딥러닝을 이용한 실시간 인코딩 효율 최적화

콘퍼런스 발표 | 여욱형 jaden.y@kakaocorp.com

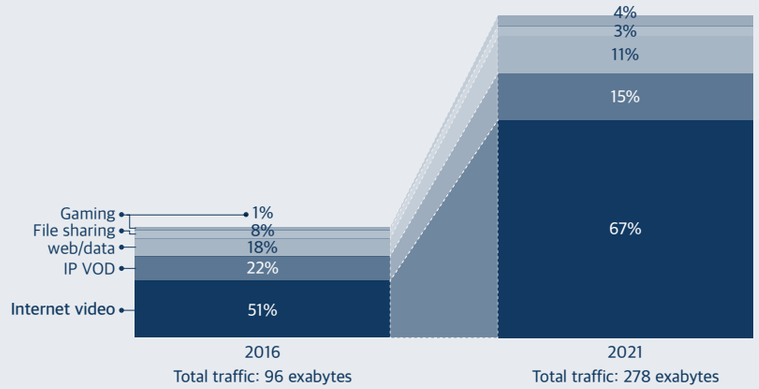
글 | 정소영 songr64@naver.com Swarm Intelligence를 이용한 Global Optimization Algorithm을 개발했고, 현재는 협업 필터링(Collaborative Filtering)을 바탕으로 추천시스템에서 Similarity Measure를 개발하는 연구를 진행 중입니다. 또 기계학습을 이용해 편미분 방정식을 푸는 연구를 진행하고 있습니다. 대학원 과정 동안 응용수학을 전공했고, 학부 때 조선공학을 전공했기 때문에 공학 기반의 수학 문제 해결에 관심이 많습니다.

동영상 분야에서 품질은 유지하면서 용량을 줄일 수 있는 방법을 딥러닝으로 개발한 내용을 다루었다. 단순히 동영상의 이미지를 학습시키는 것이 아니라, 인코딩할 때 동영상의 특성이 미치는 영향을 바탕으로 그 특성을 학습시키는 방법을 채택했다. 다시 말해 동영상을 인코딩할 때 알고리즘에서 사용되는 특징들을 분석해서 효율성을 높이는 것이다.

비트레이트(bitrate) 배분과 트래픽(traffic) 관리

하루에도 수많은 종류의 영상이 송출되고 있다. 카카오 TV 라이브 플랫폼도 그 영상을 받아서 인코딩하고, 스트리밍 서비스를 제공한다. 동영상 서비스를 제공할 때 가장 고민되는 부분은 무엇일까? 바로 트래픽이다. 트래픽은 비용과 직결되기 때문에 고화질 동영상을 무한정 서비스할 수 없으므로 트래픽과 화질을 고려하여 효율적으로 관리해야 한다. 전 세계적으로 동영상 트래픽이 폭발적으로 증가하는 추세이기 때문에 트래픽 관리는 더욱 중요한 문제가 되고 있다.

[그림 1] 글로벌 동영상 트래픽 증가 예측*1



이 문제를 해결하기 위해 카카오에서는 먼저 압축 효율이 가장 좋은 최신 인코더를 도입해서 쓰는 접근 방식을 생각해보았다고 한다. 하지만 이 코덱(codec)을 적용하기 위해서는 현재의 인프라 시스템을 교체해야 하는데, 그러기에는 리소스 비용이 많이 필요하기도 하고, 실제로 적용해보면 압축 효율성이 예상보다 떨어진다. 그래서 카카오에서는 현재의 인프라로 트래픽 효율을 최적화하기 위해서 이 프로젝트를 시작하게 되었다.

동영상이 업로드되거나 라이브로 송출되면 가장 우선적으로 하는 일이 비디오 인코딩이다. 목적은 화질을 좋게 하면서 용량은 줄이는 것이다. 여러 가지 다양한 요소가 있겠지만, 직접적으로 맞닿아 있는 문제는 이것을 위해 실질적으로 비트레이트를 얼마나 할당하느냐의 문제이다. 사실 그동안의 경험으로 해상도별로 적정 비트레이트가 어느 정도 정해져 있다. 대부분의 서비스 업체에서 이런 방식으로 인코딩을 하고 있지만 이런 인코딩 방식은 효율적이지 않다. 움직임이 없는 동영상은 움직임이 많은 동영상에 비해 적은 비트레이트로 동일한 화질을 나타낼 수 있기 때문이다. 즉, 영상마다 적절한 비트레이트를 찾으면 트래픽을 효과적으로 사용할 수 있다는

*1 참고 | Cisco

얘기가 된다. 그렇다면 적절한 비트레이트를 어떻게 찾을 수 있을까?

[그림 2] 구간별 동영상의 특성을 반영한 비트레이트 설정

단순하게 모든 필요 비트레이트와 해상도의 조합으로 인코딩을 해보면 된다. 화질 지수와 사이즈가 적절한 것을 찾으면 되는데, 이것은 많은 리소스가 필요해 현실적으로는 불가능하다. 그래서 카카오에서는 영상을 카테고리별로 분류하여 테스트했다. 애니메이션, 스포츠, 게임, 토크쇼를 분리해놓고 비트레이트를 다르게 설정해보았다. 하지만 테스트를 진행해보니 같은 영상이라도 구간별로 필요한 비트레이트가 다르다는 것을 알 수 있었다. 그래서 영상의 일정 구간마다 비트레이트에 영향을 줄 수 있는 요소를 분석하며 동영상을 더 세밀하게 분석했다.

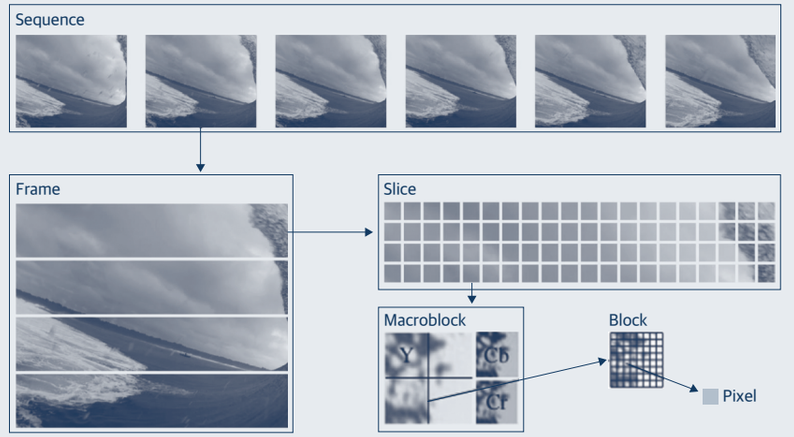
인코딩은 어떤 패턴이 있는 것이 아니라 모든 요소마다 연관이 있기 때문에 수학적으로 풀어내기가 어렵고, 모호한 부분이 존재한다. 그래서 영상의 특성과 최적 파라미터(parameter)와의 관계를 딥러닝으로 분석해보기로 했다. 동영상 인코딩의 효율 최적화는 넷플릭스(Netflix), 유튜브(Youtube) 등에서도 적용하고 있는 방식이다. 넷플릭스는 3년 동안의 연구개발 끝에 타이틀 단위 인코딩(per-title encoding)을 통해 전체 트래픽의 30%를 감소시켰다. 유튜브의 경우 인공신경망을 이용한 퀄리티 타겟(quality targets)을 개발 중이다. 이러한 시도들은 동영상에 고정적 비트레이트를 배분하는 것이 아니라 특징에 따라 유동적인 비트레이트를 배분하겠다는 시도이다.

인코딩 효율 최적화를 위한 준비

카카오에서는 어떤 패턴을 갖는 비디오의 특성들에 대해서 적절한 비트레이트의 설정값을 딥러닝으로 학습시키려 한다. 지도학습(supervised learning) 중 분류(classification) 방식이다. 비디오 특성 분석 데이터와 인코딩 파라미터가 입력(input) 데이터가 되고, 딥러닝 학습 시스템을 구동하여 비디오 특성 분류 프로그램이 출력(output)으로 나온다.

비디오 특성 분석 데이터란 영상 내 비트레이트에 영향을 주는 요소이다. 영상의 구조 단위는 크게 시퀀스(sequence), 그 안에 프레임(frame), 그 안에 다시 슬라이스(slice) 레벨로 나눌 수 있다. 슬라이스 내부는 매크로블록(Macro Block)으로 이루어져 있다. 매크로블록은 비디오 인코딩과 디코딩 연산의 최소 단위이며 주로 16×16픽셀이다. 경우에 따라서는 8×8, 4×4픽셀 등으로 더 작게 나누는 인코딩 프로세스가 수행되기도 한다.

[그림 3] 동영상 구조 단위 레벨 구분



매크로블록은 크게 세 가지 타입이 있다. 인트라 매크로블록(Intra Macro Block)은 하나의 프레임 내에서만 연관 관계를 찾아서 인코딩할 때 사용된다. 인터 매크로블록(Inter Macro Block)은 프레임 간 참조 관계를 통해서 인코딩을 할 때 사용된다. 스킵 매크로블록(Skip Macro Block)은 인터 매크로블록의 일종인데, 이전 프레임으로부터 비슷한 부분을 찾아서 데이터를 가져오는 점에서 비슷하지만, 현재의 프레임과 상관없이 이전의 프레임만을 가져와서 프레임을 구성한다는 점이 다르다. 인터 매크로블록의 경우 현재의 프레임과 이전 프레임의 상태를 분석해서 합친다.

[그림 4] 영상 속 하나의 프레임을 구성하는 요소

Type: I (▢), P (▢), Skip (▢)

[그림 4]를 보면 하나의 프레임을 구성하는 요소를 확인할 수 있다. I, P, Skip(공백)으로 나누어져 있다. I와 P의 비율은 높고 Skip의 비율이 낮은 프레임이 있지만 I와 P의 비율은 낮고, Skip의 비율이 높은 프레임도 있다. 이렇게 다른 특성을 가진 동영상은 비트레이트 배분을 달리해야 한다.

비트레이트 배분에 영향을 주는 또 다른 요소가 있다. 움직임을 나타내는 요소, 모션 벡터(Motion Vector)이다. 현재 인코딩, 디코딩하려는 매크로블록이 이전 프레임의 어느 부분과 가장 일치하는지를 나타내는 요소로, 프레임 간 움직임을 추정할 수 있는 요소다. 다음 [그림 5]의 화살표가 바로 모션 벡터다. 모션 벡터량은 당연히 비트레이트 배분에 영향을 주게 된다.

[그림 5] 움직임을 나타내는 요소(Motion Vector)

최종적으로 영상을 분석할 때 움직임에 대한 정도를 나타내는 수치(움직임의 양, 강도), 프레임을 구성하는 블록의 타입별 분포, 프레임을 구성하는 블록의 파티션(partition) 사이즈 분포, 세그먼트 내의 복잡도 수치, 비트(bit) 배분율, 프레임 레이트(frame rate), 해상도, 그 외 영상과 관련된 전반적인 수치를 더하여 20여 가지 학습 데이터로 비디오 특성 분석 데이터를 구성했다.

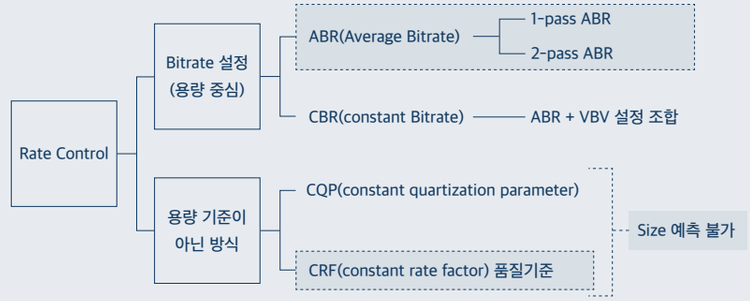
다음 인코딩 파라미터를 살펴보자. 파라미터는 두 가지로 구성했는데, CRF(Constant Rate Factor)와 Max rate를 택했다. CRF는 레이트 제어(rate control)의 한 종류이고, Max rate는 최대 비트레이트를 넘지 않도록 제한하는 수치이다.

레이트 제어는 용량을 고려하는지, 혹은 용량을 고려하지 않고 품질을 고려하는지를 기준으로 크게 두 가지 방식으로 나뉜다. 용량을 고려하는 가장 대표적인 방법으로는 ABR(Average Bitrate, 평균 비트레이트)이 가장 많이 쓰인다. ABR의 경우 미리 정해진 평균값을 중심으로 설정되는 비트레이트를 맞추는 방식이다. 레이트 제어 동작은 기본적으로 복잡도에 따라서 비트레이트를 배분하는데, ABR의 경우 복잡도를 미리 알 수 없는 것이 단점이다. 초기에 일정 비트레이트 배분을 해야 하지만 복잡도에 따라서 적극적으로 대응을 못 한다. 그래서 ABR 방식으로 2번 인코딩하는 2-Pass ABR도 존재하지만 충분히 효율적이지 못하다.

반면 CRF 방식은 용량의 제한 없이 품질을 맞추기 위해서 인코딩하는 방식이다. 현재 프레임의 복잡도에 따라서 비트 배분을 한다. 정해진 퀄리티 레벨(quality level)을 기반으로 인코딩을 하고 화질의 불균형도 발생하지 않는다. CRF는 덜 복잡한 부분은 비트 할당을 적게 하기 때문에 낭비되는 비트가 적을 수 있다.

하지만 CRF 대신 ABR이 많이 쓰이는 가장 큰 이유는 CRF의 인코딩 결과 사이즈를 예측할 수 없기 때문이다. 시스템을 운영하면서 트래픽을 예측할 수 없다는 것은 정말 치명적인 단점이다. 하지만 이러한 단점은 실험 데이터를 모아서 파악하면 극복할 수 있다. CRF 레벨은 0에서 51까지의 수치로 나뉘며, 레벨별로 비트레이트 설정 수치가 어느 정도 정해져 있다. 문제는 영상마다 CRF 레벨이 다른데, 복잡도가 비슷한 영상은 CRF 레벨과 비트레이트 곡선이 상당히 유사한 부분이 많다는 것이다. 카카오에서는 구간별로 동적인 비트레이트 설정을 할당하기 위해서 CRF를 선택했다. 한편 Max Rate의 경우 순간적으로 비트 배분이 높아지는 경우가 발생할 때를 대비하여 제한해야 하고, 이는 화질, 버퍼링 등에 상당한 영향을 미친다.

[그림 6] 레이트 제어(Rate Control) 분류 및 특성

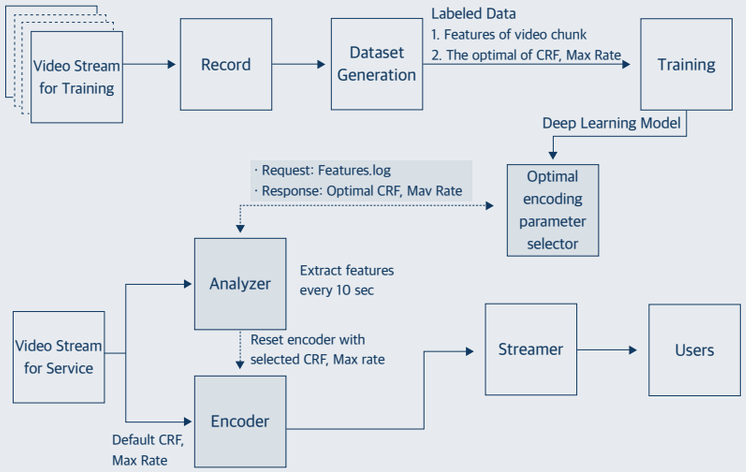


인코딩 효율 최적화의 구현

인코딩을 효율적으로 최적화하는 방법은 다음과 같다. 영상을 구간별로 잘라서 모든 CRF 레벨과 모든 Max rate 수치의 조합으로 인코딩을 실시한다. 그리고 품질 측정을 한다. 적정한 화질을 유지하면서 사이즈를 가장 작게 할 수 있는 파라미터를 고른다. 영상의 특징과 효율적인 파라미터를 매칭하고, 딥러닝을 이용하여 학습을 시킨다.

학습 결과로는 세그먼트 기준 총 30여만 개의 학습 데이터 세트를 생성해서 실험 정확도(test accuracy)를 계산했을 때 73% 수치가 나왔다. 100번 중에 73번은 학습시킨 의도대로 판별이 되었다는 뜻이다. 나머지 27번도 다른 값으로 판별되기는 하였지만, 원했던 값과 근사한 답을 주었고, 실제 서비스에 적용해도 무방한 수준이었다. 물론 알고리즘에 약간의 오류가 있어서 전혀 틀린 답을 뱉어낼 때도 있었지만 전체적으로 문제는 없는 수준이었다. 또 이전 설정값과 너무 큰 차이가 나면 이전 설정값을 사용해서 시스템의 안정성을 높였다. 이러한 방식을 적용하면 전체 트래픽의 약 20%를 줄일 수 있을 것으로 결과가 나왔다.

[그림 7] 시스템 아키텍처(System Architecture)



전체 시스템의 주요 아키텍처(architecture)는 다음과 같다([그림 7] 참고). 전체 시스템 아키텍처의 위쪽 레이어(layer)는 학습 시스템이고, 밑의 레이어는 기존에 갖고 있던 시스템이다. 애널라이저(analyzer)와 최적의 인코딩 파라미터 셀렉터(optimal encoding parameter selector)를 붙여서 시스템을 구성하였다. 애널라이저는 10초 간격으로 특징(feature)을 계속 뽑아내어(10초로 고정되어 있지 않음) 최적화된 파라미터 셀렉터에 파라미터를 질문한다. 그리고 파라미터를 받으면 해당 비트레이트의 배분에 맞게 인코딩을 한다.

이 시스템의 주요 목표는 데이터의 낭비를 줄이면서 퀄리티를 높여 데이터를 더 쓸 수 있게 하는 것이다. 고화질 서비스를 제공해야 할 때 비트레이트를 더 배분할 수 있도록 하고, 트래픽을 줄이면서 버퍼링 등의 문제를 예방해 스트리밍 서비스의 품질을 높이기 위해 노력했다. 또 원본 영상의 코덱 종류와 무관하게 동작하는 것을 목표로 했고, 지금은 H264로만 가정이 되어 시스템이 구성되어 있지만, 타깃 영상의 코덱 종류를 다양하게 지원하는 것도 염두에 두었다. 향후 H265 등의 고효율 압축 코덱도 서비스하려고 계획 중이다.

실제 일부 서비스에 적용해본 결과 화질은 -0.64% 정도로 눈으로 확인할 수 없을 만큼의 감소는 있었지만, 전체 용량은 약 20% 이상 절감 되었다. 조금 더 정적인 동영상에 적용을 해보니 화질은 겨우 1% 정도 나빠진 반면 용량은 거의 반으로 줄일 수 있었다.

이 프로젝트는 비디오의 특성과 비트레이트/화질과의 연관성을 입증하고, 비디오 특성을 분석한 데이터를 축적했다는 데 의의가 있다. 앞으로 정확도를 높이기 위해 더 많은 데이터를 확보해 분석할 수 있는 데이터를 추가하고, 모바일과 TV, 컴퓨터 등 기기별 최적화도 고려하며 좀 더 유연한 구조로 변경을 고려할 수 있도록 할 예정이다. 현재 카카오는 서비스 적용을 준비 중이며 연내 시스템 연동을 완료하는 것을 목표로 하고 있다.

〈딥러닝을 이용한 실시간 인코딩 효율 최적화〉 브런치로 연결되는 QR 코드입니다.