

专业课程设计（1）

基于人工智能求解量子力学问题的设计

学 院： 物理工程学院
姓 名： 刘宇航
学 号： 151428020212
班 级： 光电科 152 班
指导教师： 左正伟

二〇一八年六月

专业课程设计任务书

题目	基于人工智能求解量子力学问题的设计
学生班级	光电 15 级
学生姓名	李会英、刘宇航、徐莹莹、张玉岭、刘壮壮
主要设计内容 (工作)	<ol style="list-style-type: none">1. Python 语言常用功能;2. 任意势阱下薛定谔方程的数值求解;3. 机器学习的常用算法;4. 利用薛定谔方程的波函数训练神经网络, 学习和预测其他势阱下的波函数。
主要技术指标 (或研究目标)	<ol style="list-style-type: none">1. 学会 Python 语言的常用功能。2. 学会机器学习的常用算法;3. 学会人工智能求解物理问题
进度计划	<ol style="list-style-type: none">1. 第 1-2 周查资料熟悉量子力学、机器学习、Python 语言等基础知识;2. 第 3-4 周进行数值模拟、神经网络的设计、训练和学习;3. 第 5-6 周总结设计, 进行归纳总结, 并书写总结报告。

指导教师: 左正伟

填表时间: 2018 年 5 月 22 日

摘要： 简述程序语言 python 的背景，神经网络的结构及原理，人工智能学习库 TensorFlow，推导薛定谔方程的由来，灵活运用人工智能解决用薛定谔方程在一维空间中找到运动的具有势能的粒子的量子基态问题。

关键词： python 神经网络 人工智能 TensorFlow 量子基态

一、 Python 语言的简史

近几年来，随着“大数据”，“人工智能”，“云计算”等等科技的兴起，python 走入我们的视野，以至于我们认为 python 是刚兴起的一门语言，其实不然，其诞生于 1989 年。

python 的创始人：Guido van Rossum；之所以选中 python 作为该编程语言的名字，是因为他是一个叫 Monty Python 的喜剧团体的爱好者。

作者前身也是 C++ 程序员，之前也参加设计了一种叫 ABC 的教学语言，就 Guido 本人看来，ABC 这种语言非常优美和强大，是专门为非专业程序员设计的。但是 ABC 语言并没有成功，究其原因，Guido 认为是其非开放造成的（相对封闭的开发语言、扩展性、推广性相对不太成功）。Guido 决心在 Python 中避免这一错误。同时，他还想实现在 ABC 中闪现过但未曾实现的东西。

1991 年，第一个 Python 编译器诞生。它是用 C 语言实现的，并能够调用 C 语言的库文件。从一出生，Python 已经具有了：类，函数，异常处理，包含表和词典在内的核心数据类型，以及模块为基础的拓展系统，随着时间的推进，经过几十年的发展，语言体系日益完善[1]。

Python 是一门兼具简单与功能强大的编程语言，常常被称为“胶水语言”，能够把其他语言制成的各种模块（尤其 c/c++）很轻松地结合在一块。专注于如何解决问题、自由开放的社区环境以及丰富的第三方库，各种 Web 框架、爬虫框架、数据分析框架、机器学习框架应有尽有，拿来即用。Python 里面有很多集成好的标准库：比如科学计算库“numpy”、绘图库“matplotlib”，深度学习库“tensorflow”，操作 csv 文件库“csv”等等，最让人兴奋的是这些库包，都很容易理解掌握，即使你不会编写程序，看一下教程几个小时就能编写一个很好的程序满足科学计算的需求。

Python 是纯粹的自由软件， 源代码和解释器 CPython 遵循 GPL(GNU General Public License)协议。Python 语法简洁清晰， 特色之一是强制用空白符 (white space)作为语句缩进。在运算速度上也不用担心， 虽然比不上运算狂人“c”， 但是运算速度也不会慢很多， 随着设备性能的提升， 运算速度越来越快， 几乎感受不到它们之间运算的差距。

综上用 python 来解决我们物理学里面的各种科学问题再好不过了， 很简单方便。

二、 神经网络

1. 人工神经网络的来源

人类仿造蝙蝠、海豚等动物的超声波定位能力发明了了雷达；根据萤火虫的生物发光现象， 发明了无需通电， 无电磁干扰的高效冷光灯；仿照生物神经系统， 计算机科学家发明了人工神经网络。

人工神经网络 (artificial neural network, ANN)， 简称神经网络 (neural network, NN)。是一种模仿生物神经网络(动物的中枢神经系统， 特别是大脑)的结构和功能的数学模型或计算模型， 用于对函数进行评估或近似。

2. 人工神经网络的基本结构

人的神经网络

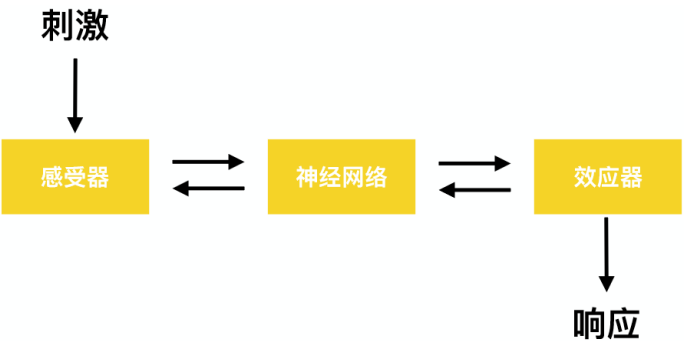


图 2-2-1 人的神经网络

人工神经网络

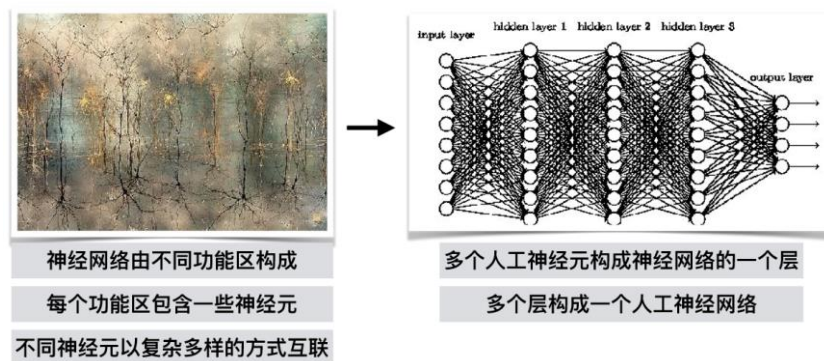


图 2-2-2 人工神经网络

生物神经元的基本结构

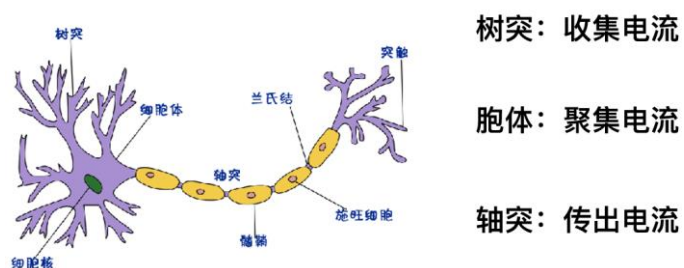


图 2-2-3 生物神经元的基本结构

模拟一个生物神经元

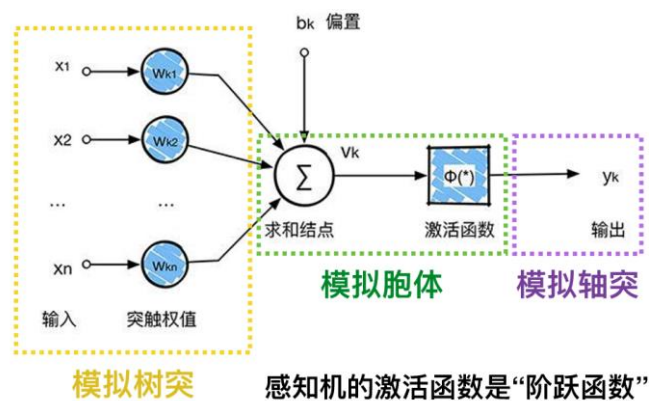


图 2-2-4 模拟一个生物神经元

X_1 、 X_2 、 X_3 、...、 X_n 输入信号

W_1 、 W_2 、 W_3 、...、 W_n 给输入信号加权参数

B_k 为偏置

ϕ (*) 为传递函数 (激活函数)，通常为非线性函数。 Y_k 为神经元输出

数学表示式： $Y_k = \phi (*) (W_k X_k + B_k)$

3. 人工神经网络的性质与能力

- 1) 非线性 (nonlinearity)
- 2) 输入输出映射 (input-output mapping)
- 3) 自适应性 (adaptivity)
- 4) 证据响应 (evidential response)
- 5) 上下文信息 (contextual information)
- 6) 容错性 (fault tolerance)

非线性：人工神经元可以是线性的，也可以是非线性的。非线性神经元构成的神经网络也是非线性的。这一特性可以用来拟合非线性物理过程。

输入输出映射：神经网络通过建立输入输出映射来从样例中学习。

自适应性：神经网络具有调整自身权值以适应外界环境变化的能力。在一个特定条件下接受训练的神经网络在环境条件变化不大时可以很容易地重新训练。

自适应性不一定总能导致鲁棒性。

证据响应：在模式分类问题中，神经网络可以设计成不仅提供选择哪一个特定模式的信息，还提供关于决策置信度信息。

上下文信息：神经网络的特定结构和激发状态代表知识。网络中每一个神经元都受网络中所有其它神经元全局活动的影响。因此，神经网络很自然的可以处理上下文信息。

知识：人或机器存储起来以备使用的信息或模型，用来对外部世界作出解释、预测和适应的反映。

容错性：神经网络具有鲁棒计算的能力，其性能在不利的运行条件下是逐渐下降的。即部分神经元的损坏不会造成灾难性的失败。

非分布式表征：

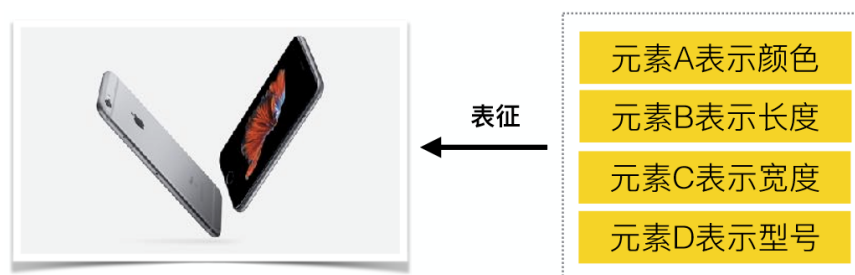


图 2-3-1 非分布式表征

一个对象使用一个特征集合表述。集合中每一个元素都是独立的， 存储特定信息的。丢失了一个信息就相当于丢失了一个特征。

分布式表征：

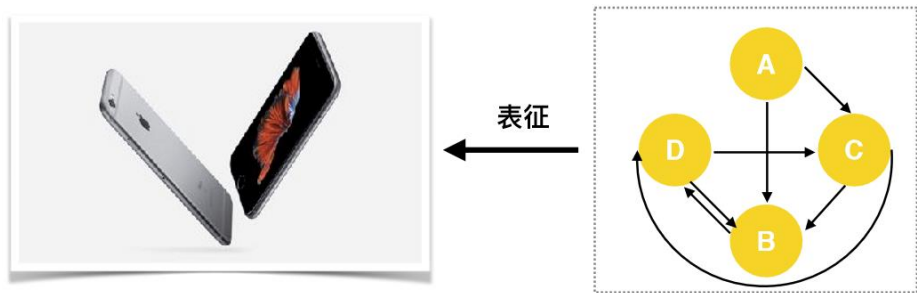


图 2-3-2 分布式表征

一个对象使用用一组无明显界限区分的特征进行表达。一个特征存 储在多个元素内，同时每个元素也可以参与不同的特征表达。部分元素丢失，通常不影响表达能力。神经网络中信息是分布式表征的。

ANN 与计算机系统的相似性：

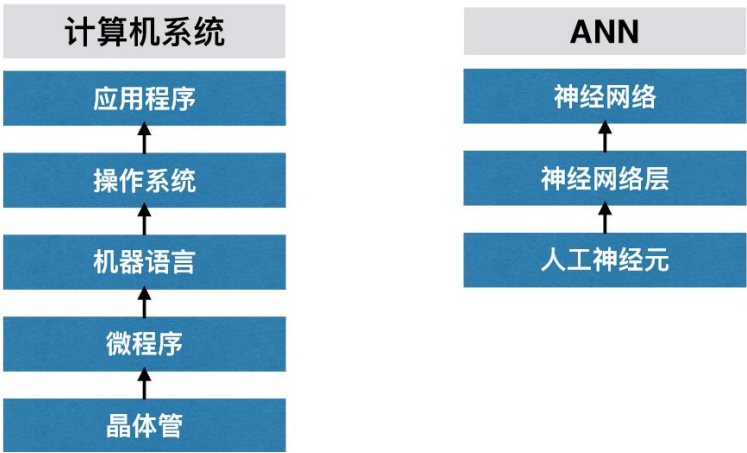


图 2-3-3 ANN 与计算机系统的相似性

三、 TensorFlow

(一) TensorFlow 的简介

TensorFlow 是由 Google Brain Team 开发的使用数据流图进行数值计算的开源机器学习库。TensorFlow 的一大亮点是支持异构设备分布式计算(heterogeneous

distributed computing)。这里的异构设备是指使用 CPU、GPU 等计算设备进行有效地协同合作。Google Brain Team 与 DeepMind 是独立运行相互合作的关系。TensorFlow 拥有众多的用户，除了 Alphabet 内部使用外，ARM、Uber、Twitter、京东、小米等众多企业均使 TensorFlow 作为机器学习的工具。[2]

（二）TensorFlow 优点

1. 可自行设计神经网络结构；
2. 不需要通过反向传播求解梯度，TensorFlow 支持自动求导；
3. 通过 C++编写核心代码，简化了线上部署的复杂度(通过 SWIG 实现 Python，Go 和 JAVA 接口)；
4. TensorFlow 中内置 TF.Learn 和 TF.Slim 等组件，并兼容 Scikit-learn estimator 接口(evaluate、grid、search、cross、validation)；
5. 数据流式图支持自由的算法表达，可实现深度学习以外的机器学习算法；
6. 可写内层循环代码控制计算图分支的计算，可将相关的分支转化为子图并执行迭代计算；
7. 可进行并行设计，充分利用硬件资源。
8. 具有灵活的移植性，编译速度较快；

（三）神经网络的算法

最速下降法（Steepest Descent），也称为梯度下降法（Gradient Descent），是一个一阶最优化算法。通过对函数某一点对应的梯度的反方向以规定步长距离进行迭代搜索以找到函数的局部极小值。[3]

迭代规则： $\mathbf{W}(n+1)=\mathbf{W}(n)-\alpha\nabla J(\mathbf{W})$

其中 α 一般被称为步长（Stepsize）或学习率（Learning-rate），用来控制迭代速率。

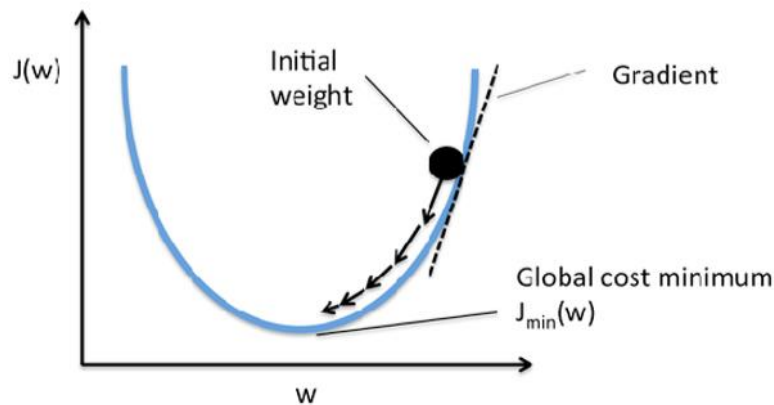


图 3-3-1 最速下降法

- 1) 导数符号的方向就是下降的反方向
- 2) 越接近极小值迭代步长越小
- 3) 迭代步长太大可能导致无法下降
- 4) 往往只能得到极小值，而非最小值

(四) 激活函数

激活函数（Activation functions）对于人工神经网络模型去学习、理解非常复杂和非线性的函数来说具有十分重要的作用。它们将非线性特性引入到我们的网络中。其主要目的是将 A-NN 模型中一个节点的输入信号转换成一个输出信号。该输出信号现在被用作堆叠中下一个层的输入。[4]

而在 A-NN 中的具体操作是这样的，我们做输入（X）和它们对应的权重（W）的乘积之和，并将激活函数 $f(x)$ 应用于其获取该层的输出并将其作为输入馈送到下一个层。不同的激活函数是用来实现不同的信息处理能力，神经元的变换函数反映了神经元输出与其激活状态之间的关系。

如果我们不运用激活函数的话，则输出信号将仅仅是一个简单的线性函数。线性函数一个一级多项式。现如今，线性方程是很容易解决的，但是它们的复杂性有限，并且从数据中学习复杂函数映射的能力更小。一个没有激活函数的神经网络将只不过是一个线性回归模型（Linear regression Model）罢了，它功率有限，并且大多数情况下执行得并不好。我们希望我们的神经网络不仅仅可以学习和计算线性函数，而且还要比这复杂得多。同样是因为没有激活函数，我们的神经网络将无法学习和模拟其他复杂类型的数据，例如图像、视

频、音频、语音等。这就是为什么我们要使用人工神经网络技术，诸如深度学习（Deep learning），来理解一些复杂的事情，一些相互之间具有很多隐藏层的非线性问题，而这也可以帮助我们了解复杂的数据。

非线性函数是那些一级以上的函数，而且当绘制非线性函数时它们具有曲率。现在我们需要一个可以学习和表示几乎任何东西的神经网络模型，以及可以将输入映射到输出的任意复杂函数。神经网络被认为是通用函数近似器

（Universal Function Approximators）。这意味着他们可以计算和学习任何函数。几乎我们可以想到的任何过程都可以表示为神经网络中的函数计算。

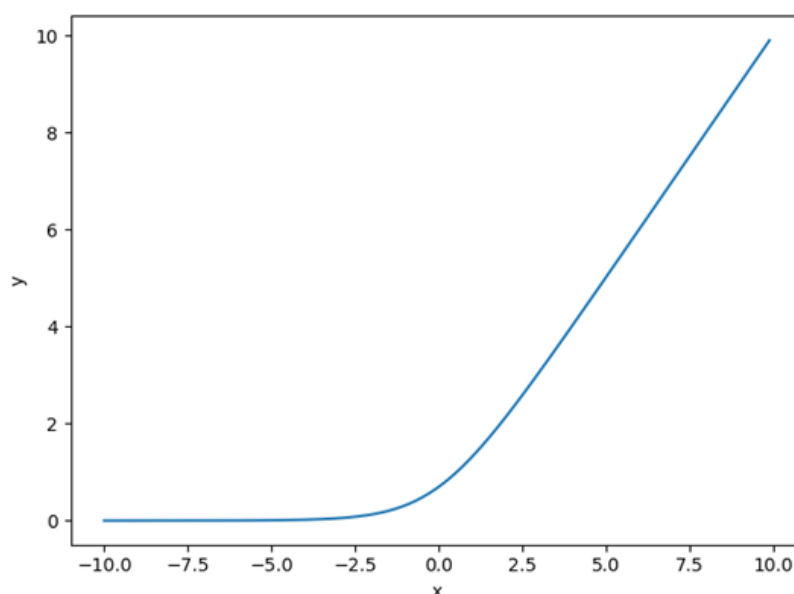
而这一切都归结于这一点，我们需要应用激活函数 $f(x)$ ，以便使网络更加强大，增加它的能力，使它可以学习复杂的事物，复杂的表单数据，以及表示输入输出之间非线性的复杂的任意函数映射。因此，使用非线性激活函数，我们便能够从输入输出之间生成非线性映射。

激活函数的另一个重要特征是：它应该是可以区分的。我们需要这样做，以便在网络中向后推进以计算相对于权重的误差（丢失）梯度时执行反向优化策略，然后相应地使用梯度下降或任何其他优化技术优化权重以减少误差。

Softplus 激活函数

Softplus 激活函数数学表达式： $y = \log(\exp(x) + 1)$.

其图像如下图



3-4-1 softplus 激活函数的图像

使用激活函数的作用：

所有激活函数输入输出的维度是一样的，即输入的是 $n \times n$ 列表，输出的依然是 $n \times n$ 列表。这保证了使用激活函数不会对我们的数据造成缺损，此外由于神经网络只是把输入信号加权偏置，通俗点来说只是把数据线性化，如果只进行线性变换，那么即使是多层的神经网络，依然只有一层的效果，而为了可以使得神经网络可以拟合任意一个函数，需要将线性转化为非线性，`softplus` 激活函数会使一部分神经元的输出为 0，这样就造成了网络的稀疏性，并且减少了参数的相互依存关系，缓解了过拟合问题的发生。

四、 用深度学习方法解决量子力学问题

（一）波函数

量子力学是反映微观粒子（分子、原子、原子核、基本粒子等）运动规律的理论，他是 20 世纪 20 年代在总结了大量实验事实和旧量子理论的基础上建立起来的，随着量子力学的出现，人类对于物质微观结构的认识日益深入，从而能较深刻地掌握物质的物理和化学的性能及其变化的规律，为利用这些规律于实际开辟了广阔的途径，原子核、固体等的性质都能从以量子力学为基础的现代理论中得到阐明，量子力学不仅是物理学中的基础理论之一，而且在化学、材料学、生物学和宇宙学等有关学科和许多近代技术中心也得到了广泛的应用。

黑体辐射和光电效应等现象使人们发现了光的波粒二象性，在此启示下，开始认识到微观粒子的波粒二象性。以微观粒子的波粒二象性，引入描述微观粒子状态的波函数，建立非相对论量子力学的基本方程——薛定谔方程

（Schrodinger equation）。

为了表示微观粒子（简称为粒子）的波粒二象性，可以用平面波来描写自由电子，平面波的频率和波长与自由粒子的能量和动量由德布罗意关系联系起来，平面波的频率和波矢都是不随时间或空间位置改变的，这和自由粒子的能量和动量不随时间或空间位置改变而相对应，如果粒子受到随时间或空间位置变化的力场的作用，其动量和能量不再是常量，这时粒子就不能用平面波来描

写，而必须用更加复杂的波来描写，在一般情况下，我们用一个波函数表示描写粒子的波，并这个函数为波函数，它是一个复数，描写自由粒子的德布罗意平面波是波函数的一个特例。

考察粒子的衍射实验，如果入射电子流的强度很大，即单位时间内有许多电子被晶体反射，则照片上很快就出现衍射图样。如果入射电子流强度很小，电子一个一个地从晶体表面反射，这时照片上就出现一个一个的点子，显示出电子的粒子性，这些点在照片的位置并不都是重合在一起的，开始时，它们看起来似乎是毫无规则地散布着，随着时间的延长，电子数目逐渐增多，它们在照片上的分布就形成了衍射图像，显示电子的波动性。

由此可见：实验所显示的电子的波动性是许多电子在同一实验中的统一结果，或者是一个电子在许多次相同实验中统计的结果。波函数正是描写粒子的这种行为而引进来的。波恩就是在这个基础上，提出了波函数的统计解释，即：波函数在空间中某一点的强度（振幅绝对值的平方）和在该点找到粒子的概率成比例。描写粒子的波是概率波。波函数的真正意义：不是什么实在的物理量，只不过是描述粒子在空间的概率分布的概率而已。

粒子被晶体反射后，描写粒子的波发生衍射，在图片中的衍射图样中，有许多衍射很大和衍射很小的地方。在衍射很大的地方，波的强度大，每个粒子投射到这里的概率也大，因而投射到这里的粒子多。相反衍射很小的地方，波的强度很小或者等于 0，粒子投射到这里的概率很小或等于 0，因而投射到这里的粒子很少或没有。

知道描写微观体系的波函数，由波函数振幅绝对值的平方，就可以得出粒子在任意一点出现的概率。波函数（也称为概率波）描写体系的量子状态（简称状态或态）。在量子力学中，微观粒子具有波粒二象性，粒子的坐标和动量不可能同时具有确定值。当粒子处于某一量子状态时，它的力学量（坐标，动量）一般有许多可能值，这些可能值各自以一定的概率出现，这些概率都可以由波函数得出。

由于粒子必定要在空间某一点出现，所以粒子在空间各点出现的概率综和为 1，因而粒子在空间各点出现的概率只决定波函数在空间各点的相对强度，而不决定强度的绝对大小。

对于一般的情况，如果 Ψ_1 和 Ψ_2 是体系的可能状态，那么，它们的线性叠加也是这个体系的一个可能状态这就是量子力学的叠加原理。

$$\Psi = c_1\Psi_1 + c_2\Psi_2 \quad (c_1, c_2 \text{ 是复数}) \quad (4-1-1)$$

上式， Ψ 表示为两个态 Ψ_1 和 Ψ_2 的线性叠加，推广到更一般的情况，态 Ψ 可以表示为许多态 $\Psi_1, \Psi_2, \dots, \Psi_n, \dots$ 的线性叠加。

$$\Psi = c_1\Psi_1 + c_2\Psi_2 + \dots + c_n\Psi_n + \dots = \sum_n c_n\Psi_n \quad (4-1-2)$$

$c_1, c_2, \dots, c_n, \dots$ 为复数，这时叠加态原理描述如下：当 $\Psi_1, \Psi_2, \dots, \Psi_n, \dots$ 是体系的可能状态时，它们的线性叠加 4-1-2 式也是体系的一个可能状态。[6]

(二) 薛定谔方程

在经典力学中，当质点在某一时刻的状态为已知时，由质点的运动方程就可以求出以后状态任一时刻质点的状态。在量子力学中也是这样，当微观粒子在某一时刻的状态为已知时，以后时刻粒子所处的状态也要由一个方程来决定，称之为薛定谔方程。[6]

由于建立的是描写波函数随时间变化的方程，因此它必须是波函数应满足的含有对时间微商的微分方程。此外还必须满足下面两个条件：

①方程是线性的，即如果 Ψ_1 和 Ψ_2 都是这方程的解，那么 Ψ_1 和 Ψ_2 的线性叠加 $a\Psi_1+b\Psi_2$ 也是方程的解。这是因为根据态叠加原理，如果 Ψ_1 和 Ψ_2 都是粒子可能的状态，那么 $a\Psi_1+b\Psi_2$ 也应是粒子可能的状态。

②这个方程的系数不应包含状态的参量，如动量、能量等，因为方程的系数如含有状态的参量，则方程只能被粒子的部分状态所满足，而不能被各种可能的状态所满足。

平面波描写自由粒子的波函数：

$$\Psi(\mathbf{r}, t) = A e^{\frac{i}{\hbar}(\mathbf{p} \cdot \mathbf{r} - Et)} \quad (\hbar \text{ 为普朗克常量}) \quad (4-2-1)$$

对时间求偏微商得到

$$\frac{\partial \psi}{\partial t} = -\frac{i}{h} \mathbf{E} \psi \quad (4-2-2)$$

对坐标求二次偏微商

$$\frac{\partial^2 \psi}{\partial x^2} = -\frac{p_x^2}{h^2} \psi \quad (4-2-3)$$

同理有

$$\frac{\partial^2 \psi}{\partial y^2} = -\frac{p_y^2}{h^2} \psi \quad (4-2-4)$$

$$\frac{\partial^2 \psi}{\partial z^2} = -\frac{p_z^2}{h^2} \psi \quad (4-2-5)$$

将式 4-2-3, 4-2-4, 4-2-5 相加得到

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} + \frac{\partial^2 \psi}{\partial z^2} = \nabla^2 \psi = -\frac{p^2}{h^2} \psi \quad (4-2-6)$$

利用自由粒子的能量和动量的关系式:

$$\mathbf{E} = \frac{p^2}{2m} \quad (4-2-7)$$

式中 m 为粒子的质量, 比较上式, 得到自由粒子波函数所满足的微分方程:

$$i \hbar \frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m} \nabla^2 \psi \quad (4-2-8)$$

设粒子在力场中的势能为 $U(r)$, 在这种情况下, 粒子的动能和动量的关系:

$$\mathbf{E} = \frac{p^2}{2m} + U(r) \quad (4-2-9)$$

上式两边乘以 $\Psi(r, t)$, 得到 $\Psi(r, t)$ 所满足的微分方程

$$i \hbar \frac{\partial \psi}{\partial t} = -\frac{\hbar^2}{2m} \nabla^2 \psi + U(r) \Psi \quad (4-2-10)$$

式 (4-2-10) 称为薛定谔波动方程, 或薛定谔方程, 也成为波动方程。

(三) 定态薛定谔方程

如果 $U(r)$ 与时间无关, (4-2-10) 可以用分离变量法进行求解, 考虑这一方程的一种特解:

$$\Psi(r, t) = \Psi(r) f(t) \quad (4-3-1)$$

方程（4-3-1）的解可以表示为许多这种特解之和，将（4-3-1）代入方程（4-2-10）中，并把方程两边用 $\Psi(r)f(t)$ 去除，得到

$$\frac{i\hbar}{f} \frac{df}{dt} = \frac{1}{\psi} \left[-\frac{\hbar^2}{2m} \nabla^2 \psi + U(r)\Psi \right] \quad (4-3-2)$$

因为这个等式的左边只是 t 的函数，右边只是 r 的函数，而 t 和 r 是相互独立的变量，所以只有当两边都等于同一常量，等式才能被满足。以 E 表是这个常量，则由等式左边等于 E ，有

$$i\hbar \frac{df}{dt} = E f \quad (4-3-3)$$

由等式右边等于 E ，有

$$-\frac{\hbar^2}{2m} \nabla^2 \psi + U(r)\Psi = E\Psi \quad (4-3-4)$$

上式就是定态薛定谔方程。

（四）解决具体问题

对于用薛定谔方程在一维空间中找到运动的具有势能的粒子的量子基态这一问题？大家都不陌生，然而现在要用一种全新的思维来处理这个问题，运用科技的成果——深度学习。

简单点来说：首先用程序生成大量的随机势能，通过数值计算出各势能下的波函数，利用薛定谔方程给出的能量泛函下降法，将势能和波函数划分为用于训练和用于验证两类数据，保存起来。

然后建立一个三层神经网络，取出上述用于训练的势能 U_i 作为输入值，进行大量的训练，不断更新权重 W_i 和偏差 B_i ，激活函数使用 `softplus`，将神经网络输出的值作为训练所得波函数 ψ'_i ，使用随机梯度下降法，将真实的波函数 ψ_i 作为基值， ψ'_i 与之对比，迭代多次找到差值最小的，进而最小化成本，达到最佳拟合。

经过多次学习，最后将用于验证的势能 U_i 作为输入值，将预测的波函数 ψ'_i 与真实的波函数 ψ_i 全部绘图，观察图像。

我们讨论的体系势能在无限远处都是无限大，波函数在无限远处为零，这

个体系使得体系的能级是分立的，属于束缚态。

在程序中生成三种类型的势能：阶跃函数势能，分段线性函数势能，傅里叶级数（由正弦波构建的图形），随着程序的进行，图像上下波澜。

一旦产生了势能，程序就会找到它的量子基态（最低能量状态）。最终的结果是量子态是一个波函数（一个概率波），显示粒子最可能被发现的位置（如果它被光子或其他粒子探测到）。

对于单位长度空间中单位质量的粒子，用定态薛定谔方程（4-3-4）推导出能量的表达式：

$$E = \int_0^1 -\frac{1}{2} \frac{d^2\varphi}{dx^2} \varphi(x) + V(X)\varphi(X)^2 dx \quad (4-4-1)$$

其中 $\psi(x)$ 是波函数， $V(x)$ 是势函数。积分的第一项表示粒子的动能对总量的影响，第二项表示其势能的影响。如果 $V(x) = 0$ ，则最小能量为 $\pi^2/2$ 或约 4.9 个单位。

通过将 x 离散成一组值 $\{x_0, x_1, \dots, x_n\}$ （这里， $n = 300$ ），并将每个 $\psi(x_i)$ 作为一个独立变量，我们可以使用 TensorFlow 的梯度下降算法来寻找一系列逐步调整每一步都会减少能量的波函数。具体来说：使用梯度下降法找到函数的局部极小值，向函数上当前点对于梯度（或者是近似梯度）的反方向的规定步长距离点进行迭代搜索，从简单的正弦波开始，经过许多迭代（20000），就可以达到与真实基态的紧密近似。

基态解在随机势能（蓝色）中的逐渐演变。较早的迭代更加趋近正弦；该解决方案最终变得更加局限在单元 30 周围。这组数据——数千个势能函数图及其相应的数值评估基态——被分成训练数据（14400 个势能和基态）和验证数据（3600 对），以供给程序中的下一步使用。

用合理的方法设置了最简单的网络：2 个隐藏层和一个输出层，全部由 softplus 函数激活，每层都有 n 个（实际上 $n-1$ 或 299 个，因为边界是固定在 0）节点。

神经网络的结构图

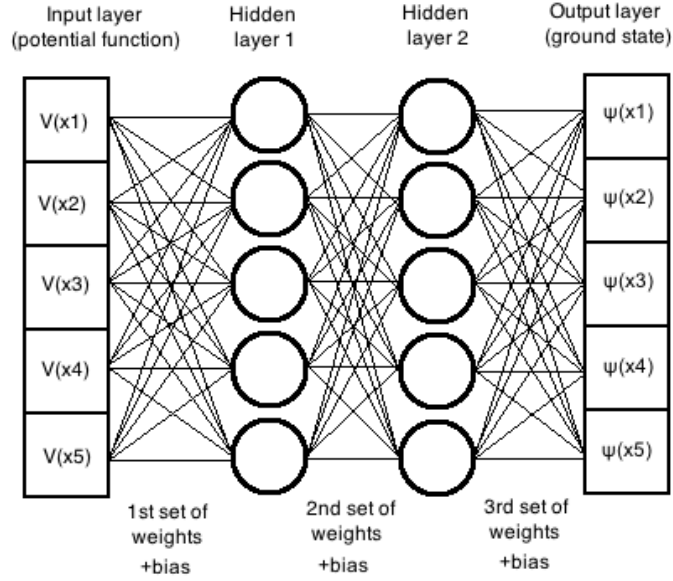


图 4-4-2 神经网络的结构图

下面我们可以将输入层视为一个 n 维向量。势能函数的每个值都被加权偏置，然后添加到进入神经元的输入中。对于隐藏层中的每个神经元都重复这一点。这些权重（上图中的每一行）都有不同的值。因此，权重包括 $n \times n$ 张量。每个神经元都会得到一个偏差（一个增加到输入的值），与来自势能函数的内容无关，因此是另一个 n 维向量。接下来，将 softplus 函数 $s(y) = \ln(1 + \exp(y))$ 应用于每个节点的值，以便将非线性引入解。这个输出成为输入到下一个隐藏层的输入，该输入又被馈送到输出层，从而为给定的输入势能产生预测的基态。在数学上，它可以简单地表达为：

$$h_1 = s(W_1 V_{(x_i)} + b_1) \quad (4-4-3)$$

$$h_2 = s(W_2 h_1 + b_2) \quad (4-4-4)$$

$$\psi_{(x_i)} = s(W_3 h_2 + b_3) \quad (4-4-5)$$

在算法的每一步中，对于 14400V's 中的每一个，应用相同的权重 w_i 和偏差 b_i 来生成 14400 个势能 ψ' s。然后将这些与之前解的“实际” ψ 进行比较，以计算成本函数：

$$C(W, b) = \sum_{k=1}^N \int_0^1 (\psi_K^{actual}(x) - \psi_k^{predicted}(x))^2 dx \quad (4-4-6)$$

W 和 b 是我们将更新的变量，以便随着梯度下降算法的进行而最小化成本。

因为有这么多（299*299 张量的 3 层和 299 个偏差=268203 个独立变量的向量），所以可能使成本最小化的解决方案过度拟合。在这种情况下，在每对训练数据中实际上有 299 个预测值（基态的每个 x 值），产生与独立权重的 25:1 预测的比率，这使得过拟合的可能性更小。作为对过度配合的进一步检查，通过网络提供验证数据，而不会降低其成本。如果验证数据产生与培训数据相当的成本，这将作为一个独立的验证，我们的网络实际上解决了一维薛定谔方程的一般问题。这里另一个微妙之处在于，算法首先更快地更新权重和偏差，然后随着算法的进行更慢地更新，但除此之外，我们已经绘制了整个过程。由于处理数据非常多，迭代次数也很多，使用随机梯度下降法对计算成本函数（具体思路是在更新每一参数时都使用一个样本来进行更新），这样就可以加速训练的速度，但也有一些缺点，准确度会下降，并不是全局最优，但迭代次数（20000）多，数据量也很大，所以这些可以忽略不计。

经过 20000 次迭代后，经过预测和验证，选择了一些好的例子和一些差的例子。势能是绿色的，实际的解决方案是蓝色的，网络的预测是红色的。训练基态和网络预测之间的良好匹配：如下图

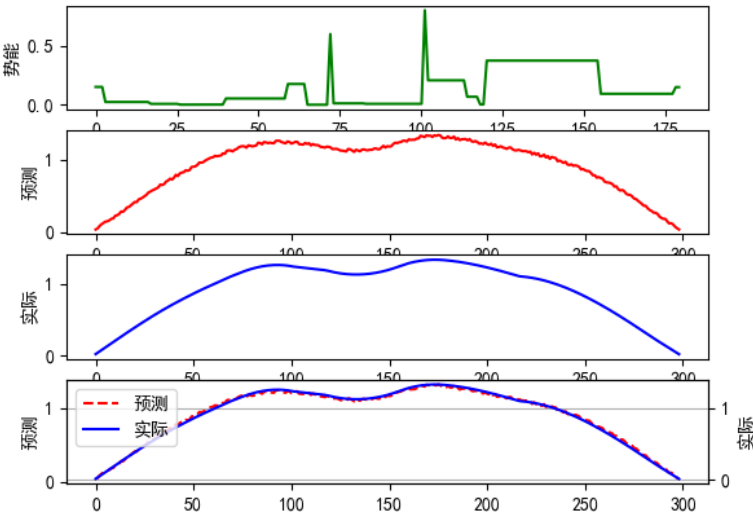


图 4-4-7 良好匹配

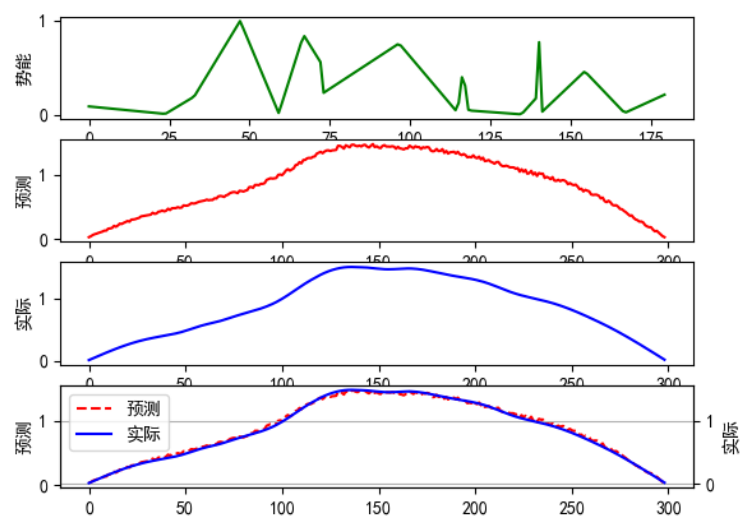


图 4-4-8 良好匹配

训练基态与网络预测的平均匹配：如下图

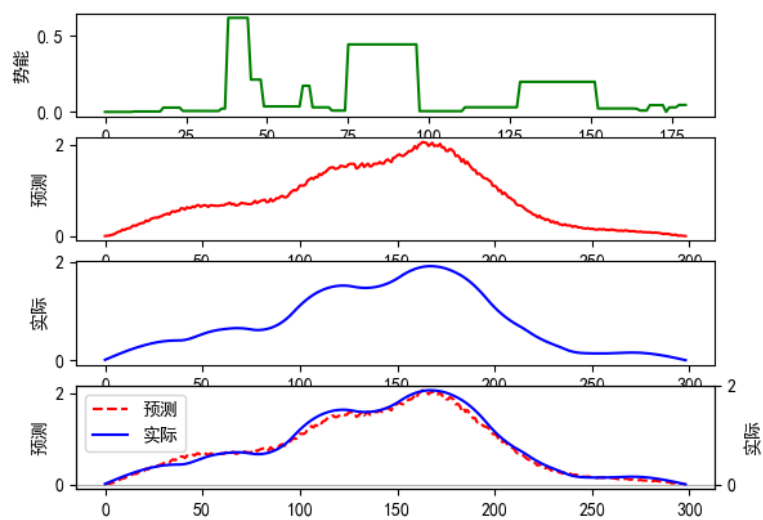


图 4-4-9 平均匹配

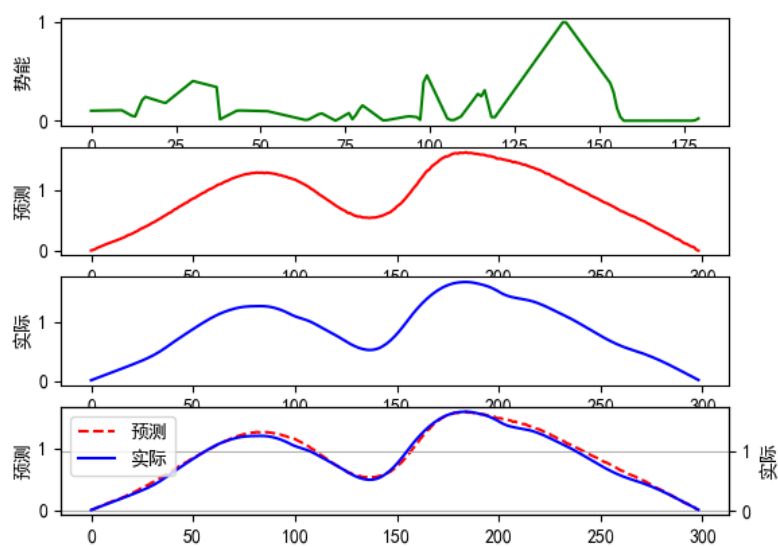


图 4-4-10 平均匹配

训练基态和网络预测之间发现的最差匹配：如下图

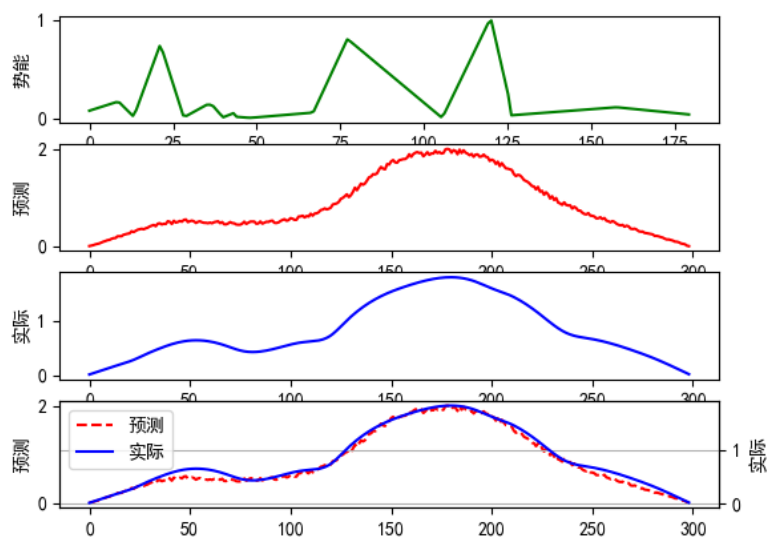


图 4-4-11 最差匹配

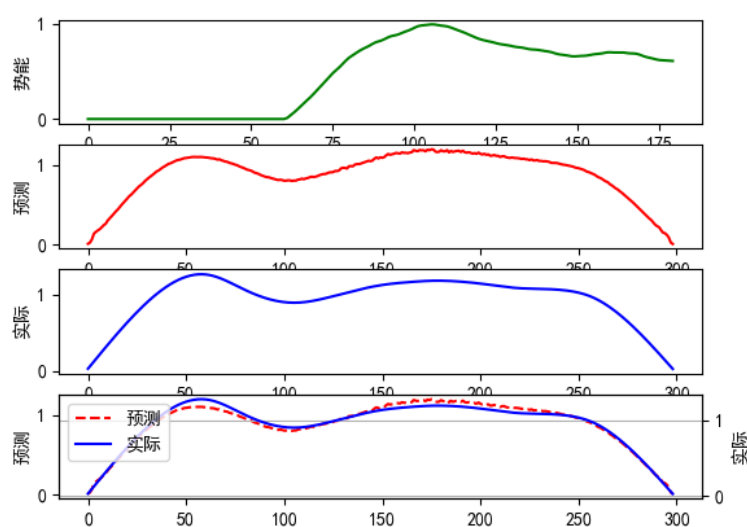


图 4-4-12 最差匹配

从上面几幅图可以很清晰看出：即便是最差匹配，通过程序所得的波函数与实际的波函数相差结果并不大，证明程序拟合的结果是比较完美的，且运算速度很快，可以广泛使用。

（五）小结

本次基于人工智能解决量子力学问题的课程设计是很有趣的，从零到有的过程让人兴奋，学会了很多知识能力，快速掌握一门程序语言（python）的基础内容：列表元组字典的操作、条件控制循环语句、迭代器与生成器、数据结构、面向对象等等；学习了 python 几个有趣的库：matplotlib（绘图）、numpy（科学计算库）、TensorFlow（人工智能深度学习）等等；学习了神经网络的原理结构性质，深度学习的实际操作流程；最重要的巩固了量子力学的基础知识，并将其结合起来，用深度学习解决用薛定谔方程在一维空间中找到运动的具有势能的粒子的量子基态这一问题。刚开始老是百度搜索答案方法，想快速解决问题，回头发现还不如自己动手自己动脑，丰衣足食，一起都需脚踏实地，稳步前行，不努力就不会有收获，知识是留给努力学习的人，最后响应了一句话：踏踏实实前进、知识就是力量、科学技术是第一生产力。

五、 参考文献

- [1] Magnus 著. python 基础教程（第二版）[M]. 北京： 司维， 曾军崑译文, 人民邮电出版社 2010. 7
- [2] 极客学院 Wiki 团队译编. [TensorFlow 官方文档中文版](#) (谷歌第二代人工智能学习系统) [Z]. 极客学院出版 2015. 11. 9.
- [3] 韩力群编著. 人工神经网络理论、设计及应用（第二版）[M]. 北京： 化学工业出版社， 2007. 7 .
- [4] （加）海金（HayKin, S）著. 神经网络与机器学习（原书第三版）[M]； 申富饶等译. 北京： 机械工业出版社， 2011. 1
- [5] the NumPy community 著. [NumPy Reference Release 1.14.2](#) [Z]. April 16, 2018.
- [6] 周世勋原著. 量子力学教程（第二版）[M]. 北京： 高等教育出版社， 2009. 6.