Learning Rate Scheduling & Label Smoothing

Winter Vacation Capstone Study

TEAM Kai.Lib

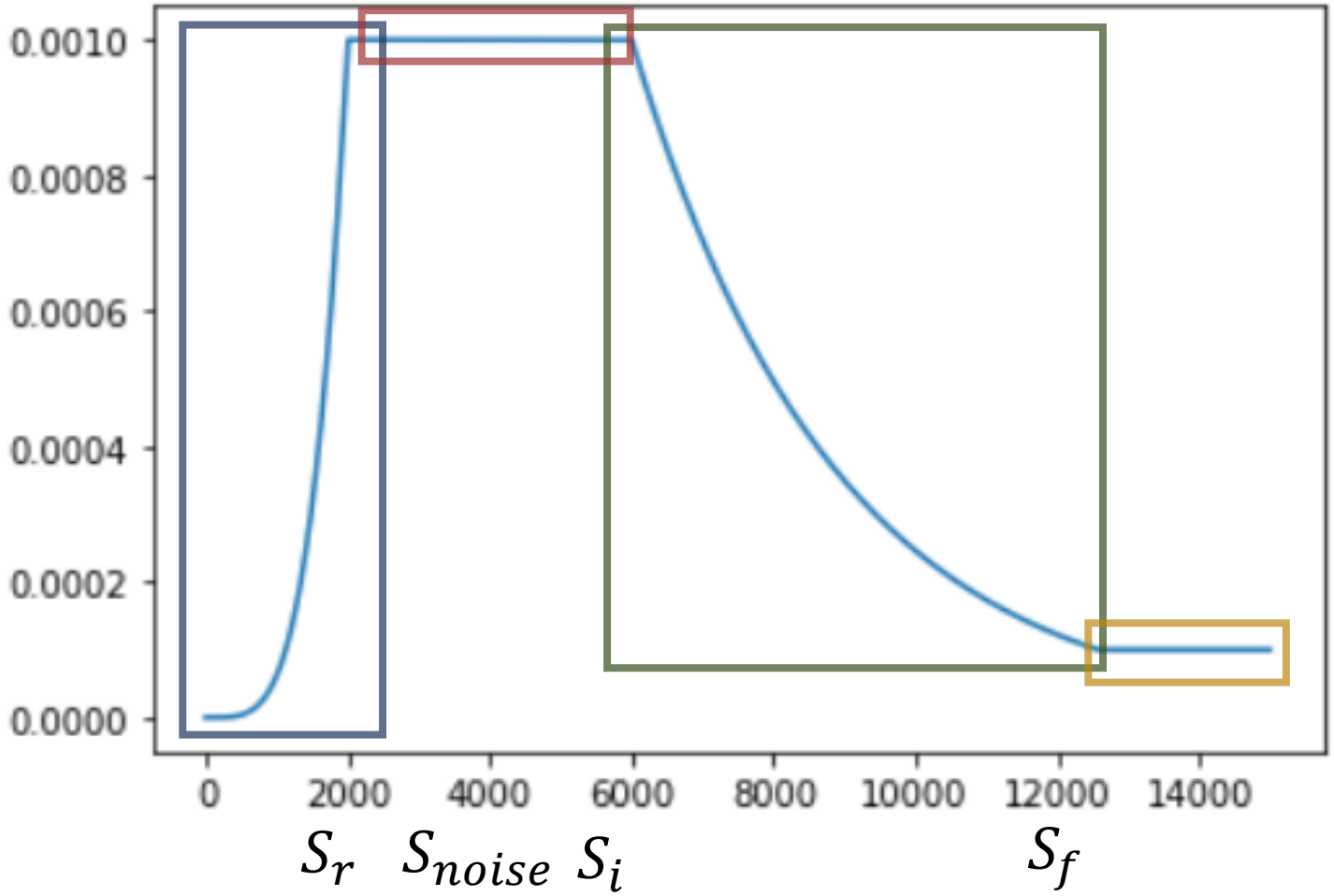
발표자 : 배세영

2020.02.03 (MON)

SpecAugment 논문 [Google Brain, 2019]

- Learning Rate Schedule (3.2) 장에서 모델 learning rate의 변화에 대하여 상세히 기술

- Ramp Up(Burn-In)
- High Plateau
- Exponential Dacay
- Low Plateau

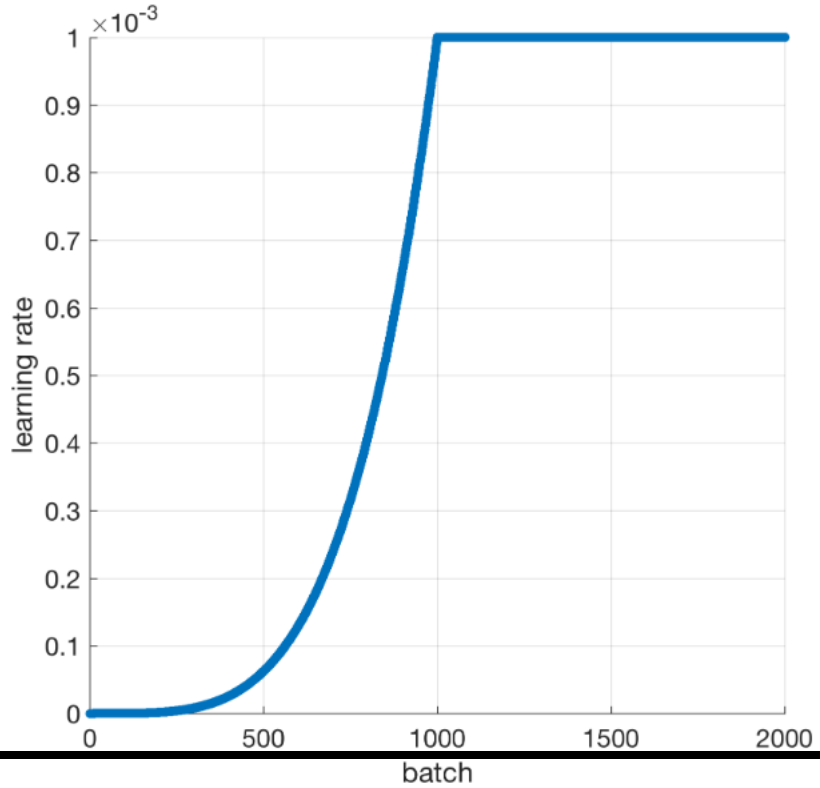


SpecAugment 논문 [Google Brain, 2019]

- Ramp Up (Burn-In)
  - Learning Rate를 점진적으로 증가시키는 단계
  - Linear하게 증가시키는 방법과 Exponential하게 증가시키는 방법이 있으나 주로 Exponential하게 증가시키는 방법을 사용

yolov3/train.py
Lines 115 to 120 in a722601

```
115     # SGD burn-in
116     if (epoch == 0) & (i <= 1000):
117         power = 4
118         lr = 1e-3 * (i / 1000) ** power
119         for g in optimizer.param_groups:
120             g['lr'] = lr
```



SpecAugment 논문 [Google Brain, 2019]

- Ramp Up (Burn-In)
  - 학습 초기 랜덤 값으로 초기화되어 있는 W값 등이 유의미한 값으로 정돈되기까지 기다려 주는 개념
  - LR은 초기에 높고 시간이 지날수록 줄어드는 것이 좋다고 알려져 있으나(multi-step), 학습 초기 짧은 기간동안은 LR이 낮아야 오히려 학습 속도가 빨라진다는 연구 결과가 있음
  - Burn-In Period를 Warm Up Period라고 부르기도 한다.

SpecAugment 논문 [Google Brain, 2019]

- High Plateau
  - Burn-In 단계 이후 최대 학습률을 유지하는 구간
  - High Plateau 단계의 어느 시점(Snoise)에서 Weight Noise 적용 시작, 학습 종료시까지 유지
- Exponential Decay
  - 학습률을 점차 낮추는 구간, 세밀한 학습을 통해 성능을 향상시킴
- Low Plateau
  - 학습 속도가 극도로 느려지는 것을 막기 위해 일정 수치 (High Plateau LR의 1/100) 에 도달하면 학습이 종료될 때 까지 더 이상 학습률을 감소시키지 않고 유지

SpecAugment 논문 [Google Brain, 2019]

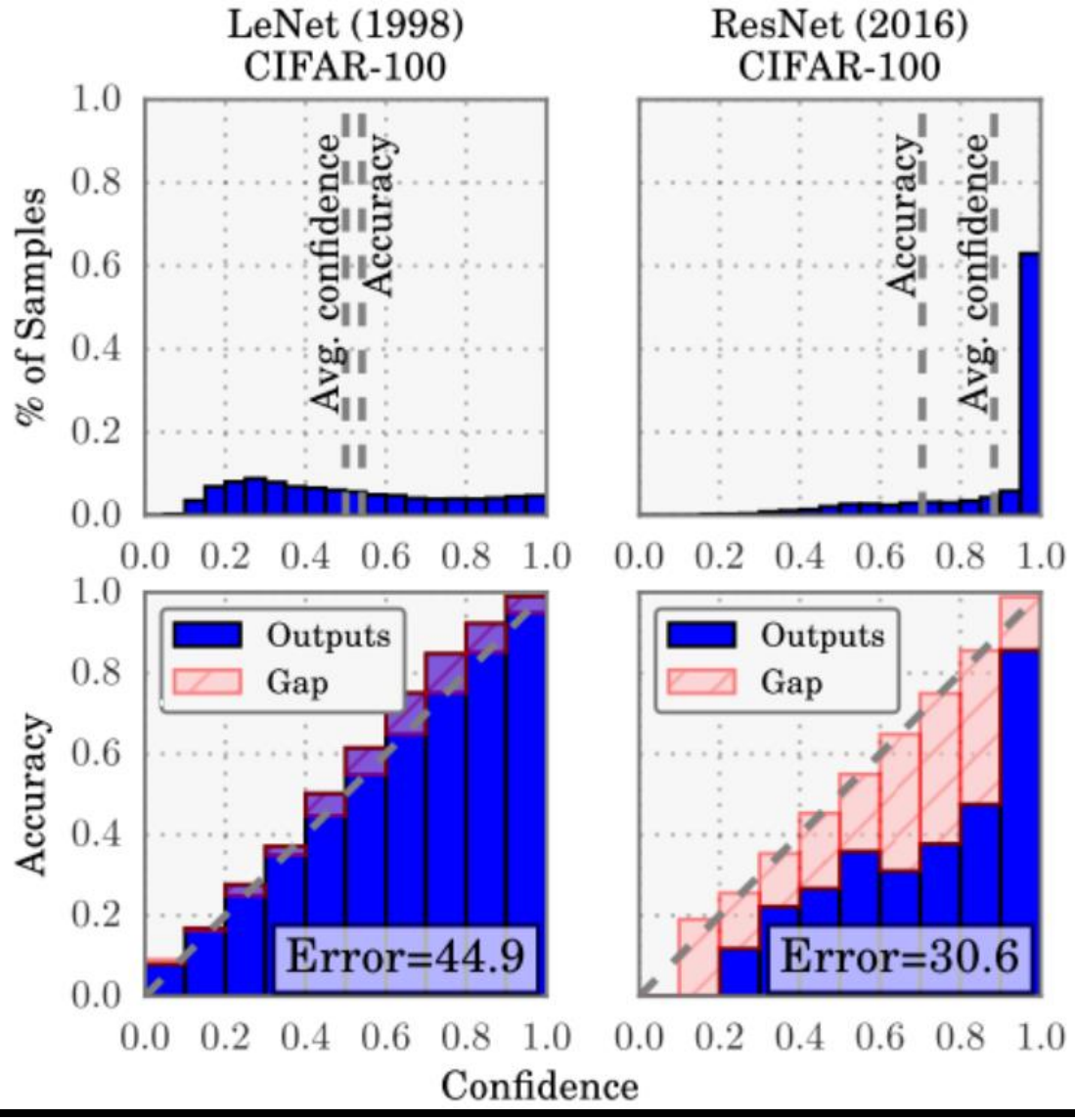
- Weight Noise (High Plateau)
  - Weight에 Noise를 추가하여 모델의 Overfitting을 방지할 수 있다.
  - 0.075의 분산값을 갖는 Noise를 적용했다고 기록되어 있으나, 정확한 적용 방식은 논문 상에서 확인할 수 없었음
  - 하지만 Weight Noise는 Activation Noise(Dropout) 기법의 적용 전에 등장한 개념이고, Activation Noise보다 더 좋은 효과를 보이는지 확실하지 않다는 의견도 있다.

SpecAugment 논문 [Google Brain, 2019]

- Label Smoothing
  - 모델의 과신(Overconfidence)을 막기 위해 도입된 개념
  - Overconfidence
    - 충분히 오래 학습한 분류기는 각각의 Class에 가까운 값을 출력하도록 학습된다.
    - 이 때문에 어느 정도의 불확실성을 갖는 입력조차도 어느 한 Class에 분류되도록 처리되고, 모델은 이를 참이라고 과신(Overconfidence)하게 된다.
  - Calibration
    - 모델의 출력 값이 실제 confidence를 반영하도록 만드는 것.
    - 이진 분류기에서 0.8의 출력이 나왔다면, 0.8의 확률로 맞을 것이 보장되는 개념
  - Calibration되지 않은 (Overconfident 한) 모델의 경우 해석 용이성과 신뢰성에 문제가 있고, 이 모델을 사용한 Downstream Decision의 Threshold를 결정하는 데 영향을 준다.
  - 또한 Ensemble, Pipeline 등의 방식으로 다른 모델과 결합되었을 때 역시 문제가 될 수 있다.

SpecAugment 논문 [Google Brain, 2019]

- **Label Smoothing**



SpecAugment 논문 [Google Brain, 2019]

- Label Smoothing
  - One-Hot Encoded Label Vector와 Uniform Distribution의 조합

$$y_{ls} = (1-\alpha) * y_{hot} + \alpha / K$$

  - K는 Class의 개수, alpha는 Hyper Parameter
  - alpha가 1이라면 단순 uniform distribution, 0이라면 one-hot encoded vector를 사용

  - EX.
    - 이진 분류기에서 alpha = 0.2이라면
    - (1-0.2)*1 + 0.2/2 = 0.8 + 0.1 = 0.9 // (1-0.2)*0 + 0.2/2 = 0 + 0.1 = 0.1
    - 이에 따라 다른 class들의 확률값 역시 상승하게 된다.

SpecAugment 논문 [Google Brain, 2019]

- Label Smoothing
  - 다만 Label Smoothing은 작은 Learning Rate에서 학습을 불안정하게 할 수 있으므로 본 논문 연구에서는 140K의 Timestep까지만 Label Smoothing을 사용하고 이후로는 비활성화하였다.