

1 Hyperparameter of Networks

ViT와 CNN 모델의 구조를 바꾸가며 여러번 실험을 진행했습니다. CNN의 경우 Convolutional layer의 개수를, ViT에서는 block의 개수를 변경해줬습니다. CNN 모델의 경우 아래 이미지(1a)처럼 3개의 convolutional layer를 이용했을 때 가장 성능이 좋았습니다. 같은 모델을 이용해 learning rate를 0.005로, epoch 수를 75, 100, 150으로 train을 진행했습니다. 실험 결과 150 epoch에서는 overfitting 되어 sharpe ratio가 떨어졌습니다. ViT 모델의 경우 아래 이미지(2b)처럼 6개의 encoder layer를 이용했을 때 가장 성능이 좋았습니다. 이를 이용해 learnign rate를 0.005로, epoch를 50으로 설정하여 train을 진행했습니다.

```
class CNN(nn.Module):
    def __init__(self):
        super(CNN, self).__init__()
        self.conv1 = nn.Conv2d(65, 128, kernel_size=3, stride=1, padding=1)
        self.conv2 = nn.Conv2d(128, 256, kernel_size=3, stride=1, padding=1)
        self.conv3 = nn.Conv2d(256, 512, kernel_size=3, stride=1, padding=1)
        self.pool = nn.MaxPool2d(kernel_size=(2,1), stride=2, padding=0)
        self.fc1 = nn.Linear(512 * 8, 3)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = self.pool(x)
        x = F.relu(self.conv2(x))
        x = self.pool(x)
        x = F.relu(self.conv3(x))
        x = self.pool(x)

        x = x.view(-1, 512 * 8)
        x = self.fc1(x)

        return x
```

```
class VisionTransformer(nn.Module):
    def __init__(self, num_classes = 128, num_patches = 13, emb_dim = 768,
                  num_heads = 12, mlp_dim = 3072, n_layers = 6):
        super(VisionTransformer, self).__init__()
        self.num_classes = num_classes
        self.num_patches = num_patches
        self.emb_dim = emb_dim
        self.num_heads = num_heads
        self.mlp_dim = mlp_dim
        self.n_layers = n_layers

        self.embedding = EmbeddingLayer(num_patches, emb_dim)
        self.blocks = Block(emb_dim, num_heads, mlp_dim, n_layers)
        self.norm = nn.LayerNorm(emb_dim)
        self.fc = nn.Linear(emb_dim, num_classes)
```

(a) 가장 성능이 좋은 CNN model 구조

(b) 가장 성능이 좋은 ViT model 구조

Figure 1: CNN과 ViT 모델 중에서 가장 좋았던 모델 구조

2 Results

아래 figure (2)는 각각 가장 좋았던 CNN과 ViT 모델의 결과입니다. Sharpe ratio로 봤을 때 성능은 CNN이 더 좋았습니다. 하지만 ViT를 훈련하는데 드는 시간이 매우 길었기 때문에 ViT를 더 많이 실험해보지 못했습니다. 그로 인해 더 좋은 모델을 찾지 못했을 가능성이 있으므로 CNN이 항상 더 좋은 모델이라고 확신하기는 어려울 것 같습니다.

```
Model: model_save/OM.pt
{'vUSD': 17597.04638215448, 'v_KRW-BTC': 0, 'v_USD': 17597.04638215448, 'v_KRW-BTC': 0.0002165562747723142, 'buy_count': 27, 'hold_count': 26, 'sell_count': 27}
Profit percentage: 6.1899261575335206
Cumulative return: 0.3579466382154071
Volatility: 0.7501653657955285
Sharma ratio: 2.806562746571629
```

```
Model: model_save/YIT.pt
1/USO: 17643.423938922147, 'X96-8TC': 0, 'y_X96-8TC': 0.0082189053281878638, 'buy_count': 79, 'hold_count': 17, 'sell_count': 7
Profit: 37445.0, 'y_hold': 0.0082189053281878638
Cumulative return: 0.7643423938922147
Volatility: 8.423938922147
Source: ratio: 1.811227987126284
```

(a) CNN model의 결과. Sharpe ratio (b) ViT model의 결과 Sharpe ratio는 3.04 1.81

Figure 2: Result of best models