# EXP4 Python神经网络手写数字识别

目录

[一、实验介绍 3](#_Toc5819)

[1.1 实验目的 3](#_Toc27409)

[1.2 实验环境 4](#_Toc13263)

[二、实验原理 4](#_Toc802)

[2.1 全连接层 4](#_Toc19746)

[2.2 非线性激活 5](#_Toc9385)

[2.3 损失函数 7](#_Toc9520)

[2.4前向传播与反向传播 8](#_Toc22325)

[三、 实验步骤 13](#_Toc2958)

[3.1 MNIST 数据集文件的读取和预处理 13](#_Toc8119)

[3.2 全连接层实现 14](#_Toc24144)

[3.3 ReLU非线性激活层实现 15](#_Toc17719)

[3.4 SoftMax损失层实现 15](#_Toc11418)

[四、实验要求 18](#_Toc21919)

[4.1实验评估 18](#_Toc25270)

[4.2实验思考 18](#_Toc29165)

## 一、实验介绍

### 1.1 实验目的

掌握神经网络的设计原理，熟练掌握神经网络的训练和推断方法，能够使用 Python语言实现一个三层全连接神经网络模型对手写数字分类的训练和使用。具体包括：

1) 实现三层神经网络模型进行手写数字分类，建立一个简单而完整的神经网络工程。通过本实验理解神经网络中基本模块的作用和模块间的关系，为后续建立更复杂的神经网络（如风格迁移）奠定基础。

2) 利用高级编程语言 Python 实现神经网络基本单元的前向传播（正向传播）和反向传播计算，加深对神经网络中基本单元的理解，包括全连接层、激活函数、损失函数等基本单元。

3) 利用高级编程语言 Python 实现神经网络训练所使用的梯度下降算法，加深对神经 网络训练过程的理解。

### **1.2 实验环境**

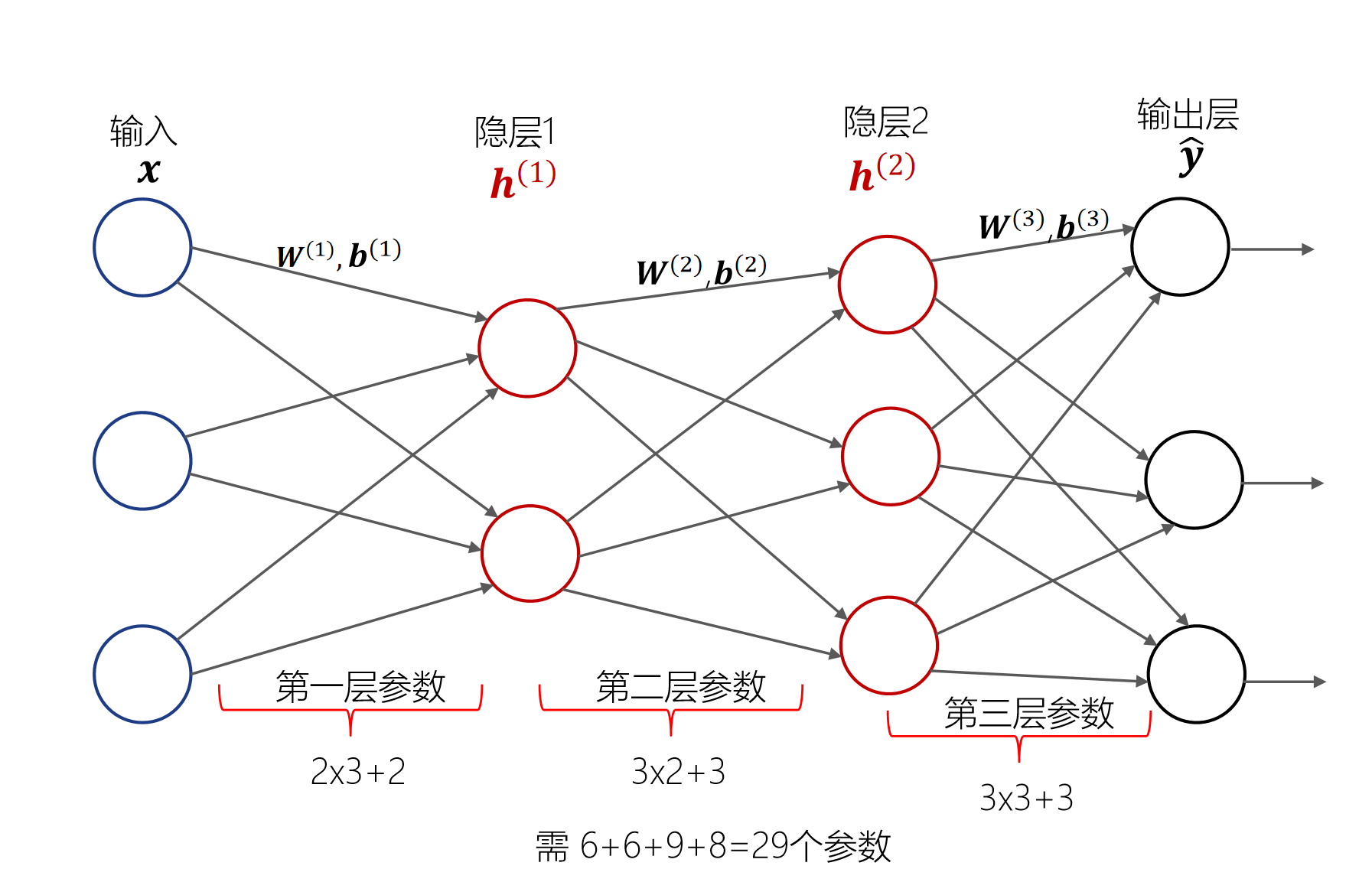
硬件环境：CPU。

软件环境：Python 编译环境及相关的扩展库，包括 Pillow，Scipy，NumPy等（本实验不能使用 Pytorch，TensorFlow 等深度学习框架）。

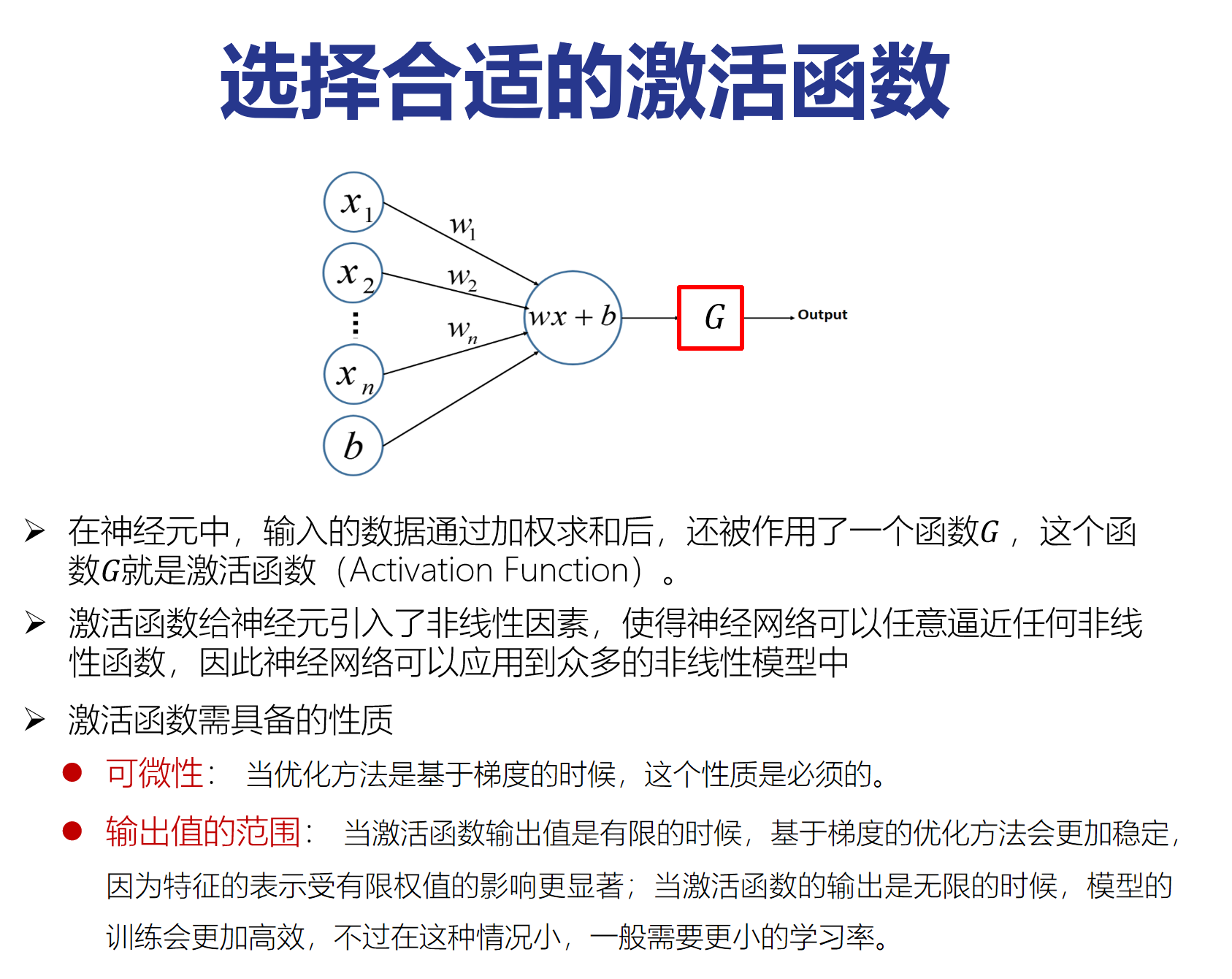
数据集：MNIST 手写数字库[2]。该数据集包含一个训练集和一个测试集，其中训练集有 60000 个样本，测试集有 10000 个样本。每个样本都由灰度图像（即单通道图像）及其标记组成，图像大小为 28 × 28。MNIST 数据集包含 4 个文件，分别是训练集图像、训练集标记、测试集图像、测试集标记。下载地址为http://yann.lecun.com/exdb/mnist/。

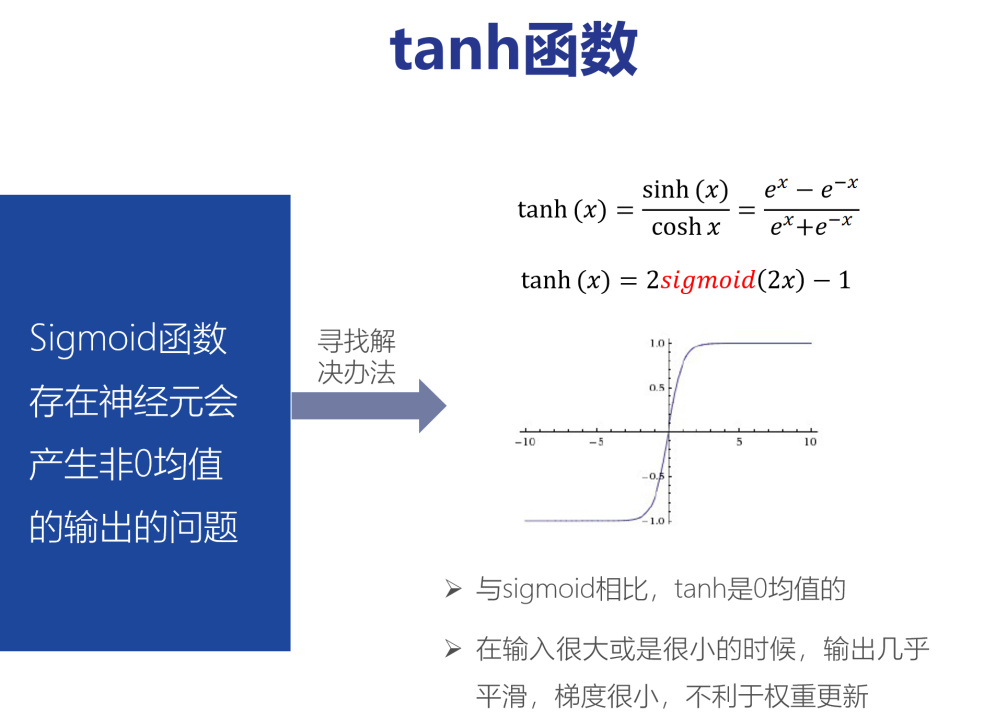
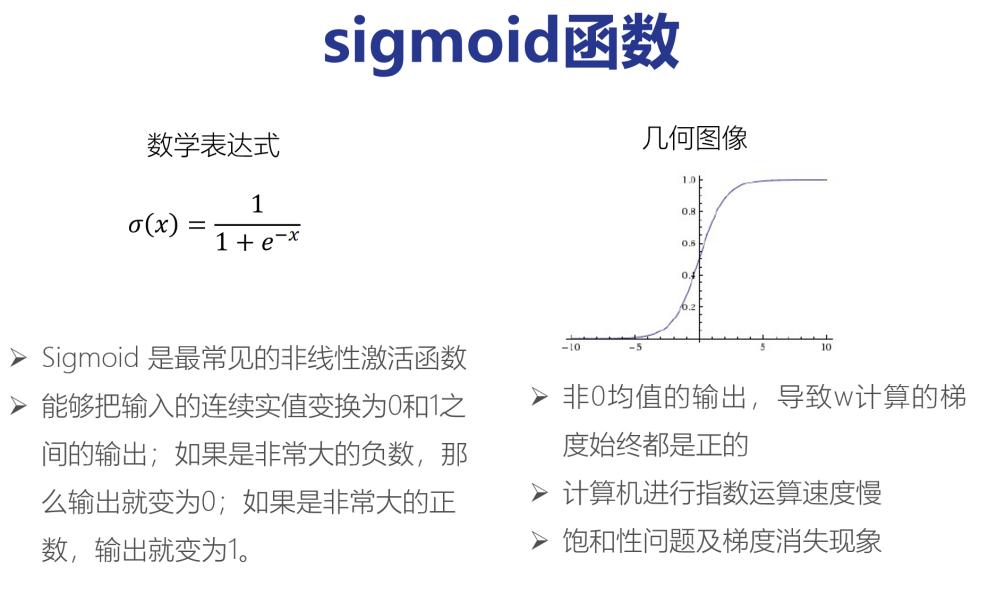
## **二、实验原理**

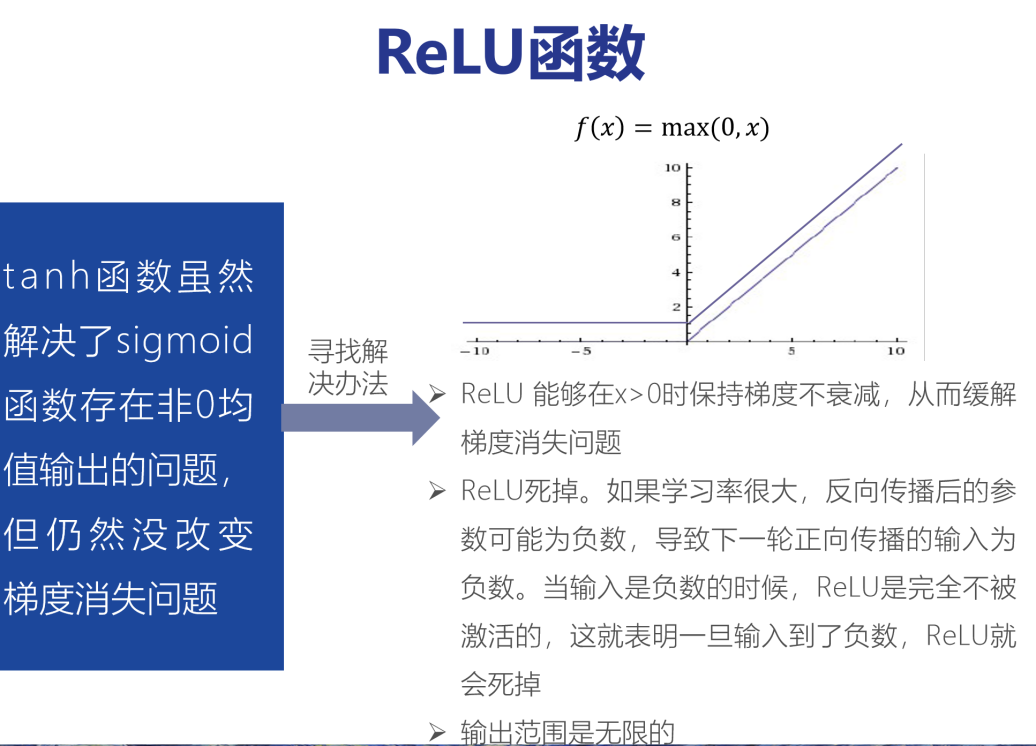
### **2.1 全连接层**



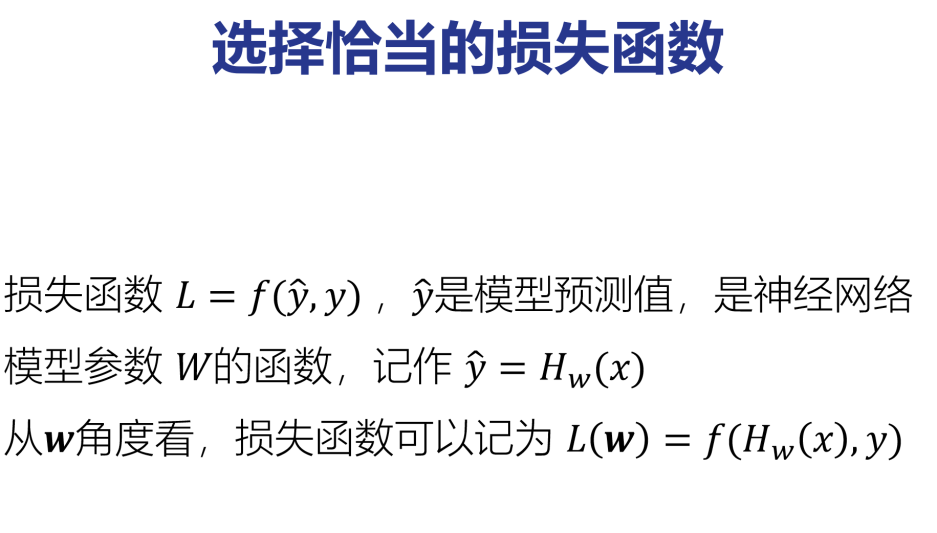
### **2.2 非线性激活**

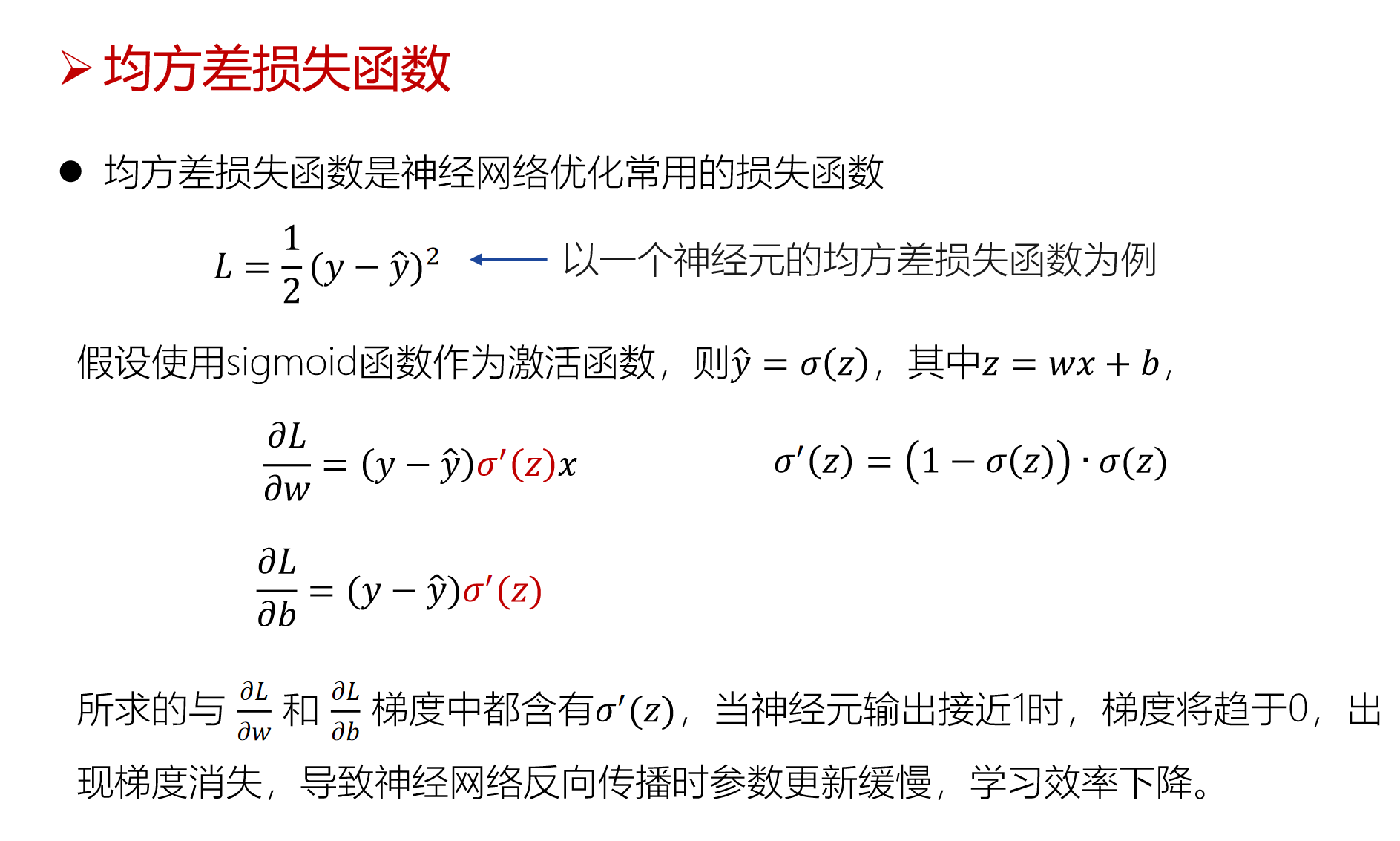




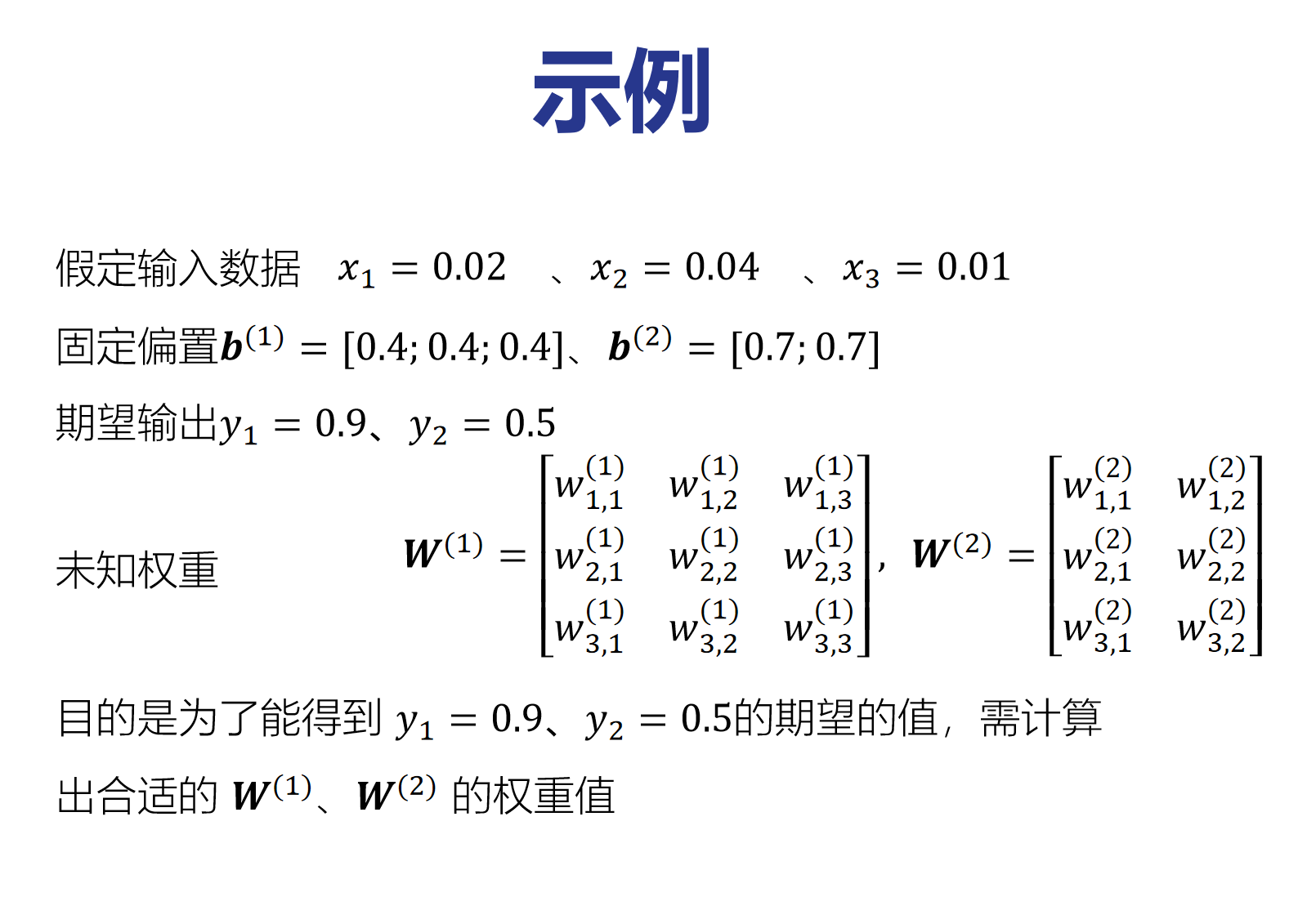


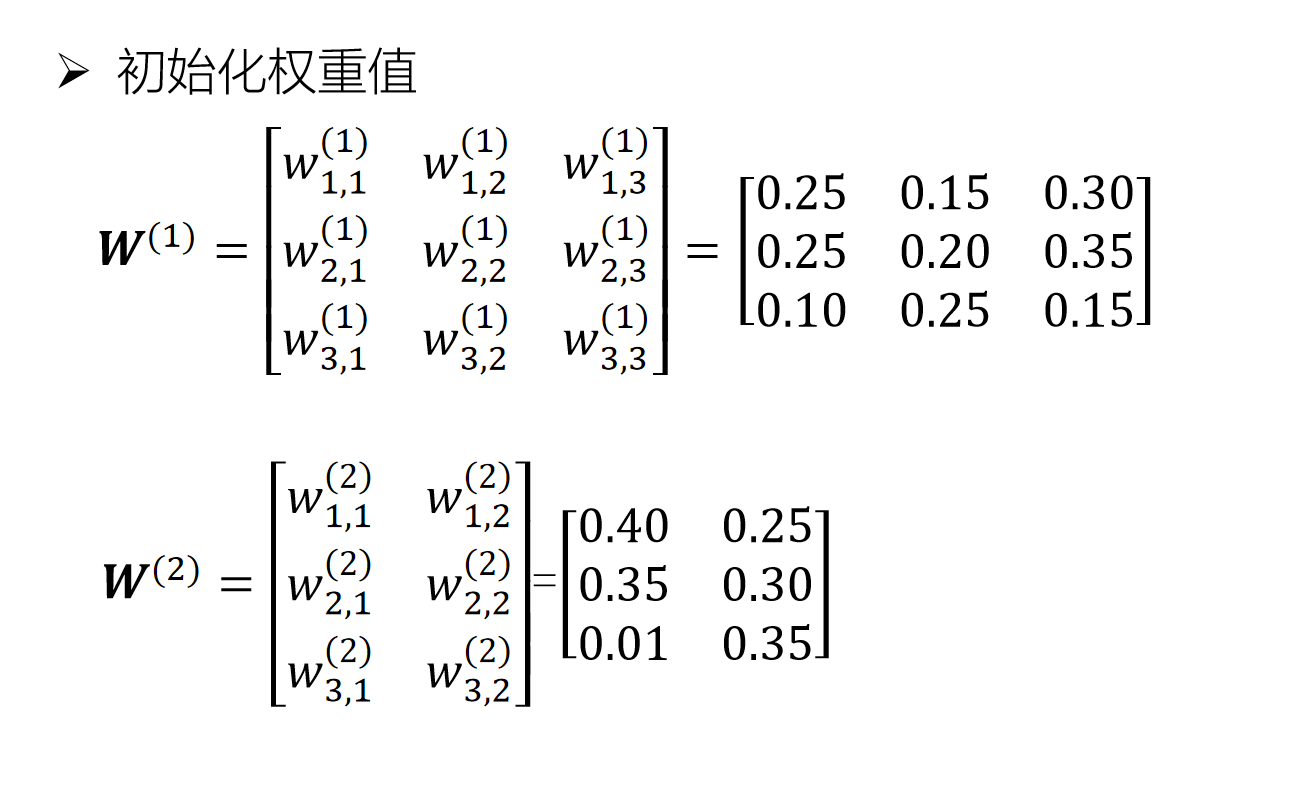
### **2.3 损失函数**

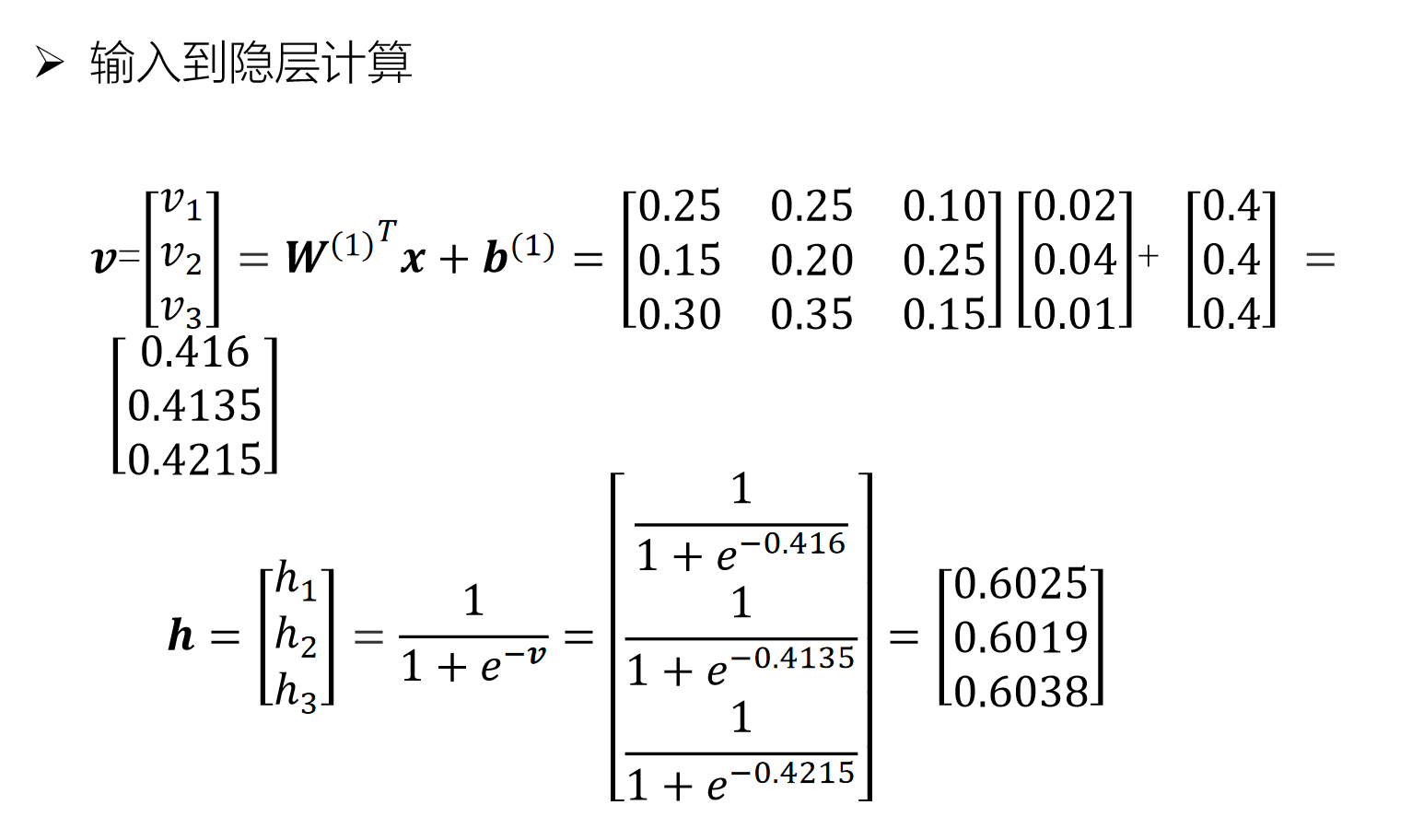


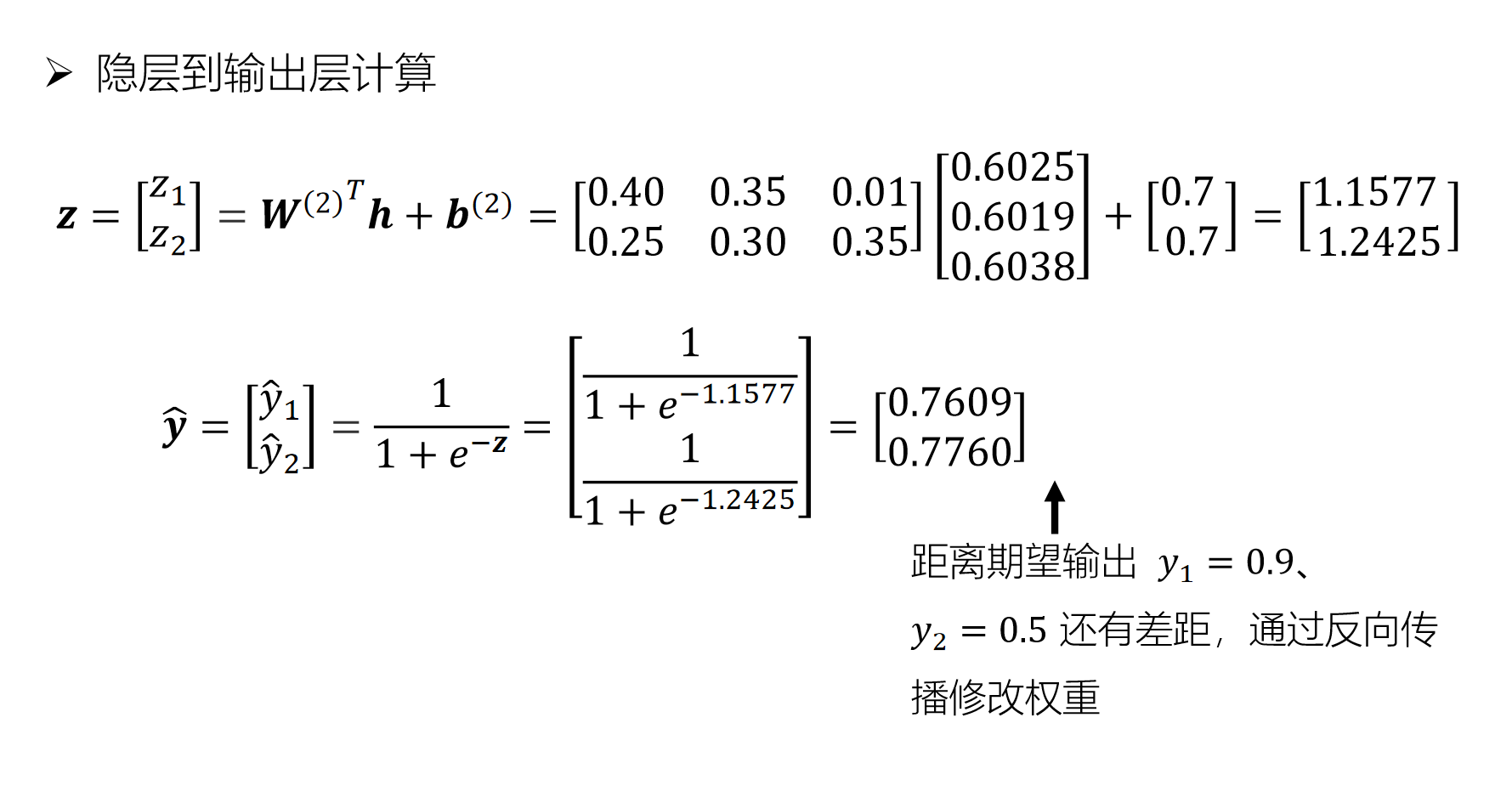


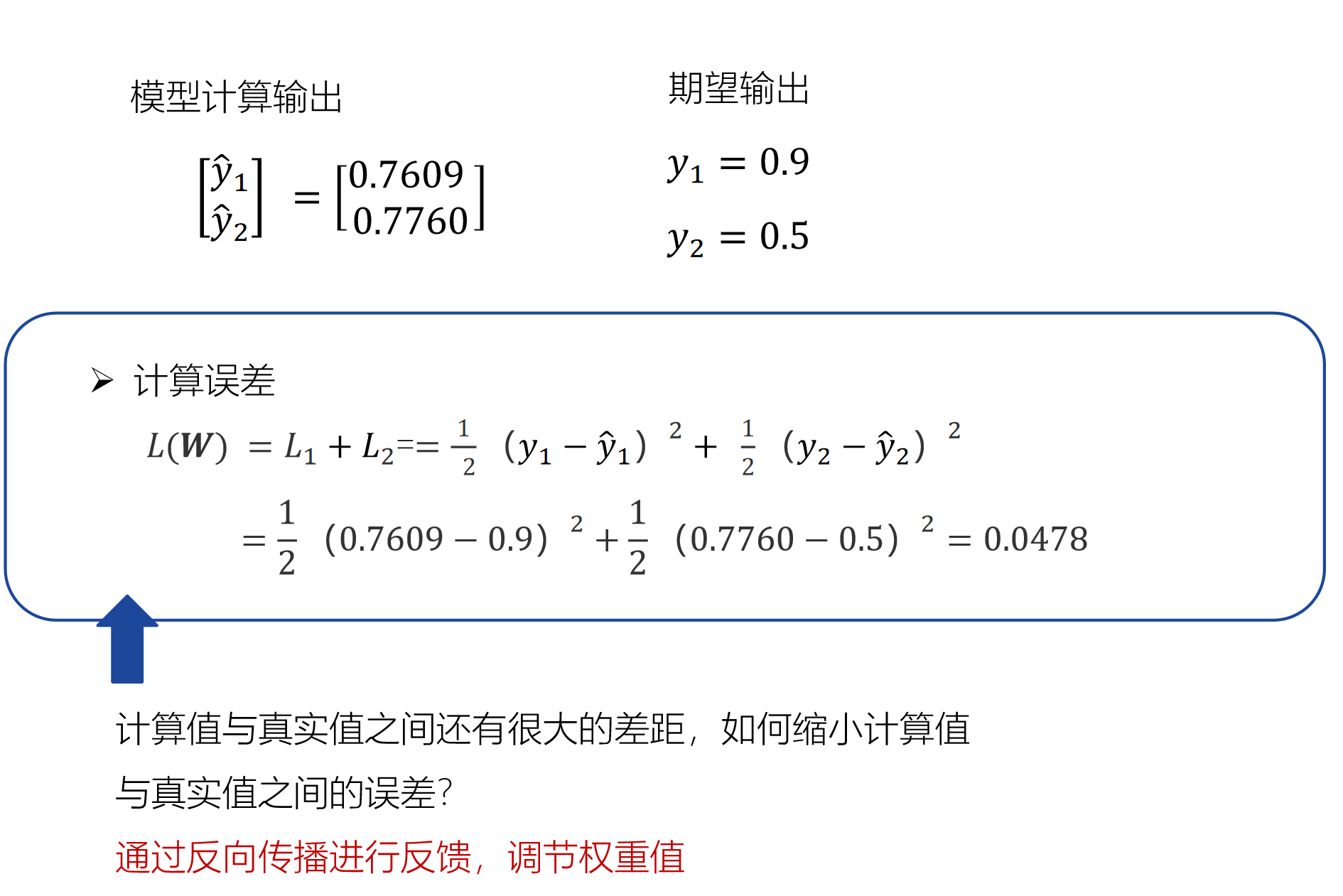
### **2.4前向传播与反向传播**

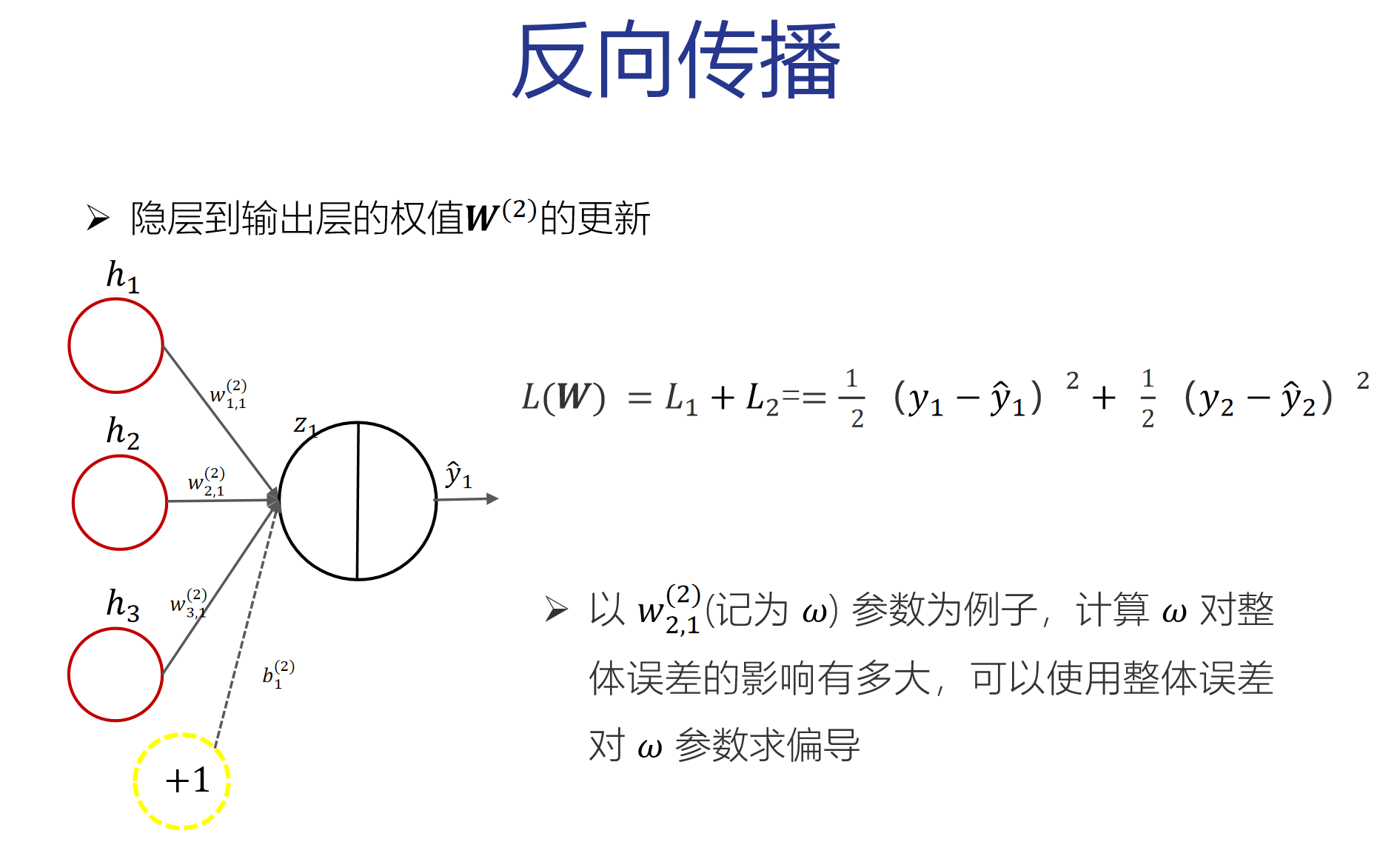


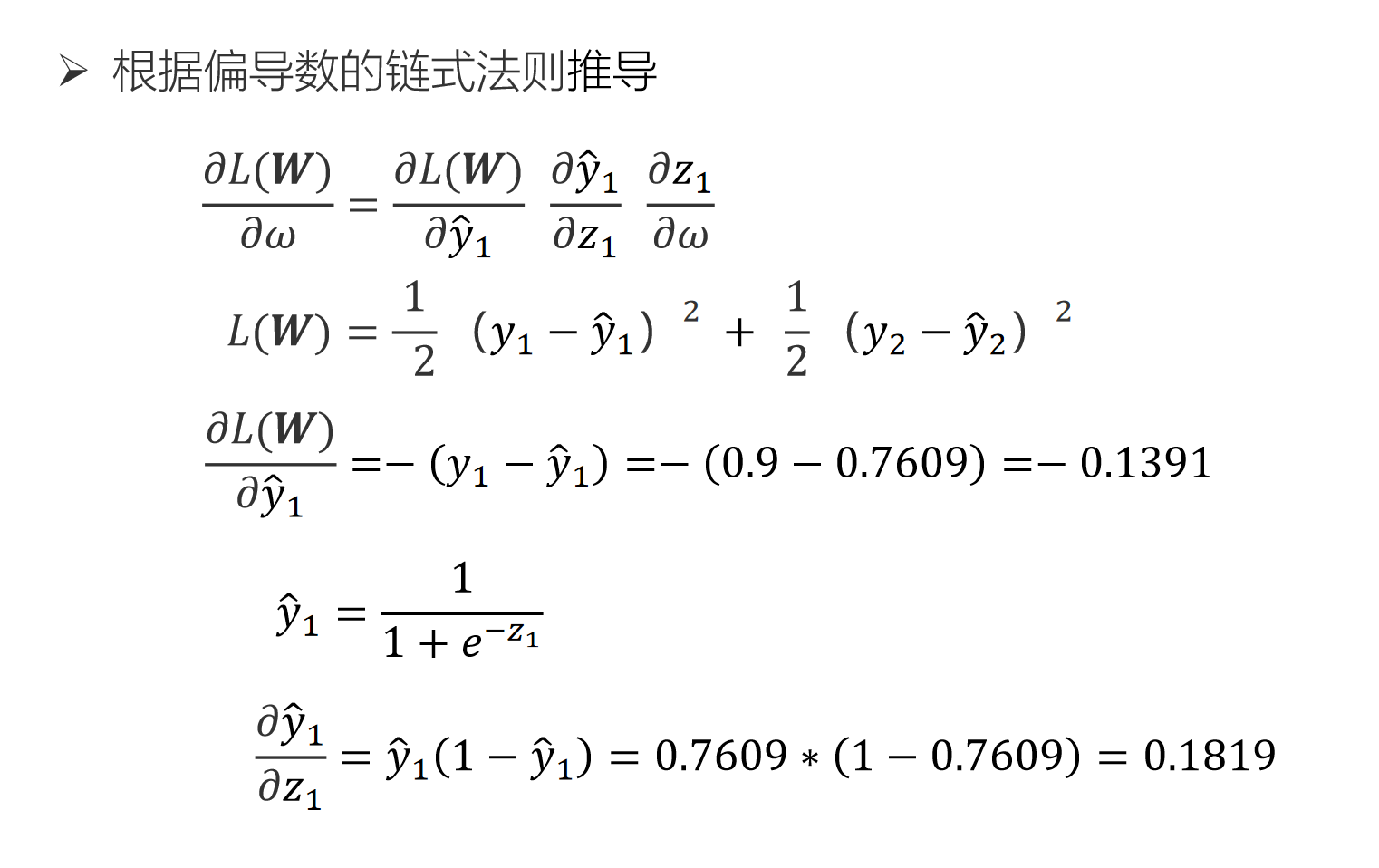


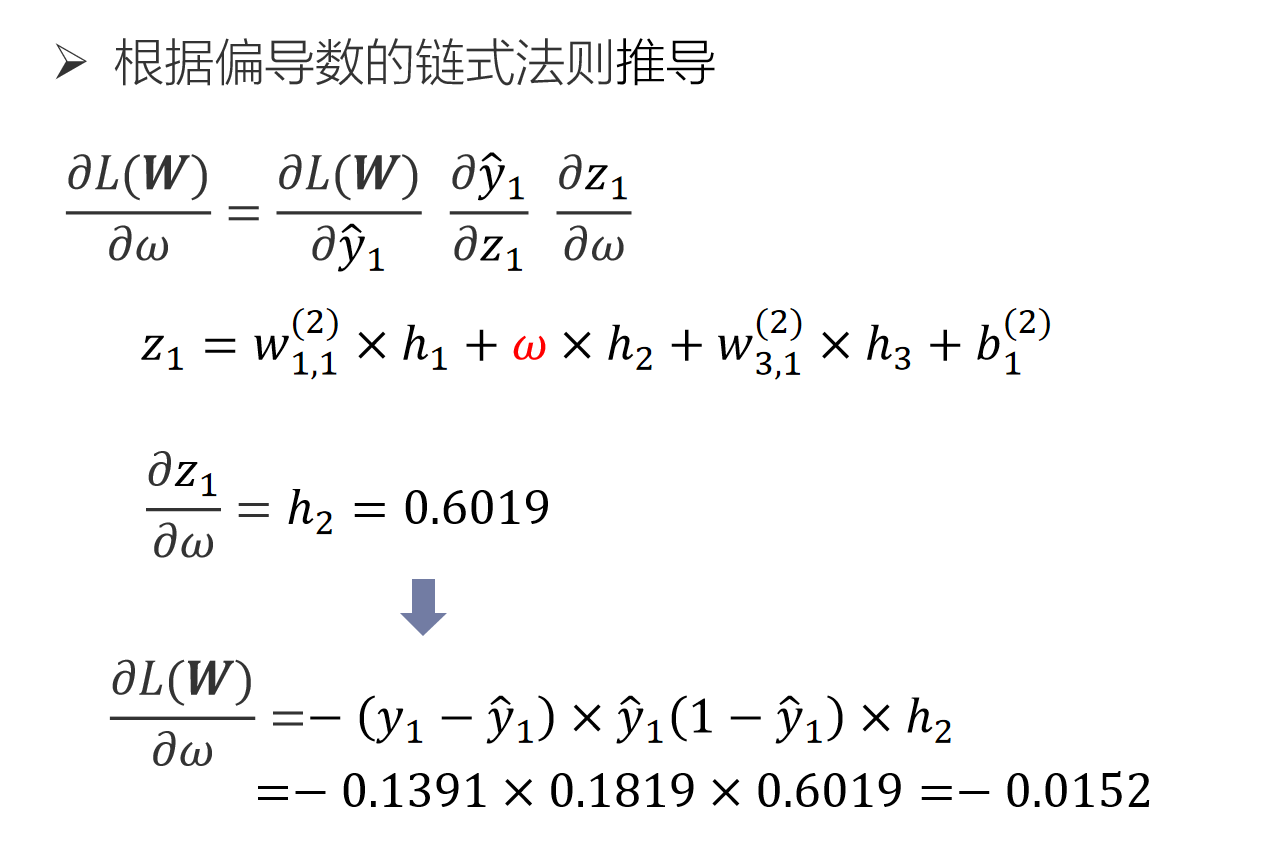


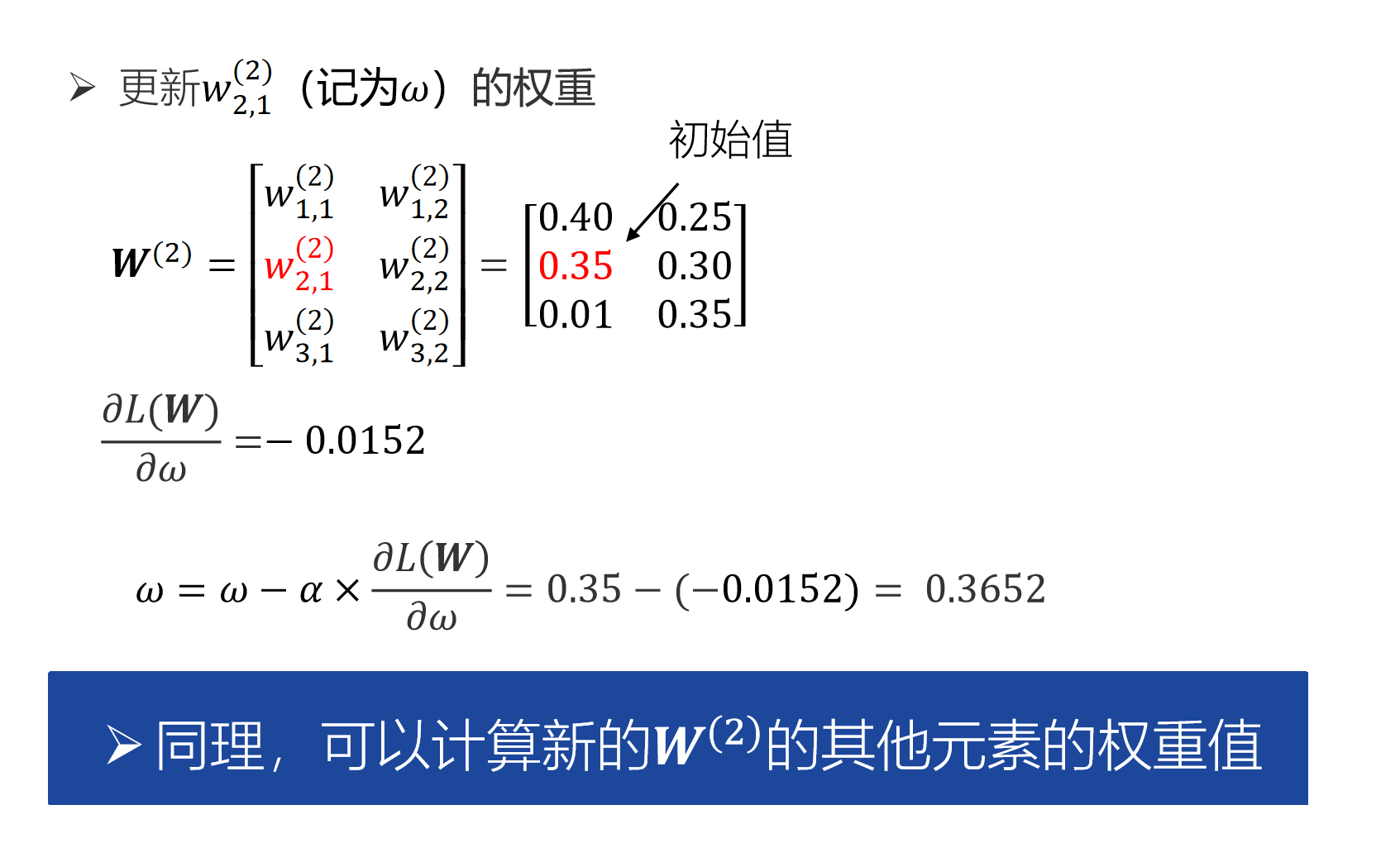






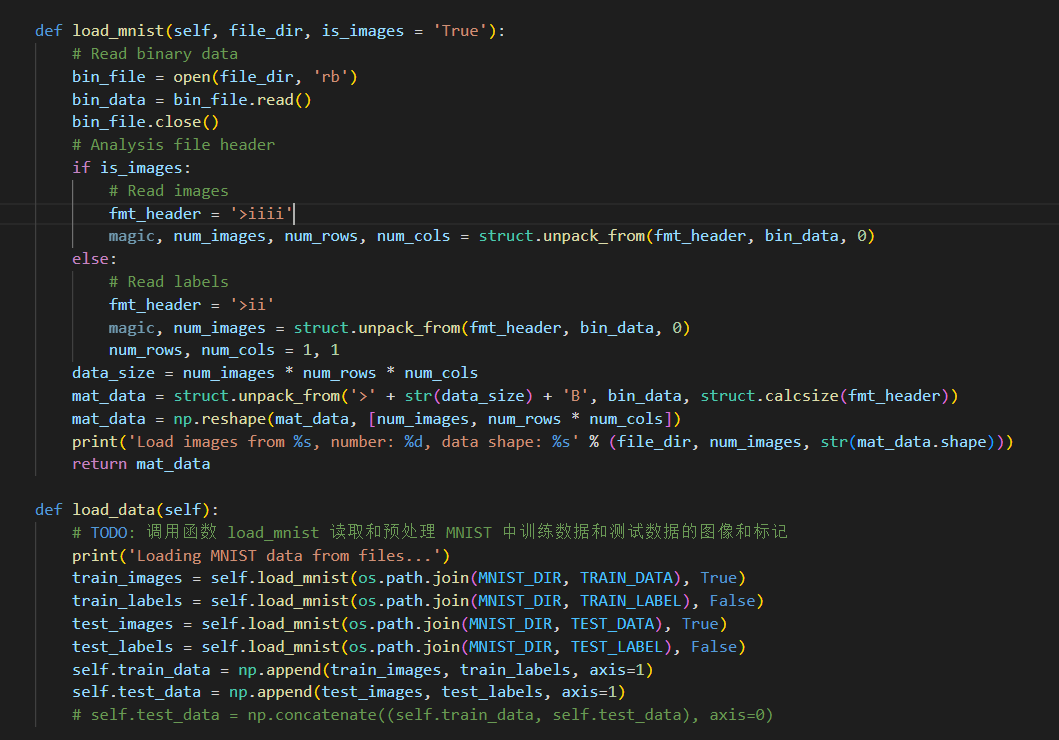




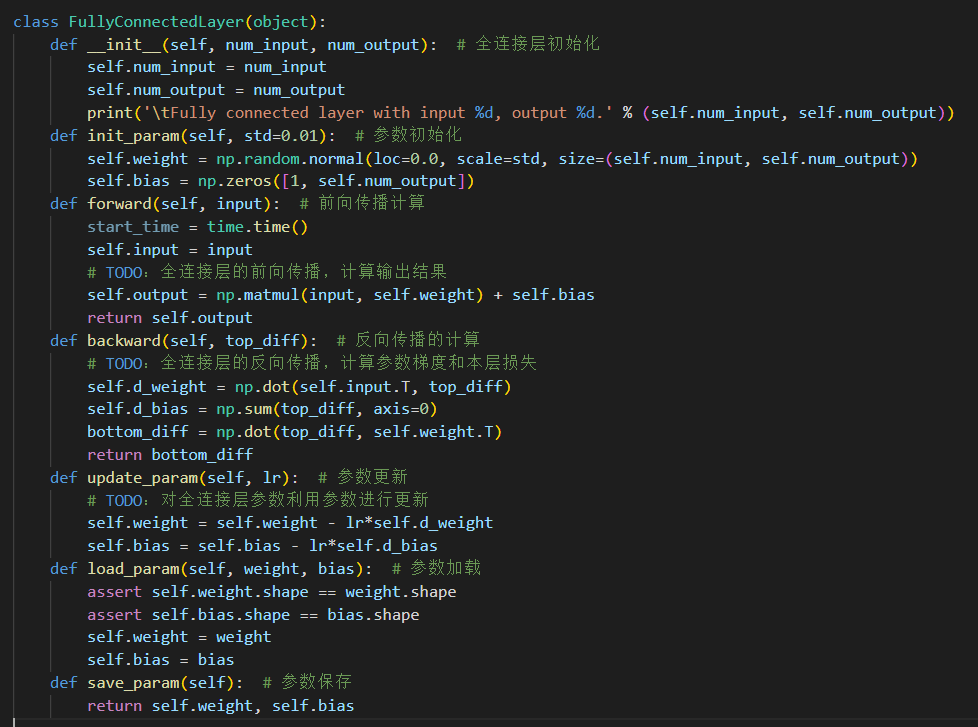


## **实验步骤**

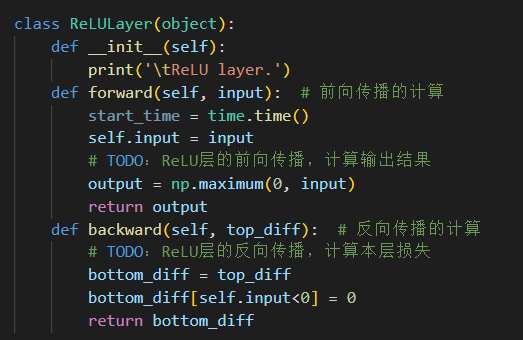
### **3.1 MNIST 数据集文件的读取和预处理**



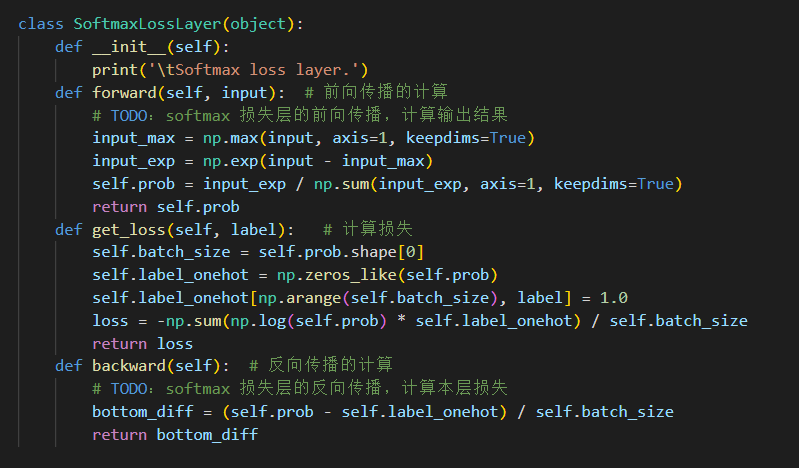
### **3.2 全连接层实现**



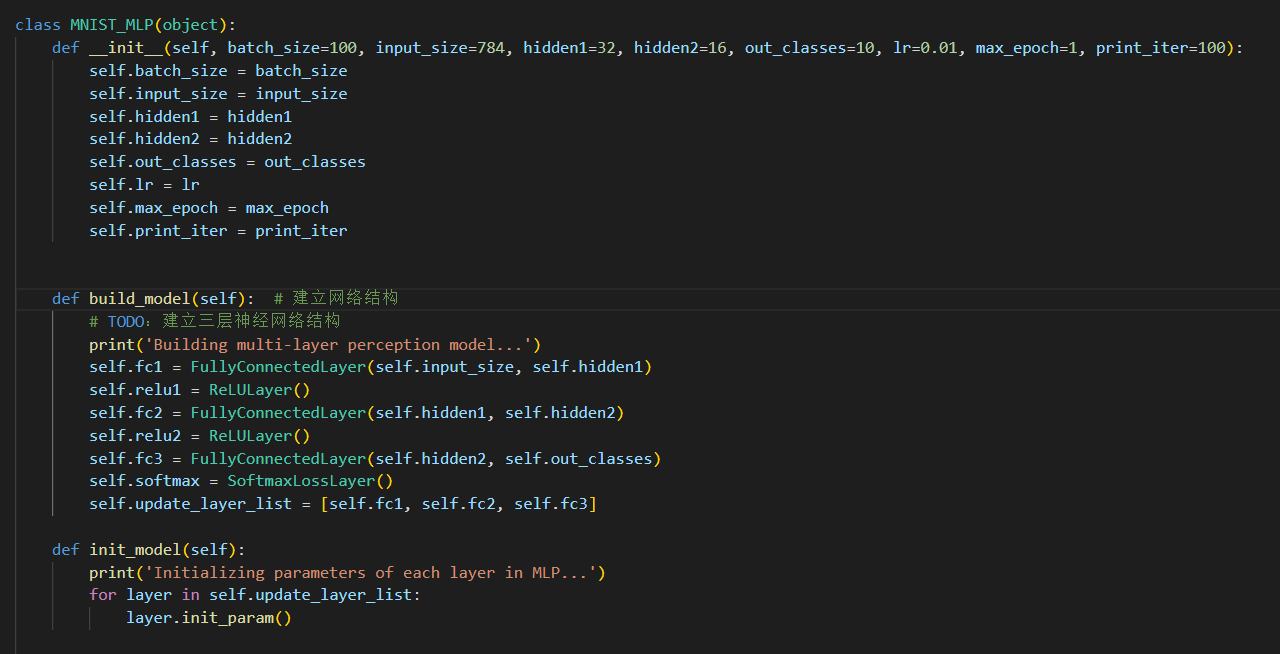
### **3.3 ReLU非线性激活层实现**



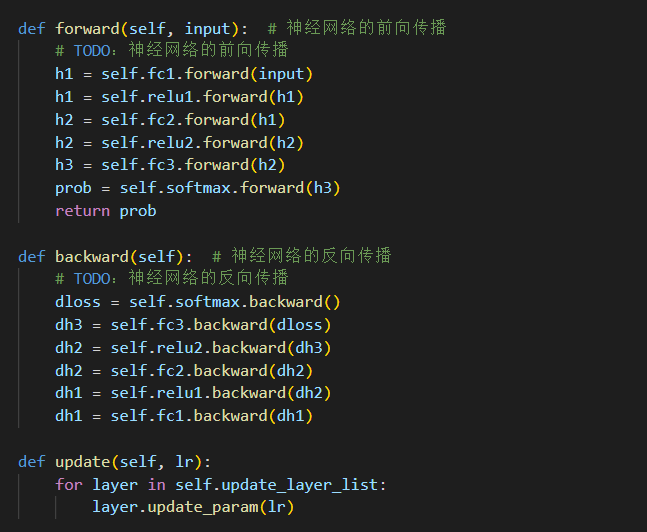
### **3.4 SoftMax损失层实现**

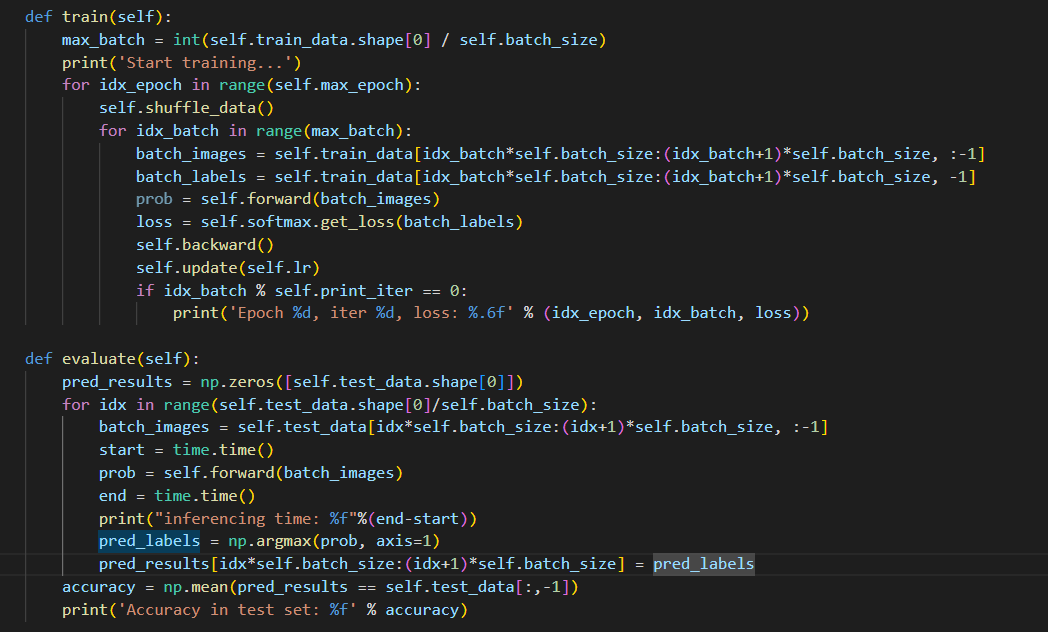


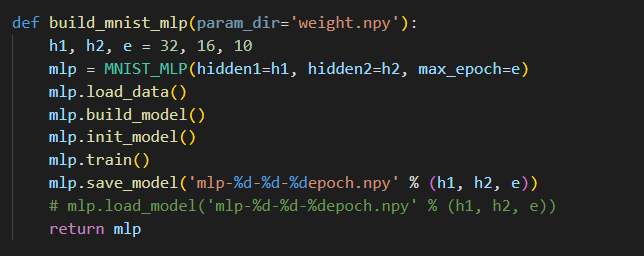
**3.5神经网络结构实现**



**3.6神经网络训练模块与评价模块实现**

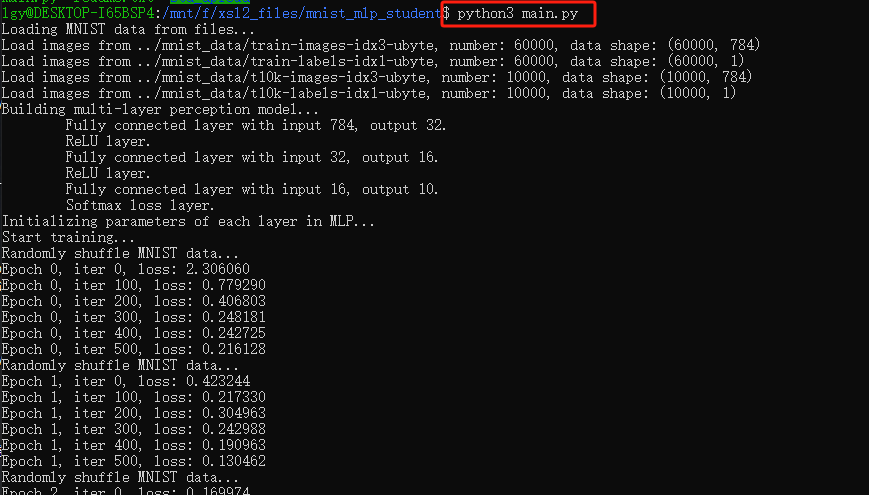






**3.7代码运行**

在mnist\_mlp\_student下输入python3 main.py 模型开始训练



## **四、实验要求**

补全 stu\_upload 中的 layer\_1.py、mnist\_mlp\_cpu.py 文件，执行 main\_exp\_2\_1.py 运行。

要求如下：

1. 根据提供的模型训练代码，训练出手写数字识别模型
2. 根据提供的模型训练代码，加载训练的模型并编写图像推理函数。

3、调整和训练相关的超参数或设计自己的神经网络结构，使最后训练得到的模型在 MNIST 测试数据集上的平均分类正确率高于 98%。