中国科学院大学

University of Chinese Academy of Sciences

学士学位论文

求解分类问题的量子神经网络方法

作者姓名：王俊彬

指导教师：杨义峰 研究员

中国科学院物理研究所

学位类别：理学学士

专　　业：物理学

学院（系）：中国科学院大学物理科学学院

2021 年 6 月

Classification with Quantum Neural Networks

A thesis submitted to the

University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Bachelor of Natural Science

in Physics

By

Wang Junbin

Supervisor: Professor Yang Yifeng

University of Chinese Academy of Sciences, School of Physical Sciences

June, 2021

中国科学院大学
学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日　　期：

中国科学院大学
学位论文授权使用声明

本人完全了解并同意遵守中国科学院大学有关保存和使用学位论文的规定，即中国科学院大学有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：　　导师签名：

日　　期：　　日　　期：

摘 要

在经典神经网络损失曲面的研究上，以 Fisher 信息矩阵为核心的理论取得了诸多进展。然而，量子神经网络的 Fisher 信息矩阵却由于量子系统的性质而难以计算。为了解决这个问题并使经典和量子网络可以直接比较，本文通过修改网络结构，找到了一种适用于经典和量子网络的 Fisher 信息矩阵计算方法。在实验测试中，本文先对经典和量子网络进行训练，发现经典和量子网络给出了不同的训练结果。然后，基于该方法，本文发现 Fisher 信息矩阵最大特征值的变化与训练的过程存在关联，而且这种关联在经典和量子情形中是不同的。这说明经典和量子网络的训练过程是不同的，两者的损失曲面存在较大差异。

关键词：量子神经网络，损失曲面，Fisher 信息矩阵

Abstract

To get more knowledge of the loss landscape of classical neural networks, the theories about the Fisher information matrix (or simply "Fisher") have made great progress. However, the characteristics of a quantum system bring some difficulties in calculating the Fisher for quantum neural networks. This paper aims to find a way to calculate the Fisher and compare classical neural networks and quantum neural networks. By training these two kinds of networks, we find that they yield quite different results. We use the method developed here to calculate the Fisher and find that the largest eigenvalues of the Fisher vary for classical and quantum networks in the training processes. It means that the training processes and the loss landscapes are different for these two kinds of networks.

Keywords: Quantum neural networks, Loss landscape, Fisher information matrix

目 录

图形列表

第 1 章　引言

1.1　研究背景

近些年，神经网络，特别是深度神经网络，在诸多领域中取得了卓越成就 (LeCun 等, 2015)。但为什么神经网络能取得如此优异的成绩以及在什么情况下神经网络能够发挥其潜能仍然没有答案。为了找到利用神经网络的合适方法，以及充分理解所使用的神经网络，我们需要给出上述问题的答案。同时，这些问题也是该领域学术研究的重点和热点。一方面，对于不同的实际问题，通过改变神经网络的结构，从经验上对结果进行分析，我们已经找到了很多在特定环境下工作得很好的网络结构，其中典型的代表包括卷积神经网络 (LeCun, 1989) 和注意力模型 (Vaswani 等, 2017) 等。另一方面，关于神经网络损失曲面的分析也引起了很多关注。神经网络损失曲面的分析针对的是损失函数在参数空间的形状，主要的分析方法是通过计算 Hessian 矩阵 (Sagun 等, 2016, 2017) 或 Fisher 信息矩阵 (Kunstner 等, 2019) 来获得损失曲面的信息。特别的，基于 Fisher 信息矩阵的分析在近些年取得了很多进展。归结起来，对 Fisher 信息矩阵的讨论主要包括两个方面：一方面是计算 Fisher 信息矩阵特征值的方法，包括基于平均场近似的 Fisher 信息矩阵特征值统计性质的计算方法 (Karakida 等, 2019)，基于随机矩阵理论的 Fisher 信息矩阵特征值的计算方法 (Pennington 等, 2018) 等；另一方面是以 Fisher 信息矩阵为基础的对神经网络的泛化能力、训练能力等方面进行评估的理论，包括有效维度理论 (Abbas 等, 2020)，Fisher-Rao 度规 (Liang 等, 2019) 等。对神经网络损失曲面与 Fisher 信息矩阵的讨论取得了很多关注，因为这种讨论可以增加我们对神经网络的了解。

除了神经网络之外，在近些年同样取得卓越进展的领域还包括量子计算领域 (Nielsen 等, 2010)。量子计算现在处于 NISQ (Noisy Intermediate-Scale Quantum) 时代 (Preskill, 2018)，即利用五十到几百个量子比特进行运算。尽管量子计算技术还不够成熟，但是将神经网络引入量子计算领域、构建量子神经网络的理论已取得诸多进展。利用量子线路计算神经网络归结为三个问题 (Pérez-Salinas 等, 2020)：如何将经典数据载入量子计算机中转化为量子态？对这些量子态应当执行什么样的运算？对最终得到的量子态进行何种测量？关于这三个问题有着许

许多多不同的回答，也就构成了不同的量子神经网络结构。不过，关于这些网络结构的讨论往往局限于对网络本身的结构进行描述和对该结构是否能够实现经典神经网络功能的经验性分析，而关于量子神经网络能力的理论分析，以及同经典网络进行比较的理论分析则相当匮乏。与经典情形一致，当考虑量子网络的损失曲面结构时，仍然需要借助 Hessian 矩阵和 Fisher 信息矩阵的帮助。对特定量子神经网络与经典网络的 Hessian 矩阵的比较已经说明了量子神经网络与经典网络的损失曲面具有差异 (Huembeli 等, 2021)。而关于 Fisher 信息矩阵的讨论，仅有基于 Fisher 信息矩阵提出的有效维度理论 (Abbas 等, 2020)，该理论给出一种比较经典和量子神经网络的可行方法。Fisher 信息矩阵作为研究神经网络的关键一环，也是比较量子神经网络和经典神经网络的关键一环，应当给予足够的重视。但是当下量子神经网络的 Fisher 信息矩阵计算方法还是沿用经典方法，这忽略了量子计算的特点，会带来一定的问题。虽然在经典领域 Fisher 信息矩阵特征值本身的意义尚不够明确，但它作为其它理论的基础，在经典和量子网络进行比较的理论尚且空缺的背景下，直接计算经典和量子网络的 Fisher 信息矩阵的特征值进行比较是有意义的。

1.2 研究目标与文章安排

出于这些考虑，本文将以一种量子的二分类神经网络结构 (Pérez-Salinas 等, 2020) 为基础，指出经典网络的 Fisher 信息矩阵计算方法直接沿用到量子情形会遇到的困难。为了求解 Fisher 信息矩阵并以此比较经典和量子网络的差异性，本文通过略微修改经典和量子网络的结构，尝试提出一种绕开这种困难的解决方案，并讨论这种解决方案的局限性以及可能带来的问题。在此之后，本文将使用该方法直接计算该模型和对应经典网络的 Fisher 信息矩阵特征值，并进行比较，以此来尝试回答三个问题：一，在训练过程中 Fisher 信息矩阵的特征值具有什么直观含义？据我们所知，这一点即使对于经典网络都没有文献进行讨论。二，训练过程中这种量子模型与经典模型的 Fisher 信息矩阵特征值存在什么差异？三，通过使用相同的算法进行训练，经典网络和这种量子分类器的表现有什么差异？这个问题能一定程度上说明经典神经网络算法对这种量子网络的适用性。

后续内容将安排如下：第二章引入经典的 Fisher 信息矩阵，并回顾其意义以及计算方法；第三章指明特定的量子神经网络结构，并论述直接将经典的 Fisher

信息矩阵计算方法套用过来将遇到的问题，然后提出一种可能的解决方法；第四章直接计算经典网络和量子网络的 Fisher 信息矩阵特征值，并对结果进行分析；第五章对前面结果进行总结和讨论，并指出尚待研究讨论的问题。

第 2 章　经典 Fisher 信息矩阵回顾

本章将从信息几何领域的 Fisher 信息矩阵的含义谈起，以提供该领域的规范化语言。然后将 Fisher 信息矩阵引入机器学习领域，最后说明其计算方法。一方面，本章希望追溯经典情形下 Fisher 信息矩阵的含义，以说明其重要性；另一方面，经典 Fisher 信息矩阵的计算方法对于量子情形也是极为重要的。

2.1　信息几何学中的 Fisher 信息矩阵

Amari (1985) 提供了一套将微分几何引入统计模型的方法，与之相关的领域被称为信息几何学。统计模型 $S = \{p(\boldsymbol{x}, \boldsymbol{\theta})\}$ 是以 $\boldsymbol{\theta}$ 为参数的概率分布族，$\boldsymbol{x}$ 为采样空间的随机变量，参数 $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_n)$ 的取值范围为 R^n 的开子集。方便起见，一般将矢量 $\boldsymbol{x}$、$\boldsymbol{\theta}$ 等简写为标量 x、θ 等，只有当需要进行矢量运算时才提及其矢量特性。这样的一个统计模型构成关于 θ 的统计流形，流形中的每一点对应一个 θ，并对应着一个概率分布 $p(x, \theta)$。

在这个流形上的每一个点 θ 可以定义切空间 T_θ，该切空间的一组自然基底为 $\left\{\frac{\partial}{\partial\theta_i}\right\} = \{\partial_i\}$，这里将 $\frac{\partial}{\partial\theta_i}$ 简写为 ∂_i。所以任意切向量 A 可以表示为 $A = \sum_i A^i\partial_i$。

定义另一个向量空间 T_θ^l，该空间的一组自然基底为 $\{\partial_i l\}$，其中 $l = \log p(x, \theta)$，所以该空间上任意一个向量可以写作 $A(x) = \sum_i A^i\partial_i l$。显然切空间 T_θ 与该空间 T_θ^l 同构。值得注意的是，切空间 T_θ 与 θ 有关，而 T_θ^l 与 θ 以及 x 都有关。

进一步，在流形上定义同一切空间 T_θ 上切向量 A 和 B 的内积 $\langle A, B\rangle$，定义内积后，该流形被称为黎曼空间。由于切空间 T_θ 与 T_θ^l 同构，可以定义内积

$$\langle A, B\rangle = \mathrm{E}[A(x)B(x)], \qquad \dots (2.1)$$

其中 $\mathrm{E}[f(x)] = \int f(x)p(x, \theta)dx$ 是对该点 θ 关于 x 满足分布 $p(x, \theta)$ 的期望值。根据内积的定义，可以得到度规张量的矩阵元为

$$g_{ij} = \langle\partial_i, \partial_j\rangle = \mathrm{E}[\partial_i l\partial_j l], \qquad \dots (2.2)$$

该矩阵就是 Fisher 信息矩阵。由此可以得到 Fisher 信息矩阵的第一个含义是黎曼空间的度规张量。

需要辨明，在欧几里得空间中，每一点 θ 代表的是自身这个值，两个点之间的距离就是欧几里得距离。而在黎曼空间中，每一点 θ 代表的是一个概率分布 $p(x,\theta)$，所以计算两个点之间的距离就需要利用式 (2.2) 中定义的黎曼度规，这个距离恰好就是这两个概率分布之间的相对熵 (Kullback-Leibler 散度)。特别，当黎曼度规 $g_{ij}=\delta_{ij}$ 时，黎曼度规与欧几里得度规一致。

2.2 机器学习中的 Fisher 信息矩阵

在机器学习问题中，假设 x 为输入，y 为该输入 x 对应的真实值，$f(x,\theta)$ 是以 θ 为参数的机器学习模型对输入 x 进行计算输出的结果。损失函数 L 是用以比较根据 $f(x,\theta)$ 进行预测的结果与真实情况 y 相似程度的函数，机器学习的目标就是降低损失函数，也即在训练中通过调整参数 θ 使得模型输出的预测结果与 y 更接近。模型对应的函数 f 的形式依赖于所选择的模型，损失函数 L 的形式依赖于所考虑的问题及其在训练中的实际表现。

可以将上一节的理论引入机器学习中 (Amari, 1998)，考虑概率模型 $\{p(z,\theta)\}$，其中 $z=(x,y)$。x 就是机器学习中定义的输入，为一个随机变量。但是 y 并不是真实值，而是用来表示该问题可能的取值，也是随机变量。所以，机器学习模型的每一个状态对应一个参数 θ 的取值，就对应着一个概率分布

$$p(z,\theta)=p_\theta(x,y)=p(x)p(y|f(x,\theta)), \qquad \ldots (2.3)$$

其中 $p(x)$ 是输入 x 满足的分布，条件概率 $p(y|f(x,\theta))=p_\theta(y|x)$ 是将 x 输入到当前模型得到 $f(x,\theta)$ 并且基于该输出预测的结果等于某一个可能取值 y 的概率。

典型的损失函数 (Amari, 1998) 可以直接定义为

$$\begin{aligned} L(\theta) &= \mathrm{E}_{x,y\sim p(x,y)}[-\log p_\theta(x,y)] \\ &= \mathrm{E}_{x\sim p(x)}[-\log p(x)]+\mathrm{E}_{x,y\sim p(x,y)}[-\log p_\theta(y|x)], \end{aligned} \qquad \ldots (2.4)$$

这里是对输入 x 和可能取值 y 满足的真实联合概率分布 $p(x,y)=p(x)p(y|x)$ 求期望值。上式中前面一项不依赖于模型参数 θ，在训练过程中相当于常数，可以直接略去，得到理论上的损失函数

$$L(\theta)=\mathrm{E}_{x,y\sim p(x,y)}[-\log p_\theta(y|x)]. \qquad \ldots (2.5)$$

但是实际过程中并不知道 $p(x,y)$，所以该式是不能计算的。在真实环境下，只能得到由 N 组输入和对应真实值组成的样本集 $\{(x_i,y_i)\}_{i=1}^N$，所以在训练过程

中使用的是经验损失函数

$$L(\theta) = \frac{1}{N} \sum_i -\log p(y_i|f(x_i,\theta)). \quad \ldots (2.6)$$

由式 (2.5) 可以看出，机器学习中降低损失函数的目标与增大 $\log p_\theta(y|x)$ 的目标是完全一致的，其背后的思想就是极大化对数似然。

值得注意的一点是，到目前为止，除了将 $p_\theta(y|x)$ 与损失函数联系在一起外，$p_\theta(y|x)$ 是没有别的要求的，所以它的具体形式还不确定。在实际过程中，并不是通过决定 $p_\theta(y|x)$ 形式来决定 $L(\theta)$ 的形式。相反，往往是根据问题中 $L(\theta)$ 常用的形式来寻找对应的 $p_\theta(y|x)$ 的形式。换言之，对于给定的问题，在给定损失函数后，便能够给出 $p_\theta(y|x)$ 的形式。

那么损失函数 L 作为 θ 的函数，应当选择什么样的策略来改变 θ 进而降低损失函数？一种非常重要的方法是自然梯度下降法 (Amari, 1998)：在移动距离相同，也即在 $|d\boldsymbol{\theta}|^2 = \epsilon^2$，$\epsilon \to 0$ 的条件下，使得 $L(\boldsymbol{\theta} + d\boldsymbol{\theta})$ 最小。

令 $d\boldsymbol{\theta} = \epsilon \boldsymbol{a}$，$|\boldsymbol{a}|^2 = \boldsymbol{a}^{\mathrm{T}} g \boldsymbol{a} = 1$，其中 g 是 Fisher 信息矩阵，也就是度规张量，其矩阵元是按照式 (2.2) 定义。$\boldsymbol{a}$ 代表更新方向的单位矢量。所以 $L(\boldsymbol{\theta} + d\boldsymbol{\theta}) = L(\boldsymbol{\theta}) + \epsilon \nabla L(\boldsymbol{\theta})^{\mathrm{T}} \boldsymbol{a}$。相当于在 $\boldsymbol{a}^{\mathrm{T}} g \boldsymbol{a} = 1$ 的限制条件下，寻找 $\boldsymbol{a}$ 的方向让 $\frac{L(\boldsymbol{\theta}+d\boldsymbol{\theta})-L(\boldsymbol{\theta})}{\epsilon} = \nabla L(\boldsymbol{\theta})^{\mathrm{T}} \boldsymbol{a}$ 最小。利用拉格朗日乘子法，即寻找 $\boldsymbol{a}$ 的方向最小化

$$\nabla L(\boldsymbol{\theta})^{\mathrm{T}} \boldsymbol{a} - \lambda \left(\boldsymbol{a}^{\mathrm{T}} g \boldsymbol{a} - 1 \right). \quad \ldots (2.7)$$

对 $\boldsymbol{a}$ 求导，可以得到

$$\nabla L(\boldsymbol{\theta}) - 2\lambda g \boldsymbol{a} = 0 \Rightarrow \boldsymbol{a} = \frac{1}{2\lambda} g^{-1} \nabla L. \quad \ldots (2.8)$$

所以更新方向 $\boldsymbol{a}$ 与参数空间的 Fisher 信息矩阵的逆矩阵同损失函数梯度乘积的方向相反。当 Fisher 信息矩阵为单位矩阵时，该方向就是损失函数的负梯度方向，对应一般的梯度下降方法。在一般情况下，Fisher 信息矩阵指明了梯度下降最快的方向，这是 Fisher 信息矩阵的第二个含义。

2.3 Fisher 信息矩阵的计算方法

在机器学习的框架下，根据式 (2.2) 定义，Fisher 信息矩阵的定义为

$$\begin{aligned} F(\theta) &= \mathrm{E}_{x,y\sim p_\theta(x,y)}\left[\nabla \log p_\theta(x,y) \nabla \log p_\theta(x,y)^{\mathrm{T}}\right] \\ &= \mathrm{E}_{x,y\sim p_\theta(x,y)}\left[\nabla \log p_\theta(y|x)p(x) \nabla \log p_\theta(y|x)p(x)^{\mathrm{T}}\right] \\ &= \mathrm{E}_{x,y\sim p_\theta(x,y)}\left[\nabla \log p_\theta(y|x) \nabla \log p_\theta(y|x)^{\mathrm{T}}\right]. \end{aligned} \quad \ldots (2.9)$$

上式第二行中 $\log p(x)$ 对 θ 的求导为 0，可以直接丢掉。上式的期望按照定义应当是对 x, y 满足联合分布 $p_\theta(x,y) = p(x)p_\theta(y|x)$ 求期望值。但是在实际问题中并不知道输入的分布 $p(x)$，需要通过样本集中的输入分布来近似该分布。所以一般认为是 Fisher 信息矩阵的量 (Kunstner 等, 2019) 为

$$F(\theta) = \frac{1}{N}\sum_i \mathrm{E}_{y\sim p_\theta(y|x_i)}\left[\nabla \log p_\theta(y|x_i) \nabla \log p_\theta(y|x_i)^{\mathrm{T}}\right]. \quad \ldots (2.10)$$

注意，此处 y 满足的分布并不是真实分布 $p(y|x)$，而是与模型和参数相关的分布 $p_\theta(y|x_i) = p(y|f(x_i,\theta))$。如前所述，当给定损失函数 L 时，可以得到 $p(y|f(x_i,\theta))$ 的形式，所以在理论上该式可以直接求解。但在实际训练过程中，由于被积函数既依赖于网络的输出 $f(x_i,\theta)$，又依赖于求导后的结果，在程序中该式是很难直接求解的。一种可行的方法是求解 Fisher 信息矩阵的蒙特卡洛近似 (Martens 等, 2015; Kunstner 等, 2019)，即只对每一个样本 x_i 根据分布 $p(y|f(x_i,\theta))$ 采样一个值 $\tilde{y}$，然后计算

$$F(\theta) = \frac{1}{N}\sum_i \nabla \log p_\theta(\tilde{y}|x_i) \nabla \log p_\theta(\tilde{y}|x_i)^{\mathrm{T}}. \quad \ldots (2.11)$$

这种计算方法的问题是结果不够准确。当然，可以通过根据分布 $p(y|f(x_i,\theta))$ 采样多个 $\tilde{y}$，通过更大的计算量来换取更准确的 $F(\theta)$。

另一种可以获得准确 $F(\theta)$ 的方法需要 p 的形式满足额外的条件 (Martens, 2014)：当 $p(y|f(x,\theta))$ 为关于自然参数 f 的指数分布族时，Fisher 信息矩阵具有简单形式

$$F(\theta) = \frac{1}{N}\sum_i -\left[\mathrm{J}_\theta f(x_i,\theta)\right]^{\mathrm{T}} \nabla_f^2 \log p(y_i|f(x_i,\theta)) \left[\mathrm{J}_\theta f(x_i,\theta)\right], \quad \ldots (2.12)$$

其中 $\mathrm{J}_\theta f(x_i,\theta)$ 指 f 对 θ 求导的雅可比矩阵。值得注意的是，由于 $p(y|f(x,\theta))$ 为关于自然参数 f 的指数分布族，实际上该式中关于 f 的二次求导结果与 y 无关，

所以关于 y 的期望值为 1。进而，表达式中的 y 可以用任意值来表示，这里用 y_i 表示。为完整起见，式 (2.12) 的证明过程将放在附录 A 中。

具体而言，对于 C-分类问题的神经网络模型，网络将输出 C 个值 $\{f_c\}_{c=1}^{C}$。通常选择交叉熵为损失函数，即

$$L = -\frac{1}{N}\sum_i \sum_c q(y=c|x_i)\log p(y=c|f(x_i,\theta)), \qquad \ldots (2.13)$$

其中对 c 的求和是在对 y 选择不同类别的情况进行求和。$q(y=c|x_i)$ 为输入 x_i 属于类别 c 的真实概率分布，$p(y=c|f(x_i,\theta))$ 为神经网络对输入 x_i 输出预测的属于类别 c 的概率分布。这样的损失函数可以衡量根据神经网络输出进行预测得到的概率分布与真实分布之间的差异性。对于样本中的输入 x_i，一般认为样本中的结果 y_i 是唯一的可能。即只有当 $c=y_i$ 时，$q(y=c|x_i)$ 为 1，否则为 0。所以损失函数简化为

$$L = -\frac{1}{N}\sum_i \log p\left(y=y_i|f(x_i,\theta)\right). \qquad \ldots (2.14)$$

该结果的形式与式 (2.6) 的形式一致。

选择 $p(y=c|f(x_i,\theta))$ 为神经网络输出 $\{f_c\}$ 经过 softmax 函数后的结果 (Kunstner 等, 2019)，即

$$p(y=c|f(x_i,\theta)) = \frac{e^{f_c}}{\sum_t e^{f_t}}, \qquad \ldots (2.15)$$

softmax 函数将输出 $\{f_c\}$ 转化为属于不同的类的概率分布。将该式带入式 (2.12) 可以得到 Fisher 信息矩阵的计算公式

$$F(\theta) = \frac{1}{N}\sum_i \left[\mathrm{J}_\theta f(x_i,\theta)\right]^{\mathrm{T}}\left(\mathrm{diag}(\pi_i) - \pi_i\pi_i^{\mathrm{T}}\right)\left[\mathrm{J}_\theta f(x_i,\theta)\right], \qquad \ldots (2.16)$$

其中 π_i 指代对于不同输入 x_i，由 $\{p(y=c|f(x_i,\theta))\}$ 构成的列向量。$\mathrm{diag}(\pi_i)$ 指将列向量 π_i 转化为相应的对角矩阵。具体计算过程将放在附录 A 中。应当强调，该式为下一章进行讨论以及后续计算的基础。

2.4　Fisher 信息矩阵与 Hessian 矩阵的关系

Hessian 矩阵和 Fisher 信息矩阵是研究损失曲面的两个关键量：Hessian 矩阵是损失函数 L 的二阶导数 $H=\nabla^2 L$，能够直接反映损失函数的局部曲率。而相

比之下，Fisher 信息矩阵是统计模型 $\{p_\theta(x, y)\}$ 构成的黎曼空间的度规，也即相对熵的二阶导数。

这两个矩阵均能反映损失曲面的性质，而且 Fisher 信息矩阵可以与 Hessian 矩阵的高斯牛顿近似联系起来 (Kunstner 等, 2019)，但这并不意味着 Fisher 信息矩阵的特征值一定会表现出与 Hessian 矩阵的特征值完全相同的行为。虽然两者确实在很多情况下有类似的行为，但是两者之间的关系的精细刻画还需要进一步研究。

第 3 章　量子模型中的 Fisher 信息矩阵

前面一章介绍了 Fisher 信息矩阵的重要性，并针对分类问题的神经网络模型给出了具体的求解方法。当考虑到量子神经网络模型时，自然而然地会想要将这种求解方法直接引入，但实际上这种引入会遇到一些问题。本章将先引入经典的单层神经网络模型和一种简单量子神经网络分类模型 (Pérez-Salinas 等, 2020)，然后说明套用经典方法求解 Fisher 信息矩阵会遇到的困难，并提出一种绕开这种困难的解决方法，最后分析这种解决方法带来的后果。

在具体讲述之前，本文希望强调一点：虽然量子神经网络的结构多种多样，但这里遇到的困难很可能是普遍的，因为它来源于量子计算中必不可少的测量过程。相比于本文提出的解决方法，这种困难本身的意义更为重要，因为这种困难与经典和量子计算之间的差异紧密相关，解决这个困难的方法也是使经典和量子能够对应比较的关键因素。

3.1　单层神经网络

考虑 C-分类问题，假设输入的维度为 n，输出维度为 C，隐藏层的神经元个数为 m，最简单的单隐藏层的神经网络的结构如图 3.1 所示。假设非线性的激活函数为 ϕ，神经网络的计算方法为：

$$h_j = \phi\left(\sum_i w_{ji} x_i + b_j\right), \qquad \ldots (3.1)$$

$$f_t = \sum_j v_{tj} h_j, \qquad \ldots (3.2)$$

其中 $\{w_{ji}\}$、$\{b_j\}$ 和 $\{v_{tj}\}$ 均为模型的参数，它们共同组成前面提到的 θ。训练过程就是通过改变这些参数来调整神经网络的输出，进而降低损失函数。参数的总数为 $nm + m + mC$，对应的就是 θ 的维度。特别，考虑输入维度为 2 的二分类问题时，参数总数为 $5m$。

由式 (3.1) 和 (3.2) 可以看出，这样的单隐藏层神经网络做的就是：将输入层的向量 $\{x_i\}$ 乘以两层之间的参数矩阵 $\{w_{ji}\}$，再加上偏移量 $\{b_j\}$，最后逐个经过激活函数 ϕ 进行非线性化得到隐藏层向量 $\{h_j\}$。隐藏层向量 $\{h_j\}$ 再乘以隐藏层

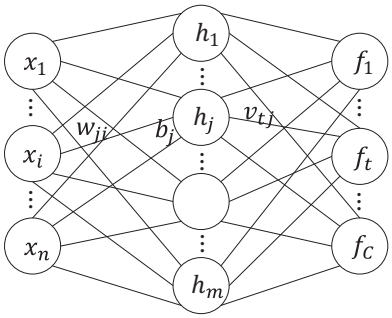


图 3.1 经典神经网络结构示意图。

Figure 3.1 The structure of the classical neural network.

和输出层之间的参数矩阵 $\{v_{tj}\}$，得到输出向量 $\{f_t\}$。对于分类问题，输出的向量还需要经过一次 softmax 函数，即按照式 (2.15) 计算来得到该输入属于不同类别的概率分布。该网络结构中极为重要的部分是激活函数 ϕ，该函数使得上述运算变成非线性的。虽然上述网络结构非常简单，但是该结构符合万能近似定理 (Cybenko, 1989)，即当隐藏层的神经元数目足够多时，该网络可以近似任何连续函数。

3.2 单量子比特分类器

在神经网络计算过程中，逐层进行运算时需要进行矩阵运算，所以不可避免的是单个神经元的数据被多次使用。这对经典的比特是很正常的，但是对于量子比特，由于不可克隆原理 (Nielsen 等, 2010)，即不能将一个量子比特复制多次，重复使用一个神经元的数据的操作是不可行的。

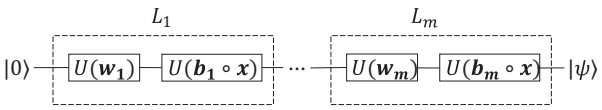


图 3.2 量子神经网络结构示意图。

Figure 3.2 The structure of the quantum neural network.

基于这个原因，Pérez-Salinas 等 (2020) 提出利用单个量子比特进行运算的重载入的神经网络分类器结构。该结构与上述单层神经网络结构相似，其示意图如图 3.2 所示。这种设计的核心思想是将经典神经网络中多次利用一个神经元的数据的行为，看作是数据的多次载入。进而将该思想引入量子神经网络中，每一次

对量子比特进行运算后，就再次将数据载入，以此作为一层。这里考虑输入数据的维度为 2 的情况。每一层的运算是对单个比特先进行一次 SU(2) 旋转，然后再将输入的数据通过另一个 SU(2) 旋转载入，具体表达式为

$$L_j = U\left(b_{j1}x_1, b_{j2}x_2, 0\right) U\left(w_{j1}, w_{j2}, w_{j3}\right), \qquad \ldots (3.3)$$

其中的 U 代表 SU(2) 群中的群元，可以用欧拉角具体表示为

$$\begin{aligned} U(a,b,c) &= R_z(a)R_y(b)R_z(c) \\ &= \exp\left(-\frac{iaZ}{2}\right)\exp\left(-\frac{ibY}{2}\right)\exp\left(-\frac{icZ}{2}\right), \end{aligned} \qquad \ldots (3.4)$$

其中 i 为虚数单位，Y 和 Z 为泡利矩阵。

可以简单地令量子比特的初始状态为 $|0\rangle$，则经过 m 层的运算后，该网络得到的态为

$$|\psi\rangle = L_m L_{m-1} \ldots L_1 |0\rangle. \qquad \ldots (3.5)$$

该网络的参数为 $\{w_{j1}, w_{j2}, w_{j3}, b_{j1}, b_{j2}\}$，共同组成参数 θ。假如网络有 m 层，参数总数也恰好为 $5m$。

这种网络结构属于 VQC (Varitional Quantum Circuit)，即将数据输入带参数的量子线路中进行计算并测量，对测量的结果利用经典程序计算损失函数并进行梯度下降得到更新后的参数，再将参数放回量子线路中，通过重复这一系列操作对该模型进行训练。

但是现在只是指明了量子线路的计算方法，还需要选择合适测量方法。对于分类问题，Pérez-Salinas 等 (2020) 提出的方案是：为每一个标签 t 指定一个相应的目标状态 $|\psi_t\rangle$，通过测量量子线路计算出的态 $|\psi\rangle$ 在目标状态上的投影 $|\langle\psi_t|\psi\rangle|^2$ 来计算损失函数。

按照这种方案，当类别的个数大于 2 的时候，对于单个量子比特而言，所选择的目标状态不可能两两正交，所以输出态在不同目标状态的投影加起来并不为 1。但是当考虑二分类问题时，可以简单地选择 $|0\rangle$ 和 $|1\rangle$ 作为两个类的目标状态。由于这两个目标状态是正交的，对于叠加态 $|\psi\rangle = a|0\rangle + b|1\rangle$ 在这两个状态上的投影 $|a^2|$ 和 $|b|^2$ 满足 $|a|^2 + |b|^2 = 1$，恰好就是叠加态处于这两种状态的概率。所以在这种情况下，对线路测量的投影分布与概率分布是完全一致的。

更重要的是，这种正交性使得期望的投影分布与真实分布一致：假设输入 x 类别为 0 (或 1)，期望得到的状态 $|\psi_e\rangle = |0\rangle$ (或$|1\rangle$) 满足条件 $|\langle 0|\psi_e\rangle|^2 = 1$ (或0) 和 $|\langle 1|\psi_e\rangle|^2 = 0$ (或1)。这满足了交叉熵表达式 (2.13) 简化为式 (2.14) 的条件，从而使得经典模型中的交叉熵可以直接迁移过来。

为了使经典和量子网络的损失函数具有相同的形式，本文考虑二分类问题，选择 $|0\rangle$ 和 $|1\rangle$ 作为量子线路进行测量的目标状态，并选交叉熵 (2.14) 作为经典和量子情形的损失函数。这样的选择使得经典和量子的结果可以通过损失函数直接进行比较，同时，这样的选择也为经典和量子网络的 Fisher 信息矩阵的比较提供了可能。

不失一般性，任何多分类问题都可以归结为二分类问题。但是仍有一点应当指出：考虑多分类问题时，如果采用式 (2.13) 定义的交叉熵形式，将概率分布用投影分布代替，此时式 (2.13) 就不能简单地简化为式 (2.14) 。这种做法或许能够起到意外的效果，但是并不在本文的讨论范围内。

3.3 Fisher 信息矩阵的计算问题

由于损失函数为交叉熵的形式,所以自然会考虑通过式 (2.12) 导出的式 (2.16) 计算 Fisher 信息矩阵。但是利用这种方法计算 Fisher 信息矩阵，需要知道网络的输出 $f(x_i, \theta)$ 以及概率分布 $p(y = c|f(x_i, \theta))$。

对于经典情形，这完全不是问题，网络输出即为 $f(x_i, \theta)$，再根据式 (2.15) 就可以得到概率分布 p。既然经典和量子的损失函数完全一致，那么选定量子情形的概率分布 $p(y = c|f(x_i, \theta))$ 为在目标状态上的投影分布，也即概率分布，是理所当然的。但这样带来的问题就是，量子情形没有对应的网络输出 $f(x_i, \theta)$。

是什么导致了这个问题？这是本文希望引起关注的一点：量子线路的测量究竟意味着什么？在将一个态转化为某些目标状态上的投影的过程中，究竟发生了什么？与经典网络的输出没有范围限制不同，量子线路测量的值只能处于 0 到 1 之间。这一点来源于量子态密度的归一化条件，是量子本身的性质。这样的性质将会带来什么后果，以及与经典过程能否对应起来，都是值得思考的问题。

回到当前讨论的问题，这种性质带来的后果就是使得量子情形难以与经典产生对应关系，难以确定网络的输出。如果考虑网络的输出为测量前的量子态 $|\psi\rangle$，那么式 (2.16) 的计算方法将完全不适用于量子情形。Stokes 等 (2020) 提出

了另一种思路：经典情形的 Fisher 信息矩阵是衡量不同参数 θ 下 p_θ 之间的距离的度规张量，在量子情形下就可以定义类似的度规张量用以衡量不同参数 θ 下输出态 $|\psi_\theta\rangle$ 之间的距离。这样做的后果是这种量子的 Fisher 信息矩阵将与曲面性质没有直接关系，抛弃了经典已有的理论，更无法与经典情形对应起来。

如果将测量后的值看做是网络的输出 f，就会遇到前面提到的问题：在经典网络中输出 f 的取值范围是没有要求的，但是量子网络中测量后的值的范围仅仅在 0 到 1 之间。这样一来，很难将经典和量子情形直观地进行比较。

在提供具体的方法之前，需要强调，我们希望找到这种对应关系，一方面是为了计算量子网络的 Fisher 信息矩阵，并且充分继承经典情况下它的丰富内涵以及理论。另一方面，这种对应关系使得经典和量子网络的直接比较成为可能。合适的对应关系，应当使得损失函数的值以及损失函数的变化范围完全一致，并且在这种对应关系下计算的 Fisher 信息矩阵也可以对应起来。这样一来，对于经典和量子网络，无论是损失函数，还是 Fisher 信息矩阵的特征值，值大小和变化趋势都是可以直接比较的。更进一步的，基于经典 Fisher 信息矩阵的理论就可以直接用以衡量比较经典和量子网络。在输入和输出的维度均合适的情况下，两种网络的参数个数可以较为接近，从而可以直观地比较经典和量子网络。

3.4　添加 softmax 层的解决方法

量子测量的结果范围在 0 到 1 之间，自然而然地可以认为这组值对应的是经典网络的输出 f 经过 softmax 函数作用后的结果，也仅仅在这种情况下，两者才可能对应起来。在这种认知下，一种可行的解决方法是：对于经典和量子网络，将这样对应处在 0 到 1 之间的值，看作是网络的输出 f，并将再次经过 softmax 函数的结果作为 p。对应于经典网络，就是在原来的输出层后添加一层 softmax 层，以此结果作为网络的输出 f，而对该结果进行二次 softmax 处理后的结果作为概率分布 p；对应于量子网络，就是将测量结果作为输出 f，然后再添加额外的经典线程对结果进行 softmax 处理作为 p。这样一来，两个网络的输出 f 均为两个处于 0 到 1 之间、并且相加为 1 的值，网络的概率分布为该值的进行 softmax 结果，处于 $\frac{1}{e+1}$ 到 $\frac{e}{e+1}$ 之间。这样一来，两者的 Fisher 信息矩阵均可以通过式 (2.16) 进行计算，并且相应的损失函数大小、损失函数变化范围和 Fisher 信息矩阵的特征值均可以直接比较。

这种计算方法非常简单，通过对经典和量子网络结构进行简单的修改，以增加少量计算量为代价，使得经典和量子网络可以直接计算并比较。同时，由于网络的两个输出和为 1，所以式 (2.16) 中求解两个输出 f_1 和 f_2 对参数求导的过程就可以简化：两个导数存在简单的相反数关系。所以在计算 Fisher 信息矩阵时，计算量还会有所降低。

但是这种做法有什么问题？首先，这种做法有局限性。这种做法是为了解决量子网络测量结果范围为 0 到 1，难以与经典输出直接匹配的问题。为经典网络添加 softmax 层，使得经典的输出在 0 到 1 之间，并且相加为 1。对于分类问题这背后的本质是，只有在二分类问题中，两个正交的目标状态才能保证期望的投影分布与真实分布相统一，进而使得交叉熵满足式 (2.14) 的形式，才使得经典和量子网络可以进行比较。这一特点杜绝了利用 POVM (Positive Operator-Valued Measure) 测量 (Nielsen 等, 2010) 进行改进的可能。但在回归问题中，常见的损失函数 $L = \frac{1}{N}\sum_i \frac{1}{2}|y_i - f(x_i,\theta)|^2$ 并不需要用到真实分布，所以该方法可以很简单地泛化过去。因此，这种处理方法实际上是适用于所有监督学习问题的，但是并不能直接处理多分类问题。与之相比，蒙特卡洛近似的计算方法 (2.11) 很难用于连续的 y 的情况。

不过，这种方法修改了网络结构。就算是仅仅在网络最后增加了一层 softmax 层，但也确确实实改变了网络的输出 f、概率分布 p 以及损失函数 L，进而改变了神经网络的训练过程、改变了损失曲面的几何结构。这会带来两个问题：

首先，经典情形下，网络通过输出 f 计算 p 直接计算损失函数用于训练，而修改后的网络相当于将正常的输出 f 经过两次 softmax 操作后的结果用来计算损失函数。所以，在实际问题中表现得很好的网络经过这样的修改后是否依旧表现得很好？这个问题很难回答，需要进行很多经验性的验证。而且如果修改后的网络无法取得较好的表现，那么这种修改后的结构就是没有意义的。相反，如果修改后的网络仍然能表现得很好，那么修改后的网络同量子网络进行比较的结果就是有参考价值的。直觉上可以认为在网络最后添加一层 softmax 层对网络能力的影响并不是非常大，所以可以认为第二种情况是合理的，但是这仍然需要更多的实验检测。幸运的是，实际上训练过程和计算 Fisher 信息矩阵的过程是独立的，两者所需的导数是不同的，所以可以将训练和计算 Fisher 信息矩阵的过程分开。在训练过程中，按照没有修改的网络进行训练；在计算 Fisher 信息矩阵时，

按照修改后的网络进行计算。具体而言，经典网络 (图 3.3) 的输出 f 为式 (3.2) 的计算结果，而量子网络 (图 3.4) 没有这样的输出。经典网络输出 f 经过一次 softmax 函数后得到 p_1，该结果对应于量子网络的测量结果。经典网络输出 f 经过两次 softmax 函数后得到 p_2，对应于量子测量结果经过一次 softmax 函数的结果。损失函数 (2.14) 利用 p_1 进行运算

$$L = -\frac{1}{N}\sum_i \log p_1(y = y_i|f(x_i,\theta)), \qquad \ldots (3.6)$$

并根据该结果对网络进行训练。Fisher 信息矩阵 (2.16) 利用 p_1 和 p_2 进行运算

$$F(\theta) = \frac{1}{N}\sum_i \left[\mathrm{J}_\theta p_1(x_i,\theta)\right]^{\mathrm{T}} \left(\mathrm{diag}(\pi_i) - \pi_i\pi_i^{\mathrm{T}}\right) \left[\mathrm{J}_\theta p_1(x_i,\theta)\right], \qquad \ldots (3.7)$$

其中 π_i 指由 $\{p_2(y = c|f(x_i,\theta)\}$ 构成的列向量。这样一来，既充分利用了经典网络结构分析的经验性结论，又使得经典和量子网络可以进行比较。

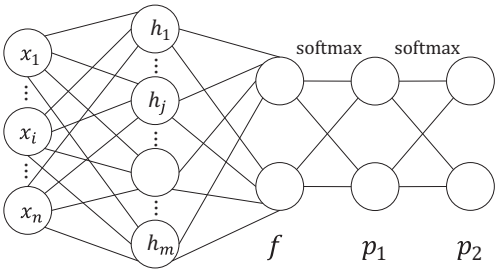


图 3.3 修改后的经典神经网络结构示意图。

Figure 3.3 The structure of the modified classical neural network.

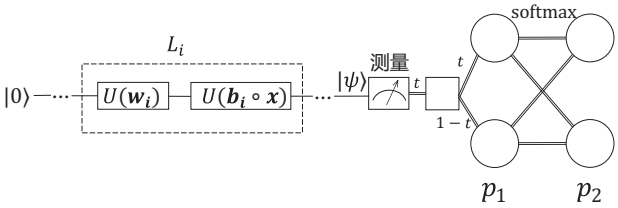


图 3.4 修改后的量子神经网络结构示意图。

Figure 3.4 The structure of the modified quantum neural network.

但是这么做，也就与第二个问题相关：这样计算的 Fisher 信息矩阵是修改后的网络的 Fisher 信息矩阵。需要指明，一方面，将训练和计算分开的方法只是利用原来的网络帮助训练，比较的仍然是修改后的网络，使得经典和量子网络可以

直接进行比较才是这种方法的关键。另一方面，我们希望修改后的网络的 Fisher 信息矩阵特征值仍然能反映原来网络矩阵特征值的性质。这样一来，对修改后经典和量子网络的比较结果也适用于未修改的情况。所以如果能够证明这一点，这种解决方法将更具意义。下一节将通过实验具体观察添加 softmax 层对 Fisher 信息矩阵最大特征值的影响。

3.5 添加 softmax 层的影响

直觉上讲，修改后的网络由于添加了一层 softmax 层，如果对 θ 进行微扰，网络最终输出 p_1 的改变相比 f 的改变更小，也就说明损失曲面变得平缓，对应于曲面的度规，也即 Fisher 信息矩阵的值变小。并且这种修改很可能不会在损失曲面上添加新的鞍点或极值点，也不会使得训练过程中 Fisher 信息矩阵特征值的变化趋势发生改变。

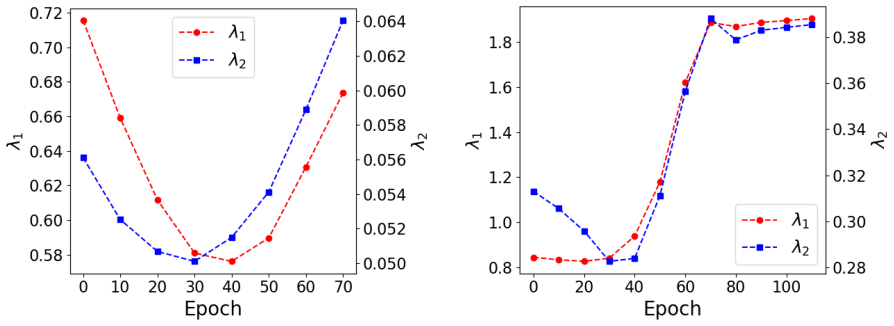
对上述经典网络，根据式 (3.6) 进行训练，计算两个 Fisher 信息矩阵

$$F_1(\theta) = \frac{1}{N}\sum_i \left[\mathrm{J}_\theta f(x_i,\theta)\right]^{\mathrm{T}} \left(\mathrm{diag}(\pi_{1i}) - \pi_{1i}\pi_{1i}^{\mathrm{T}}\right) \left[\mathrm{J}_\theta f(x_i,\theta)\right], \qquad \ldots (3.8)$$

$$F_2(\theta) = \frac{1}{N}\sum_i \left[\mathrm{J}_\theta p_1(x_i,\theta)\right]^{\mathrm{T}} \left(\mathrm{diag}(\pi_{2i}) - \pi_{2i}\pi_{2i}^{\mathrm{T}}\right) \left[\mathrm{J}_\theta p_1(x_i,\theta)\right], \qquad \ldots (3.9)$$

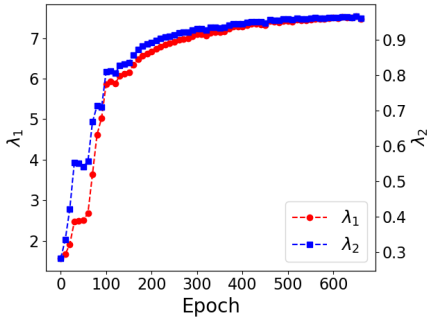
其中 π_{ji} 指代由 $\{p_j(y=c|f(x_i,\theta)\}$ 构成的列向量。

针对下一章考虑的问题，对经典网络进行训练，对 $m = 1,2,3,4,5,6,7,8$ 的情况进行测试，计算两个 Fisher 信息矩阵的最大特征值在训练过程中的变化趋势，结果如图 3.5 所示。可以看出，式 (3.9) 的 Fisher 信息矩阵最大特征值相比式 (3.8) 的最大特征值确实是变得更小了，但是两者的变化趋势基本一致，所以这种方法可以用于后续研究。

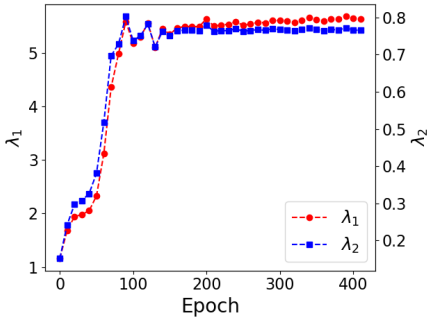


(a) 1 neuron

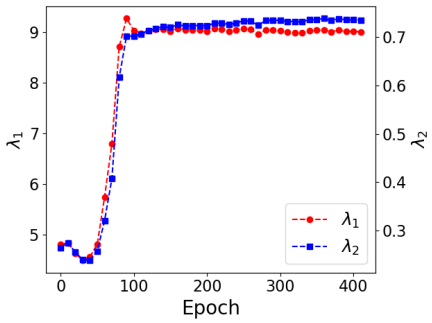
(b) 2 neurons



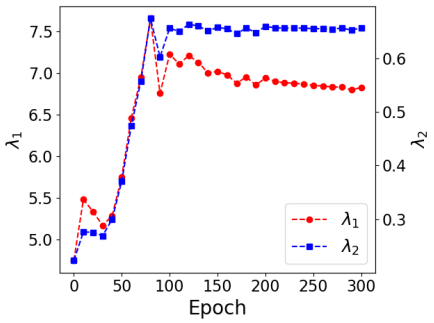
(c) 3 neurons



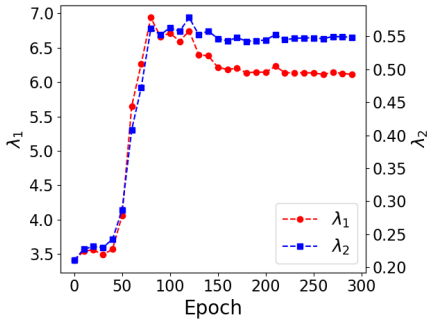
(d) 4 neurons



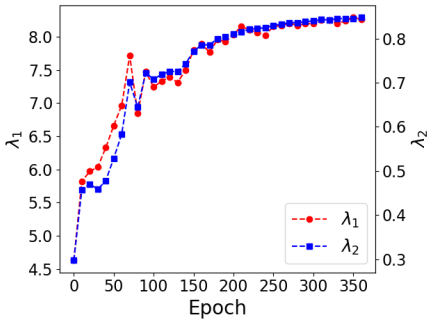
(e) 5 neurons



(f) 6 neurons



(g) 7 neurons



(h) 8 neurons

图 3.5 不同的神经元个数下经典网络训练过程中两个 Fisher 信息矩阵最大特征值的变化。λ_1 和 λ_2 分别对应式 (3.8) 和 (3.9) 计算出的 Fisher 信息矩阵的最大特征值。

Figure 3.5 The largest eigenvalues of the two Fisher information matrices in the training process of classical neural networks with different numbers of neurons. λ_1 and λ_2 are the largest eigenvalues of the Fisher calculated according to Eq. (3.8) and (3.9) respectively.

第 4 章　实验训练

本章将以前面提到的经典和量子网络结构为基础，利用前面提到的算法进行训练和计算。第一节将具体讲述测试的环境以及用到的算法，之后的章节将具体讲述实验结果。本章重点关注三个问题：第一，Fisher 信息矩阵的特征值在训练过程中有什么直观意义？第二，在对经典网络适用的算法条件下，该量子模型表现如何？第三，在训练过程中，经典和量子模型的 Fisher 信息矩阵特征值表现出什么差异？下面将试图回答这三个问题。

4.1　测试环境

本实验测试环境在 Python 3.6.8 版本下，利用 TensorFlow 2.1.0 (Abadi 等, 2015) 构建并训练经典网络模型，利用 TensorFlow_quantum 0.3.1 (Broughton 等, 2020) 模拟量子线路配合 TensorFlow 构建并训练量子网络模型。

本实验考虑 Pérez-Salinas 等 (2020) 提到的简单的二分类问题 (图 4.1)：二维平面上的点 $(x_1, x_2) \in [-1, 1]^2$ 处于一个正方形内，当 $x_1^2 + x_2^2 \leq \frac{2}{\pi}$，即处于圆内和边界上时，该点属于类 0；否则，该点属于类 1。该圆形半径的选择使得圆内和圆外的面积相同，即考虑的是均等的二分类问题。

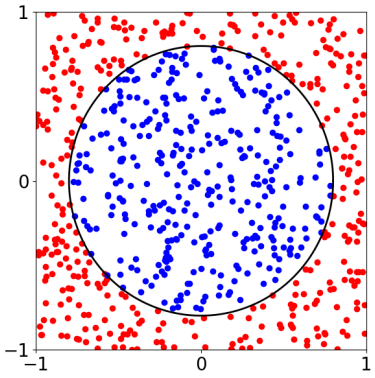


图 4.1 圆形分类问题。

Figure 4.1 Classification of a circle.

网络的参数 $\{\theta_i\}$ 从均值为 0、标准差为 1 的标准正态分布中随机取样作为初始值。在训练前，随机生成 700 组样本数据 $\{(x_1, x_2), y\}$ 作为训练集用于训练。每个训练周期，模型将对 700 组数据全部训练一次，使用的训练方法为批量梯度

下降 (BGD)：先将训练集随机打乱，然后以 50 组数据为一批输入模型，计算该批次损失函数的和，以此计算梯度，进而更新参数。每批次训练的损失函数和为

$$L_{\text{batch}} = -\sum_{i}^{\text{batch}} \log p_1\left(y = y_i^{\text{batch}} | f\left(x_i^{\text{batch}}, \theta\right)\right). \quad \ldots (4.1)$$

与式 (3.6) 相比，这里的求和是对一个批中的数据进行的，而且没有对样本数求平均。以该结果计算损失函数的梯度，更新参数

$$\theta_{\text{new}} = \theta_{\text{old}} - \alpha \nabla_\theta L_{\text{batch}}, \quad \ldots (4.2)$$

其中 α 为学习率，设定初始学习率为 0.0001。由于式 (4.1) 没有对损失函数取平均，所以初始学习率的选择与批的大小有关。每个训练周期结束后，将对训练结果进行评估，并根据结果修改学习率：计算整个训练集的总损失函数，如果损失函数增大，学习率就减半；如果损失函数降低，学习率就增加 5%。这种准则能够自动地调整学习率，帮助网络更好地收敛。

每经过 5 个训练周期，根据式 (3.7) 计算一次 Fisher 信息矩阵并计算其特征值，用于后续观察比较。网络训练 300 个周期后停止，将结果保存下来。经验上讲，300 个周期网络已经基本收敛。

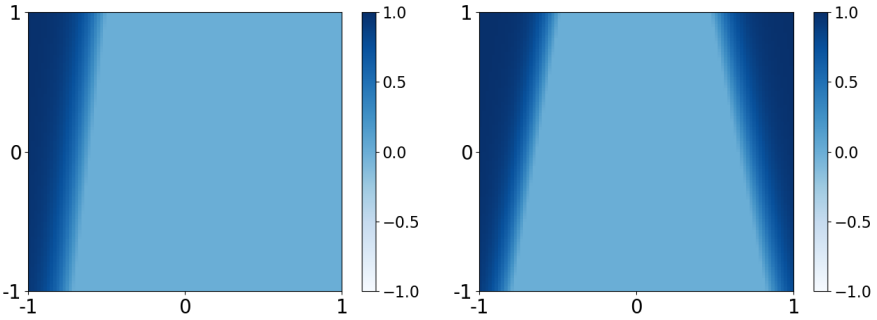
经典网络选择的激活函数为 ReLu 函数，该函数提供非线性性，为该领域的常见选择。Huembeli 等 (2021) 提到激活函数的选择可能会对结果产生影响。

将训练后的模型对正方形内的每个点进行计算得到概率分布 p_1，将其中对应类别为 1 的概率减去类别为 0 的概率作为结果。当结果小于 0 时，说明网络更有把握认为该点属于类 0；结果大于 0 时，更有把握认为该点属于类 1。将该值用颜色深浅表示，通过这种方式将结果图示出来。

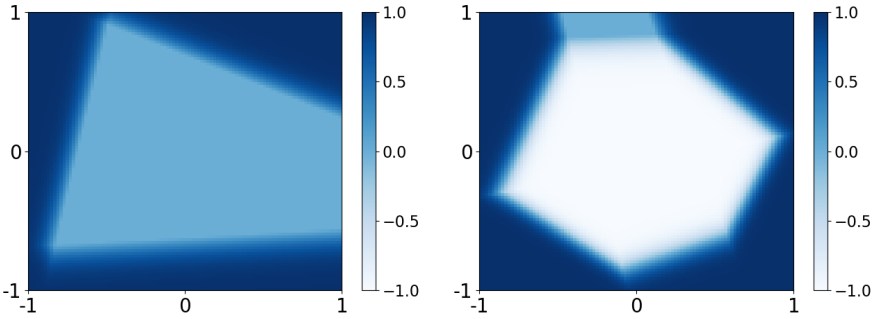
正如前一章所提到的，当经典网络隐藏层神经元个数为 m 时，参数总数为 $5m$；单比特量子神经网络的层数为 m 时，参数总数也为 $5m$。由于个人计算机性能的限制，将对 $m = 1, 2, 3, 4, 5, 6, 7, 8$ 的情况进行训练，比较相同算法条件、相同参数总数条件下经典和量子神经网络体现出的差异性。

4.2 经典情形

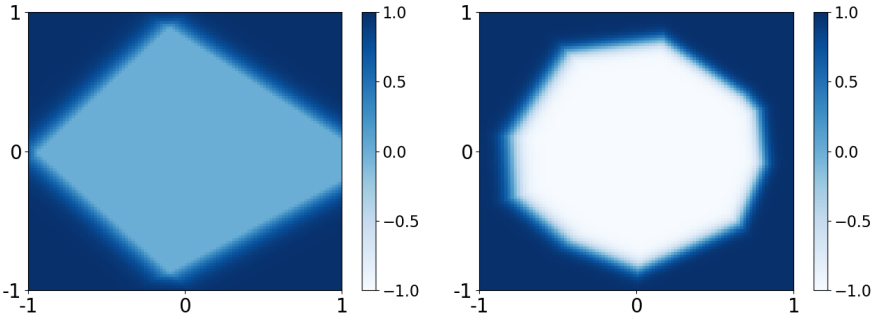
对于不同神经元数目，一次训练结果如图 4.2 所示。可以发现，随着神经元的数目增加，该网络结构通过边数逐渐增加的多边形来逼近圆形分类面。该结果与理论分析的结果 (Shalev-Shwartz 等, 2014) 一致。



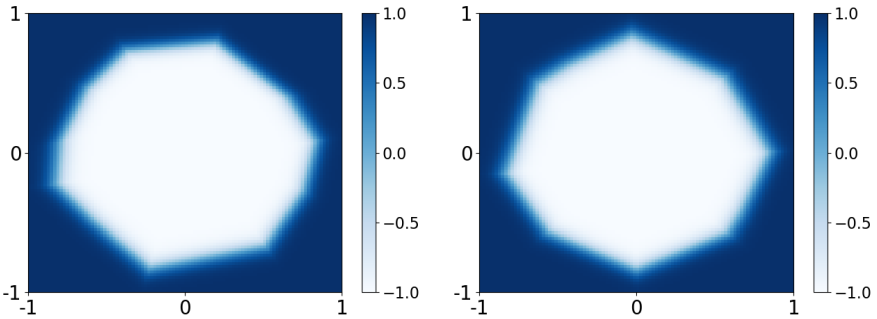
(a) 1 neuron　　(b) 2 neurons



(c) 3 neurons　　(d) 4 neurons



(e) 5 neurons　　(f) 6 neurons

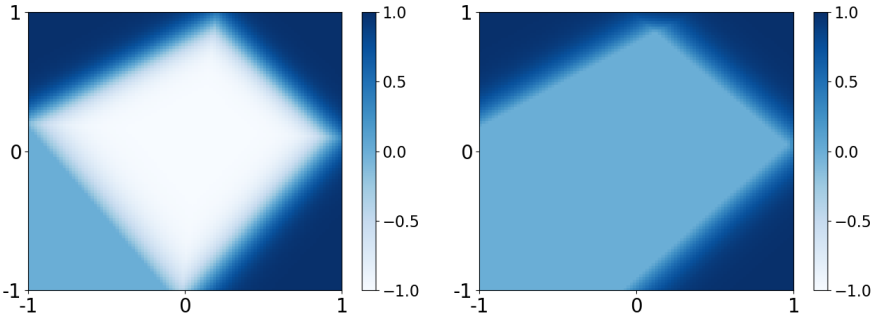


(g) 7 neurons　　(h) 8 neurons

图 4.2 不同神经元个数下经典网络的训练结果。该点的颜色越深代表网络越有把握认为该点属于类 1 (位于圆外)。

Figure 4.2 The training results of classical neural networks with different numbers of neurons. Points are more likely to be labeled as Class 1 (outside the circle) in darker places.

图 4.3 是对 3 个神经元的不同的测试结果，可以发现当隐藏层神经元的数目为 3 个左右时，该网络结构能够开始给出较好的分类结果。在神经元个数为 3 个的时候，网络可能收敛得较好，也可能收敛较差。



(a) A good result **(b) A bad result**

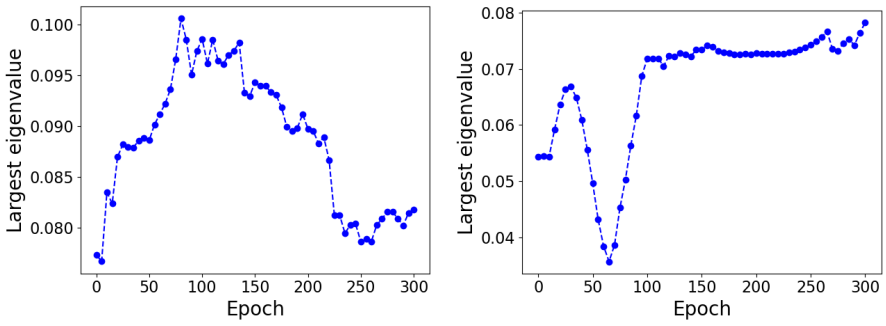
图 4.3 3 个神经元的经典网络给出不同的训练结果。

Figure 4.3 The different training results of classical neural networks with 3 neurons.

这些结果可以说明，在这种算法条件下，虽然有时会出现如图 4.2(e) 这样收敛得不太好的结果，但是在大多数情况下，该网络结构可以较好地工作。当然，更好的算法可以帮助网络更好地收敛，逃离不好的鞍点或者极值点，但是这方面的内容并不在本文的讨论范围内。更进一步地，下面开始考察训练过程中 Fisher 信息矩阵特征值的变化情况。

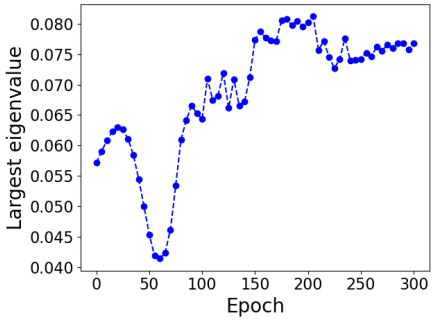
图 4.4 给出的 Fisher 信息矩阵最大特征值在训练过程中的变化，该结果与图 4.2 相对应。对于训练结果较好的情况 (图 4.4(d)、图 4.4(f)、图 4.4(g) 和图 4.4(h))，在训练过程中，Fisher 信息矩阵最大特征值一开始逐渐增大，然后在一个值的附近开始小范围波动，并且该值的趋势是逐渐降低。结合图 4.5 将训练过程中最大特征值与损失函数进行比较的结果，可以发现总损失函数开始缓慢下降的时间与最大特征值开始小范围波动的时间基本吻合，这说明最大特征值在小范围内波动的行为对应着模型开始逼近一个鞍点或极值点并逐渐收敛到这个点。在这几组测试中，最大特征值的变化范围以及收敛过程中的值的大小并没有体现出与神经元数目有明显关系。但是与训练结果较差的情况相比，最大特征值的变化范围会大一个数量级。

当网络训练结果较差时 (图 4.4(a)、图 4.4(b)、图 4.4(c) 和图 4.4(e))，最大特征值实际上是在一个小范围内波动，这种行为与上述结果较好的情况下训练后期逐渐收敛时最大特征值小范围波动的行为类似，这可能意味着这几个图对应

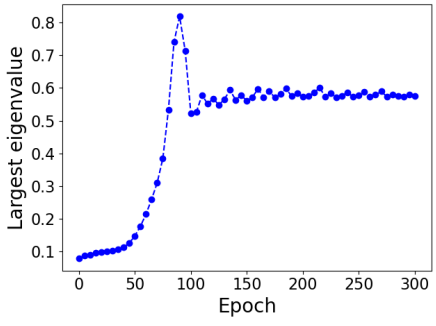


(a) 1 neuron

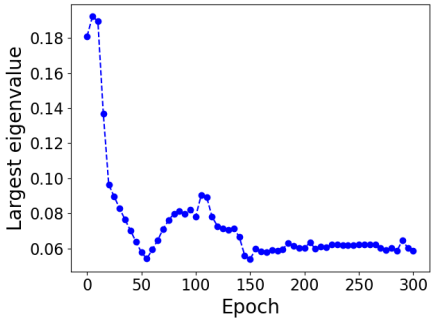
(b) 2 neurons



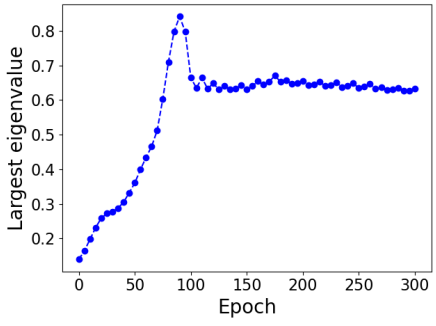
(c) 3 neurons



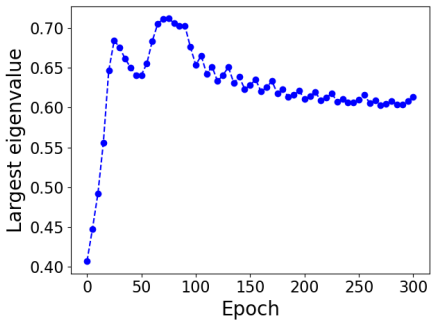
(d) 4 neurons



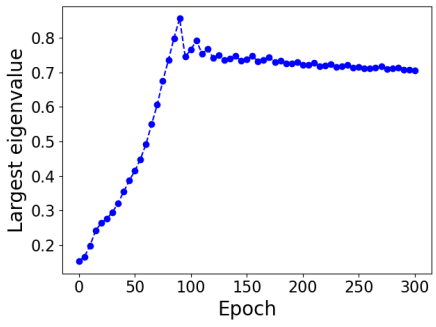
(e) 5 neurons



(f) 6 neurons



(g) 7 neurons



(h) 8 neurons

图 4.4 不同神经元个数下经典网络 Fisher 信息矩阵最大特征值在训练过程中的变化。

Figure 4.4 The largest eigenvalues of the Fisher in the training process of classical neural networks with different numbers of neurons.

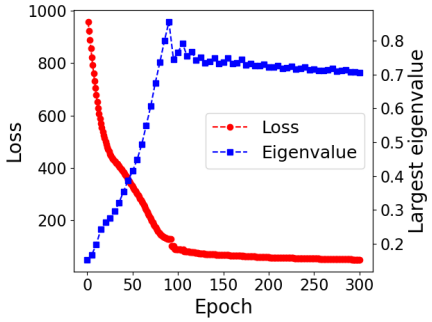


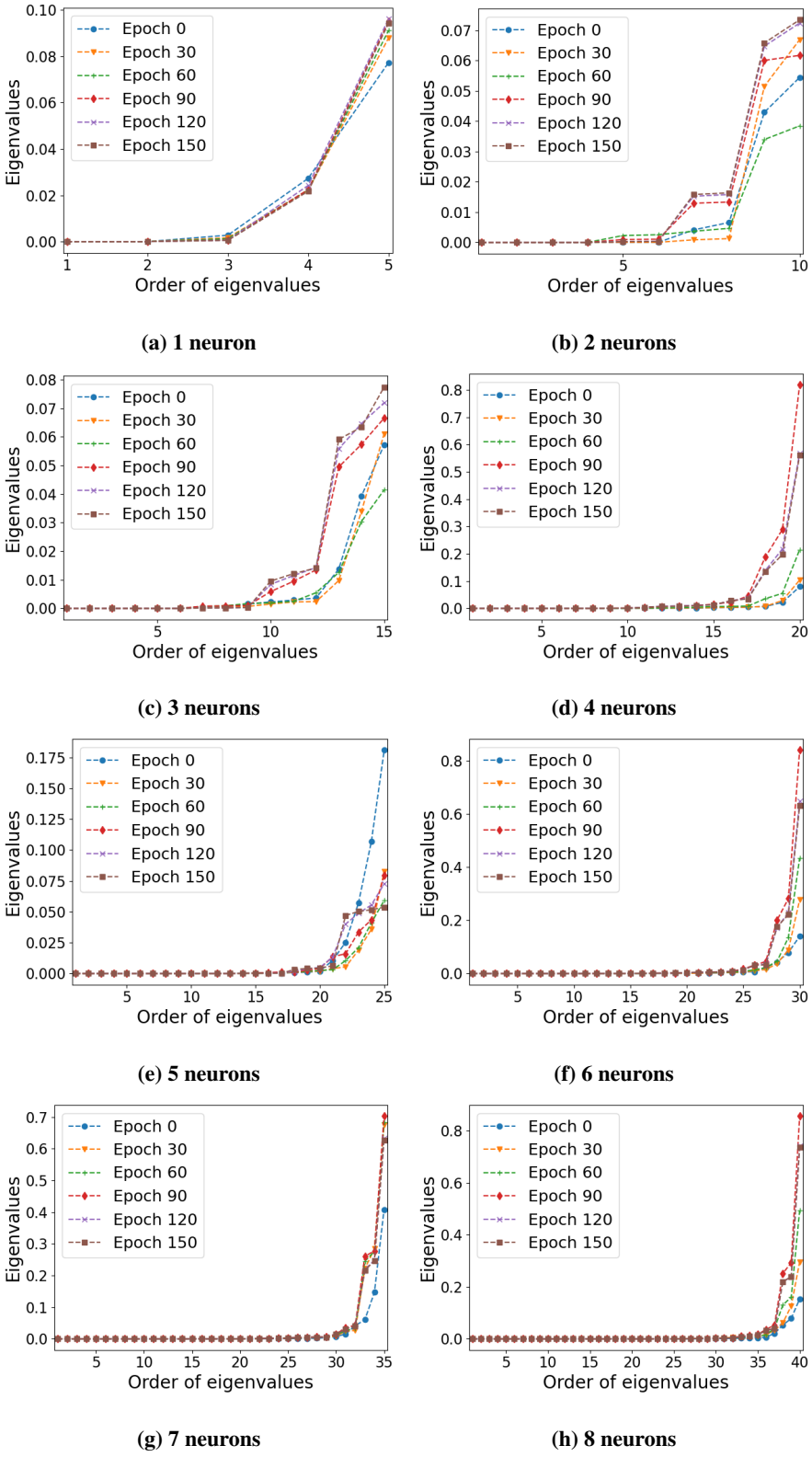
图 4.5 8 个神经元的网络训练过程中损失函数的变化与 Fisher 信息矩阵最大特征值的变化。

Figure 4.5 The loss and the largest eigenvalue of the Fisher in the training process of classical neural networks with 8 neurons.

着模型一开始就处于鞍点或极值点附近，并且直接收敛到这个不好的点。

这些结果说明，经典情况下，Fisher 信息矩阵最大特征值很可能与训练状态对应：在训练还未收敛的过程中，最大特征值逐渐增大；当处于鞍点或极值点附近时，最大特征值会在一个值附近波动，并且该值逐渐降低。

图 4.6 展示在训练过程中所有特征值的变化，该结果也与图 4.2 相对应。可以看出除最大值以外的其它特征值的变化较为复杂：在未收敛的训练过程中，这些值整体表现为逐渐增大，该结果与 Sagun 等 (2016) 提到 Hessian 矩阵特征值在训练过程中向 0 靠拢的趋势是不同的；在模型逐渐收敛时，这些值有时增大，有时减小。不同值的变化趋势也不尽相同，并且与最大特征值的变化趋势没有直接联系。该结果没有表现出明显的规律，因此后面将主要研究最大特征值的变化。



(a) 1 neuron　**(b) 2 neurons**

(c) 3 neurons　**(d) 4 neurons**

(e) 5 neurons　**(f) 6 neurons**

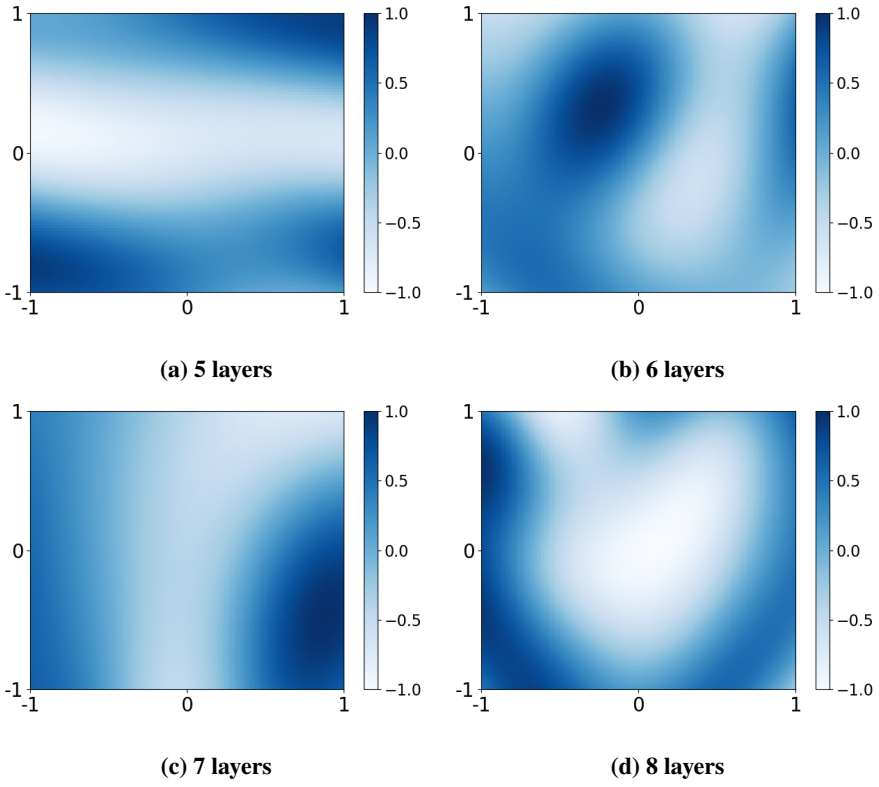
(g) 7 neurons　**(h) 8 neurons**

图 4.6 不同神经元个数下 Fisher 信息矩阵特征值在训练过程中的变化。

Figure 4.6 The eigenvalues of the Fisher in the training process of classical neural networks with different numbers of neurons.

4.3 量子情形

在相同条件下，对量子网络进行测试，部分结果如图 4.7 所示。与经典结果 (图 4.2) 相比，该结果很不理想，给出的分类结果较差。这说明在本实验的设计条件下，量子网络模型无法收敛到一个较好的点。实际上，Pérez-Salinas 等 (2020) 提到对该模型使用批量梯度下降算法进行训练得到的结果较差，而使用 L-BFGS-B 算法 (Byrd 等, 1995) 进行训练会取得很好的结果。这一点可能是由于 L-BFGS-B 算法是二阶算法，而批量梯度下降算法是一阶算法，更高阶的算法可以帮助网络收敛到一个更好的点。这正说明这种量子模型的损失曲面可能更为复杂，所以针对该模型的算法设计或参数选择需要更多考虑。



(a) 5 layers **(b) 6 layers**

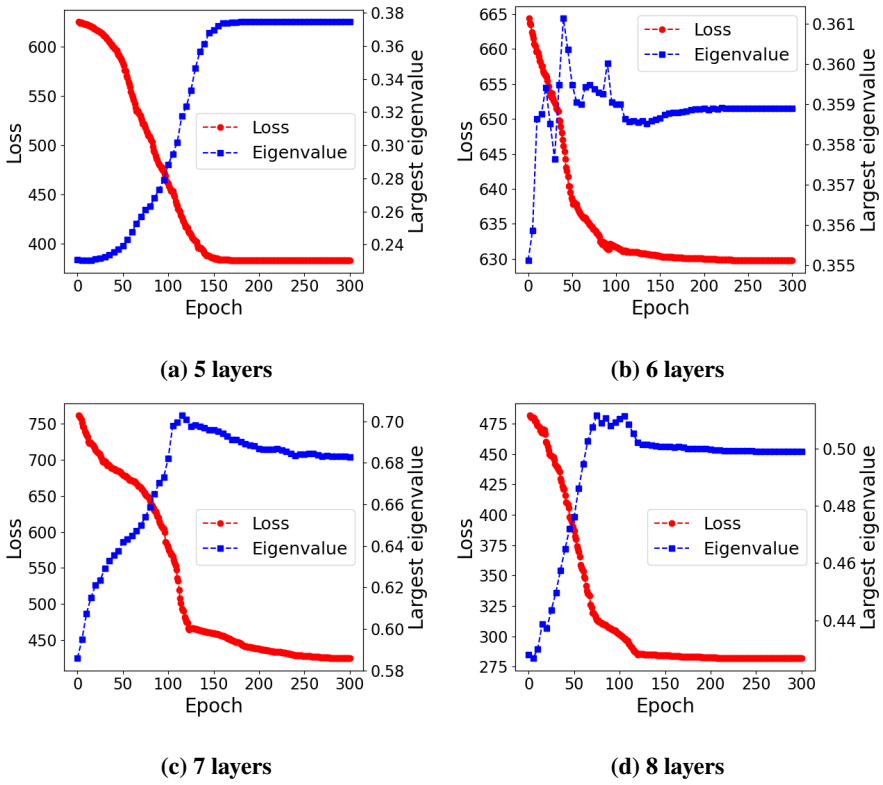
(c) 7 layers **(d) 8 layers**

图 4.7 不同层数下量子网络的训练结果。该点的颜色越深代表网络越有把握认为该点属于类 1 (位于圆外)。

Figure 4.7 The training results of quantum neural networks with different numbers of layers. Points are more likely to be labeled as Class 1 (outside the circle) in darker places.

图 4.8 展示训练过程中 Fisher 信息矩阵最大特征值和损失函数的变化情况，可以看出量子情形下最大特征值的行为与经典情形 (图 4.5) 展现出相同的特点：在尚未收敛的训练过程中，最大特征值逐渐增大；当靠近鞍点或极值点逐渐开始

收敛时，最大特征值在小范围内波动。但是与经典情况相比，图中量子网络的最大特征值的变化范围较小，这很可能意味着这些网络很快地逼近了一个较为不好的鞍点或极值点，并且没有逃离出来，图 4.7 的结果并不能代表该模型能力的上限。



(a) 5 layers　　**(b) 6 layers**

(c) 7 layers　　**(d) 8 layers**

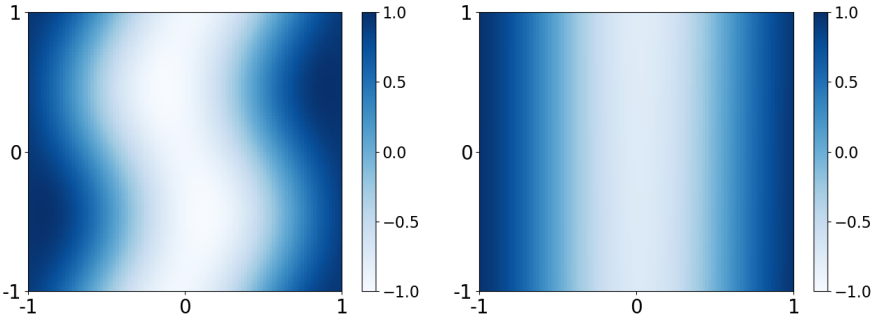
图 4.8 不同层数下量子网络训练过程中损失函数的变化与 Fisher 信息矩阵最大特征值的变化。

Figure 4.8 The losses and the largest eigenvalues of the Fisher in the training process of quantum neural networks with different numbers of layers.

4.4　随机梯度下降结果

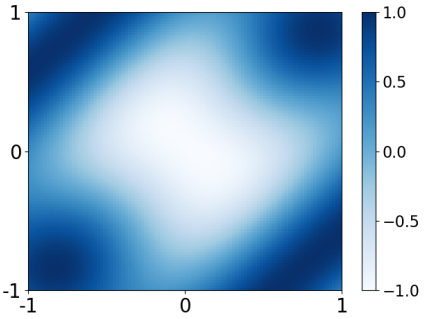
为了使量子网络收敛得更好，将批大小修改为 1，也即使用随机梯度下降 (SGD) 进行训练。需要注意的是，由于批大小变小，所以实际上在一个训练周期中，参数更新的次数会更多。并且这种方法会引入更多随机性，从而使得网络更有可能逃出不好的鞍点或极值点。

量子网络训练结果如图 4.9 所示，可见结果得到了明显的好转。当层数大于 3 层时，该网络就可以得到较好的分类结果。与经典的结果 (图 4.2) 相比，量子网络的结果有几点明显的差异：首先，该量子模型并不是通过增加多边形边数来

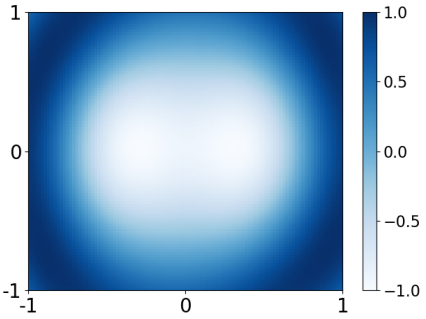


(a) 1 layer

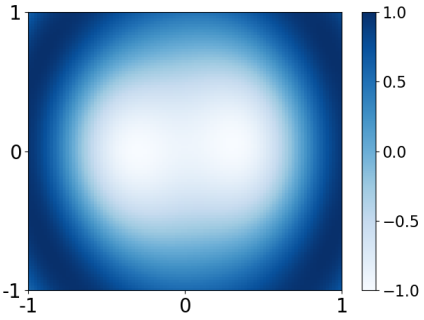
(b) 2 layers



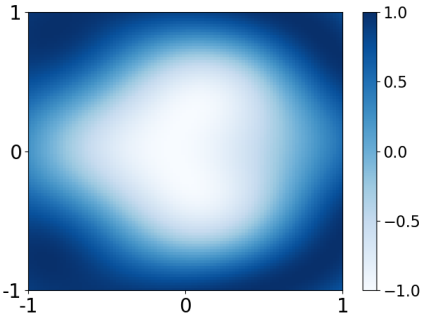
(c) 3 layers



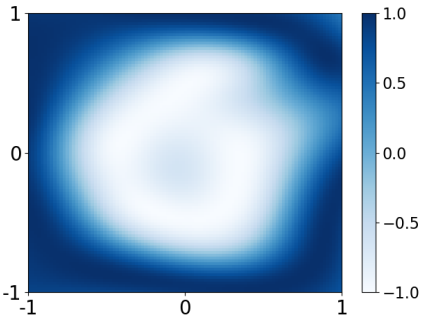
(d) 4 layers



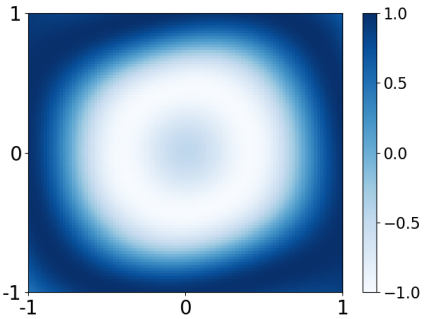
(e) 5 layers



(f) 6 layers



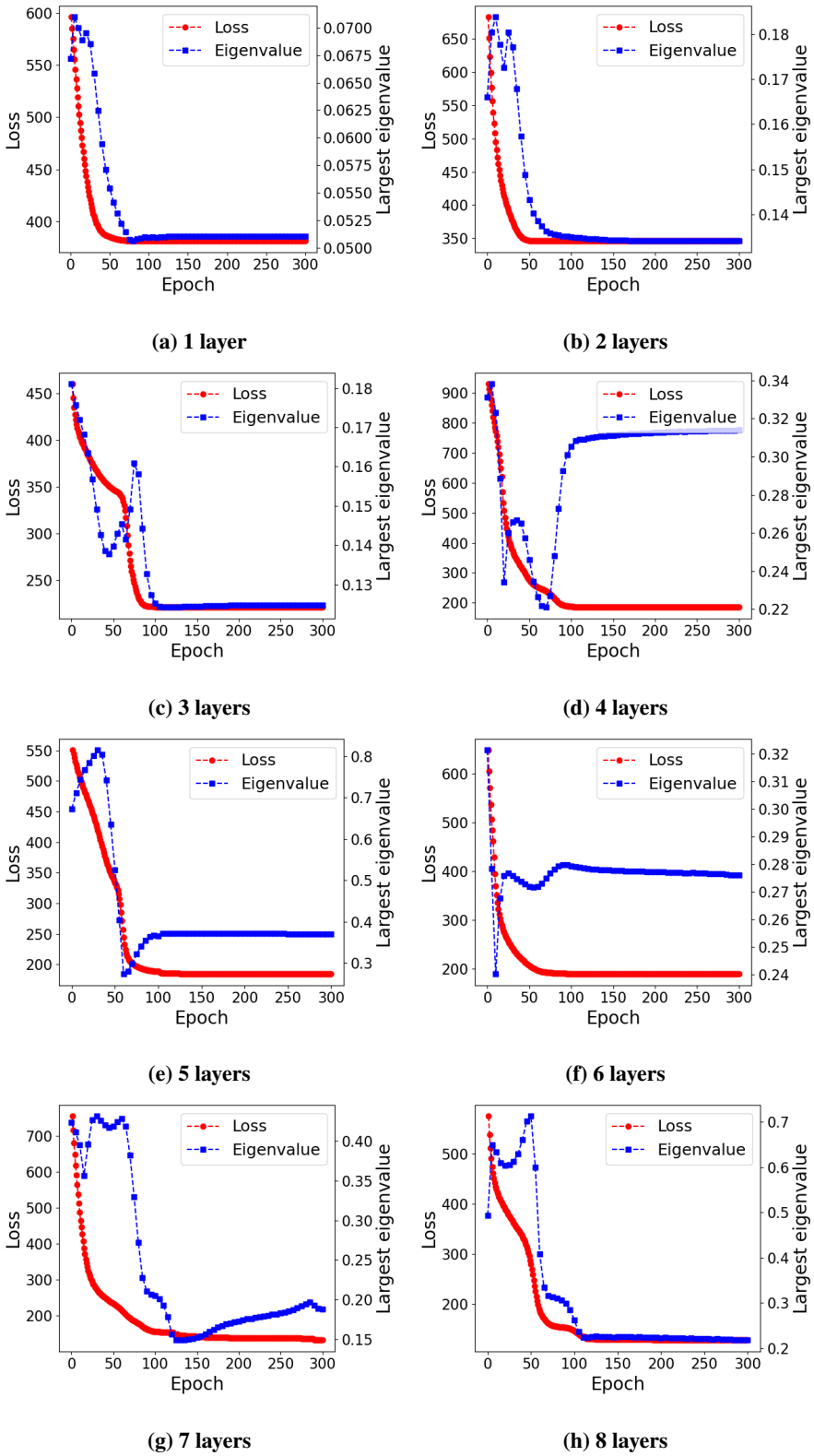
(g) 7 layers



(h) 8 layers

图 4.9 不同层数下用 SGD 对量子网络进行训练的结果。该点的颜色越深代表网络越有把握认为该点属于类 1 (位于圆外)。

Figure 4.9 The training results of quantum neural networks with different numbers of layers using SGD. Points are more likely to be labeled as Class 1 (outside the circle) in darker places.



(a) 1 layer　(b) 2 layers

(c) 3 layers　(d) 4 layers

(e) 5 layers　(f) 6 layers

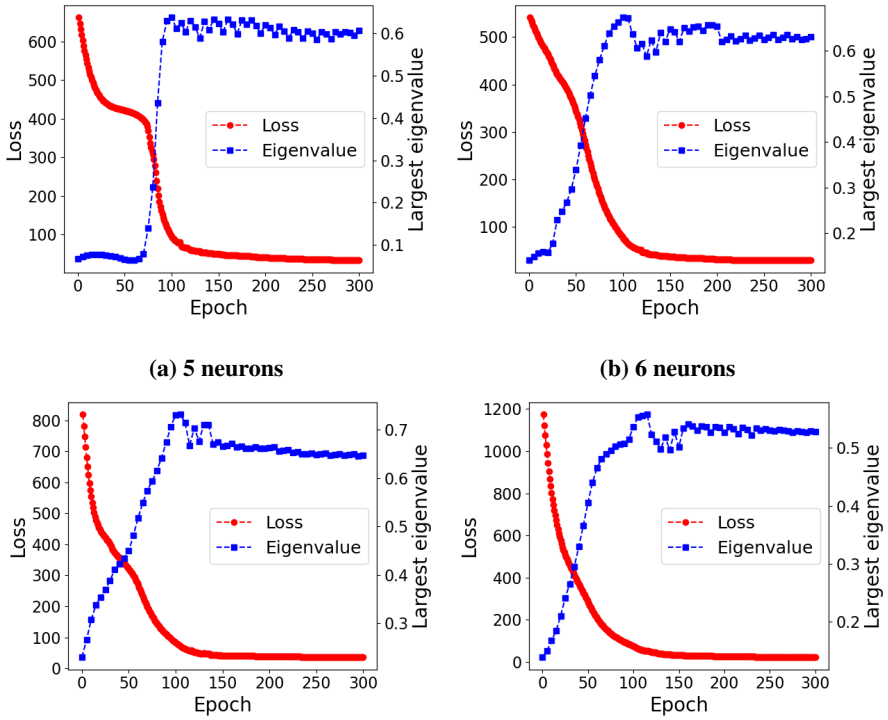
(g) 7 layers　(h) 8 layers

图 4.10 不同层数下量子网络 SGD 训练过程中损失函数的变化与 Fisher 信息矩阵最大特征值的变化。

Figure 4.10 The losses and the largest eigenvalues of the Fisher in the training process of quantum neural networks with different numbers of layers using SGD.

逐步近似圆形的，而是直接捕捉圆形的特征；其次，在分类面的附近，量子模型的输出并不会随输入坐标的细微改变而剧烈变化，与经典结果相比该变化更为平缓；最后，在远离分类面的圆内或圆外，量子网络对这些位置的输入分为 0 或 1 的把握也不是非常大。这些现象与 Huembeli 等 (2021) 训练 192 个参数的多量子比特分类器得到的结果类似。

值得注意的是，当量子模型的层数为 1 时 (图 4.9(a))，不同于经典情形 (图 4.2(a)) 只能简单地以一条直线作为分类面，该模型能得到由两条曲线构成的分类面。同时，在参数更多的情况下，该模型抓取圆形分类面的方式也与经典不同。这说明该量子模型引入的非线性性与经典网络中由激活函数引入的非线性性不同。



(a) 5 neurons **(b) 6 neurons**

(c) 7 neurons **(d) 8 neurons**

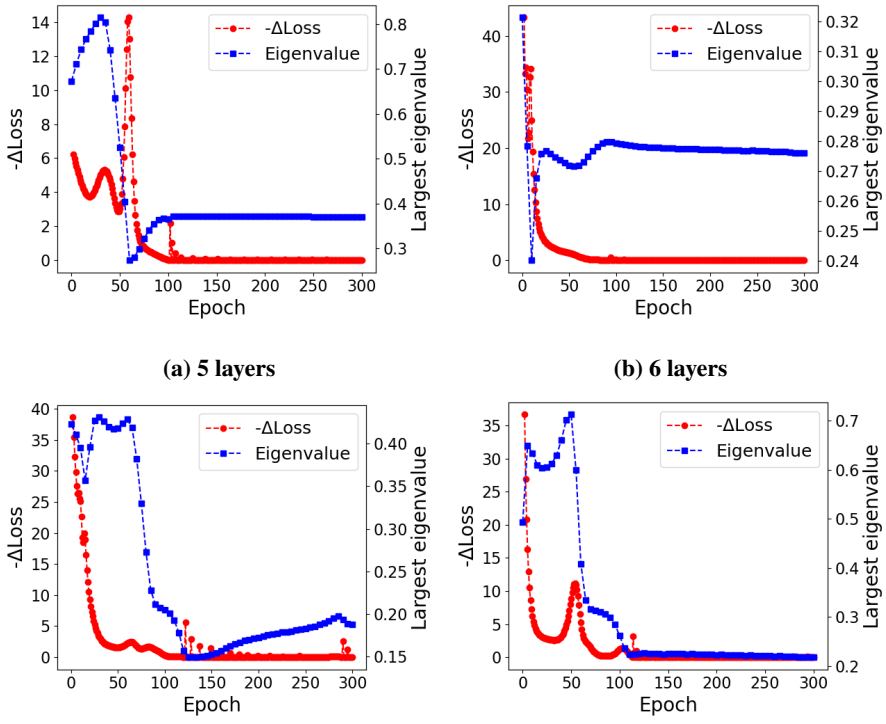
图 4.11 不同神经元个数下经典网络 SGD 训练过程中损失函数的变化与 Fisher 信息矩阵最大特征值的变化。

Figure 4.11 The losses and the largest eigenvalues of the Fisher in the training process of classical neural networks with different numbers of neurons using SGD.

图 4.10 展示 SGD 训练过程中最大特征值与损失函数的变化情况，可以看出该结果与前面的结果有较大的差异。在具体描述差异之前，为了避免算法对结果产生影响，同样使用 SGD 算法对经典网络进行训练。结果如图 4.11 所示，该结

果与批量梯度下降的结果 (图 4.5) 相比没有明显变化，仍然表现出与前面一致的现象：在未收敛的训练过程中，最大特征值持续增加；在逐渐收敛的过程中，最大特征值在一个值附近波动。

与之相比，量子网络给出了不同的结果 (图 4.10)，具体表现为：在未收敛的训练过程中，最大特征值没有明显的上升段，而是有着较大的波动；在逐渐收敛的过程中，最大特征值在多数情况下先会出现一个可以与变化范围相比拟的较大幅度的下降，然后在小范围内波动。



(a) 5 layers　　**(b) 6 layers**

(c) 7 layers　　**(d) 8 layers**

图 4.12 不同层数下量子网络 SGD 下降训练过程中损失函数下降率的变化与 Fisher 信息矩阵最大特征值的变化。

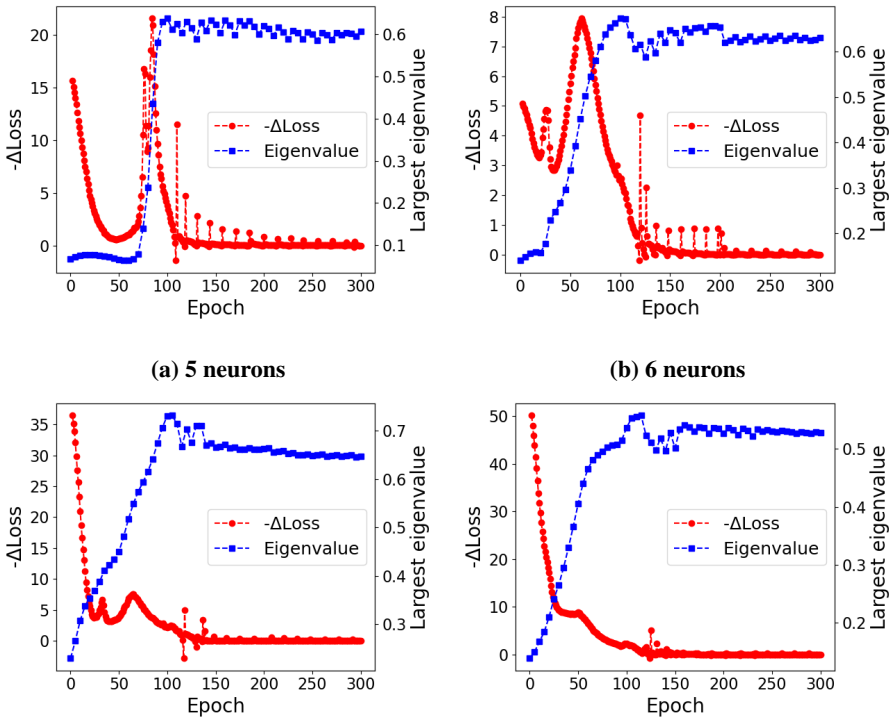
Figure 4.12 The decreasing rates of the losses and the largest eigenvalues of the Fisher in the training process of quantum neural networks with different numbers of layers using SGD.

同时，由于量子情形下最大特征值变化比较剧烈，可以看出最大特征值似乎与损失函数的下降速率有一定的相关性。为此，图 4.12 展示了训练过程中用差分近似的损失函数下降速率与最大特征值的比较，可以发现，部分情况下两者的峰和谷出现的时间基本一致，这说明是两者确实有一定的相关性。

对经典情形进行同样的处理，结果如图 4.13 所示，可以发现在经典情况下

最大特征值与损失函数的下降速率并没有表现出明显的相关性。

经典和量子 Fisher 信息矩阵的最大特征值在训练过程中表现出不同的行为，直接导致的结果是：一开始量子情形的最大特征值比经典对应值大，而在训练后比经典对应值小。这个结果与 Huembeli 等 (2021) 的实验中 Hessian 矩阵的最大特征值表现出的现象相似。Fisher 信息矩阵的特征值能反映损失曲面的性质，这意味经典模型和量子模型的损失曲面具有较大差异。同时，在相同的随机初始化条件和相同的算法设计下，训练过程中最大特征值行为的不同意味着两者的训练过程也是不同的，更是说明了经典和量子网络有着很大的差异性。



(a) 5 neurons **(b) 6 neurons**

(c) 7 neurons **(d) 8 neurons**

图 4.13 不同神经元个数下经典网络 SGD 下降训练过程中损失函数下降率的变化与 Fisher 信息矩阵最大特征值的变化。

Figure 4.13 The decreasing rates of the losses and the largest eigenvalues of the Fisher in the training process of classical neural networks with different numbers of neurons using SGD.

第 5 章　结论与展望

经典神经网络的 Fisher 信息矩阵的计算方法已经较为完备，但是这些理论难以直接引入量子情形。这是由于量子线路需要进行测量才能得到结果，并且该结果是有范围限制的，而经典没有这样的约束条件。对于二分类问题，对量子线路采取正交投影测量，可以将经典情形中常见的交叉熵形式引入量子情形。为了解决量子网络 Fisher 信息矩阵的计算问题并使经典和量子相对应，本文提出了在原网络结构的基础上在最后添加一层 softmax 层的解决方法，这种方法可以简单推广到所有的监督学习问题。通过对经典网络进行训练测试，我们发现这种网络结构的修改并不会影响 Fisher 信息矩阵最大特征值的变化趋势，这说明了该方法的可行性。进而，对经典和量子网络进行训练并通过该方法计算 Fisher 信息矩阵，将结果进行比较，最终可以得到三个经验性结论：

1. 随着参数的增多，经典网络通过增加多边形的边数来逼近圆形分类面，而量子网络是直接试图抓住圆形的特征。这说明量子网络引入的非线性性非常特别，使得它具有这样的能力。

2. 量子网络的损失曲面更为复杂，在经典情形下可以正常工作的批量梯度下降算法甚至可能不适用于结构简单的量子网络。该算法很可能使得量子模型在训练过程中收敛到不好的鞍点或极值点，所以量子模型的算法设计和参数选择需要更为小心。

3. 经典和量子模型的 Fisher 信息矩阵最大特征值在训练过程中具有直观意义，但是两者的意义不完全相同。当最大特征值在小范围波动时，意味着模型逼近了一个鞍点或极值点并逐渐收敛，这点对于经典和量子网络都成立。但是在还未收敛的训练过程中，经典和量子的行为完全不同：经典的 Fisher 信息矩阵最大特征值整体趋势为逐渐增大；量子的最大特征值则在大范围内波动，并在趋于收敛前会出现一个大幅度的降低。这意味着经典和量子网络的训练过程是不同的，损失曲面也有很大的差异。

这三个结论说明经典神经网络与量子网络有较大差异性，也意味着量子神经网络很可能是与经典网络不同的模型，而不是简单的量子推广。我们对经典网络的了解不足，对于量子世界的了解也不足。在这种情况下，将一个未知推向另

一个未知，又得到一个新的未知。这种新的未知能否帮助我们理解以前的未知还没有答案，但是这确实是一个极为有趣的话题。在这个过程中，必不可少的就是了解新的未知与旧的未知之间的关系，也需要判明新的未知又带来了多少问题。这一点，本文尝试通过修改网络结构将两者联系起来，起到了一定的效果，但是也抛弃了网络直接进行多分类的能力。尽管在这一次实验中，不同层数的量子神经网络的 Fisher 信息矩阵最大值在训练过程中表现出了相似的行为，并且该结果与 Huembeli 等 (2021) 的实验结果可以匹配，但是更多的重复实验可以帮助我们更充分地理解所有可能出现的情况。同时，在实验比较过程中，更多的问题还有待研究。不同的算法设计、不同的参数条件、不同的问题输入等等都有可能带来新的发现。本文希望这种比较引起更多的关注，带我们领略到更多有趣的现象。

附录 A　简化 Fisher 信息矩阵形式的证明

该部分的的推导参考了 Kunstner 等 (2019)。

命题 A.1. *当 $p(y|f(x,\theta))$ 为关于自然参数 f 的指数分布族时，* Fisher 信息矩阵具有简单形式

$$F(\theta) = \frac{1}{N}\sum_i - \left[\mathrm{J}_\theta f(x_i,\theta)\right]^{\mathrm{T}} \nabla_f^2 \log p(y_i|f(x_i,\theta)) \left[\mathrm{J}_\theta f(x_i,\theta)\right], \qquad \ldots (A.1)$$

其中 $\mathrm{J}_\theta f(x_i,\theta)$ 指 f 对 θ 求导的雅可比矩阵，∇_f 值对 f 进行求导。

证明. Fisher 信息矩阵的表达式为

$$F(\theta) = \frac{1}{N}\sum_i \mathrm{E}_{y\sim p_\theta(y|x_i)} \left[\nabla \log p_\theta(y|x_i) \nabla \log p_\theta(y|x_i)^{\mathrm{T}}\right], \qquad \ldots (A.2)$$

该式中的 ∇ 指对 θ 进行求导。由链式法则，该式可化为

$$F(\theta) = \frac{1}{N}\sum_i \mathrm{E}_{y\sim p_\theta(y|x_i)} \left[\left[\mathrm{J}_\theta f(x_i,\theta)\right]^{\mathrm{T}} \left[\nabla_f \log p_\theta(y|x_i) \nabla_f \log p_\theta(y|x_i)^{\mathrm{T}}\right] \left[\mathrm{J}_\theta f(x_i,\theta)\right]\right], \qquad \ldots (A.3)$$

其中 $\mathrm{J}_\theta f(x_i,\theta)$ 指 f 对 θ 求导的雅可比矩阵，∇_f 指对 f 进行求导。由于 $\mathrm{J}_\theta f(x_i,\theta)$ 与 y 无关，所以可以直接移出期望值之外

$$\begin{aligned} F(\theta) &= \frac{1}{N}\sum_i \left[\mathrm{J}_\theta f(x_i,\theta)\right]^{\mathrm{T}} \mathrm{E}_{y\sim p_\theta(y|x_i)} \left[\nabla_f \log p_\theta(y|x_i) \nabla_f \log p_\theta(y|x_i)^{\mathrm{T}}\right] \left[\mathrm{J}_\theta f(x_i,\theta)\right] \\ &= \frac{1}{N}\sum_i \left[\mathrm{J}_\theta f(x_i,\theta)\right]^{\mathrm{T}} \mathrm{E}_{y\sim p_\theta(y|x_i)} \left[\frac{1}{p_\theta^2(y|x_i)} \nabla_f p_\theta(y|x_i) \nabla_f p_\theta(y|x_i)^{\mathrm{T}}\right] \left[\mathrm{J}_\theta f(x_i,\theta)\right]. \end{aligned} \qquad \ldots (A.4)$$

利用求导的性质，上式可化为

$$\begin{aligned} F(\theta) = \frac{1}{N}\sum_i \left[\mathrm{J}_\theta f(x_i,\theta)\right]^{\mathrm{T}} \mathrm{E}_{y\sim p_\theta(y|x_i)} \Bigg[&\frac{1}{p_\theta(y|x_i)} \nabla_f^2 p_\theta(y|x_i) \\ &- \nabla_f^2 \log p_\theta(y|x_i)\Bigg] \left[\mathrm{J}_\theta f(x_i,\theta)\right]. \end{aligned} \qquad \ldots (A.5)$$

其中第一项的期望值

$$\begin{aligned} \mathrm{E}_{y\sim p_\theta(y|x_i)} \left[\frac{1}{p_\theta(y|x_i)} \nabla_f^2 p_\theta(y|x_i)\right] &= \int dy\, p_\theta(y|x_i) \frac{1}{p_\theta(y|x_i)} \nabla_f^2 p_\theta(y|x_i) \\ &= \nabla_f^2 \int dy\, p_\theta(y|x_i) = 0, \end{aligned} \qquad \ldots (A.6)$$

所以只剩下第二项

$$F(\theta)=\frac{1}{N}\sum_i\left[\mathrm{J}_\theta f(x_i,\theta)\right]^{\mathrm{T}}\mathrm{E}_{y\sim p_\theta(y|x_i)}\left[-\nabla_f^2\log p_\theta(y|x_i)\right]\left[\mathrm{J}_\theta f(x_i,\theta)\right]. \quad \dots (A.7)$$

由于 $\boldsymbol{y}$ 和 $\boldsymbol{f}$ 的维度相同，当 $p(y|f(x,\theta))$ 为关于自然参数 f 的指数分布族时，$p(y|f)$ 具有形式

$$p(y|f)=K\exp\left(\boldsymbol{y}^{\mathrm{T}}\boldsymbol{f}-b(f)+c(y)\right), \quad \dots (A.8)$$

其中 K 表示归一化系数，$b(f)$ 表示关于 f 的某一个函数，$c(y)$ 表示关于 y 的某一个函数。所以对 $\log p$ 求 f 的二阶导数与 y 无关

$$\nabla_f^2\log p(y|f)=\nabla_f^2\log K-\nabla_f^2 b(f)=\nabla_f^2\log p(y_i|f), \quad \dots (A.9)$$

求期望的项与 y 无关，得到最后的结果

$$F(\theta)=\frac{1}{N}\sum_i-\left[\mathrm{J}_\theta f(x_i,\theta)\right]^{\mathrm{T}}\nabla_f^2\log p(y_i|f(x_i,\theta))\left[\mathrm{J}_\theta f(x_i,\theta)\right]. \quad \dots (A.10)$$

□

命题 A.2. 对于 C-分类问题的神经网络模型，当选择损失函数为交叉熵(2.14)时，Fisher 信息矩阵具有形式

$$F(\theta)=\frac{1}{N}\sum_i\left[\mathrm{J}_\theta f(x_i,\theta)\right]^{\mathrm{T}}\left(\mathrm{diag}(\pi_i)-\pi_i\pi_i^{\mathrm{T}}\right)\left[\mathrm{J}_\theta f(x_i,\theta)\right], \quad \dots (A.11)$$

其中 π_i 指代对于不同输入 x_i，由 $\{p(y=c|f(x_i,\theta))\}$ 构成的列向量。

证明. 当选择损失函数为交叉熵时，对于分类问题，$p(y=c|f(x,\theta))$ 中 $y=c$ 表示着矢量 $\boldsymbol{y}$ 仅在第 c 个分量上为 1，在其余分量上为 0。选择

$$p(y=c|f(x_i,\theta))=\frac{e^{f_c}}{\sum_t e^{f_t}}, \quad \dots (A.12)$$

对应于 $K=\sum_i\exp(f_i)$，$b(f)=0$，$c(y)=0$。求对数后

$$\log p(y=c|f(x_i,\theta))=f_c-\log\left(\sum_t e^{f_t}\right), \quad \dots (A.13)$$

对 f 求二阶导数时，第一项消失，所以有

$$\nabla_f^2\log p(y=c|f(x_i,\theta))=-\nabla_f^2\log\left(\sum_t e^{f_t}\right). \quad \dots (A.14)$$

具体的求导过程为

$$\frac{\partial \log\left(\sum_t e^{f_t}\right)}{\partial f_i} = \frac{e^{f_i}}{\sum_t e^{f_t}}, \quad \ldots \text{(A.15)}$$

$$\frac{\partial^2 \log\left(\sum_t e^{f_t}\right)}{\partial f_i \partial f_j} = \frac{-e^{f_i} e^{f_j}}{\left(\sum_t e^{f_t}\right)^2}, \quad \ldots \text{(A.16)}$$

$$\frac{\partial^2 \log\left(\sum_t e^{f_t}\right)}{\partial f_i^2} = \frac{e^{f_i}\left(\sum_t e^{f_t}\right) - e^{2f_i}}{\left(\sum_t e^{f_t}\right)^2}. \quad \ldots \text{(A.17)}$$

带回式 (A.1) 就可以得到最后的结果

$$F(\theta) = \frac{1}{N}\sum_i \left[\mathrm{J}_\theta f(x_i,\theta)\right]^{\mathrm{T}} \left(\mathrm{diag}(\pi_i) - \pi_i \pi_i^{\mathrm{T}}\right) \left[\mathrm{J}_\theta f(x_i,\theta)\right] \quad \ldots \text{(A.18)}$$

其中 π_i 指代对于不同输入 x_i，由 $\{p(y=c|f(x_i,\theta))\}$ 构成的列向量。 □

参考文献

Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems [Z]. 2015.

Abbas A, Sutter D, Zoufal C, et al. The Power of Quantum Neural Networks [J]. arXiv e-prints, 2020: arXiv:2011.00027.

Amari S i. Differential Geometry of Statistical Models [M]. Springer New York, 1985: 11.

Amari S i. Natural Gradient Works Efficiently in Learning [J]. Neural Computation, 1998, 10: 251.

Broughton M, Verdon G, McCourt T, et al. TensorFlow Quantum: A Software Framework for Quantum Machine Learning [J]. arXiv e-prints, 2020: arXiv:2003.02989.

Byrd R, Lu P, Nocedal J, et al. A Limited Memory Algorithm for Bound Constrained Optimization [J]. SIAM Journal of Scientific Computing, 1995, 16: 1190.

Cybenko G. Approximation by Superpositions of a Sigmoidal Function [J]. Mathematics of Control, Signals and Systems, 1989, 2: 303.

Huembeli P, Dauphin A. Characterizing the Loss Landscape of Variational Quantum Circuits [J]. Quantum Science and Technology, 2021, 6: 025011.

Karakida R, Akaho S, Amari S i. Universal Statistics of Fisher Information in Deep Neural Networks: Mean Field Approach [C]//Chaudhuri K, Sugiyama M. Proceedings of Machine Learning Research: volume 89. PMLR, 2019: 1032.

Kunstner F, Hennig P, Balles L. Limitations of the Empirical Fisher Approximation for Natural Gradient Descent [C]//Wallach H, Larochelle H, Beygelzimer A, et al. Advances in Neural Information Processing Systems: volume 32. Curran Associates, Inc., 2019: 4133.

LeCun Y. Generalization and Network Design Strategies [J]. Connectionism in Perspective, 1989, 19: 143.

LeCun Y, Bengio Y, Hinton G. Deep learning [J]. Nature, 2015, 521: 436.

Liang T, Poggio T, Rakhlin A, et al. Fisher-Rao Metric, Geometry, and Complexity of Neural Networks [C]//Chaudhuri K, Sugiyama M. Proceedings of Machine Learning Research: volume 89. PMLR, 2019: 888.

Martens J. New insights and perspectives on the natural gradient method [J]. arXiv e-prints, 2014: arXiv:1412.1193.

Martens J, Grosse R B. Optimizing Neural Networks with Kronecker-Factored Approximate Curvature [C]//Bach F R, Blei D M. Proceedings of the 32nd International Conference on Machine Learning: volume 37. PMLR, 2015: 2408.

Nielsen M A, Chuang I L. Quantum Computation and Quantum Information: 10th Anniversary Edition [M]. Cambridge University Press, 2010: 17.

Pennington J, Worah P. The Spectrum of the Fisher Information Matrix of a Single-Hidden-Layer Neural Network [C]//Bengio S, Wallach H, Larochelle H, et al. Advances in Neural Information Processing Systems: volume 31. Curran Associates, Inc., 2018: 5410.

Pérez-Salinas A, Cervera-Lierta A, Gil-Fuster E, et al. Data Re-Uploading for a Universal Quantum Classifier [J]. Quantum, 2020, 4: 226.

Preskill J. Quantum Computing in the NISQ era and beyond [J]. Quantum, 2018, 2: 79.

Sagun L, Bottou L, LeCun Y. Eigenvalues of the Hessian in Deep Learning: Singularity and beyond [J]. arXiv e-prints, 2016: arXiv:1611.07476.

Sagun L, Evci U, Ugur Guney V, et al. Empirical Analysis of the Hessian of Over-Parametrized Neural Networks [J]. arXiv e-prints, 2017: arXiv:1706.04454.

Shalev-Shwartz S, Ben-David S. Understanding Machine Learning: From Theory to Algorithms [M]. Cambridge University Press, 2014: 273.

Stokes J, Izaac J, Killoran N, et al. Quantum Natural Gradient [J]. Quantum, 2020, 4: 269.

Vaswani A, Shazeer N, Parmar N, et al. Attention is All You Need [C]//Guyon I, Luxburg U V, Bengio S, et al. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach, California, USA: Curran Associates Inc., 2017: 6000.

致　谢

感谢黄东宸师兄在毕设期间给予了我莫大的帮助，感谢他同我一起讨论课题、讨论结果并给予我很多指导。感谢龙卓青师兄、陈佳林师兄以及许多同学毕设期间在方方面面提供的帮助。感谢杨义峰老师在我心中种下了物理学的种子，改变了我对世界的理解。感谢苏湛老师，他的课程设计和杨老师的教导帮助我一点点构建了与科学相关的思维方式。感谢叶静静带我走进计算机的世界，让我看到了另一种思维方式，感谢她使我对世间的诸多事物产生兴趣，并不断给予我前进的力量。感谢我的父母和我的姐姐，无论什么时候他们都在支撑着我。