심층학습
중간 과제

SW융합학부 양희경

CIFAR10 인식 정확도 챌린지

- 제출할 것: **코드(.ipynb, .html), 모델(.pkl), 리포트(.docx)**
  - Google Colab 으로 구현시 '.html' 대신 '.pdf' 제출 가능
- 방법
- 리포트
- 주의
- 조건
- 평가
- 지난학기 수강생 정확도 분포

CIFAR10 인식 정확도 챌린지

- 방법
  - '[04실습]심층신경망 훈련' 커스터마이징하기
  - '[04]심층신경망 훈련 ' 에서 배운 훈련 방법 조합을 사용하여 가능한 만큼 test 세트에 대한 정확도를 높인다.
    - 원래 validation 세트에 대해 정확도를 높인 후 test 세트 정확도는 마지막에 한 번만 측정하는 것이나, 이 과제에서는 생략함

CIFAR10 인식 정확도 챌린지

- 리포트
  - 워드파일(.docx)로 작성할 것(PDF 제출도 가능)
  - 최종 정확도 표시(가장 윗 줄에 적을 것)
  - 실험 일지 형식으로 시도한 방법들 간단히 서술하고 (try & error), 선택한 방법 정리하여 작성
  - 최대 3페이지 이하로 요약

CIFAR10 인식 정확도 챌린지

- 주의
  - 학습 전에 **'model.train()'** 을,
    모든 성능측정 전에 **'model.eval()'** 을 추가할 것
  - 이유
    - test mode 에서는 dropout 의 확률 p 만큼 유닛들이 비활성화되고, train mode 에서는 모든 유닛이 활성화되기 때문.
    - 참고로 따로 지정해주지 않으면 train mode 가 디폴트 세팅이 됨

```
model.train()   # train mode 로 바꾸기 -> dropout, batch normalization 에 영향을 줌.
# === 6. 학습 ===
for i in range(num_epoch):
    for j,[image, label] in enumerate(train_loader):
        x=Variable(image).cuda()
        y_=Variable(label).cuda()

        optimizer.zero_grad()
        output=model.forward(x)
        loss=loss_func(output,y_)
        loss.backward()
        optimizer.step()

        if j%1000==0:
            print(j,loss)

model.eval()  # evaluation(test) mode 로 바꾸기 -> dropout, batch normalization 에 영향을 줌.
ComputeAccr(test_loader, model)
```

CIFAR10 인식 정확도 챌린지

- 조건
  - **CNN 구조 고정**
    - CNN 은 아직 배우지 않았으므로 고정할 것
    - 최신 CNN 사용 금지(Alex, VGG, ResNet 등)
  - 변경 가능한 것
    - 구조적
      - Activation function, fully-connected layer
    - 구조적인 것 제외하고 epoch 수, learning rate 등 하이퍼파라미터와 activation function, data augmentation 방법 등은 자유롭게 변경 가능

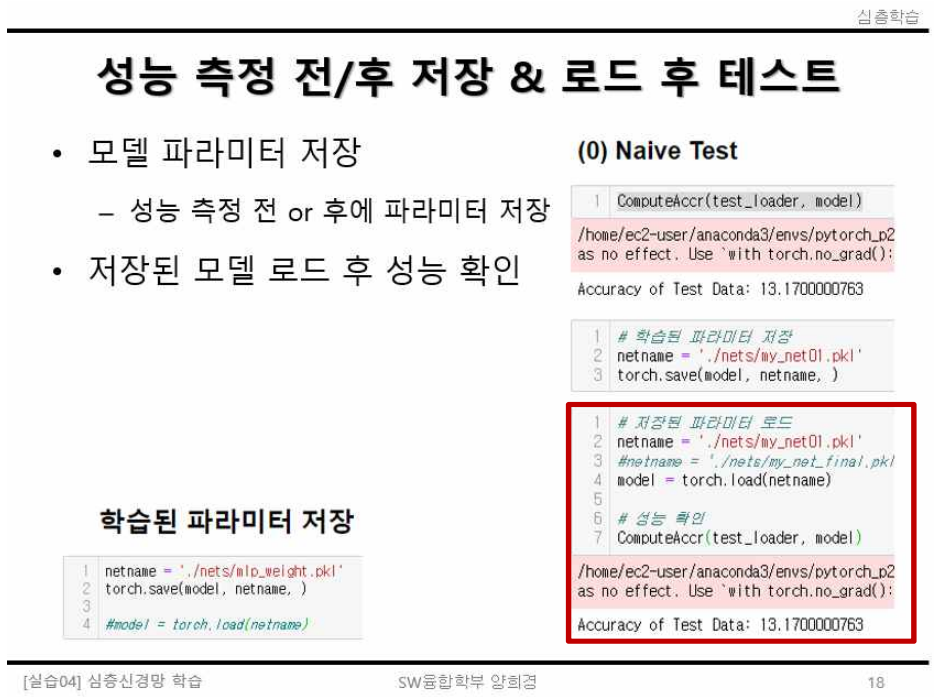
```
# === 4. 모델 선언 ===
class CNN(nn.Module):
    def __init__(self):
        super(CNN,self).__init__()
        self.layer=nn.Sequential(
            nn.Conv2d(3,16,3,padding=1),
            nn.ReLU(),
            #nn.Dropout2d(0.2),     # (1) drop out
            #nn.BatchNorm2d(16),    # (5) Batch normalization
            nn.Conv2d(16,32,3,padding=1),
            nn.ReLU(),
            #nn.Dropout2d(0.2),
            #nn.BatchNorm2d(32),
            nn.MaxPool2d(2,2),
            nn.Conv2d(32,64,3,padding=1),
            nn.ReLU(),
            #nn.Dropout2d(0.2),
            #nn.BatchNorm2d(64),
            nn.MaxPool2d(2,2)
        )
        self.fc_layer=nn.Sequential(
            nn.Linear(64*8*8, 100),
            nn.ReLU(),
            #nn.Dropout2d(0.2),
            #nn.BatchNorm1d(100),
            nn.Linear(100,10)
        )

    def forward(self,x):
        out=self.layer(x)
        out=out.view(batch_size,-1)
        out=self.fc_layer(out)

        return out
model=CNN().cuda()
```

CIFAR10 인식 정확도 챌린지

- 평가
  - 조교의 Colab 또는 AWS 인스턴스에 업로드하여 채점 예정
    - '.ipynb', '.pkl'
  - AWS 또는 Colab 등 작성 및 훈련 환경을 코드 파일 이름에 쓸 것
    - 예) AWS.Py27.ipynb
      Colab.Py27. ipynb
      my_gpu.Py27.ipynb
  - **<u>본인 모델 로드 & 성능 측정 코드 포함</u>**



CIFAR10 인식 정확도 챌린지

- 평가

	점수	평가 요소
주의사항 반영	2	
채점 편의	2	제출한 .pkl 파일을 로드하여 성능확인 코드 포함
리포트	1	첫 장에 성능 기입, 가독성, 실험 충실도 등
정확도 랭킹	5	-제출된 정확도들을 1~5점 구간으로 feature normalization 을 하여 점수 부여
합	10	※ CNN 구조 고정 조건 어길 시 0점 환산되어 최종 과제 점수에 반영

  - 마감일: e-campus 마감일 참조
    - 마감일 이후 제출시 감점

CIFAR10 인식 정확도 챌린지

- 지난학기 수강생 정확도 분포
  - 최고 정확도 훈련 전략
    - Epoch 늘리기(보통 학생 3배)
    - 한계: Test dataset 에 overfitting
    - 무작정 epoch 늘리는 걸 권장하지는 않음

- 학습 시간 많이 소요되는 관계로 Google Colab 으로 학습 시키길 권장
  - '[01실습] 실습도구소개(공통).pdf' 의 '1.2 Colab' 참조

정확도	
	0.00
75.73	78.68
0.00	0.00
75.47	0.00
0.00	76.20
0.00	0.00
76.05	73.92
0.00	75.18
	79.35
0.00	77.32
75.57	76.55
76.19	79.79
74.98	
74.60	0.00
10.04	71.54
87.40	0.00
75.75	
72.00	