

MLP 实验经验分享

个人自我介绍

- 姓名：崔行健
- 专业：信息与计算科学
- Github 仓库：<https://github.com/Triglang/Artificial-Intelligence>

理论知识预备

我们的 AILab 和理论课是配套进行的。对我个人而言，我更愿意把实验当作理论知识的一个补充，更希望实验课是为了我自己的理论课所服务。因此，做实验前可以先仔细先看一遍理论的课件、做一些笔记。

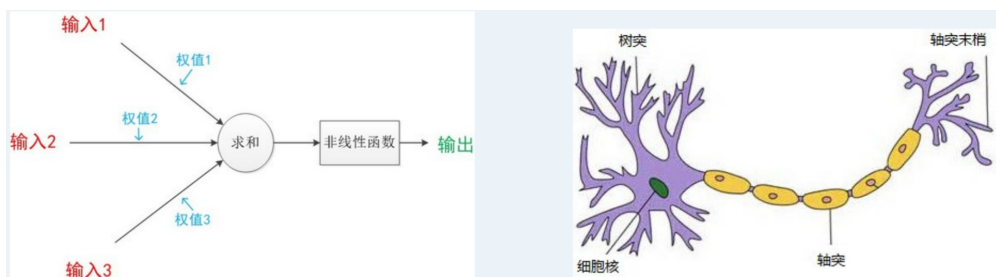
- 在笔记里，虽然大多数都是记录一些课件里面已有的知识，但也偶尔可以对一些重点难点的内容做上自己的一些标注，以及根据知识的内容重构知识结构。这样有利于加深对知识的理解。

人工神经元模型 (MP)

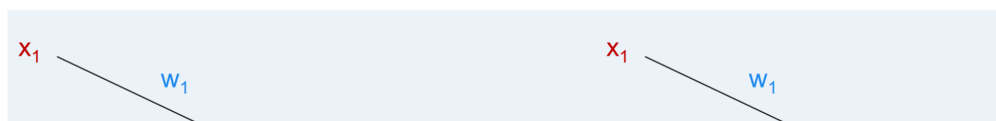
一个典型的神经元模型：包含有 N 个输入，1 个输出，以及 2 个计算功能，每个连接上有一个权值。

神经元的计算公式：用 $\mathbf{x} = [x_1, x_2, x_3]^T$ 表示输入，用 $W = [w_1, w_2, w_3]$ 表示权值，经过加权计算末端信号为 $W \cdot \mathbf{x}$ ，再叠加上非线性函数 g ，即是输出值 z

- 在MP模型里，非线性函数 g 设置是符号函数 (sign)
- 由于起到类似神经元的“激活”作用，非线性函数也称作激活函数 (Active Function)



对神经元模型的图进行简化：将求和函数 sum 与激活函数 sign 合并到一个圆圈里，代表神经元的内部计算。此时，一个“节点”就是一个神经元。



有些时候，前面用到的知识的原理会放在后面才讲，这个时候我们可以把后面的知识的笔记做到前面。这样复习一遍下来，对理论知识的印象就能更加深刻。

实验代码编写

作为 AI 领域的初学者，其实 AILab 的很多代码写法我都从来没写过，pytorch 我也从来没用过，调库我也不知道有什么库可调，我也不知道这个库是什么意思。

```
from torch import nn
import torch
import torch.nn.functional as F
```

对此，我觉得可以抱着学习的心态去写代码，而不是纯粹的想着我有从零开始手搓一段代码。所以，在每次开始写代码之前，我们都可以借助一些工具去学习一些代码：

1. [CSDN](#)
2. [Github](#)
3. [AI](#)

对于 AI 工具，经常用它做实验的的同学（bushi）应该有过一种体会：如果将整段的代码直接 copy 给 AI 让它填写 TODO 部分，或者直接将实验要求发送给它的话，它大概率吐出来的代码是有 bug 的。所以每次使用它的时候，我们要将任务切分成小部分（比如说单独某个函数），一步步去实现。

其实大部分时候，AI 都不能代替我们的工作，就是说它很难从零开始写出一份优雅的代码。它能起到的作用只是，我知道这段代码怎么写，但我懒得写，然后让 AI 写。

实验报告

建议软件：typora + markdown

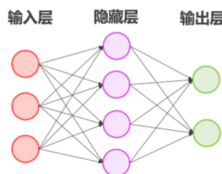
对于实验报告的编写，我认为它的**理论性**应该更高于它的**实践性**。正如我前面说过，实验应该是对理论的一个补充。具体什么意思呢？

第一，我认为对[实验原理的阐述](#)需要足够清晰。

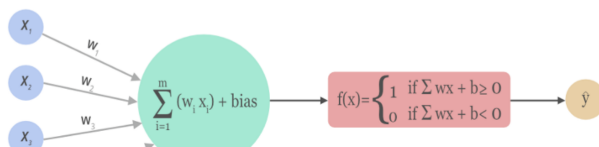
1.1 算法原理

1.1.1 人工神经网络

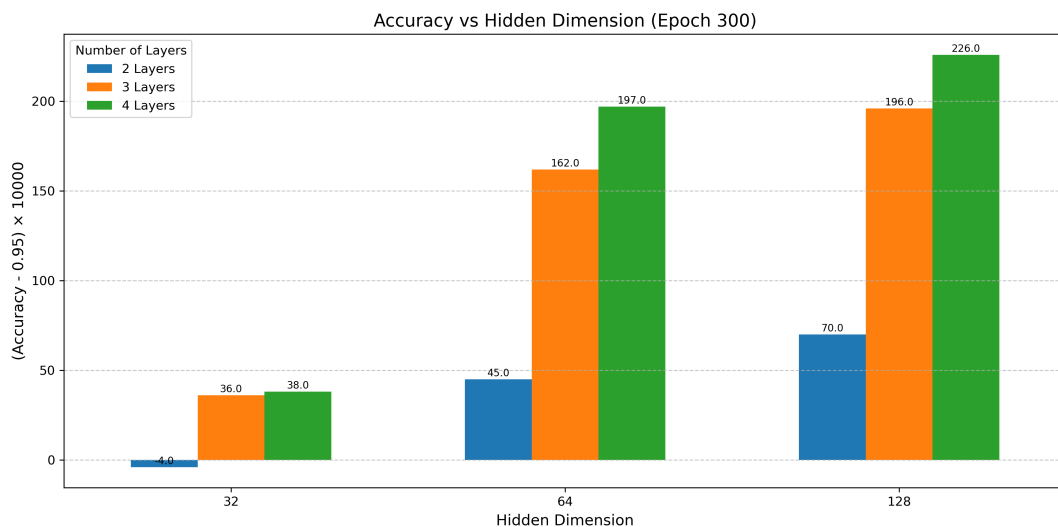
多层感知机包含三个层次：一个输入层，一个或多个中间层（也叫隐藏层，hidden layer）和一个输出层。输入层与输出层的节点数是固定的，中间层则可以自由指定。



可以看见，神经元之间是全连接的。对每个神经元，上一层的数据通过加权线性求和后，经过当前神经元的激活函数，得到当前神经元的输出。每一层的计算依次向前，即可得到输出。如下图所示：



第二，我认为需要对[实验结果的分析 and 思考](#)足够深刻。



第三，代码的实现是其次，不必过多的将源代码 copy 到报告中水字数，将伪代码阐述清晰即可。

