机器学习task0

笔记本：	我的第一个笔记本		
创建时间：	2023/9/18 12:44	**更新时间：**	2023/9/18 13:22
作者：	3446994995@qq.com		
URL：	https://d.jotang.club/t/topic/881/11		

1.什么是机器学习？机器学习和深度学习有什么区别？

什么是机器学习？

机器学习（Machine Learning，简称ML）是人工智能（Artificial Intelligence，简称AI）领域的一个分支，它关注如何通过计算机系统使机器能够从数据中学习并改进性能，而无需明确地编程。机器学习的核心思想是让计算机系统具备从数据中提取模式、进行自动学习和适应的能力，以便能够做出预测、做出决策或完成特定任务。

机器学习和深度学习有什么区别？

机器学习（Machine Learning）和深度学习（Deep Learning）是密切相关的概念，深度学习是机器学习的一个子领域，它们之间存在一些区别，主要体现在以下方面：

1. **方法和技术的范围**：
   - **机器学习**：这是一个更广泛的领域，涵盖了多种不同的方法和技术，包括传统的统计方法、决策树、支持向量机、聚类等等。机器学习方法可以包括浅层模型和深层模型，但不限于深度神经网络。
   - **深度学习**：深度学习是机器学习的一个分支，其核心是构建和训练深度神经网络。深度学习侧重于多层神经网络（深度神经网络）的使用，这些网络可以自动从数据中学习特征表示，通常用于处理复杂的非结构化数据，如图像、语音和自然语言文本。
2. **特征表示的学习**：
   - **机器学习**：在传统机器学习中，特征工程是一个关键的步骤，需要人工设计和选择合适的特征来表示数据。这意味着特征的质量直接影响了模型的性能。
   - **深度学习**：深度学习的一个主要优势是可以自动地学习特征表示。深度神经网络可以通过多个层次自动抽取和转换数据的特征，从而减轻了特征工程的负担。
3. **数据量和计算资源**：
   - **机器学习**：传统机器学习方法通常对数据量和计算资源的要求较低，可以在相对较小的数据集上运行。

- **深度学习**：深度学习通常需要大量的标记数据和更强大的计算资源（如GPU或TPU），因为深度神经网络通常包含大量参数，需要大规模训练来获得良好的性能。

4. **应用领域**：
   - **机器学习**：机器学习方法广泛应用于各种领域，包括传统的统计建模、推荐系统、聚类、回归等等。
   - **深度学习**：深度学习在计算机视觉、自然语言处理、语音识别等领域取得了显著的成功，尤其是处理大规模非结构化数据时表现出色。

总之，深度学习是机器学习的一个子集，它专注于使用深度神经网络来自动学习数据的特征表示，适用于处理大规模和复杂的数据，但需要更多的数据和计算资源。机器学习则是一个更广泛的领域，包括多种方法和技术，可以在各种应用中找到用武之地。选择机器学习还是深度学习通常取决于具体问题和可用资源。

2.请简述监督学习与无监督学习的概念，给出你对无监督学习和监督学习的自己理解和看法。

监督学习（Supervised Learning）：

- 监督学习是一种机器学习任务，其核心思想是从有标签的训练数据中学习一个模型，以便模型能够对新的未见数据进行预测或分类。
- 在监督学习中，训练数据包括输入特征和相应的目标标签。模型通过学习输入和目标标签之间的关系，以便能够根据输入来预测或分类目标标签。
- 常见的监督学习任务包括分类（将输入数据分为不同的类别）和回归（预测连续值）。

无监督学习（Unsupervised Learning）：

- 无监督学习是一种机器学习任务，其目标是从未标记的数据中发现模式、结构或关系，而不需要目标标签的指导。
- 无监督学习通常用于数据探索、聚类、降维和生成模型等任务。聚类是将数据分组到相似的簇中，降维是减少数据的维度以保留关键信息，生成模型则是学习生成数据的分布模式。
- 无监督学习在数据挖掘和特征工程等领域有广泛应用。

自己的理解和看法：

1. **监督学习**：监督学习是一种强大的学习方法，因为它允许我们通过提供标签数据来教导模型，使其能够进行准确的预测和分类。这对于许多实际问题非常有用，例如垃圾邮件过滤、图像识别和自然语言处理。然而，监督学习的一个限制是需要大量的标记数据，而且对于某些任务，获取标签数据可能非常昂贵或困难。

2. **无监督学习**：无监督学习在数据探索和理解中扮演着重要的角色，它可以帮助我们揭示数据中的隐藏模式和结构。无监督学习在聚类、降维和生成模型等任务中都非常有用，可以帮助我们更好地理解数据。此外，无监督学习还可以用于半监督学习和自监督学习等领域，为监督学习提供额外的帮助。

总的来说，监督学习和无监督学习是机器学习中的两个重要分支，各自适用于不同类型的问题。理解它们的原理和应用场景可以帮助选择适当的学习方法来解决特定的任务。此外，深度学习已经使得监督学习和无监督学习之间的界限变得模糊，因为深度神经网络可以用于各种类型的学习任务，包括有监督和无监督学习。

3.偏导数是什么？链式法则是什么？梯度又是什么？矩阵乘法怎么操作？请仔细思考他们在机器学习/深度学习中的作用。

1. **偏导数**：偏导数是多元函数中的导数的一种形式，用于衡量函数关于其中一个变量的变化率，同时将其他变量保持不变。如果一个函数有多个输入变量（例如，函数是一个多元函数），则可以计算每个变量的偏导数。偏导数通常用符号∂表示。
2. **链式法则**：链式法则是微积分中的一个重要原理，用于计算复合函数的导数。它允许我们将一个复杂的函数拆分为多个简单的函数，并计算它们的导数，然后通过链式法则将它们组合起来以计算整个函数的导数。在机器学习中，链式法则用于计算神经网络中每个参数的梯度，这对于梯度下降等优化算法至关重要。
3. **梯度**：梯度是多元函数的偏导数组成的向量，表示函数在每个变量方向上的变化率。对于单变量函数，梯度是导数；而对于多变量函数，梯度告诉我们函数在每个方向上的变化速度。在机器学习中，梯度通常用于计算损失函数相对于模型参数的导数，以便进行梯度下降等优化。
4. **矩阵乘法**：矩阵乘法是一种线性代数操作，用于将两个矩阵相乘以产生新的矩阵。左横右列。

在机器学习/深度学习中的作用：

- **偏导数**：在机器学习中，我们需要计算损失函数相对于模型参数的偏导数，以便进行参数更新。这些偏导数告诉我们如何调整模型参数以最小化损失函数。
- **链式法则**：链式法则用于计算复杂神经网络中每个参数的梯度。它允许我们有效地传播梯度并计算模型中每个参数的更新方向。
- **梯度**：梯度是损失函数相对于模型参数的导数，它告诉我们如何在参数空间中移动以减小损失。梯度下降是深度学习中常用的优化算法，用于训练神经网络。
- **矩阵乘法**：矩阵乘法在神经网络中用于权重和输入数据之间的线性变换。它允许网络层之间的信息传递，并在前向传播和反向传播中用于计算梯度和更新权重。

这些数学概念和操作是深度学习和机器学习中不可或缺的部分，它们帮助我们理解模型如何学习和优化，以及如何进行高效的参数更新和训练。深刻理解它们对于从事机器学习和深度学习的工作非常重要。

4.什么是损失函数？梯度下降的原理？反向传播的原理？

损失函数（Loss Function）是机器学习和深度学习中的一个关键概念，它是用来衡量模型预测与实际目标之间差异的函数。损失函数的目标是在训练过程中衡量模型的性能，帮助

优化算法调整模型参数以最小化这个损失值。不同的任务和问题可能需要不同的损失函数。

梯度下降（Gradient Descent）是一种优化算法，用于最小化损失函数。其基本原理如下：

1. **初始化**：选择初始模型参数的值，通常是随机选择。
2. **计算梯度**：计算损失函数相对于模型参数的梯度（导数）。梯度告诉我们损失函数在参数空间中的变化速率，即损失函数增加最快的方向。
3. **更新参数**：根据梯度的反方向，以一定的学习率（learning rate）调整模型参数。学习率决定了每次参数更新的步长。
4. **重复迭代**：重复步骤2和3，直到满足停止条件，如达到最大迭代次数或损失函数收敛到某个阈值。

梯度下降的目标是找到损失函数的局部最小值或全局最小值，从而使模型能够更好地拟合训练数据。

反向传播（Backpropagation）是用于训练神经网络的一种特定形式的梯度下降。其基本原理如下：

1. **前向传播**：将输入数据通过神经网络，计算每一层的输出。
2. **计算损失**：计算模型的预测与实际标签之间的差异，从而得到损失值。
3. **反向传播误差**：从输出层开始，计算每一层的梯度，这是损失函数相对于每个参数的导数。这是通过链式法则来计算的。
4. **参数更新**：使用梯度下降的原理，根据梯度的反方向和学习率来更新每一层的参数。
5. **重复迭代**：重复前向传播、计算损失、反向传播误差和参数更新，直到满足停止条件，通常是达到最大迭代次数或损失函数收敛到某个阈值。

反向传播允许神经网络通过调整权重和偏差来减小预测值与实际标签之间的误差，从而提高模型的性能。

总之，损失函数是用于衡量模型性能的函数，梯度下降是一种优化算法，用于最小化损失函数，而反向传播是用于训练神经网络的梯度下降算法的特例。这些原理是深度学习中的基础，使神经网络能够逐渐调整参数以学习从数据中提取特征和进行预测。

5.什么是样本（sample）？什么是特征（feature）？为什么要使用激活函数?

在机器学习和深度学习中，有几个重要的概念，包括样本（sample）、特征（feature）和激活函数（activation function）：

1. **样本（Sample）**：
    - 样本是指在数据集中的单个数据点或观测值。在监督学习中，每个样本通常包括输入数据和相应的目标标签。例如，对于图像分类任务，一个样本可能是一张图

像（输入数据）以及该图像所属的类别标签（目标标签）。

2. **特征（Feature）**：
   - 特征是用来描述样本的属性或信息的变量。特征可以是数值、类别、文本或任何可以表示样本的属性。在机器学习中，我们通常将每个样本表示为一个特征向量，其中每个特征对应特定的属性或测量值。例如，在房价预测任务中，特征可以包括房屋的面积、卧室数量、浴室数量等。
3. **激活函数（Activation Function）**：
   - 激活函数是神经网络中的一个关键组成部分，用于在神经元中引入非线性性。神经网络的层中的每个神经元都具有激活函数，它将神经元的输入转换为输出。激活函数的引入使神经网络能够学习复杂的非线性关系，从而增加了模型的表达能力。
   - 常见的激活函数包括Sigmoid、ReLU（Rectified Linear Unit）、Tanh（双曲正切）等。每种激活函数都具有不同的性质，适用于不同类型的任务和网络结构。

为什么要使用激活函数：

激活函数的主要作用是引入非线性性。如果没有激活函数，整个神经网络将由线性变换组成，多层线性变换的叠加仍然是线性变换。这将导致神经网络无法学习复杂的非线性关系，因为多层线性变换等效于单层线性变换。

通过使用非线性激活函数，神经网络可以模拟更复杂的函数，并从数据中学习到非线性模式。这增加了神经网络的表达能力，使其能够解决更广泛的问题，例如图像识别、自然语言处理和语音识别等。因此，激活函数是神经网络中不可或缺的组成部分，它使网络能够适应各种复杂的数据分布和任务。

6.请简述线性回归和逻辑回归的概念与基本原理。通过学习，总结出线性回归和逻辑回归的联系与区别。

线性回归和逻辑回归是两种常见的机器学习算法，它们用于不同类型的问题，并具有不同的概念和基本原理。以下是它们的概念、基本原理以及联系与区别的总结：

线性回归（Linear Regression）：

概念：
线性回归是一种用于解决回归问题的监督学习算法。它的目标是建立一个线性模型，用于预测连续数值输出（如房价、销售额等），基于输入特征的线性组合。

基本原理：

- 模型形式：线性回归模型的形式为
  $$y = b_0 + b_1x_1 + b_2x_2 + \ldots + b_px_p$$
  ，其中 y 是输出变量，x_1, x_2, ..., x_p 是输入特征，b_0, b_1, b_2, ..., b_p 是模型参数。

- 目标：通过调整模型参数，使预测值与实际观测值的平方差最小化，通常使用最小二乘法。
- 输出：线性回归的输出是一个连续数值，它可以是任意实数。

逻辑回归（Logistic Regression）：

概念：
逻辑回归是一种用于解决分类问题的监督学习算法。尽管名字中带有"回归"，但实际上它是一种分类模型。逻辑回归通过建立一个逻辑函数来估计观测数据点属于某个类别的概率，然后根据概率进行分类。

基本原理：

- 模型形式：逻辑回归模型使用逻辑函数（Sigmoid函数）建模，其形式为
  $$P(Y=1) = \frac{1}{1+e^{-(b_0+b_1x_1+b_2x_2+\ldots+b_px_p)}}$$
  ，其中 P(Y=1) 表示观测数据点属于类别1的概率，b_0, b_1, b_2, ..., b_p 是模型参数。
- 目标：通过最大化似然函数或最小化对数损失函数来学习模型参数，通常使用梯度下降等优化算法。
- 输出：逻辑回归的输出是一个介于0和1之间的概率值，通常用于二分类问题的阈值判定。

联系与区别：

联系：

1. 两者都是监督学习算法，需要有标签的训练数据进行训练。
2. 两者都使用线性模型，但逻辑回归引入了逻辑函数来建模概率。

区别：

1. 问题类型：线性回归用于解决回归问题，逻辑回归用于解决分类问题。
2. 输出类型：线性回归的输出是连续数值，逻辑回归的输出是表示概率的值。
3. 模型形式：线性回归使用线性方程，逻辑回归使用逻辑函数。
4. 损失函数：线性回归通常使用均方误差作为损失函数，逻辑回归使用对数损失或交叉熵损失。

总结：线性回归和逻辑回归都是基于线性模型的算法，但用途和模型形式不同，前者用于回归问题，后者用于分类问题。逻辑回归的输出经过逻辑函数映射，表示为概率值，可以用于二分类问题。

在机器学习中，为什么要将数据集划分成训练集、验证集和测试集？三者之间区别与联系，以及数据集不正确划分造成的影响？我们该如何正确的划分和使用数据集

将数据集划分成训练集、验证集和测试集是为了有效评估和改进机器学习模型的性能。这个划分有助于模型的开发和调优，以及防止过拟合（overfitting）问题。以下是它们之间的区别与联系，以及不正确划分数据集可能带来的影响：

1. **训练集（Training Set）**：
    - 训练集是用于训练机器学习模型的数据子集。
    - 模型根据训练集中的数据来学习特征和关系，调整模型参数，以使模型能够拟合训练数据。
    - 训练集用于模型的参数估计和训练过程。
2. **验证集（Validation Set）**：
    - 验证集是用于评估模型性能和进行超参数调优的数据子集。
    - 在训练过程中，使用验证集来验证模型在未见过的数据上的性能，以及选择最佳的模型超参数（如学习率、正则化参数等）。
    - 验证集有助于避免模型在训练集上过拟合，帮助选择更泛化的模型。
3. **测试集（Test Set）**：
    - 测试集是用于最终评估模型性能的数据子集。
    - 模型在测试集上的性能评估通常在模型训练和验证之后进行，用于模型的最终性能报告。
    - 测试集通常应该与训练集和验证集是相互独立的，模型未曾在测试集上进行过训练或验证。

联系与区别：

- 训练集用于模型的训练和参数估计，验证集用于模型性能的评估和超参数调优，而测试集用于最终模型性能的评估。
- 训练集和验证集通常在模型开发和调优过程中反复使用，而测试集只用于最终性能评估，模型不应在测试集上进行调优。
- 这三个子集的数据应该相互独立，确保模型在未曾见过的数据上的性能评估是准确的。

不正确划分的影响：
如果数据集没有正确划分为训练集、验证集和测试集，或者划分不合理，可能会导致以下问题：

1. **过拟合问题**：模型可能在训练集上表现良好，但在未见过的数据上性能差，因为没有独立的验证集用于调优。
2. **无法评估泛化性能**：没有独立的测试集，无法准确评估模型在真实世界中的性能。
3. **超参数选择困难**：没有验证集，超参数的选择将更具随机性，可能导致子优的超参数选择。
4. **模型泄漏**：如果模型在测试集上进行了调优，测试集可能不再是独立的，模型性能评估将失去可信度。

正确划分和使用数据集：

1. 将数据集划分为训练集、验证集和测试集，并确保它们相互独立。

2. 使用训练集训练模型，使用验证集进行超参数调优，最终使用测试集评估模型的性能。
3. 不要在测试集上进行任何形式的调优，以确保性能评估的客观性。
4. 可以采用交叉验证等技术来更充分地利用有限的数据，特别是在数据量有限的情况下。
   划分数据集成训练集、验证集和测试集的比例通常没有一个固定的标准，它会依赖于你的具体问题、数据的大小和性质以及训练模型的需求。然而，有一些常见的比例建议可以作为参考：

常见的比例建议：

- 70%训练集、15%验证集、15%测试集：这是一个常见的划分比例，特别适用于中等大小的数据集。
- 60%训练集、20%验证集、20%测试集：这是另一个常见的比例，也适用于中等大小的数据集。
- 80%训练集、10%验证集、10%测试集：如果数据集较大，可以考虑分配更多数据给训练集。

数据量有限时的考虑：

- 如果数据量非常有限，可以考虑使用交叉验证，将数据多次划分为训练集和验证集，以更充分地评估模型性能。
- 例如，5折交叉验证将数据划分为5个部分，每次将其中4个作为训练集，1个作为验证集，重复5次，然后取平均性能作为最终评估。

不同数据集特点的考虑：

- 数据集的特点可能会影响划分比例。例如，如果数据集中有严重的类别不平衡问题（某一类样本数量远大于其他类），则可能需要更谨慎地划分验证集和测试集，以确保每个子集中都有足够的样本来代表不同的类别。

总之，划分数据集的比例应该根据具体情况来调整，并且需要考虑到数据集的大小、问题的复杂性、类别平衡等因素。在进行机器学习项目时，通常会根据实验和交叉验证的结果来优化划分比例，以确保模型性能的客观评估和泛化能力的提高。

8.softmax函数是什么？其在机器学习中有什么应用？

Softmax函数是一种常用的数学函数，通常用于多类别分类问题的机器学习模型中。它将一组数值转化为表示概率分布的形式，其中每个数值表示一个类别的概率。

Softmax函数的定义如下，对于给定的输入向量z = z_1, z_2, ..., z_k]，它将每个 z_i 转化为对应类别 i 的概率P(y=i|z)

$$P(y=i|\mathbf{z}) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

其中，k 表示类别的总数。Softmax函数对输入向量中的每个元素进行指数化（取幂），然后将它们归一化为总和为1的概率分布。

Softmax函数在机器学习中的应用主要涉及多类别分类问题，特别是在深度学习中的神经网络模型中，如多层感知器（MLP）和卷积神经网络（CNN）。以下是Softmax函数在机器学习中的主要应用：

1. **多类别分类**：Softmax函数常用于多类别分类任务，其中模型需要将输入数据分为多个不同的类别。通过Softmax函数，模型可以将原始的类别得分转化为概率分布，然后选择具有最高概率的类别作为最终的预测结果。
2. **神经网络输出层**：在深度学习中，神经网络的输出层通常使用Softmax函数，以生成每个类别的概率分布。这对于图像分类、文本分类、语音识别等任务非常有用。
3. **多标签分类**：Softmax函数可以扩展到多标签分类问题，其中每个样本可以属于多个类别。在这种情况下，Softmax函数会生成每个类别的概率，模型可以根据概率来决定样本是否属于某个类别。
4. **模型评估**：Softmax函数还常用于模型评估，特别是在多类别分类问题中。通过将模型的输出转化为概率分布，可以计算交叉熵损失（Cross-Entropy Loss）来评估模型的性能。

总之，Softmax函数在多类别分类问题中起着关键作用，它将模型的原始输出转化为概率分布，使模型能够进行分类预测并进行训练和评估。