

# 实验报告：构建三层神经网络分类器进行Fashion-MNIST分类

王子天 23210980082

[github repo链接](#), [模型-百度网盘链接](#)

## 1. 引言

本实验不依赖现有的深度学习框架如PyTorch或TensorFlow等，仅使用NumPy库，自主构建了一个三层神经网络分类器。实验使用Fashion-MNIST数据集，最终提交的代码中包含了**模型**、**训练**、**测试**和**参数查找**四个部分，进行了模块化设计。其中，**模型部分**允许自定义隐藏层大小、激活函数类型，支持通过反向传播计算给定损失的梯度；**训练部分**实现了SGD优化器、学习率权重衰减、交叉熵损失和L2正则化，并能根据验证集指标自动保存最优的模型权重；**参数查找**环节可以使用网格搜索法调节学习率、学习率权重衰减强度、各隐藏层大小、L2正则化强度（weight\_decay）等超参数；**测试部分**支持导入训练好的模型，输出在测试集上的分类准确率等指标。

## 2. 数据集

Fashion-MNIST数据集是一个包含10个不同时尚商品类别的图像集合。每个图像都是28x28像素的灰度图，涵盖了从T恤、裤子到鞋子、包等多种时尚商品。与MNIST手写数字数据集相比，Fashion-MNIST提供了更复杂的图像分类任务，有助于评估神经网络在不同场景下的性能。

## 3. 模型构建

- 网络架构**：本实验搭建的神经网络架构包括1个输入层、2个隐藏层和1个输出层。输入层接收28x28像素的图像数据，并将其展平为一维向量，作为网络的输入。隐藏层采用全连接结构，通过权重矩阵和偏置项与前一层的输出进行线性组合，并应用激活函数引入非线性。输出层使用softmax函数，将隐藏层的输出转换为概率分布，以表示不同类别的预测概率。
- 激活函数**：在隐藏层中，我选择了Sigmoid和ReLU作为激活函数。ReLU函数能够改善Sigmoid中的梯度消失问题，并加快训练速度。它通过将输入值映射到非负区域，引入非线性特性，使网络能够学习更复杂的特征表示。在输出层中，我使用了softmax函数将隐藏层的输出转换为概率分布，以便进行多分类任务。
- 参数初始化**：为了打破对称性并提高模型的泛化能力，我使用了随机初始化方法对网络参数（权重和偏置）进行初始化。权重矩阵使用小的随机数进行初始化，而偏置项需要设置为零。

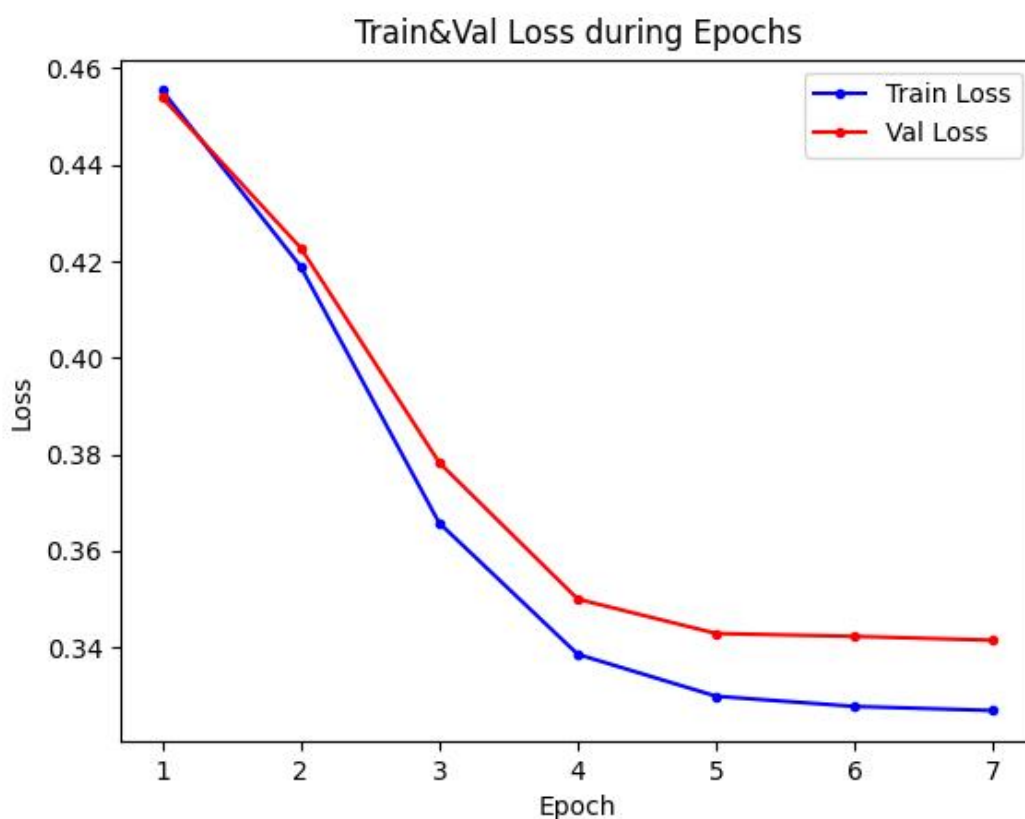
## 4. 训练流程

- 前向传播**：在前向传播阶段，首先将输入数据通过网络层进行前向计算。输入数据经过输入层后，进入第一个隐藏层。在隐藏层中，根据权重矩阵和偏置项计算每个神经元的输出值，并应用ReLU激活函数进行非线性变换。然后，将隐藏层的输出作为下一个隐藏层的输入，重复上述过程直至到达输出层。在输出层中，使用了softmax函数将隐藏层的输出转换为概率分布，以表示不同类别的预测概率。
- 损失函数**：为了衡量模型预测与真实标签之间的差异，本实验采用交叉熵损失函数作为训练目标。交叉熵损失函数能够量化预测概率分布与真实概率分布之间的差距，并通过最小化该差距来优化网络参数。此外，在损失函数构建中，还对参数进行了L2范数惩罚，以控制模型的复杂度，提升样本外泛化性能。通过计算损失函数值，评估模型在当前配置下的性能，并据此调整网络参数以改进模型。

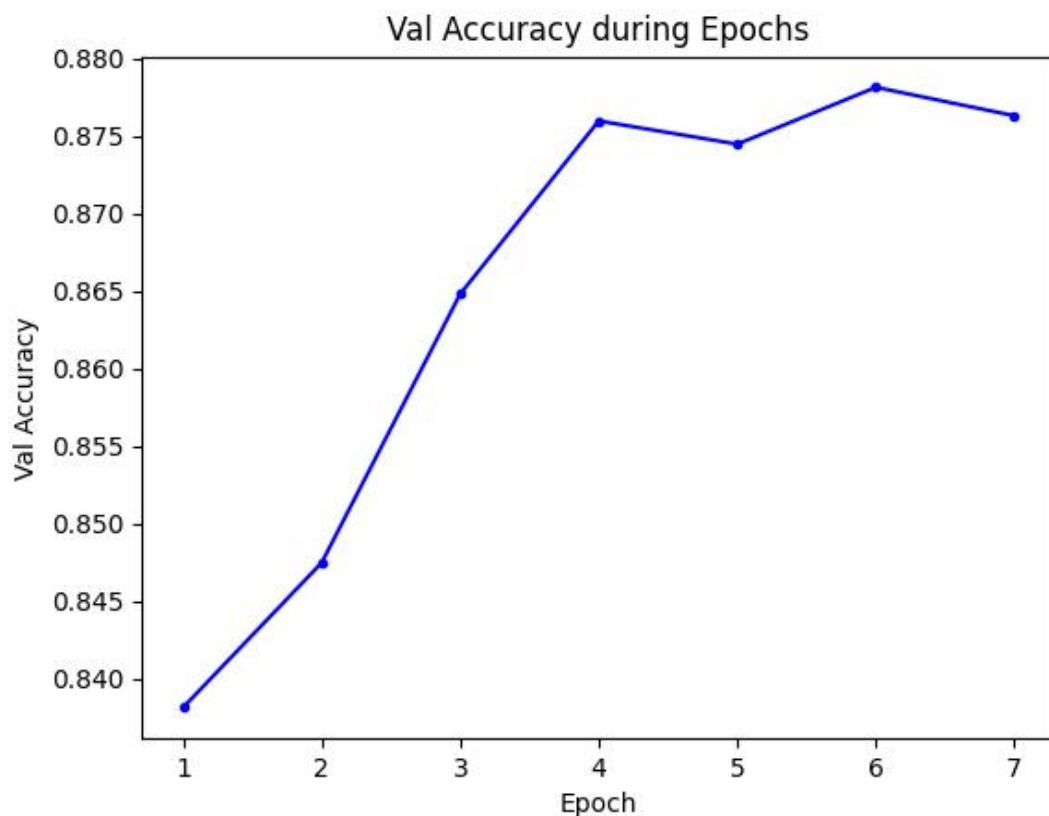
3. **反向传播**: 在反向传播过程中, 我使用了矩阵求导的链式法则计算损失函数关于网络参数的梯度。首先, 计算输出层对损失函数的梯度 (即误差项), 然后逐层向前传递梯度, 计算每个隐藏层对损失的贡献。通过反向传播算法, 可以将损失函数关于输出层的梯度逐层传递回输入层, 并据此更新网络参数以最小化损失函数。
4. **优化器**: 本实验选择随机梯度下降 (SGD) 作为优化算法来更新网络参数。在每个训练批次中, 随机选择一部分数据进行迭代训练, 并计算该批次数据的损失函数值及其关于网络参数的梯度。然后, 使用SGD算法更新网络参数, 以减小损失函数值。为了加速训练过程和防止过拟合, 实验引入了学习率衰减策略, 使学习率随训练轮次的增加而逐渐减小。此外, 还采用了L2正则化项对网络参数进行约束, 以减小过拟合的风险。
5. **验证评估**: 在每个训练周期结束后, 使用验证集评估模型的性能。通过计算验证集上的损失值和准确率等指标, 监控模型的训练进度并检查是否存在过拟合现象。根据验证集性能的变化情况, 进行超参数调整以优化模型性能。最后, 我保存了最佳模型权重, 以便在测试集上进行评估。

#### 6. 训练过程可视化

在训练集和验证集上的loss曲线



验证集上的accuracy曲线



## 5. 超参数调优

为了找到最佳的超参数配置，实验进行了超参数调优。首先，确定了需要调优的超参数范围，如学习率、隐藏层大小、正则化强度等。然后，采用网格搜索的方法，在超参数空间中选择不同的配置组合进行训练。在每个配置下，记录模型在验证集上的性能表现（如损失值和准确率），并比较不同配置下的结果以找到最优配置。

所得最优模型的超参数配置为：

lr	hidden1_dim	hidden2_dim	weight_decay
0.1	256	256	1e-3

不同超参数组合下，模型在测试集和训练集上的具体表现如下：

lr	hidden1_dim	hidden2_dim	weight_decay	train_loss	val_loss	train_acc	val_acc
0.1	64	64	1	0.365	0.399	86.98%	85.60%
0.1	64	64	0.1	0.348	0.389	87.54%	85.60%
0.1	64	64	0.001	0.358	0.400	87.25%	85.43%
0.1	64	128	1	0.365	0.403	87.09%	85.28%
0.1	64	128	0.1	0.338	0.378	87.77%	85.68%
0.1	64	128	0.001	0.337	0.379	87.95%	85.98%
0.1	128	64	1	0.359	0.395	87.23%	85.52%
0.1	128	64	0.1	0.336	0.379	87.99%	85.93%
0.1	128	64	0.001	0.338	0.384	87.98%	85.93%
0.1	128	128	1	0.355	0.390	87.44%	85.35%
0.1	128	128	0.1	0.333	0.374	88.00%	85.85%
0.1	128	128	0.001	0.326	0.372	88.37%	86.45%
0.1	256	64	1	0.355	0.391	87.49%	85.48%
0.1	256	64	0.1	0.330	0.373	88.04%	85.97%
0.1	256	64	0.001	0.330	0.373	88.21%	86.43%
0.1	256	128	1	0.347	0.386	87.79%	85.87%
0.1	256	128	0.1	0.323	0.369	88.50%	85.97%
0.1	256	128	0.001	0.322	0.371	88.43%	86.13%
0.1	256	256	1	0.346	0.383	87.76%	85.68%
0.1	256	256	0.1	0.316	0.361	88.74%	86.27%
0.1	256	256	0.001	0.317	0.366	88.63%	86.47%
0.01	64	64	1	0.519	0.545	81.77%	80.72%
0.01	64	64	0.1	0.529	0.556	80.95%	80.20%
0.01	64	64	0.001	0.520	0.549	81.63%	80.33%
0.01	64	128	1	0.500	0.525	82.35%	81.25%
0.01	64	128	0.1	0.503	0.535	82.11%	80.67%
0.01	64	128	0.001	0.507	0.537	81.80%	80.37%
0.01	128	64	1	0.510	0.533	82.20%	81.65%
0.01	128	64	0.1	0.505	0.528	82.30%	81.15%
0.01	128	64	0.001	0.498	0.526	82.49%	81.33%
0.01	128	128	1	0.493	0.518	82.58%	81.43%
0.01	128	128	0.1	0.492	0.516	82.72%	81.50%
0.01	128	128	0.001	0.493	0.521	82.67%	81.70%
0.01	256	64	1	0.492	0.519	82.65%	81.37%
0.01	256	64	0.1	0.489	0.518	82.74%	81.78%
0.01	256	64	0.001	0.495	0.515	82.54%	81.17%
0.01	256	128	1	0.491	0.519	82.77%	81.28%
0.01	256	128	0.1	0.488	0.517	83.04%	81.43%
0.01	256	128	0.001	0.484	0.514	82.98%	81.57%

## 6. 测试与评估

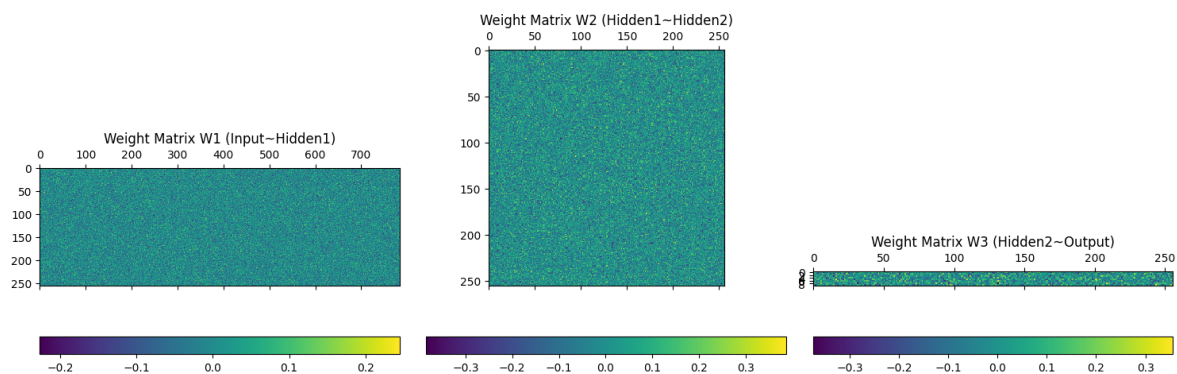
在测试阶段，我们使用独立的测试集对训练好的模型进行评估。首先，我们将测试集中的图像数据输入到模型中，并通过前向传播得到每个类别的预测概率。然后，我们根据预测概率确定每个图像的预测类别，并与真实标签进行比较，计算分类准确率等评估指标。

下图是最佳模型在测试数据集上，各类别的分类准确率数据：

category	0	1	2	3	4	5	6	7	8	9
precision	81.19%	98.46%	76.10%	84.66%	77.48%	94.79%	68.40%	91.17%	95.17%	93.56%

## 7. 网络参数可视化

为了深入理解模型的内部工作机制，本实验对训练好的网络参数进行了可视化分析。首先，绘制了权重矩阵的热力图，展示了不同神经元之间的连接强度和权重分布模式，以帮助了解模型在学习过程中的特征提取和表示能力。



此外，实验还采用了主成分分析（PCA）等方法对隐藏层的输出进行降维处理，并在二维平面上绘制了降维后的数据点，以展示隐藏层学习到的特征表示，并帮助理解模型如何将输入数据映射到高维空间中的特征表示。

