COSE474 Project #3: Encoder-Decoder Implementation

2021320322 윤민서

This project focuses on implementing Encoder-Decoder architecture. The project can be divided into two parts: implementing the Original U-Net and implementing an architecture that replaces the Encoder part of the Original U-Net with ResNet-50. Incidentally, there are some things that need to implement the main part and module.

What can be considered an important part of the first part, Original U-Net, is Skip connection, which copies feature maps. In the process of upsampling in the decoder, the encoder's tensor is considered as another input and is concatenated. PyTorch provides a function that operates the concatenation of the tensor, and the application example is as follows.

```
x = torch.cat([conv4, x], dim = 1)
```

Overall, it is structured by repeating 3x3 convolutions, batch normalization, and ReLU, so in the actual U-Net implementation, only the numbers of input features and output features need to be considered well.

The second part, ResNet-Encoder-UNet, the implementation understanding of ResNet of which was implemented in Project 2 became an important part. Since the actual structure of ResNet is not all brought, but only the structure up to layer3, it must be well determined whether downsampling is implemented. Skip connection of the Original U-Net was also an important part of ResNet-Encoder-UNet. Except for the bridge part between out3 and Upconv1, the rest of the Encoder and the corresponding parts of Decoder should also be connected through concatenation operations.

Training and validation accuracy of Original U-Net are as follows.

```
epoch 1 train loss :  0.7502652553662862 train acc :  0.7971079346548724
epoch 1 val loss :  0.9929620075870205 val acc :  0.7404603614463463
```

Training and validation accuracy ResNet-Encoder-UNet are as follows.

```
epoch 1 train loss :  1.0756327885470978 train acc :  0.7151189372964102
epoch 1 val loss :  1.1457486893679645 val acc :  0.7031383858070717
```

It is surprising that the current U-Net, which can perform the semantic segmentation task well through ideas such as upsampling created through basic operations used in CNN implementation, can be applied to binding site prediction of protein structure, 3D segmentation, etc. Since the network is a separate form, the disadvantage of slow learning is somewhat regrettable.