



OUTTA AI

딥러닝반 _ basic _ 이재빈, 김준영, 류송화, 김성현



In [3]:

```
# 하이퍼파라미터
args = {
    "train_path" : "/kaggle/input/2024-outta-basic-p-1/train.csv",      # train 데이터 경로
    "test_path" : "/kaggle/input/2024-outta-basic-p-1/test.csv",      # test 데이터 경로
    "submit_path" : "/kaggle/input/2024-outta-basic-p-1/sample_submission.csv",  # submit 파일 경
로
    "batch_size" : 64,
    "epochs" : 10,
    "lr" : 2e-5, #원래 2e-5
    "seed_val" : 42          # 절대 수정하지 마세요.
}
```

#	Team	Members	Score	Entries
1	전형진_2545		1.000000000000	26
2	이재빈_0967		1.000000000000	10
3	전승민_5419		0.999581706636	10

P1

Epoch 1/10

100%|██████████| 429/429 [05:50<00:00, 1.22it/s]

CheckPoint : model_state_dict_epoch_1.pth
train_loss : 1.6388346754309737, train_acc : 0.6789038507788507

Epoch 2/10

100%|██████████| 429/429 [05:53<00:00, 1.21it/s]

CheckPoint : model_state_dict_epoch_2.pth
train_loss : 0.6035569190006433, train_acc : 0.9903481934731935

Epoch 3/10

100%|██████████| 429/429 [05:53<00:00, 1.21it/s]

CheckPoint : model_state_dict_epoch_3.pth
train_loss : 0.37411279500503364, train_acc : 0.9995993589743589

Epoch 4/10

100%|██████████| 429/429 [05:53<00:00, 1.21it/s]

CheckPoint : model_state_dict_epoch_4.pth
train_loss : 0.2798570653636417, train_acc : 1.0

Epoch 7/10

100%|██████████| 429/429 [05:53<00:00, 1.21it/s]

CheckPoint : model_state_dict_epoch_7.pth
train_loss : 0.15430610222277386, train_acc : 1.0

Epoch 8/10

100%|██████████| 429/429 [05:53<00:00, 1.21it/s]

CheckPoint : model_state_dict_epoch_8.pth
train_loss : 0.13065275760033193, train_acc : 1.0

Epoch 9/10

100%|██████████| 429/429 [05:53<00:00, 1.21it/s]

CheckPoint : model_state_dict_epoch_9.pth
train_loss : 0.10928513015900458, train_acc : 1.0

Epoch 10/10

100%|██████████| 429/429 [05:52<00:00, 1.22it/s]

CheckPoint : model_state_dict_epoch_10.pth
train_loss : 0.09168697539326194, train_acc : 1.0

P2

Epoch 1/5

100%|██████████| 352/352 [03:41<00:00, 1.59it/s]

Checkpoint : model_state_dict_epoch_1.pth
train_loss : 0.14314281148042277, train_acc : 0.9613813920454546

Epoch 2/5

100%|██████████| 352/352 [03:50<00:00, 1.53it/s]

Checkpoint : model_state_dict_epoch_2.pth
train_loss : 0.05173582624576573, train_acc : 0.9858842329545454

Epoch 3/5

100%|██████████| 352/352 [03:51<00:00, 1.52it/s]

Checkpoint : model_state_dict_epoch_3.pth
train_loss : 0.025225948509449318, train_acc : 0.9937855113636364

Epoch 4/5

100%|██████████| 352/352 [03:52<00:00, 1.52it/s]

Checkpoint : model_state_dict_epoch_4.pth
train_loss : 0.010831700461163895, train_acc : 0.9976029829545454

Epoch 5/5

100%|██████████| 352/352 [03:52<00:00, 1.52it/s]

100%|██████████| 352/352 [03:52<00:00, 1.52it/s]
/opt/conda/lib/python3.10/site-packages/transformers/optimization.py:591: FutureWarning: mplementation of AdamW is deprecated and will be removed in a future version. Use the PyTorch implementation torch.optim.AdamW instead, or set `no_deprecation_warning=True` to disable the future warning
warnings.warn(

Checkpoint : model_state_dict_epoch_5.pth
train_loss : 0.005964267727325601, train_acc : 0.9988458806818182

Epoch 1/5

100%|██████████| 352/352 [03:51<00:00, 1.52it/s]

Checkpoint : model_state_dict_epoch_1.pth
train_loss : 0.026103471202759465, train_acc : 0.9937855113636364

Epoch 2/5

100%|██████████| 352/352 [03:51<00:00, 1.52it/s]

Checkpoint : model_state_dict_epoch_2.pth
train_loss : 0.012479425533249676, train_acc : 0.9976029829545454

Epoch 3/5

100%|██████████| 352/352 [03:51<00:00, 1.52it/s]







Checkpoint : model_state_dict_epoch_3.pth
train_loss : 0.004691906741796133, train_acc : 0.9990234375

Epoch 4/5

100%|██████████| 352/352 [03:51<00:00, 1.52it/s]

In [3]:

```
# 하이퍼파라미터
args = {
    "train_path" : "/kaggle/input/2024-outta-basic-p-2/train.csv",      # train 데이터 경로
    "test_path" : "/kaggle/input/2024-outta-basic-p-2/test.csv",      # test 데이터 경로
    "submit_path" : "/kaggle/input/2024-outta-basic-p-2/sample_submission.csv",
    "max_len" : 128,
    "batch_size" : 32,
    "num_labels" : 2,
    "epochs" : 5,
    "lr" : 2e-5,
    "eps" : 1e-8,
    "seed_val" : 42          # 절대 수정하지 마세요.
}
```

#	Team	Members	Score
1	이재빈_0967	   	0.992915734526
2	박균_3459	 	0.990678598061

P2

-> for문을 이용해 5epoch단위로 learning rate을 줄이며 3번 train 진행

```
In [37]: def train(train_dataloader, valid_dataloader, model, device, args):
        """
        주어진 데이터로 모델을 학습시키는 함수입니다.

        Args:
            train_dataloader (DataLoader): 학습 데이터를 제공하는 DataLoader 객체
            valid_dataloader (DataLoader): 검증 데이터를 제공하는 DataLoader 객체
            model (torch.nn.Module): 학습할 모델
            device (torch.device): 사용할 디바이스 (CPU 또는 GPU)
            args (dict): 학습 관련 인자들을 포함한 딕셔너리

        Returns:
            None
        """
        for i in range(3):
            # (12-1) 옵티마이저와 스케줄러 초기화
            optimizer = AdamW(model.parameters(), lr=args["lr"]) #lr
            scheduler = get_linear_schedule_with_warmup(optimizer, num_warmup_steps=0, num_training_steps=len(train_dataloader) * args["epochs"]) #num_training_steps=128

            # (12-2) 모델의 그래디언트 초기화
            model.zero_grad()

            for epoch in range(args["epochs"]):
                # (12-3) 모델을 훈련 모드로 설정
                model.train()

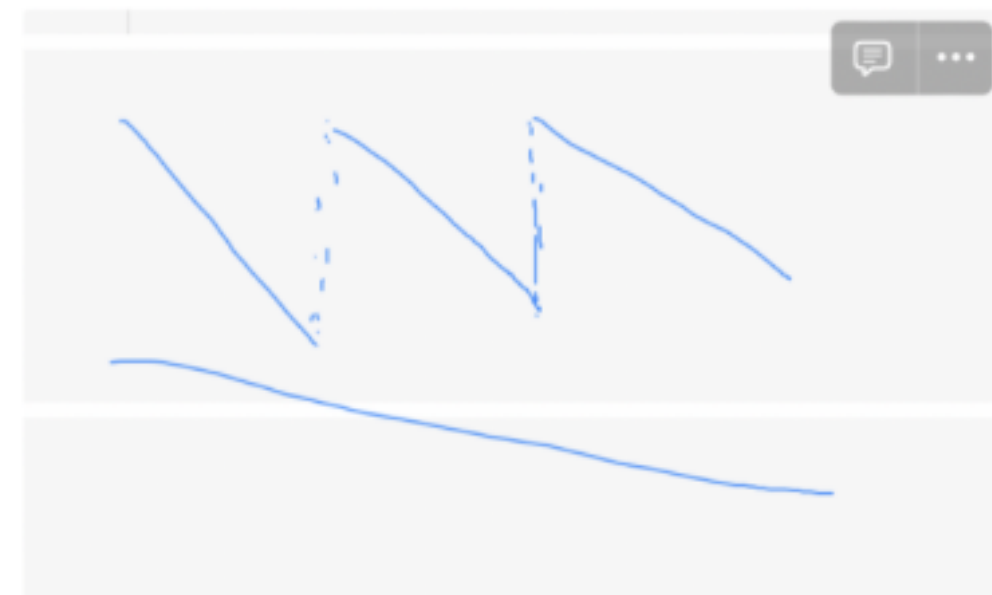
                total_loss = 0 # 전체 손실 초기화
                total_accuracy = 0 # 전체 정확도 초기화

                print(f'Epoch {epoch + 1}/{args["epochs"]}')

                for step, batch in enumerate(tqdm(train_dataloader)):
                    batch = tuple(index.to(device) for index in batch)
                    ids, masks, labels = batch

                    # (12-4) Forward pass
                    outputs = model(ids, attention_mask=masks, labels=labels)
```

- P2
- 전처리 최적화
- 에폭수: 1, 5, 5, 10 (20쯤 넘게)



P3

In [2]:

```
# 하이퍼파라미터
args = {
    "train_path" : "/kaggle/input/2024-outta-basic-p-3/train/train",
    "test_path" : "/kaggle/input/2024-outta-basic-p-3/test/test",
    "submit_path" : "/kaggle/input/2024-outta-basic-p-3/sample_submission.csv",
    "extract_features" : "spectral", # "rhythm"과 "spectral" 중에 선택하세요.
    "batch_size" : 32,
    "num_labels" : 2,
    "epochs" : 40, #40
    "lr" : 2e-5,
    "eps" : 1e-8,
    "seed_val" : 42 # 절대 수정하지 마세요.
}
```

Epoch 1/40

100%|██████████| 29/29 [01:51<00:00, 3.83s/it]

CheckPoint : model_state_dict_epoch_1.pth
Train Loss : 2.1953949517217177, Train Accuracy : 0.20568927789934355

Epoch 2/40

100%|██████████| 29/29 [01:40<00:00, 3.47s/it]

CheckPoint : model_state_dict_epoch_2.pth
Train Loss : 2.0025071234538636, Train Accuracy : 0.3479212253829322

Epoch 3/40

100%|██████████| 29/29 [01:40<00:00, 3.47s/it]

CheckPoint : model_state_dict_epoch_3.pth
Train Loss : 1.759318105105696, Train Accuracy : 0.36542669584245074

Epoch 4/40

100%|██████████| 29/29 [01:39<00:00, 3.44s/it]

CheckPoint : model_state_dict_epoch_4.pth
Train Loss : 1.5800691185326412, Train Accuracy : 0.437636761487965

Epoch 37/40

100%|██████████| 29/29 [01:42<00:00, 3.53s/it]

CheckPoint : model_state_dict_epoch_37.pth
Train Loss : 0.1255034889126646, Train Accuracy : 0.961706783

Epoch 38/40

100%|██████████| 29/29 [01:41<00:00, 3.51s/it]

CheckPoint : model_state_dict_epoch_38.pth
Train Loss : 0.12522499465608392, Train Accuracy : 0.962800875273523

Epoch 39/40










100%|██████████| 29/29 [01:46<00:00, 3.68s/it]

CheckPoint : model_state_dict_epoch_39.pth
Train Loss : 0.13901149468689128, Train Accuracy : 0.9584245076586433

Epoch 40/40

100%|██████████| 29/29 [01:45<00:00, 3.65s/it]

CheckPoint : model_state_dict_epoch_40.pth
Train Loss : 0.12245290928745065, Train Accuracy : 0.9638949671772429

#	Team	Members	Score
1	권혁준_3696	  	0.8800000000000
2	박균_3459	 	0.8733333333333
3	이재빈_0967	   	0.8733333333333

P3

Layer (type)	Output Shape	Param #
Conv1d-1	[-1, 64, 32]	256
BatchNorm1d-2	[-1, 64, 32]	128
ReLU-3	[-1, 64, 32]	0
Conv1d-4	[-1, 64, 32]	12,352
ReLU-5	[-1, 64, 32]	0
Conv1d-6	[-1, 64, 32]	12,352
ReLU-7	[-1, 64, 32]	0
Conv1d-8	[-1, 64, 32]	12,352
ReLU-9	[-1, 64, 32]	0
MaxPool1d-10	[-1, 64, 16]	0
Conv1d-11	[-1, 128, 16]	24,704
BatchNorm1d-12	[-1, 128, 16]	256
ReLU-13	[-1, 128, 16]	0
Conv1d-14	[-1, 128, 16]	49,280
ReLU-15	[-1, 128, 16]	0
Conv1d-16	[-1, 128, 16]	49,280
ReLU-17	[-1, 128, 16]	0
Conv1d-18	[-1, 128, 16]	49,280
ReLU-19	[-1, 128, 16]	0
MaxPool1d-20	[-1, 128, 8]	0
Conv1d-21	[-1, 256, 8]	98,560
BatchNorm1d-22	[-1, 256, 8]	512
ReLU-23	[-1, 256, 8]	0
Conv1d-24	[-1, 256, 8]	196,864
ReLU-25	[-1, 256, 8]	0
Conv1d-26	[-1, 256, 8]	196,864
ReLU-27	[-1, 256, 8]	0
Conv1d-28	[-1, 256, 8]	196,864
ReLU-29	[-1, 256, 8]	0
MaxPool1d-30	[-1, 256, 4]	0

Conv1d-31	[-1, 512, 4]	393,728
BatchNorm1d-32	[-1, 512, 4]	1,024
ReLU-33	[-1, 512, 4]	0
Conv1d-34	[-1, 512, 4]	786,944
ReLU-35	[-1, 512, 4]	0
Conv1d-36	[-1, 512, 4]	786,944
ReLU-37	[-1, 512, 4]	0
Conv1d-38	[-1, 512, 4]	786,944
ReLU-39	[-1, 512, 4]	0
MaxPool1d-40	[-1, 512, 2]	0
Conv1d-41	[-1, 1024, 2]	1,573,888
BatchNorm1d-42	[-1, 1024, 2]	2,048
ReLU-43	[-1, 1024, 2]	0
Conv1d-44	[-1, 1024, 2]	3,146,752
ReLU-45	[-1, 1024, 2]	0
Conv1d-46	[-1, 1024, 2]	3,146,752
ReLU-47	[-1, 1024, 2]	0
Conv1d-48	[-1, 1024, 2]	3,146,752
AdaptiveMaxPool1d-49	[-1, 1024, 1]	0
Flatten-50	[-1, 1024]	0
Dropout-51	[-1, 1024]	0
Linear-52	[-1, 10]	10,250

=====
Total params: 14,681,930
Trainable params: 14,681,930
Non-trainable params: 0

Input size (MB): 0.00
Forward/backward pass size (MB): 0.74
Params size (MB): 56.01
Estimated Total Size (MB): 56.75

- 합성곱 블록으로 구성
- 각 블록은 4개의 convolution layer와 1개의 pooling layer로 구성
- 채널 : 64 ~ 1024개까지 확장
- 각 블록의 첫번째 convolution 이후 batch normalization을 적용
- 모든 convolution 이후 ReLU 사용

< 하나의 블록 >

필터 개수를 2배로 늘리는 convolution층
1개

필터 개수를 유지시키는 convolution층
3개

=> 신경망의 유연함 증가



THANK YOU!

