{w}^k} \log[\frac{1}{2}(1 + yM^z)] = \frac{\frac{1}{2} y \frac{\partial M^z}{\partial \mathbf{w}^k}}{[\frac{1}{2}(1 + yM^z)]}.$$

To derive $\partial M^z / \partial \mathbf{w}^k$ we use the chain rule:

$$\frac{\partial M^z}{\partial \mathbf{w}^k} = \frac{\partial}{\partial \mathbf{w}^k} L(h) \frac{h^z}{h} = \left(\frac{\partial}{\partial \mathbf{w}^k} L(h) \right) \frac{h^z}{h} + L(h) \left(\frac{\partial}{\partial \mathbf{w}^k} \frac{h^z}{h} \right),$$

with the two derivative terms,

$$\frac{\partial}{\partial \mathbf{w}^k} L(h) = \left(\frac{1}{h^2} - \frac{1}{\sinh^2 h} \right) \frac{\partial}{\partial \mathbf{w}^k} h = \left(\frac{1}{h^2} - \frac{1}{\sinh^2 h} \right) \frac{h^k}{h} \mathbf{x},$$
$$\frac{\partial}{\partial \mathbf{w}^k} \frac{h^z}{h} = \frac{\left(\frac{\partial}{\partial \mathbf{w}^k} h^z \right) h - h^z \left(\frac{\partial}{\partial \mathbf{w}^k} h \right)}{h^2} = \frac{\left(\delta_{kz} h - h^z \frac{h^k}{h} \right) \mathbf{x}}{h^2}.$$

Combining everything gives the final expression for the gradient of $\mathcal{L}$,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}^k} = \sum_\mu q(\mathbf{x}) \sum_y q(y|\mathbf{x}) \frac{y}{1 + y M^z} \left(\left(\frac{1}{h^2} - \frac{1}{\sinh^2 h} \right) \frac{h^k h^z}{h^2} + L(h) \left(\frac{\delta_{kz}}{h} - \frac{h^z h^k}{h^3} \right) \right) \mathbf{x}.$$

Although this expression looks complicated, we can understand the first term as moment matching between the empirical distribution and M^z. For some sample $\mathbf{x}$ we have

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{w}^k} &\propto q(y=1|\mathbf{x}) \frac{1}{1+M^z} + (1 - q(y=1|\mathbf{x}) \frac{-1}{1-M^z} \\ &= \frac{q(y=1|\mathbf{x})(1-M^z) - (1-q(y=1|\mathbf{x}))(1+M^z)}{1-(M^z)^2} \\ &= \frac{2q(y=1|\mathbf{x}) - 1 - M^z}{1-(M^z)^2} = \frac{b(\mathbf{x}) - M^z}{1-(M^z)^2}. \end{aligned}$$

In other words, when M^z equals the expected value $b(\mathbf{x})$ for each sample $\mathbf{x}$ we have reached the minimum of the cost function. The separation boundary of the magnetization perceptron is broader than that of the quantum perceptron, due to the shape of the Langevin function. This difference can be seen in figure 1.22.

Even though we get a similar model separation boundary as for the quantum perceptron, the magnetization perceptron loses some of the elegance of the quantum description. The qubit is an inherently binary system, with states that can be measured and directly correspond to the class labels. For the magnetization perceptron we have to choose the xz-plane as the threshold for classification. The σ^x operator provides a very natural way of considering flipped labels in the quantum perceptron. Also, the extension to qudits is very natural if we consider the $\mathrm{SU}(C)$ representations for C classes. For a paramagnetic material this is less obvious, since we would have to extend physical space to $1 + C$ dimensions.

1.10 Entangled Qubit Regression

As another followup of the critique that the model is not quantum we ask the question: can we utilize quantum effects besides the superposition principle? The answer turns out to be yes. With a simple extension we can build a machine learning model from an entangled state. To start, we remember that a general pure state can be written as (see Appendix A.3)

$$|\psi\rangle = \frac{1}{Z} \sum_{i,j} h_{ij} |\psi_i\rangle \otimes |\psi_j\rangle ,$$

where $h_{ij} \in \mathbb{C}$ and $\{|\psi_{i,j}\rangle\}$ form an orthonormal basis for the subspaces $\mathcal{H}_A$, $\mathcal{H}_B$ respectively. The subspaces have dimension 2 for qubits so $i, j \in \{0, 1\}$. We must

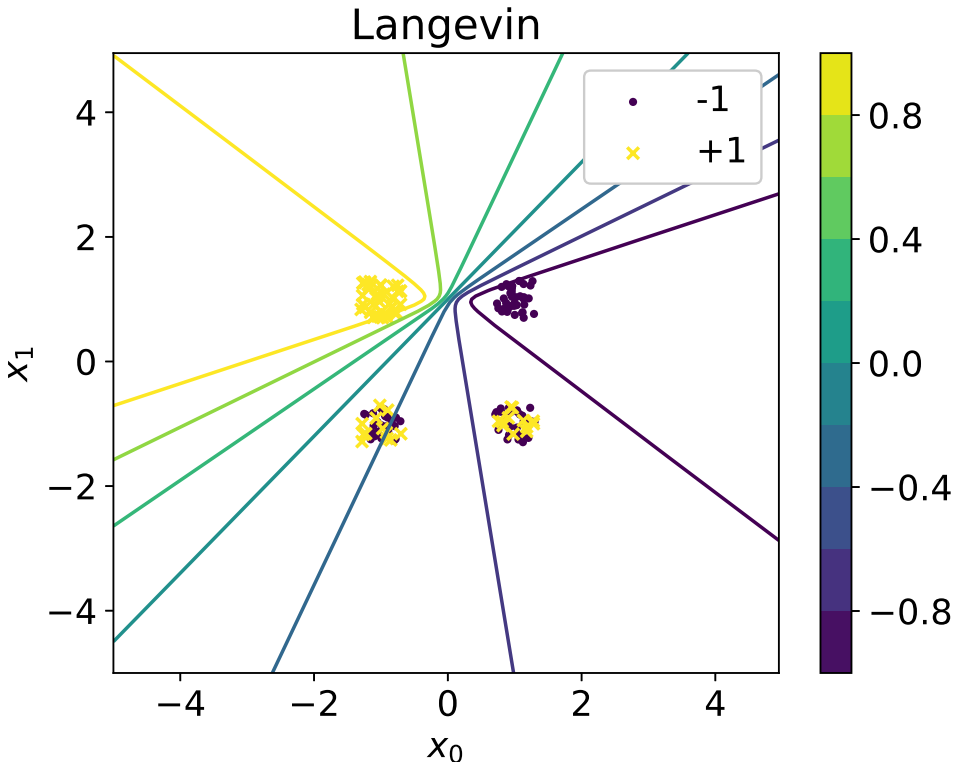


FIGURE 1.22: Separation boundary of Langevin perceptron. The assigned boundaries are broader than for the quantum perceptron. Also, the two noisy classes get different expectation values assigned to them even though they are equally noisy. In the quantum perceptron these samples are assigned an equal probability, so the uncertainty in the data is captured correctly.

normalize $|\psi\rangle$ accordingly, so $Z = \sqrt{\langle\psi|\psi\rangle} = \sqrt{\sum_{ij} h_{ij}^* h_{ij}}$. This state can be entangled (see Appendix A.6) and we can describe it with a density matrix,

$$\begin{aligned}\rho_{\mathbf{x}} &= |\psi\rangle\langle\psi| \\ &= \sum_{i,j,i',j'} h_{ij}^* h_{i'j'} \, |\psi_i\rangle \otimes |\psi_j\rangle \langle\psi_{i'}| \otimes \langle\psi_{j'}| \, .\end{aligned}$$

This matrix is of course rank one since we are dealing with a pure state. If we look at the reduced density matrix ρ_B we end up with a mixed state,

$$\begin{aligned}\rho_B &= \mathrm{Tr}_A \left\{ \sum_{i,j,i',j'} h_{ij}^* h_{i'j'} \, |\psi_i\rangle \otimes |\psi_j\rangle \langle\psi_{i'}| \otimes \langle\psi_{j'}| \right\} \\ \rho_B &= \sum_{i,j,i',j'} h_{ij}^* h_{i'j'} \underbrace{\mathrm{Tr}_A \left\{ |\psi_i\rangle \langle\psi_{i'}| \right\}}_{\delta_{i,i'}} \otimes \left(|\psi_j\rangle \langle\psi_{j'}| \right) \\ \rho_B &= \sum_{i,j,j'} h_{ij}^* h_{ij'} \, |\psi_j\rangle \langle\psi_{j'}| \, .\end{aligned}$$

If we take $h_{ij} = \mathbf{w}^{ij} \cdot \mathbf{x}$ with $\mathbf{w}^{ij} \in \mathbb{C}$, then we have an entangled state parametrized by the data. In matrix form the reduced density matrix is given by

$$\rho_B = \frac{1}{\sum_{ij}(h_{ij}^2)} \begin{pmatrix} h_{00}^2 + h_{10}^2 & h_{00}^* h_{01} + h_{10}^* h_{11} \\ h_{00}^* h_{01} + h_{10}^* h_{11} & h_{01}^2 + h_{11}^2 \end{pmatrix}.$$

As before, we represent the data as a density matrix $\eta_{\mathbf{x}} = \sqrt{q(y|x)q(y'|\mathbf{x})}$ and minimize the quantum log-likelihood in equation 1.13,

$$NLL = -\sum_{\mathbf{x}} q(\mathbf{x}) \operatorname{Tr}\{\eta_x \log \rho_B\}.$$

To update the parameters we will use the numerical gradient $f'(x) = (f(x+\epsilon) - f(x))/\epsilon$ for all weights $\mathbf{w}^{ij}$ of the density matrix. The matrix logarithm of ρ_B can be calculated by diagonalizing ρ_B,

$$\log(\rho_B) = \log\left(UDU^\dagger\right) = U \log(D) U^\dagger.$$

Here, $\log(D)$ is a matrix with the logarithm of the eigenvalues on the diagonal, since ρ_B is positive semidefinite these eigenvalues will always be greater than zero and so the logarithm will be well defined. With this model, we can learn nonlinear problems, such as the XOR problem as seen in figure in 1.23

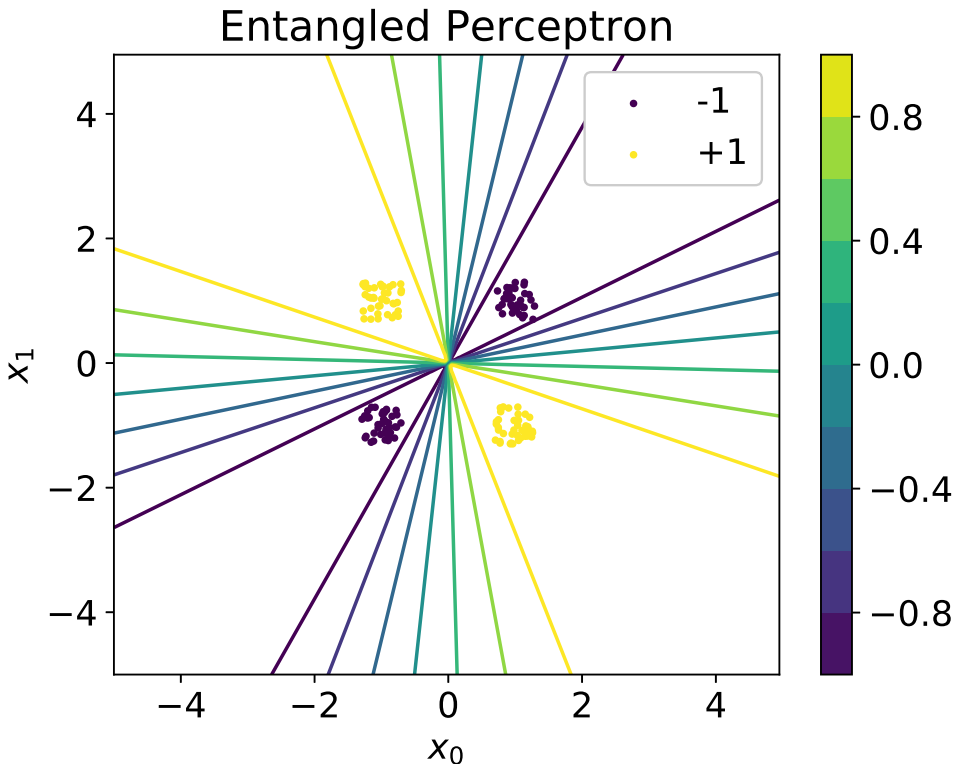


FIGURE 1.23: The XOR problem solved by two entangled qubits. The hyperbolic separation boundaries allow for the classification of a nonlinear problem.

Again, let us try to find an analytic expression for the probability boundaries of this model. The reduced density matrix is dependent on four parameters. To study the probability boundaries we study the eigenvectors of the system. First, redefine the

density matrix in terms of parameters a, b, c,

$$\rho_B = \frac{1}{a+b}\begin{pmatrix} a & c \\ c & b \end{pmatrix}, \tag{1.32}$$

which we can identify with the previously found single qubit density matrix,

$$\rho_{\mathbf{x}} = \frac{1}{2}\begin{pmatrix} 1+m^z & m^x - im^y \\ m^x + im^y & 1-m^z \end{pmatrix},$$

to find,

$$\begin{aligned} \frac{2a}{a+b} - 1 &= m^z \\ \mathrm{Re}(\frac{c}{a+b}) &= \frac{1}{2}\frac{c+c^*}{a+b} = m^x \\ \mathrm{Im}(\frac{c}{a+b}) &= \frac{1}{2i}\frac{c-c^*}{a+b} = m^y. \end{aligned}$$

This allows us to reuse the analysis in sections 1.7.1. With the use of the eigenvectors we get again that $p(y) = \frac{1}{2}(1 + y\frac{m^z}{\tilde{m}})$. Setting $p(y=1|\mathbf{x};\mathbf{w}) = p(y=-1|\mathbf{x};\mathbf{w})$ we find

$$\begin{aligned} \frac{1}{2}(1+\frac{m^z}{\tilde{m}}) &= \frac{1}{2}(1-\frac{m^z}{\tilde{m}}) \\ \frac{m^z}{\tilde{m}} &= 0, \end{aligned} \tag{1.33}$$

which is solved for $m^z = 0$. But unlike before the trivial solution $m^z = 0$ requires us to solve an additional equation,

$$\begin{aligned} \frac{2a}{a+b} - 1 &= 0 \\ a - b &= 0 \\ h_{00}^2 + h_{10}^2 - h_{01}^2 - h_{11}^2 &= 0. \end{aligned} \tag{1.34}$$

Before we continue, we need to introduce the concept of quadratic forms. The square of a dot product can be written as

$$h_{ij}h_{kl} = (\mathbf{w}_{ij}\cdot\mathbf{x})(\mathbf{w}_{kl}\cdot\mathbf{x}) = \sum_{\mu,\nu} w_{ij}^{\mu}x^{\mu}w_{kl}^{\nu}x^{\nu} = \mathbf{x}^T A_{ijkl}\mathbf{x} = 0.$$

If A is symmetric, then $\mathbf{x}^T A\mathbf{x}$ is a quadratic form. However, the form $\mathbf{x}^T A_{ijkl}\mathbf{x}$, is not symmetric since in general $w_{ij}^{\mu} \neq w_{kl}^{\nu}$. We can redefine $w_{ijkl}^{sym} = \frac{1}{2}(w_{ij}^0 w_{kl}^1 + w_{ij}^1 w_{kl}^0)$, $w_{ijkl}^{00} = w_{ij}^0 w_{kl}^0$ and $w_{ijkl}^{11} = w_{ij}^1 w_{kl}^1$ so that we can define a matrix B that is symmetric in terms of the weights w_{ijkl}^{sym}, w_{ijkl}^{00} and w_{ijkl}^{11} so that $\mathbf{x}^T B_{ijkl}\mathbf{x}$ is a quadratic form. We can use the fact that a quadratic form describes a quadric surface, e.g., surfaces that are described by the equation

$$\mathbf{x}^T Q_{ij}\mathbf{x} + \mathbf{R}\mathbf{x} + \mathbf{P} = 0. \tag{1.35}$$

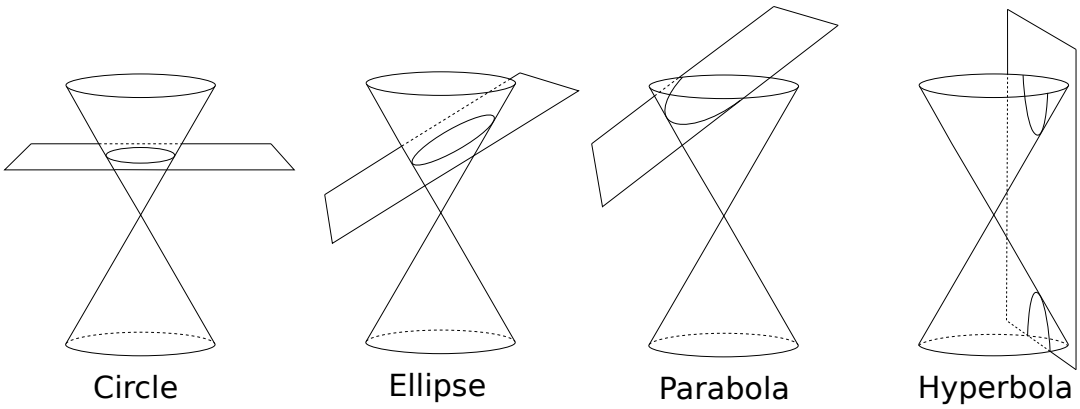


FIGURE 1.24: 4 of the non-degenerate (real) conic sections. We can also have a single point, if we take a plane at the apex of the cones, or a line the plane is tangent to the surface of the cone.

Let us consider the two-dimensional projection of some high-dimensional manifold. The quadric surface (setting $x_1 = x$ and $x_2 = y$ for the moment) is given by

$$ax^2 + 2bxy + cy^2 + 2dx + 2fy + g = 0, \tag{1.36}$$

where $a = w^{00}_{ijkl}$, $b = w^{sym}_{ijkl}$, $c = w^{11}_{ijkl}$ and $d = f = g = 0$. If we would add a bias, with $x_3 = 1$, then d, f and g would be nonzero. The different types of curves that we get for different values of these parameters are called conic sections, since they are created by the intersection of a plane with a cone. See figure 1.24. The coefficient b can always be set to zero by rotation of the coordinate system [62]. Equation 1.34 is a linear combination of quadratic forms, which is clearly also a quadratic form. The classification of which conic section we obtain is dependent on the following quantities [63]:

$$\Delta = \det \begin{pmatrix} a & b & d \\ b & c & f \\ d & f & g \end{pmatrix}, \; J = \det \begin{pmatrix} a & b \\ b & c \end{pmatrix}, \; K = \det \begin{pmatrix} a & d \\ d & g \end{pmatrix} + \det \begin{pmatrix} c & f \\ f & g \end{pmatrix}, \; I = a + b$$

Based on these we can determine the different geometric shapes of the quadric as can be seen in table 1.1.

TABLE 1.1: Since we only have real weights, we can only have real conics, so the shapes with a star are excluded.

Δ	J	Δ / I	K	Conic Type
$\neq 0$	< 0			Hyperbola
$\neq 0$	0			Parabola
$\neq 0$	> 0	< 0		Ellipse
$\neq 0$	> 0	> 0		Imaginary ellipse *
0	< 0			Intersecting lines
0	> 0			Point
0	0		< 0	Distinct Parallel lines
0	0		> 0	Imaginary parallel lines *
0	0		0	Coincident lines

What can we say about the probability curves $p(y=1|\mathbf{x};\mathbf{w}) = \frac{1}{2}(1+\epsilon)$? We have

$$\begin{aligned} \frac{m^z}{\tilde{m}} &= \epsilon \\ m^z &= \epsilon\tilde{m}, \end{aligned} \tag{1.37}$$

which looks complicated, because calculating $\tilde{m} = \sqrt{\sum_k (m^k)^2}$ with the previously made identifications between a, b, c and m^x, m^y, m^z looks cumbersome. We remember that $\tilde{m}$ also shows up in the eigenvalues of the density matrix as $\lambda_\pm = \frac{1}{2}(1 \pm \tilde{m})$ in equation 1.19. If the eigenvalues of equation 1.32 admit a similar form, we might be able to find a simple form for $\tilde{m}$. The matrix in equation 1.32 has characteristic polynomial $(a-\lambda)(b-\lambda) - c^2 = \lambda^2 - \lambda(a+b) + ab - c^2$. Solving this equation and taking the factor in front into account gives $\lambda'_\pm = \frac{1}{2}(1 \pm \sqrt{(a-b)^2 + 4c^2}/(a+b)$. We can identify $\tilde{m} = \sqrt{(a-b)^2 + 4c^2}/(a+b)$. Substituting these identifications in equation 1.37 gives

$$\begin{aligned} \frac{2a}{a+b} - 1 &= \epsilon \frac{\sqrt{(a-b)^2 + 4c^2}}{(a+b)} \\ a - b &= \epsilon \sqrt{(a-b)^2 + 4c^2} \\ a^2 - b^2 - 2ab &= \epsilon^2 (a^2 + b^2 - 2ab + 4c^2) \\ (a-b)^2(1-\epsilon^2) - \epsilon^2 4c^2 &= 0, \end{aligned}$$

which gives back the original condition $a - b = 0$ in the case that $\epsilon = 0$,

$$\begin{aligned} (a-b)\sqrt{1-\epsilon^2} &= \pm\epsilon 2c \\ (a-b) &= \pm \frac{\epsilon}{\sqrt{1-\epsilon^2}} 2c. \end{aligned}$$

Since this is again a linear combination of quadrics, we expect the separation boundaries to be quadric as well. We can observe some different quadratic separation boundaries in figure 1.25.

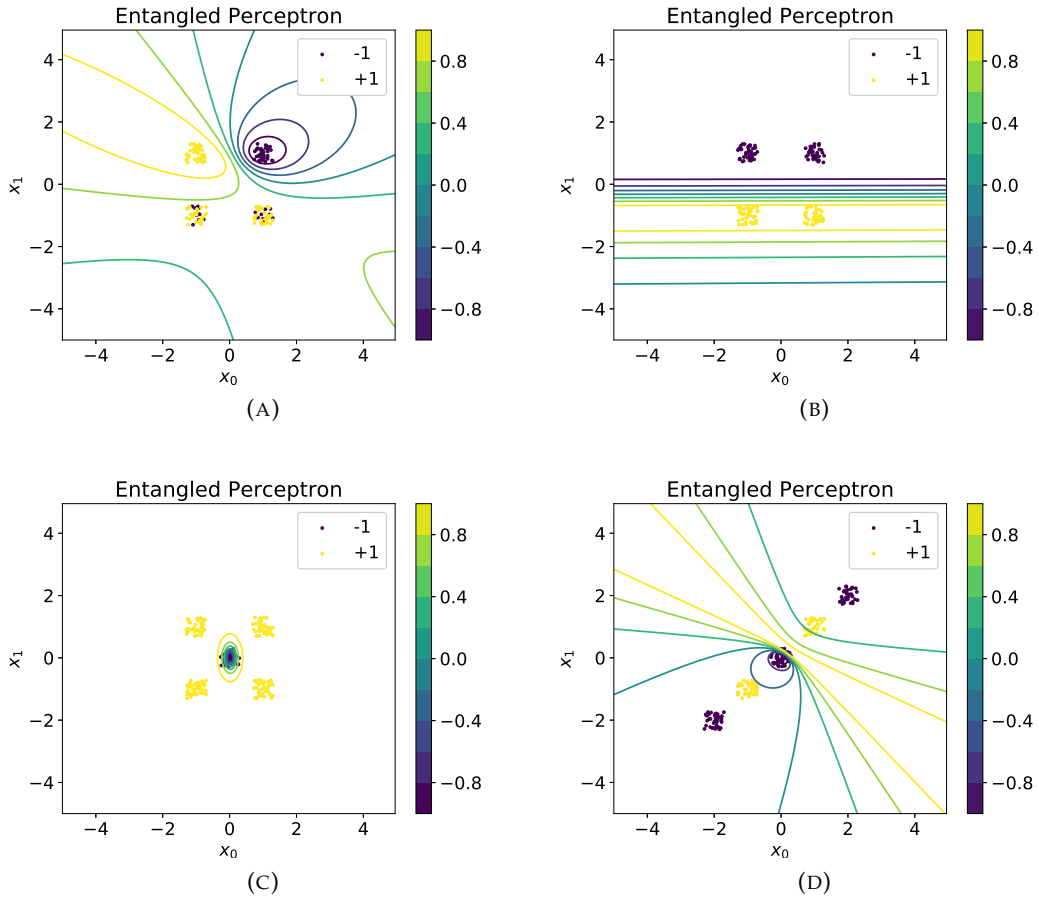


FIGURE 1.25: Separation boundaries of the entangled perceptron for additional two-dimensional problems. **(a)** We can still learn the same noisy problem that we studied in section 1.8, only now with quadric surfaces. **(b)** A quadric surface can also consist of parallel lines, allowing us to learn linearly separable problems. **(c)** For this specific problem we can find an elliptical separation boundary to perfectly classify the data. **(d)** Problems that cannot be solved with a quadric surface are still problematic and lead to bad solutions.

It is clear that the method of entangling qubits can also be used for multiple classes and continuous classes. See figures 1.26 and 1.27.

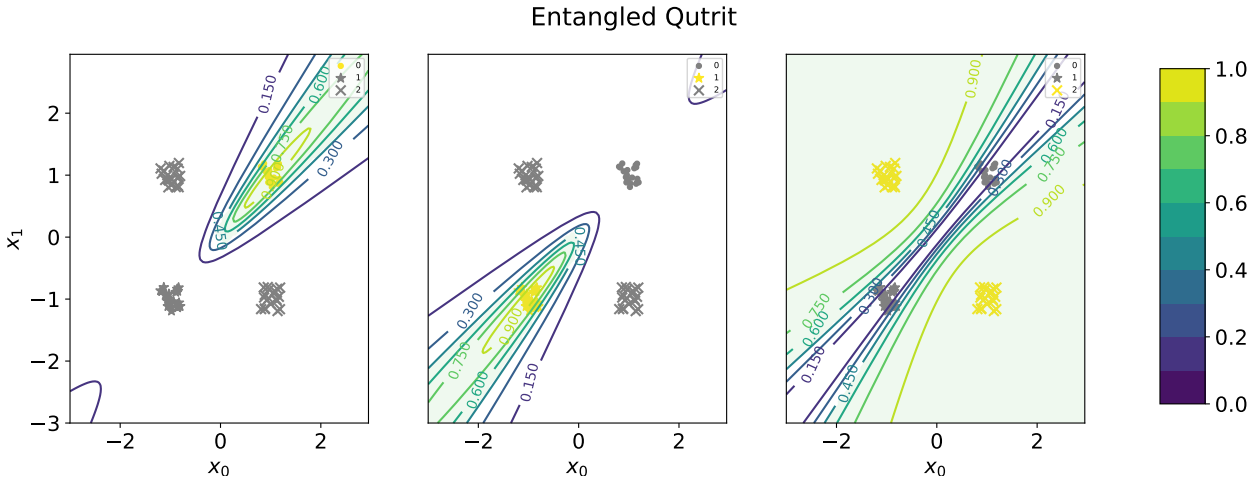


FIGURE 1.26: Separation boundary of entangled qutrit. For non-noisy nonlinear data we obtain perfect classification.

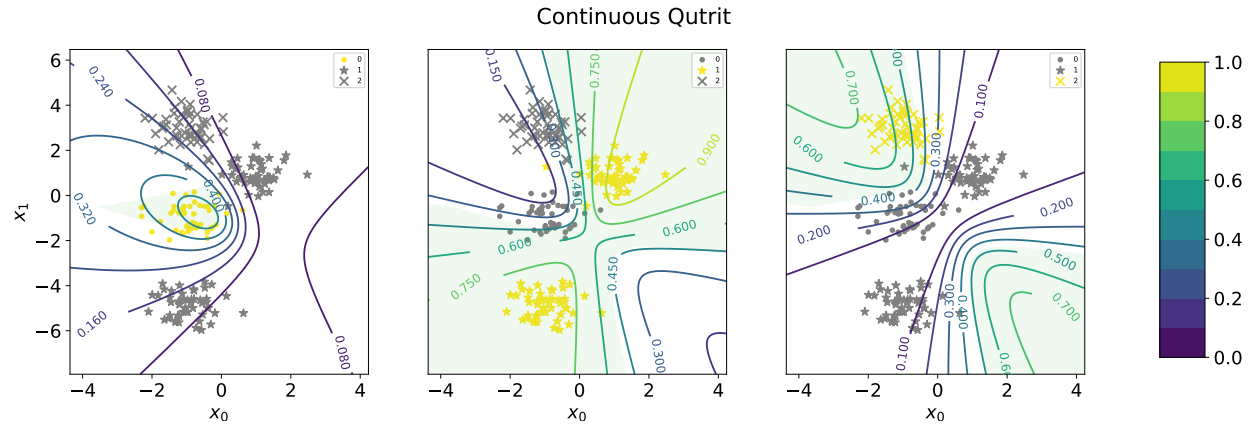


FIGURE 1.27: Separation boundary of entangled qutrit with continuous data. A combination of elliptic and hyperbolic separation boundaries makes it possible to learn this continuous data set.

Entangling more than 2 qubits with this method might seem worthwhile. However, this will not give any additional features. The wave function for multiple qubits is given by

$$|\psi\rangle = \frac{1}{Z} \sum_{i_1,\ldots i_n} h_{i_1,\ldots i_n} |\psi_{i_1}\rangle \otimes \ldots \otimes h_{i_n} |\psi_{i_n}\rangle ,$$

which contains fields h_{ij} corresponding to the basis state $|\psi_{i_1}\rangle \otimes \ldots \otimes h_{i_n} |\psi_{i_n}\rangle$. When we trace out a part of the system to create a reduced density matrix, we always end up with a linear combination of quadratic terms h_{ij}^2. The number of terms in this linear combination scales with the dimension of the total Hilbert space. But since all these terms are at most quadratic, the boundaries are always a quadric surface, since a linear combination of quadric surfaces is still a quadric surface, as noted before. Entangling more qubits thus only increases the number of parameters that determine the shape of the quadric surface.

To summarize, the quantum perceptron can be extended to learn nonlinear boundaries, such as the XOR problem. We have shown that this is possible because the entangled perceptron can learn quadric surfaces due to the quadratic parametrization of the reduced density matrix. This extension can be combined with the previous results to learn multiclass continuous data. We retain the robustness with respect to noise, because we still minimize the quantum log-likelihood with respect to a density matrix with off-diagonal elements.

1.11 Physical System

Up to this point we have taken a lot of freedom with the description of the model $\rho_{\mathbf{x}}$. We have parametrized density matrices and wave functions in the most general way possible by choosing the system Hamiltonian to be

$$H = \sum_k h^k \sigma^k.$$

While this description is convenient, it does not take the physical constraints of a system into account. As it turns out, we can still learn non-trivial problems if we limit the degrees of freedom significantly. Consider the Hamiltonian of two coupled qubits, both with transverse fields h_1 and h_2,

$$H_1 = h_1 {\sigma^x}_1 + h_2 {\sigma^x}_2 + J_{12} {\sigma^z}_1 {\sigma^z}_2,$$

with corresponding density matrix $1/Z e^{\beta H}$. The subindex of the Pauli matrix indicates the spin location. We are interested in this system, because it can be realized easily experimentally. By tuning the transverse fields we can control entanglement, which would be an interesting way to implement the perceptron. In order to minimize the quantum likelihood in equation 1.13 we have to obtain a single qubit density matrix, since $\eta_{\mathbf{x}}$ is a single qubit data density matrix. By tracing out one qubit we get $\rho_{red} = \mathrm{Tr}_2\{\rho\}$, which allows us to minimize the quantum log-likelihood,

$$\mathcal{L}_{phys} = -\sum_{\mathbf{x}} q(\mathbf{x}) \, \mathrm{Tr}_1\{\eta_{\mathbf{x}} \log(\rho_{red})\}.$$

Calculating $\mathrm{Tr}_2\{\rho\}$ is not easy. We either have to diagonalize a 4×4 matrix or approximate $\exp\{H\}$. Both can be done numerically, but we are interested in an analytic expression of the reduced exponent. Finding an analytic expression is cumbersome, since we have to write out

$$e^H = \sum_{i=0}^{\infty} \frac{H_1^n}{n!},$$

to some order, perform the partial trace over the second subspace and try to extract a nice expression for the sum generating this expression (see Appendix A.5). However, this might give some insight in the problems that we can learn with this physical system setup. The hope is that calculating the first five orders of H might allow us to write down an infinite series expansion for the partial trace of the exponent. If this retains the infinite radius of convergence of the exponent, we might be able to study the partial trace of an exponential operator in the limit $\beta \to \infty$ in an

analytic way. We calculate the first five non-trivial terms,

$$\frac{1}{Z}e_1^H \approx \frac{1}{Z}\left(1 + H_1 + \frac{1}{2}H_1^2 + \frac{1}{6}H_1^3 + \frac{1}{24}H_1^4 + \frac{1}{120}H_1^5\right) + \mathcal{O}(h^6).$$

If we trace over the second qubit we get the reduced density matrix. The full derivation can be found in the Appendix C. The first five orders of H are

$$\begin{aligned}
H_1 &= h_1\sigma_1^x + h_2\sigma_2^x + J_{12}^z\sigma^z{}_1\sigma^z{}_2,\\
H_1^2 &= c + 2h_1h_2\sigma_1^x\sigma_2^x,\\
H_1^3 &= cH_1 + 2\left(h_1^2h_2\sigma_2^x + h_1h_2^2\sigma_1^x + J_{12}^z h_1h_2\sigma_1^y\sigma_2^y\right),\\
H_1^4 &= (c^2 + 4(h_1h_2)^2 + 4(h_1h_2)^2),\\
H_1^5 &= c^2H_1 + 4c\left(h_1^2h_2\sigma_2^x + h_1h_2^2\sigma_1^x\right) + 4h_1h_2(h_1^2h_2\sigma_1^x + h_1h_2^2\sigma_2^x - J_{12}h_1h_2\sigma_1^z\sigma_2^z).
\end{aligned}$$

Calculating the partial trace and taking the factorial coefficients into account then gives

$$\begin{aligned}
\mathrm{Tr}_2\{\rho\} &\approx \frac{1}{2}\left(1 + \overbrace{h_1\left(1 - \frac{1}{3}\left[h_1^2 + J_{12}^2\right] + \frac{2}{15}\left(\left[h_1^2 + J_{12}^2\right]^2\right) + 2h_2^2J_{12}^2\right)}^{m^x}\sigma_1^x\right) + O(h^6) \\
&= \frac{1}{2}(1 + m^x\sigma_1^x).
\end{aligned} \tag{1.38}$$

Note that in the case that $J_{12} \to 0$ we only get back a dependence on parameters of spin 1. This is the expected behaviour, because if there is no interaction the full exponent factorizes and all dependence on h_2 should be eliminated by the trace. Our goal is now to find a sum which generates equation 1.38. Remember that the hyperbolic tangent is given by

$$\tanh(x) = x\left(1 - \frac{x^2}{3} + \frac{2x^4}{15}\right) + O(h^7).$$

Substituting $x = [h_1^2 + J_{12}^2]^{\frac{1}{2}}$ gives

$$\tanh\left([h_1^2 + J_{12}^2]^{\frac{1}{2}}\right) = [h_1^2 + J_{12}^2]^{\frac{1}{2}}\left(1 - \frac{1}{3}[h_1^2 + J_{12}^2] + \frac{2}{15}[h_1^2 + J_{12}^2]^2\right).$$

Multiplying with $\frac{h_1}{[h_1^2+J_{12}^2]^{\frac{1}{2}}}$ gives back the original expression, except for the $2h_2^2J_{12}^2$ term,

$$\tanh\left([h_1^2 + J_{12}^2]^{\frac{1}{2}}\right)\frac{h_1}{[h_1^2 + J_{12}^2]^{\frac{1}{2}}} = h_1\left(1 - \frac{1}{3}\left[h_1^2 + J_{12}^2\right] + \frac{2}{15}\left[h_1^2 + J_{12}^2\right]^2\right) + O(h^7).$$

Using a Mathematica script, we were able to determine even more terms. Our hope was that some structure might appear that would allow us to resum the $2h_2^2J_{12}^2$ in some way. But alas, it seems that the extra terms cannot be resummed easily to a nice

algebraic expression with our approach[2]. Nonetheless, for small our approximation will do fine. The reduced density matrix can be written as

$$\mathrm{Tr}_2\{\rho\} \approx \frac{1}{2}\left(1 + \tanh\left([h_1^2 + J_{12}^2]^{\frac{1}{2}}\right)\frac{h_1}{[h_1^2 + J_{12}^2]^{\frac{1}{2}}}\sigma_1^x\right) = \frac{1}{2}(1 + m^x\sigma_1^x). \tag{1.39}$$

If we want to learn a classical distribution, we need to have at least have a σ_z field so that we can control the diagonal elements of the density matrix. Luckily we can exchange $x \Leftrightarrow z$ by a simple basis transformation to obtain this,

$$H_1' = h_1\sigma^z{}_1 + h_2\sigma^z{}_2 + J_{12}\sigma^x{}_1\sigma^x{}_2,$$

which gives

$$\mathrm{Tr}_2\{\rho\} \approx \frac{1}{2}(1 + m^z\sigma_1^z),$$

with $m^x = m^z$ with $h_1 \Leftrightarrow h_2$ by symmetry. Based on this derivation, we hypothesize that constructing a system with Hamiltonian H_1' where the field h_1 is parametrized as $h_1 = \mathbf{x}\cdot\mathbf{w}_1$ will allow us to learn data under the constraints of a physical system. In figure 1.28 we see that this is indeed possible. Adding a field $g_1\sigma_1^x$ to H_1' also allows us to also learn the off diagonal elements. The derivation that shows this can again be found in Appendix C (with $x \Leftrightarrow z$). Adding noise to the data points gives us the curved probability boundaries as expected, as can be seen in figure 1.29.

In this section we have shown that the quantum perceptron model can still be learned for a physical system. By tuning a single transverse field we can construct a quantum system whose statistics give us the correct class probability. Adding another external field gives us the curved probability from the original quantum perceptron. Further investigation of exact expressions of reduced density matrices might be an fruitful direction of future research. Also, realizing the setup with a physical system experimentally might be interesting. Although for each field we still need to calculate $h_i = \mathbf{x}\cdot\mathbf{w}_i$ on a classical computer, the rest can be realized on an actual spin system.

[2]Using a different approach [64], Eduardo Dominiquez was able to determine an algebraic expression for the reduced density matrix. However, this method relies on calculating the algebraic expressions of the eigenvalues of a H, which becomes quite involved for more complex Hamiltonians. For our simple system this becomes

$$\mathrm{Tr}_2\{\rho\} = \frac{1}{2}\left(1 + \frac{1}{\cosh E_1 + \cosh E_2}\left(\frac{h_1 + h_2}{E_1}\sinh E_1 + \frac{h_1 - h_2}{E_2}\sinh E_2\right)\sigma_1^x\right),$$

with

$$E_1 = \sqrt{(h_1 + h_2)^2 + J_{12}}$$
$$E_2 = \sqrt{(h_1 - h_2)^2 + J_{12}}.$$

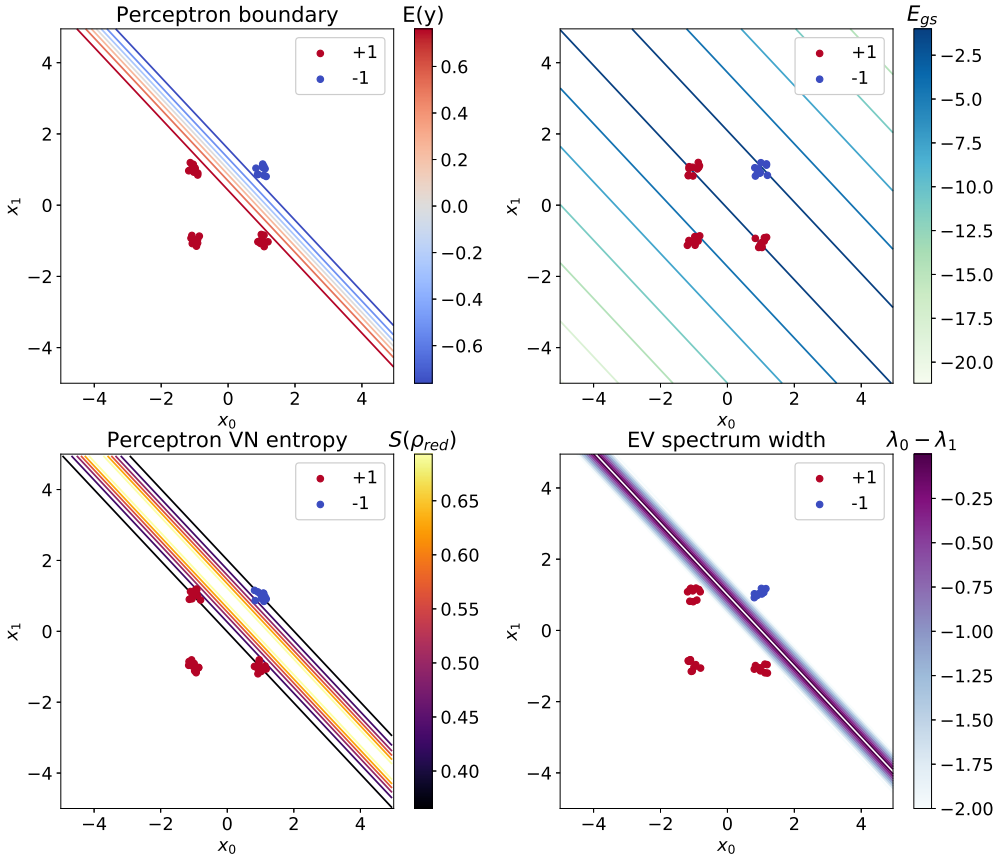


FIGURE 1.28: The simple binary classification problem from section 1.8 learned with a physical quantum system with $J_{12} = 1$. To break the degeneracy of the system we add a very small field $0.05\sigma^x$. The probability $p(y|\mathbf{x};\mathbf{w})$ is again constructed by measuring $\langle\sigma_1^z\rangle$. The top left figure shows the separation boundaries, which are equal to the logistic regression boundaries as expected, but $\langle\sigma_1^z\rangle$ is limited to the interval $[-0.7, 0.7]$. The top right shows us the ground state energy, which is spread symmetrically around the boundary. The bottom left shows the von Neumann entropy (see Appendix A.6). The closer we are to the boundary, the higher the entropy since the state uncertainty increases.In the bottom right we see the how the difference between the energies of the system's state determines the separation boundaries. When the energies are close, there is a large uncertainty about the class label.

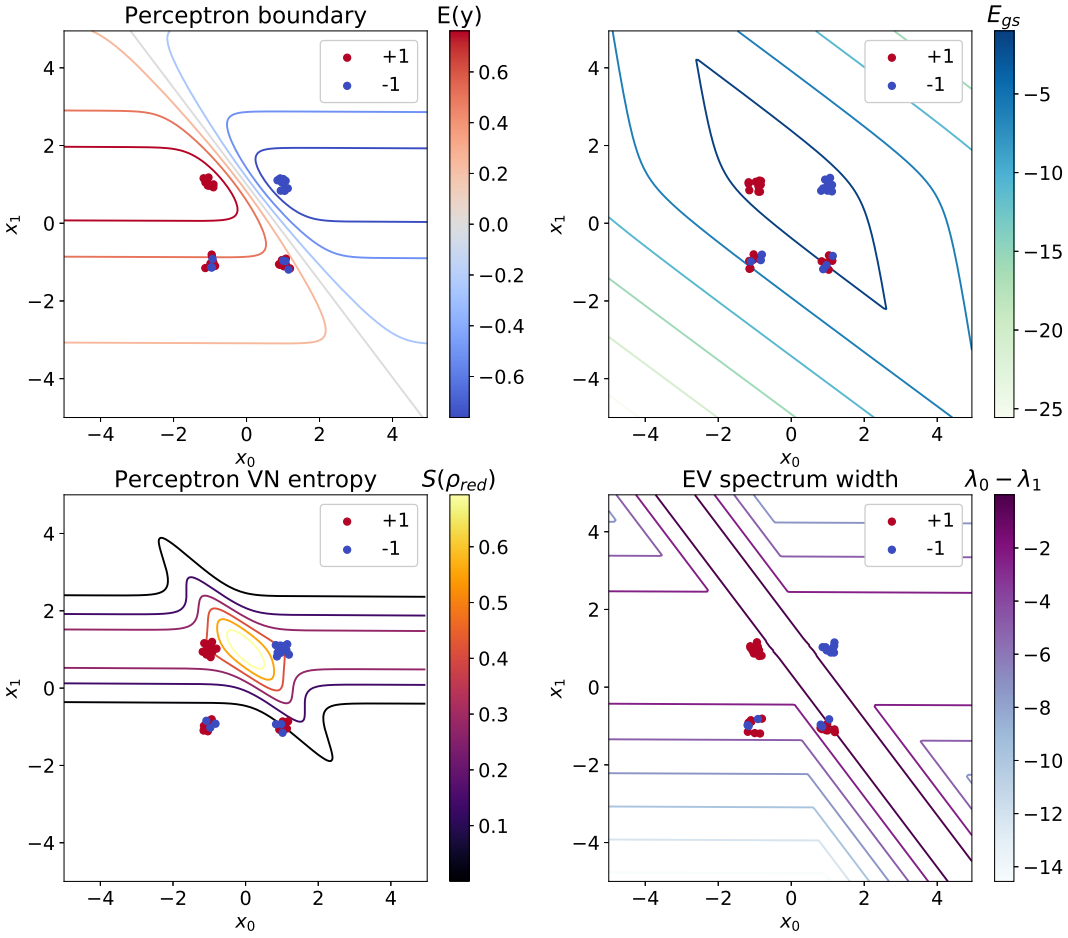


FIGURE 1.29: In the top left figure we see that the curved separation boundaries. The probability curves are no longer hyperbolic but look more like flattened parabolas. The maximal von Neumann entropy in the bottom left figure is no longer located on the boundary, but concentrated between the two non-noisy samples to account for the uncertainty about the two noisy samples. The top and bottom right figures reflect the changes in the separation boundary and uncertainty.

1.12 Λ-operator perceptron

In Amin's proposal for the quantum boltzmann a density matrix likelihood is constructed by limiting the trace of a modeled density matrix to the diagonal by applying a projection operator $\Lambda(y)$ [28]. This operator is used to reduce the trace to the visible units of the quantum boltzmann machine. We can use this operator for the perceptron to project the off diagonal elements of the density matrix onto the diagonal, giving a classical optimization criterion. We use the same $\rho_{\mathbf{x}}$ as in equation 1.12 and the same likelihood as in equation 1.13,

$$\mathcal{L}_\Lambda = \sum_{\mathbf{x},y} q(\mathbf{x})q(y|\mathbf{x}) \log(\mathrm{Tr}\{\Lambda(y)\rho_{\mathbf{x}}\}). \tag{1.40}$$

The projection operator is given by

$$\Lambda(y) = \frac{1}{2}(1 + y\sigma^z),$$

and the expression of traces in the likelihood in equation 1.40 becomes

$$\mathrm{Tr}\{\Lambda(y)\rho_{\mathbf{x}}\} = \mathrm{Tr}\Big\{\Lambda(y)e^{-H}/Z\Big\} = \frac{\mathrm{Tr}\Big\{\Lambda(y)e^{\sum_k h^k\sigma^k}\Big\}}{2\cosh h}.$$

Taking the log gives

$$= \log \mathrm{Tr}\Big\{\Lambda(y)e^{\sum_k h^k\sigma^k}\Big\} - \log 2\cosh h. \tag{1.41}$$

For the original quantum perceptron this expression simplified significantly, because the exponent would disappear due to the log, however the pesky Λ_y prevents us from doing this. If we want to minimize the log-likelihood we have to derive the gradient with respect to the parameter $\mathbf{w}^k$. Doing so gives the expression

$$\frac{\partial \mathcal{L}_\Lambda}{\partial \mathbf{w}^k} = \frac{\mathrm{Tr}\Big\{\Lambda(y)\frac{\partial}{\partial \mathbf{w}^k}e^{-H}\Big\}}{\mathrm{Tr}\{\Lambda(y)e^{-H}\}} - \tanh h \frac{h^k}{h}\mathbf{x}.$$

Since H and $\frac{\partial H}{\partial \mathbf{w}^k}$ do not commute, we have that $\frac{\partial}{\partial \mathbf{w}^k}e^{-H} \neq e^{-H}\frac{\partial}{\partial \mathbf{w}^k}H$. However in the limit of large n we can write

$$\frac{\partial}{\partial \mathbf{w}^k}e^{-H} = \sum_{m=1}^{n} e^{-m\delta\tau H}(-\frac{\partial}{\partial \mathbf{w}^k}H\delta\tau)e^{-\delta\tau(n-m)H}.$$

Taking $m\delta\tau = t$ infinitesimally small and $n \to \infty$, we can change the sum into an integral:

$$= -\int_0^1 dt\, e^{-tH}(\frac{\partial}{\partial \mathbf{w}^k}H)e^{-(1-t)H}.$$

The parameter t can be absorbed in the fields $tH \to \sum_k th^k\sigma^k = H(t)$. The full derivation of the gradient update rules is rather lengthy and can be found in Appendix B.

The final result is

$$\frac{\partial \mathcal{L}_\Lambda}{\partial \mathbf{w}^x} = \sum_{\mathbf{x},y} q(\mathbf{x})q(y|\mathbf{x}) \left(\frac{1}{4Z} \left(\sinh h \frac{h^x}{h} + \frac{2y}{h^2} \left(\frac{1}{2}\cosh h - \frac{1}{2h}\sinh h \right) h^x h^z \right) - \tanh h \frac{h^x}{h} \right) \mathbf{x},$$

$$\frac{\partial \mathcal{L}_\Lambda}{\partial \mathbf{w}^y} = \sum_{\mathbf{x},y} q(\mathbf{x})q(y|\mathbf{x}) \left(\frac{1}{4Z} \left(\sinh h \frac{h^y}{h} + \frac{2y}{h^2} \left(\frac{1}{2}\cosh h - \frac{1}{2h}\sinh h \right) h^y h^z \right) - \tanh h \frac{h^y}{h} \right) \mathbf{x},$$

$$\frac{\partial \mathcal{L}_\Lambda}{\partial \mathbf{w}^z} = \sum_{\mathbf{x},y} q(\mathbf{x})q(y|\mathbf{x}) \left(\frac{1}{4Z} \left(\sinh h \frac{h^z}{h} + y \left(1 + \frac{1}{h^2} \left(\frac{1}{2}\cosh h - \frac{1}{2h}\sinh h \right) \left(-h^x h^x - h^y h^y + h^z h^z \right) \right) \right) - \tanh h \frac{h^z}{h} \right) \mathbf{x},$$

with $Z = \frac{1}{2}(1 + y \tanh h \frac{h^z}{h})$. For a single sample $\mathbf{x}$ we rewrite these equations in the general form,

$$\begin{aligned}
\frac{\partial \mathcal{L}_\Lambda}{\partial \mathbf{w}^k} &= \sum_y q(y|\mathbf{x}) \frac{1}{2(1 + y m^z)} \left(m^k \cosh h + y C^k \right) - m^k \\
&= q(y = 1|\mathbf{x}) \frac{1}{2(1 + m^z)} \left(m^k \cosh h + y C^k \right) \\
&+ (1 - q(y = 1|\mathbf{x})) \frac{1}{2(1 - m^z)} \left(m^k \cosh h - y C^k \right) - m^k \\
&= m^k \left(\cosh h \frac{1 + m^z(1 - 2q(y = 1|\mathbf{x}))}{2(1 - (m^z)^2)} - 1 \right) + C^k \frac{2q(y = 1|\mathbf{x}) - m^z - 1}{2(1 - (m^z)^2)},
\end{aligned}$$

with $m^k = \tanh h \frac{h^k}{h}$ and C^k a constant that differs per gradient. For the second term, the numerator becomes zero if $m^k = 2q(y = 1|\mathbf{x}) - 1$, e.g. when the fields equal the empirical expectation values. We saw this principle of moment matching also for the magnetization perceptron. For the first term this is not as simple, only if the term proportional to $\cosh h$ is equal to one does it become zero. The fixed point behaviour of the boundaries beyond these two observations is difficult to probe due to the complex nature of the gradient rules. Fitting the model on our favorite toy model, the simple binary classification problem, provides some insight into what is going on. See figure 1.30.

The Λ-operator approach projects the quantum probability distribution onto the diagonal of the density matrix. We have seen that this produces some ugly looking gradients, which complicates the analysis of this model. Nevertheless we have shown that the model has similar characteristics as the quantum perceptron, but that even for a simple example it performs worse.

1.13 Note on Quantum Computing

Modern quantum computing architectures leave a lot to be desired, but are capable of performing a calculations coherently on small systems. Integrating quantum computing into this thesis would be a considerable amount of work that would differ conceptually a lot from the previous sections, which is why a deep analysis is not presented here. Instead, we discuss the outlines of some ideas that might be interesting for future work.

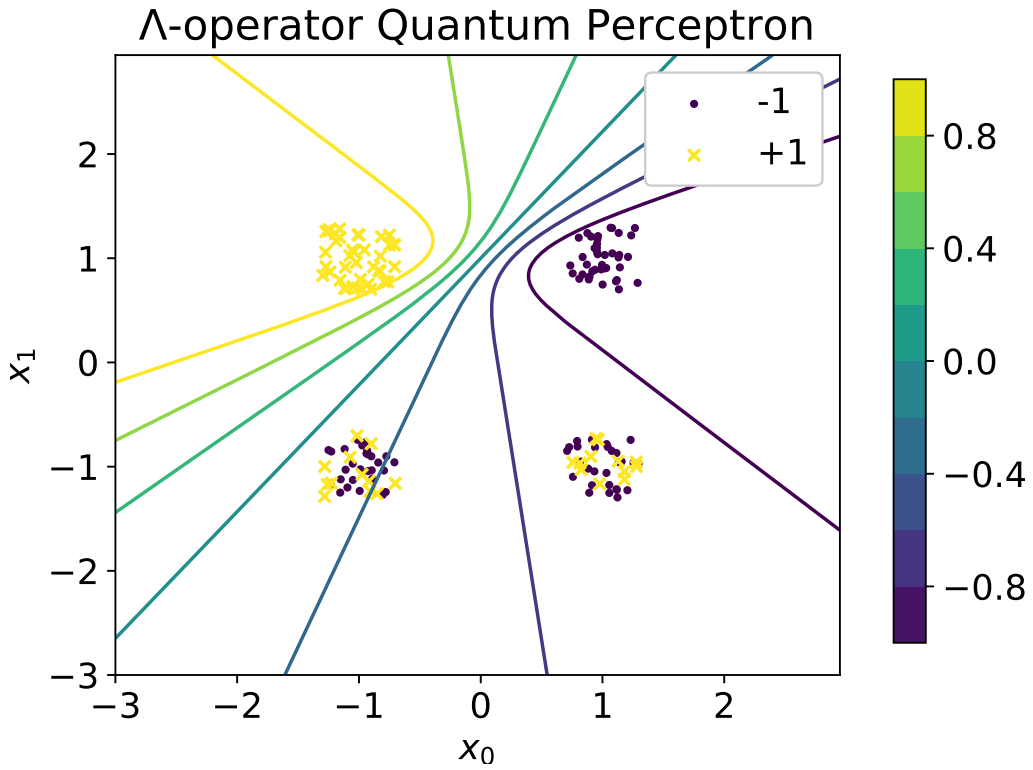


FIGURE 1.30: The Λ-operator perceptron has MSE(Λ) ≈ 0.111 which is higher than the MSE(*quantum*) ≈ 0.106 but still lower than MSE(*classical*) ≈ 0.154. We can attribute this difference in performance to the fact that the model cannot assign the same uncertainty to the two bottom data points. The point $(1, -1)$ has an expected value of $\mathbb{E}[y|\mathbf{x}; \mathbf{w}]_p \approx -0.8$ while the point $(-1, -1)$ has an expected value of $\mathbb{E}[y|\mathbf{x}; \mathbf{w}]_p \approx -0.4$. The Λ-operator perceptron is similar to the magnetization perceptron in this regard. The quantum perceptron assigned both these points an equal probability as desired, since both points have to same uncertainty.

The learned density matrix matrix of the quantum perceptron is a mixed state, which is difficult to prepare on a quantum computer, which is designed to work on pure states. But we can use a trick: we know that tracing out subsystems of entangled states can give mixed states. In fact, one can show that for any mixed state ρ_A there exists a pure density matrix $\rho = |\phi_{AR}\rangle\langle\phi_{AR}|$ such that $\text{Tr}_B(\rho) = \rho_A$ with $|\phi_{AR}\rangle \in \mathcal{H}_\mathcal{A} \otimes \mathcal{H}_R$. The system $\mathcal{H}_R$ is a fictitious system known as the reference system. Finding the pure density matrix ρ belonging to a mixed state is a process known as purification [32]. Consider the pure state $|\phi_{AR}\rangle$, which can be decomposed into a linear combination of tensorproduct states (see Appendix A.2),

$$|\phi_{AR}\rangle = \sum_{jk} a_{jk} |j\rangle |k\rangle , \tag{1.42}$$

where $a_{jk} \in \mathbb{C}$ so that $\langle\phi_{AR}|\phi_{AR}\rangle = 1$. With singular value decomposition we can write the matrix $a_{jk} = u_{ji} d_{ii} v_{ik}$,

$$|\phi_{AR}\rangle = \sum_{ijk} u_{ji} d_{ii} v_{ik} |j\rangle |k\rangle .$$

Define $\left|\psi_i^A\right\rangle \equiv \sum_j u_{ji} \left|j\right\rangle$, $\left|\psi_i^R\right\rangle \equiv \sum_k v_{ik} \left|k\right\rangle$ and $\sqrt{p_i} \equiv d_{ii}$, which gives

$$\left|\phi_{AR}\right\rangle = \sum_i \sqrt{p_i} \left|\psi_i^A\right\rangle \left|\psi_i^R\right\rangle,$$

with $p_i \in \mathbb{R}_+$. This is called the Schmidt decomposition. We can write the reduced density matrix as

$$\begin{aligned}
\mathrm{Tr}_R(\left|\phi_{AR}\right\rangle\!\left\langle\phi_{AR}\right|) &= \sum_{ij} \sqrt{p_i p_j} \left|\psi_i^A\right\rangle \left\langle\psi_j^A\right| \mathrm{Tr}_R(\left|\psi_i^R\right\rangle \left\langle\psi_j^R\right|) \\
&= \sum_{ij} \sqrt{p_i p_j} \left|\psi_i^A\right\rangle \left\langle\psi_j^A\right| \delta_{ij} \\
&= \sum_i p_i \left|\psi_i^A\right\rangle\!\left\langle\psi_i^A\right| = \rho_A.
\end{aligned}$$

$\left|\phi_{AR}\right\rangle$ is known as the purification of ρ_A. The purification scheme shows us that we can measure an effective mixed system by looking at the subsystem of a larger pure system. This means that we can simulate the quantum perceptron, which is a mixed state, on a quantum computer, such as the IBM Q Experience [65].

Additionally, researchers affiliated with Xanadu, a Canadian photonic quantum computing company, have developed a Python framework that enables automatic differentiation through quantum circuit called PennyLane [66, 67]. This would obviate the usage of a classical computer to calculate gradients, although updating parameters would still require a classical computer. At the moment, we are investigating the possibilities of integrating the quantum log-likelihood with PennyLane. The code and research for this project can be found on Github [68].

1.14 Conclusion

We extended the classical likelihood to a quantum log-likelihood and constructed a quantum perceptron from density matrices. The resulting algorithm is more resistant to noisy data when learning and takes this noisiness into account when predicting. This is due to the fact that there is a cost for flipped output labels in the quantum log-likelihood. For toy data sets we observed that the quantum perceptron is better at assigning probability to noisy samples, which resulted in improved performance. We focused on the analysis of possible separation boundaries for the model, to better understand what problems could be learned.

We have not only considered binary classification. The quantum perceptron could be extended to multiclass regression for $C > 2$ classes by considering the fundamental $\mathrm{SU}(C)$ representations instead of the Pauli matrices. A caveat of our model is that in order to get better results than a classical perceptron, we require multiple copies of a sample $\mathbf{x}$ with conflicting labels y to be present in the data. If this is not the case, our model reduces to logistic regression. By introducing a continuous similarity measure between samples that replaces the binary $b(\mathbf{x})$ statistic, we showed that non-discrete data could be learned while retaining the interesting boundaries of the quantum perceptron. Additionally, we have also shown that an entangled perceptron can be constructed with two qubits, and that an extension to n qudits is trivial as long as we trace out $n-1$ qudits so that we can use the quantum log-likelihood. However, with the current construction, entangling more qubits does

not give a more complex model, it simply increases the number of parameters that determine the shape of the quadric separation boundary.

The quantum perceptron also provides a good starting point for constructing other physically-inspired models. Firstly, we looked into a possible classical analogue, inspired by paramagnetic materials. Even though this model had some qualitative similarities with the quantum perceptron, the current construction we came up with lacked some elegance. Secondly, we showed that physical constraints pose no significant limitation for our model. By tuning the transverse fields of a two qubit system, we were able to learn the XOR problem. This shows that we do not require all degrees of freedom that the original entangled perceptron has access to. Finally, we briefly explored the relation with Amin's quantum boltzmann machine, and outlined the possibilities for integrating quantum computing with our research.

Appendix A

Quantum Mechanics

Isaac Newton's famed remark, "if I have seen further, it is by standing on the shoulders of giants" could not be more true for the development of quantum mechanics [69]. The 19th century gave birth to Maxwell's theory of electromagnetism which provided a theoretical framework for describing the transfer of energy through electromagnetic waves. At the same time, the likes of Boltzmann, Thomson, Joule and Clausius worked on combining heat with mechanical energy in the theory of thermodynamics. Together with classical mechanics these theories provided a very powerful framework for physics that led to numerous experiments being conducted to validate the theories to the highest precision possible. However, despite its successes there were still some unresolved issues. One of these issues was the origin and shape of the spectrum of a black body. Attempts by Rayleigh and Wien to explain the experimental data proved to be only sufficient in either the long or the short wavelength range. Max Planck had derived Wien's law in 1900 from entropic principles but as more data about the high frequency part of the spectrum became available he was forced to alter his arguments to account for the experimental data. By proposing that heated objects could only emit light in discrete chunks of energy $E_\gamma = h\nu$, where h is Planck's constant and ν the frequency of the photon, he successfully derived a law that could explain the the spectrum for long and short wavelengths, now known as Planck's law,

$$u(\nu, T) = \frac{8\pi h\nu^3}{c^3} \frac{1}{e^{h\nu/kT} - 1}.$$

Planck did not think that the quantization of energy implied that light had some discrete nature, he thought it a property of the radiation mechanism itself. It was not until 1905 that the discrete nature of light was proposed by Einstein who states in his paper [70]: "The energy [of a light ray] is not continuously distributed over an ever increasing volume, but it consists of a finite number of energy quanta, localized in space". The idea of quantizing energy also led to a new model for the atom, introduced by Bohr in 1913. The previous model was Rutherford's planetary atom model, which could explain experiments about scattering quite well, but fell short when describing atomic line spectra. Between the publication of Bohr's model and the introduction of wave mechanics in the second decade, a lot of effort was put in developing a model of orbital shells that could explain the spectra of all elements up to Xenon. Quantum numbers were introduced to characterize the orbits, but there were still problems with the model. The Zeeman effect was still not well understood until in 1924 Pauli introduced the concept of spin. Around the same time, both Schrödinger and Heisenberg developed their own formalism to explain the atomic spectra. Heisenberg developed a strange method to calculate the atomic spectra and wrote a, in his own words, "crazy" paper about his findings. Born and his student Jordan recognized the procedures Heisenberg used as matrix operations, and so the theory got the name "matrix mechanics". Parallel to this, Schrödinger wrestled with

the idea that if particles behaved like waves, as proposed by De Broglie, then their must be some wave equation describing said waves. While Skiing in the Swiss Alps he found that equation that now carries his name,

$$i\hbar\frac{\partial\Psi(\mathbf{r},t)}{\partial t} = \left(\frac{\hbar^2}{2m}\nabla^2 - V(\mathbf{r})\right)\Psi(\mathbf{r},t).$$

Paul Dirac showed the equivalence between the theories of Heisenberg and Schrödinger in 1927, developing a more general mathematical framework that contained the work of both physicists. However there were still some kinks to work out, as stated by von Neumann: "The method of Dirac, in no way satisfies the requirements of mathematical rigor" [71]. Von Neumann's 1932 "Mathematische Grundlagen der Quantenmechanik" would contain this mathematical rigor and to this day still forms the basis for a study into the mathematics of quantum mechanics. Parallel to the development of quantum mechanics, Einstein developed his theory of relativity. In thirty years the landscape of physics was transformed in a drastic manner, paving the way for what would be known as modern physics [72, 73].

A.1 Postulates of Quantum Mechanics

We will discuss the mathematical framework of quantum mechanics at lightning speed. The postulates of quantum mechanics can be boiled down to the following statements:

a) A physical state ϕ corresponds to a vector in a complex Hilbert space $\mathcal{H}$.

b) The observable quantities of a quantum system are defined by self-adjoint operators $\hat{A}$ on $\mathcal{H}$.

c) The expectation value of an observable quantity $\hat{A}$ for a system in a state ϕ is given by the inner product $(\phi, \hat{A}\phi)$.

A complex vector space V is a set of elements $v \in V$ over a field $\mathbb{C}$, which is closed under addition and multiplication by scalars $\alpha \in \mathbb{C}$. This means that any linear combination of vectors $v = \sum_i \alpha_i v_i$, $v_i \in V$, $\alpha \in \mathbb{C}$ is also contained in V. A complex Hilbert space contains additional structure in the form of an inner product (x, x) that produces a scalar in the complex plane. This inner product of $\mathcal{H}$ automatically defines a norm $\|x\| \equiv \sqrt{(x, x)}$. To avoid unnecessary complication of the required mathematical background we will consider a finite Hilbert space $\mathbb{C}^n$ so that all operators are bounded by definition. This obviates the need to fully describe infinite dimensional metric spaces and the completeness of such spaces, which we do not need anyway. We also consider only systems with a non-degenerate eigenvalue spectrum.

An orthonormal set of vectors is always linearly independent and spans the full Hilbert space. It is common to write the vectors x in the Dirac notation, where kets are denoted by $|x\rangle \in \mathcal{H}$ and bras by $\langle x| \in \mathcal{H}^*$. The starred Hilbert space $\mathcal{H}^*$ refers to the dual space, and its elements are called covectors. The operation $\langle x|y\rangle$ is not really an inner product, but a dual form. However, just like the inner product, it produces a scalar over the field of the respective vector space, which in the case is $\mathbb{C}^n$. If the Hilbert space is n-dimensional, then the standard basis $\mathbf{e}_i = \{\delta_{ij}\}^n$ is an orthonormal basis of the Hilbert space. We can write kets as vectors $|\psi\rangle = \sum_i a_i \mathbf{e}_i$ and bras as covectors $\langle\psi| = \sum_i a_i^* \mathbf{e}_i^\dagger$ with the additional operation of the standard dot-product. This definition ensures that the norm $\sqrt{\langle\psi|\psi\rangle}$ produces a real number larger than zero. We can convert a bra into ket by taking the conjugate transpose operation $\langle\psi|^\dagger \to |\psi\rangle$.

In order to perform measurements or transform a physical system, we require operators that act on these vectors (also known as state vectors). Operators in quantum physics are described in the language of abstract linear operators that act as basis transformations between two vector spaces. These operators are most often denoted by a hat: $\hat{A}$. In general two operators do not commute $\left[\hat{A}, \hat{B}\right] = \hat{A}\hat{B} - \hat{B}\hat{A} \neq 0$. Similar to bras and kets, there is a dual operator that works on elements of the dual vector space. These operators act on the states as $\left\langle\psi\middle|(\hat{A}\psi)\right\rangle = \langle\psi|\,\hat{A}\,|\psi\rangle$ and $\left\langle(\hat{A}\psi)\middle|\psi\right\rangle = \langle\psi|\,\hat{A}^\dagger\,|\psi\rangle$. In other words, $\hat{A}$ works on kets, while $\hat{A}^\dagger$ works on bras. It can be shown that operators can be represented by matrices and their operations of matrix addition and matrix multiplication, e.g. they are isomorphic to the abstract language of linear operators $\hat{A} \equiv \mathbf{A}$. Writing these matrices in terms of the standard basis gives $A_{ij}^\dagger = \mathbf{e}_i^T A^\dagger \mathbf{e}_j = (\mathbf{e}_j^T A \mathbf{e}_i)^* = A_{ji}^*$ by conjugate symmetry of the inner product, so $\mathbf{A}^\dagger$ is the conjugate transpose of $\mathbf{A}$.

Define a projector operator as $P_i = |v_i\rangle \langle v_i|$, this operator projects the component of some arbitrary state $|\phi\rangle$ out with respect to the basis $\{|v_i\rangle\}$,

$$P_i |\phi\rangle = |v_i\rangle \langle v_i| \sum_j a_j |v_j\rangle = |v_i\rangle \delta_{ij} a_j = a_i |v_i\rangle .$$

Furthermore, the sum of projectors equals the identity operation,

$$\left(\sum_i |v_i\rangle \langle v_i| \right) |\phi\rangle = \sum_{i,j} |v_i\rangle \langle v_i| a_j |v_j\rangle = \sum_{i,j} a_j |v_i\rangle \delta_{ij} = |\phi\rangle .$$

An operator is self-adjoint if $\hat{A} = \hat{A}^\dagger$. This is also referred to as a Hermitian operator. For every physical property $\mathcal{A}$ there exists an observable $\hat{A}$ that is a Hermitian operator on $\mathcal{H}$. The eigenvalues of this operator correspond to the possible values of the underlying physical property. If an operator $\hat{A}$ is Hermitian, there exists an orthonormal basis $\{|u_i\rangle\}$ of eigenvectors of $\hat{A}$, so that the matrix representation of $\hat{A}$ is diagonal with eigenvalues filling the diagonal of the matrix (assuming a non-degenerate spectrum). From the definition of the Hermitian operator we see that these eigenvalues must be real,

$$\hat{A} = \sum_i \lambda_i |u_i\rangle \langle u_i| . \tag{A.1}$$

This theorem is called the Spectral Theorem and is quite convenient because it allows to work with diagonal matrices, simplifying computations considerably. It also provides a way to relate the mathematical framework of quantum mechanics to the real world by measuring the eigenvalues of the operator we are interested in. Most often this operator is the Hamiltonian $\hat{H} \equiv H$ of the system, which contains the sum of potential and kinetic energies. The Schrödinger equation then tells us how the physical states of the system evolve,

$$i\hbar \frac{\partial |\psi\rangle}{\partial t} = H |\psi\rangle .$$

Obtaining information about some observable of a quantum system can be done through measurement. However, to obtain information about this system we must somehow interfere with the system. The Born rule bridges the quantum world with the classical world. Let $p_i = |\langle u_i|\psi\rangle|^2$ be the probability that we measure the eigenvalue λ_i for some self-adjoint operator $\hat{A}$, where $|u_i\rangle$ is the corresponding eigenvector of $\hat{A}$. The complex numbers a_i in $|\psi\rangle = \sum_i a_i |v_i\rangle$ are called probability amplitudes. After the measurement, the system collapses to the state $|u_i\rangle$ and measuring it again will return the eigenvalue λ_i with probability 1, this is known as wave function collapse. We can understand this behaviour by considering measurements to be projection operators consisting of eigenstates of observables,

$$p_i = \langle \psi|u_i\rangle \langle u_i|\psi\rangle = \langle \psi| P_i |\psi\rangle .$$

Since the result of single measurement is probabilistic, we are usually more interested in the average behaviour of an observable,

$$\langle \hat{A} \rangle = \langle \psi| \hat{A} |\psi\rangle .$$

The interpretation of quantum mechanics as a phenomenological theory is known

as the “Copenhagen Interpretation". According to this interpretation, there is a discontinuity between the quantum regime and the classical regime, bridged by invoking an observer that collapses the wave function of the system. This explanation is unsatisfactory to many physicists and as such, there exist many proposals to solve this so called “measurement problem" [74]. The philosophical debate on this topic spans almost a century and we will refrain from adding anything to it, for a detailed overview the reader is referred to [75].

A.2 Tensor Products

In classical physics we can describe the total phase space of a composite system as the Cartesian product of the subsystem phase spaces. For example Consider two systems $H(x)$ and $G(y)$, then the possible states for the composite system $K = H \times G = \{(x,y)|x \in H \text{ and } y \in G\}$ with $\dim K = \dim H + \dim G$. To describe composite quantum systems we require more mathematical tools, in the form of the tensor product. The formal definition is a bit abstract, but for completeness we will include it here. The tensor product of two vector spaces V and W is the map $\otimes : V \times W \to V \otimes W$ so that for $v \in V$, $w \in W$ and $\hat{v} \in V^*$, $\hat{w} \in W^*$ we have $(v \otimes w)(\hat{v}, \hat{w}) = \hat{v}(v) \otimes \hat{w}(w)$. For linear operators $\hat{A} : V \to V'$, $\hat{B} : W \to W'$ we have that $(\hat{A} \otimes \hat{B})(v \otimes w) = (\hat{A}v) \otimes (\hat{B}w)$.

This product establishes a connection between two vector spaces, its basis vectors and the operators that can be constructed in this new space. Let $\mathcal{H}_A$ and $\mathcal{H}_B$ be two Hilbert spaces. The tensor product space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$ is a vectorspace of dimension $\dim(\mathcal{H}) = \dim(\mathcal{H}_A) \cdot \dim(\mathcal{H}_B)$, with elements of $\mathcal{H}$ being linear combinations of $|\psi_A\rangle \otimes |\psi_B\rangle$, where $|\psi_A\rangle \in \mathcal{H}_\mathcal{A}$ and $|\psi_B\rangle \in \mathcal{H}_\mathcal{B}$. If $\{|v_i\rangle\}$ and $\{|v_j\rangle\}$ are orthonormal bases for $\mathcal{H}_A$ and $\mathcal{H}_B$, then $\{|v_i\rangle \otimes |v_j\rangle\}$ is a basis of $\mathcal{H}$. In the language of matrix representations we use the Kronecker product to calculate the tensor product of vectors and matrices,

$$|\psi\rangle = |\psi_A\rangle \otimes |\psi_B\rangle = \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} a_1 \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \\ \vdots \\ a_n \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \end{pmatrix},$$

where $|\psi\rangle$ is indexed by $a_i b_j$. For operators this is extended to

$$\hat{U} = \hat{A} \otimes \hat{B} \equiv \mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \otimes \begin{pmatrix} b_{11} & \dots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{m1} & \dots & b_{mm} \end{pmatrix}$$
$$= \begin{pmatrix} a_{11} \begin{pmatrix} b_{11} & \dots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{m1} & \dots & b_{mm} \end{pmatrix} & \dots & a_{1n} \begin{pmatrix} b_{11} & \dots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{m1} & \dots & b_{mm} \end{pmatrix} \\ \vdots & \ddots & \vdots \\ a_{n1} \begin{pmatrix} b_{11} & \dots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{m1} & \dots & b_{mm} \end{pmatrix} & \dots & a_{nn} \begin{pmatrix} b_{11} & \dots & b_{1m} \\ \vdots & \ddots & \vdots \\ b_{m1} & \dots & b_{mm} \end{pmatrix} \end{pmatrix},$$

so $\hat{U}$ is indexed by $A_{ij} B_{kl}$.

A.3 Density Matrix

Up to this point we have only considered state vectors as a description of a system, with probability amplitudes assigned to each possible state. In an experimental setting we have to repeat a certain measurement a number of times to obtain statistics of some observable. This assumes that we can prepare the system identically for each measurement, but this might prove to be difficult in practice. To deal with this problem we define a density matrix ρ to describes a statistical ensemble of quantum systems. A density matrix ρ for an n-dimensional quantum system $\mathcal{H}$ is a self-adjoint matrix with the properties

a) $\mathrm{Tr}\{\rho\} = 1$

b) ρ is non-negative, e.g. has only eigenvalues ≥ 0

c) $\langle \hat{A} \rangle = \mathrm{Tr}\{\rho \hat{A}\}$

d) $\rho = \sum_i \lambda_i |v_i\rangle \langle v_i|$ where $\{|v_i\rangle\}$ is an orthogonal basis

In terms of statevectors we have for two systems A and B that the combined state can always be written in the form

$$|\psi\rangle = \sum_i c_{ij} \left|\psi_i^A\right\rangle \otimes \left|\psi_j^B\right\rangle . \tag{A.2}$$

We say that the system is in a pure state if the density matrix is given by a single state vector,

$$\rho^{pure} = |\psi\rangle \langle\psi| . \tag{A.3}$$

The system is in a mixed state if it can be written in terms of a linear combination of pure states,

$$\rho^{mixed} = \sum_i p_i |\psi\rangle_i \langle\psi|_i . \tag{A.4}$$

with $\sum_i p_i = 1$ to ensure that the trace is 1. In other words, p_i is the probability $0 \leq p_i \leq 1$ of obtaining a certain pure state when preparing the system. If $n = 1$ and $p = 1$ then the system is in a pure state. If the state is pure then the matrix ρ

is idempotent, $\rho^2 = |\psi\rangle\langle\psi|\psi\rangle\langle\psi| = \rho$ and rank one. The average behaviour of a mixed system is calculated with the trace,

$$\langle \hat{A} \rangle = \mathrm{Tr}\{\hat{A}\rho\}. \tag{A.5}$$

A.4 Qubits

The study of quantum information concerns itself with understanding how information is contained and propagated in quantum systems. An essential ingredient for this study is the qubit: A two level quantum system and quantum equivalent of the classical bit. The easiest examples of a qubit in the physical world is the a spin-$\frac{1}{2}$ particle or the vertical and horizontal polarization of a photon. When working with spin-$\frac{1}{2}$ states the canonical choice of basis are the eigenstates of the Pauli σ^z operator,

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \quad \text{with} \quad |0\rangle = \mathbf{e}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad |1\rangle = \mathbf{e}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The two basis define the computational basis $\{|i\rangle\}$, which is commonly used in quantum computing. We can write the total state of a qubit as $|\psi\rangle = a|0\rangle + b|1\rangle$ with $|a|^2 + |b|^2 = 1$. Because the only constraint on the state is the normalization of the probability amplitudes $a, b \in \mathbb{C}$ we can rewrite this more generally as

$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle\right),$$

with $\theta, \phi, \gamma \in \mathbb{R}$. Due to the complex conjugation in the Born rule the term $e^{i\gamma}$ has no observable effects it can be left out, so the qubit is fully defined by the two angles θ and ϕ [32],

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle. \tag{A.6}$$

The angles θ and ϕ describe a point on a sphere with radius 1 called the Bloch sphere which can be seen in figure A.1. For mixed states we can also use the Bloch sphere. We know that a density matrix ρ is a trace 1, Hermitian matrix. This means that it can be written in terms of the Pauli basis,

$$\rho = \frac{1}{2}(\mathbb{1} + \sum_k r^k\sigma^k) = \frac{1}{2}(\mathbb{1} + \mathbf{r}\cdot\boldsymbol{\sigma}) = \begin{pmatrix} 1+r^z & r^x - ir^y \\ r^x + ir^y & 1-r^z \end{pmatrix},$$
$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad , \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad , \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{A.7}$$

The vector $\mathbf{r}$ is called the Bloch vector with $\|\mathbf{r}\| \leq 1$ and equality when the state $|\psi\rangle$ is pure. The Bloch sphere visualization can be useful to provide some insight into how individual qubits are affected by operators. We will use this in chapter 3 to show how the quantum a classical distribution over a qubit differs from the quantum equivalent.

When working with a multi-qubit system we often use the shorthand notation $|00\rangle \equiv |0\rangle \otimes |0\rangle$, for instance a state $|00\rangle\langle 11| = (|0\rangle \otimes |0\rangle)(\langle 1| \otimes \langle 1|) = |0\rangle\langle 1| \otimes |0\rangle\langle 1|$ by the previous definition of the tensorproduct. Visualizing the density matrix becomes harder since the length of the vector scales with $(n^2 - 1)$ for a $n \times n$ density matrix.

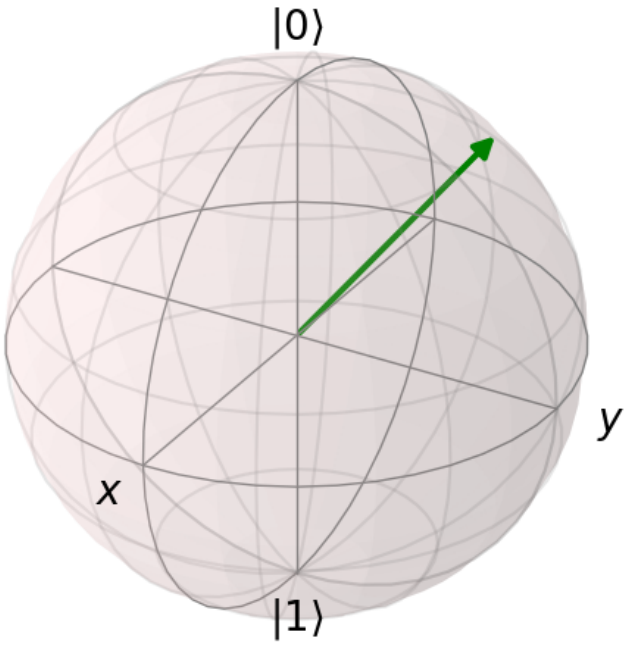


FIGURE A.1: The Bloch sphere of a qubit state with $\theta = \frac{1}{4}\pi$ and $\phi = \frac{3}{4}\pi$. The vector is in spherical coordinates written as $\mathbf{a} = (\sin(\theta)\cos(\phi), \sin(\theta)\sin(\phi), \cos(\theta))$ [76].

For 2 qubits this would already give a 15 dimensional vector that we would have to visualize.

A.5 Partial Traces

Just like in classical mechanics, we might be interested in only a smaller section of the entire phase space, for instance a single particle. We can trace out degrees of freedom in quantum physics by taking the partial trace of the density matrix. This will outline some of the strange properties of quantum physics, where looking at subsystems decreases the knowledge of the entire system, as opposed to increasing it.

Let ρ_{AB} be the density matrix in a complex Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$. We can find the reduced density matrix $\rho_A \in \mathcal{H}_A$ by taking the partial trace over the subsystem B,

$$\rho_A \equiv \mathrm{Tr}_B(\rho_{AB}). \tag{A.8}$$

The partial trace is the linear map $\mathrm{Tr}_B : \mathcal{L}(\mathcal{H}_A \otimes \mathcal{H}_B) \to \mathcal{L}(\mathcal{H}_A)$ where $\mathcal{L}(V)$ is the space of linear operators on the vector space V. This operation is then given by

$$\mathrm{Tr}_B(\rho_{AB}) \equiv \rho_A \,\mathrm{Tr}_B(\rho_B). \tag{A.9}$$

This approach looks straightforward enough, we remove the degrees of freedom belonging to the subsystem we are not interested in. Consider the system in the product state $\rho_{AB} = \rho_A \otimes \rho_B$ with $\rho_A \in \mathcal{H}_A$, $\rho_B \in \mathcal{H}_B$ and $\rho_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$. Performing the partial trace over B gives gives $\rho_A \,\mathrm{Tr}_B(\rho_B) = \rho_A$, and vice versa for A.

We can also imagine a system $\rho_{AB} = \sum_i p^i \rho_A^i \otimes \rho_B^i$ consisting of two qubits. For instance the famous Bell State,

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \tag{A.10}$$

has a corresponding density matrix

$$\rho = |\psi\rangle\langle\psi| = \frac{1}{2}(|00\rangle\langle 00| + |11\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 11|). \tag{A.11}$$

Remember that a state $|00\rangle\langle 11| \equiv |0\rangle_A\langle 1|_A \otimes |0\rangle_B\langle 1|_B$. If we trace out qubit B, we get

$$\begin{aligned}\mathrm{Tr}_B(\rho) =& \frac{1}{2}(\mathrm{Tr}_B(|0\rangle_A\langle 0|_A \otimes |0\rangle_B\langle 0|_B) + \mathrm{Tr}_B(|1\rangle_A\langle 0|_A \otimes |1\rangle_B\langle 0|_B) + \\ & \mathrm{Tr}_B(|0\rangle_A\langle 1|_A \otimes |0\rangle_B\langle 1|_B) + \mathrm{Tr}_B(|1\rangle_A\langle 1|_A \otimes |1\rangle_B\langle 1|_B)) \\ =& \frac{1}{2}(|0\rangle_A\langle 0|_A + |1\rangle_A\langle 1|_A).\end{aligned}$$

Notice that this state is a mixed state, a linear combination of density matrices. For the composite system ρ we have a pure state, which indicates full knowledge of the system. But, for the subsystem A we are uncertain of the state that it is in, since we have probability $\frac{1}{2}$ of finding the system in either state $|0\rangle$ or $|1\rangle$. Looking at a subsystem decreases the knowledge of the entire system [32, 77]!

A.6 Entanglement

In the previous paragraph we have introduced the notion of entanglement: The properties of a system cannot be described solely in terms of the individual subsystems. Measuring subsystem A impacts the information we can obtain about subsystem B. Despite its success, this "spooky action at a distance" was a fundamental problem for quantum physics according to Einstein, Podolsky and Rosen [78]. It would take 30 years before Bell would prove that a local theory could not be consistent with quantum theory. The concept of entanglement has received a lot of renewed interest in the 1990s and 2000s with the rise of quantum computing and quantum information. A full theory of entanglement is still not realized, mostly because of its complex structure.
The origin of entanglement is the tensor product of subsystem Hilbert spaces. For some composite system $\mathcal{H} = \bigotimes_i^n \mathcal{H}_i$ we have an orthonormal basis $\{|v_i\rangle = |v_{i_1}\rangle \otimes |v_{i_2}\rangle \otimes \ldots \otimes |v_{i_n}\rangle\}$ with indices $i_j = 1, \ldots, \dim(H_i)$. The total state of the system is described by

$$|\psi\rangle = \sum_i a_i |v_i\rangle .$$

with $\sum_i a_i^2 = 1$, $a_i \in \mathbb{C}$ and this can in general not be described as a product of states of individual subsystems,

$$|\psi\rangle \neq |\psi_1\rangle \otimes |\psi_2\rangle \otimes \ldots \otimes |\psi_n\rangle .$$

This inequality expresses formally the phenomenon of entanglement. If the total system consists of two subsystems, the system is called bipartite. If it consists of n subsystems it is called multipartite. The above definition is only valid for a pure

system $\rho = |\psi\rangle\langle\psi|$. In the case of a mixed state, we have to extend the definition with density matrices. A mixed state of n systems is entangled if it cannot be written as a linear combination of product states,

$$\rho \neq \sum_i p_i \rho_1^i \otimes \ldots \otimes \rho_n^i. \tag{A.12}$$

If such a combination does exist, then the states are called separable. In practice it is hard to decide if states are separable or entangled based on this definition, we therefore look to other ways of measuring entanglement. The study of the quantification of entanglement is a vast field of research, and the proposals for entanglement measures are numerous [79].
A key concept in classical information theory is the so called Shannon entropy which says something about the average information we gain when we learn the value of a random variable X. We can also view this as a measure of uncertainty about X before we learn its value [32],

$$H_s(X) \equiv H_s(p_1, ..., p_n) \equiv -\sum_i p_i \log_2 p_i. \tag{A.13}$$

This definition is motivated by Shannon to represent the minimal amount of bits required to reconstruct the information produced by some source. There is a quantum equivalent of this called the von Neumann entropy which extends this definition to density matrices,

$$S(\rho) \equiv -\operatorname{Tr}\{\rho \log_2 \rho\} = -\sum_i \lambda_i \log_2 \lambda_i,$$

by diagonalization of ρ where λ_i are the eigenvalues of ρ. This definition reduces to the classical definition in the case that ρ is diagonal. Note that both the Shannon and von Neumann entropy lie between 0 and 1 and we assume that $0 \log_2 0 = 0$.
For a bipartite system we can quantify the amount of entanglement with the relative entanglement entropy, which is defined as the von Neumann entropy of the reduced density matrix,

$$S(\rho_A) = -\operatorname{Tr}\{\rho_A \log_2 \rho_A\}.$$

Consider a system of two qubit A and B of the form,

$$|\psi\rangle = \frac{1}{\sqrt{2a^2 - 2a + 1}} \left(a\,|00\rangle + (1-a)\,|11\rangle\right).$$

We can trace out qubit B to obtain the reduced density matrix ρ_A and calculate the entropy as a function of a, this can be seen in figure A.2. This definition of a quantum entropy works only for pure states. If we consider a simple mixed state of non-entangled systems we can see that the von Neumann entropy becomes maximal, where we would like it to be zero. For instance, consider the following systems and corresponding mixed state ensemble,

$$\rho_1^A = |0\rangle\langle 0|, \rho_1^B = |1\rangle\langle 1|, \rho_2^A = |1\rangle\langle 1|, \rho_2^B = |0\rangle\langle 0|$$

$$\text{choose } p_1 = p_2 = \frac{1}{2}$$

$$\rho = \sum_i p_i(\rho_i^A \otimes \rho_i^B) = \frac{1}{2}\left(\overbrace{\rho_1^A \otimes \rho_1^B}^{\text{pure state}} + \overbrace{\rho_2^A \otimes \rho_2^B}^{\text{pure state}}\right).$$

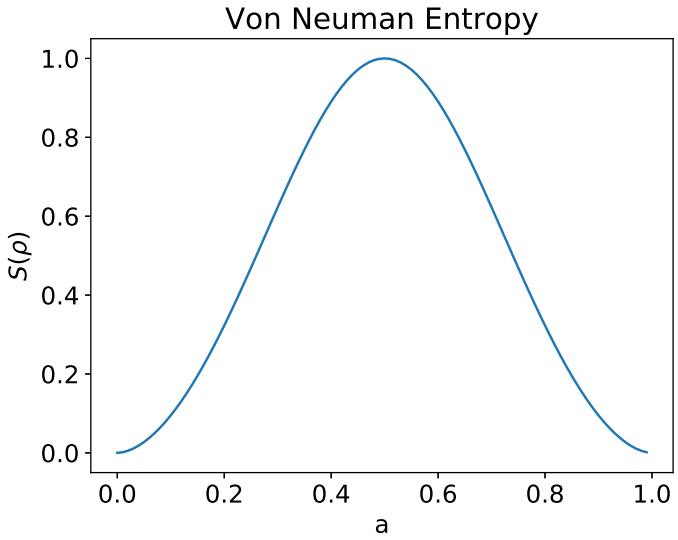


FIGURE A.2: The von Neumann entropy for a parameterized state. For $a = 0.5$ the relative entropy is maximal and corresponds to the Bell state in equation A.10.

If we now try to calculate the entropy for this mixed system, we will find that the subsystem entropy is maximal, while according to the definition of equation A.12 this system is not entangled since it is a linear combination of density matrix product states,

$$\mathrm{Tr}_A\,\rho = \frac{1}{2}\left(\rho_1^B + \rho_2^B\right)$$
$$S(\rho_B) = -2(\frac{1}{2}\log_2\frac{1}{2}) = 1.$$

So a bipartite mixed state has maximum entropy while being not entangled. This indicates that finding a good measure of entanglement might not be as simple as it seems. The literature on this subject is extensive [79].

Appendix B

Derivation Λ-operator perceptron

Inspired by Amin [28] we consider here the classical likelihood generalization to density matrices. In the previous proposal we start with the KL divergence and a semi-classical conditional density matrix, here we start with the classical log-likelihood,

$$\mathcal{L}_\Lambda = -\sum_{\mathbf{x}} q(\mathbf{x}) q(y|\mathbf{x}) \log(p(y|\mathbf{x};\mathbf{w})). \tag{B.1}$$

The model probability distribution $p(y|\mathbf{x};\mathbf{w})$ is replaced by the trace over a density matrix ρ times an observable operator Λ. Amin uses this operator to reduce the trace to the visible units of the quantum Boltzmann machine. Instead of minimizing the distance between the density matrices of data and model, we minimize the distance between the true label y and the projected probability of an up or down state $\Lambda\rho$, which would generalize to the classical case of minimizing $\mathcal{L}_\Lambda = -\sum_{\mathbf{x}} q(\mathbf{x}) \log p(y;\mathbf{x})$. We use the log-likelihood given by

$$\mathcal{L}_\Lambda = -\sum_{\mathbf{x}} q(\mathbf{x}) q(y|\mathbf{x}) \log(\mathrm{Tr}\{\Lambda\rho\}). \tag{B.2}$$

The operator Λ limits the trace only to diagonal terms that correspond to the variables being in state y. So for a certain sample x^μ we have label $y = y$. This operator can be written as

$$\Lambda = \frac{1}{2}(1 + y\sigma^z).$$

However, Λ corresponds to the two matrices

$$y = 1 \to \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$$
$$y = -1 \to \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}.$$

As proposal for the density matrix we take

$$\rho = \frac{1}{Z} e^{-H} \tag{B.3}$$
$$\text{where } H = \sum_k h^k \sigma^k \text{ with } h^k \in \mathbb{R} \text{ and } k = (x, y, z) = (1, 2, 3),$$

which is the grand canonical ensemble description at some temperature β, which is absorbed in the fields h^k. We parameterize these fields as follows:

$$\begin{aligned}\frac{m^k}{m} &= \frac{h^k}{h},\\ h &= \sqrt{\sum_k (h^k)^2},\\ m &= \tanh(h),\end{aligned}$$

so the total input h is squashed between $(-1, 1)$. We further parameterize the field h^k by the dot product of inputs times weights: $h^k = \mathbf{w}^k \cdot \mathbf{x}$. We use that ρ is a Hermitian matrix which can be written as

$$\rho = \frac{1}{2}(1 + \sum_k m^k \sigma^k) = \frac{1}{2}(1 + \tanh(h) \sum_k \frac{h^k}{h} \sigma^k),$$

or

$$2\rho \cosh(h) = \cosh(h) + \sinh(h) \sum_k \frac{h^k}{h} \sigma^k.$$

We also use that we can write an exponential of Pauli matrices as

$$e^{a \sum_k v^k} = \cosh(a) + \sinh(a) \sum_k v^k,$$

to write

$$\begin{aligned}2\rho \cosh(h) &= e^{\sum_k h^k \sigma^k}\\ \rho &= \frac{e^{\sum_k h^k \sigma^k}}{2\cosh(h)},\end{aligned}$$

so $Z = 2\cosh(h)$ in equation B.3. The expression of traces in the likelihood in equation reads

$$\mathrm{Tr}\{\Lambda\rho\} = \mathrm{Tr}\Big\{\Lambda e^{-H}/Z\Big\} = \frac{\mathrm{Tr}\Big\{\Lambda e^{\sum_k h^k \sigma^k}\Big\}}{2\cosh(h)}.$$

Taking the log gives

$$= \log \mathrm{Tr}\Big\{\Lambda e^{\sum_k h^k \sigma^k}\Big\} - \log 2\cosh(h). \tag{B.4}$$

For the original quantum perceptron this expression simplified significantly, because the exponent would disappear due to the log, however the pesky Λ prevents us from doing this. If we want to minimize the log-likelihood we have to derive the gradient with respect to the parameter w. Doing so gives the expression

$$\frac{\partial}{\partial \mathbf{w}^k} \mathcal{L}_\Lambda = \frac{\mathrm{Tr}\Big\{\Lambda \frac{\partial}{\partial \mathbf{w}^k} e^{-H}\Big\}}{\mathrm{Tr}\{\Lambda e^{-H}\}} - \tanh(h)\frac{h^k}{h}\mathbf{x}.$$

Since H and $\frac{\partial}{\partial \mathbf{w}^k} H$ do not commute, we have that $\frac{\partial}{\partial \mathbf{w}^k} e^{-H} \neq e^{-H} \frac{\partial}{\partial \mathbf{w}^k} H$. However, if we define $e^{-H} = [e^{-\delta\tau H}]^n$ with $\delta\tau \equiv 1/n$ where $\frac{\partial}{\partial \mathbf{w}^k} e^{-\delta\tau H} = e^{-\delta\tau H} \frac{\partial}{\partial \mathbf{w}^k} H \delta\tau$ does

commute,

$$\frac{\partial}{\partial \mathbf{w}^k} e^{-H} = \sum_{m=1}^{n} e^{-m\delta\tau H} (-\frac{\partial}{\partial \mathbf{w}^k} H \delta\tau) e^{-\delta\tau(n-m)H}.$$

Taking $m\delta\tau = t$ infinitesimally small and $n \to \infty$, we can change the sum into an integral,

$$= -\int_0^1 dt\, e^{-tH} (\frac{\partial}{\partial \mathbf{w}^k} H) e^{-(1-t)H}.$$

The introduction of an imaginary time requires us to redefine the Hamiltonian that we use. The parameter t can be absorbed in the fields $tH \to \sum_k th^k\sigma^k = H(t)$. The time independent Hamiltonian gives a linear dependency on the inputs $\mathbf{x}$ with $\frac{\partial}{\partial \mathbf{w}^k} H = \sigma^k \mathbf{x}$. Combining definitions B en B.3 gives

$$e^{-H(t)} = \cosh h \left(1 + \sum_k m^k(t)\sigma^k\right),$$

with the fields

$$\begin{aligned} \frac{m^k(t)}{m(t)} = \frac{th^k}{h(t)} &\to m^k(t) = \frac{h^k}{h}\tanh(th) \\ h(t) &= t\sqrt{\sum_k (h^k)^2} \\ m(t) &= \tanh(th). \end{aligned}$$

Putting everything together starts our journey,

$$= \frac{\mathrm{Tr}\left\{\Lambda \frac{\partial}{\partial \mathbf{w}^k} e^{-H}\right\}}{\mathrm{Tr}\{\Lambda e^{-H}\}} = -\int_0^1 dt\, \frac{\mathrm{Tr}\left\{\Lambda e^{-H(t)} (-\frac{\partial}{\partial \mathbf{w}^k} H) e^{-H(1-t)}\right\}}{\mathrm{Tr}\{\Lambda e^{-H}\}}. \tag{B.5}$$

Only the numerator is dependent on t, so isolate this term,

$$\begin{aligned} &= \int_0^1 dt\ \mathrm{Tr}\left\{\Lambda e^{-H(t)} (\frac{\partial}{\partial \mathbf{w}^k} H) e^{-H(1-t)}\right\} \\ &= \int_0^1 dt\ \mathrm{Tr}\left\{\Lambda \cosh(th)\cosh((1-t)h)\frac{1}{2}\left(1 + \sum_{k'} m^{k'}(t)\sigma^{k'}\right)(\sigma^k \mathbf{x})\frac{1}{2}\left(1 + \sum_{k''} m^{k''}(1-t)\sigma^{k''}\right)\right\} \\ &= \int_0^1 dt\, \frac{1}{4}\cosh(th)\cosh((1-t)h)\,\mathrm{Tr}\left\{\frac{1}{2}(1 + y\sigma^z)\left(\sigma^k + \sum_{k'} m^{k'}(t)\sigma^{k'}\sigma^k\right)\left(1 + \sum_{k''} m^{k''}(1-t)\sigma^{k''}\right)\right\}\mathbf{x} \\ &= \int_0^1 dt\, \frac{1}{8}\cosh(th)\cosh((1-t)h)\,\mathrm{Tr}\left\{(\overbrace{1}^{\mathbf{1}} + \overbrace{y\sigma^z}^{\mathbf{2}})(\sigma^k + \sum_{k''} m^{k''}(1-t)\sigma^k\sigma^{k''} + \sum_{k'} m^{k'}(t)\sigma^{k'}\sigma^k \right. \\ &\left. + \sum_{k'k''} m^{k'}(t)m^{k''}(1-t)\sigma^{k'}\sigma^k\sigma^{k''})\right\}\mathbf{x}. \end{aligned}$$

The full term is split into two parts, indicated in bold with **1** and **2**. Although this looks like a mess, we can make use of the trace relations of the Pauli matrices to simplify the mathematics considerably.

Term 1

We look only at the trace, which can be reduced to

$$= \mathrm{Tr}\left\{\sigma^k + \sum_{k''} m^{k''}(1-t)\sigma^k\sigma^{k''} + \sum_{k'} m^{k'}(t)\sigma^{k'}\sigma^k + \sum_{k'k''} m^{k'}(t)m^{k''}(1-t)\sigma^{k'}\sigma^k\sigma^{k''}\right\}$$
$$= \underbrace{\mathrm{Tr}\left\{\sigma^k\right\}}_{=0} + \sum_{k''} m^{k''}(1-t)\underbrace{\mathrm{Tr}\left\{\sigma^k\sigma^{k''}\right\}}_{2\delta_{kk''}} + \sum_{k'} m^{k'}(t)\underbrace{\mathrm{Tr}\left\{\sigma^{k'}\sigma^k\right\}}_{2\delta_{k'k}} + \sum_{k'k''} m^{k'}(t)m^{k''}(1-t)\underbrace{\mathrm{Tr}\left\{\sigma^{k'}\sigma^k\sigma^{k''}\right\}}_{2i\epsilon_{k'kk''}}.$$

The final term with the Levi-Cevita symbol becomes zero, since for each k all the terms are zero. Since $m^{k'}(t)m^{k''}(1-t)$ is symmetric under the exchange $t \Leftrightarrow (1-t)$ we get the following terms

$$k = x \rightarrow \epsilon_{yxz}m^y(t)m^z(1-t) + \epsilon_{zxy}m^z(t)m^y(1-t) = -m^y(t)m^z(1-t) + m^z(t)m^y(1-t) = 0,$$
$$k = y \rightarrow \epsilon_{xyz}m^x(t)m^z(1-t) + \epsilon_{zyx}m^z(t)m^x(1-t) = m^x(t)m^z(1-t) - m^z(t)m^x(1-t) = 0,$$
$$k = z \rightarrow \epsilon_{xzy}m^x(t)m^y(1-t) + \epsilon_{yzx}m^y(t)m^x(1-t) = -m^x(t)m^y(1-t) + m^y(t)m^x(1-t) = 0.$$

Putting back the integration and additional terms we have

$$= \int_0^1 dt\, \frac{1}{8}\cosh(th)\cosh((1-t)h)\left(\sum_{k''} m^{k''}(1-t)2\delta_{kk''} + \sum_{k'} m^{k'}(t)2\delta_{k'k}\right)$$
$$= \frac{1}{4}\int_0^1 dt\, \cosh(th)\cosh((1-t)h)(m^k(1-t) + m^k(t)).$$

Plugging in the redefined time dependent fields gives

$$= \frac{1}{4}\int_0^1 dt\, \cosh(th)\cosh((1-t)h)\left[\tanh((1-t)h) + \tanh(ht)\right]\frac{h^k}{h}\mathbf{x}$$
$$= \frac{1}{4}\int_0^1 dt\, \left[\cosh(th)\sinh((1-t)h) + \cosh((1-t)h)\sinh(ht)\right]\frac{h^k}{h}\mathbf{x}.$$

Integrating time-dependent parts in the first term gives

$$= \int_0^1 dx\, \cosh(ax)\sinh(a(1-x)) = \frac{1}{4}\int dx\left[(e^{ax} + e^{-ax})(e^{a(1-x)} - e^{-a(1-x)})\right]$$
$$= \frac{1}{4}\int_0^1 dx\, \left[(e^a - e^{-a(1-2x)} + e^{a(1-2x)} - e^{-a}\right] = \frac{1}{2}\sinh(a) - \frac{1}{4}\int_0^1 dx\, \left[e^{-a(1-2x)} - e^{a(1-2x)}\right]$$
$$= \frac{1}{2}\sinh(a) - \frac{1}{2}\int_0^1 dx\, \sinh(a(1-2x)) = \frac{1}{2}\sinh(a) - \frac{1}{2}\left[\frac{1}{2a}\cosh(a(1-2x))\right]_0^1$$
$$= \frac{1}{2}\sinh(a) - \frac{1}{2}\left[\frac{1}{2a}\cosh(-a) - \frac{1}{2a}\cosh(a)\right] = \frac{1}{2}\sinh(a),$$

where we have used that $\rightarrow \cosh(x) = \cosh(-x)$. Integrating the second term gives

$$= \int_0^1 dx\, \cosh(a(1-x))\sinh(ax) = \frac{1}{4}\int dx\left[(e^{a(1-x)} + e^{-a(1-x)})(e^{ax} - e^{-ax})\right]$$
$$= \frac{1}{4}\int_0^1 dx\, \left[(e^a - e^{a(1-2x)} + e^{-a(1-2x)} - e^{-a}\right].$$

This is the same expression as in line two of the integration of the first term, so this integral also gives a contribution of $\frac{1}{2}\sinh(a)$. Plugging this in gives

$$\mathbf{1} \rightarrow \frac{1}{4}\sinh(h)\frac{h^k}{h}\mathbf{x}. \tag{B.6}$$

Term 2

The second term has a more complicated structure, because of the σ^z matrix in front. This results in traces of 4 Pauli matrices with 4 indices,

$$\begin{aligned}
&= \mathrm{Tr}\left\{ y\sigma^z\sigma^k + \sum_{k''} m^{k''}(1-t)y\sigma^z\sigma^k\sigma^{k''} + \sum_{k'} m^{k'}(t)y\sigma^z\sigma^{k'}\sigma^k + \sum_{k'k''} m^{k'}(t)m^{k''}(1-t)y\sigma^z\sigma^{k'}\sigma^k\sigma^{k''} \right\} \\
&= y\Bigg[\underbrace{\mathrm{Tr}\left\{\sigma^z\sigma^k\right\}}_{2\delta_{zk}} + \sum_{k''} m^{k''}(1-t)\underbrace{\mathrm{Tr}\left\{\sigma^z\sigma^k\sigma^{k''}\right\}}_{2i\epsilon_{zkk''}} + \sum_{k'} m^{k'}(t)\underbrace{\mathrm{Tr}\left\{\sigma^z\sigma^{k'}\sigma^k\right\}}_{2i\epsilon_{zk'k}} \\
&+ \sum_{k'k''} m^{k'}(t)m^{k''}(1-t) \quad \underbrace{\mathrm{Tr}\left\{\sigma^z\sigma^{k'}\sigma^k\sigma^{k''}\right\}}_{2(\delta_{zk'}\delta_{kk''}-\delta_{zk}\delta_{k'k''}-\delta_{zk''}\delta_{k'k})} \Bigg].
\end{aligned}$$

This complicates matters quite a bit, since we obtain a scaling with the output y and trace over 4 Pauli matrices.

The first term is only non-zero for $k = z$.

The second and third terms are given by

$$= 2i\sum_{k''} m^{k''}(1-t)\epsilon_{zkk''} + 2i\sum_{k'} m^{k'}(t)\epsilon_{zk'k},$$

and are zero for $k = z$ by definition of the Levi-Cevita symbol and each contain a single non-zero term for $k = x, 2$. Then, we have

$$\begin{aligned}
k = x &\rightarrow 2i\left(m^2(1-t)\epsilon_{zxy} + m^2(1-t)\epsilon_{zyx}\right) = 2i\left(m^2(1-t) - m^2(t)\right) \\
k = y &\rightarrow 2i\left(m^1(1-t)\epsilon_{zyx} + m^1(1-t)\epsilon_{zxy}\right) = 2i\left(-m^1(1-t) + m^1(t)\right).
\end{aligned}$$

However, we know the resulting integral will give a $\frac{1}{2}\sinh(h)$ for both $m^k(t)$ terms, so for $k = x, 2$ these terms are zero.
The fourth term is given by

$$2\sum_{k'k''} m^{k'}(t)m^{k''}(1-t)(\delta_{zk'}\delta_{kk''} - \delta_{zk}\delta_{k'k''} + \delta_{zk''}\delta_{k'k}),$$

which we split into three parts for $k = x, y, z$. We perform the sum over k'. The underbraces specify for which indices a term is non-zero,

$$\begin{aligned}
k = x \rightarrow \sum_{k''} & 2m^x(t)[\underbrace{\delta_{zx}\delta_{xk''}}_{=0} - \underbrace{\delta_{zx}\delta_{xk''}}_{=0} + \underbrace{\delta_{zk''}\delta_{x1}}_{k''=z}]m^{k''}(1-t) \\
& + 2m^y(t)[\underbrace{\delta_{zy}\delta_{xk''}}_{=0} - \underbrace{\delta_{zx}\delta_{yk''}}_{=0} + \underbrace{\delta_{zk''}\delta_{y1}}_{=0}]m^{k''}(1-t) \\
& + 2m^z(t)[\underbrace{\delta_{zz}\delta_{xk''}}_{k''=x} - \underbrace{\delta_{zx}\delta_{zk''}}_{=0} + \underbrace{\delta_{zk''}\delta_{zx}}_{=0}]m^{k''}(1-t) \\
& = 2[m^x(t)m^z(1-t) + m^z(t)m^x(1-t)],
\end{aligned}$$

$$\begin{aligned}
k = y \rightarrow \sum_{k''} & 2m^x(t)[\underbrace{\delta_{zx}\delta_{yk''}}_{=0} - \underbrace{\delta_{zy}\delta_{xk''}}_{=0} + \underbrace{\delta_{zk''}\delta_{xy}}_{=0}]m^{k''}(1-t) \\
& + 2m^y(t)[\underbrace{\delta_{zy}\delta_{yk''}}_{=0} - \underbrace{\delta_{zy}\delta_{yk''}}_{=0} + \underbrace{\delta_{zk''}\delta_{y2}}_{k''=z}]m^{k''}(1-t) \\
& + 2m^z(t)[\underbrace{\delta_{zz}\delta_{yk''}}_{k''=y} - \underbrace{\delta_{zy}\delta_{zk''}}_{=0} + \underbrace{\delta_{zk''}\delta_{zy}}_{=0}]m^{k''}(1-t) \\
& = 2[m^y(t)m^z(1-t) + m^z(t)m^y(1-t)],
\end{aligned}$$

$$\begin{aligned}
k = z \rightarrow \sum_{k''} & 2m^x(t)[\underbrace{\delta_{zx}\delta_{zk''}}_{=0} - \underbrace{\delta_{zz}\delta_{xk''}}_{k''=x} + \underbrace{\delta_{zk''}\delta_{x3}}_{=0}]m^{k''}(1-t) \\
& + 2m^y(t)[\underbrace{\delta_{zy}\delta_{zk''}}_{=0} - \underbrace{\delta_{zz}\delta_{yk''}}_{k''=y} + \underbrace{\delta_{zk''}\delta_{y3}}_{=0}]m^{k''}(1-t) \\
& + 2m^z(t)[\underbrace{\delta_{zz}\delta_{zk''} - \delta_{zz}\delta_{zk''}}_{=0} + \underbrace{\delta_{zk''}\delta_{zz}}_{k''=z}]m^{k''}(1-t) \\
& = 2[-m^x(t)m^x(1-t) - m^y(t)m^y(1-t) + m^z(t)m^z(1-t)].
\end{aligned}$$

We need to perform the integral over the square of the fields $m^k(t)m^{k'}(1-t)$,

$$\begin{aligned}
&= \int_0^1 dt\, \frac{1}{8}\cosh(th)\cosh((1-t)h)2m^k(t)m^{k'}(1-t) \\
&= \frac{1}{4}\int_0^1 dt\ \cosh(th)\cosh((1-t)h)\tanh(ht)\tanh((1-t)h)\frac{h^k h^{k'}}{h^y} \\
&= \frac{1}{4}\int_0^1 dt\ \sinh(th)\sinh((1-t)h)\frac{h^k h^{k'}}{h^y}.
\end{aligned}$$

Integrating the time-dependent parts gives

$$\begin{aligned}
&= \int_0^1 dx\ \sinh(ax)\sinh(a(1-x)) = \frac{1}{4}\int dx\left[(e^{ax} - e^{-ax})(e^{a(1-x)} - e^{-a(1-x)})\right] \\
&= \frac{1}{2}\cosh(a) - \frac{1}{2}\int_0^1 dx\cosh(a(1-2x)) = \frac{1}{2}\cosh(a) + \frac{1}{2}[\sinh(a(1-2x))\frac{1}{2a}]_0^1 \\
&= \frac{1}{2}\cosh(a) + \frac{1}{4a}[\sinh(-a) - \sinh(a)] = \frac{1}{2}\cosh(a) - \frac{1}{2a}\sinh(a).
\end{aligned}$$

For $k = x, 2$ we have by symmetry $m^k(t)m^{k'}(1-t) = m^k(1-t)m^{k'}(t)$, so we pick up a factor 2. Putting all these things together gives the three rules:

$$\mathbf{2} \rightarrow k = x \rightarrow \frac{2y}{h^y}\left[\frac{1}{2}\cosh(h) - \frac{1}{2h}\sinh(h)\right] h^x h^z \mathbf{x},$$
$$\mathbf{2} \rightarrow k = y \rightarrow \frac{2y}{h^y}\left[\frac{1}{2}\cosh(h) - \frac{1}{2h}\sinh(h)\right] h^y h^z \mathbf{x},$$
$$\mathbf{2} \rightarrow k = z \rightarrow \frac{y}{4}\left\{1 + \frac{1}{h^y}\left[\frac{1}{2}\cosh(h) - \frac{1}{2h}\sinh(h)\right](-h^x h^x - h^y h^y + h^z h^z)\right\}\mathbf{x}.$$

Normalization

The denominator in equation B.5 is responsible for the normalization,

$$\begin{aligned}\mathrm{Tr}\left\{\Lambda e^{-H}\right\} &= \frac{1}{4}\mathrm{Tr}\left\{(1+y\sigma^z)(1+\sum_k \sigma^k m^k)\right\} = \frac{1}{4}\mathrm{Tr}\left\{(1+\sum_k \sigma^k m^k + y\sigma^z + y\sigma^z\sum_k \sigma^k m^k)\right\} \\ &= \frac{1}{4}\left(\underbrace{\mathrm{Tr}\{1\}}_{=y} + \sum_k m^k \underbrace{\mathrm{Tr}\left\{\sigma^k\right\}}_{=0} + y\underbrace{\mathrm{Tr}\{\sigma^z\}}_{=0} + y\sum_k m^k \underbrace{\mathrm{Tr}\left\{\sigma^z\sigma^k\right\}}_{2\delta_{zk}}\right) = \frac{1}{2}(1+ym^z) \\ &= \frac{1}{2}(1 + y\tanh(h)\frac{h^z}{h}) \equiv Z.\end{aligned}$$

The gradient rules

Putting together term **1** and **2**, adding the $\tanh h \frac{h^k}{h}\mathbf{x}$ term from equation B.4, adding the sum over the samples and the empirical probabilities $q(\mathbf{x})q(y|\mathbf{x})$ gives the following gradient update rules:

$$\begin{aligned}\frac{\partial \mathcal{L}_\Lambda}{\partial \mathbf{w}^x} &= \sum_{\mathbf{x}} q(\mathbf{x})q(y|\mathbf{x})\left\{\frac{1}{4Z}\left(\sinh(h)\frac{h^x}{h} + \frac{2y}{h^y}\left[\frac{1}{2}\cosh(h) - \frac{1}{2h}\sinh(h)\right]h^x h^z\right) - \tanh(h)\frac{h^x}{h}\right\}\mathbf{x}, \\ \frac{\partial \mathcal{L}_\Lambda}{\partial \mathbf{w}^y} &= \sum_{\mathbf{x}} q(\mathbf{x})q(y|\mathbf{x})\left\{\frac{1}{4Z}\left(\sinh(h)\frac{h^y}{h} + \frac{2y}{h^y}\left[\frac{1}{2}\cosh(h) - \frac{1}{2h}\sinh(h)\right]h^y h^z\right) - \tanh(h)\frac{h^y}{h}\right\}\mathbf{x}, \\ \frac{\partial \mathcal{L}_\Lambda}{\partial \mathbf{w}^z} &= \sum_{\mathbf{x}} q(\mathbf{x})q(y|\mathbf{x})\left\{\frac{1}{4Z}\left(\sinh(h)\frac{h^z}{h} + y\left(1 + \frac{1}{h^y}\left[\frac{1}{2}\cosh(h) - \frac{1}{2h}\sinh(h)\right](-h^x h^x - h^y h^y + h^z h^z)\right)\right)\right. \\ &\quad \left. - \tanh(h)\frac{h^z}{h}\right\}\mathbf{x}.\end{aligned}$$

Appendix C

Derivation Physical System

C.1 Ising Model With Transverse Fields

Consider 2-qubit the Hamiltonian

$$\boxed{H_1 = h_1\sigma_1^x + h_2\sigma_2^x + J_{12}^z\sigma^z{}_1\sigma^z{}_2}.$$

We want to estimate the value of

$$e^{H_1} = \sum_{i=0}^{\infty}\frac{H_1^n}{n!},$$

and perform the partial trace over the second subspace,

$$\mathrm{Tr}_2\{\rho\} = \frac{1}{Z}\,\mathrm{Tr}_2\{e^{H_1}\}.$$

We need to keep a couple of things in mind. First off, we also need to estimate the value of the partition function Z up to the correct order. Only then will we see that if $J_{12} \to 0$, the approximation will be independent of variables linked to the second spin variable. We can calculate the first 5 orders in H_1,

$$\rho = \frac{1}{Z}\exp(H_1) \approx \frac{1}{Z}\left\{1 + H_1 + \frac{1}{2}H_1^2 + \frac{1}{6}{H_1}^3 + \frac{1}{14}{H_1}^4 + \frac{1}{120}{H_1}^5\right\} + O(h^6).$$

We start with H_1^2,

$$\begin{aligned}H_1^2 = {}&(h_1)^2\,(\sigma_1^x)^2 + h_1h_2\,(\sigma_2^x\sigma_1^x + \sigma_1^x\sigma_2^x) + h_1J_{12}^z\,(\sigma_1^x\sigma^z{}_1\sigma^z{}_2 + \sigma^z{}_1\sigma^z{}_2\sigma_1^x) + (h_2)^2\,(\sigma_2^x)^2\\ &+ h_2J_{12}^z\,(\sigma_2^x\sigma^z{}_1\sigma^z{}_2 + \sigma^z{}_1\sigma^z{}_2\sigma_2^x) + (J_{12}^z)^2\,(\sigma^z{}_1)^2{\sigma^z}_2^2.\end{aligned}$$

Using that

$$\{\sigma_i^a, \sigma_i^b\} = 2\delta_{ab}I, \tag{C.1}$$
$$(\sigma_i^a)^2 = I, \tag{C.2}$$
$$[\sigma_i^a, \sigma_i^b] = 2i\epsilon_{abc}\sigma_i^c \quad \text{or} \quad \sigma_i^a\sigma_i^b = i\epsilon_{abc}\sigma^c, \tag{C.3}$$
$$[\sigma_i^a, \sigma_j^b] = 0, \tag{C.4}$$

we can simplify this greatly,

$$\boxed{H_1^2 = c + 2h_1h_2\sigma_1^x\sigma_2^x},$$

with $c = (J_{12}^z)^2 + (h_1)^2 + (h_2)^2$. For ${H_1}^3 = H_1^2 H_1$:

$$\begin{aligned} H_1^2 H_1 &= \left(c + 2h_1 h_2 \sigma_1^x \sigma_2^x\right) H_1 \\ &= cH_1 + \left(2h_1^2 h_2 \sigma_2^x + 2h_1 h_2^2 \sigma_1^x + 2J_{12} h_1 h_2 \sigma_1^x \sigma_2^x {\sigma^z}_1 {\sigma^z}_2\right) \\ &= cH_1 + 2\left(h_1^2 h_2 \sigma_2^x + h_1 h_2^2 \sigma_1^x - iJ_{12} h_1 h_2 \sigma_1^x {\sigma^z}_1 \sigma_2^y\right) \\ &= cH_1 + 2\left(h_1^2 h_2 \sigma_2^x + h_1 h_2^2 \sigma_1^x + J_{12} h_1 h_2 \sigma_1^y \sigma_2^y\right). \end{aligned}$$

So

$$\boxed{H_1^3 = cH_1 + 2\left(h_1^2 h_2 \sigma_2^x + h_1 h_2^2 \sigma_1^x + J_{12} h_1 h_2 \sigma_1^y \sigma_2^y\right)}.$$

We can look at the density matrix ρ_{red} by tracing out all the operators living in the second subspace. Remember the trace relations

$$\mathrm{Tr}\{\sigma^a\} = 0, \tag{C.5}$$

$$\mathrm{Tr}\left\{\sigma^a \sigma^b\right\} = \delta_{ab}, \tag{C.6}$$

$$\mathrm{Tr}\left\{\sigma^a \sigma^b \sigma^c\right\} = 2i\epsilon_{abc}. \tag{C.7}$$

We calculate

$$\rho_{red} = \frac{1}{Z}\,\mathrm{Tr}_2\left\{e^{H_1}\right\} \approx \frac{1}{Z}\,\mathrm{Tr}_2\left\{1 + H_1 + \frac{1}{2}H_1^2 + \frac{1}{6}{H_1}^3\right\},$$

with

$$\begin{aligned} \mathrm{Tr}_2\{H_1\} &= 2h_1\sigma_1^x, \\ \mathrm{Tr}_2\{H_1^2\} &= 2c, \\ \mathrm{Tr}_2\{{H_1}^3\} &= 2h_1\left(c + 2h_2^2\right)\sigma_1^x. \end{aligned}$$

To second order this gives

$$\mathrm{Tr}_2\{\rho\} \approx \frac{1}{Z}\left\{2 + 2h_1\sigma_1^x + \frac{1}{2}2c\right\},$$

with $Z = 4 + 2c$. Remember that $c \propto O(h^2)$. We approximate the normalization constant to order $O(h^2)$, which gives

$$\frac{1}{Z} = \frac{1}{2(2+c)} \approx \frac{1}{4}(1 - \frac{1}{2}c) + O(h^4),$$

so

$$\begin{aligned} \mathrm{Tr}_2\{\rho\} &\approx \frac{1}{4}(1 - \frac{1}{2}c)\left(2 + c + 2h_1\sigma_1^x\right) \\ &\approx \frac{1}{4}\left(2 + 2h_1\sigma_1^x\right) + O(h^3) \\ &= \frac{1}{2}\left(1 + h_1\sigma_1^x\right) + O(h^3), \end{aligned}$$

which to order $O(h^3)$ only contains terms dependent on the fields of spin 1 as we expect.

For the third order expansion we do the same,

$$\begin{aligned}\mathrm{Tr}_2\{\rho\} &\approx \frac{1}{Z}\left\{2 + 2h_1\sigma_1^x + \frac{1}{2}2c + \frac{1}{6}\left(2h_1\left(c + 2h_2^2\right)\right)\sigma_1^x\right\} \\ &= \frac{1}{Z}\left\{2 + c + 2\left(h_1 + \frac{1}{6}h_1\left(c + 2h_2^2\right)\right)\sigma_1^x\right\}.\end{aligned}$$

Which is again normalized by $Z = 4 + 2c$. Define

$$a^x = 2\left(h_1 + \frac{1}{6}h_1\left(c + 2h_2^2\right)\right),$$

and approximate the normalization constant,

$$\frac{1}{Z} = \frac{1}{2(2+c)} \approx \frac{1}{4}(1 - \frac{1}{2}c) + O(h^4) = Z_1.$$

We then calculate the full term to order $O(h^4)$,

$$\begin{aligned}\mathrm{Tr}_2\{\rho\} &= \frac{1}{4}(1 - \frac{1}{2}c)\left\{2 + c + 2\left(h_1 + \frac{1}{6}h_1\left(c + 2h_2^2\right)\right)\sigma_1^x\right\} \\ &= \frac{1}{4}\left\{2 + \left(2\left(h_1 + \frac{1}{6}h_1\left(c + 2h_2^2\right)\right) - ch_1\right)\sigma_1^x\right\} + O(h^4) \\ &= \frac{1}{4}\left\{2 + h_1\left(2 + \frac{1}{3}\left(h_1^2 + 3h_2^2 + J_{12}^2\right) - h_1^2 - h_2^2 - J_{12}^2\right)\right)\sigma_1^x\right\} + O(h^4) \\ &= \frac{1}{2}\left\{1 + h_1\left(1 - \frac{1}{3}\left[h_1^2 + J_{12}^2\right]\right)\sigma_1^x\right\} + O(h^4).\end{aligned}$$

which for $J_{12} \to 0$ is independent of spin 2. We and write the reduced density matrix as

$$\mathrm{Tr}_2\{\rho\} \approx \frac{1}{2}(1 + \frac{a^x}{Z_1}\sigma^x) + O(h^4).$$

So with this physical system we can only control the off diagonal elements of the density matrix, and not the diagonal. Flipping the x fields and z fields should allow us to learn the diagonal and thus any problem. This is under the assumption that there will not pop up any σ^z scaling in higher order terms, which seems unlikely.

The fourth order term is given by

$$H_1^4 = H_1^2 H_1^2 = (c + 2h_1h_2\sigma_1^x\sigma_2^x)(c + 2h_1h_2\sigma_1^x\sigma_2^x),$$

which is simplified to

$$\boxed{H_1^4 = (c^2 + 4(h_1h_2)^2 + 4(h_1h_2)^2)}.$$

This gives the traced out term of

$$\mathrm{Tr}_2\{H_1^4\} = 2c^2 + 8(h_1h_2)^2.$$

This term is constant and contributes only to to Z and not to the field a^x,

$$\begin{aligned}\mathrm{Tr}_2\{\rho\} &\approx \frac{1}{Z}\left(2 + 2h_1\sigma_1^x + \frac{1}{2}2c + \frac{1}{6}\left(2h_1\left(c + 2h_2^2\right)\right)\sigma_1^x + \frac{1}{24}(2c^2 + 8(h_1h_2)^2)\right) + O(h^5) \\ &= \frac{1}{Z}\left(2 + c + a^x\sigma_1^x + \frac{1}{12}c^2 + \frac{1}{3}(h_1h_2)^2)\right),\end{aligned}$$

with a^x as in equation C.1. This density matrix is normalized by $Z = 4 + 2c + \frac{1}{6}c^2 + \frac{2}{3}(h_1h_2)^2$. Approximate $1/Z$ to order $O(h^5)$,

$$\frac{1}{Z} = \frac{1}{4}\frac{1}{1 + \underbrace{\frac{1}{2}\left(c + \frac{1}{12}c^2 + \frac{1}{3}(h_1h_2)^2\right)}_{a}} \approx \frac{1}{4}\left(1 - \frac{a}{2} + \frac{a^2}{4}\right) + O(h^5).$$

Luckily, the term proportional to a^2 only contributes a single term c^2 since the other terms are at least of order $O(h^6)$. The normalization constant becomes

$$\begin{aligned}\frac{1}{Z} &\approx \frac{1}{4}\left(1 - \frac{1}{2}(c + \frac{1}{12}c^2 + \frac{1}{3}(h_1h_2)^2) + \frac{1}{4}c^2\right) + O(h^6) \\ &= \frac{1}{4}\left(\underbrace{1 - \frac{1}{2}c}_{Z_1} + \underbrace{\frac{5}{24}c^2 - \frac{1}{6}(h_1h_2)^2)}_{Z_2}\right) = \frac{1}{4}(Z_1 + Z_2)\end{aligned} \tag{C.8}$$

Continuing with $\mathrm{Tr}_2\{\rho\}$,

$$\begin{aligned}\mathrm{Tr}_2\{\rho\} &\approx \frac{1}{4}\left(1 - \frac{1}{2}c + \frac{5}{24}c^2 - \frac{1}{6}(h_1h_2)^2)\right)\left(2 + c + \frac{1}{12}c^2 + \frac{1}{3}(h_1h_2)^2 + a^x\sigma_1^x)\right) \\ &= \frac{1}{4}\left(2 + \overbrace{c + \frac{1}{12}c^2 + \frac{1}{3}(h_1h_2)^2 - c - \frac{1}{2}c^2 + \frac{10}{24}c^2 - \frac{1}{3}(h_1h_2)^2}^{=0}\right) \\ &+ \frac{1}{4}\left(1 - \frac{1}{2}c + \frac{5}{24}c^2 - \frac{1}{6}(h_1h_2)^2)\right)a^x\sigma_1^x \\ &= \frac{1}{2}\left[1 + \left(1 - \frac{1}{2}c + \frac{5}{24}c^2 - \frac{1}{6}(h_1h_2)^2)\right)\left(h_1 + \frac{1}{6}h_1\left(c + 2h_2^2\right)\right)\right] \\ &= \frac{1}{2}\left[1 + \left(h_1 + \frac{1}{6}h_1\left(c + 2h_2^2\right) - \frac{1}{2}ch_1\right)\right] + O(h^5) \\ &= \frac{1}{2}\left[1 + h_1(1 - \frac{1}{3}(h_1^2 + J_{12}^2))\right].\end{aligned}$$

This final term is independent of the field h_2. The fifth order term becomes

$$\begin{aligned}
H_1^5 =& H_1^3 H_1^2 = (cH_1 + 2\left(h_1^2 h_2 \sigma_2^x + h_1 h_2^2 \sigma_1^x + J_{12} h_1 h_2 \sigma_1^y \sigma_2^y\right))(c + 2h_1 h_2 \sigma_1^x \sigma_2^x) \\
=& c^2 H_1 + 2c\left(h_1^2 h_2 \sigma_2^x + h_1 h_2^2 \sigma_1^x + J_{12} h_1 h_2 \sigma_1^y \sigma_2^y\right) \\
&+ 4h_1 h_2 (h_1^2 h_2 \sigma_2^x \sigma_1^x \sigma_2^x + h_1 h_2^2 \sigma_1^x \sigma_1^x \sigma_2^x + J_{12} h_1 h_2 \sigma_1^y \sigma_2^y \sigma_1^x \sigma_2^x) + 2ch_1 h_2 H_1 \sigma_1^x \sigma_2^x \\
=& c^2 H_1 + 2c\left(h_1^2 h_2 \sigma_2^x + h_1 h_2^2 \sigma_1^x + J_{12} h_1 h_2 \sigma_1^y \sigma_2^y\right) \\
&+ 4h_1 h_2 (h_1^2 h_2 \sigma_1^x + h_1 h_2^2 \sigma_2^x - J_{12} h_1 h_2 \sigma_1^z \sigma_2^z) \\
&+ 2ch_1 h_2 (h_1 \sigma_2^x + h_2 \sigma_1^x - J_{12} \sigma_1^y \sigma_2^y),
\end{aligned}$$

which is simplified to

$$\boxed{H_1^5 = c^2 H_1 + 4c\left(h_1^2 h_2 \sigma_2^x + h_1 h_2^2 \sigma_1^x\right) + 4h_1 h_2 (h_1^2 h_2 \sigma_1^x + h_1 h_2^2 \sigma_2^x - J_{12} h_1 h_2 \sigma_1^z \sigma_2^z)}.$$

Taking the subtrace gives

$$\mathrm{Tr}_2\{H_1^5\} = (2c^2 h_1 + 8ch_1 h_2^2 + 8h_1^3 h_2^2)\sigma_1^x.$$

There are no new contributions to Z, so we can use the derivation from the H_1^4 term,

$$\begin{aligned}
\mathrm{Tr}_2\{\rho\} &\approx \frac{1}{Z}\left(2 + c + a^x \sigma_1^x + \frac{1}{24}(2c^2 + 8(h_1 h_2)^2) + \frac{1}{120}(2c^2 h_1 + 8ch_1 h_2^2 + 8h_1^3 h_2^2)\sigma_1^x\right) + O(h^6) \\
&= \frac{1}{Z}\left(2 + c + \frac{1}{12}c^2 + \frac{1}{3}(h_1 h_2)^2 + \underbrace{\left(a^x + \frac{1}{60}c^2 h_1 + \frac{1}{15}ch_1 h_2^2 + \frac{1}{15}h_1^3 h_2^2\right)}_{b^x}\sigma_1^x\right),
\end{aligned}$$

with a^x as in equation C.1. We also determined that the approximate form of Z contains no terms of order $O(h^5)$, so we we can again use equation C.8. We already know from the H_1^4 term that the constant part becomes 1 when multiplied with the approximate form of Z, so we will only look at the part b^x/Z,

$$\begin{aligned}
\mathrm{Tr}_2\{\rho\} &\approx \frac{1}{4}(Z_1 + Z_2)\left(a^x + \frac{1}{60}c^2 h_1 + \frac{1}{15}ch_1 h_2^2 + \frac{1}{15}h_1^3 h_2^2\right) + O(h^6) \\
&= \frac{1}{4}\left(Z_1 a^x + Z_2 a^x + Z_1\left(\frac{1}{60}c^2 h_1 + \frac{1}{15}ch_1 h_2^2 + \frac{1}{15}h_1^3 h_2^2\right)\right).
\end{aligned}$$

Multiplying Z_2 with b^x only gives terms of order $< O(h^5)$ for $Z_2 a^x$. We calculate the remaining three terms separately,

$$\begin{aligned}
Z_1 a^x &= 2(1 - \frac{1}{2}c)\left(h_1 + \frac{1}{6}h_1\left(c + 2h_2^2\right)\right) \\
&= 2h_1 + \frac{1}{3}h_1\left(c + 2h_2^2\right) - ch_1 - \frac{1}{6}ch_1\left(c + 2h_2^2\right) \\
&= 2h_1\left(1 - \frac{1}{3}\left[h_1^2 + J_{12}^2\right]\right) - \frac{1}{6}ch_1\left(c + 2h_2^2\right).
\end{aligned}$$

The first term is the same as the H_1^3 term. Since $Z_2 \propto O(h^4)$, only terms of a^x linear in h can contribute,

$$Z_2 a^x = h_1(\frac{5}{12}c^2 - \frac{1}{3}(h_1 h_2)^2).$$

The final term to the right of Z_1 is proportional to $O(h^5)$ so only the 1 in Z_1 can contribute. Combining this gives

$$Z_1\left(\frac{1}{60}c^2h_1 + \frac{1}{15}ch_1h_2^2 + \frac{1}{15}h_1^3h_2^2\right) = \frac{1}{60}c^2h_1 + \frac{1}{15}ch_1h_2^2 + \frac{1}{15}h_1^3h_2^2.$$

We collect all the positive and negative terms while scaling the denominator to $\frac{1}{60}$,

$$\begin{aligned}
&= h_1\left(\frac{25}{60}c^2 + \frac{1}{60}c^2 + \frac{4}{60}ch_2^2 + \frac{4}{60}h_1^2h_2^2 - \frac{10}{60}c^2 - \frac{20}{60}ch_2^2 - \frac{20}{60}h_1^2h_2^2\right)\\
&= h_1\frac{16}{60}\left(c^2 - ch_2^2 - h_1^2h_2^2\right) = h_1\frac{4}{15}\left(c(h_1^2 + J_{12}^2) - h_1^2h_2^2\right)\\
&= h_1\frac{4}{15}\left(h_1^4 + J_{12}^4 + h_1^2h_2^2 + 2h_1^2J_{12}^2 + 2h_2^2J_{12}^2) - h_1^2h_2^2\right)\\
&= h_1\frac{4}{15}\left(\left[h_1^2 + J_{12}^2\right]^2 + 2h_2^2J_{12}^2\right).
\end{aligned}$$

So the whole thing becomes

$$\mathrm{Tr}_2\{\rho\} \approx \frac{1}{2}\left(1 + h_1\left(1 - \frac{1}{3}\left[h_1^2 + J_{12}^2\right] + \frac{2}{15}\left(\left[h_1^2 + J_{12}^2\right]^2 + 2h_2^2J_{12}^2\right)\right)\sigma_1^x\right) + O(h^6).$$

Remember that the hyperbolic tangent is given by

$$\tanh(x) = x\left(1 - \frac{x^2}{3} + \frac{2x^4}{15}\right) + O(h^7).$$

Substituting $x = [h_1^2 + J_{12}^2]^{\frac{1}{2}}$ gives

$$\tanh\left([h_1^2 + J_{12}^2]^{\frac{1}{2}}\right) = [h_1^2 + J_{12}^2]^{\frac{1}{2}}\left(1 - \frac{1}{3}[h_1^2 + J_{12}^2] + \frac{2}{15}[h_1^2 + J_{12}^2]^2\right),$$

and multiplying with $\frac{h_1}{[h_1^2+J_{12}^2]^{\frac{1}{2}}}$ gives back the original expression, except for the term $+2h_2^2J_{12}^2$,

$$\tanh\left([h_1^2 + J_{12}^2]^{\frac{1}{2}}\right)\frac{h_1}{[h_1^2 + J_{12}^2]^{\frac{1}{2}}} = h_1\left(1 - \frac{1}{3}\left[h_1^2 + J_{12}^2\right] + \frac{2}{15}\left[h_1^2 + J_{12}^2\right]^2\right).$$

So the reduced density matrix can be written as

$$\mathrm{Tr}_2\{\rho\} \approx \frac{1}{2}\left(1 + \left(\tanh\left([h_1^2 + J_{12}^2]^{\frac{1}{2}}\right)\frac{h_1}{[h_1^2 + J_{12}^2]^{\frac{1}{2}}} + \frac{4}{15}h_2^2J_{12}^2\right)\sigma_1^x\right) + O(h^6). \quad \text{(C.9)}$$

As a final check, we note that for $J_{12} \to 0$ the full exponent factorizes

$$\exp(h_1\sigma_1^x + h_2\sigma_2^x) = \exp(h_1\sigma_1^x)\exp(h_2\sigma_2^x).$$

Taking the trace gives

$$\begin{aligned}\mathrm{Tr}_2\{\exp(h_1\sigma_1^x)\exp(h_2\sigma_2^x)\} &= \exp(h_1\sigma_1^x)\,\mathrm{Tr}_2\{\exp(h_2\sigma_2^x)\}\\ &= \exp(h_1\sigma_1^x) = \frac{1}{2}(1+\tanh|h_1|\frac{h_1}{|h_1|}\sigma_1^x),\end{aligned}$$

which is equivalent to expression C.9 for $J_{12}\to 0$. While this looks further study of the behaviour of higher order behaviour of $+2h_2^2J_{12}^2$ is required.

C.2 Adding σ_1^z

We add a term $g_1\sigma_1^z$ to the Hamiltonian of section 1

$$\boxed{H_2 = H_1 + g_1\sigma_1^z}.$$

The square term is given by

$$H_2^2 = (H_1+g_1\sigma_1^z)^2 = H_1^2 + g_1(H_1\sigma_1^z+\sigma_1^zH_1) + g_1^2.$$

The two terms of order g_1 are unknown

$$\begin{aligned}H_1\sigma_1^z &= h_1\sigma_1^x\sigma_1^z + h_2^x\sigma_1^x\sigma_1^z + J_{12}\sigma_1^z\sigma_2^z\sigma_1^z\\ \sigma_1^zH_1 &= h_1\sigma_1^z\sigma_1^x + h_2^x\sigma_1^z\sigma_1^x + J_{12}\sigma_1^z\sigma_1^z\sigma_2^z,\end{aligned}$$

using that $\{\sigma_1^x,\sigma_1^z\}=0$ we get

$$H_1\sigma_1^z+\sigma_1^zH_1 = 2h_2^x\sigma_1^z\sigma_1^x + 2J_{12}\sigma_2^z,$$

which gives

$$\boxed{H_2^2 = H_1^2 + 2g_1(h_2^x\sigma_1^z\sigma_1^x + J_{12}\sigma_2^z) + g_1^2}.$$

For the third order term we get

$$\begin{aligned}H_2^3 &= (H_1+g_1\sigma_1^z)^3 = (H_1^2+g_1(H_1\sigma_1^z+\sigma_1^zH_1)+g_1^2)(H_1+g_1\sigma_1^z)\\ H_2^3 &= H_1^3 + g_1(H_1\sigma_1^zH_1+\sigma_1^zH_1^2) + g_1^2H_1 + g_1H_1^2\sigma_1^z + g_1^2(H_1+\sigma_1^zH_1\sigma_1^z) + g_1^3\sigma_1^z.\end{aligned} \tag{C.10}$$

The unknown terms are $H_1\sigma_1^zH_1$, $\sigma_1^zH_1^2$, $H_1^2\sigma_1^z$ and $\sigma_1^zH_1\sigma^z$.

a) $\boldsymbol{H_1\sigma_1^zH_1}$
We know the part $H_1\sigma_1^z$ from the squared term,

$$\begin{aligned}H_1\sigma_1^zH_1 &= (h_1\sigma_1^x\sigma_1^z + h_2^x\sigma_2^x\sigma_1^z + J_{12}\sigma_1^z\sigma_2^z\sigma_1^z)(h_1\sigma_1^x + h_2^x\sigma_1^x + J_{12}\sigma_1^z\sigma_2^z)\\ &= h_1^2\sigma_1^x\sigma_1^z\sigma_1^x + h_2h_1\sigma_2^x\sigma_1^z\sigma_1^x + J_{12}h_1\sigma_2^z\sigma_1^x + h_1h_2\sigma_1^z\sigma_2^x + h_2^2\sigma_2^x\sigma_1^z\sigma_2^x\\ &+ J_{12}h_2\sigma_2^z\sigma_x^x + h_1J_{12}\sigma_1^x\sigma_1^z\sigma_1^z\sigma_2^z + h_2J_{12}\sigma_2^x\sigma_1^z\sigma_1^z\sigma_2^z + J_{12}^2\sigma_2^z\sigma_1^z\sigma_2^z\\ &= -h_1^2\sigma_1^z + J_{12}^2\sigma_1^z + (h_2^z)^2\sigma_1^z\\ &= (c-2h_1^2)\sigma_1^z.\end{aligned}$$

b) $\sigma_1^z H_1^2$
We know H_1^2 from the squared term of H_1,

$$\sigma_1^z H_1^2 = \sigma_1^z(c + 2h_1h_2\sigma_1^x\sigma_2^x) = c\sigma_1^z + 2h_1h_2\sigma_1^z\sigma_1^x\sigma_2^x).$$

c) $\sigma_1^z H_1^2$
Similar as the previous term,

$$H_1^2\sigma_1^z = (c + 2h_1h_2\sigma_1^x\sigma_2^x)\sigma_1^z = c\sigma_1^z + 2h_1h_2\sigma_1^x\sigma_2^x\sigma_1^z).$$

Where we see can already see that the final term is canceled by anti-commutation with the final term in 2.).

d) $\sigma_1^z H_1 \sigma^z$
Again, we know the part $H_1\sigma_1^z$ from the squared term,

$$\begin{aligned}\sigma_1^z H_1^2\sigma^z &= \sigma_1^z(h_1\sigma_1^x\sigma_1^z + h_2^x\sigma_2^x\sigma_1^z + J_{12}\sigma_1^z\sigma_2^z\sigma_1^z)\\ &= h_1\sigma_1^z\sigma_1^x\sigma_1^z + h_2^x\sigma_2^x + J_{12}\sigma_2^z\sigma_1^z.\end{aligned}$$

Substituting all these terms in equation C.10 gives

$$\boxed{H_2^3 = H_1^3 + g_1((c - 2h_2^2)\sigma_1^z + \sigma_1^z + 2c\sigma_1^z) + g_1^2(3H_1 - h_1\sigma_1^x - h_1\sigma_1^x) + g_1^3\sigma_1^z}.$$

We take the trace of each order of H, which gives

$$\begin{aligned}\mathrm{Tr}_2\{H_2\} &= 2h_1\sigma_1^x + 2g_1\sigma_1^z\\ \mathrm{Tr}_2\{H_2^2\} &= 2(c + g_1^2)\\ \mathrm{Tr}_2\{H_2^3\} &= (2h_1c + 4h_1h_2^2 + 2g_1^2h_1)\sigma_1^x + (6g_1c - 4h_1^2g_1 + 2g_1^3)\sigma_1^z.\end{aligned}$$

This gives for the third order expansion of $\mathrm{Tr}_2\{\rho\} = \mathrm{Tr}_2\{\exp(H_2)\}$ the following:

$$\begin{aligned}\mathrm{Tr}_2\{\rho\} &\approx \frac{1}{Z}\mathrm{Tr}_2\left\{1 + H_2 + \frac{1}{2}H_2^2 + \frac{1}{6}H_2^3\right\} + O(h^4)\\ &= \frac{1}{Z}\Bigg(2 + c + 2g_1\sigma_1^z + \frac{1}{2}(2c + 2g_1^2) + \frac{1}{6}(2h_1c + 4h_1h_2^2 + 2g_1^2h_1)\sigma_1^x\\ &+ \frac{1}{6}(6g_1c - 4h_1^2g_1 + 2g_1^3)\sigma_1^z\Bigg)\\ &= \frac{1}{Z}\Bigg(2 + c + g_1^2 + \overbrace{2(h_1 + \frac{1}{12}(2h_1c + 4h_1h_2^2 + 2g_1^2h_1))}^{a^x}\sigma_1^x\\ &+ \underbrace{2(g_1 + \frac{1}{12}(6g_1c - 4h_1^2g_1 + 2g_1^3))}_{a^z}\sigma_1^z\Bigg)\\ &= \frac{1}{Z}(2 + c + g_1^2 + a^x\sigma_1^x + a^z\sigma_1^z),\end{aligned}$$

with normalization $Z = 2 + c + g_1^2$. We again approximate the normalization constant to order $O(h^4)$,

$$Z \approx \frac{1}{2}(1 - \frac{1}{2}(c + g_1^2)) + O(h^4),$$

and then calculate the approximations of the fields a^x/Z and a^z/Z,

$$\begin{aligned}\frac{a^x}{2+c+g_1^2} &\approx \frac{1}{2}(1-\frac{1}{2}(c+g_1^2))a^x \\ &= h_1(1-\frac{1}{3}\left(h_1^2+g_1^2+J_{12}^2\right)) \qquad \text{(C.11)} \\ \frac{a^z}{2+c+g_1^2} &\approx \frac{1}{2}(1-\frac{1}{2}(c+g_1^2))a^z \\ &= g_1(1-\frac{1}{3}\left(g_1^2+h_1^2\right)). \qquad \text{(C.12)}\end{aligned}$$

Both terms are independent of h_2. The term $h_1 J_{12}^2$ pops up in the a^x field due to the $\sigma_1^z\sigma_2^z$ interaction. Adding a $\sigma_1^z\sigma_2^z$ interaction will likely give a similar term in the a^z field.

Bibliography

[1] R. C. Wiersema and H. J. Kappen. "Implementing perceptron models with qubits". In: *Phys. Rev. A* 100 (2019), p. 020301.

[2] Seaton R.C. *Apollonius Rhodius: Argonautica*. 1997 edition. Harvard University Press, 1912.

[3] A. M. Turing. "I.—Computing Machinery and Intelligence". In: *Mind* LIX.236 (1950), pp. 433–460.

[4] Hebb D.O. *The Organization of Behavior: A Neuropsychological Theory*. 1st edition. John Wiley & Sons, 1949.

[5] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65 (1958), pp. 386–408.

[6] L. G. Valiant. "A Theory of the Learnable". In: *Commun. ACM* 27 (1984), pp. 1134–1142.

[7] Giuseppe Carleo et al. *Machine learning and the physical sciences*. 2019. eprint: arXiv:1903.10563.

[8] Yann LeCun et al. "Efficient BackProp". In: *Neural Networks: Tricks of the Trade*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 9–50.

[9] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 2nd ed. Pearson Education, 2003.

[10] Pedro Larrañaga et al. "Machine learning in bioinformatics". In: *Briefings in Bioinformatics* 7 (2006), pp. 86–112.

[11] Gabriel Chartrand et al. "Deep Learning: A Primer for Radiologists". In: *RadioGraphics* 37 (2017), pp. 2113–2131.

[12] Adrian A. Collister and Ofer Lahav. "ANNz: Estimating Photometric Redshifts Using Artificial Neural Networks". In: 116 (2004), pp. 345–351.

[13] Alexander Radovic et al. "Machine learning at the energy and intensity frontiers of particle physics". In: *Nature* 560 (2018), pp. 41–48.

[14] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529 (2016), pp. 484–503.

[15] *Google Translate World Lens*. https://googleblog.blogspot.com/2015/01/hallo-hola-ola-more-powerful-translate.html. Accessed: 22-04-2019.

[16] W. Ouyang and X. Wang. "Joint Deep Learning for Pedestrian Detection". In: *2013 IEEE International Conference on Computer Vision*. 2013, pp. 2056–2063.

[17] Z. Zhu et al. "Traffic-Sign Detection and Classification in the Wild". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2110–2118.

[18] Jacob Biamonte et al. "Quantum machine learning". In: *Nature* 549 (2017), pp. 195–202.

[19] Yann LeCun, Y Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521 (2015), pp. 436–44.

[20] Jie Zhou et al. "Recognition of handwritten numerals by Quantum Neural Network with fuzzy features". In: *International Journal on Document Analysis and Recognition* 2 (1999), pp. 30–36.

[21] Noriaki Kouda et al. "Qubit neural network and its learning efficiency". In: *Neural Computing & Applications* 14 (2005), pp. 114–121.

[22] R. Zhou and Q. Ding. "Quantum M-P Neural Network". In: *International Journal of Theoretical Physics* 46 (2007).

[23] M.V. Altaisky. *Quantum neural network.* arXiv:quant-ph/0107012v2. 2001.

[24] H. Xuan. "Research on Quantum Adaptive Resonance Theory Neural Network". In: *Proceedings of 2011 International Conference on Electronic Mechanical Engineering and Information Technology*. Vol. 8. Institute of Electrical and Electronics Engineers, 2011, pp. 3885–3888.

[25] Shang Fuhua. "Quantum-Inspired Neural Network with Quantum Weights and Real Weights". In: *Open Journal of Applied Sciences* Vol.05 No.10 (2015), pp. 609–617.

[26] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. "The quest for a Quantum Neural Network". In: *Quantum Information Processing* 13 (2014), pp. 2567–2586.

[27] S. K. Jeswal and S. Chakraverty. "Recent Developments and Applications in Quantum Neural Network: A Review". In: *Archives of Computational Methods in Engineering* (2018), pp. 1 –15.

[28] M. H. Amin et al. "Quantum Boltzmann Machine". In: *Phys. Rev. X* 8 (2018), p. 021050.

[29] H.J. Kappen. *Learning quantum models from quantum or classical data*. arXiv:1803.11278. 2018.

[30] N. J. Cerf and C. Adami. "Quantum extension of conditional probability". In: *Phys. Rev. A* 60 (1999), pp. 893–897.

[31] Manfred K. Warmuth and Dima Kuzmin. "A Bayesian Probability Calculus for Density Matrices". In: *Machine Learning* 78 (2009), pp. 63–101.

[32] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. 10th. Cambridge University Press, 2011.

[33] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[34] Giuseppe Carleo and Matthias Troyer. "Solving the quantum many-body problem with artificial neural networks". In: *Science* 355 (2017), pp. 602–606.

[35] Hiroki Saito and Masaya Kato. "Machine Learning Technique to Find Quantum Many-Body Ground States of Bosons on a Lattice". In: *Journal of the Physical Society of Japan* 87 (2018), p. 014001.

[36] T. L. Jacobsen, M. S. Jørgensen, and B. Hammer. "On-the-Fly Machine Learning of Atomic Potential in Density Functional Theory Structure Optimization". In: *Phys. Rev. Lett.* 120 (2018), p. 026102.

[37] Ganesh Hegde and R. Chris Bowen. "Machine-learned approximations to Density Functional Theory Hamiltonians". In: *Scientific Reports* 7 (2017), p. 42669.

[38] Juan Carrasquilla and Roger G. Melko. "Machine learning phases of matter". In: *Nature Physics* 13 (2017), 431 EP –.

[39] Hartmut Neven et al. *Training a Binary Classifier with the Quantum Adiabatic Algorithm*. arXiv:0811.0416. 2008.

[40] M. Schuld and N. Killoran. "Quantum Machine Learning in Feature Hilbert Spaces". In: *Phys. Rev. Lett.* 122 (2019), p. 040504.

[41] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. "Quantum Algorithm for Linear Systems of Equations". In: *Phys. Rev. Lett.* 103 (2009), p. 150502.

[42] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. *Quantum algorithms for supervised and unsupervised machine learning*. arXiv:1307.0411. 2013.

[43] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. "Quantum Support Vector Machine for Big Data Classification". In: *Phys. Rev. Lett.* 113 (2014), p. 130503.

[44] J. Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (2018), p. 79.

[45] David Gross et al. "Quantum State Tomography via Compressed Sensing". In: *Phys. Rev. Lett.* 105 (2010), p. 150401.

[46] Edwin Stoudenmire and David J Schwab. "Supervised Learning with Tensor Networks". In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016, pp. 4799–4807.

[47] Zeke Xie and Issei Sato. "A Quantum-Inspired Ensemble Method and Quantum-Inspired Forest Regressors". In: *Proceedings of the Ninth Asian Conference on Machine Learning*. Vol. 77. Proceedings of Machine Learning Research, 2017, pp. 81–96.

[48] S. Yang, M. Wang, and L. Jiao. "A quantum particle swarm optimization". In: *Proceedings of the 2004 Congress on Evolutionary Computation*. Vol. 1. 2004, pp. 320–324.

[49] R.C. Wiersema. *Code: PennyLane and the Quantum Log-Likelihood*. https://github.com/therooler/pennylane-qllh. 2019.

[50] G. Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2 (1989), pp. 303–314.

[51] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. "Deep Sparse Rectifier Neural Networks". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Vol. 15. Proceedings of Machine Learning Research, 2011, pp. 315–323.

[52] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)". In: *International Conference on Learning Representations*. PMLR, Nov. 2015, pp. 1–14.

[53] Sepp Hochreiter. "The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 06.02 (1998), pp. 107–116.

[54] Lieven Vandenberghe Stephen Boyd. *Convex Optimization*. 1st. Cambridge University Press, 2004.

[55] Eric A. Carlen. "Trace Inequalities and Quantum Entropy: An introductory course". In: 2009.

[56] Archimedes. "On Conoids and Spheoids". In: *The Works of Archimedes: Edited in Modern Notation with Introductory Chapters*. Cambridge University Press, 2009, 99–150.

[57] E. T. Jaynes. "Information Theory and Statistical Mechanics". In: *Phys. Rev.* 106 (1957), pp. 620–630.

[58] Peter B R Nisbet-Jones et al. "Photonic qubits, qutrits and ququads accurately prepared and delivered on demand". In: *New Journal of Physics* 15 (2013), p. 053007.

[59] Thomas L. Curtright and Cosmas K. Zachos. "Elementary Results for the Fundamental Representation of SU(3)". In: *Reports on Mathematical Physics* 76 (2015), pp. 401 –404.

[60] Reinhold A Bertlmann and Philipp Krammer. "Bloch vectors for qudits". In: *Journal of Physics A: Mathematical and Theoretical* 41 (2008), p. 235303.

[61] *David Lidzey's Advanced Electrodynamics and Magnetism course, Lecture 4*. https://www.sheffield.ac.uk/physics/teaching/phy331/magnetism. Accessed: 2019-05-29.

[62] Eric W Weisstein. *Quadratic Curve*. From MathWorld – A Wolfram Web Resource. URL: http://mathworld.wolfram.com/QuadraticCurve.html.

[63] Daniel Zwillinger. "CRC Standard Mathematical Tables and formulae". In: 31st. Chapman & Hall/CRC, 2002, 4.6 Conics.

[64] Francisco De Zela. "Closed-Form Expressions for the Matrix Exponential". In: *Symmetry* 6 (2014), pp. 329–344.

[65] Gadi Aleksandrowicz et al. *Qiskit: An Open-source Framework for Quantum Computing*. 2019.

[66] Ville Bergholm et al. *PennyLane: Automatic differentiation of hybrid quantum-classical computations*. 2018. eprint: arXiv:1811.04968.

[67] Maria Schuld et al. "Evaluating analytic gradients on quantum hardware". In: *Phys. Rev. A* 99 (2019), p. 032331.

[68] R.C. Wiersema. *Code: Implementing perceptron models with qubits*. https://github.com/therooler/qperceptron. 2019.

[69] I. Newton and R. Hooke. *Isaac Newton letter to Robert Hooke*. elecronic resource. 1675.

[70] A. Einstein. "Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt". In: *Annalen der Physik* 322 (1905), pp. 132–148.

[71] Michael A. Nielsen and Isaac L. Chuang. *Mathematische Grundlagen der Quantenmechanik*. 2014 edition. Princeton University Press, 1932.

[72] John G. Cramer. "The Curious History of Quantum Mechanics". In: *The Quantum Handshake: Entanglement, Nonlocality and Transactions*. Springer International Publishing, 2016, pp. 9–38.

[73] D. ter Haar. *The Old Quantum Theory*. Pergamon Press, 1967.

[74] Maximilian Schlosshauer. "Decoherence, the measurement problem, and interpretations of quantum mechanics". In: *Rev. Mod. Phys.* 76 (2005), pp. 1267–1305.

[75] Wayne Myrvold. "Philosophical Issues in Quantum Theory". In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Fall 2018. Metaphysics Research Lab, Stanford University, 2018.

[76] J.R. Johansson, P.D. Nation, and Franco Nori. "QuTiP 2: A Python framework for the dynamics of open quantum systems". In: *Computer Physics Communications* 184 (2013), pp. 1234 –1240.

[77] Anna Vershynina, Eric Carlen, and Elliott Lieb. "Matrix and Operator Trace Inequalities". In: *Scholarpedia* 8 (Apr. 2013), p. 30919.

[78] A. Einstein, B. Podolsky, and N. Rosen. "Can Quantum-Mechanical Description of Physical Reality Be Considered Complete?" In: *Phys. Rev.* 47 (1935), pp. 777–780.

[79] Ryszard Horodecki et al. "Quantum entanglement". In: *Rev. Mod. Phys.* 81 (2009), pp. 865–942.