

深度学习 | 第1讲：深度学习简介和感知机原理与实现

原创：louwill 狗熊会 2018-08-06



新朋友点蓝色字免费订阅



大家好！从本期开始，狗熊会公众号将陆续推出深度学习系列的推文，旨在帮助大家了解和学习当下火热的深度学习技术的基本原理、实践项目、一些优秀的开源深度学习框架以及工业生产环境下的深度学习项目是如何打通数据和价值之间关系的。

而对于阅读深度学习系列文章的广大数据爱好者而言，小编希望大家能有一些机器学习基础，而且小编不会去刻意用很多通俗的语言去描述数学和计算机科学相关的术语和概念，当然小编也会尽力把主要的知识点说的够敞亮，希望大家理解。那么闲话少说，我们正式开启深度学习的学习之旅~



—— 1 —— 机器学习与深度学习

要是说到深度学习，恐怕不得不先提一下机器学习，解释好二者之间的关系。相信大家心中应该有自己对于机器学习概念的理解。小编这里就一句话简单概括一下：机器学习就是从历史数据中探索和训练出数据的普遍规律，将其归纳为相应的数学模型，并对未知的数据进行预测的过程。至于在这个过程中我们碰到各种各样的问题，比如数据质量、模型评价标准、训练优化方法、过拟合等一系列关乎机器学习模型生死的问题，小编就不展开来说了，自己去补机器学习知识哈。

在机器学习中，我们有很多很多已经相当成熟了的模型和算法。（这里厘一下模型和算法的概念，小编认为，通常我们所说的像SVM之类的所谓机器学习十大算法其实不应该称之为算法，更应该称其为模型，机器学习的算法应该是在给定模型和训练策略的情况下采取的优化算法，比如梯度下降、牛顿法之类。当然，一般情况下将模型和算法混合称呼也不碍事，毕竟模型中本身就包含着计算规则的意思。）在这很多种机器学习模型中，有一种很厉害的模型，那就是人工神经网络。这种模型从早期的感知机发展而来，对任何函数都有较好的拟合性，但自上个世纪90年代一直到2012年深度学习集中爆发前夕，神经网络受制于计算资源的限制和较差的可解释性，一直处于发展的低谷阶段。之后大数据兴起，计算资源也迅速跟上，加之2012年ImageNet竞赛冠军采用的AlexNet卷积神经网络一举将图片预测的 top5 错误率降至16.4%，震惊了当时的学界和业界。从此之后，原本处于研究边缘状态的神经网络又迅速热了起来，深度学习也逐渐占据了计算机视觉的主导地位。

扯了这么多，无非就是想让大家知道，以神经网络为核心的深度学习理论是机器学习的一个领域分支，所以深度学习其本质上也必须是遵循一些机器学习的基本要义和法则的。传统的机器学习中，我们需要训练的是结构化的数值数据，比如说预测销售量、预测某人是否按时还款等等。但在深度学习中，我们的训练输入就不大是常规的数据了，它可能是一张图像、一段语言、一段对话语料或是一段视频。深度学习要做的就是我丢一张猫的图片到神经网络里，它的输出是猫或者cat这样的标签，丢进去一段语音，它输出的是你好这样的文本。所以机器学习/深度学习的核心任务就是找（训练）一个模型，它能够将我们的输入转化为正确的输出。

Machine Learning

≈ Looking for a Function

- Speech Recognition

$$f(\text{audio waveform}) = \text{"How are you"}$$

- Image Recognition

$$f(\text{cat image}) = \text{"Cat"}$$

- Playing Go

$$f(\text{Go board state}) = \text{"5-5" (next move)}$$

- Dialogue System

$$f(\text{"Hi" (what the user said)}) = \text{"Hello" (system response)}$$

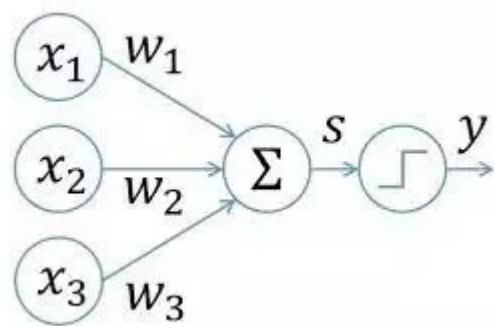
(图片来自台湾大学李宏毅教授的deep learning tutorial ppt)

2

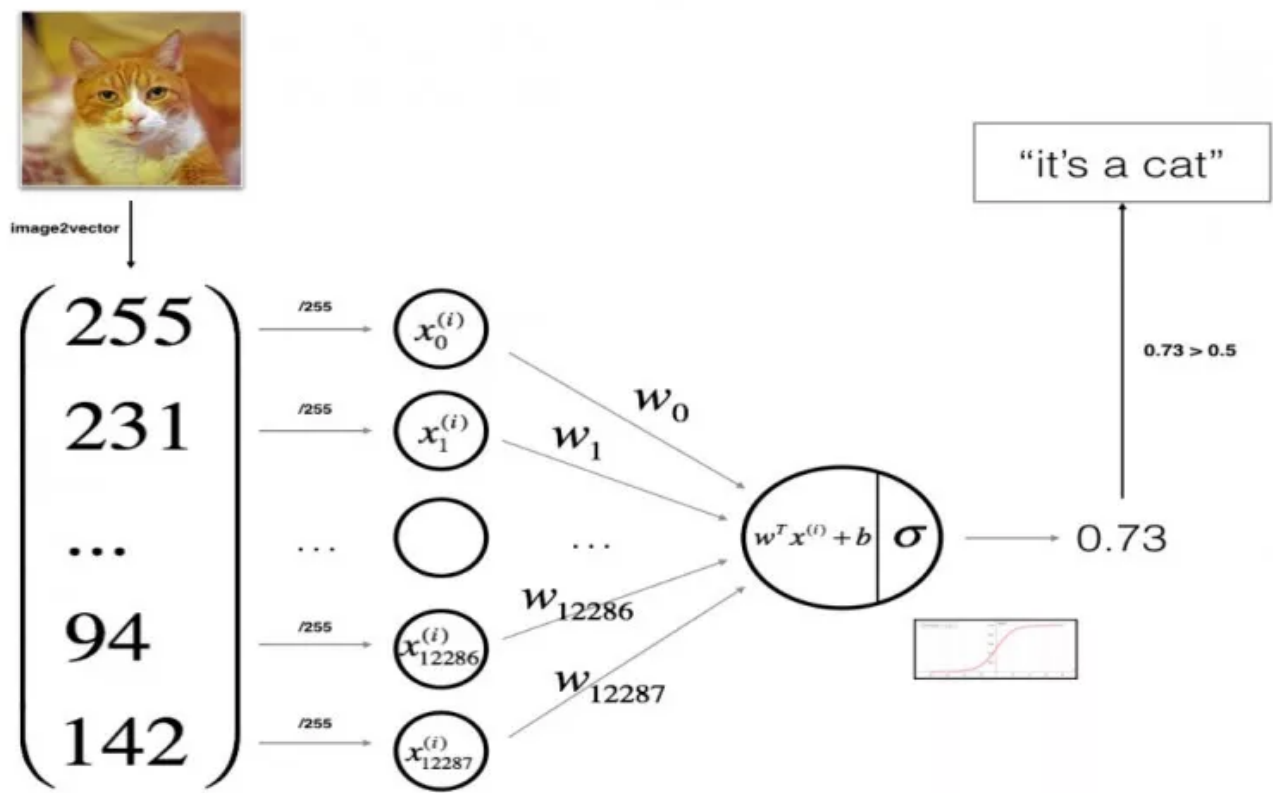
感知机与神经网络

就像上面那幅图展示的一样，深度学习看起来就像是一个黑箱子，给定输入之后就出来预测结果，中间的细节很难搞清楚。在实际生产环境下，调用像 tensorflow 这样优秀的深度学习计算框架能够帮助我们快速搭建起一个深度学习项目，但在学习深度学习的过程中，小编并不建议大家一开始就上手各种深度学习框架，希望大家能和小编一道，在把基本的原理搞明白之后利用 python 或者 R 自己手动去编写模型和实现算法细节。

所以，为了学习各种结构的神经网络，我们需要从头开始。感知机作为神经网络和支持向量机的理论基础，相信任何有机器学习基础的同学都清楚其模型细节。简单而言，感知机就是一个旨在建立一个线性超平面对线性可分的数据集进行分类的线性模型。其基本结构如下所示：



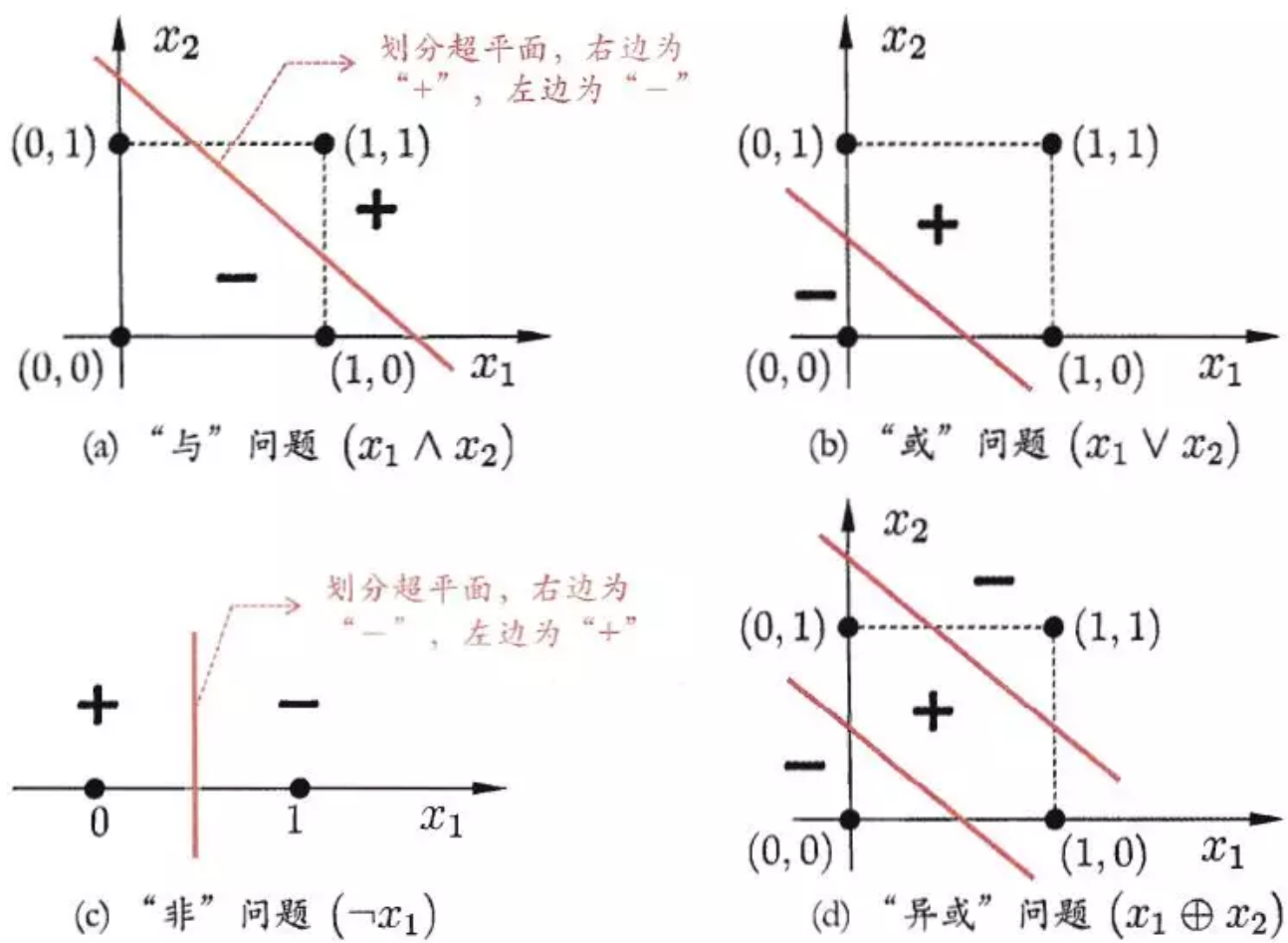
上图从左到右为感知机模型的计算执行方向，模型接受了 x_1 、 x_2 、 x_3 三个输入，将输入与权值参数 W 进行加权求和并经过 sigmoid 函数进行激活，将激活结果作为 y 进行输出。这便是感知机执行前向计算的基本过程。这样就行了吗？当然不行。按照李航老师的统计学习三要素来打分，刚刚我们只解释了模型，对策略和算法并未解释。当我们执行完前向计算得到输出之后，模型需要根据你的输出和实际的输出按照损失函数计算当前损失，计算损失函数关于权值和偏置的梯度，然后根据梯度下降法更新权值和偏置。经过不断的迭代调整权值和偏置使得损失最小，这便是完整的单层感知机的训练过程。



输入为图像的感知机计算过程（图片来自吴恩达老师deeplearningai作业截图）

上述的单层感知机包含两层神经元，即输入与输出神经元，可以非常容易的实现逻辑与、或和非等线性可分情形，但终归而言，这样的一层感知机的学习能力是非常有限的，对于像异或这样的非线性情形，单层感知机就搞不定了。其学习过程会呈现一定程度的振荡，权值参数 w 难以稳定下来，最终不能求得合适的

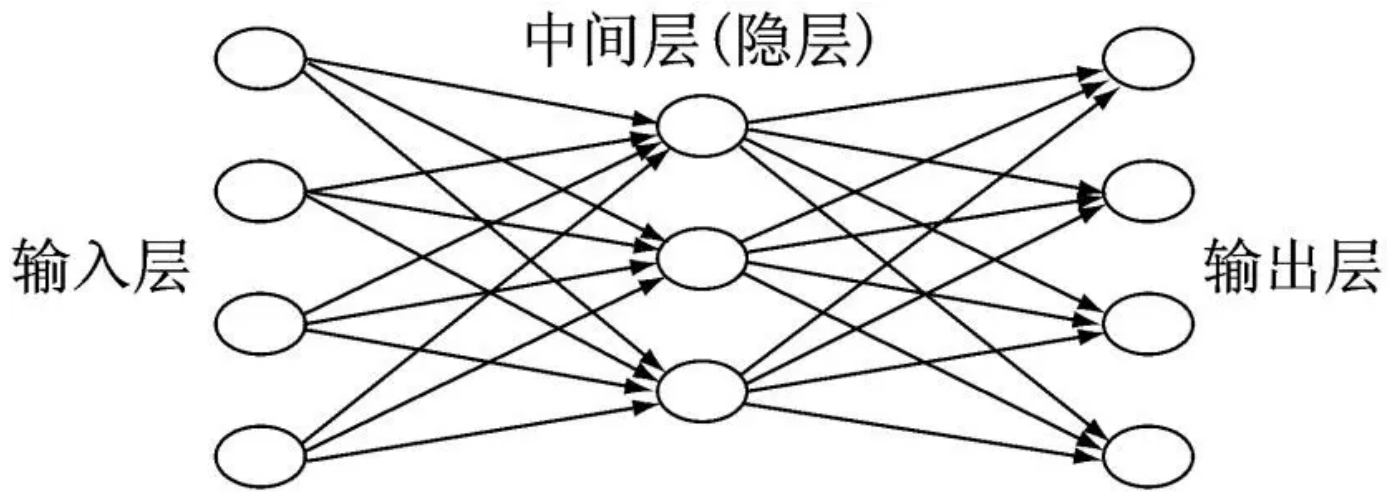
解。



单层感知机难以解决异或问题（截图于周志华老师的《机器学习》）

对于非线性可分的情况，在感知机基础上一般有了两个解决方向，一个就是著名的支持向量机模型，旨在通过核函数映射来处理非线性的情况，这里我们不多谈，读者朋友们可以去回顾复习机器学习中的有关的内容，而另一种就是神经网络模型。这里的神经网络模型也叫多层感知机（MLP: Muti-Layer Perception），与单层的感知机在结构上的区别主要在于 MLP 多了若干隐藏层，这使得神经网络对非线性的情况拟合能力大大增强。

一个单隐层的人工神经网络的结构如下图所示：



可以看到相较于两层神经元的单层感知机，多层感知机中间多了一个隐藏层，称为隐藏层的含义在于神经网络的训练过程中我们只能观察到输入和输出层的数据，对于中间的隐藏层我们是看不见的，因而在深度神经网络（DNN）中，对于中间看不见又难以进行解释的隐藏层又有个黑箱子的称呼。

含隐藏层的神经网络是如何训练的呢？跟感知机一样，神经网络的训练依然是包含前向计算和反向传播两个主要过程。当然，单层感知机没有反向传播的概念，通常是直接建立损失函数对权值和偏置参数的梯度优化。前向计算过程这里不再细述，就是权值偏置与输入的线性加权和激活操作，在隐藏层上有个嵌套的过程。这里我们重点讲一下反向传播算法（Error BackPropagation，因而也叫误差逆传播），作为神经网络的训练算法，反向传播算法可谓是目前最成功的神经网络学习算法了。我们通常说的 BP 神经网络也就是指应用反向传播算法进行训练的神经网络模型。

那反向传播算法究竟是怎样个工作机制呢？前方高能，需要大家自己补习微积分知识。因为小编实在是没有不借助公式把反向传播讲清楚的能力。假设以一个两层（即单隐层）网络为例，也就是上图中的网络结构，小编带大家详细推导一下反向传播的基本过程。

我们假设输入层为 X ，输入层与隐藏层之间的权值和偏置分别为 $W1$ 和 $b1$ ，线性加权计算结果为 $Z1 = W1 * X + b1$ ，采用 sigmoid 激活函数，隐藏层是激活输出为 $a1 = \sigma(Z1)$ 。而隐藏层到输出层的权值和偏置分别为 $W2$ 和 $b2$ ，线性加权计算结果为 $Z2 = W2 * a1 + b2$ ，激活输出为 $a2 = \sigma(Z2)$ 。所以这个两层网络的前向计算过程为 $X \rightarrow Z1 \rightarrow a1 \rightarrow Z2 \rightarrow a2$ 。

所以反向传播的直观理解就是将上述前向计算过程反过来，但必须是梯度计算的方向反过来，假设我们这里采用交叉熵损失函数：

$$L(y, a) = -y \log a + (1 - y) \log(1 - a)$$

反向传播是基于梯度下降策略的，主要是以目标参数的负梯度方向对参数进行更新，所以基于损失函数对前向计算过程中各个变量进行梯度计算就是非常必要的了。将前向计算过程反过来，那基于损失函数的梯度计算顺序就是 $da_2-dZ_2-dW_2-db_2-da_1-dZ_1-dW_1-db_1$ 。一大堆微分符号！聪明如你应该可以看到我们马上要进行一波链式求导操作。我们从输出 a_2 开始进行反向推导。输出层激活输出为 a_2 ，那首先计算损失函数 $L(y, a)$ 关于 a_2 的微分 da_2 ，影响输出 a_2 的是谁呢？由前向传播可知 a_2 是由 Z_2 经激活函数激活计算而来的，所以计算损失函数关于 Z_2 的导数 dZ_2 必须经由 a_2 进行复合函数求导，即微积分上常说的链式求导法则。然后继续往前推，影响 Z_2 的又是哪些变量呢？由前向计算 $Z_2 = W_2 * a_1 + b_2$ 可知影响 Z_2 的有 W_2 、 a_1 和 b_2 ，继续按照链式求导法则进行求导即可。最终以交叉熵损失函数为代表的两层神经网络的反向传播向量化求导计算公式如下所示：

$$\frac{\partial L}{\partial a_2} = \frac{d}{da_2} L(a_2, y) = (-y \log a_2 - (1 - y) \log(1 - a_2))' = -\frac{y}{a_2} + \frac{1-y}{1-a_2}$$

$$\frac{\partial L}{\partial Z_2} = \frac{\partial L}{\partial a_2} \frac{\partial a_2}{\partial Z_2} = a_2 - y$$

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial a_2} \frac{\partial a_2}{\partial Z_2} \frac{\partial Z_2}{\partial W_2} = \frac{1}{m} \frac{dL}{dZ_2} a_1 = \frac{1}{m} (a_2 - y) a_1$$

$$\frac{\partial L}{\partial b_2} = \frac{\partial L}{\partial a_2} \frac{\partial a_2}{\partial Z_2} \frac{\partial Z_2}{\partial b_2} = \frac{dL}{dZ_2} = a_2 - y$$

$$\frac{\partial L}{\partial a_1} = \frac{\partial L}{\partial a_2} \frac{\partial a_2}{\partial Z_2} \frac{\partial Z_2}{\partial a_1} = (a_2 - y) w_2$$

$$\frac{\partial L}{\partial Z_1} = \frac{\partial L}{\partial a_2} \frac{\partial a_2}{\partial Z_2} \frac{\partial Z_2}{\partial a_1} \frac{\partial a_1}{\partial Z_1} = (a_2 - y) w_2 \sigma'(Z_1)$$

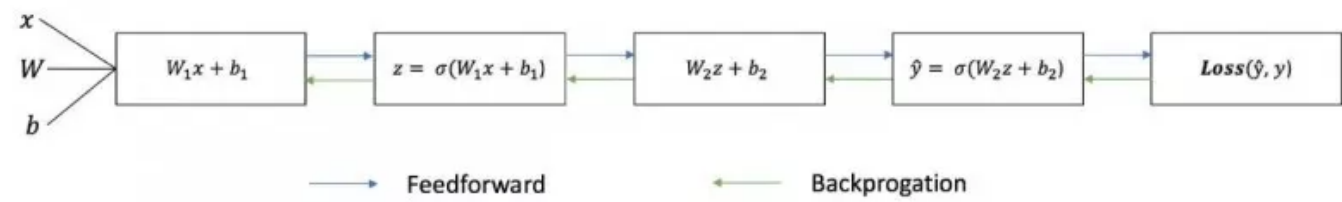
$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial a_2} \frac{\partial a_2}{\partial Z_2} \frac{\partial Z_2}{\partial a_1} \frac{\partial a_1}{\partial Z_1} \frac{\partial Z_1}{\partial W_1} = (a_2 - y) w_2 \sigma'(Z_1) x$$

$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial a_2} \frac{\partial a_2}{\partial Z_2} \frac{\partial Z_2}{\partial a_1} \frac{\partial a_1}{\partial Z_1} \frac{\partial Z_1}{\partial b_1} = (a_2 - y) w_2 \sigma'(Z_1)$$

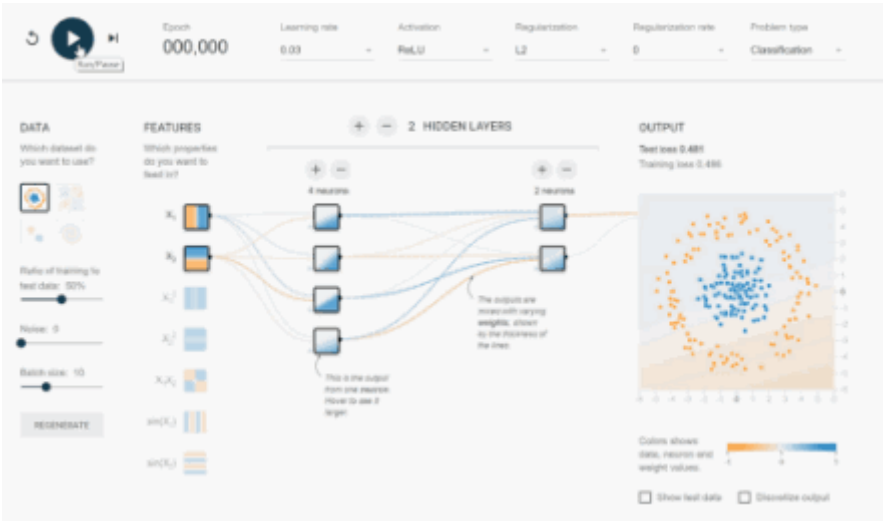
在有了梯度计算结果之后，我们便可根据权值更新公式对权值和偏置参数进行更新了，具体计算公式如下，其中 η 为学习率，是个超参数，需要我们在训练时手动指定，当然也可以对其进行调参取得最优超参数。

$$w = w - \eta dw$$

以上便是 BP 神经网络模型和算法的基本工作流程，简单而言就是前向计算得到输出，反向传播调整参数，最后以得到损失最小时的参数为最优学习参数。神经网络的基本总结流程如下图所示：



训练一个 BP 神经网络并非难事，我们有足够优秀的深度学习计算框架通过几行代码就可以搭建起一个全连接网络。但是为了学习和掌握神经网络的基本思维范式和锻炼实际的编码能力，希望大家能够利用 python 或者 R 在不调用任何算法包的情况下根据算法原理手动实现一遍神经网络模型。最后以一个神经网络可视化的动图给大家动态的展示一下神经网络的训练过程：



动图依据 <https://playground.tensorflow.org/>制作而成



第一讲的内容到这里就结束了，在深度学习第一讲中，我们了解了深度学习和机器学习的基本关系和发展历程，对神经网络的理论基础有了更深层次的学习和掌握。咱们下期见！



【参考资料】

李航 统计学习方法
周志华 机器学习
<https://www.deeplearning.ai/>

作者简介

鲁伟，狗熊会人才计划一期学员。目前在杭州某软件公司从事数据分析和深度学习相关的研究工作，研究方向为贝叶斯统计、计算机视觉和迁移学习。



识别二维码，查看作者更多精彩文章



识别下方二维码成为狗熊会会员！

友情提示：

个人会员**不提供数据、代码**，

视频**only**！

个人会员网址：<http://teach.xiong99.com.cn>



点击“[阅读原文](#)”，成为狗熊会会员！

[阅读原文](#)