



南开大学
Nankai University

《人工智能物联网导论》实验报告

Attention U-Net 图像分割

组员 1: 梁涵杰 2120230780

组员 3: 李时玉 2120230778

组员 2: 杨 毅 2120230799

2023~2024 第二学期

目 录

1	模型背景概述	2
1.1	U-Net	2
1.2	注意力机制	2
1.3	Attention U-Net	3
2	数据集介绍	3
2.1	Synapse 数据集	3
2.2	数据集目录结构	3
2.3	DataSet 类	4
3	参数设置情况	5
4	模型实验结果	5
4.1	评价指标	5
4.2	实验结果	6
5	相关分析和结论	7
5.1	结果分析	7
5.2	结论	7
6	相关问题和解决方法	8
6.1	问题: 测试数据集加载异常	8
6.2	解决方法	8
7	小组成员分工描述和占比	9

1 模型背景概述

1.1 U-Net

U-Net 是一种基于卷积神经网络（CNN）的模型，最早由 Olaf Ronneberger 等人在 2015 年提出，专为生物学医学图像分割设计。U-Net 的结构呈 U 形，包含一个编码器（encoder）和一个解码器（decoder）。U-Net 的关键特性在于其跳跃连接（skip connections），即将编码器每一层的特征图直接连接到相应解码器层。这种结构能够有效保留图像的细节信息，提高分割精度。

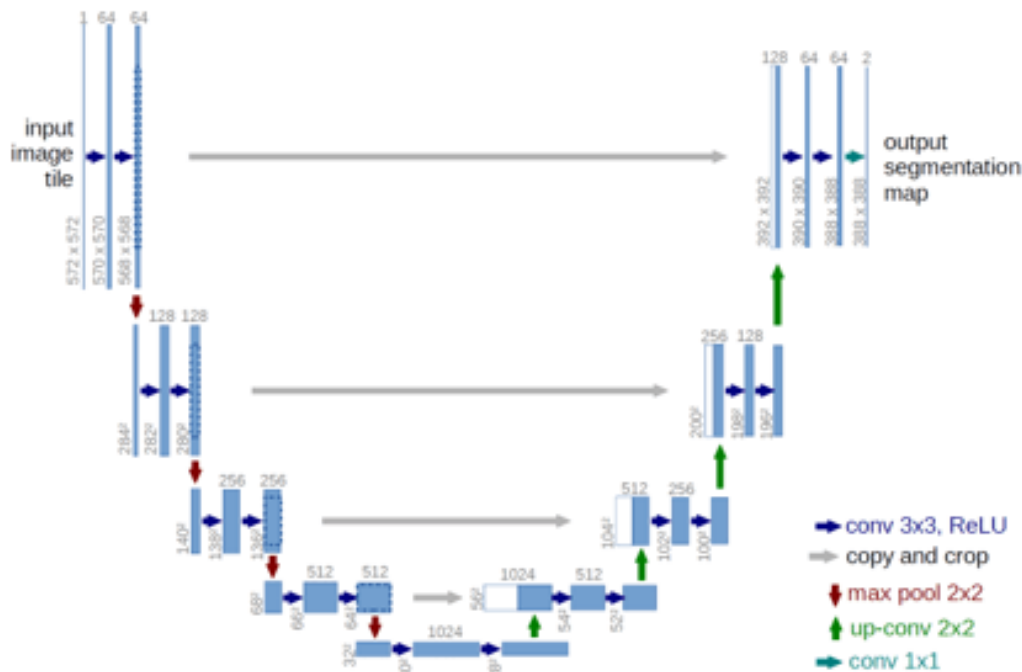


图 1: U-Net

1.2 注意力机制

注意力机制起源于自然语言处理领域，旨在模拟人类视觉选择性注意的过程。通过为输入数据的不同部分分配不同的权重，注意力机制能够突出重要特征，抑制不相关信息。其主要类型包括：

- 自注意力（Self-Attention）：关注输入序列中各位置间的相关性。
- 注意力加权（Attention Weighting）：为输入数据的各部分分配不同的权重。

1.3 Attention U-Net

Attention U-Net 是一种用于医学图像分割的深度学习模型，其主要在传统 U-Net 的基础上进行了改进，引入了注意力机制（Attention Mechanism）以增强模型的性能。

Attention U-Net 主要应用于医学图像分割任务，如肿瘤检测、器官分割等。由于其在保留细节和强调重要特征方面的优势，Attention U-Net 在许多医学图像分割任务中表现出色，显著提升了分割精度和鲁棒性。

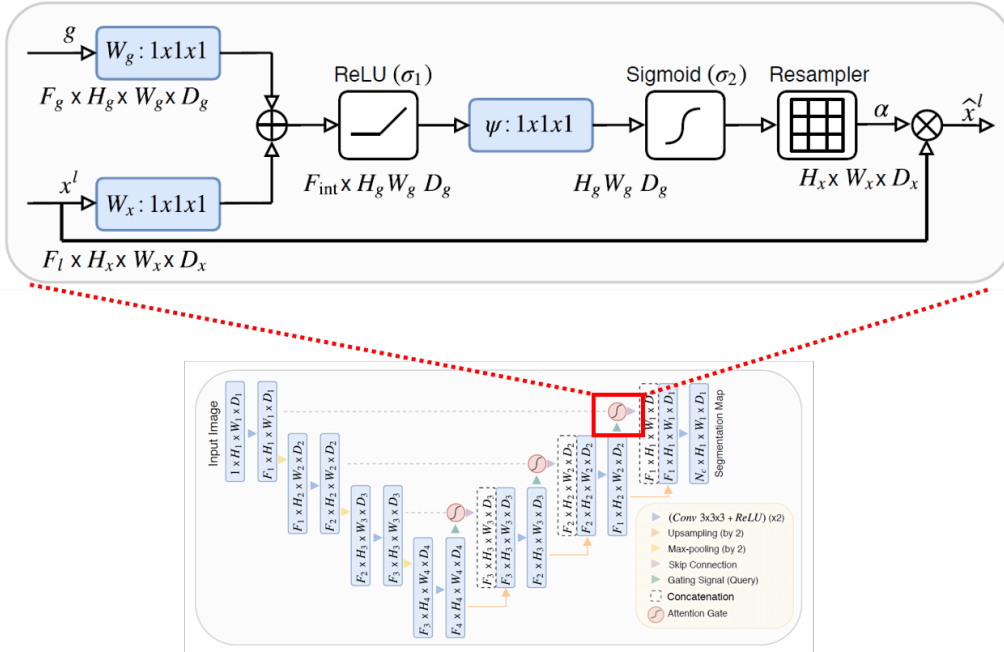


图 2: Attention U-Net

2 数据集介绍

本实验使用提供的 Synapse 数据集，训练集大小为 2000，测试集大小为 1500。

2.1 Synapse 数据集

Synapse 数据集由腹部 CT 扫描图像组成，包含多个器官的手动标注。这些器官包括肝脏、脾脏、胰腺、肾脏和其他重要器官。Synapse 数据集是一个重要的医学图像分割数据集，广泛应用于多器官分割任务中。其高质量的图像和详细的器官标注为开发和评估分割模型提供了宝贵的资源，对于推动医学影像分析和相关技术的发展具有重要作用。

2.2 数据集目录结构

```
1 Synapse/  
2     lists/  
3         lists_Synapse/  
4             all.lst  
5             test_vol.txt  
6             train.txt  
7     test_vol_h5/  
8         case0001.npy.h5  
9         case0002.npy.h5  
10        ...  
11     train_npz/  
12         case0005_slice000.npz  
13         case0005_slice001.npz  
14        ...
```

2.3 DataSet 类

```
1 class SynapseDataset(Dataset):  
2     def __init__(self, data_list, data_dir, transform=None):  
3         self.data_list = data_list  
4         self.data_dir = data_dir  
5         self.transform = transform  
6  
7     def __len__(self):  
8         return len(self.data_list)  
9  
10    def __getitem__(self, idx):  
11        file_name = self.data_list[idx]+' '  
12        file_path = os.path.join(self.data_dir, file_name)  
13        # print(file_path)  
14        if file_path.endswith('.npz'):  
15            data = np.load(file_path)  
16            image = data['image']  
17            label = data['label']  
18  
19        elif file_path.endswith('.h5'):  
20            with h5py.File(file_path, 'r') as f:  
21                images = np.array(f['image'])  
22                labels = np.array(f['label'])  
23                for img in images:
```

```
24         image = img
25
26         for lab in labels:
27             label = lab
28
29     else:
30         raise ValueError("Unsupported file format")
31
32     if self.transform:
33         image, label = self.transform(image, label)
34
35     image = torch.tensor(image, dtype=torch.float32).unsqueeze(0)
36     label = torch.tensor(label, dtype=torch.float32).unsqueeze(0)
37
38     return image, label
```

3 参数设置情况

本网络模型基于 python3.8 和 pytorch1.9 实现。输入图像尺寸为 512×512 像素。在一张具有 8 GB 内存的 Nvidia RTX 2070 GPU 上训练 U-net、Att-Unet、FCN 模型。训练参数设置为：epoch 为 100，batch size 为 3，初始学习率为 0.15，使用 SGD（stochastic gradient descent）优化器（动量为 0.9，权重衰减 10^{-5} ）来优化反向传播模型。

4 模型实验结果

4.1 评价指标

采用 Dice 相似系数 (Dice Similarity Coefficient, DSC) 和交并比 (Intersection over Union, IoU) 作为评价指标，对当前模型的性能优劣进行全面评估。Dice 相似系数特别强调图像分割中像素的内部填充，能够对模型的分割精度进行有效约束。而交并比则能衡量预测区域与真实区域之间的重叠程度。将 DSC 与 IoU 结合使用，不仅能够从不同角度反映模型的分割效果，还能相互补充，提高评估的全面性和准确性，从而有助于获得更加精确和可靠的分割结果。这种综合评价方法可以更全面地评价模型在图像分割任务上的表现，为进一步优化模型提供有力的指导。

4.2 实验结果

在 Synapse 数据集上，我们对三种不同的图像分割方法进行了实验，分别是 FCN、U-Net 和 Attention U-Net。实验数据如下表所示：

方法	Train Dice	Train Iou	Val Dice	Val Iou
FCN	0.56	0.55	0.39	0.34
U-Net	0.57	0.56	0.45	0.41
Attention U-Net	0.57	0.55	0.47	0.43

表 1: 不同方法在 Synapse 数据集上的分割精度

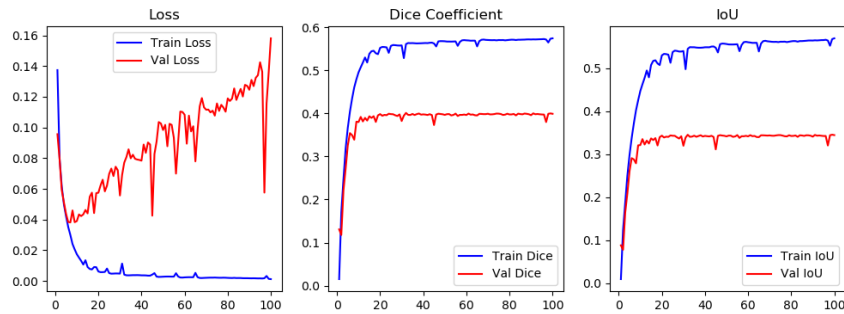


图 3: FCN

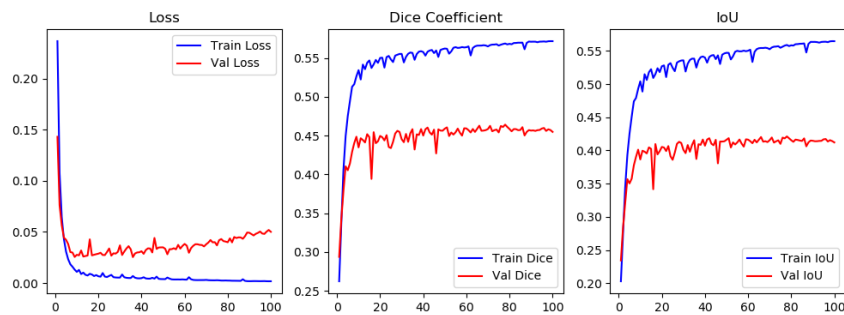


图 4: U-Net

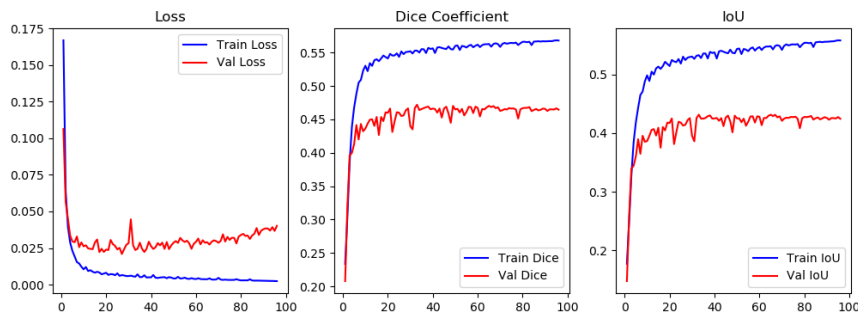


图 5: Attention U-Net

5 相关分析和结论

5.1 结果分析

- 训练集表现:

- FCN 在训练集上的 Dice 系数和 IoU 分别为 0.56 和 0.55，表现较为稳定。
- U-Net 的训练集表现稍优于 FCN，Dice 系数为 0.57，IoU 为 0.56。
- Attention U-Net 的训练集表现与 U-Net 相似，Dice 系数为 0.57，IoU 为 0.55。

- 验证集表现:

- 在验证集上，FCN 的表现最差，Dice 系数为 0.39，IoU 为 0.34。
- U-Net 在验证集上的表现明显优于 FCN，Dice 系数为 0.45，IoU 为 0.41。
- Attention U-Net 在验证集上的表现最好，Dice 系数为 0.47，IoU 为 0.43。

5.2 结论

从实验结果可以看出，Attention U-Net 在 Synapse 数据集上的整体表现最好，其验证集上的 Dice 系数和 IoU 均高于其他两种方法。具体分析如下：

- FCN:

- 在训练集和验证集上的表现均较差，特别是在验证集上的表现远低于其他两种方法，说明其泛化能力较弱。

- U-Net:

- 在训练集和验证集上的表现均好于 FCN，特别是在验证集上的提升明显，说明 U-Net 较好地捕捉到了数据的特征，具有较强的泛化能力。

- Attention U-Net:
 - 在训练集上的表现与 U-Net 相似，但在验证集上的表现最好，说明引入注意力机制后，模型能够更有效地关注重要的特征区域，提高了分割精度。

6 相关问题和解决方法

6.1 问题：测试数据集加载异常

实验中编写 dataset 类构建数据集时，使用相同的方式构建了训练集和测试集中的数据，在训练过程中发现训练集中的数据正常，但是测试集中的张量出现 4 个维度 [1,89,512,512] 不符合预期，在检查代码后发现由于测试集中的一个文件中包含了多组 image 和 label，所以出现了异常的数据。

6.2 解决方法

重新编写了 dataset 类，在读取测试集的时候，预先读取每一个文件中的 images 和 labels，存到数组中，再进行处理，代码实现如下：

```
1 class SynapseDataset_test(Dataset):
2     def __init__(self, data_list, data_dir, transform=None):
3         self.data_list = data_list
4         self.data_dir = data_dir
5         self.transform = transform
6         # self.resize = transforms.Resize((512, 512))
7         self.data_list_len = len(self.data_list)
8
9         self.images = []
10        self.labels = []
11
12
13        for idx in range(self.data_list_len):
14            file_name = self.data_list[idx]+' '
15            file_path = os.path.join(self.data_dir, file_name)
16            # print(file_path)
17
18            with h5py.File(file_path, 'r') as f:
19                images = np.array(f['image'])
20                labels = np.array(f['label'])
21                for img in images:
```

```

22         self.images.append(img)
23
24     for lab in labels:
25         self.labels.append(lab)
26
27     def __len__(self):
28         return len(self.images)
29
30     def __getitem__(self, idx):
31         image = self.images[idx]
32         label = self.labels[idx]
33
34         image = torch.tensor(image, dtype=torch.float32).unsqueeze(0) # ...
            Add channel dimension
35         label = torch.tensor(label, dtype=torch.float32).unsqueeze(0)
36         # print("image_size:{}, ...".format(image.shape), ...)
            label_size:{}}".format(image.shape, label.shape))
37
38     return image, label

```

7 小组成员分工描述和占比

- 梁涵杰 (33.3%) : 数据集处理、模型训练、文档编写
- 李时玉 (33.3%) : FCN 模型构建、模型测试评估、文档编写
- 杨毅 (33.3%) : U-Net 模型构建、Attention U-Net 模型构建、文档编写