|  |  |  |
| --- | --- | --- |
| 第二章 神经网络设计实验 | | |
| 姓名： | 学号： | 班级： |
| **注意：课程实验代码和报告需独立完成，严禁抄袭!如发现有抄袭情况（同年级或者抄袭高年级），相应的实验或报告部分直接0分。** 实验2.1基于三层神经网络实现手写数字分类(60’’) **1.介绍一下“手写数字分类任务”和本实验中使用的数据集。**  "手写数字分类任务" 是指一个常见的机器学习任务，其目标是将手写数字的图像分类为0到9之间的数字。这种任务通常用于计算机视觉和深度学习领域，是一个入门级别的图像分类问题。手写数字分类任务有很多应用，包括邮政服务的邮编识别、银行支票的自动处理、手写数字识别软件以及数字验证码识别等。  在本实验中，我们通过三层全连接神经网络，在前两个全连接层后添加了激活函数RELU层，最后一个全连接层添加损失softmax层的方式来实现手写数字分类任务。  MNIST MNIST 手写数字库。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 数据集通常被用来测试和验证机器学习和深度学习算法的性能。  MNIST 数据集的主要特点包括：  图像小而简单：每张图像都是28x28像素，只包含单一的手写数字，使它们易于处理。  标签明确：每张图像都有与之关联的正确标签，即包含的数字，这使得监督学习任务容易实施。  大量的样本：数据集中包含了足够多的图像，可以用于训练、验证和测试模型的性能。  广泛应用：MNIST 数据集被广泛用于教育、研究和算法验证，它已经成为了一个经典的基准测试数据集。  机器学习和深度学习研究人员通常使用MNIST数据集来评估他们的分类算法，并研究不同模型的性能。此外，MNIST数据集也可以用于教育，作为入门级别的练习，帮助学生理解图像分类和机器学习的基本概念。  **2.本实验的神经网络都包含哪些层？各个层的功能是什么？**  在该神经网络中，包含三个全连接层，前两个连接层后有relu层，最后一个全连接层后有softmax层。  三个全连接层在神经网络中的作用如下：  第一个全连接层：  输入层：接受原始输入数据，通常是扁平化的图像数据或特征向量。  作用：将输入数据通过权重和偏差的线性变换映射到中间特征空间，使模型能够提取输入数据中的关键特征。  第二个全连接层：  中间层：接受来自第一个全连接层的输出。  作用：继续对来自第一个全连接层的特征进行线性变换，进一步提取和组合特征，以更好地捕捉数据中的模式和信息。  第三个全连接层：  输出层：接受来自第二个全连接层的输出。  作用：将中间特征映射到输出空间，通常用于多类别分类问题。每个神经元代表一个类别，并对应一个可能的类别标签。  ReLu层作用  非线性激活：ReLU 激活函数对于正数输入，保持不变，而对于负数输入，将输出置为零。这种非线性操作允许神经网络学习非线性映射，因为它在输入的不同范围内产生不同的输出，从而有助于捕捉数据中的非线性关系。  特征提取：每个 ReLU 层都有多个神经元，每个神经元对应于网络中的不同特征。通过 ReLU 激活，网络可以选择性地激活或抑制这些特征，从而允许网络更好地学习和表示数据中的不同特征。这有助于提取数据中的有用特征，以用于后续任务，如分类或回归。  Softmax层作用：  多类别分类：Softmax 层用于多类别分类任务，其中有多个类别需要进行分类。它将原始的类别分数映射为每个类别的概率，这些概率表示输入数据属于每个类别的可能性。  概率归一化：通过应用 Softmax 函数，原始分数被转化为正数值，并进行概率归一化，以确保所有类别的概率之和等于 1。这使得每个类别的输出可以被解释为概率分布，便于理解和解释模型的预测。  预测类别：在推断或测试阶段，Softmax 层将神经网络的输出转化为每个类别的概率，然后可以根据最高概率来预测输入数据属于哪个类别。通常，具有最高概率的类别被认为是模型的最终预测。  **3.网络的训练(training)过程和网络的推理(inference)过程有什么区别？**  训练过程：  目标：训练神经网络的主要目标是学习从输入数据到输出数据之间的映射关系，以便网络可以执行特定的任务，如图像分类、语音识别、自然语言处理等。在这个过程中，网络要不断地调整其参数，以最小化损失函数的值，损失函数衡量了网络的输出与实际标签之间的误差。  数据：训练过程需要大量的标记数据，包括输入数据和对应的目标标签。这些数据用于通过梯度下降等优化算法来更新网络的权重和偏置，以提高网络在训练数据上的性能。  优化：在训练过程中，神经网络通过反向传播算法来计算损失函数相对于网络参数的梯度，然后使用这些梯度来更新参数。通常，这个过程需要多次迭代，以不断改进网络的性能。  计算成本：训练过程通常需要大量的计算资源和时间，因为每个数据点都要经过前向传播和反向传播来进行参数更新。通常，这会在专门的硬件上进行，如图形处理单元或专用的深度学习加速器。  推理过程：  目标：推理是在已经训练好的神经网络上进行的，其目标是使用网络来进行实际任务的预测或分类，而不是调整网络的参数。推理过程的主要目标是高效地处理输入数据，产生输出。  数据：推理过程通常使用来自真实应用场景的数据，而不是训练时使用的数据。这些数据可以是实时生成的或来自外部源。  优化：与训练不同，推理过程无需进行反向传播和参数更新。因此，它通常可以更快地执行，因为网络的权重已经固定。  计算成本：推理过程通常需要较少的计算资源，因为不需要更新参数，而只需要进行前向传播以获取输出。这可以在通用的计算硬件上执行，如中央处理单元或较小规模的硬件设备。  **4.参数和超参数的区别是什么？本实验中都使用了哪些超参数？**  参数：参数是模型内部可学习的组成部分，它们直接影响模型的性能。参数通常由训练数据来调整，通过反向传播算法（例如梯度下降）来更新，以使模型能够适应训练数据。  超参数：超参数是模型训练过程中需要手动设置的配置选项，它们不是直接从训练数据中学习的，而是通过试验和经验来确定。  在本实验中，超参数有  batch\_size ： 每个训练迭代（或批次）中包含的样本数量  hidden1 hidden2 ： 两个隐藏层神经元数量  Lr ： 学习率  max\_epoch ： 最大迭代次数  **5.什么是正则化？有哪些正则化技术？**  正则化是一种用于机器学习和深度学习的技术，旨在帮助防止模型过度拟合训练数据。过度拟合是指模型在训练数据上表现得很好，但在未见过的数据上表现不佳的情况。正则化通过向模型的损失函数中添加额外的项或约束来对模型进行限制，以防止它过度拟合。  以下是一些常见的正则化技术：  L1 正则化（L1 Regularization）：也称为Lasso正则化，它通过在损失函数中添加模型权重的绝对值之和，来鼓励模型中的一些权重变为零。这有助于特征选择，因为某些特征的权重会被归零，从而减少了模型的复杂性。  L2 正则化（L2 Regularization）：也称为Ridge正则化，它通过在损失函数中添加模型权重的平方和，来防止权重过大。这有助于减小权重之间的差异，从而提高模型的稳定性。  弹性网络（Elastic Net）：弹性网络是L1和L2正则化的组合，它在损失函数中同时使用L1和L2正则化项。这提供了一种平衡，可以在某些情况下更好地处理多重共线性。  最大范数正则化（Max-Norm Regularization）：最大范数正则化通过限制每个权重的范数，以防止它们过大。这有助于防止模型过拟合。 实验2.2 基于 DLP 平台实现手写数字分类(30’’) **1.什么是量化？量化会带来什么好处/坏处？**  量化是在人工智能领域中一种优化神经网络模型的技术，其主要目的是减小模型的计算和内存需求，从而提高模型的运行效率。在模型量化中，通常会将原始的浮点数参数和权重转化为较低精度的整数或定点数，从而减小模型的内存占用和计算复杂度。  量化好处：  内存和计算效率：通过减小模型参数的位宽，模型量化可以大幅减小模型的内存占用和计算需求，使得模型可以在资源有限的设备上运行，例如移动设备、嵌入式系统等。  加速推理：量化模型通常更容易在硬件上并行执行，因此可以提高推理速度，特别是在边缘计算场景下，可以实现更低的延迟。  节省能源：模型量化能够减小模型的计算需求，这在移动设备和便携式设备上特别有益，因为它可以延长电池寿命，降低能耗。  量化坏处：  模型精度损失：量化会导致模型参数的信息损失，从而可能降低模型的准确性。因此，需要权衡模型的准确性和效率。  实现复杂性：模型量化需要精心设计和实施，以确保不会过度损害模型的性能。这可能需要更多的工程工作。  **2.pycnml、CNML库以及DLP之间的关系是什么？**  深度学习编程库 pycnml 通过调用 dlp 上 cnml库中的高性能算子实现了全连接层、 卷积层、池化层、ReLU 激活层、softmax 损失层等常用的网络层的基本功能， 并提供了常用络层的 python 接口。pycnml 提供的编程接口可以用于在dlp上加速神经网络算法， 具体接口说明如表2.3所示。pycnml 用 python 封装了一个 c++类 cnmlNet，该类的成员函数 定义了神经网络中层的创建、网络前向传播、参数加载等操作。  **3.使用DLP相对于CPU有哪些优势和劣势?**  DLP相对于CPU的优势：  专门设计： DLP是专门设计用于深度学习任务的硬件，通常包括专用的处理单元、内存和指令集，因此能够更高效地执行深度学习计算，而通用CPU则是设计用于广泛的计算任务。  高性能： DLP通常具有比CPU更高的计算性能，能够处理大规模深度神经网络，加速训练和推理任务。这对于深度学习应用中的大数据和复杂模型至关重要。  低功耗： DLP通常能够以较低的功耗执行深度学习任务，这对于便携式设备、嵌入式系统和边缘计算应用非常有利，可以延长电池寿命。  并行计算： DLP通常支持高度并行的计算，可以同时处理多个神经网络层的计算，从而提高计算效率。  降低延迟： 在实时应用和边缘计算场景中，DLP可以减小计算延迟，因为它专注于深度学习任务，无需处理其他冗余任务。  DLP相对于CPU的劣势：  通用性受限： DLP是专用的硬件，仅适用于深度学习任务。相比之下，CPU具有更广泛的应用范围，可以处理多种不同类型的计算任务。  高成本： 开发和制造DLP通常需要昂贵的研发和制造成本，这可能导致硬件价格较高。  灵活性不足： 由于DLP是专门设计的，因此难以适应新的深度学习模型或任务，而通用CPU可以通过软件更新实现更大的灵活性。  生态系统支持： CPU拥有庞大的生态系统和开发工具支持，而DLP的生态系统可能相对较小，这可能使开发和部署深度学习应用更加复杂。 | | |