

國立政治大學理學院應用物理研究所

碩士論文

Graduate Institute of Applied Physics

College of Science

National ChengChi University

Master Thesis

機器學習識別古典及量子自旋模型相態

Identifying phases of classical and quantum spin models with
machine learning

The logo of National Chengchi University is a circular emblem. It features a central five-petaled flower shape. Inside the flower, the Chinese characters '政大' (Chengchi University) are written in a stylized font. The outer ring of the emblem contains the text 'National Chengchi University' in English. The entire logo is rendered in a light gray, semi-transparent watermark style.

林恆毅

Heng-Yi Lin

指導教授：林瑜瑋 博士

Advisor: Yu-Cheng Lin, Dr. rer. nat.

中華民國 一零九年 七月

July, 2020



致謝

首先感謝林瑜琤老師這些年的指導和鼓勵，不辭辛勞的幫助論文以及程式，並耐心的教導學業上的觀念，讓我有能力完成論文，由衷感激，也感謝許琇娟老師在我詢問問題時給予我解答，並感謝口試委員台大物理高英哲老師給我論文上寶貴的意見。感謝何政緯、郭庭愷給我很多幫忙，解決許多觀念，以及張太乙學長協助電腦及硬體相關的作業，讓我能夠順利地跑論文的數據。感謝行政人員筱嘉姐、雅淑姐在行政和生活細節上，提供我協助和建議。感謝所上老師們，所上學弟妹蕭邦、喬雯、家豪、楊于廷學姐的支持陪伴，能夠有討論的對象十分感謝你們。最後感謝父母支撐我求學，給我慰藉和支持，讓我能順利的完成學業這條路。



摘要

三角易辛 (Ising) 反鐵磁在絕對零度因幾何挫折性而不具磁性。有趣的是，具量子效應的橫向磁場可誘發易辛反鐵磁零溫基態之有序性，產生具 Z_6 對稱破缺的時鐘態；這個零溫有序態可由更強的橫向磁場或有限溫度破壞。在絕對零度，一量子臨界點區分弱場下的有序時鐘態與強場下的無序順磁態。而在有限溫度，一 Kosterlitz-Thouless 相態區隔了低溫的時鐘態及高溫的順磁態。我們以量子蒙地卡羅方法針對許多不同溫度值及橫場值產生自旋組態，接著藉機器學習技術的多層感知器和捲積神經網路訓練機器辨識自旋組態與相態的關係，再以更多的自旋組態使神經網路識別其對應的相態。上述機器學習方法可頗精確辨識古典易辛模型的簡單相態，但對我們主要考慮的三角反鐵磁相態卻無法呈現良好的辨識力。

關鍵字：

深度學習、多層感知器、捲積神經網路、三角量子反鐵磁、二維古典易辛模型



Abstract

The triangular Ising antiferromagnet has no magnetic order down to zero temperature due to geometrical frustration. Interestingly, a weak transverse field, introducing quantum fluctuations, can induce magnetic order in the triangular antiferromagnet at zero temperature, resulting in the clock phase with a broken Z_6 symmetry; this ordered clock phase can be destroyed by a strong transverse field or at finite temperature. At $T = 0$, there is a quantum critical point separating the clock phase in weak fields and a paramagnetic phase in strong fields; at finite temperature, the antiferromagnet exhibits an extended Kosterlitz-Thouless (KT) phase intervening between the clock and paramagnetic phases. We generate spin configurations of the triangular antiferromagnet at different temperatures and transverse fields by quantum Monte Carlo (QMC) simulations. We attempt to use supervised machine learning techniques via multilayer perceptrons and convolutional neural networks to classify the phases of the antiferromagnetic system, solely based on spin configurations sampled with QMC. We find that the neural network models perform the classification task with a 70% accuracy for the triangular quantum antiferromagnet, while successfully distinguishing the classical Ising states with more than 90% accuracy.

Keywords:

deep learning, multilayer perceptron, convolutional neural network, triangular quantum Ising antiferromagnet, two-dimensional classical Ising model



Contents

致謝	i
摘要	iii
Abstract	v
Contents	vii
1 緒論	1
2 自旋模型	3
2.1 二維古典易辛模型	3
2.2 三角量子易辛模型	4
3 自旋組態採樣	7
3.1 古典易辛模型採樣方法	7
3.2 量子三角反鐵磁模型採樣方法	10
4 深度學習自旋組態	13
4.1 多層感知器	15
4.2 捲積神經網路	17
4.3 古典易辛模型相態的分類	19
4.3.1 MLP 模型的學習結果	20
4.3.2 CNN 模型的學習結果	22
4.4 量子三角反鐵磁相態的分類	23

5 結論

29

參考文獻

31



Chapter 1

緒論

本論文探討機器學習演算法辨識自旋組態與相態的關係之能力。目標相態為二維方晶格易辛 (Ising) 自旋模型高溫時的順磁態及低溫鐵磁態，以及二維量子三角易辛反鐵磁不同溫度及橫場下的相態，包含順磁態、時鐘態 (clock phase) 及 Kosterlitz-Thouless (KT) 態。我們先利用蒙地卡羅 (Monte Carlo) 方法獲取上述相態的自旋組態。接著藉機器學習技術的多層感知器 (Multilayer Perceptron, MLP) 和捲積神經網路 (Convolutional Neural Network, CNN) 訓練機器辨識自旋組態與相態的關係，再以更多的自旋組態使神經網路識別其對應的相態。

本論文的架構如下：下一章我們先簡單介紹上述兩個自旋模型及其相態；第三章我們敘述獲得對應不同相態的自旋組態之方法；於第四章我們敘述多層感知器和捲積神經網路，及這些模型應用在我們自旋相態的成效。



Chapter 2

自旋模型

自旋模型 (spin models) 常用來描述磁性物質。又因模型定義簡單，且可描述許多複雜的現象及物質相態，所以不僅是在凝態物理、統計物理、高能物理領域常用的模型，亦廣泛應用於資訊科學、類神經網路、乃至於社會科學。本論文討論的兩個自旋模型及其相態分述於本章的以下兩節。

2.1 二維古典易辛模型

我們考慮二維方晶格古典易辛 (Ising) 自旋模型，由以下的哈密頓函數定義：

$$H = - \sum_{\langle i,j \rangle} \sigma_i \sigma_j, \quad (2.1)$$

其中 $\sigma_i = \pm 1$ 為座落在晶格點 i 上的「自旋值」， \sum 符號下標 $\langle i,j \rangle$ 表示 i 及 j 為相鄰晶格點。若晶格邊長為 L ，共 $N = L^2$ 自旋，自旋組態共 2^N 個。我們考慮具週期性邊界條件 (periodic boundary conditions) 的方晶格，也就是東西向及南北向各以環狀銜接起來的「甜甜圈」表面。此系統只存在兩個基態，為所有 σ 值均為 1 或均為 -1 的兩完美鐵磁態。當系統處於一有限溫度時，部分自旋將因熱擾動導致翻轉而偏離基態的組合，溫度越高則自旋組態將趨向 $\sigma_i = 1$ 及 $\sigma_i = -1$ 各半的無序狀態。

磁化量 (magnetization) 用來描述上述隨溫度改變的相態，其定義如下：

$$m = \frac{1}{N} \sum_i^N \sigma_i. \quad (2.2)$$

在溫度 T 下之平均磁化量為：

$$\langle m \rangle = \sum_{c=\{\sigma_i\}} m(c) P_B(c), \quad (2.3)$$

上述 c 指在熱平衡下某個特定自旋組態 $c = \{\sigma_i\}$ ，其出現的機率 $P_B(c)$ 為波茲曼分布 (Boltzmann distribution)：¹

$$P_B(c) = \frac{1}{Z} e^{-\frac{H(c)}{T}}, \quad (2.4)$$

而其中 Z 為使 $P_B(c)$ 滿足歸一化條件配分函數 (partition function)：

$$Z = \sum_c e^{-\frac{H(c)}{T}}. \quad (2.5)$$

當溫度由零溫持續上升，磁化量將由對應基態的 $|\langle m \rangle| = 1$ 漸漸變成 $\langle m \rangle = 0$ ，此過程稱為相變 (phase transition)。在 $N \rightarrow \infty$ 情形下，有限磁化量 $\langle m \rangle \neq 0$ 對應的相態為鐵磁態 (或有序態)，對應零磁化量的稱為無序態，而磁化量正消失的那一點為相變點，此相變點位於臨界溫度 $T_c = 2/(\ln(1 + \sqrt{2})) \approx 2.269$ 。

2.2 三角量子易辛模型

另一個本論文關注的自旋模型為二維三角量子易辛模型，以下列的哈密頓算符 (Hamiltonian) 定義於二維三角晶格 (圖 2.1(a))：

$$\hat{H} = \sum_{\langle i,j \rangle} \hat{\sigma}_i^z \hat{\sigma}_j^z - h \sum_i \hat{\sigma}_i^x \quad (2.6)$$

¹在本論文，我們將波茲曼常數設為： $k_B \equiv 1$

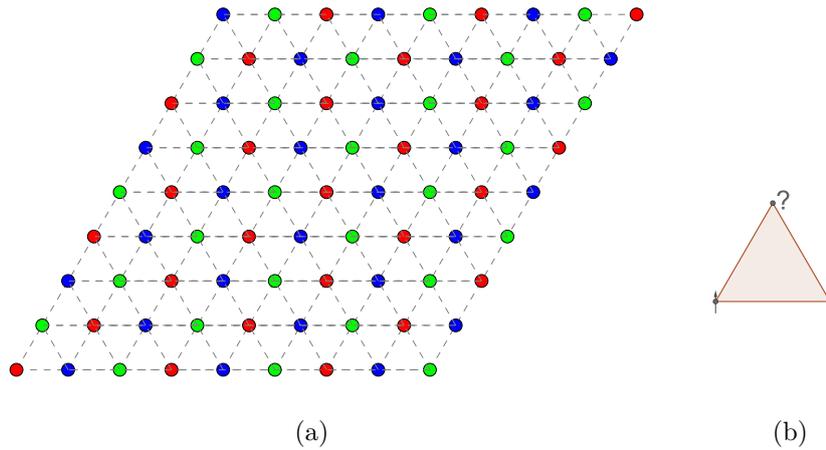


Figure 2.1: (a) 二維三角晶格。三個不同顏色分別表示三個子晶格；(b) 當自旋排列於三角晶格的角時，只有任意兩相鄰自旋可滿足反鐵磁性的排列，而第三個自旋無論向上向下均將使其與其中一相鄰自旋共同違背反鐵磁性的排列，此為挫折性。

其中 $\hat{\sigma}_i^{z,x}$ 為晶格點上的厄立矩陣 (Pauli matrices) 之 z 或 x 分量， h 為垂直 z 軸之橫場強度， \sum 符號下標 $\langle i, j \rangle$ 表示相鄰兩晶格點。若沒有外加橫場，上述模型可如同式 (2.1) 被視為古典易辛模型，因為每個自旋自由度取 $\hat{\sigma}_i^z$ 本徵值 $\sigma_i = +1$ 或 $\sigma_i = -1$ (分別代表 z 軸自旋向上或向下)。這裡 \sum 前的符號為正號，代表反鐵磁性，因為最低能量狀態傾向使每相鄰兩自旋值異號 (一為 $+1$ ，另一取 -1)；然而因為晶格的三角結構，並無法使上述的反鐵磁條件完美滿足 (見圖 2.1(b))，這個因晶格結構導致自旋排列上的衝突稱為幾何挫折性 (geometrical frustration)。三角古典易辛模型 (在無橫場下) 的幾何挫折性效應是如此的大，在任何溫度下甚至在絕對零溫時系統仍無法呈現具反鐵磁性的有序態 [1]；但有別於 $T > 0$ 的順磁態 (paramagnetic state)，於 $T = 0$ 的基態是臨界態，也就是基態自旋關聯函數隨距離呈冪次方遞減 [2]： $C(r) = \langle \sigma_i \sigma_{i+r} \rangle \propto r^{-2}$ 。

哈密頓算符 (2.6) 的橫場項使得此自旋模型呈現量子性質，因為 $[\hat{\sigma}^x, \hat{\sigma}^z] \neq 0$ ，使得哈密頓算符無法簡化成如古典易辛模型。因為 x 分量的自旋矩陣 σ^x 將翻轉 z 軸自旋，為另一可妨礙有序的反鐵磁排列的因子，不同於溫度帶來的熱擾動，橫場翻轉自旋的行為為純量子性的，也可發生於 $T = 0$ 。加上溫度及橫場的影響，三角自旋模型的相態頗為豐富，圖 2.2 呈現模型在 $h - T$ 平面的相圖；圖中兩條曲線區分了三個相態，從高溫至低溫分別為順磁相 (paramagnetic phase/PM phase)、Kosterlitz-Thouless (KT) 相，及時鐘相態 (clock phase)。其中位於半月形內的 KT 相為一臨界態 (或所謂具準長程有序，quasi-long-range order)，具隨

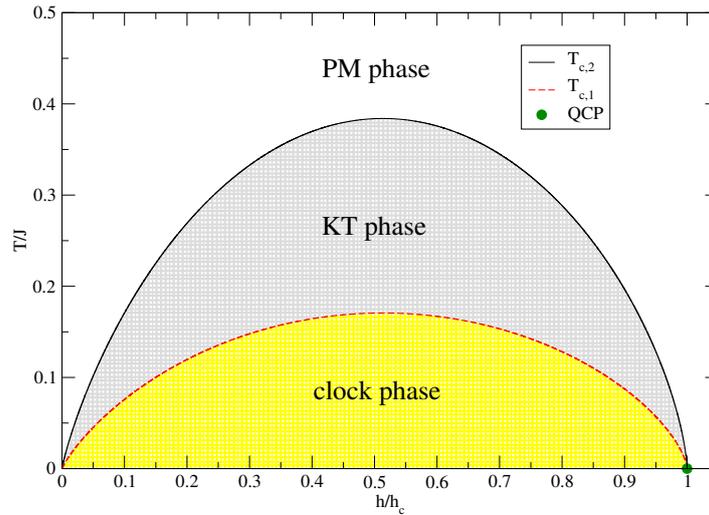


Figure 2.2: 三角量子反鐵磁之相態圖。兩彎型的相邊界 $T_{c,1}$ 及 $T_{c,2}$ 為依據式 (2.7) 畫出。兩條相邊界交會於量子臨界點 $T = 0, h/h_c = 1$ ，及 $T = 0, h = 0$ 處。

距離呈冪次方遞減的自旋關聯函數 $C(r) \propto r^{-\eta(T)}$ ，且指數 $\eta(T)$ 隨溫度變化。而時鐘態為有序態；描述這個有序態我們可先將三角晶格分成三個子晶格： A, B 及 C （見圖 2.1(a)），三個子晶格磁化量

$$m_\alpha = \langle \hat{\sigma}_{j \in \alpha}^z \rangle, \quad \alpha = A, B, C,$$

將呈現一正值 (> 0)、一負號 (< 0) 及一零的 $(+, 0, -)$ 有序排列 [2-5]。在 $T = 0$ ，時鐘態位於橫場強度小於某臨界值 h_c 的區域 ($0 < h < h_c$)，當橫場強度大於該臨界值，系統基態將經過量子相變進入無序的順磁相。 $h-T$ 平面相圖 2.2 的兩條相變界依據下述公式繪出 [2]：

$$\begin{aligned} T_{c,2} &= b \frac{h}{h_c} \ln^\nu \left(\frac{h_c}{h} \right), \quad b = 0.98, \nu = \frac{2}{3}, \\ T_{c,1} &= \frac{4}{9} T_{c,2}. \end{aligned} \quad (2.7)$$

為依據一等效模型的分析及配合量子蒙地卡羅 (Quantum Monte Carlo) 的計算結果得出 [2]。

Chapter 3

自旋組態採樣

我們利用蒙地卡羅方法針對二維古典易辛模型及量子三角反鐵磁模型各相態的自旋組態作採樣，以作為機器學習的資料。以下二節分述對於古典模型的蒙地卡羅方法及量子模型的量子蒙地卡羅方法。

3.1 古典易辛模型採樣方法

在統計力學的應用，蒙地卡羅演算法主要用以作多體系統期望值的計算。在固定溫度 T 熱平衡下，某觀察量 O 的期望值為

$$\begin{aligned}\langle O \rangle &= \sum_c O(c) P_B(c), \\ &= \frac{\sum_c O(c) e^{-H(c)/T}}{\sum_c e^{-H(c)/T}}.\end{aligned}$$

蒙地卡羅方法根據 $P_B(c)$ 隨機採樣， $\langle O \rangle$ 即可以大量如此獲取的組態 $\{c_1, c_2, \dots, c_M\}$ 近似為

$$\langle O \rangle \approx \frac{1}{M} \sum_i^M O(c_i), \quad M \gg 1 \quad (3.1)$$

這就是重要性採樣 (importance sampling)。原則上我們可取 $M \rightarrow \infty$ ，使式 (3.1) 精確 (等號成立)。

為進行上述的重要性採樣，蒙地卡羅演算法建立一離散時間馬爾可夫鏈 (discrete-time Markov chain) 來漸近獲得靜態分布 (stationary distribution) $P^*(c)$ ，

使之 $P^*(c) = P_B(c)$ 。Metropolis 蒙地卡羅演算法設計具遍歷性 (ergodicity) 的馬爾可夫過程，從組態 c 到組態 c' 的轉移機率 $w_{c,c'}$ 滿足細緻平衡 (detailed balance)：

$$P_B(c)w_{c,c'} = P_B(c')w_{c',c}. \quad (3.2)$$

轉移機率 $w_{c,c'}$ 可再分解成選擇試圖轉移 $c \rightarrow c'$ 的機率 $G(c \rightarrow c')$ 及接受此轉移的機率 $A(c \rightarrow c')$ 之乘積：

$$w_{c,c'} = G(c \rightarrow c')A(c \rightarrow c'). \quad (3.3)$$

如此，細緻平衡條件 (3.2) 可改寫成：

$$\frac{A(c \rightarrow c')}{A(c \rightarrow c)} = \frac{P_B(c') G(c' \rightarrow c)}{P_B(c) G(c \rightarrow c')}. \quad (3.4)$$

在 Metropolis 演算法中，接受機率 $A(c \rightarrow c')$ 被選為：

$$A(c \rightarrow c') = \min \left(1, \frac{P_B(c') G(c' \rightarrow c)}{P_B(c) G(c \rightarrow c')} \right) \quad (3.5)$$

來滿足式 (3.4)。在 $G(c \rightarrow c') = G(c' \rightarrow c)$ 情況下，式 (3.5) 變成

$$A(c \rightarrow c') = \min \left(1, \frac{P_B(c')}{P_B(c)} \right). \quad (3.6)$$

帶入波茲曼機率分佈，得出

$$A(c \rightarrow c') = \min \left(1, e^{-(E_{c'} - E_c)/T} \right), \quad (3.7)$$

其中 $E_c = H(c)$ 為組態 c 對應的能量值。

Metropolis 演算法中組態的轉移 $c \rightarrow c'$ 常為局域自由度的更新，例如在我們的自旋系統中選取一自旋翻轉；局域更新組態造成的能量差 $E_{c'} - E_c$ 一般來說並不大，如此接受組態更新的機率也不會太小。這裡整理古典易辛模型的 Metropolis 演算法步驟：對一選取的固定溫度 T ，我們由任一初始自旋組態 c 開始，隨機選取一個自旋 σ_i 並試圖翻轉 ($\sigma_i \rightarrow -\sigma_i$)，使得系統嘗試變換到另一個組態 c' ，這

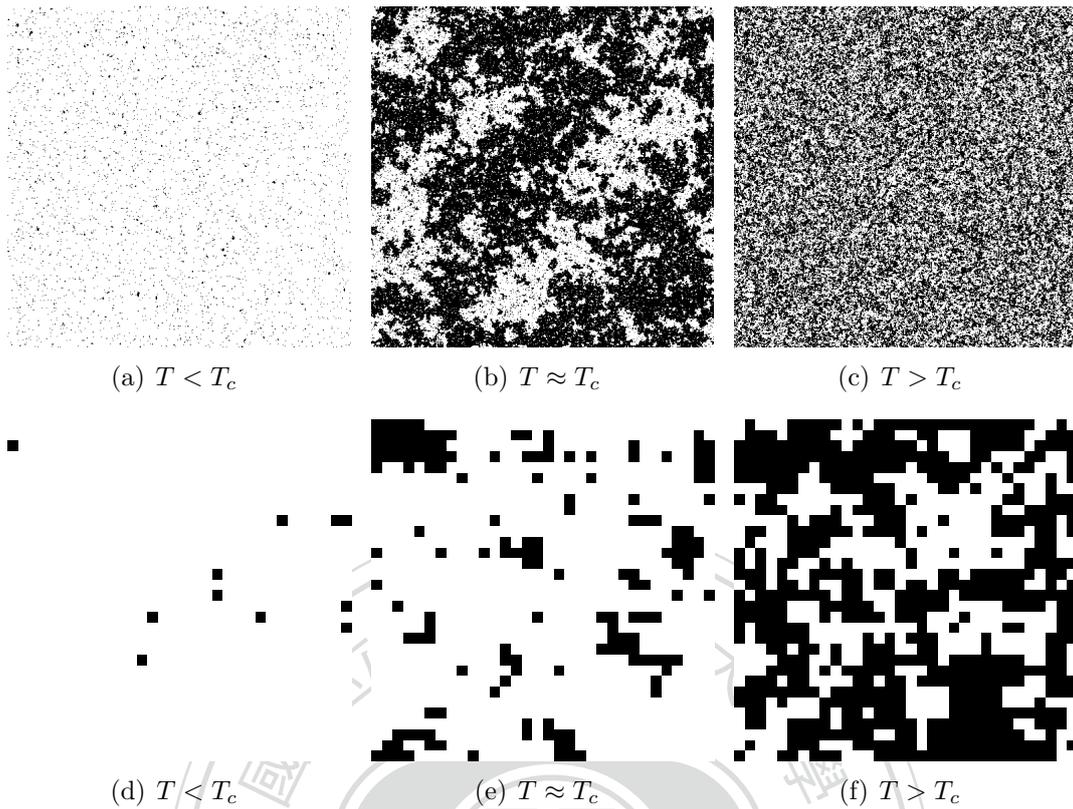


Figure 3.1: 古典易辛模型自旋組態圖，上排圖為系統尺寸 $L = 512$ ，下排圖為 $L = 32$ ，對應溫度各分別為 $T = 1.8, 2.3$ 及 3.5 。黑色色點及白色色點分別表示兩不同符號的自旋值。

個組態的改變造成的能量差為

$$E_{c'} - E_c = 2\sigma_i \sum_{j \in \text{nn}(i)} \sigma_j,$$

其中 \sum 下標 $j \in \text{nn}(i)$ 代表與自旋 σ_i 相鄰的晶格點（以二維方晶格為例，共 4 個相鄰晶格點）；利用式 (3.7) 的接受機率 $A(c \rightarrow c')$ 來判定組態 c 到組態 c' 的轉換接受與否，如果接受則 $c = c'$ ，若不接受則保持原組態。重複執行上述步驟 N 次（ N 為模型的自旋數量），整個過程被稱為一次完整的蒙地卡羅步驟（Monte Carlo sweep）。對於我們作為機器學習每一個溫度的自旋組態資料，我們捨棄預跑的 100,000 蒙地卡羅步驟來達到熱平衡，之後以每筆間隔 100 步，跑出 10,000 筆組態，而用於訓練的資料為 10,000 筆組態中的後 5,000 筆資料；其他資料則用以提供訓練後的機器作相態的辨識。我們共進行從 $T = 0.1$ 到 $T = 3.55$ 間共 70 個不同溫度值（相間隔 $\Delta T = 0.05$ ）的自旋組態採樣。

圖 3.1 展示古典易辛模型的自旋組態圖，黑色色點表示自旋 $\sigma_i = 1$ ，白色色點

表示自旋 $\sigma_i = -1$ 。以 $L = 32$ 及 $L = 512$ 為例，分別展示溫度低於臨界溫度、處於臨界溫度 ($T_c \approx 2.26$) 及高於臨界溫度的組態。觀察大尺寸系統 $L = 512$ ，可以看到溫度高於 T_c ，自旋組態的對稱性越高，黑白點均勻分佈而不形成同色的大區塊；溫度低於 T_c ，則大多數自旋朝同方向而使某一特定顏色占據組態圖的大部份面積；處於臨界溫度時，黑白兩色各形成許多不同大小的區塊。較小尺寸 $L = 32$ 的組態圖較無法呈現上述熱力學極限下不同相態的特徵，但亦可看出組態圖像在不同溫度範圍的差異。組態圖像差異明顯有利於機器學習辨識。

3.2 量子三角反鐵磁模型採樣方法

蒙地卡羅方法亦用於對量子三角反鐵磁模型作自旋組態作採樣。對於以哈密頓算符 \hat{H} 定義的量子系統，溫度 T 熱平衡下觀察量 \hat{O} 期望值為

$$\langle \hat{O} \rangle = \frac{1}{Z} \text{Tr} \left(\hat{O} e^{-\hat{H}/T} \right), \quad (3.8)$$

其中配分函數 Z 表示為

$$Z = \text{Tr} \left(e^{-\hat{H}/T} \right), \quad (3.9)$$

針對量子多體系統設計的量子蒙地卡羅 (QMC) 演算法中，所謂隨機級數展開 (Stochastic Series Expansion, SSE) QMC 方法 [6] 應屬最有效率的方法之一。SSE 方法首先將 $e^{-\hat{H}/T}$ 作級數展開：

$$e^{-\beta \hat{H}} = \sum_{n=0}^{\infty} \frac{(-\beta)^n}{n!} \hat{H}^n, \quad (3.10)$$

其中我們引入溫度倒數 $\beta = 1/T$ 。考慮將哈密頓算符 \hat{H} 改寫成局域算符 $\{\hat{H}_t\}$ 的組合：

$$\hat{H} = - \sum_{t_i} \hat{H}_{t_i}, \quad (3.11)$$

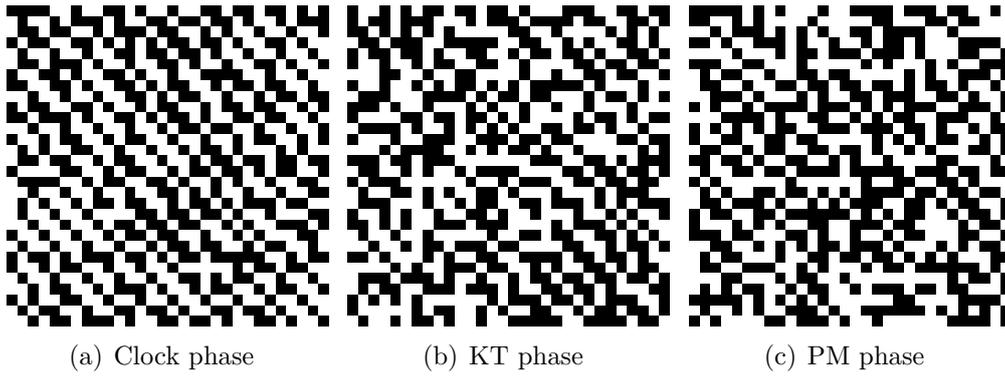


Figure 3.2: 三角反鐵磁模型 ($L = 30$) 各相態組態圖。(a) $h = 0.65, T = 0.1$ (b) $h = 1.6, T = 0.05$ (c) $h = 1.8, T = 0.2$

如此在自旋標準基底 $|x\rangle = |\sigma_1\sigma_2\cdots\sigma_N\rangle$ 下，我們可將配分函數 Z 表示為

$$\begin{aligned}
 Z &= \sum_{n=0}^{\infty} \frac{\beta^n}{n!} \sum_{\{S_n\}} \sum_{\{x\}} \langle x | \hat{H}_{t_n} \cdots \hat{H}_{t_2} \hat{H}_{t_1} | x \rangle \\
 &= \sum_{n=0}^{\infty} \frac{\beta^n}{n!} \sum_{\{S_n\}} \sum_{\{x\}} \prod_{j=1}^n \langle x' | \hat{H}_{t_j} | x \rangle
 \end{aligned} \tag{3.12}$$

其中 S_n 標示一串 n 算符乘積的組合；第二等式為於算符間插入完備基底結果（利用 $\hat{I} = \sum_{x'} |x'\rangle \langle x'|$ ）。SSE 方法隨機更新算符序列 S_n 。在計算上， n 為浮動的，為固定算符序列長度，我們可選擇 $M > n$ ，並插入 $M - n$ 個單位算符 \hat{I} 來使算符序列長度固定為 M 。考慮 \hat{I} 插入算符序列 S_M 不同位置的排列組合數，最終我們可將配合函數如下改寫：

$$Z = \sum_{\{\alpha\}} \sum_{\{S_M\}} \frac{\beta^n (M - n)!}{M!} \prod_j^M \langle x' | \hat{H}_{t_j} | x \rangle. \tag{3.13}$$

如此觀察量期望值 $\langle \hat{O} \rangle$ 在 SSE 法中可表示為

$$\langle \hat{O} \rangle = \frac{\sum_X O(X) W(X)}{\sum_X W(X)}, \quad W(X) = \frac{\beta^n (M - n)!}{M!} \prod_j^M \langle x' | \hat{H}_{t_j} | x \rangle, \tag{3.14}$$

其中 $X = \{|x\rangle, S_M\}$ 為狀態向量及算符序列的組態。

SSE 蒙地卡羅方法即根據式 (3.14) 中 $W(X)$ 隨機對算符序列 S_M 作抽樣及更新，進而更新自旋組態。針對我們的量子三角反鐵磁模型，我們除了可如 Metropolis 演算法作局域自旋組態的更新，也可大範圍的作自旋叢集的更新 [7]。

對於每對選取的溫度值與橫場值 (T, h) ，我們同樣預跑一些蒙地卡羅步驟來達平衡態，之後以每筆間隔 128 步方式蒐集 10,000 筆自旋組態作為機器學習的資料。

圖 3.2 展示以 SSE 方法獲得三角反鐵磁模型在三個相態的自旋組態樣本。考慮的系統大小為 $L = 30$ 。與前一節展示的古典易辛模型自旋組態比較，這裡量子組態似乎無法以圖像看出各相態的特徵。



Chapter 4

深度學習自旋組態

本章描述我們用以學習自旋組態的神經網路模型架構，並討論預測相態的結果。神經網路模型可視為一組安排成層狀互相連結的神經元 (neurons)。層數多的神經網路模型稱為深層神經網路，為作所謂深度學習 (deep learning) 所使用的模型 [8,9]。我們採用兩類型的神經網路模型，分別是多層感知器，及捲積神經網路。Google 所開發的 TensorFlow [10] 為廣泛被應用於深度學習的程式庫，而 Python 套件 Keras [11] 提供一便捷使用 TensorFlow 的介面。本論文的神經網路模型均藉 Keras 建構而成。

我們採監督學習 (supervised learning) 的方式讓神經網路機器辨識自旋組態所對應的相態。監督學習需要一組有標籤的訓練資料；在我們的問題，這組資料為由蒙地卡羅方法產生的自旋組態 (見第 3 章的討論)。我們將 n 組加標籤的訓練資料表示為 $\{(\mathbf{X}_\alpha, y_\alpha)\}_{\alpha=1,2,\dots,n}$ ，其中 \mathbf{X}_α 為元素值 ± 1 (代表自旋朝上或朝下) 的向量，其維度為自旋數目 N ，例如，對 $L = 30$ 的三角反鐵磁而言 \mathbf{X}_α 維度為 900； y_α 為對應 \mathbf{X}_α 的標籤。我們以慣用的 one-hot encoding 方式製標籤 [8]，對 C 類 (C classes) 問題，我們共有 C 個標籤，對編號為 α 的那筆資料，one-hot 向量形式的標籤為 $[y_{\alpha 1}, y_{\alpha 2}, \dots, y_{\alpha C}]$ ，其中

$$y_{\alpha c} = \begin{cases} 1 & \text{if } y_\alpha = c, \\ 0 & \text{otherwise.} \end{cases} \quad (4.1)$$

例如對於有三個相態的三角反鐵磁，標示 $[1, 0, 0]$ 為順磁態， $[0, 1, 0]$ 為 KT 態，

$[0, 0, 1]$ 為時鐘態。

監督學習的目標在於找到一函數 f_w 使其可將一組新的、未曾訓練過的資料 \mathbf{X}' 經由 $f_w(\mathbf{X}')$ 找出對應的標籤。描述函數 f_w 的一組參數 w 稱為權重 (weights)。判斷學習成效的一指標函數為所謂的損失函數 (loss function)，用以度量預測的標籤與真實標籤的差距；若差距小，損失函數值則小。我們以負的平均對數似然 (negative log-likelihood) 來作為與損失函數相關的欲極小化的目標函數 (代價函數) [9]：

$$\mathcal{L} = -\frac{1}{n} \sum_{\alpha=1}^n \ln[p_w(y_\alpha|\mathbf{X}_\alpha)]. \quad (4.2)$$

訓練過程，參數 w 將持續調整來極小化 \mathcal{L} 。對於分類問題，上述 \mathcal{L} 與所謂的交叉熵 (cross-entropy) 相似。若一筆資料 \mathbf{X}_α 被預測為第 c 類的機率為 $\hat{y}_{ac}(w) \equiv p_w(y_{ac} = 1|\mathbf{X}_\alpha)$ ，我們定義類別交叉熵 (categorical cross-entropy) 為 [9]

$$S = -\sum_{\alpha=1}^n \sum_{c=1}^C y_{\alpha c} \ln \hat{y}_{\alpha c} + (1 - y_{\alpha c}) \ln[1 - \hat{y}_{\alpha c}]. \quad (4.3)$$

極小化損失函數的一種方法為梯度下降法 (Gradient Descent)，也就是根據損失函數梯度方向迭代更新參數 w ：

$$w^{t+1} = w^t - \eta \nabla_w \mathcal{L}, \quad (4.4)$$

上式中的 η 為可調整的參數，稱為學習率 (learning rate)；調整 w 的方向為 \mathcal{L} 負梯度大的方向，如此可在訓練過程調整 w 流向目標函數的一個局部最小值 (local minimum)。機器學習技術中常用的最佳化 (optimization) 方法為梯度下降法的擴充版本，例如，隨機梯度下降法 (Stochastic Gradient Descent, 以下簡稱 SGD) [12] 及 Adam 演算法 (Adaptive Moment Estimation) [13]，這兩種最佳化演算法也將運用在本論文。簡單說，SGD 就是隨機挑選訓練資料的一小部分資料 (稱為 minibatch) 作為梯度的依據。SGD 可另外加上一慣性項，所謂的動量 (momentum)，當作在參數空間移動方向的記憶 [9]。加動量的 SGD 可表示為

$$\Delta w^{t+1} = \gamma \Delta w^t - \eta \nabla_w \mathcal{L}, \quad (4.5)$$

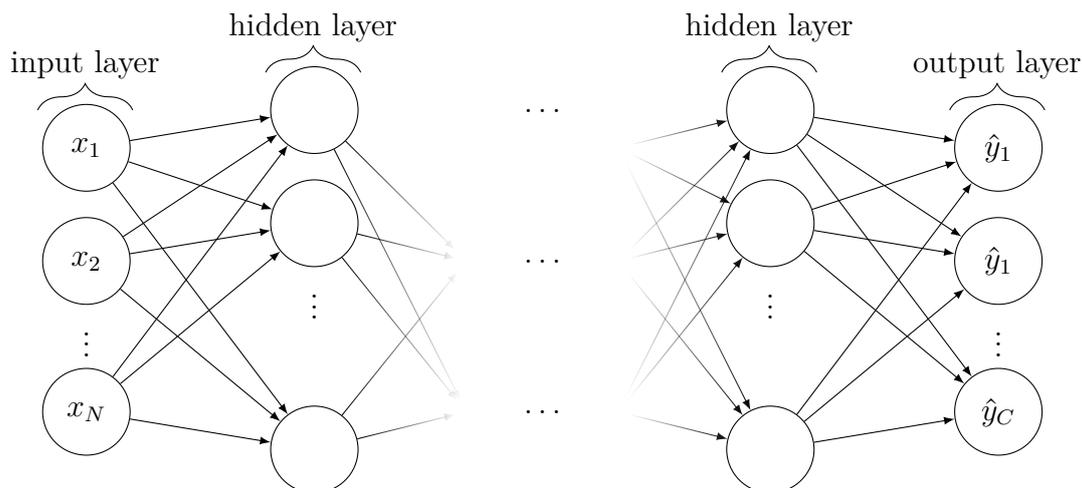


Figure 4.1: 多層感知器基本架構。

其中 $\Delta \mathbf{w}^t = \mathbf{w}^t - \mathbf{w}^{t-1}$ ，參數 γ 為動量參數。Adam 演算法主要同時考量梯度的期望值 $\langle \nabla_{\mathbf{w}} \mathcal{L} \rangle$ 及其二階矩 (second moment) $\langle (\nabla_{\mathbf{w}} \mathcal{L})^2 \rangle$ ，也因此多了相關的參數 β_1 及 β_2 。Keras 程式套件也提供上述 (及更多) 最佳化方法 (optimizer) 的選用及相關參數的設定。

神經網路模型結合以上學習指標及最佳化的技術提供有效的監督式機器學習的方法。以下兩節我們介紹我們使用的兩類型深層神經網路。

4.1 多層感知器

多層感知器 (Multilayer Perceptron) 網路，簡稱 MLP，包含輸入層、隱藏層和輸出層三層基本結構，其中至少一層隱藏層，且其數目可為任意多。每一層由不同數目的神經元組成，每層間為完全連結 (fully-connected) 的架構 (見示意圖 4.1)。以分類問題來說，輸出層的神經元數目即為類別數目 C 。資訊的傳遞是由輸入層向輸出層單向進行的，即所謂前饋神經網路 (feed-forward neural network)。

一個神經元的功能主要將接受到的資料向量與權重作純量積 (可能再加偏權值 (bias) b)，再經過非線性激活函數 (activation function) g 輸出。如此，神經元

轉換資訊的功能可表示為

$$\begin{aligned} z_j^{(\ell)} &= \sum_k w_{jk}^{(\ell)} x_k^{(\ell-1)} + b_j^{(\ell)}, \\ x_j^{(\ell)} &= g^{(\ell)}(z_j^{(\ell)}), \end{aligned} \quad (4.6)$$

這裡 $w_{jk}^{(\ell)}$ 代表連結第 $(\ell-1)$ 層第 k 個神經元及第 (ℓ) 層第 j 個神經元的權重， $b_j^{(\ell)}$ 為第 ℓ 層第 j 個神經元的偏權值， $g^{(\ell)}$ 為第 ℓ 層的激活函數。式 (4.6) 也可更簡潔地以向量形式表示為：

$$\mathbf{X}^{(\ell)} = g^{(\ell)}(\mathbf{w}^{(\ell)} \mathbf{X}^{(\ell-1)} + \mathbf{b}^{(\ell)}). \quad (4.7)$$

常用的激活函數包含 sigmoid 函數：

$$g(z) = \frac{1}{1 + e^{-z}}, \quad (4.8)$$

ReLU (rectified linear unit) 函數：

$$g(z) = \max(0, z), \quad (4.9)$$

等。對於一個 M 層感知器 (M 層前饋神經網路)，我們可將其 $f_{\mathbf{w}}(\mathbf{X})$ 的運作如下表示 [8]：

$$f_{\mathbf{w}}(\mathbf{X}) = g^{(M)}(\mathbf{w}^{(M)} \dots g^{(2)}(\mathbf{w}^{(2)} g^{(1)}(\mathbf{w}^{(1)} \mathbf{X})). \quad (4.10)$$

當神經網路的深度 M 大時，可見找尋最佳的一組大量參數的計算工程之浩大。值得一提，所謂反向傳遞演算法 (backpropagation algorithm) [14] 在深層神經網路的參數最佳化過程中扮演關鍵性的角色。基本上，反向傳遞演算法可由偏微分的連鎖律 (chain rule) 導出 [9, 15]。在 Keras 套件中反向傳遞演算法已包含於最佳化計算過程。

除了上述提到的一些神經網路基礎及最佳化技術，實務上執行深度學習其他常用到的技術還有 Dropout 及批量正規化 (Batch Normalization, 以下簡稱 BatchNorm)。Dropout 藉隨機消除部份神經元間的連結 (圖 4.2)，來避免可

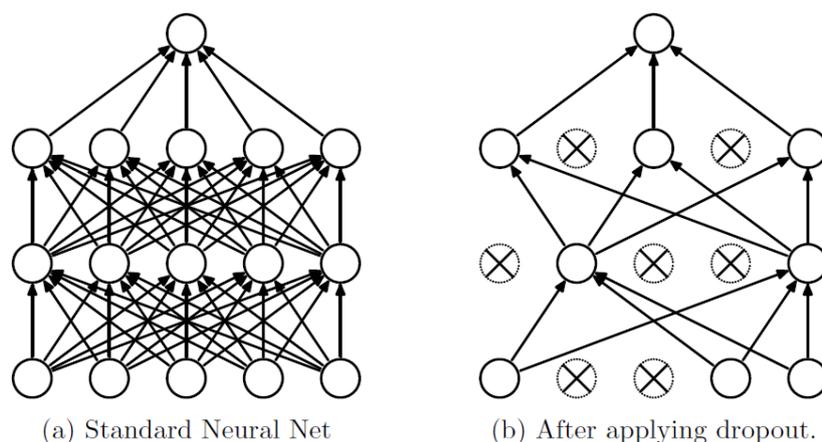


Figure 4.2: Dropout 技術的示意圖，取自 [16]。右圖 (b) 為從完全連結的神經網路中 (a) 隨機丟棄一些神經元間的連結的一個例子。

能因模型參數過多及複雜度太高而衍生的過度擬合 (overfitting) 問題 [16]。而 BatchNorm 的作用在於將輸入一層神經元待訓練的資料分小批 (minibatch) 作標準化，使其平均值為零且標準差歸一 [17]，藉以避免資訊在神經網路層層傳遞中分佈改變造成梯度消失進而導致最佳化方法沒效率 [17]，或另一被提出的解釋：可使目標函數的景觀變平滑而有利尋找極小值 [18]。

最後關於輸出層 (第 M 層)。在分類問題上，輸出層一般為 softmax 層，並有 C 個神經元，其中 C 為類別個數。如同式 (4.6) 的描述，具權重的輸入值為 $z_c^M = \sum_k w_{ck}^{(M)} x_k^{(M-1)} + b_c^{(M)}$ ，再由 softmax 激活函數轉換為 [9, 15]

$$\hat{y}_{c'} = \frac{e^{-z_{c'}^M}}{\sum_{c=1}^C e^{-z_c^M}} \quad (4.11)$$

所獲得的 $\hat{y}_{c'}$ 即代表歸屬於第 c' 類的機率；也不難看出此機率滿足 $\sum_{c'} \hat{y}_{c'} = 1$ 。

4.2 捲積神經網路

捲積神經網路 (Convolutional Neural Network, 簡稱 CNN) 具有兩個基本層，分別為捲積層 (convolution layer) 及池化層 (pooling layer)。捲積層利用捲積核 (kernel)，又稱過濾器 (filter)，對輸入的資料作捲積 (convolution) 運算；池化層則將輸入的資料「粗粒化」 (coarse-graining)，常被比擬為理論物理的重整化群 (renormalization group) 方法中的消除步驟 (decimation step) [19]。以上兩層可

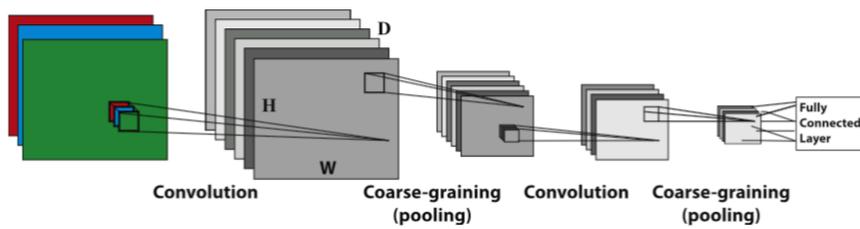


Figure 4.3: 捲積神經網路架構圖例。圖取自 [9]。

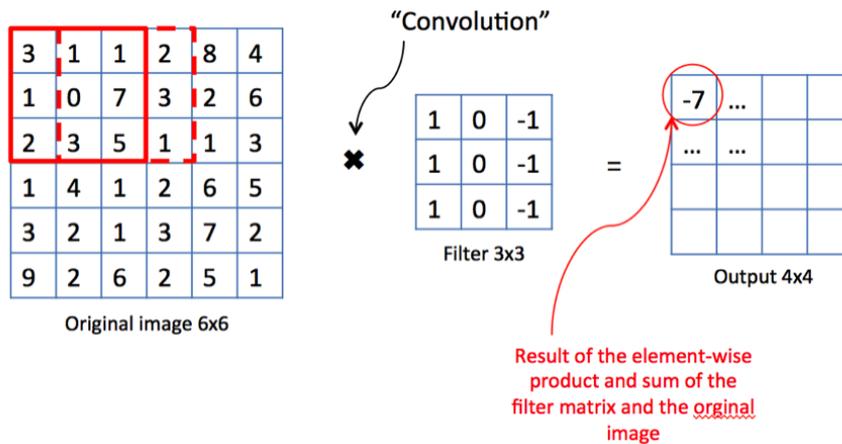


Figure 4.4: 捲積核的運算。這裡捲積核大小為 3×3 ，移動的步幅為 1，也就是每次移動一個晶格長度；虛線紅框為捲積核在實線紅框位置運作後將移動至的新位置。產生的 4×4 矩陣中的元素（如-7）為原矩陣中紅框區域的數字與捲積核同位置的數字兩兩相乘後加總所得。本捲積層沒有偏權值 $b = 0$ 。圖取自 [20]。

多次加入模型中。完成捲積及池化後，常再加完全連結層。與上節描述的多層感知器相似，捲積網路為前饋神經網路，包含一輸入層及一輸出層。這裡我們使用二維度捲積網路，故先將每筆 N 自旋組態還原成 $L \times L$ 的二維矩陣，再讀入輸入層；輸入層一般位於捲積層之前。輸出層則為 softmax 層，由 C 個神經元組成，分別代表 C 個類別。

在捲積層，我們定義捲積核的尺寸；以二維捲積網路為例，捲積核為一 $K \times K$ 的方矩陣， $K (< L)$ 一般為奇數如 $K = 1, 3, 5 \dots$ 。我們也定義不同捲積核的數目 (D)，這將產生 D 張特徵圖 (feature maps)。每一捲積核在資料平面上以一定的步幅移動，並作如圖 4.4 所示的運算，即所謂的捲積運算；我們如下表示捲積層的運算：

$$\mathbf{X}' = g(\mathbf{w} * \mathbf{X} + \mathbf{b}), \quad (4.12)$$

其中 \mathbf{w} 及 \mathbf{b} 為在該捲積層定義的共享權重及偏權值， g 為激活函數， \mathbf{X}' 為產生的特徵圖激活值。

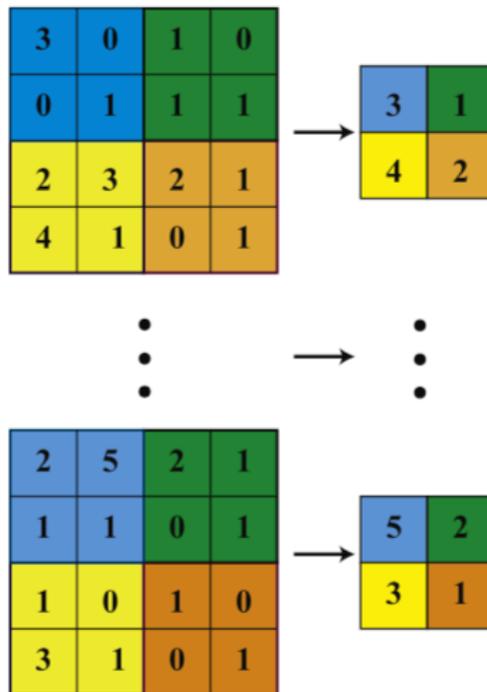


Figure 4.5: 最大池化圖示。左側特徵圖每一色塊內的 4 個激活值將由其中最大值取代，新的特徵圖（右圖）因而變小。池化將在每特徵圖獨立進行。圖取自 [9]。

捲積層之後常接池化層，其運作圖示於圖 4.5。在池化層我們將資料（圖像）切割成許多小區塊，每一區塊為如 2×2 的小方塊。我們採用如圖 4.5 所示的「最大池化」（max pooling），也就是將每一小區塊的所有激活值以其中最大值取代，如此區塊內的數個神經元將簡化成一個帶最大激活值的神經元。文獻上提到的其他池化方式包含平均池化（取平均激活值）、L2 池化（取激活值總和的平方根）等 [15]。

完全連結的神經元層常接在捲積層及池化層後（見圖 4.3 末端），這裡我們將展開（flatten）神經元成一維陣列，並可再接如上節所描述的多層感知器。

4.3 古典易辛模型相態的分類

這節我們討論使用 MLP 及 CNN 模型根據古典易辛模型的自旋組態預測所對應的相態之結果。用來預測的自旋模型之邊長為 $L = 32$ 及 $L = 128$ ，前者共 $N = 1024$ 個自旋，後者共 $N = 16384$ 個自旋。我們用以訓練的資料（自旋組態）共 $n = 175,000$ 筆，涵蓋 35 個溫度值，每個溫度值平分 5000 筆資料。用以預測（測試）的資料則有 70 筆，分別代表 70 個溫度值；這 70 筆自旋組態均未曾讓神

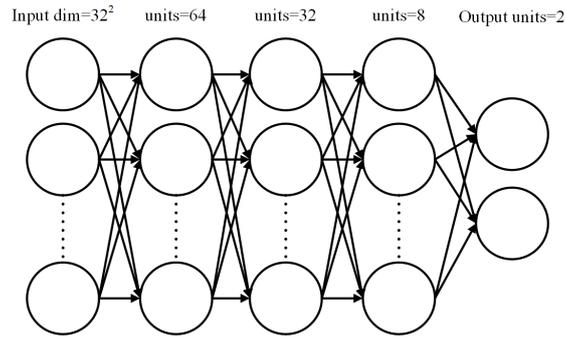


Figure 4.6: 用以訓練古典易辛模型的 MLP 架構。輸入層的神經元數目為自旋個數 L^2 (本圖以 $L = 32$ 為例)，第一層隱藏神經元數目為 64，第二層隱藏神經元數目為 32，第三層隱藏神經元數目為 8，以上激活函數都採用 ReLU。而最後輸出層為 softmax 層，神經元數目為 2，代表兩種分類。

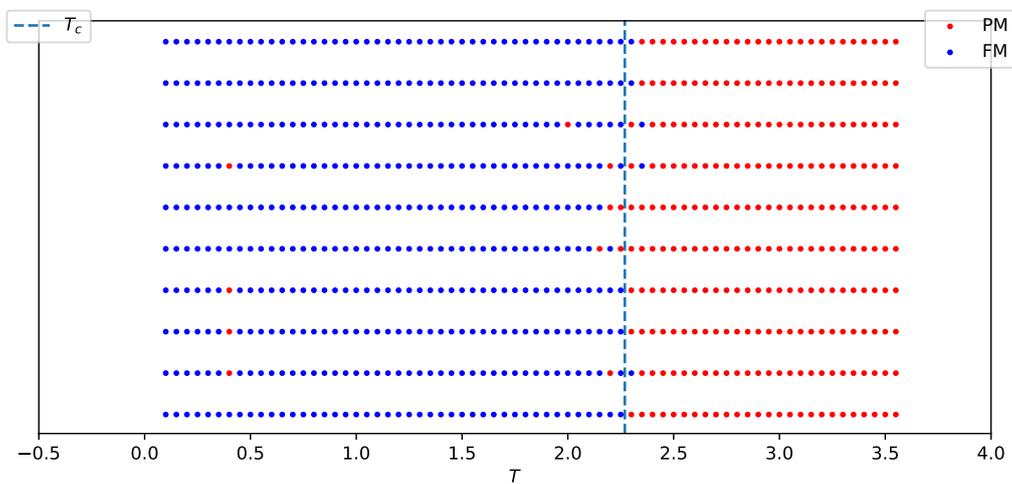


Figure 4.7: MLP 學習 $L = 32$ 古典易辛模型自旋組態的成果，共 10 次 (由上至下) 獨立的訓練及相態分類的驗證。藍色點被分類為有序態 (鐵磁態)，紅色點被分類為無序態 (順磁態)。直的虛線標示臨界溫度 $T_c = 2.269$ 位置。

經模型訓練過，也就是說，這些對神經模型而言是新資料。神經模型將僅依據自旋組態來分類相態，溫度值並不揭露給神經模型。對兩種神經模型，選擇的最佳化方法均為 Adam。

4.3.1 MLP 模型的學習結果

我們使用的 MLP 模型架構如下 (圖 4.6)：輸入層共 N 個神經元來接收輸入的 N 自旋狀態，接著三層隱藏層，分別各有 64、32 及 8 個神經元，最後的輸出層為含兩個神經元的 softmax 層。隱藏層的激活函數採用 ReLU。我們採用 one-hot 向量標籤，損失函數使用 categorical cross-entropy。用以作最佳化的方法為 Adam。

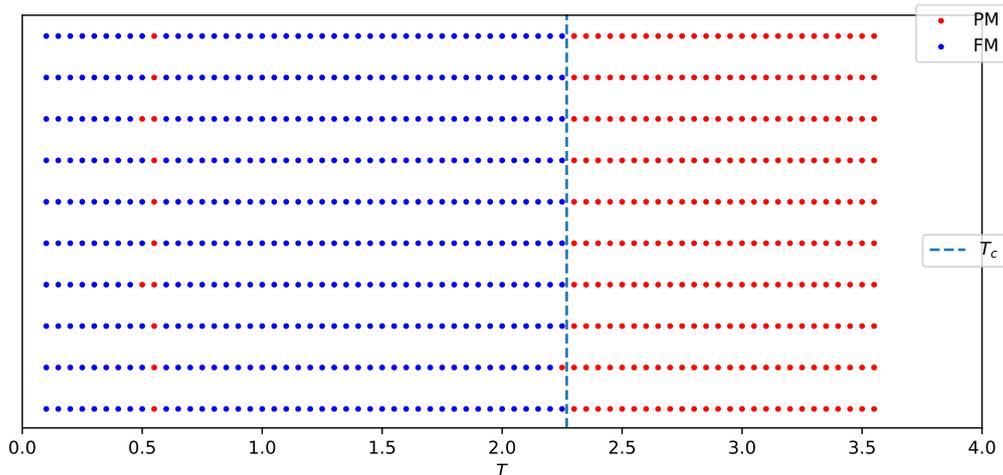


Figure 4.8: MLP 學習 $L = 128$ 古典易辛模型自旋組態的成果，共 10 次（由上至下）獨立的訓練及相態分類的驗證。藍色點被分類為有序態（鐵磁態），紅色點被分類為無序態（順磁態）。直的虛線標示臨界溫度 $T_c = 2.269$ 位置。

另外，待輸入的參數為 batch size (m) 及 epochs；其中 batch size 用以選擇每次優化的資訊量，一般我們選遠比總資料筆數 n 少的量來分批優化，即 minibatch 的概念。這裡我們設定 $m = 64$ ，如此我們共有 2734 束資料。所謂一個 epoch（期）即利用所有 $[n/m]$ 束資料來完成一次最佳化的過程。 m 值若小，資料束數目則大，一個 epoch 的最佳化過程將較耗時。

因為我們建立的 MLP 模型含有三個隱藏層，涉及的神經元數及權重參數也多；對於 $L = 32$ ，我們僅選擇一個 epoch 即觀察到很好的學習結果，驗證準確度 (validation accuracy) 高達 97%。我們將 MLP 測試結果展示在圖 4.7，利用一樣的參數 (epoch、batch size 等) 我們共進行 10 次獨立的訓練及驗證，10 次結果均展示在圖 4.7。歸類為有序態（鐵磁態）以藍色點表示，歸類為無序態（順磁態）則以藍色點表示；色點的位置依照其溫度值安排。我們可觀察到，僅有極少數錯位的色點發生在臨界溫度 T_c 附近及低溫 $T = 0.4$ 附近。在臨界點附近，自旋組態圖可能因有限尺度效應而特徵不明顯，故提高相態辨識的困難度。然而低溫相態誤判的情形，並沒有顯而易見的解釋。圖 4.8 展示大系統 $L = 128$ 的相態分類的學習結果，共 10 次結果，每次訓練期為 epochs = 10。同樣地，我們看到臨界溫度及低溫處 ($T = 0.55$ 附近) 出現零星色點錯位的情形。

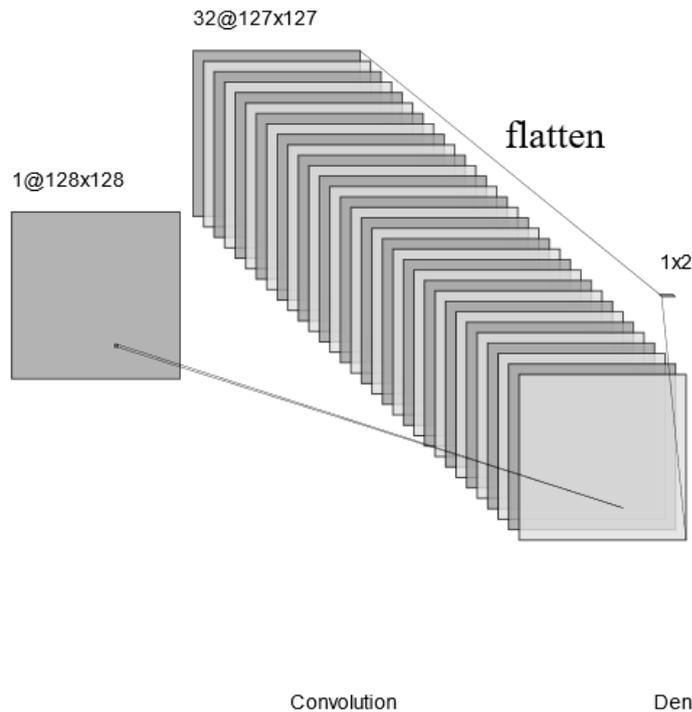


Figure 4.9: 用以訓練古典易辛模型的 CNN 架構。神經模型除輸入層及輸出層外，包含一捲積層，給定 32 個 2×2 捲積核，產生 32 張特徵圖。捲積層後展開神經元成一維陣列，直接連結到含 2 個神經元的輸出層。本例子顯示輸入的自旋組態資料維度為 $128 \times 128 \times 1$ ，表示 $L = 128$ 的自旋模型組態，輸入的組態圖為單色（黑白圖）。

4.3.2 CNN 模型的學習結果

接著我們以 CNN 模型來學習自旋組態，這裡我們使用「淺層」的二維 CNN 模型，除輸入層及輸出層外，僅含一層有 32 個 2×2 捲積核的捲積層，沒有再加池化層；神經模型架構示意於圖 4.9。

圖 4.10 及圖 4.11 分別展示對 $L = 32$ 及 $L = 128$ 的相態預測結果，對 $L = 32$ 訓練期同樣為僅 1 個 epoch，對 $L = 128$ 訓練期為 epochs = 10。我們可觀察 CNN 模型的學習結果明顯與 MLP 模型好；圖 4.10 及圖 4.11 錯位的色點僅零星出現在臨界溫度附近，而 MLP 模型在低溫處誤判的情形，在 CNN 模型的成果圖完全不出現。也就是說，以我們的古典易辛模型為例，所使用的淺層 CNN 模型優於多層的 MLP 模型。

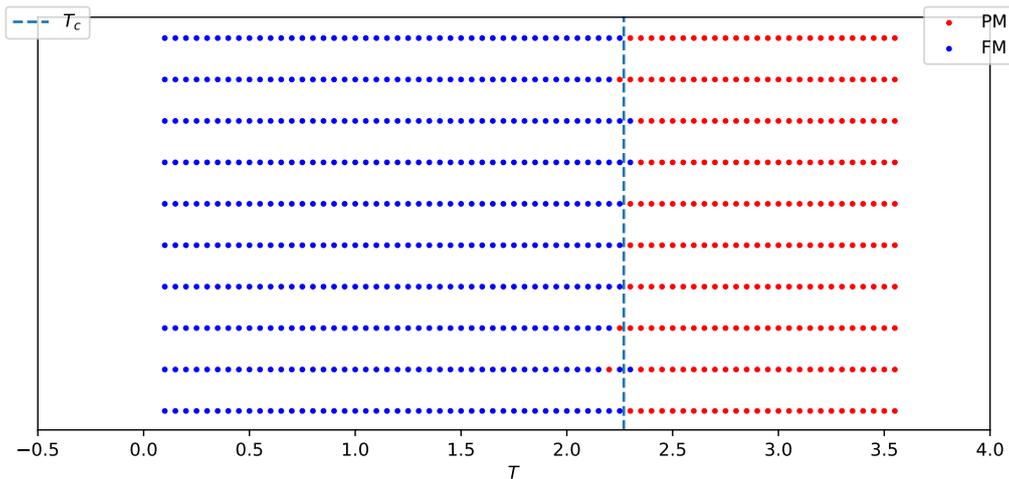


Figure 4.10: CNN 學習 $L = 32$ 古典易辛模型自旋組態的成果，共 10 次（由上至下）獨立的訓練及相態分類的驗證。藍色點被分類為有序態（鐵磁態），紅色點被分類為無序態（順磁態）。直的虛線標示臨界溫度 $T_c = 2.269$ 位置。

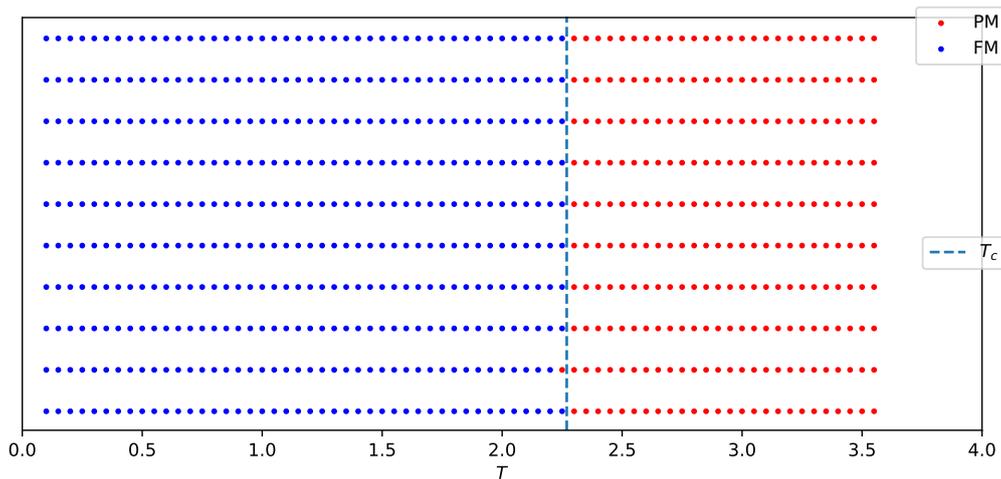


Figure 4.11: CNN 學習 $L = 128$ 古典易辛模型自旋組態的成果，共 10 次（由上至下）獨立的訓練及相態分類的驗證。藍色點被分類為有序態（鐵磁態），紅色點被分類為無序態（順磁態）。直的虛線標示臨界溫度 $T_c = 2.269$ 位置。

4.4 量子三角反鐵磁相態的分類

這節討論 MLP 模型與 CNN 模型對 $L = 30$ 量子三角反鐵磁依據自旋組態作相態分類的結果。用以進行監督學習的訓練資料為包含 320 個溫度-橫場值 $((T, h))$ 的自旋組態；每個 (T, h) 值各含 5000 筆自旋組態，也就是共 $n = 1,600,000$ 筆訓練資料。溫度-橫場值的訓練範圍標示於圖 4.12。

經多次實驗後，我們採用深層的神經網路模型。MLP 及 CNN 模型的架構分別圖示於圖 4.13 及圖 4.14。MLP 模型包含 7 個完全連結的隱藏層，依序分別有

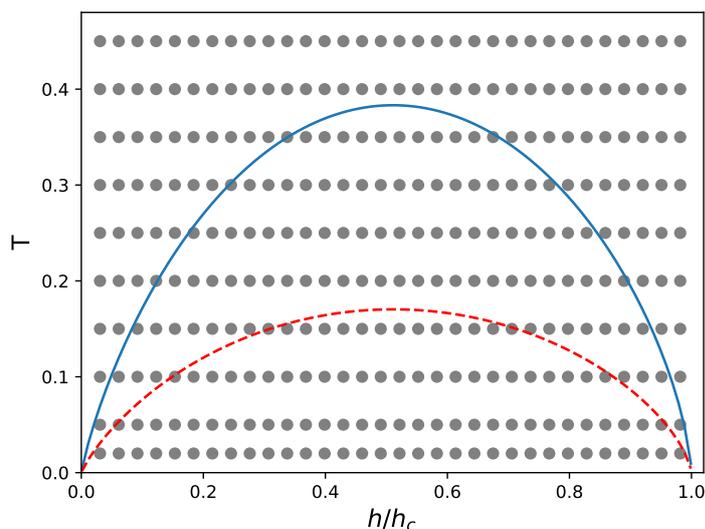


Figure 4.12: 量子三角反鐵磁模型用以訓練的溫度-橫場值範圍。灰色點標示 320 個參數點，每個參數點分別有 5000 筆自旋組態用以作訓練資料。

Input dim= 30^2 units=1024 units=512 units=256 units=128 units=64 units=32 units=16

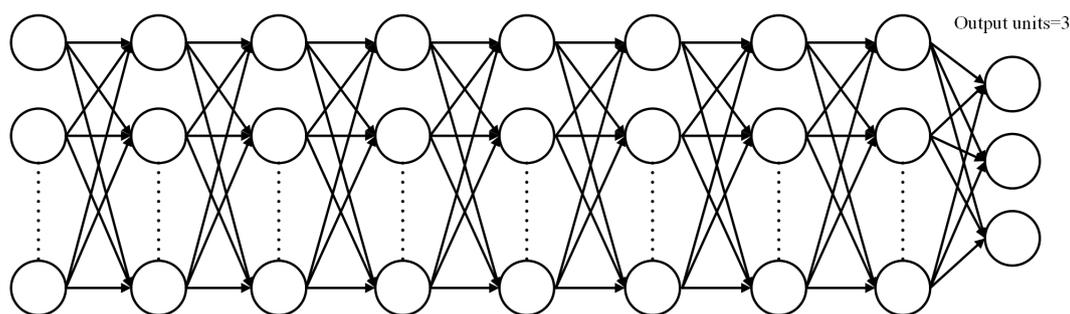


Figure 4.13: 用以訓練量子三角反鐵磁的 MLP 架構。輸入的神經元數目為 $30^2 = 900$ ，第一層隱藏神經元數目為 1024，第二層隱藏神經元數目為 512，第三層隱藏神經元數目為 256，第四層隱藏神經元數目為 128，第五層隱藏神經元數目為 64，第六層隱藏神經元數目為 32，第七層隱藏神經元數目為 32，以上激活函數都採用 ReLU，且在每一層隱藏層中加入 Dropout=0.5，用來避免過度擬合，而最後的輸出層神經元數目為 3。

1024、512、256、128、64、32 及 16 個神經元，每層均以 ReLU 作為激活函數，且每一層隱藏層中加入 Dropout=0.5，並採用 BatchNorm，最後輸出層為 3 個神經元的 softmax 層。CNN 模型則建立在三系列的捲積層及池化層，每捲積層的捲積核均為 3×3 ，池化層採 2×2 區塊的最大池化。

首先我們先就 CNN 模型比較 SGD 與 Adam 兩種最佳化方法在訓練過程的表現，圖 4.15 我們將每一個訓練期（每一個 epoch）MLP 神經模型精確度及損失（loss）作圖，比較 SGD（加動量）與 Adam 兩演算法的表現。這裡我們區別兩種精確度：訓練精確度（training accuracy）及驗證精確度（validation accuracy）。

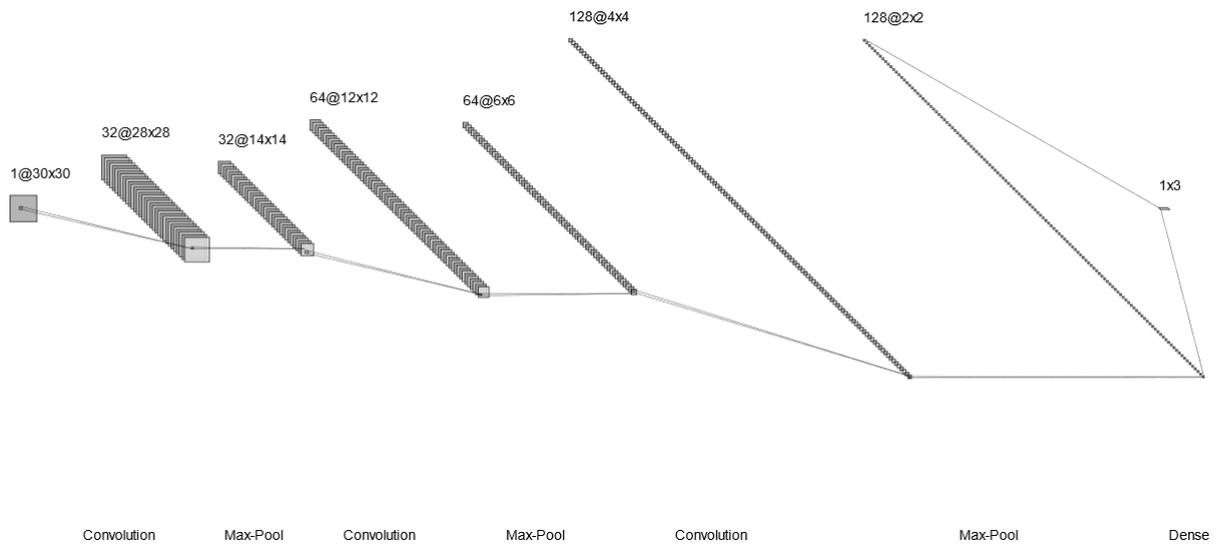


Figure 4.14: 用以訓練量子三角反鐵磁的 CNN 架構。利用三層捲積層及池化層；每捲積層的捲積核均為 3×3 ，池化層採 2×2 區塊的最大池化；除輸出層外，每層均以 ReLU 作為激活函數。

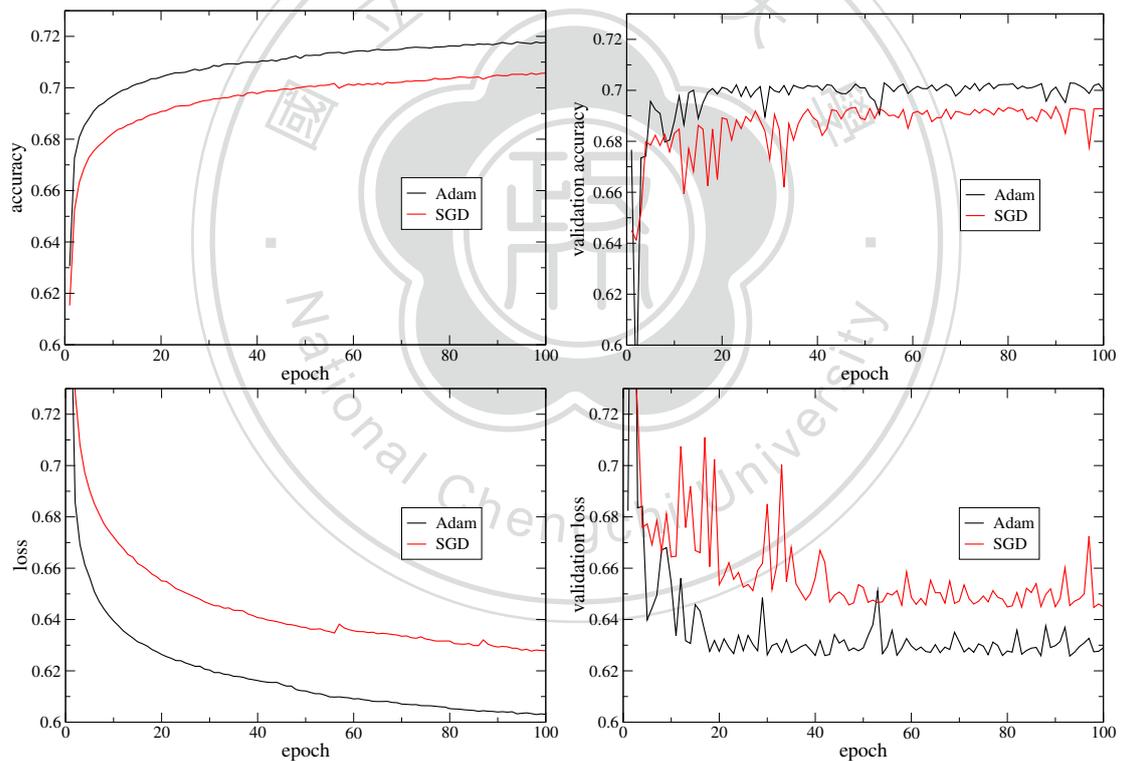


Figure 4.15: Adam 及加動量的 SGD 之精確度及損失在 CNN 模型訓練過程隨訓練期增長之變化。

我們設定 `validation_split=0.3`，表示 30% 的訓練資料將被取出作為驗證用，訓練精確度是根據保留的 70% 訓練資料所分析的精確度，而驗證精確度則是根據 30% 的驗證資料。同樣地，損失也區分出訓練損失 (training loss) 及驗證損失 (validation loss)。由 SGD 與 Adam 的比較看來，Adam 表現稍好；我們將採用 Adam 來執行後續的計算。

我們利用 3200 組測試用自旋組態來檢驗 MLP 及 CNN 模型學習後的分類相態能力。十次獨立訓練的測試結果分別呈現在圖 4.16 及圖 4.17，以三個不同顏色的色點來標示分類結果：紅色為順磁態 (PM)，綠色為 KT 態，藍色為時鐘態 (Clock)。無論是 MLP 或是 CNN 的測試結果均可觀察到不少錯位的色點，成績遠不如上一節討論的古典自旋模型。針對這個量子自旋模型，我們也嘗試其他許多不同架構的 MLP 及 CNN 模型，但尚未產生比圖 4.16 及圖 4.17 所展示的明顯更好的結果。



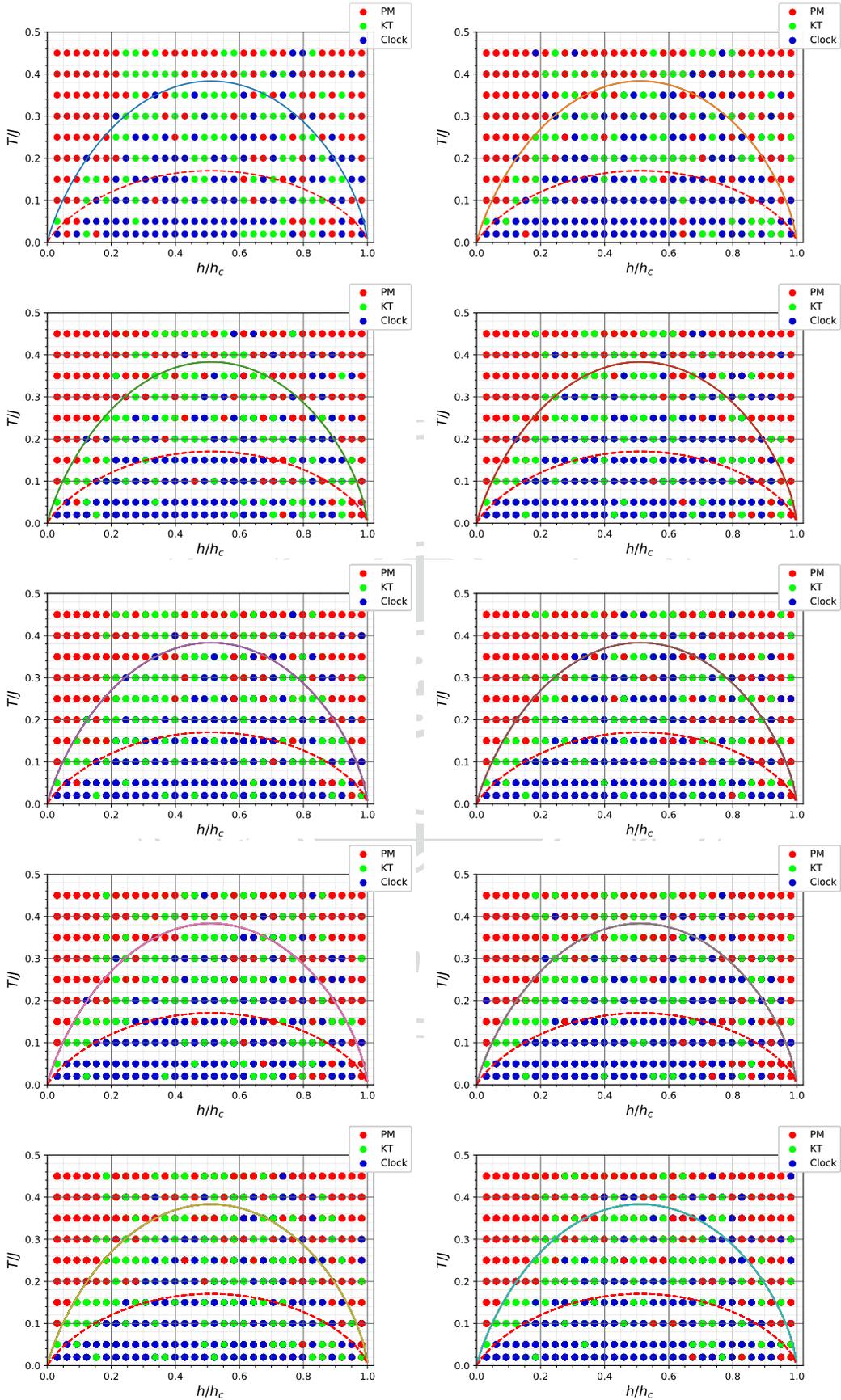


Figure 4.16: MLP 模型對量子三角反鐵磁自旋組態作分類的結果。呈現的十張圖為根據十次獨立訓練後的結果，且分別對十組包含 320 個 (T, h) 值的不同自旋組態作測試。訓練期各為 epochs = 50。

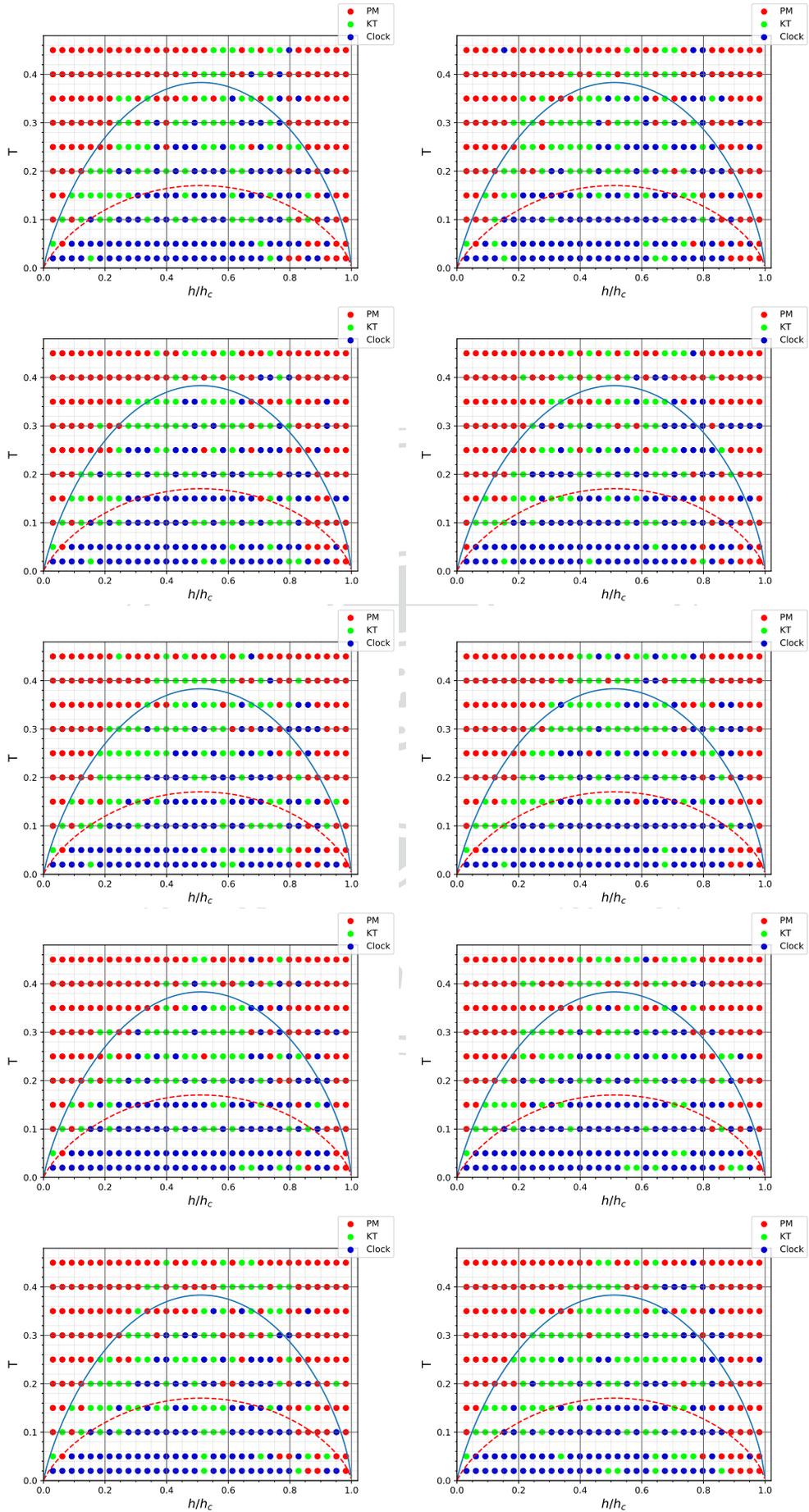


Figure 4.17: CNN 模型對十組量子三角反鐵磁自旋組態作分類的結果。訓練期各為 epochs = 50。

Chapter 5

結論

本論文測試神經網路模型根據自旋模型的組態判斷其所處相態的能力。我們檢視的自旋模型包含古典易辛模型及量子三角反鐵磁；前者自旋模型隨溫度變化僅有簡單的兩種相態：鐵磁態及順磁態，後者隨溫度與外加橫場強度變化具有三個相態：順磁態、KT 態及所謂的時鐘態。用以進行監督式學習的兩類型神經網路模型為多層感知器及捲積神經網路。

測試結果顯示，古典易辛模型因為自旋組態與相態的配對關係明顯，簡單的神經網路模型及短暫的訓練期即可達到近 100% 的正確判斷率，甚至對於沒有學習過的溫度下的組態。然而量子三角反鐵磁因其視覺上不明顯的自旋組態與相態的配對關係，加上相態種類較多，我們未能成功建立合適的神經網路模型來達到高正確判斷率。但量子模型仍獲得的 70% 精確度，意味機器學習技術不容小看，值得嘗試更多種學習方式。



參考文獻

- [1] G. H. Wannier, Phys. Rev. 79, 357 (1950).
- [2] Y. Jiang and T. Emig, Phys. Rev. B 73,104452 (2006).
- [3] S. V. Isakov and R. Moessner, Physical Review B **68** (2003).
- [4] M. Žukovič, L. Mižišin, and A. Bobák, Acta Physica Polonica A **126**, 40 (2014).
- [5] 張鎮宇, 三角晶格易辛反鐵磁之量子相變, Master's thesis, 國立政治大學, 2017.
- [6] A. W. Sandvik and J. Kurkijärvi, Phys. Rev. B **43**, 5950 (1991).
- [7] R. G. Melko, *Stochastic Series Expansion Quantum Monte Carlo*, pages 185–206, Springer, Berlin, Heidelberg, 2013.
- [8] G. Carleo et al., Rev. Mod. Phys. **91**, 045002 (2019).
- [9] P. Mehta et al., Physics Reports **810**, 1 (2019).
- [10] *TensorFlow*, <https://www.tensorflow.org/>.
- [11] *Keras*, <https://keras.io/>.
- [12] L. Bottou, Stochastic gradient descent tricks, in *Neural networks: Tricks of the trade*, pages 421–436, Springer, 2012.
- [13] D. P. Kingma and J. Ba, arXiv: **1412.6980** (2014).
- [14] D. E. Rumelhart and D. Zipser, Cognitive science **9**, 75 (1985).

- [15] M. A. Nielsen, *Neural networks and deep learning*, Determination press San Francisco, CA, 2015.
- [16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, *J. Mach. Learn. Res.* **15**, 1929 (2014).
- [17] S. Ioffe and C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 448, JMLR.org, 2015.
- [18] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, How does batch normalization help optimization?, in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 2488, Red Hook, NY, USA, 2018, Curran Associates Inc.
- [19] P. Mehta and D. J. Schwab, arXiv **1410.3831** (2014).
- [20] *DeepLearning series: Convolutional Neural Networks*, <https://mc.ai/deeplearning-series-convolutional-neural-networks/>.