딥 러닝에 사용되는 매개변수들 간의 상관관계 분석 및 최적화 방법

김연규*, 박호준*, 이상걸*, 차의영*
*부산대학교 전기전자컴퓨터공학과
e-mail : ykim1024@pusan.ac.kr, pn1012@naver.com,
leesg@pusan.ac.kr, eycha@pusan.ac.kr

Correlation Analysis and Optimization between Parameters using with Deep Learning

Yeon-Gyu Kim*, Ho-Jun Park*, Sang-Geol Lee*, Eui-Young Cha*
*Dept of Electrical and Computer Science Engineering,
Pusan National University

요 약

본 논문에서는 영상인식을 위한 딥 러닝에서 사용되는 매개변수 최적화 방법을 제안한다. 학습 성능에 영향을 미치는 매개변수 중 이미지 배치 사이즈 값, 초기 학습율, 최대 학습 반복 횟수에 대해 상호간의 관계를 분석하고 성능을 개선시키기 위해 값을 최적화하는 방법을 연구한다. 제안된 방법을 통한 개선 정도를 분석하기 위해 매개변수의 변화에 따른 학습 소요 시간, 정확도 향상 추이, 메모리 사용량의 변화를 측정한다. 측정된 학습 소요 시간, 정확도 향상 추이, 메모리 사용량의 변화를 분석한 결과 배치 사이즈와 초기 학습 율은 같은 비율로 반비례하게 값을 적용할 때가 이상적 이였으며 서로 다른 환경에서 각각의 학습 소요시간을 측정하는 것으로 배치 사이즈 값과 초기 학습 율에 따른 최적의 최대 학습 반복 횟수를 획득할 수 있었다.

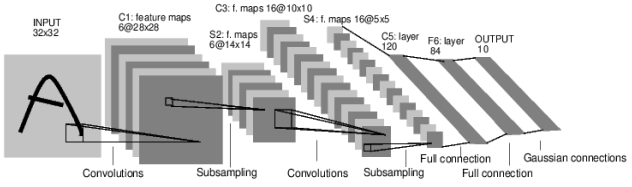
1. 서론

최근 영상 인식을 위해 다양한 방법의 딥 러닝 기술이 적용되고 있다. CNN(Convolutional Neural Network)은 현재 영상 인식 분야 딥 러닝에서 높은 성과를 보이는 구조들이 대부분 채택하고 있는 기본적인 학습 형태이다. CNN을 기반으로 많은 변형구조가 제안되고 있는데 Ciresan[1] 등은 Multi-column 개념을 딥 러닝 CNN구조에 적용하였으며 Krizhevsky[2] 등은 두 개의 GPU를 동시에 사용하여 각각 CNN을 동작하게 하는 것으로 빠르고 더 정확한 학습을 가능하도록 하였다. 또한 최근 Google은 Bhaskara[3] 등이 제안한 원리에 착안하여 크기가 서로 다른 3~4개의 Convolutional mask를 영상에 적용한 후 9개의 모듈에 통과시켜 결과를 조합하는 방법으로 학습 성능을 크게 향상시킨 GoogLeNet을 발표하였다.

[1]~[3]에서는 CNN의 성능을 향상시키기 위해 구조를 개선한 것과 달리 본 논문에서는 CNN의 결과에 영향을 미치는 매개 변수들에 대해 연구하고자 한다. Denil[4]등이 매개변수의 값을 예상하기 위한 시도를 하였지만 대부분 가중치를 중심으로 연구되었다. 가중치 이외에도 영상인식을 위한 CNN구조에는 반드시 공통적으로 사용되는 매개변수들이 존재한다. 그들 중에서 학습 성능에 가장 크게 영향을 끼치는 매개변수들인 초기 학습율(base learning rate), 이미지 배치 사이즈(image batch size), 최대 학습 반복 횟수(max iteration) 에 대해 실험하고 연구한다. 각각의 매개변수 값들을 변경하면서 CNN에 반영한 후 하나의 매개변수가 변경될 때 다른 매개변수들은 어떤 방식으로 다시 계산되어질 필요가 있는지 상관관계를 분석한다. 제안한 방법으로 분석한 결과 매개변수들의 상관관계를 정확히 알 수 있었고 상황에 따라 최적에 가까운 값을 유추할 수 있었다.

2. 매개변수들의 상관관계 분석 및 최적화 방법

본 논문에서는 매개 변수들 간의 상관관계를 분석하고 상황에 맞는 최적의 매개변수 값을 찾는 방법을 연구하기 위해 <표 1>과 같이 매개변수들을 설정하고 실험한다. 실험을 통해 각각의 경우의 수에 따른 학습에 걸리는 소요시간, 학습의 정확도의 변화 추이를 모두 파악한다. 특정 경우의 수를 기준으로 매개 변수의 값을 가감하였을 때 발생하는 효과를 그래프 등으로 제시 및 비교하여 어떤 경우의 수에서 최적의 학습 성능을 나타내는지 분석한다. 제안된 모든 실험은 (그림 1)과 같이 Yann이 제안한 CNN구조의 LeNet[5]에서 이루어진다.



(그림 1) LeNet의 구조

<표 1> 실험에 사용된 매개변수 설정 값

	배치 사이즈 (학습)	배치 사이즈 (테스트)	초기 학습율 (α)	최대 반복횟수
CASE1	64	128	0.01	1000
CASE2	32	64	0.01	1000
CASE3	128	256	0.01	1000
CASE4	64	128	0.04	1000
CASE5	64	128	0.001	1000
CASE6	32	64	0.02	1000
CASE7	64	128	0.0025	1000
CASE8	64	128	0.1	1000
CASE9	32	64	0.005	1000

<표 1>에 제시한 9가지의 매개변수 설정 값들은 배치 사이즈만을 변수로 하였을 때 결과에 미치는 영향, 초기 학습율만을 변수로 하였을 때 결과에 미치는 영향, 두 매개변수를 동시에 변경하였을 때 결과에 미치는 영향을 분석하기 위해 구성하였다. 자세한 분석내용은 3.실험 결과 및 분석에서 설명한다.

실험을 진행하기 위해서는 학습이 진행될 때마다 학습율을 감소시키는 방법에 대한 논의가 필요하다. 본 실험에 사용되는 MNIST(Mixed National Institute of Standards and Technology database)[5] 데이터 셋은 60,000개의 학습데이터와 10,000개의 테스트 데이터로 이루어져 있어 학습율의 변화를 모든 세트에 대해 계산하여 반영하는 것 대신 학습이 반복될 때 마다 임의로 선택된 몇 개의 샘플들에 대해서만 변화율을 계산해서 다음 가중치에 반영하는 Stochastic Gradient Descent(SGD)[6] 방식을 사용한다. 식(1)은 학습이 반복될 때마다 학습율이 감소하는 정도를 나타낸 식이다.

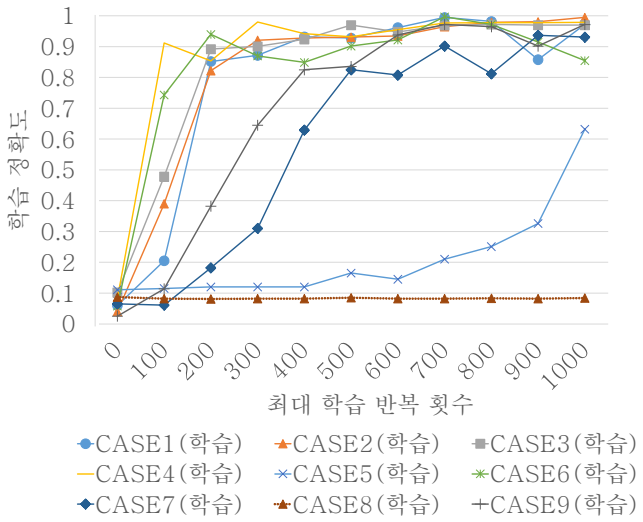
$$\alpha_{next} = \alpha_{current} \times (1 + \gamma \times iteration_{current})^{-power} \quad (1)$$

α_{next}와 $\alpha_{current}$는 각각 다음 세대의 학습율과 현재 세대의 학습율을 의미하고 $iteration_{current}$는 현재까지 진행된 학습 반복 횟수를 의미한다. 감마(γ)값으로 0.001을 사용하고 power값으로 0.75를 사용하여 (식 1)을 통해 다음 세대에 사용할 학습율을 계산하게 된다.

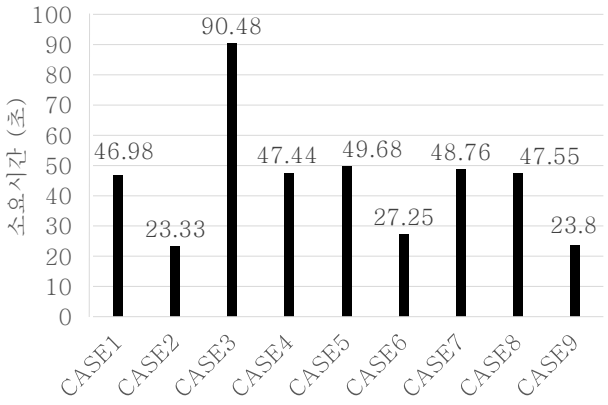
실험 결과의 신뢰성을 위해 배치 사이즈, 초기 학습 율을 제외한 매개변수와 식은 고정 값으로 실험이 진행되는 동안 일절 변경되지 않는다.

3. 실험 결과 및 분석

(그림 2)를 통해 <표 1>에 제시된 설정 값들에 대한 학습 정확도의 변화 값을 나타내었다. 최대 1000회의 학습이 반복될 때까지 100회의 학습마다 측정된 학습 정확도의

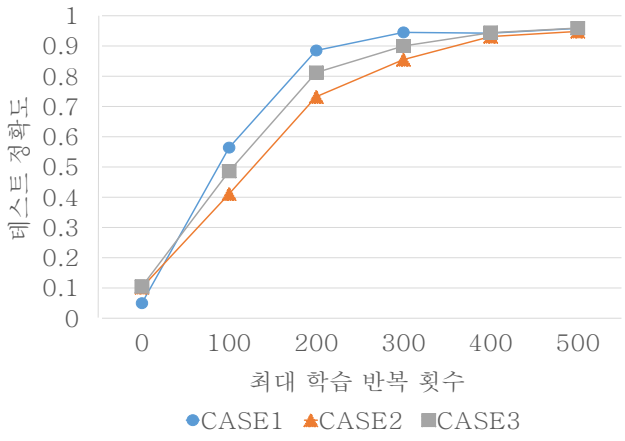


(그림 2) <표 1>에 제시된 모든 매개변수 설정 값들에 대한 학습 정확도

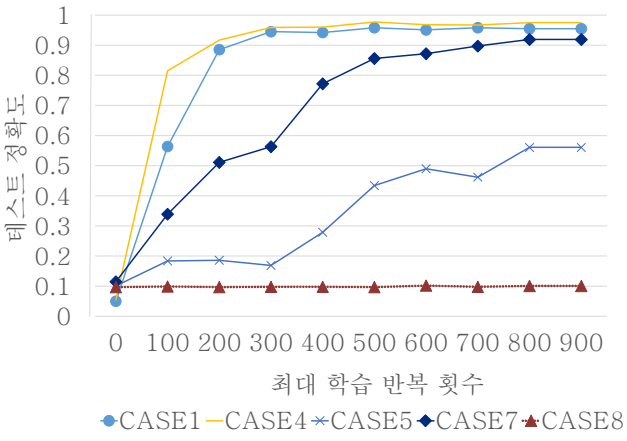


(그림 3) 학습 횟수 100회에 소요되는 평균 시간

값은 9가지의 CASE마다 모두 다른 것으로 나타났다. 주목할만한 점으로 CASE6과 같은 배치 사이즈 값이 다른 CASE들에 비해 상대적으로 작고 초기 학습 율이 다른 CASE보다 큰 경우가 학습될 때 200회 이상의 학습 시점에서는 학습 정확도가 10%의 오차로 계속 진동하는 모습을 확인할 수 있다. 즉 200회 이상의 시점부터는 높은 초기 학습 율로 인한 오버피팅(Overfitting) 현상이 나타난다고 추측할 수 있다. 반대로 CASE5, CASE7, CASE9의 경우 초기 학습율이 다른 CASE와 비교하여 상당히 낮은 경우이다. 이들의 경우에는 다른 CASE에 비해 느리게 학습 정확도 수치가 올라가고 있다. 하지만 CASE5를 제외하면 1,000회의 학습이 이루어진 시점에서는 모두 0.95의 높은 학습율을 나타내었고 CASE5 또한 900~1000회의 시점부터는 가파른 학습 정확도 상승을 보여주었다. 하지만 CASE8의 경우처럼 다른 CASE들보다 초기 학습 율이 10배 이상 크다면 전혀 학습이 진행이 되지 않고 학습 정확도가 0.1미만에 머무는 것을 확인할 수 있다.



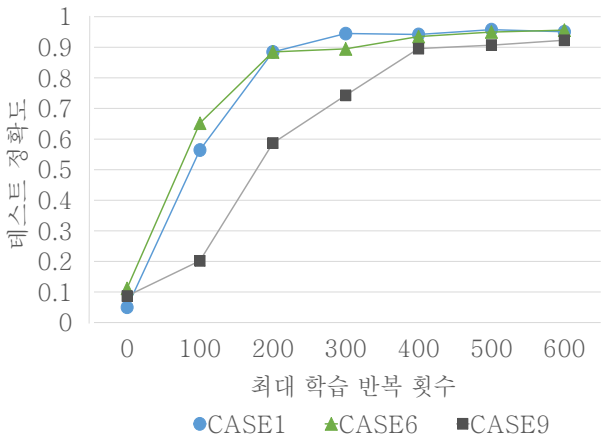
(그림 4) 배치사이즈가 변경될 때의
학습 횟수 별 테스트 정확도



(그림 5) 초기 학습율이 변경될 때의
학습 횟수 별 테스트 정확도

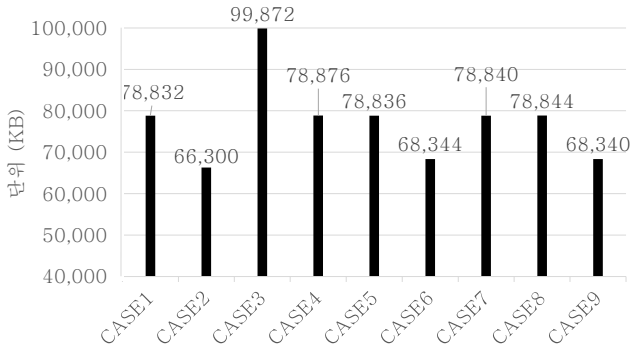
다음으로 <표 1>의 설정 값들에 대해 최대 1,000번의 학습이 진행되는 동안 100번의 학습이 진행되는데 걸리는 평균 소요 시간을 (그림 3)에 나타내었다. CASE1, CASE2, CASE3의 매개변수 값을 비교해보면 초기 학습율과 최대 반복 횟수는 고정하고 배치 사이즈 값을 CASE1에 기준하여 CASE2, CASE3에 각각 1/2배, 2배를 적용하였다. 결과적으로 소요시간은 배치 사이즈 크기에 선형적으로 비례하여 증가하거나 감소하였다. 즉, 매 학습마다 임의로 선택되어 사용되는 영상의 개수를 뜻하는 배치 사이즈 개수가 증가하면 그 만큼 전체 학습 소요시간도 선형적으로 증가하는 것을 알 수 있다. 단, 배치 사이즈를 고정하고 학습율만을 변경시켰을 때는 주목할 만한 소요시간 변화는 나타나지 않았다.

다음으로 배치 사이즈의 변화가 테스트 정확도에 미치는 영향을 확인해 보았다. (그림 4)는 CASE1,2,3에 대해 학습이 진행되면서 보이는 테스트 정확도에 대한 발전 과정을 보여준다. 표면적으로는 CASE1에서 가장 빠르게 테스트 정확도가 향상되며 한계 값에 다다르는 것으로 보인다. 하지만 (그림 3)의 100번의 학습 반복에 소요되는 시간적 비용을 고려해야한다. 테스트 정확도의 최종 한계 값으로 보이는 0.95근처의 값에 도달하는 시간이 CASE1이 CASE2보다 2배 이상 걸린다는 점이다. 배치사이즈가 더 큰 CASE3의 경우에는 CASE1보다 2배의 소요시간이 소요된다. 단, (그림 4)의 CASE1,2 곡선에서 확인할 수 있는 것은 배치사이즈가 큰 CASE1의 경우 단위 반복 횟수당 증가하는 테스트 정확도는 분명 높아지다는 점이다. 만일 학습 소요시간을 충분히 줄일 수 있는 하드웨어 환경을 갖출 수 있다면 배치 사이즈를 늘려 작은 학습 반복으로도 한계 테스트 정확도에 다다를 수 있도록 하는 것이 좋다. 또한 최대 학습 반복 횟수를 1000으로 고정시켜 놓았으므로 어느 지점에서 테스트 정확도가 한계점에 도달하는지 확인할 수 있다. 배치 사이즈가 달라지더라도 모두 400회의 학습 반복이 이루어졌을 때 테스트 정확도 한계점에 도달하는 것을 확인할 수 있다.



(그림 6) 초기 학습율과 배치사이즈가 동시에
변경될 때의 학습 횟수 별 테스트 정확도

다음으로 (그림 5)에서는 초기 학습 율에만 변화를 주었을 때 테스트 정확도의 변화 추이를 살펴볼 수 있다. 먼저 CASE5의 경우에는 다른 CASE들 보다 월등히 초기 학습율이 낮은 경우이다. 이 경우 다른 CASE와 비교해서 절반 이하의 테스트 정확도를 보여주고 있다. CASE8은 초기 학습 율이 지나치게 큰 관계로 전혀 학습이 진행되지 않는 모습이다. 이와 같은 현상은 (그림 2)에서 학습 정확도를 확인할 때도 나타난 증상이다. 나머지 CASE1, CASE4, CASE7을 비교해 보면 최종 테스트 정확도의 한계점은 대략 유사하나 한계점에 다다르는데 필요한 학습 반복 횟수가 CASE7의 경우 CASE1, CASE4에 비해 많이 필요했다. 즉, CASE1, CASE4가 가지는 학습 율인 0.01이나 0.04값이 최적의 값으로 보인다. CASE1 과 CASE4의 경우 학습 반복 횟수에 소요되는 시간은 (그림 2)에서 확인할 수 있듯이 CASE4에서 100회 반복 횟수 당 5.6초의 시간이 더 필요하다. 따라서 시간적인 비용과 학습 반복 횟수 당 테스트 정확도 증가율을 비교해 자신의 환경에



(그림 7) CASE에 따른 평균 메모리 사용량

적합한 매개변수 값을 설정해야 한다.

또한 초기 학습 율과 배치 사이즈 크기를 상호보완적으로 변경하려면 값을 어떻게 바꿔야 하는지 확인해볼 수 있다. (그림 6)은 CASE1을 기준으로 배치사이즈 값을 1/2로 줄였을 때 초기 학습 율을 CASE6과 CASE9에 각각 2배, 1/2배로 설정하여 학습 정확도 변화 추이를 보이고 있다. 그래프 곡선을 보면 배치 사이즈 크기를 기준인 CASE1의 1/2배로 설정하고 초기 학습 율은 CASE1의 2배로 설정한 CASE6의 경우가 CASE1의 그래프 곡선을 거의 유사하게 따라갔다. 배치 사이즈 크기와 초기 학습 율을 모두 CASE1의 1/2로 설정한 CASE9는 학습 반복이 400회까지 진행되는 동안 CASE1, CASE6과 비교하여 많이 낮은 테스트 정확도를 보여주었다.

마지막으로 (그림 7)에 매개변수들에 따른 평균 메모리 사용량을 나타내었다. CASE1,4,5,7,8을 통해 배치 사이즈 크기만이 메모리 사용량에 직접적인 영향을 주는 것을 확인할 수 있다. CASE1 과 CASE3을 통해 배치 사이즈가 2배 커지면 메모리 사용량은 126%만큼 커지고 CASE1 과 CASE2를 통해 배치 사이즈가 1/2배가 되면 메모리 사용량은 84%만큼 작아진 것을 확인할 수 있다. 따라서 만약 대규모의 딥 러닝 환경에서 메모리가 부족한 상황에 직면한다면 배치사이즈를 줄여보는 것을 추천한다.

4. 결론 및 향후과제

CNN구조를 가지는 영상인식 딥 러닝 환경에는 유저가 직접 값을 조작해야하는 많은 매개변수들이 존재한다. 아직도 많은 변수나 공식들이 정확히 분석되지 못한 채 일방적으로 실험 결과 성능이 좋게 나온 다는 이유로 모두가 획일화된 값과 공식을 사용하는 경우가 많다. 본 논문에서는 실험을 통해 배치 사이즈와 학습율이 구체적으로 어떤 형태로 성능에 영향을 미치는지 알아보았다. 학습 성능을 유지하면서 속도를 빠르게 하기 위해선 배치 사이즈를 줄이고 초기 학습율은 그 비율만큼 늘이는 것이 이상적이라는 것을 여러 방식의 실험을 통해 보여 주었다. 하지만 지나치게 초기 학습율을 높이면 일정 학습 횟수 이상부터는 테스트 정확도가 진동하거나 전혀 학습이 되지 않는 경우도 보였다. 또한 본 실험에서 사용된 LeNet에서는 적은 양의 메모리만 사용되었지만 현재 시중의 하드웨어 사양은 아무리 좋은 GPU를 사용한다 하더라도 다른 모든 딥 러닝 구조를 원활히 빠르게 처리 하는데 한계가 있다. 따라서 딥 러닝을 연구한다면 반드시 메모리 관리 측면도 고려해야한다. 본 논문에서는 배치 사이즈 변경으로 메모리 사용량의 변화를 유도할 수 있는 방법을 제시하였다.

앞으로의 연구 방향으로는 SGD방식[6]에서 학습이 진행될 때 학습 율을 감소시키는 공식들에 대해 연구해볼 것이다. 상황에 따라 학습 율을 감소시켜 나가는 공식은 여러 가지 방식이 제안된바가 있지만 어떤 과정을 통해 성능에 영향을 주는지는 정확히 알려진 바가 없다. 또한 본 논문에서는 MNIST만을 데이터 셋으로 사용하였지만 추후의 연구에서는 CIFAR-10[7]등 다른 대규모 영상 데이터 셋도 함께 사용하여 비교 실험을 진행할 것이다.

참고문헌

[1] Dan Cires¸an, Ueli Meier and Jurgen Schmidhuber "Multi-column Deep Neural Networks for Image Classification" in Proc. CVPR, 2012

[2] Alex Krizhevsky, Ilya Sutskever and Geoffrey E. Hinton "ImageNet Classification with Deep Convolutional Neural Networks" in Proc. NIPS, 2012

[3] Sanjeev Arora, Aditya Bhaskara, Rong Ge and Tengyu Ma “Provable bounds for learning some deep representations” CoRR, 2013

[4] Misha Denil, Babak Shakibi, Laurent Dinh, Marc'Aurelio Ranzato and Nando de Freitas "Predicting Parameters in Deep Learning" In Advances in Neural Information Processing Systems, pp. 2148 - 2156, 2013

[5] Yann LeCun, Leon Bottou, Yoshua Bengio and Patrick Haffner "Gradient-based learning applied to document recognition" in Proc. IEEE, vol.86, no.11, pp. 2278-2324, 1998

[6] L´eon Bottou “Stochastic Gradient Descent Tricks” In G. Montavon, G. Orr, and K.-R. Muller, editors Neural Networks: Tricks of the Trade, volume 7700 of Lecture Notes in Computer Science, pp. 421 - 436. Springer Berlin Heidelberg, 2012

[7] Alex Krizhevsky "Learning multiple layers of features from Tiny Images" Tech. Rep, University of Toronto, 2009