

# Notes

Emil Sayahi

3 May 2023

## 1 Chapter 1 — Introduction to Superposition

**Definition 1.1.** *Classical superposition* refers to when two physical quantities are added together to make a new, third one. An example would be constructive or destructive interference with two waves.

**Definition 1.2.** On the quantum scale, objects have properties which can exist in a state of *quantum superposition*. We say that some of their properties, rather than being continuous, can have values inside a set of possible values—such properties are *quantised*. A quantum superposition refers to when a quantised property is not in one discrete state. For example, think of a coin being tossed. The coin, when it lands, can only have two values: heads or tails—this is referred to as having a *definite state*. Until it lands, however, it is both heads and tails simultaneously.

## 2 Chapter 2 — What Is a Qubit?

**Definition 2.1.** Qubits are the quantum equivalent of classical bits. They can be expressed with *Dirac notation* (eg,  $|a\rangle$  refers to a column vector,  $a$ , while  $\langle b|$  refers to a row vector,  $b$ ).

**Definition 2.2.** Quantum states in a superposition can be expressed as  $\alpha|a\rangle + \beta|b\rangle$ , where  $\alpha$  and  $\beta$  are the probability amplitudes for the  $|a\rangle$  and  $|b\rangle$  states, respectively. The *normalisation rule* is that  $\alpha^2 + \beta^2 = 1$ ; the amplitudes are the square-roots of the probabilities.

We can express the following quantum state,  $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , with the vector  $|\Psi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ , where  $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  and  $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ .

**Definition 2.3.** If you multiply a matrix,  $A$ , by its conjugate transpose,  $A^\dagger$ , and the result is a matrix with 1s in its main diagonal but 0s elsewhere (an *identity matrix*), then the original matrix ( $A$ ) is called a *unitary matrix*.

**Definition 2.4.** A *Bloch sphere* can be used to visually represent a quantum state. If an arrow in the sphere is pointing toward one of the poles, it is a definite state. Otherwise, the state is in a superposition.

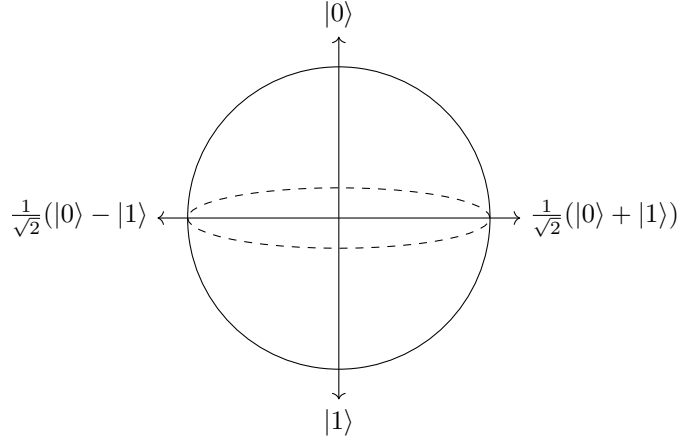


Figure 1: A Bloch sphere representation of the quantum state  $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$ .

## 2.1 Single Qubit Gates

### 2.1.1 $X$ ('NOT') Gate

This gate flips a definite state ( $|0\rangle \leftrightarrow |1\rangle$ ); if a qubit is in a superposition, then the superposition also flips.

$$\alpha|0\rangle + \beta|1\rangle \longrightarrow \boxed{X} \longrightarrow \beta|0\rangle + \alpha|1\rangle$$

Figure 2: The probability amplitudes flip.

The quantum 'NOT' gate can be represented as the following matrix:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

### 2.1.2 Hadamard Gate

The Hadamard gate takes a qubit in a definite  $|0\rangle$  or  $|1\rangle$  state, and puts it in a superposition of  $|0\rangle$  and  $|1\rangle$  states.

$$|0\rangle \longrightarrow \boxed{H} \longrightarrow \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

Figure 3: The initial definite  $|0\rangle$  state of the qubit becomes a  $|+\rangle$  superposition state.

$$|1\rangle \xrightarrow{H} \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

Figure 4: The initial definite  $|1\rangle$  state of the qubit becomes a  $|-\rangle$  superposition state.

The Hadamard gate can be represented as the following matrix:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

### 2.1.3 Z Gate

The  $Z$  gate leaves the  $|0\rangle$  state unchanged, but flips the sign of the  $|1\rangle$  state.

$$\alpha|0\rangle + \beta|1\rangle \xrightarrow{Z} \alpha|0\rangle - \beta|1\rangle$$

Figure 5: The state was flipped from a  $|+\rangle$  state to a  $|-\rangle$  state.

The  $Z$  gate can be represented as the following matrix:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

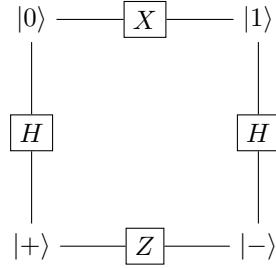


Figure 6: The effects of the  $X$ ,  $H$ , and  $Z$  gates are related.

## 2.2 Two-Qubit Gates

### 2.2.1 ‘CNOT’ (Controlled ‘NOT’) Gate

The controlled ‘NOT’ gate is used to entangle two qubits together. The gate takes in a control qubit, a target qubit, and outputs two qubits. The control qubit is untouched, but determines how the target qubit is effected; if the control

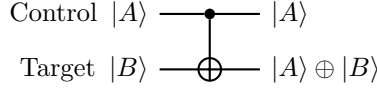


Figure 7: A quantum circuit representing the behaviour of the ‘CNOT’ gate.

qubit is  $|0\rangle$ , then the target qubit is unchanged, but if the control qubit is  $|1\rangle$ , then the target qubit is flipped ( $|0\rangle \leftrightarrow |1\rangle$ ).

The ‘CNOT’ gate can be represented as the following matrix:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

## 3 Section 2.1 — Examples of Typical Machine Learning Problems

### 3.1 Image recognition

A machine learning model can be given images as a matrix of pixels, and tasked with determining what human-recognisable objects are present in the image. In other words, the model can take in a matrix of pixels and determine the appropriate label for the pixel matrix.

### 3.2 Time series forecasting

A machine learning model can be given a series of data points recorded in consecutive time intervals, tasked with the task of forecasting future data points.

### 3.3 Hypothesis guessing

A machine learning model can be given data with some pattern or rules, which the model must determine to predict future elements of the data set. For example, given the vector  $(4 \ 16 \ 36 \ 100)$ , a model could be tasked with completing the pattern until 200.

### 3.4 Board games

In a context where a machine learning model is performing actions within an action space, the model is referred to as an *agent*. Often times, an agent can be taught to play a board game using *reinforcement learning*, where the agent is rewarded for performing positive actions and punished for performing negative actions, where ‘negative’ and ‘positive’ refer to following pre-determined rules

for how to behave, which can include how the action impacts the ultimate outcome—in the case of a board game, whether the action eventually leads to a win, loss, or, if the game allows, a draw. With this style of learning, an agent learns by trial and error.

**Definition 3.1.** *Deep learning* is a type of machine learning model, where there are several layers in a ‘neural network’, each layer having at least one ‘neurone’. All of these artificial neurones are connected to every neurone in the adjacent layers. These neurones have ‘weights’ (a tensor which is updated by training the model), ‘biases’ (a constant term which is updated by training the model), and an ‘activation function’ (which determines if the neurone ‘fires’, or is activated, within the network). With a trained neural network, the input neurons (in the input layer) have their values tensor-multiplied ( $\otimes$ ) with the weights of each connected layer, with the biases of those connected layers added to the resulting tensors. This is done with each successive layer until the output layer, where the values of each output neurone are collected.

## 4 Section 2.2 — The Three Ingredients of a Learning Problem

**Definition 4.1.** A *supervised learning task* is one in which the data set is labelled, and the task is to predict the appropriate label for a new, unclassified input.

**Definition 4.2.** An *unsupervised learning task* is one in which the data set is not labelled, and the task is to determine the appropriate classes present in the data set, and the elements which belong to each class.

### 4.0.1 Subsection 2.2.1 — Data

**Definition 4.3.** A machine learning model, in a supervised learning context, attempts to minimise its mistakes; its accuracy can be modelled with another function, called a *loss function*.

In the context of unsupervised learning, clustering of elements in a data set is done using the distance between the elements.

**Definition 4.4.** *Feature selection* refers to the process of determining what variables, predictors, or features are relevant in constructing the appropriate model.

A data set that has biases within it will result in models that reflect those same biases.

**Definition 4.5.** A *classification task* is concerned with labelling data elements with the class that they belong to.

**Definition 4.6.** A *regression task* is concerned with fitting a model to a data set, so that new data elements can be predicted.

#### 4.0.2 Subsection 2.2.2 — Model

**Definition 4.7.** A *model* is something which can be used to approximate observed behaviour. In the context of statistics, data science, and machine learning, a model is a function which attempts to fit the behaviour of a data set.

**Definition 4.8.** A *deterministic model* is a model which provides predictions without a probability attached to it; the model is certain in its predictions, regardless of whether or not the predictions are accurate.

**Definition 4.9.** A *probabilistic model* is a model which is not certain in its predictions, but, instead, has some probability attached with its predictions.

#### 4.0.3 Subsection 2.2.3 — Loss

**Definition 4.10.** We can evaluate the performance of a model by using a *loss function*—a common metric used to define loss is accuracy ( $\text{Accuracy} = \frac{\text{Number of correctly classified examples}}{\text{Total number of examples}}$ ). Regardless of what function we use to define loss, the *error* is the complement of the loss ( $\text{Error} = 1 - \text{Loss}$ , where  $\text{Loss} = [0, 1]$ ).

## 5 2.4 — Training in Unsupervised Learning

There is no one standard approach to training in unsupervised learning, as the goal of ‘minimising loss’ is difficult to define when we cannot automatically calculate a metric such as accuracy—a model which generates images of zebras, for example, cannot have its imagery automatically evaluated at the end. There are, however, some common approaches, including *generative adversarial training* and *two-sample testing*.

**Definition 5.1.** A *generative adversarial network* is a type of unsupervised learning model, where two models are trained simultaneously: a *generator* and a *discriminator*. The generator attempts to generate data which is similar to the training data, while the discriminator attempts to distinguish between the generated data and the training data. The generator is trained to fool the discriminator, while the discriminator is trained to not be fooled by the generator. Essentially, the generator is trained to generate data which is similar to the training data, while the discriminator is trained to distinguish between the training data and the generated data.

**Definition 5.2.** *Two-sample testing* is a method of unsupervised learning, where the goal is to determine whether two data sets are drawn from the same distribution. This is done by training a model to distinguish between the two data sets, and then evaluating the accuracy of the model.