

# Homework 2

李雨硕2024316200

October 2024

## 1 BP必做

- Neuron transfer function select sigmoid.
- Weight update from the bias neuron to the hidden layer neurons:

$$\Delta w_{0j} = \eta \cdot (d_1 - o_1) \cdot o_1(1 - o_1) \cdot w_{j1} \cdot y_j(1 - y_j)$$

- Weight update from the input neurons to the hidden layer neurons:

$$\Delta w_{ij} = \eta \cdot (d_1 - o_1) \cdot o_1(1 - o_1) \cdot w_{j1} \cdot y_j(1 - y_j) \cdot x_i$$

- Weight update from the hidden layer neurons to the output layer neuron:

$$\Delta w_{j1} = \eta \cdot (d_1 - o_1) \cdot o_1(1 - o_1) \cdot y_j$$

- Code: BP\_required.py

- Result:

Input: [1 1] - Predicted Output: [[0.0381571]]

Input: [1 -1] - Predicted Output: [[0.96580801]]

Input: [-1 1] - Predicted Output: [[0.96581007]]

Input: [-1 -1] - Predicted Output: [[0.03960205]]

## 2 BP选做

- Neuron transfer function select sigmoid.

### 1. Bias to Output Layer:

$$\Delta w_{01} = \eta \cdot (d_1 - o_1) \cdot o_1 \cdot (1 - o_1)$$

### 2. Bias to Hidden Layer:

$$\Delta w_{02} = \eta \cdot (\delta_1 \cdot w_{21}) \cdot h_2 \cdot (1 - h_2)$$

### 3. Input Layer to Hidden Layer:

$$\Delta w_{31} = \eta \cdot (\delta_1 \cdot w_{21}) \cdot h_2 \cdot (1 - h_2) \cdot x_1$$

$$\Delta w_{41} = \eta \cdot (\delta_1 \cdot w_{21}) \cdot h_2 \cdot (1 - h_2) \cdot x_2$$

### 4. Hidden Layer to Output Layer:

$$\Delta w_{21} = \eta \cdot (d_1 - o_1) \cdot o_1 \cdot (1 - o_1) \cdot h_2$$

- **Code:** BP\_choose.py

- **Result**

Input: [1 1] - Predicted Output: [[0.00207906] [0.00207906]]

Input: [1 -1] - Predicted Output: [[0.00162791] [0.00162791]]

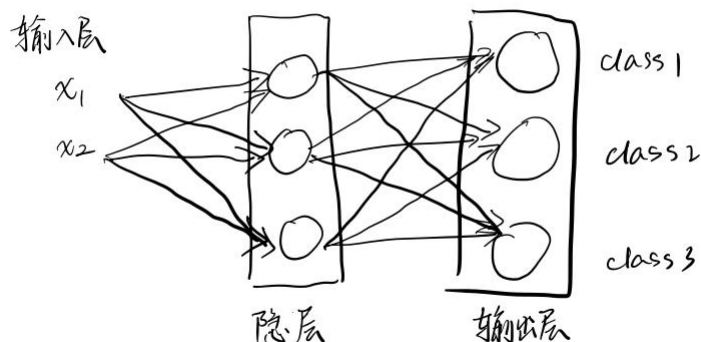
Input: [-1 1] - Predicted Output: [[0.93984773] [0.93984773]]

Input: [-1 -1] - Predicted Output: [[0.00207906] [0.00207906]]

## 二、分类问题（三类）

### 1. 单隐层 BP 网络

#### ➤ 结构绘制



#### ➤ 算法流程描述

1. 前向传播：计算输入层到隐层的加权和，应用激活函数得到隐层输出；将隐层输出传递到输出层，计算加权和并通过激活函数得到最终输出。
2. 误差计算：使用目标值（类别标签）与网络输出的差值，计算误差。
3. 反向传播：计算输出层误差，调整隐层到输出层的权重；计算隐层的误差，调整输入层到隐层的权重。
4. 权重更新：使用梯度下降法，根据误差的导数更新每一层的权重。
5. 循环训练：对所有训练样本进行多个迭代，重复执行上述步骤，直到网络收敛或达到最大迭代次数。

#### ➤ 不同隐层节点个数的讨论

##### 隐层节点较少：

- 优点：网络较简单，计算开销低，训练速度快。
- 缺点：若隐层节点太少，可能导致网络不足以学习数据中的复杂模式，出现欠拟合现象，无法正确分类复杂样本。

##### 隐层节点适中：

- 优点：网络能够较好地捕捉数据的非线性关系，分类精度较高，并且能在复杂度和准确度之间取得平衡。
- 缺点：需要通过实验确定合适的节点数，比较难把控。

##### 隐层节点过多：

- 优点：理论上，更多的隐层节点可以提高网络的表达能力，使网络能够学习到非常复杂的模式。
- 缺点：节点过多可能导致网络过拟合，即网络过度记忆训练数据而失去对新数据的泛化能力。此外，网络训练时间和计算资源需求也会显著增加。

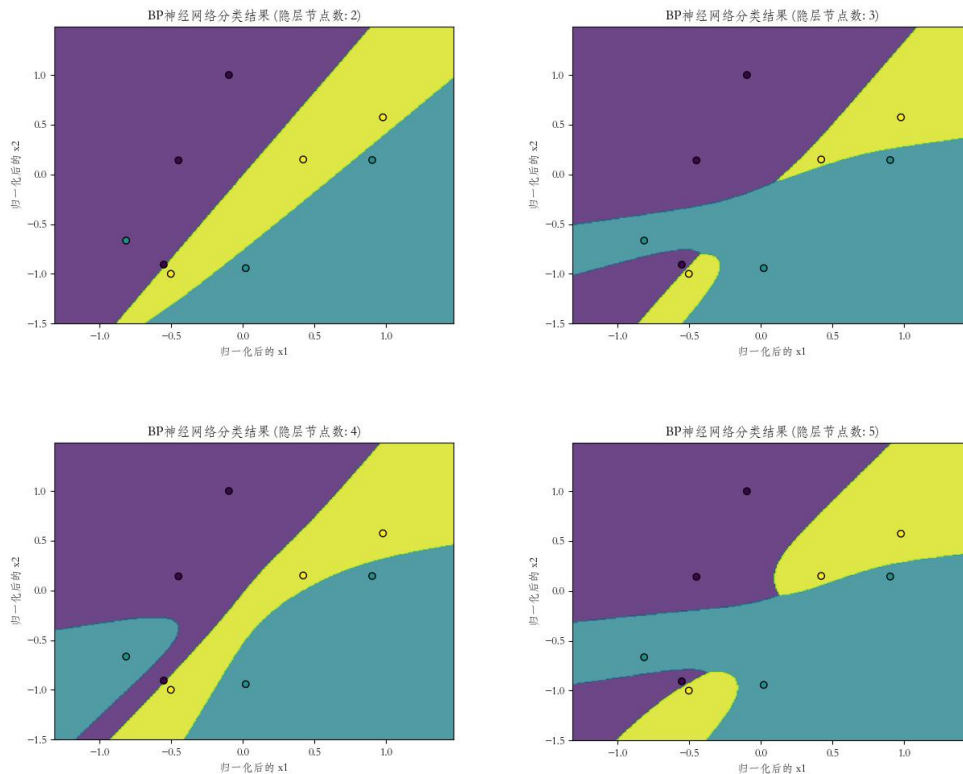
#### ➤ 综上所述，最少隐层节点数为 3。

#### ➤ Code: Single\_hidden\_layer\_BP.py

#### ➤ 分析讨论：

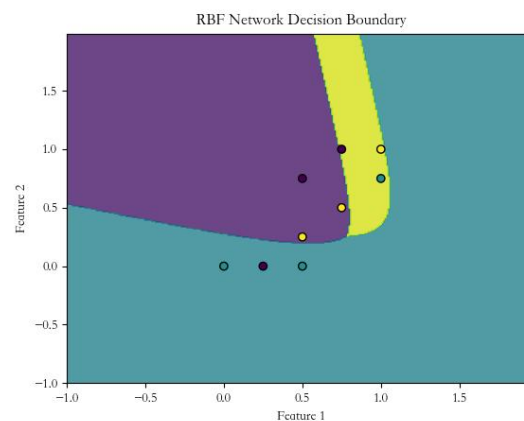
- 如图所示
- 隐层节点数 2：分类边界较为简单，网络可以进行基本的分类，但灵活性较差，决策区域比较大且单一。

- 隐层节点数 3: 分类边界变得更加复杂, 能够更精细地区分不同的类别, 决策区域更加符合数据的分布。
- 隐层节点数 4: 分类边界进一步复杂化, 但仍保持一定的平滑性, 说明网络的分类准确性有所提升, 同时过拟合的风险还不小。
- 隐层节点数 5: 分类边界更加复杂, 网络能够很好地拟合数据, 但过拟合风险也增加, 决策边界可能过度贴合训练数据。



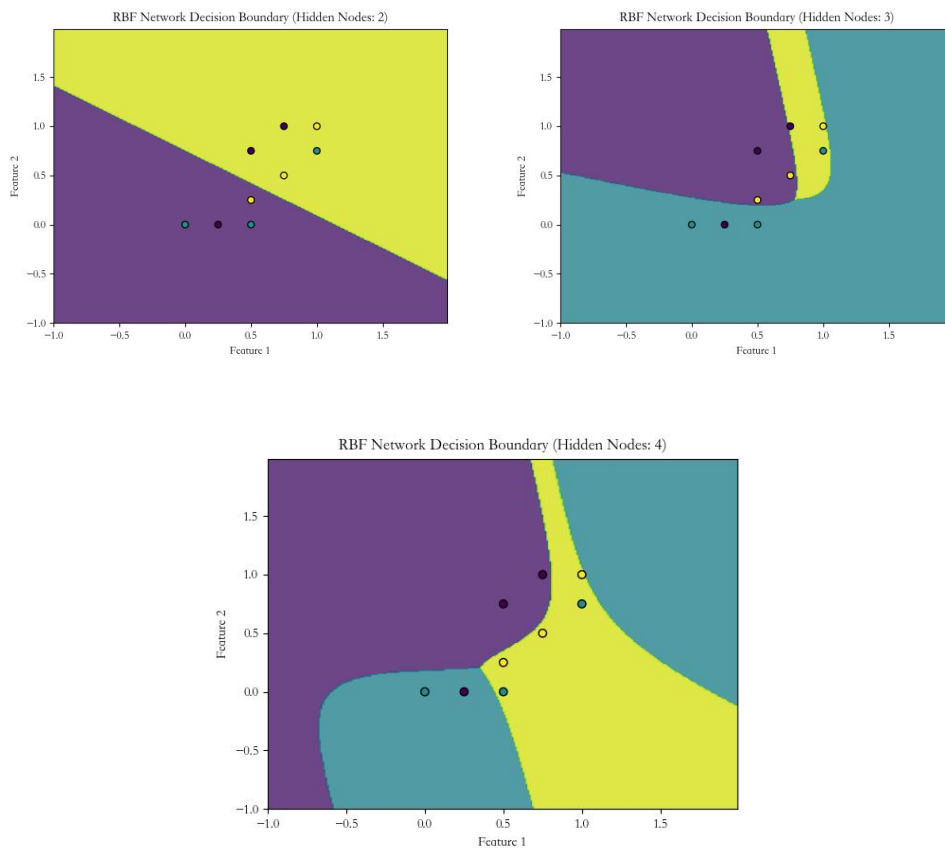
## 2. 正规化 RBF

- Code: NormalizedRBF.py
- 参数:
  - 隐层节点数: 设定为 3。
  - 中心: 通过 KMeans 聚类从输入数据中选取的中心点
- 仿真结果:



### 3. 广义 RBF

➤ Code: GeneralizedRBF.py



## 三、 MNIST 分类

➤ Code: MNIST.py

➤ 网络结构设计:

使用两层隐藏层的 BP 神经网络，输入层有 784 个节点（对应 28x28 的像素），第一隐藏层有 128 个节点，第二隐藏层有 64 个节点，输出层有 10 个节点，对应数字 0-9。网络使用 ReLU 激活函数，输出层使用 Softmax 激活函数。

➤ 训练方法:

使用 MNIST 数据集进行训练，训练样本为 50,000 个，测试样本为 10,000 个。损失函数使用交叉熵损失函数，优化器选择 Adam 优化器，学习率为 0.001。模型训练共进行 10 个 epoch，每个 batch 大小为 128。

➤ 识别结果:

在测试集上评估模型性能，最终测试集上的准确率为约 96.53%。

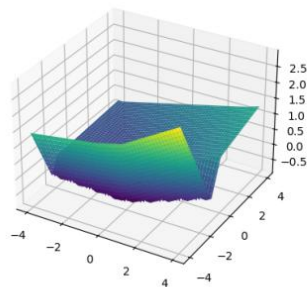
```
Epoch [1/10], Loss: 1.7034
Epoch [2/10], Loss: 1.5554
Epoch [3/10], Loss: 1.5368
Epoch [4/10], Loss: 1.5257
Epoch [5/10], Loss: 1.5170
Epoch [6/10], Loss: 1.5103
Epoch [7/10], Loss: 1.5055
Epoch [8/10], Loss: 1.5029
Epoch [9/10], Loss: 1.4987
Epoch [10/10], Loss: 1.4973
测试集上的准确率: 96.53%
```

## 三、 peaks 逼近

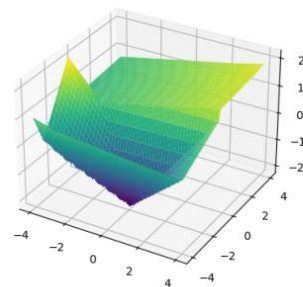
### 1. BP 逼近

- Code: peaks\_BP.py
- 在样本数量较少（150 个随机样本）的情况下，模型可能无法很好地捕捉到 **peaks** 函数的复杂性，特别是在变化剧烈的区域（如极值和边界附近）。此时的逼近效果相对较差，尤其在函数波动较大的位置。同时，完全随机采样的情况下，样本可能集中在某些区域而导致函数逼近效果不稳定。
- 当样本数量增加到 200 个且包含一部分特征区域的样本时，模型更容易捕捉到 **peaks** 函数的关键特征。此时，模型能够更好地逼近函数的极值区域，使整体效果更为精准。

Approximation with 150 samples (random sampling)

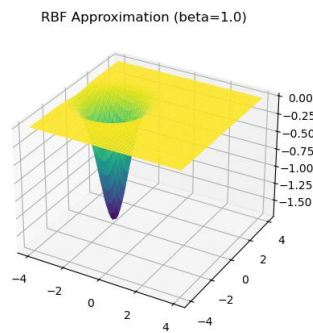
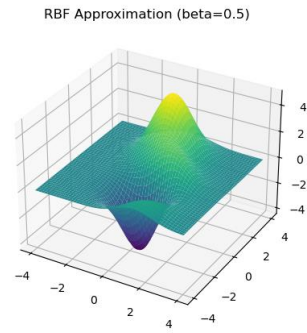
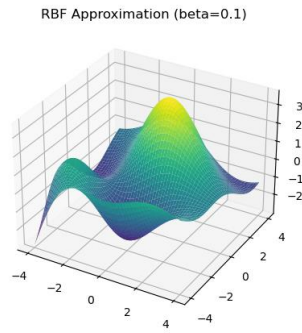


Approximation with 200 samples (feature\_based sampling)



### 2. RBF 逼近

- 较小的 **beta** 值: 当 **beta** 较小时，RBF 网络的每个神经元影响范围较小，网络更关注局部细节变化。因此，网络能在局部区域更精准地逼近复杂的变化，但容易产生过拟合，对噪声更为敏感，整体逼近效果不平滑。
- 较大的 **beta** 值: 当 **beta** 较大时，神经元的影响范围增大，网络逼近的曲面更为平滑。这使模型对整体趋势的拟合更好，对噪声不敏感，但在变化剧烈区域可能难以准确捕捉局部特征。



## 四、 ASCII 压缩

➤ Code: ASCII.py

➤ 讨论

隐层节点数少（高压缩）：高压缩率，重构误差大，适合提取主要特征的任务，但细节损失严重。

隐层节点数适中（平衡）：实现压缩与重构精度的平衡，能捕捉主要特征和细节，重构误差较小。

隐层节点数多（低压缩，高精度）：低压缩率，重构误差极小，接近恒等映射，压缩意义不大且泛化能力差。

➤ 恢复后数据图像

