

8장. 정규화

부산대학교 전기컴퓨터공학부 정보컴퓨터공학전공
201724579 정현모

1. 개요

지금까지 과제에서 진행했던 많은 학습들의 특징은, epoch를 많이 할수록 학습 능력이 좋아진다는 것이다. 하지만 특정 epoch를 넘어가면 오히려 test dataset에서 정확도가 낮아지는 현상을 발견했다. 이는 overfitting(과적합)때문인데, 학습 모델이 train dataset에 과도하게 학습되어서 test dataset에서 정확도가 낮아지는 현상을 의미한다.

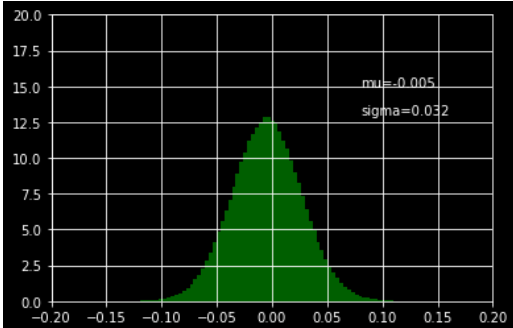
오늘 과제에서 이러한 과적합을 방지하는 정규화에 대해 알아보고, 여러 정규화 기법들, 각 기법들의 특징에 대해 알아본다.

2. 학습 결과

2.1. Flowers 모델 학습 비교

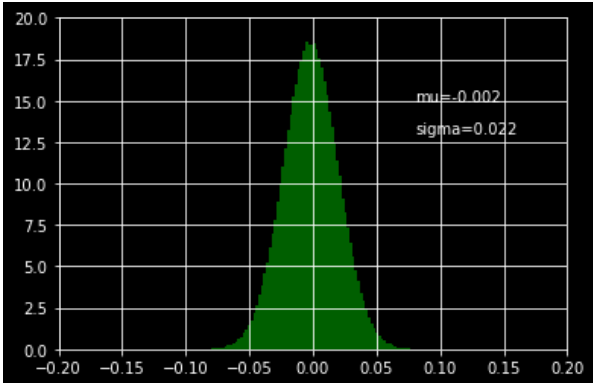
2.1.1. 정규화 기법(L2 손실, L1 손실)에 따른 학습 비교

flowers_model_1

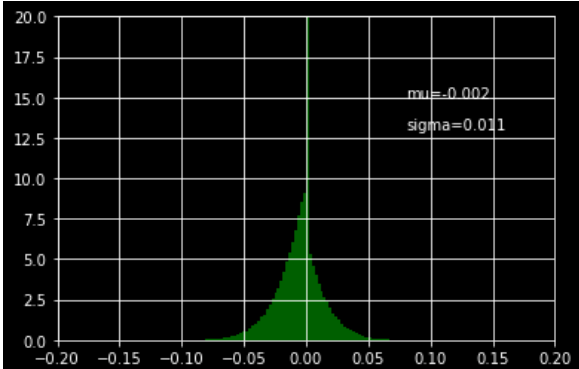
구분	내용	값
조건	Epoch	10
	hidden layer	[30, 10]
	정규화 기법	None
최종 결과	Accuracy	31.1%
	Train time	154 secs
결과 화면	<div>Model flowers_model_1 train started: Epoch 10: cost=1.403, accuracy=0.358/0.240 (30/154 secs) Model flowers_model_1 train ended in 154 secs: Model flowers_model_1 test report: accuracy = 0.311, (0 secs)</div> 	

8장. 정규화

flowers_model_2

구분	내용	값
조건	Epoch	10
	hidden layer	[30, 10]
	정규화 기법	L2_decay = 0.1
최종 결과	Accuracy	23.2%
	Train time	527 secs
결과 화면	<p>Model flowers_model_2 train ended in 527 secs: Model flowers_model_2 test report: accuracy = 0.232, (0 secs)</p> 	

flowers_model_3

구분	내용	값
조건	Epoch	10
	hidden layer	[30, 10]
	정규화 기법	L1_decay = 0.01
최종 결과	Accuracy	23.2%
	Train time	543 secs
결과 화면	<p>Model flowers_model_3 train ended in 543 secs: Model flowers_model_3 test report: accuracy = 0.232, (1 secs)</p> 	

8장. 정규화

2.1.2. 정규화 기법(드롭아웃, 잡음주입, 배치정규화)에 따른 학습 비교

Model flowers_cnn_1

구분	내용	값
조건	Epoch	10
	hidden layer	['conv', {'ksize':3, 'chn':6}], ['max', {'stride':2}], ['conv', {'ksize':3, 'chn':12}], ['max', {'stride':2}], ['conv', {'ksize':3, 'chn':24}], ['avg', {'stride':3}]
	정규화 기법	None
최종 결과	Accuarcy	64.9 %
	Train time	692 secs
결과 화면	Model flowers_cnn_1 train started: Epoch 2: cost=1.118, accuracy=0.569/0.530 (138/138 secs) Epoch 4: cost=0.913, accuracy=0.658/0.570 (139/277 secs) Epoch 6: cost=0.786, accuracy=0.701/0.620 (137/414 secs) Epoch 8: cost=0.669, accuracy=0.748/0.620 (138/552 secs) Epoch 10: cost=0.578, accuracy=0.787/0.700 (140/692 secs) Model flowers_cnn_1 train ended in 692 secs: Model flowers_cnn_1 test report: accuracy = 0.649, (3 secs)	

Model flowers_cnn_2

구분	내용	값
조건	Epoch	10
	hidden layer	['conv',{'ksize':3, 'chn':6}], ['max',{'stride':2}], ['dropout', {'keep_prob':0.6}], ['conv',{'ksize':3, 'chn':12}], ['max',{'stride':2}], ['dropout', {'keep_prob':0.6}], ['conv',{'ksize':3, 'chn':24}], ['avg',{'stride':3}], ['dropout', {'keep_prob':0.6}]
	정규화 기법	Drop out
최종 결과	Accuarcy	55.7 %
	Train time	823 secs
결과 화면	Model flowers_cnn_2 train started: Epoch 2: cost=1.207, accuracy=0.490/0.500 (162/162 secs) Epoch 4: cost=1.064, accuracy=0.572/0.570 (163/325 secs) Epoch 6: cost=0.986, accuracy=0.621/0.520 (161/486 secs) Epoch 8: cost=0.918, accuracy=0.641/0.570 (172/658 secs) Epoch 10: cost=0.898, accuracy=0.658/0.550 (165/823 secs) Model flowers_cnn_2 train ended in 823 secs: Model flowers_cnn_2 test report: accuracy = 0.557, (5 secs)	

8장. 정규화

Model flowers_cnn_3

구분	내용	값
조건	Epoch	10
	hidden layer	<pre>['noise', {'type':'normal','mean':0,'std':noise_std}], ['conv',{'ksize':3, 'chn':6}], ['max',{'stride':2}], ['noise', {'type':'normal','mean':0,'std':noise_std}], ['conv',{'ksize':3, 'chn':12}], ['max',{'stride':2}], ['noise', {'type':'normal','mean':0,'std':noise_std}], ['conv',{'ksize':3, 'chn':24}], ['avg',{'stride':3}]</pre>
	정규화 기법	잡음 주입
최종 결과	Accuarcy	61.3 %
	Train time	890 secs
결과 화면	<p>Model flowers_cnn_3 train started:</p> <p>Epoch 2: cost=1.099, accuracy=0.580/0.610 (180/180 secs)</p> <p>Epoch 4: cost=0.932, accuracy=0.646/0.620 (177/357 secs)</p> <p>Epoch 6: cost=0.805, accuracy=0.697/0.660 (176/533 secs)</p> <p>Epoch 8: cost=0.707, accuracy=0.740/0.680 (176/709 secs)</p> <p>Epoch 10: cost=0.590, accuracy=0.780/0.630 (181/890 secs)</p> <p>Model flowers_cnn_3 train ended in 890 secs:</p> <p>Model flowers_cnn_3 test report: accuracy = 0.613, (4 secs)</p>	

Model flowers_cnn_4

구분	내용	값
조건	Epoch	10
	hidden layer	<pre>['batch_normal'], ['conv',{'ksize':3, 'chn':6}], ['max',{'stride':2}], ['batch_normal'], ['conv',{'ksize':3, 'chn':12}], ['max',{'stride':2}], ['batch_normal'], ['conv',{'ksize':3, 'chn':24}], ['avg',{'stride':3}]</pre>
	정규화 기법	batch 정규화
최종 결과	Accuarcy	53.0 %
	Train time	688 secs
결과 화면	<p>Model flowers_cnn_4 train started:</p> <p>Epoch 4: cost=0.878, accuracy=0.661/0.400 (128/296 secs)</p> <p>Epoch 6: cost=0.785, accuracy=0.699/0.480 (132/428 secs)</p> <p>Epoch 8: cost=0.689, accuracy=0.733/0.540 (134/562 secs)</p> <p>Epoch 10: cost=0.608, accuracy=0.774/0.540 (126/688 secs)</p> <p>Model flowers_cnn_4 train ended in 688 secs:</p> <p>Model flowers_cnn_4 test report: accuracy = 0.530, (4 secs)</p>	

8장. 정규화

3. 결론

3.1. 과적합

3.1.1. 과적합이란

과적합은 신경망이 학습 데이터에 지나치게 맞추어지는 것을 의미한다. Test dataset에서 정확도가 떨어지게 되는데, 이는 곧 일반화 단계에서 성능이 저하되는 것을 의미한다. 데이터의 질이 떨어지거나 양이 적으면 이러한 과적합은 더욱 두드러지지만, 현재 대처 방안이 많이 나와 있어 대처 방향을 잡기는 쉬운 편이다.

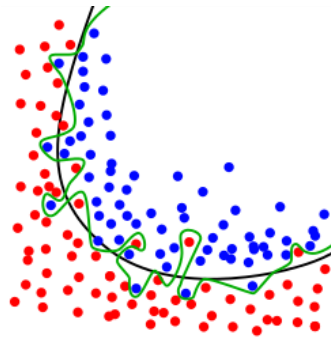


그림 1. 과적합 예시

3.1.2. 과적합 상태

과적합 상태는 그림2와 같이 validation dataset에서의 loss와 train dataset에서의 loss가 크게 차이나는 부분에서 과적합 여부를 판단할 수 있다. Train dataset의 loss는 계속해서 감소하지만 validation dataset의 loss가 어느순간 증가한다는 것은 학습 모델이 과하게 학습되었다는 것을 의미한다.

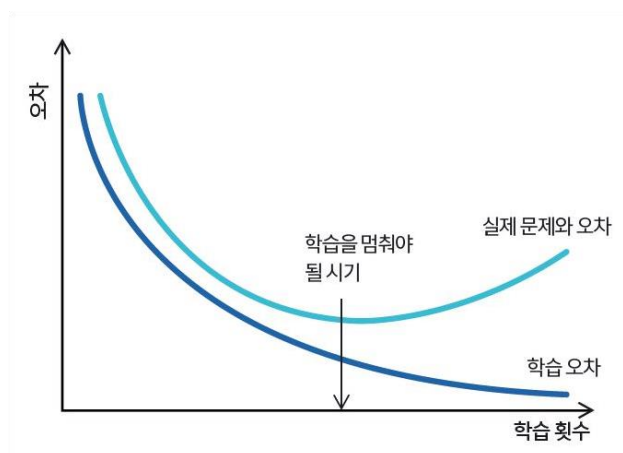


그림 2. 과적합 판단 예시

8장. 정규화

3.2. 정규화

3.2.1. 정규화란

3.2.2. 각 정규화 기법의 특징

- L2 정규화

L2 정규화는 절대값이 큰 파라미터에 불이익을 주는 정규화 기법으로 파라미터 값의 폭주를 막고 작은 절댓값을 갖도록 압박하는 정규화 기법이다. 본연의 학습 효과 때문에 모든 파라미터의 절댓값이 줄어들진 않는다. 아래 수식에서 알 수 있듯이 절댓값이 클수록 패널티를 많이 부여하는 것을 알 수 있다.

$$L' = L + P_{L2} = L + \frac{1}{2} \lambda \sum_{i=1}^n w_i^2 \quad (\lambda : \text{L2 손실률})$$

※ w_i : 전체 계층의 가중치 파라미터 및 커널 파라미터

- L1 정규화

L1 정규화는 전체적으로 L2 정규화와 비슷하지만 더욱 강한 압박을 준다. 수식은 아래와 같으며, 파라미터 갱신 때 기존의 손실 기울기에 $\alpha \text{sign}(w_i)$ 값을 더해서 처리하기 때문에 항상 어느정도 패널티를 가진 채로 학습을 진행한다는 특징이 있다.

$$L' = L + P_{L1} = L + \alpha \sum_{i=1}^n |w_i| \quad (\alpha : \text{L1 손실률})$$

※ w_i : 전체 계층의 가중치 파라미터 및 커널 파라미터

- Drop out

드롭아웃은 이름에서 유추할 수 있듯이 계층 입력 중 일부만 이용해 신경망을 학습시키는 규제화 기법이다. 가장 간단하면서도 효과가 커서 널리 이용된다. 계산량이 줄어드는 기법이 아니며, 오히려 기존 처리 과정에서 마스킹 처리가 추가되어 계산량이 약간 더 증가한다는 특징이 있다.

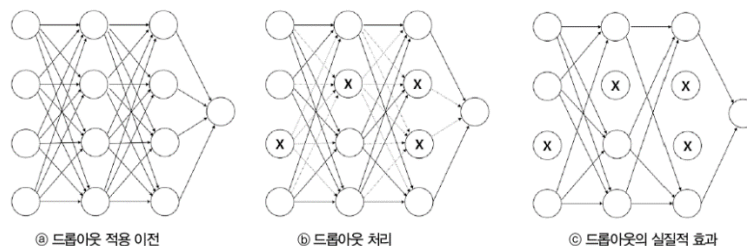


그림 3. 드롭아웃 기법의 동작방식과 효과

8장. 정규화

- 잡음 주입

두 계층사이에 삽입되어 은닉벡터에 잡음을 추가하여 전달하는 기법이다. 학습단계에서 랜덤한 잡음을 생성해 주입하는 방식으로 그만큼 학습에 혼란을 겪게 되지만 이후 다양한 입력에서 더 잘 처리할 수 있게 된다.

- 배치정규화

배치 정규화는 미니배치 내에서 데이터들에 대해 벡터 성분 별로 정규화를 수행하는 것을 의미한다. 동일한 선형변환으로 데이터들을 평균 0, 표준편차 1 분포로 만드는 처리로, 입력의 성분 별 분포가 심하게 다를 때 특히 장점을 갖는다.

3.3. 실험 결과

L2 정규화와 L1 정규화를 진행한 그래프를 보면, 정규화를 진행하지 않은 파라미터는 0에 가까운 파라미터가 약 83만개 중에 192개로 상당히 적은 수를 가졌음을 알 수 있다. 하지만 L2 정규화를 진행했을 경우 343개로 192개보다 좀더 많은 0에 가까운 파라미터를 가짐을 알 수 있다. 또한 그래프 역시 0에 가까운 값들이 더 많아졌음을 시각적으로 확인할 수 있다. L1 정규화의 경우 압도적으로 0에 가까운 파라미터를 가졌는데 83만개중 약 60만개의 개수를 보였다. 이로써 절대값이 파라미터에 패널티를 준다는 의미를 확인할 수 있다.

Drop out과 잡음 주입 기법을 사용했을 때에는 같은 epoch에서 시간이 더 오래 걸렸음을 확인할 수 있다. 이는 위에서 언급했듯이 Drop out은 마스크를 만드는데 시간이 조금 더 소요되는 것이고, 잡음 주입은 계층사이에 삽입함으로써 시간이 더 소요되는 것이다. 그에 반해 배치 정규화는 시간상 차이가 거의 나지 않았는데, 이는 입력 배치의 정규화에 큰 시간이 소요되지 않음을 알 수 있다.

정규화를 포함한 학습을 진행한 학습 결과에서 예상과 너무나도 다른 결과가 나왔다. 정규화를 진행한 학습모델이 정규화를 진행하지 않은 학습 결과보다 훨씬 못 미치는 정확도를 보였다. 이는 정규화가 epoch가 적은 환경에서 오히려 정확도를 감소시킬 수 있다는 뜻인데, 대부분의 정규화 기법이 학습을 방해하면서 일반화를 시키는 방식을 사용하기 때문이다. Epoch가 늘어나면 늘어날 수록 정규화한 모델의 정확도가 높아질 것으로 예상된다.