DATASET

一、内容

Fashion MNIST数据集是一个包含60,000个训练样本和10,000个测试样本的28x28图像集合，包括10种不同的服装物品，如鞋子、靴子、衬衫、包包等。

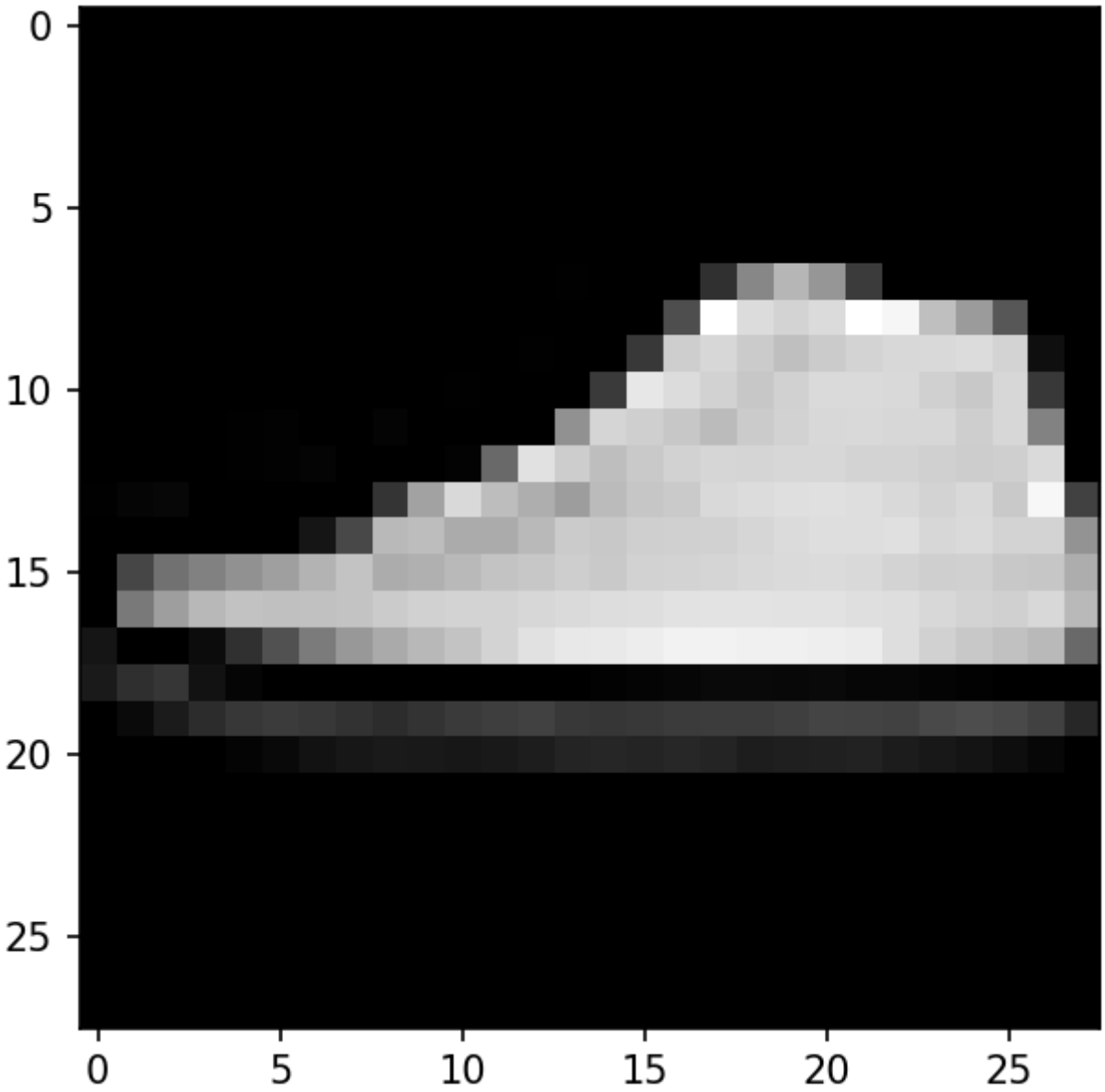
二、代码

下载数据

```
# 数据集下载地址
URL = 'https://nnfs.io/datasets/fashion_mnist_images.zip'
# 存放地址
FILE = 'fashion_mnist_images.zip'
# 解压地址
FOLDER = 'fashion_mnist_images'
# 将网上数据存在当前文件夹的FILE中
# 如果本地没有文件，就下载
if not os.path.isfile(FILE):
      print(f'下载 {URL} 并存在 {FILE}...')
      urllib.request.urlretrieve(URL, FILE)

      print('解压文件')
      with ZipFile(FILE) as zip_images:
            zip_images.extractall(FOLDER)

# image_data = cv2.imread('fashion_mnist_images/train/7/0002.png',cv2.IMREAD_UNCHANGED)
# np.set_printoptions(linewidth=200)
# print(image_data)
#
# plt.imshow(image_data, cmap='gray')
# plt.show()
```



鞋

加载数据

```
# 加载MNIST dataset
def load_mnist_dataset(dataset, path):

    # 输入数据集的名称和地址
    # 得到类文件
    labels = os.listdir(os.path.join(path, dataset))

    X = []
    y = []

    # 打开每个类文件夹
    for label in labels:
```

```
            # 循环其中每个文件
            for file in os.listdir(os.path.join(path, dataset, label)):
                # 读文件
                image = cv2.imread(os.path.join(path, dataset, label, file),
cv2.IMREAD_UNCHANGED)

                # 存到list中
                X.append(image)
                y.append(label)

    return np.array(X), np.array(y).astype('uint8')

# 创建数据集，内部调用load_mnist_dataset
def create_data_mnist(path):

    # 加载训练和测试集
    X, y = load_mnist_dataset('train', path)
    X_test, y_test = load_mnist_dataset('test', path)

    return X, y, X_test, y_test
```

预处理数据

```
def data_preprocess():
    X, y, X_test, y_test =
create_data_mnist('D:/python_workplace/pycharm/workplace/NNFS_py38_NNFS/fashion_mnist_imag
es')

    # 归一化，让数据分布在[-1.1],利于训练
    X = (X.astype(np.float32) - 127.5) / 127.5
    X_test = (X_test.astype(np.float32) - 127.5) / 127.5

    # 因为网络模一型是全连接网络，要将二维图片展成一维
    X = X.reshape(X.shape[0],-1)
    X_test = X_test.reshape(X_test.shape[0], -1)

    # 打乱数据顺序
    key = np.array(range(X.shape[0]))
    np.random.shuffle(key)
    X = X[key]
    y = y[key]

    return X, y, X_test, y_test
```

> 预处理数据包括：归一化、二维图片展成一维、打乱数据顺序。

实例

```
model = Model()
model.add(Layer_Dense(X.shape[1], 64, weight_L2=5e-4,bias_L2=5e-4))#,weight_L2=5e-
4,bias_L2=5e-4
model.add(Activation_ReLu())
model.add(Layer_Dense(64, 64))
model.add(Activation_ReLu())
model.add(Layer_Dense(64, 10))
model.add(Activation_Softmax())
model.set(loss=Loss_CategoricalCrossentropy(),
                optimizer=Optimizer_Adam(decay=5e-7),
                accuracy=Accuracy_Classification())


model.finalize()

model.train(X, y, batch_size=100, validation_data=(X_test, y_test), epochs=5,
print_every=10)
model.evaluate(X_test, y_test, batch_size=10)
# 反回各类别的概率
confidence = model.predict(X_test[95:105])
prediction = model.output_layer.prediction(confidence)

print('预测分类：',prediction)
print('ground truth：',y_test[95:105])
```

```
training 5, acc: 0.882, loss: 0.404 (data_loss: 0.325, reg_loss: 0.079), lr: 0.000998502745133672
validation, acc: 0.866, loss: 0.372
validation, acc: 0.866, loss: 0.372
预测分类： [[6]
 [4]
 [4]
 [5]
 [1]
 [1]
 [0]
 [2]
 [2]
 [8]]
ground truth： [6 4 4 5 1 1 0 2 2 8]
```

> 表现非常好。