

《模式识别与机器学习 A》实验报告

实验题目： 卷积神经网络

学号： _____

姓名： _____

1.实验目的

采用任意一种课程中介绍过的或者其它卷积神经网络模型（如 Lenet-5、AlexNet 等）用于解决某种媒体类型的模式识别问题。

2. 实验内容

1. 卷积神经网络可以基于现有框架如 TensorFlow、Pytorch 或者 Mindspore 等构建，也可以自行设计实现。
2. 数据集可以使用手写体数字图像标准数据集，也可以自行构建。预测问题可以包括分类或者回归等。实验工作还需要对激活函数的选择、dropout 等技巧的使用做实验分析。必要时上网查找有关参考文献。
3. 用不同数据量，不同超参数，比较实验效果，并给出截图和分析。

3. 实验环境

我在本实验中使用了 python 3.10, 利用了 numpy 库进行辅助矩阵运算，使用 matplotlib 库进行图像绘制。本次实验使用的额外数据是 MNIST 数据集。

4. 实验过程、结果及分析（包括代码截图、运行结果截图及必要的理论支撑等）

4.1 实验原理

卷积神经网络（Convolutional Neural Network, CNN）是受生物学上感受野机制的启发而提出的。目前的卷积神经网络一般是由卷积层、汇聚层和全连接层交叉堆叠而成的前馈神经网络，有三个结构上的特性：局部连接、权重共享以及汇聚。这些特性使得卷积神经网络具有一定程度上的平移、缩放和旋转不变性。和前馈神经网络相比，卷积神经网络的参数更少。卷积神经网络主要应用在图像和视频分析的任务上，其准确率一般也远远超出了其他的神经网络模型。近年来卷积神经网络也广泛地应用到自然语言处理、推荐系统等领域。

考虑到使用全连接前馈网络来处理图像时，会出现如下问题：

①模型参数过多，容易发生过拟合。在全连接前馈网络中，隐藏层的每个神经元都要跟该层所有输入的神经元相连接。随着隐藏层神经元数量的增多，参数的规模也会急剧增加，导致整个神经网络的训练效率非常低，也很容易发生过拟合。

②难以提取图像中的局部不变性特征。自然图像中的物体都具有局部不变性特征，比如尺度缩放、平移、旋转等操作不影响其语义信息。而全连接前馈网络很难提取这些局部不变性特征。

卷积神经网络有三个结构上的特性：局部连接、权重共享和汇聚。这些特性使得卷积神经网络具有一定程度上的平移、缩放和旋转不变性。和前馈神经网络相比，卷积神经网络的参数也更少。因此，通常会使用卷积神经网络来处理图像信息。

在机器学习和图像处理领域，卷积的主要功能是在一个图像（或特征图）上滑动一个卷积核，通过卷积操作得到一组新的特征。在计算卷积的过程中，需要进行卷积核的翻转，而这也带来一些不必要的操作和开销。因此，在具体实现上，一般会以数学中的互相关（Cross-Correlation）运算来代替卷积。

在神经网络中，卷积运算的主要作用是抽取特征，卷积核是否进行翻转并不会影响其特征抽取的能力。特别是当卷积核是可学习的参数时，卷积和互相关在能力上是等价的。因此，很多时候，为方便起见，会直接用互相关来代替卷积。

4.2 实验过程

```
# 加载数据集
mnist = tf.keras.datasets.mnist
(train_x, train_y), (test_x, test_y) = mnist.load_data()

# 对属性进行归一化，使它的取值在0-1之间，同时转换为tensor张量，类型为tf.float32
X_train = train_x.reshape(60000, 28, 28, 1)
X_test = test_x.reshape(10000, 28, 28, 1)
```

这部分代码在加载数据和将数据归一化。

```
X_train, X_test = tf.cast(X_train / 255.0, tf.float32), tf.cast(X_test / 255.0, tf.float32)
y_train, y_test = tf.cast(train_y, tf.int32), tf.cast(test_y, tf.int32)
```

这部分将数据分成了训练集和测试集。

```
# 建立模型
model = tf.keras.Sequential([
    # unit1
    tf.keras.layers.Conv2D(16, kernel_size=(3, 3), padding="same", activation=tf.nn.relu, input_shape=(28, 28, 1)),
    tf.keras.layers.MaxPool2D(pool_size=(2, 2)),

    # unit2
    tf.keras.layers.Conv2D(32, kernel_size=(3, 3), padding="same", activation=tf.nn.relu),
    tf.keras.layers.MaxPool2D(pool_size=(2, 2)),

    # unit3
    tf.keras.layers.Flatten(),

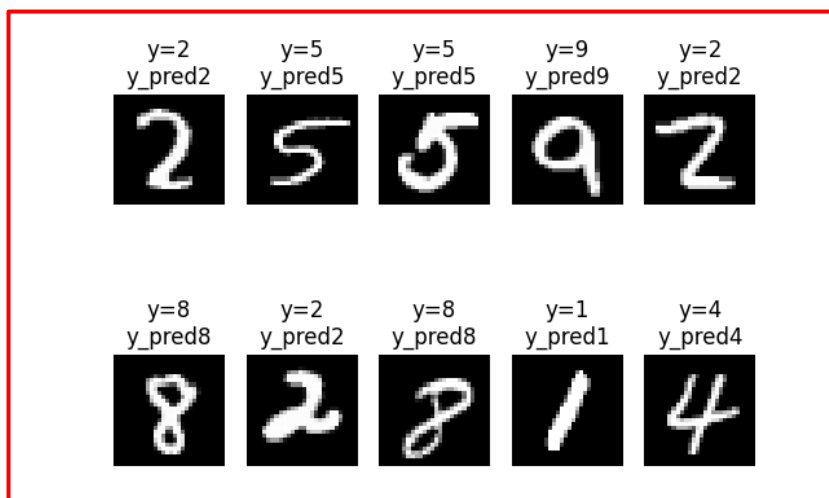
    # unit4
    tf.keras.layers.Dense(128, activation="relu"),
    tf.keras.layers.Dense(10, activation="softmax")
])
```

这部分在设置模型的参数。

.3 实验结果展示

```
Epoch 1/5
750/750 [=====] - 12s 15ms/step - loss: 0.2200 - sparse_categorical_accuracy: 0.9340 - val_loss: 0.0787 -
val_sparse_categorical_accuracy: 0.9764
Epoch 2/5
750/750 [=====] - 11s 15ms/step - loss: 0.0607 - sparse_categorical_accuracy: 0.9807 - val_loss: 0.0573 -
val_sparse_categorical_accuracy: 0.9813
Epoch 3/5
750/750 [=====] - 11s 15ms/step - loss: 0.0427 - sparse_categorical_accuracy: 0.9870 - val_loss: 0.0451 -
val_sparse_categorical_accuracy: 0.9855
Epoch 4/5
750/750 [=====] - 11s 15ms/step - loss: 0.0324 - sparse_categorical_accuracy: 0.9897 - val_loss: 0.0400 -
val_sparse_categorical_accuracy: 0.9878
Epoch 5/5
750/750 [=====] - 11s 15ms/step - loss: 0.0255 - sparse_categorical_accuracy: 0.9919 - val_loss: 0.0479 -
val_sparse_categorical_accuracy: 0.9853
```

上图是训练过程，可以发现，经过了 5 轮迭代，模型的损失不断下降，测试集上的精度不断提升。



上图是训练过程中，模型对于测试集上 10 个样本的预测。可以发现我们的模型效果不错。

5. 实验总体结论

在本次实验中，我尝试使用卷积神经网络完成 MNIST 数据集分类任务。相比之前的全连接神经网络，卷积神经网络参数更少，表征能力更强，从而取得了更好的效果。

6. 完整实验代码

```
import matplotlib.pyplot as plt
import numpy as np
import tensorflow as tf
import pandas as pd

# 加载数据集
mnist = tf.keras.datasets.mnist
(train_x, train_y), (test_x, test_y) = mnist.load_data()

# 对属性进行归一化，使它的取值在 0-1 之间，同时转换为 tensor 张量，类型为 tf.float32
X_train = train_x.reshape(60000, 28, 28, 1)
X_test = test_x.reshape(10000, 28, 28, 1)

X_train, X_test = tf.cast(X_train / 255.0, tf.float32), tf.cast(X_test / 255.0, tf.float32)
y_train, y_test = tf.cast(train_y, tf.int32), tf.cast(test_y, tf.int32)

# 建立模型
model = tf.keras.Sequential([
    # unit1
    tf.keras.layers.Conv2D(16, kernel_size=(3, 3), padding="same",
        activation=tf.nn.relu, input_shape=(28, 28, 1)),
    tf.keras.layers.MaxPool2D(pool_size=(2, 2)),

    # unit2
    tf.keras.layers.Conv2D(32, kernel_size=(3, 3), padding="same",
        activation=tf.nn.relu),
    tf.keras.layers.MaxPool2D(pool_size=(2, 2)),

    # unit3
    tf.keras.layers.Flatten(),

    # unit4
    tf.keras.layers.Dense(128, activation="relu"),
    tf.keras.layers.Dense(10, activation="softmax")
])

# 配置训练方法
model.compile(optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['sparse_categorical_accuracy'])
```

```
# 训练模型
history = model.fit(X_train, y_train, batch_size=64, epochs=5,
validation_split=0.2)

# 评估模型
model.evaluate(X_test, y_test, verbose=2)
pd.DataFrame(history.history).to_csv("training_log.csv", index=False)
graph = pd.read_csv('training_log.csv')

# 使用模型
for i in range(10):
    num = np.random.randint(1, 10000)

    plt.subplot(2, 5, i + 1)
    plt.axis("off")
    plt.imshow(test_x[num], cmap="gray")
    demo = tf.reshape(X_test[num], (1, 28, 28, 1))
    y_pred = np.argmax(model.predict(demo))
    plt.title("y=" + str(test_y[num]) + "\ny_pred" + str(y_pred))

plt.show()
```

7. 参考文献

[1] 《深度学习基础》高等教育出版社 刘远超