第4章 神经网络基础

苏统华
软件学院
哈尔滨工业大学

本章内容

1. 神经元细胞的数学模型
2. 感知机模型和多层感知机
3. 激活函数
4. 损失函数
5. 参数优化方法
6. 模型评估方法
7. 延展与讨论

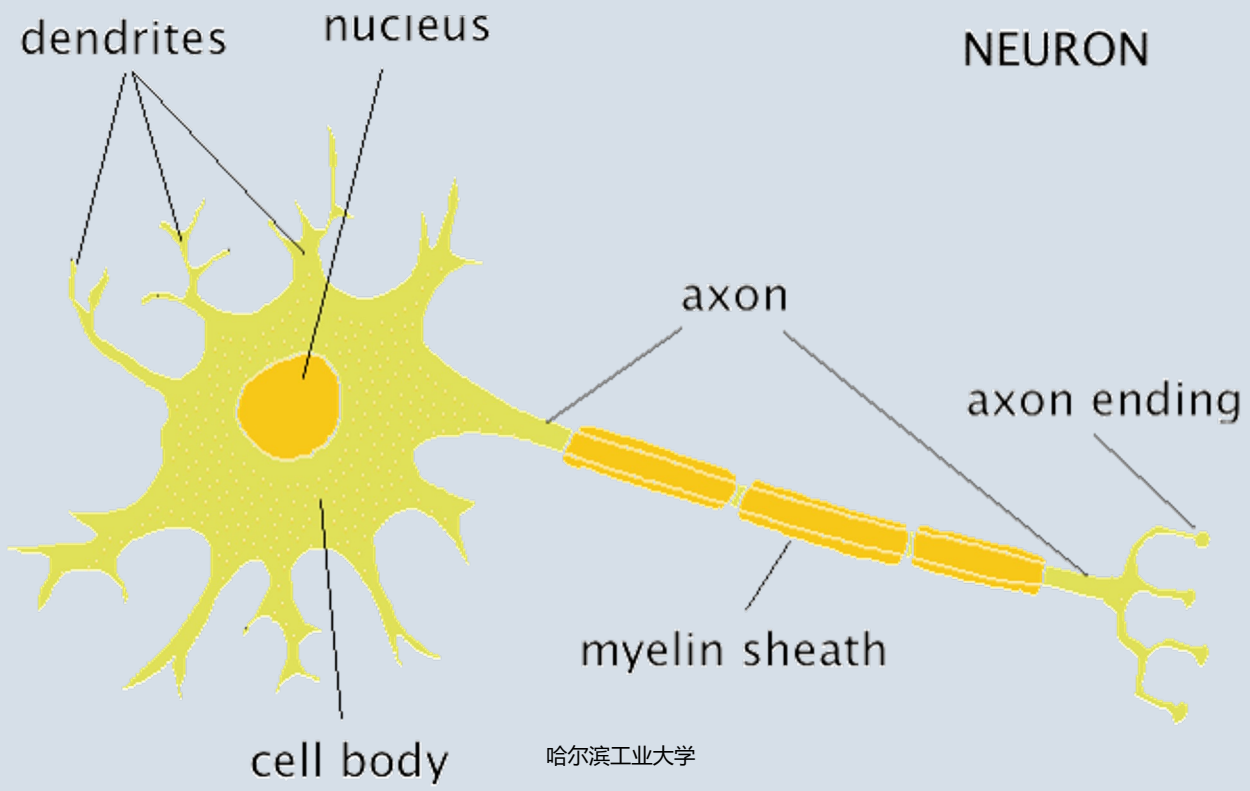
4.1 神经元细胞的数学模型

苏统华
软件学院
哈尔滨工业大学

生物神经元的结构和工作原理

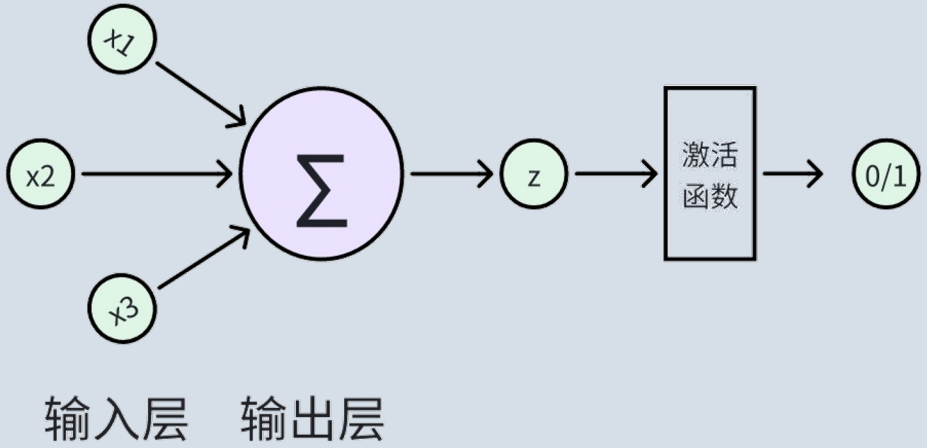
生物神经元主要由以下三个部分组成

- ✓ **树突：接收来自其他神经元的信号**
- ✓ **细胞体：整合并处理来自树突的信号**
- ✓ **轴突：将处理后的信号传输到下一个神经元或目标组织**



M-P模型相关概念

□M-P模型用于模拟神经元的激活过程，如下图所示。该模型将神经元行为简化为一个二进制过程，即输入信号要么使神经元激活（输出为1），要么不激活（输出为0）。



M-P模型相关概念

神经元

- ✓ 每个神经元有若干个输入（x_i）和一个输出（z）。
- ✓ 输入x_i是上一个神经元经过激活函数后的输出，或者是原始的外界信号。
- ✓ 输入和输出均为二进制信号。

权重

- ✓ 每个输入信号对应一个权重w_i，表示输入信号的重要性。

加权和

- ✓ 神经元对所有输入信号乘以其权重值，然后进行累加得到加权和。

激活函数

- ✓ 在M-P模型中，激活函数使用阶跃函数，将加权和转换为二进制输出。

M-P模型设计与生物神经元的对应关系

- **输入与树突：** M-P模型中的输入对应于生物神经元的树突接收的信号。

- **权重与突触强度：** 权重反映了生物神经元中不同突触对输入信号的能力，这类似于突触强度的概念。

- **加权和与细胞体的整合：** 在M-P模型中，各个输入信号通过权重加权后相加，这类似于细胞体对来自树突的信号进行整合的过程。

- **阈值与动作电位的触发：** 阈值对应于生物神经元中触发动作电位所需的信号强度，当加权和超过阈值时，模型输出1，这对应于神经元发放动作电位的信号。

M-P模型工作流程

假设使用M-P模型来实现一个简单的逻辑门，比如AND门

□定义输入、权重和偏移

- ✓ 输入：x_1和x_2
- ✓ 权重：ω_1和ω_2
- ✓ 阈值：θ

□神经元计算输出

- ✓ 神经元计算输出公式为$z = \omega_1 \cdot x_1 + \omega_2 \cdot x_2$

□激活函数

- ✓ 激活函数公式为：$y = \begin{cases} 0 \ \ if \ z < \theta \\ 1 \ \ if \ z \geq \theta \end{cases}$

M-P模型工作流程

工作流程示例

- ✓ 情况1：$x_1 = 0, x_2 = 0$，经过计算得$y = 0$
- ✓ 情况2：$x_1 = 1, x_2 = 0$，经过计算得$y = 0$
- ✓ 情况3：$x_1 = 0, x_2 = 1$，经过计算得$y = 0$
- ✓ 情况4：$x_1 = 0, x_2 = 0$，经过计算得$y = 0$

4.2 感知机模型和多层感知机

苏统华
软件学院
哈尔滨工业大学

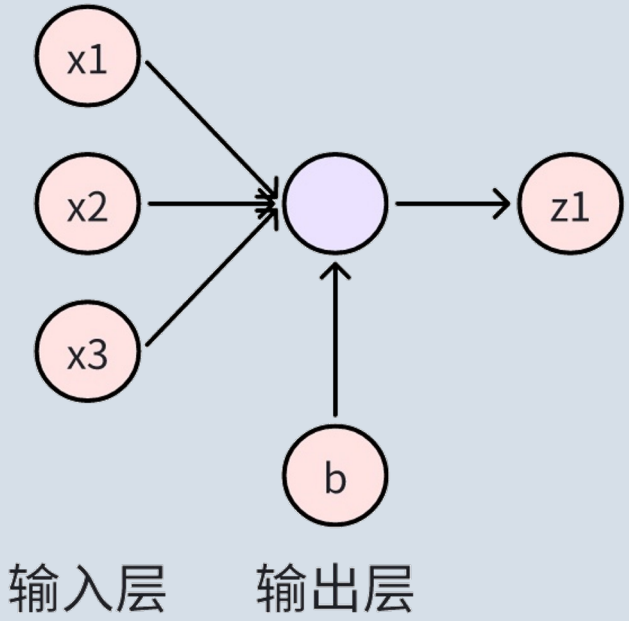
单层感知机

- M-P模型的权重和阈值都是人为设定且固定的，无法学习算法，不具备自我调整的能力，而且输入信号也只能是二进制输入，不能处理连续值输入。

- 1957年，弗兰克•罗森布拉特提出了感知机算法，它比M-P模型具有如下优点：
  - ✓ 可以通过学习算法自动调整权重
  - ✓ 可以处理更多输入情况，输入可以是连续值

单层感知机

口单层感知机模型是感知机算法的具体实现，模型结构如下图所示



单层感知机

- 感知机与M-P模型最大的不同就是感知机的权重和偏置值并非人工设定，而是通过学习进行自我调整。

- 感知机的训练过程通过调整权重和偏置，能够使模型正确分类训练数据。训练过程如下：
  - ✓ 初始化权重和偏置：初始化为0或小的随机值
  - ✓ 对每个训练样本进行如下操作：
    - 计算感知机的输出
    - 根据实际输出与预测输出之间的差异，更新权重和偏置

$$\omega_i \leftarrow \omega_i + \eta\left(y_{true} - y_{pred}\right)x_i$$
$$b \leftarrow b + \eta\left(y_{true} - y_{pred}\right)$$

其中η是学习率，控制更新步伐

单层感知机优缺点

口优点

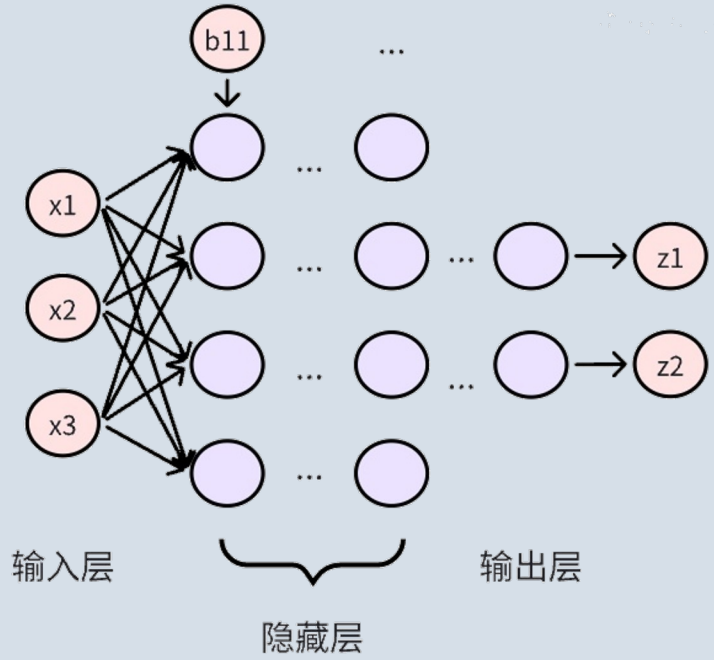
- ✓ 简单易懂：模型结构简单，易于实现
- ✓ 快速训练：训练速度较快，适用于线性可分问题

口缺点

- ✓ 只能解决线性可分问题：单层感知机模型只能解决线性可分的分类问题，对于非线性问题（如异或问题）无能为力
- ✓ 易受噪声影响：训练数据中的噪声和异常值可能会对模型的性能产生较大影响

多层感知机

□单层感知机只能解决线性可分问题，为了克服单层感知机的局限性，研究人员提出了多层感知机（MLP），模型结构如下图所示：



多层感知机

□从单层感知机到多层感知机，模型主要有以下三方面不同

- ✓ **引入隐藏层：**隐藏层位于输入层和输出层之间，增加了网络的复杂性和表达能力。
- ✓ **非线性激活函数：**引入非线性使网络能够学习和表示复杂的关系。
- ✓ **反向传播算法：**计算损失函数对每个权重的梯度，并据此调整权重。

通用近似定理

一个具有至少一个隐藏层且使用非线性激活函数的前馈神经网络，在给定足够多的神经元时，可以以任意精度逼近定义在有限维实数空间上的任何连续函数。

换句话说，单隐藏层神经网络，只要有足够多的神经元，可以逼近任意的连续函数。

通用近似定理

- 假设有连续函数$f(x) = sin(x)$，希望使用单隐藏层的神经网络来逼近它
  - ✓ 选择激活函数：使用Sigmoid函数$\sigma(x) = \frac{1}{1+e^{-x}}$
  - ✓ 构建神经网络：选择合适数量的隐藏层神经元，构建单隐藏层神经网络
  - ✓ 训练神经网络：通过梯度下降和反向传播算法，调整神经网络的权重和偏置，使得输出$g(x)$尽可能接近$f(x)$
- 经过训练后，神经网络的输出$g(x)$将逼近$sin(x)$函数

4.3 激活函数

苏统华
软件学院
哈尔滨工业大学

激活函数及其作用

激活函数的主要作用是为神经网络引入非线性性，以便学习各种复杂的模式和特征，更好地拟合现实世界的高维数据分布。主要作用可以概况为以下两点：

□引入非线性性

如果一个由若干层全连接层构建的神经网络，每层之间不引入激活函数，那么构建的神经网络将仅执行线性变换，无论堆叠多少层，其本质上仍是单层神经网络，也无法解决非线性问题。

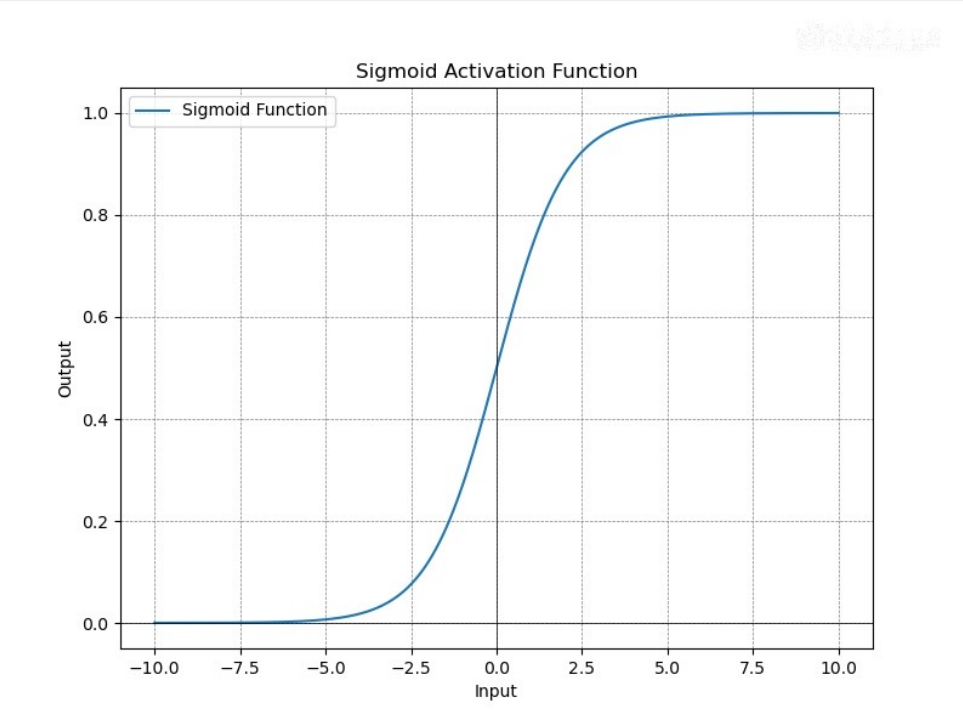
□解决梯度消失和梯度爆炸

梯度消失是指在反向传播过程中，随着网络层数的增加，梯度逐渐变小最终接近于0。梯度爆炸是指梯度在反向传播过程中越来越大，甚至数值溢出。选用合适的激活函数，可以有效缓解梯度消失和梯度爆炸问题。

常用激活函数及其图像

□Sigmoid激活函数

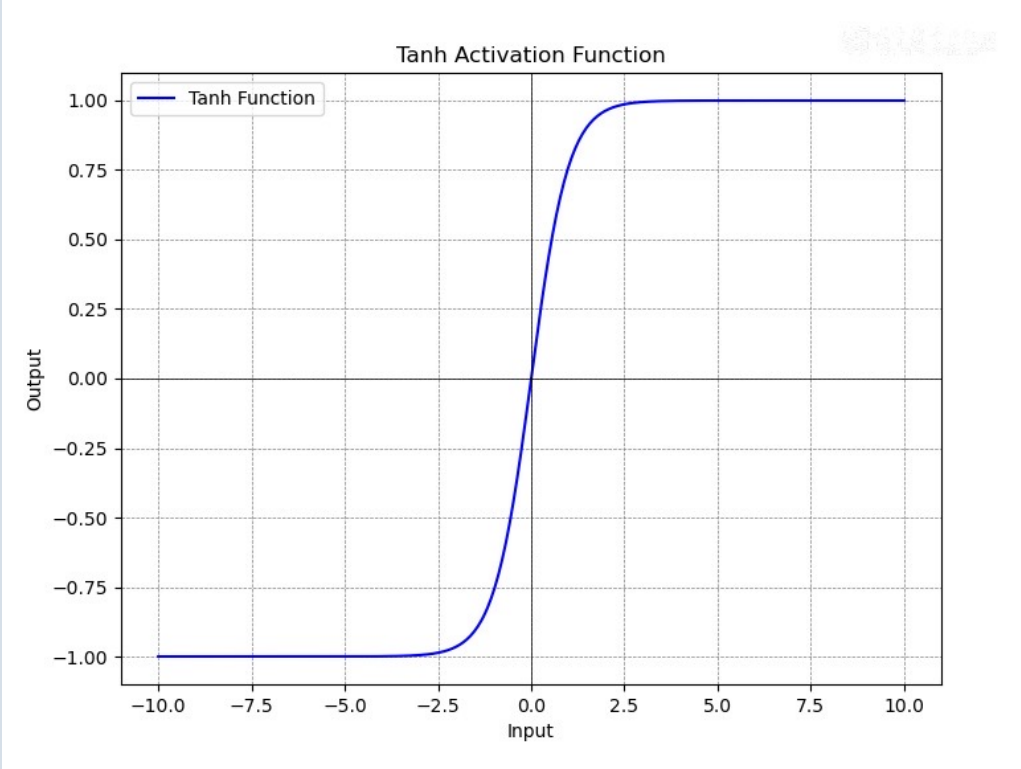
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



常用激活函数及其图像

□Tanh激活函数

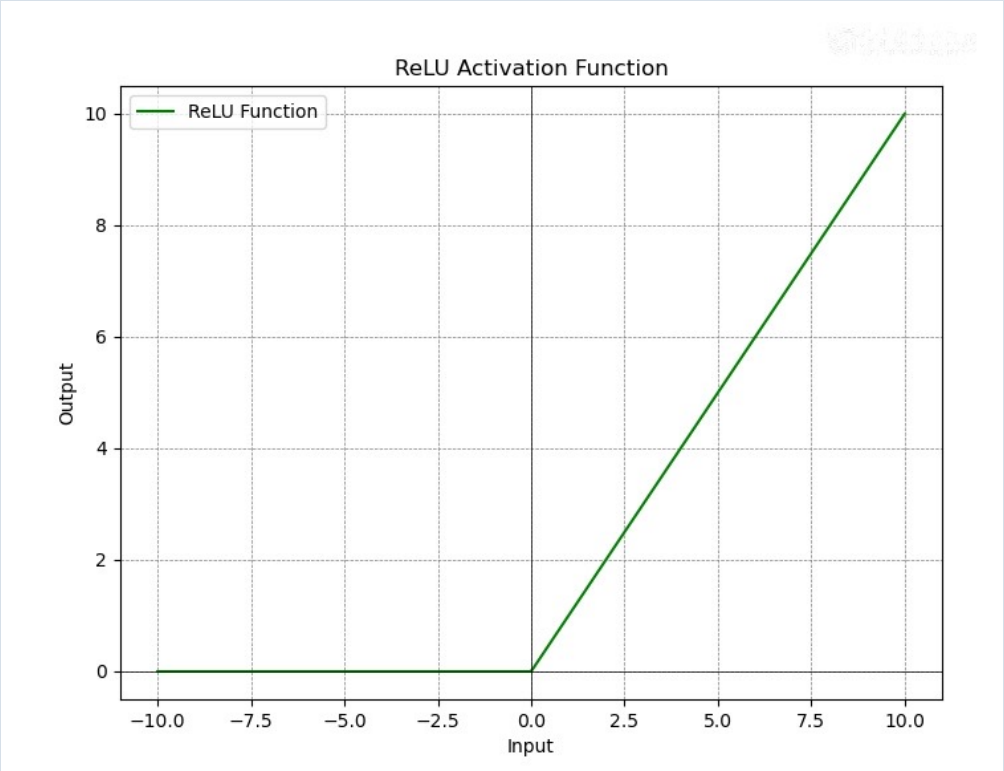
$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$



常用激活函数及其图像

□ReLU激活函数

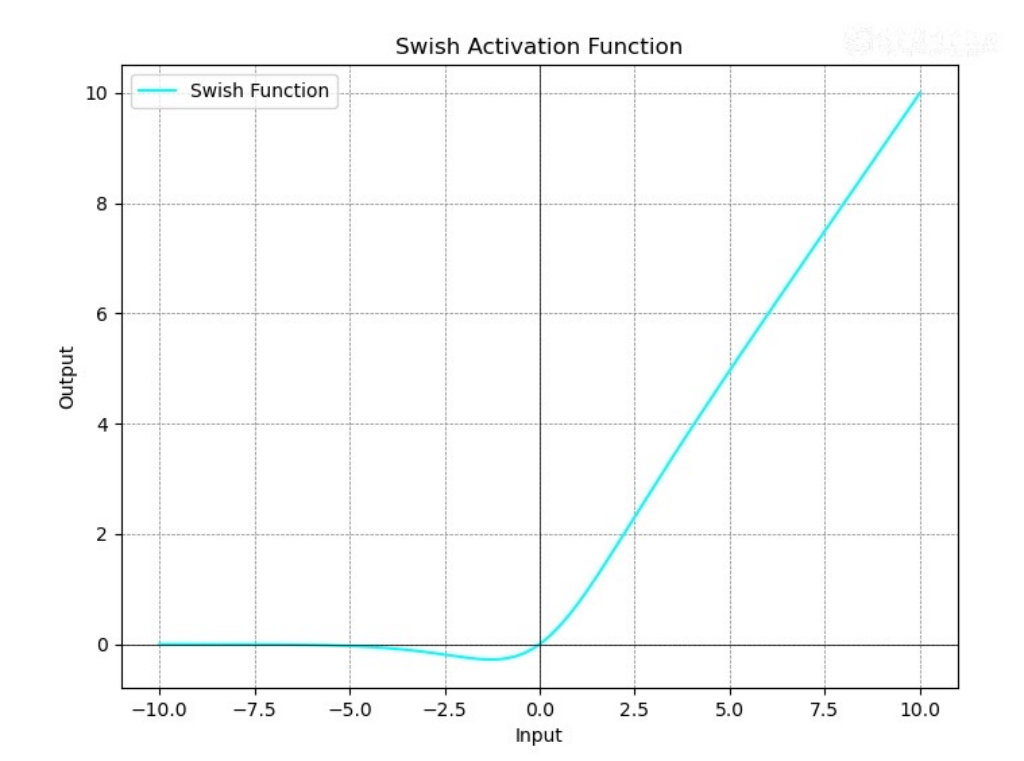
$$ReLU(x) = max(0, x)$$



常用激活函数及其图像

□Swish激活函数

$$Swish(x) = x \cdot \sigma(x) = x \cdot \frac{1}{1 + e^{-x}}$$



4.4 损失函数

苏统华
软件学院
哈尔滨工业大学

损失函数的作用

- 损失函数的主要作用是衡量模型输出的预测值和真实值之间的差异，损失值越小，说明模型的输出越接近真实值，往往也说明模型的性能更好。

- 损失函数的作用可以总结为：
  - ✓ 衡量模型性能
  - ✓ 指导模型训练
  - ✓ 帮助选择模型结构和超参数
  - ✓ 提高模型对噪声和异常值的鲁棒性：

常用损失函数

□0-1损失函数

$$L(y,\hat{y}) = \begin{cases} 0 \ \ if \ y = \hat{y} \\ 1 \ \ if \ y \neq \hat{y} \end{cases}$$

常用于分类问题中，其中y是真实标签，$\hat{y}$是预测标签

常用损失函数

□绝对值损失函数

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

常用于研究回归问题，通过计算预测值与真实值之间的绝对差异来评估模型的性能，对异常值不敏感，具有较好的鲁棒性，其中n是样本数量，y_i是第i个样本的值，$\hat{y}_i$是第i个样本的预测值，$|y_i - \hat{y}_i|$表示真实值与预测值之间的绝对差异。

常用损失函数

□平方损失函数

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

常用于回归任务，通过计算预测值与真实值之间的平方差来评估模型的性能，对异常值较为敏感，计算简单且具有良好的数学性质，其中n是样本数量，y_i是第i个样本的值，$\hat{y}_i$是第i个样本的预测值，$(y_i - \hat{y}_i)^2$表示真实值和预测值之间的平方差

常用损失函数

口交叉熵损失函数

$$Cross-Entropy\ Loss=-\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k}y_{ij}log(\widehat{p_{ij}})$$

主要通过计算预测概率分布和真实标签分布之间的差异来评估模型的性能，其中k是类别数量，y_{ij}是第i个样本的真实标签，$\widehat{p_{ij}}$是第i个样本属于类别j的预测概率。

4.5 参数优化方法

苏统华
软件学院
哈尔滨工业大学

优化方法

目标函数与优化概述

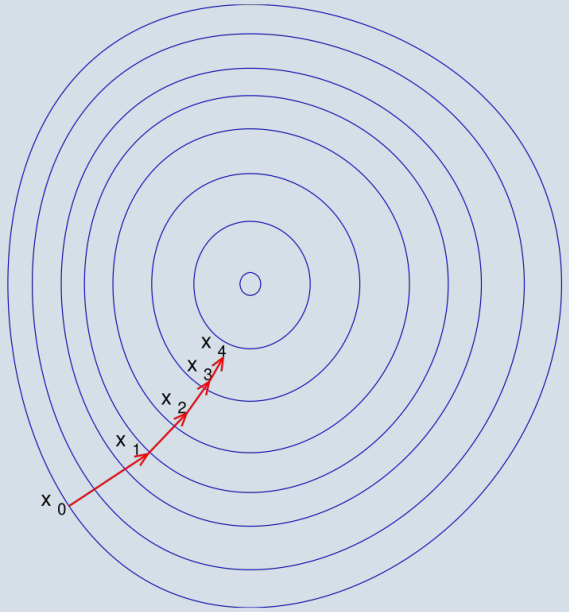
优化是神经网络训练过程中的一个至关重要的环节，其目标是调整网络的参数，以最小化一个称为目标函数（或损失函数）的数学表达式。其数学表达为，寻找一个参数向量x，使得目标函数$f(x)$达到最小值

通常我们记$x^* = \arg min\, f(x)$，表示使$f(x)$最小化的参数向量

优化方法

□梯度下降算法

梯度下降算法是一种基于迭代的方法，用于找到目标函数的最小值。该算法通过沿函数梯度的反方向移动，逐步接近最小值，如下图所示。



优化方法

口动量方法

在非凸损失函数优化过程中，梯度下降算法可能会面临一些问题，例如收敛速度慢、在鞍点或局部最小值处停滞不前等。

为此，研究者引入动量方法，通过在参数更新中加入历史梯度的加权平均数据来加速收敛过程，从而加快收敛速度。

优化方法

自适应学习率优化器

研究人员发现，学习率是训练神经网络中最难设置的超参数之一，它控制着参数更新的步长，能够显著影响模型的性能。因此研究人员提出了多种自适应学习率优化器，例如AdaGrad、RMSProp、Adam等。

以AdaGrad为例，它使用以往梯度的平方来调整学习率，但随着梯度平方和的不断累积，学习率会随着时间不断减小，可能导致后期学习过慢，具体更新规则如下图

$$g_k = \nabla f(x_k)$$

$$G_k = G_{k-1} + g_k \odot g_k$$

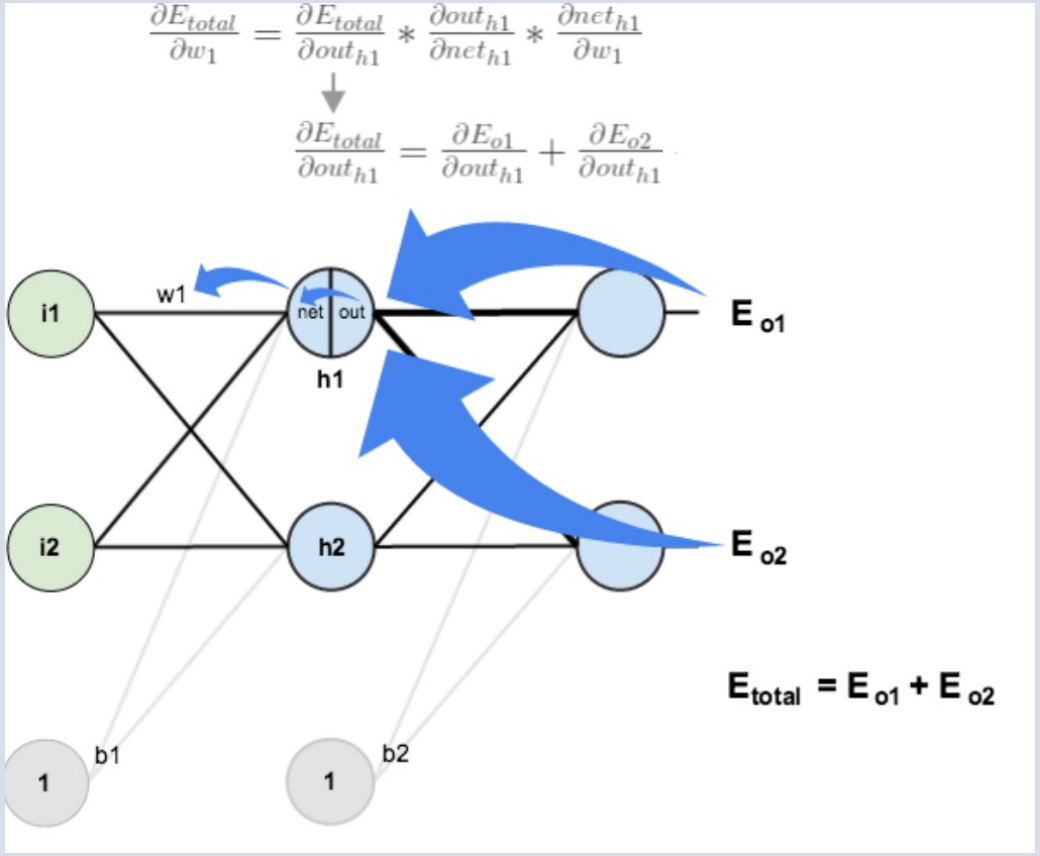
$$x_{k+1} = x_k - \frac{\eta}{\sqrt{G_k + \epsilon}} \odot g_k$$

反向传播算法

□反向传播算法是训练人工神经网络的核心算法之一，它利用链式法则来高效地计算梯度，从而将误差从输出层传递回输入层，逐层更新每个参数的梯度。

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out_{h1}} * \frac{\partial out_{h1}}{\partial net_{h1}} * \frac{\partial net_{h1}}{\partial w_1}$$

$$\frac{\partial E_{total}}{\partial out_{h1}} = \frac{\partial E_{o1}}{\partial out_{h1}} + \frac{\partial E_{o2}}{\partial out_{h1}}$$



参数正则化

- 在神经网络训练过程中，模型很容易产生过拟合现象，即模型在训练数据上表现良好，但在未见过的测试数据上表现较差。
- 为了避免过拟合现象的发生，我们可以使用正则化方法，常见的有：
  - ✓ L2正则化
  - ✓ L1正则化
  - ✓ Dropout正则化

参数正则化

□L2正则化

L2正则化也称为岭回归，是指通过在目标函数中加入权重的平方和来防止过拟合现象的发生。

正则化项为$\Omega(\mathbf{w}) = \frac{1}{2}\|w\|_2$

正则化后的目标函数为$\widehat{L}(w; X, y) = L(w; X, y) + \alpha\Omega(\mathbf{w})$

参数正则化

□L1正则化

L1正则化通过在目标函数中加入权重的绝对值之和来防止过拟合

正则化项为$\Omega(\mathbf{w}) = \|w\|_1$

正则化后的目标函数为$\widehat{L}(w; X, y) = L(w; X, y) + \alpha\Omega(\mathbf{w})$

与L2正则化不同，L1正则化可以导致一些权重完全变为0，从而实现特征选择，这种正则化方法也被称为LASSO回归

参数正则化

□Dropout正则化

Dropout也是一种正则化技术，通过在训练过程中随机“丢弃”一部分神经元来防止过拟合。其实施方法为：

1. 在训练过程中，对于每一层的每个神经元，以概率p将其设置为0

2. 在测试过程中，使用所有神经元但是将每个神经元的输出按比例缩小，即乘以$1-p$，以模拟训练时的行为

4.6 模型评估方法

苏统华
软件学院
哈尔滨工业大学

评估指标

准确率

准确率是指模型预测正确的样本数量占总样本数量的比例，是分类模型最常用的评估指标之一，其计算公式为：

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

其中，TP是预测为正类且实际为正类的样本数，TN是预测为负类且实际为负类的样本数，FP是预测为正类但实际为负类的样本数，FN是预测为负类但实际为正类的样本数。

评估指标

口精确率

精确率是指在所有被预测为正类的样本中，实际为正类的比例，其计算公式为

$$Precision = \frac{TP}{TP + FP}$$

精确率高，意味着模型在预测正类时错误较少

评估指标

□召回率

召回率是指在所有实际为正类的样本中，被预测为正类的样本比例，其计算公式为

$$Recall = \frac{TP}{TP + FN}$$

召回率高，意味着模型能够识别出更多的正类样本

评估指标

□F1分数（F1 Score）

F1分数是精确率和召回率的调和平均数，综合了二者的性能，其计算公式为

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

F1分数高，意味着模型在精确率和召回率之间有较好的平衡

模型评估方法

口基础方法

将数据集划分为训练集、验证集、测试集的方法是最基础的模型评估方法之一

- ✓ 训练集用于模型训练
- ✓ 验证集用于调参和避免过拟合
- ✓ 测试集用于评估模型的最终性能

模型评估方法

□交叉验证

交叉验证是一种可靠的模型评估方法，常见的交叉验证方法包括K折交叉验证方法和留一法交叉验证。

- ✓ K折交叉验证：将数据集随机划分为K份，每次用其中一份作为验证集，其余K-1份作为训练集，重复K次，最后取K次评估结果的平均值
- ✓ 留一法交叉验证：每次只取一个样本作为验证集，其余样本作为训练集，重复N次（N为样本总数），最后取N次评估结果的平均值，该方法开销较大但结果稳定

评估模型泛化能力

模型泛化能力是指模型在处理未见过的数据时所表现出来的能力，即模型在训练数据之外的新数据上的表现效果。

评估模型泛化能力的过程至关重要，因为它直接影响模型在实际应用中的效果。通过一系列评估指标和方法，例如交叉验证、测试集评估、过拟合与欠拟合分析等，可以定量的衡量模型的泛化能力。

4.7 延展与讨论

苏统华
软件学院
哈尔滨工业大学

延展与讨论

□模型训练效率和资源消耗

- ✓ 深度学习模型的训练需要消耗大量算力资源
- ✓ 未来可能在分布式环境中提升训练效率与减少能耗

□神经网络的可解释性

- ✓ 神经网络的决策过程难以被直观解释
- ✓ 未来可能将开发可解释AI等技术

□神经网络的鲁棒性问题

- ✓ 深度学习模型在面对对抗性攻击时表现出脆弱性
- ✓ 未来可能研究更强健的模型，应对数据扰动，提升鲁棒性