7 综述——（前馈）神经网络

本章概述

1943年神经生理学家**沃伦·麦克洛克**（McCulloch）和数学家**沃尔特·皮茨**（Pitts）合作提出的**MP神经元模型**是现代神经网络发展的最初来源，尽管现在已经提出了各种更复杂精细的网络结构和训练方式，但神经元之间的连接、激活以及反馈的思想仍然存在。多个神经元可以形成一张神经网络，其中最自然或最符合直觉的结构便是**前馈神经网络**，即信息从输入层通过隐藏层传递到输出层而不形成反馈回路，单向流动。根据**通用近似定理**（Universal Approximation Theorem），一个前馈神经网络，只要有足够多的隐藏神经元，可以在任意精度下近似任何连续函数。这说明了即使是一个简单的单隐藏层神经网络，也具备足够的表达能力来处理任意复杂的连续函数，奠定了人工神经网络作为一种强大的非线性模型的理论基础，也进一步为更复杂的**深度神经网络**的应用提供了理论基础。当隐藏层的层数达到一定深度，此时神经元的数量也达到一定数量级，便会形成深度神经网络，而这也已经成为了现代神经网络的重要特征。在自然语言处理中，我们可以将前馈神经网络用于**分类**如情感分类，也可以用于构建神**经语言模型**。

最简单的MP模型

MP模型是最早的人工神经网络模型在1943年提出，其用数学的形式描述了神经元的工作方式，可以认为是现代神经网络理论的起源。 MP模型的核心思想是将神经元抽象为一种二值输出的处理单元，通过输入加权求和，然后与一个阈值比较来决定输出。其本质是一个简单的**线性阈值单元**。

MP模型将一个神经元的功能简单化地描述为以下几个部分：

1. 输入：
   输入为一个向量，表示神经元接收的信号。每一个输入通常用 $x_1, x_2, \ldots, x_n$ 来表示。输入可以是二进制的（0或1）或实数值。
2. 权重：
   对每个输入赋予一个权重 $w_1, w_2, \ldots, w_n$ ，表示每个输入对神经元最终输出的贡献度。权重值可以是正的或负的。
3. 线性组合：
   所有输入和权重的乘积进行加和，得到一个总的激活值： $s = \sum_{i=1}^{n} w_i \cdot x_i$
4. 阈值比较（激活函数）：
   MP模型假设每个神经元都有一个阈值 θ 。激活值 s 与阈值 θ 进行比较，决定神经元是否被激活：

$$y = \begin{cases} 1 & \text{if} s \geq \theta \\ 0 & \text{if} s < \theta \end{cases}$$

综上MP模型的数学表达式如下：

$$y = \begin{cases} 1 & \text{if} \sum_{i=1}^{n} w_i \cdot x_i \geq \theta \\ 0 & \text{if} \sum_{i=1}^{n} w_i \cdot x_i < \theta \end{cases}$$

特点与局限性：

- 简单的二值输出： MP模型中的神经元输出只有0和1两种状态，用于模拟神经元“激活”或“抑制”。
- 线性组合与阈值： 该模型使用线性组合来计算神经元的激活状态，通过与阈值比较来确定输出。
- 没有学习能力： MP模型没有学习机制，所有的权重和阈值是预先设定的，而非通过训练数据学习得来的。
- 表达能力有限： 由于模型的简单性，它只能表达一些线性可分的模式（类似于逻辑“与”、“或”、“非”等简单逻辑运算）。

MP模型主要贡献在于，它首次用数学形式模拟了生物神经元的工作机制，这种将神经元抽象为一个“线性加权-阈值激活-二值输出”单元的方式为后来的神经网络模型奠定了核心思想。1958年弗兰克·罗森布拉特（Frank Rosenblatt） 提出的感知机（Perceptron）模型是MP模型的直接延伸与改进。感知机的关键在于引入了一个学习算法，它能够根据输入数据调整神经元的权重，以减少分类误差。感知机模型的提出解决了MP模型静态权重的问题，迈出了自适应神经网络的第一步。

MP模型虽然简单，但其基本思想贯穿于现代神经网络的各个阶段。现代神经网络的各种改进和发展，都是在MP模型所奠定的“线性组合+非线性激活”的基础上不断创新和演化的。

前馈神经网络

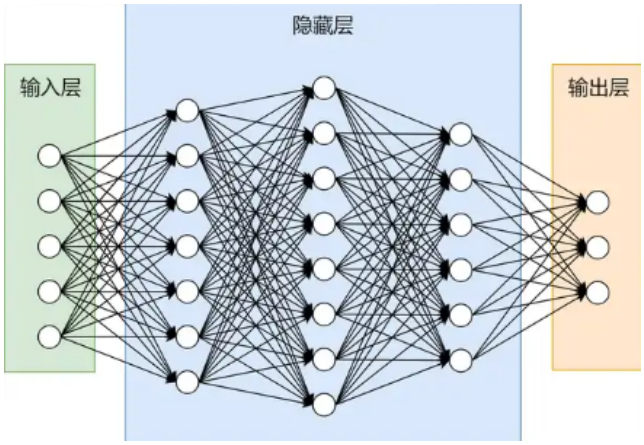
MP模型和感知机的提出奠定了人工神经网络的基础。MP模型通过模拟生物神经元的工作方式，提出了神经元的“线性组合+阈值激活”的机制。感知机在MP模型的基础上引入了权重学习的方法，使得神经元能够通过训练数据进行学习。然而，感知机只能解决线性可分的问题（如 AND、OR 等），而无法处理更复杂的非线性问题（如 XOR 问题）。

为了克服感知机的局限性，我们开始探讨多层神经网络的可能性。多层网络通过在输入层和输出层之间增加隐藏层，使得网络能够进行更复杂的映射。多层感知机是前馈神经网络的雏形，而反向传播算法的出现使训练多层网络结构成为可能。

网络结构及数学形式表达

前馈神经网络是一种最基础的神经网络结构，其主要特点是信息在网络中单向前向传播，从输入层到输出层。具体结构包括：

- 输入层：接收输入数据，输入层的节点数等于数据的特征数。
- 隐藏层（一个或多个）：前馈神经网络中的每个隐藏层节点对上一层的所有节点进行线性组合，并通过激活函数进行非线性转换。隐藏层的数量和节点数可以是可调的。
- 输出层：输出神经网络的结果，输出层的节点数取决于具体任务（如回归问题的一个节点，分类问题的多个节点）。



网络中每一层的节点与上一层的所有节点是全连接的。

假设一个前馈神经网络有 L 层，每一层的节点数为 $n_1, n_2, \ldots, n_L$ ，则前馈神经网络的计算可以表示为：

1. 输入层：设输入数据为 $\mathbf{x}$ ，即第0层的输出 $\mathbf{a}^{(0)} = \mathbf{x}$ 。
2. 隐藏层和输出层：对每一层 l ，有权重矩阵 $\mathbf{W}^{(l)}$ 和偏置向量 $\mathbf{b}^{(l)}$,则第 l 层的线性组合表示为：

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \cdot \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}$$

然后经过激活函数 f ，得到第 l 的输出：

$$\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)})$$

3. 输出层的计算：经过所有的层后，得到输出层的输出 $\mathbf{a}^{(L)}$ ，该输出可以根据具体任务映射为最终的预测结果。

激活函数的选择可以是Sigmoid、ReLU、tanh 等，具体取决于任务和模型的设计。

通用近似定理下的理论能力

通用近似定理指出，一个具有单隐藏层的前馈神经网络（即带有非线性激活函数的多层感知机）可以在任意精度下逼近任意一个连续函数。这一结论为前馈神经网络提供了强大的理论支持，使得它在理论上能够表示任意的复杂函数。通用近似定理的数学表达如下：

设 $f : \mathbb{R}^n \to \mathbb{R}$ 是一个连续函数，则对于任意的误差 $\epsilon > 0$ ，存在一个具有单隐层的前馈神经网络 $F(\mathbf{x})$ ，使得对于 $\mathbf{x} \in \mathbb{R}^n$ ，都有：

$$|f(\mathbf{x}) - F(\mathbf{x})| < \epsilon$$

上式的证明主要利用激活函数的非线性特性和逼近理论Stone-Weierstrass定理来进行。通用近似定理说明前馈神经网络具有足够的表达能力来近似任何连续函数。因此，只要网络结构和参数足够，前馈神经网络能够应对各种复杂的任务。同时赋予了模型设计的灵活性。由于前馈神经网络在理论上可以表示任意函数，因此设计者可以通过选择适当的网络层数、每层的神经元数量以及激活函数来实现不同的任务。

反向传播算法

反向传播算法是训练神经网络的关键方法，核心思想是通过计算损失函数的梯度来更新神经网络中的权重和偏置，从而逐步减少误差。BP算法是现代神经网络的基础，几乎所有的前馈神经网络训练都依赖于这一方法，其应用梯度下降法，通过逐层反向计算误差的梯度，从而更新每一层的权重和偏置。主要包含两个过程：

1. 前向传播：输入数据通过神经网络从输入层到输出层进行传递，并计算每一层的激活值和最终的输出。
2. 反向传播：根据损失函数的导数，从输出层向输入层逐层反向计算误差的梯度，并利用这些梯度更新网络中的权重和偏置。

假设我们有一个前馈神经网络，具有 L 层（包括输入层和输出层）。每一层 l 的神经元数量为 n_l。网络的输入为 $\mathbf{x}$，输出为 $\hat{y}$，目标值为 y。激活函数为 f，损失函数为 $\mathcal{L}(\hat{y}, y)$。

首先通过输入层将输入数据 $\mathbf{x}$ 传递到网络的每一层，计算每一层的线性组合和激活值。对于第 l 层，先线性组合再进行激活输出：

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)} \cdot \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}$$
$$\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)})$$

其中，$\mathbf{W}^{(l)}$ 是第 l 层的权重矩阵，$\mathbf{b}^{(l)}$ 是偏置项，$\mathbf{a}^{(l-1)}$ 是第 $l-1$ 层的激活值。

网络的输出层的输出值为 $\hat{y}$，损失函数 $\mathcal{L}(\hat{y}, y)$ 计算网络输出 $\hat{y}$ 与目标值 y 之间的误差。例如，常用的损失函数有均方误差和交叉熵等。这里假设损失函数为：

$$\mathcal{L} = \frac{1}{2}(\hat{y} - y)^2$$

计算输出层的误差： 输出层的误差是由损失函数的导数引起的。设输出层为第 L 层，基于**链式法则**则输出层的误差表示为：

$$\delta^{(L)} = \frac{\partial \mathcal{L}}{\partial \mathbf{z}^{(L)}} = (\hat{y} - y) \cdot f'(\mathbf{z}^{(L)})$$

其中，f' 是激活函数的导数。

逐层反向计算误差：对于第 l 层（从输出层逐层反向传递），误差 $\delta^{(l)}$ 的计算公式为：

$$\delta^{(l)} = \left(\mathbf{W}^{(l+1)}\right)^T \delta^{(l+1)} \cdot f'(\mathbf{z}^{(l)})$$

这里使用了上一层的误差和当前层的激活函数的导数。

计算权重和偏置的梯度： 通过计算得到每一层的误差 $\delta^{(l)}$，我们可以得到每一层的权重梯度和偏置梯度：

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} = \delta^{(l)} \cdot \left(\mathbf{a}^{(l-1)}\right)^T$$
$$\frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(l)}} = \delta^{(l)}$$

进而使用梯度下降法，根据梯度来更新每一层的权重和偏置：

$$\mathbf{W}^{(l)} \leftarrow \mathbf{W}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}}$$
$$\mathbf{b}^{(l)} \leftarrow \mathbf{b}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial \mathbf{b}^{(l)}}$$

其中，η 是学习率，控制每次更新的步长大小。

局限性分析

前馈神经网络作为最基础的人工神经网络结构之一，有着广泛的应用和理论价值。通过多层神经元的连接，构建了线性和非线性组合的计算结构，使其具有不少优势。当然也存在以下一些局限：

1. 无法处理序列数据： 前馈神经网络的结构是层次化的单向流动，因此它无法处理具有时间或序列特征的数据。例如，对于自然语言、语音信号或时间序列预测等需要考虑上下文关系的任务，前馈神经网络很难捕捉到数据的时序特性。这使得它在这类任务上表现不如递归神经网络或长短期记忆网络。
2. 深度增加时的梯度消失问题： 在前馈神经网络中，随着网络的层数加深，反向传播算法中的梯度可能会逐层衰减，导致网络训练效率低下，无法有效更新参数。这一问题被称为梯度消失问题。尽管激活函数（如 ReLU）和网络结构（如残差网络）可以缓解这一问题，但FFNN仍在深度学习的应用中表现出局限。
3. 对高维数据的处理能力有限： FFNN对高维数据的处理能力较为有限。相比于卷积神经网络（CNN），FFNN在图像等高维数据上无法高效提取局部特征。此外，FFNN的全连接结构使得参数量较大，在处理高维数据时容易导致计算资源的浪费和过拟合问题。
4. 缺乏内在的记忆能力： 前馈神经网络没有内部状态或记忆机制，无法保留和利用历史信息。这使得它无法在需要记忆信息的任务中表现良好，例如自然语言处理中的句子级依赖、机器翻译中的上下文保持等。
5. 无法有效处理长程依赖： 前馈神经网络难以处理具有长程依赖关系的输入数据，例如语音识别、文本生成等任务。对于这种长程依赖，递归神经网络（如 LSTM 和 GRU）和注意力机制（如 Transformer）更加适合。

本章问题

传统的神经元是多输入、单输出的处理单元，即每个神经元接收多个输入后输出一个值。有没有可能设计一种多数输入且多输出的神经元结构，从而带来处理更复杂的任务时具有更高的灵活性和计算效率？