# [COSE474] Deep Learning Project #3: Encoder-Decoder Implementation

2022320315 데이터과학과 김승주

## # Description of code

In the UNet_skeleton, the convDown blocks are part of the contracting path (downsampling), while the convUp blocks are part of the expansive path (upsampling). Each convDown block reduces the spatial resolution and increases the number of channels, and each convUp block does the opposite. In the code, x = torch.cat([conv4, x], dim=1) concatenates two tensors, conv4 and x, along the second dimension (channel dimension).

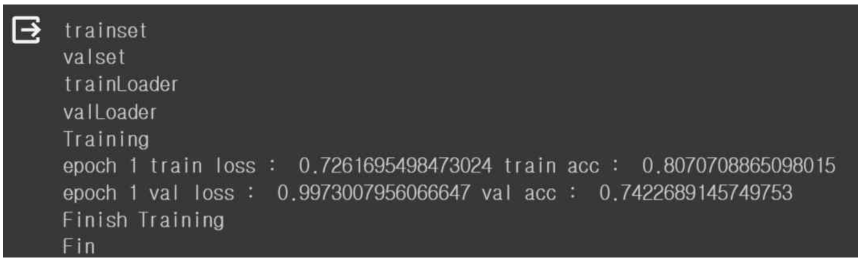
In the modules, usually Zeroing Gradients, Backpropagation, Optimization process is mainly necessary during the training phase. So when testing, I only calculate the model's output and loss.

In resnet_encoder_unet, I overlap my project2. And I also add x = torch.cat([out2, x], dim = 1) code to concatenate two tensors.

In main_skeleton, I use 'criterion = nn.CrossEntropyLoss()' to compute loss and try to minimize this loss. I use 'model.load_state_dict(...)' function. This method of the neural network (model) loads the state dictionary into the model. And I also use 'model.state_dict()' function to return a dictionary representing the current state of the model, including parameters like weights and biases.

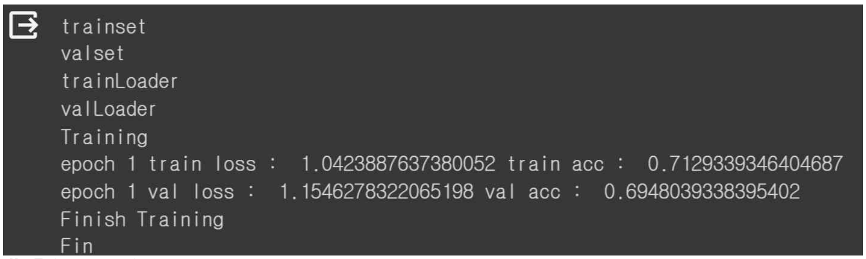
## # Results

UNet



```
trainset
valset
trainLoader
valLoader
Training
epoch 1 train loss :  0.7261695498473024 train acc :  0.8070708865098015
epoch 1 val loss :  0.9973007956066647 val acc :  0.7422689145749753
Finish Training
Fin
```

ResNet-encoder-Unet



```
trainset
valset
trainLoader
valLoader
Training
epoch 1 train loss :  1.0423887637380052 train acc :  0.7129339346404687
epoch 1 val loss :  1.1546278322065198 val acc :  0.6948039338395402
Finish Training
Fin
```

## # Discussions

To improve performance, there are several approaches I can consider. Maybe I can fine-tune Learning Rate Scheduling. The current code uses StepLR to adjust the learning rate every 4 epochs. To achieve better performance, fine-tune learning rate scheduling and experiment with different scheduling strategies. Also I can use deeper or wider Network Architecture I can modify the UNet or ResNet architecture or use a deeper network to improve the model's expressiveness.