Focal Loss for Dense Object Detection

Two-stage-method는 바운딩 박스와 클래스를 순차적으로 찾는 방법입니다. 여기서 바운딩 박스를 찾는 것을 localization이라고 하고 클래스 찾는 것을 classification이라고 합니다.
localization은 selective search나 RPN 등을 통해 후보 박스를 제안하는 것을 말합니다.
이러한 후보 박스를 만드는 과정에서 대부분의 background samples가 필터링됩니다.

One-stage-method는 바운딩 박스와 클래스를 한 번에 찾는 방법입니다. 속도는 빠르지만 정확도가 낮은 방법입니다. YOLO를 예시로 말하면 각 셀마다 B개의 박스를 제안하기는 하지만 selective search나 RPN처럼 background samples가 필터링되지는 않습니다. 따라서 one-stage-method에서는 foreground-background class imbalance가 극명하게 발생합니다. imbalance가 발생된 상태에서 일반적인 Loss Function을 사용하면 객체 검출 정확도가 떨어지게 됩니다.

일반적으로 사용되는 Cross Entropy를 알아 보겠습니다.

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases} \quad (1)$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases} \quad (2)$$

y : ground-truth class $y \in \{\pm 1\}$
p : model's estimated probability $p \in [0, 1]$

P는 **foreground에 해당하는 class의 probability**를 의미합니다.
만약 P의 값이 1이면 CE는 0이 됩니다. 그렇지 않고 P의 값이 0이면 CE는 ∞가 됩니다.

Foreground 케이스에 대해 잘 예측해서 P가 높은 경우(1과 가까운 경우) 패널티는 없고, 잘못 예측해서 P가 낮은 경우(0과 가까운 경우) 패널티는 크게 됩니다.

ex) –log(1)=0 , -log(0)=∞

Background 케이스에 대해 잘 예측해서 P가 낮은 경우(0과 가까운 경우) 패널티는 없고,
잘못 예측해서 P가 높은 경우(1과 가까운 경우) 패널티는 크게 됩니다.

ex) –log(1-0)=0, -log(1-1)=∞

ex) Foreground case:  , Background case:  <강아지 class인 상황에서 case 예시>

여기까지만 보면 큰 문제는 없어 보입니다.

이번엔 예시로 foreground 케이스를 0.95로 예측하고 background 케이스를 0.05로 예측했다고 가정하겠습니다. 그럼 CE(foreground) = -log(0.95) = 0.05가 될 것이고 CE(background) = - log(1-0.05) = 0.05가 될 것입니다.
여기서 문제는 background 케이스의 수가 훨씬 많다는 점입니다.
이 말은 (foreground 케이스에 대한 누적 loss)와 (background 케이스에 대한 누적 loss)를 비교하면 후자가 더 높을 수 밖에 없다는 뜻입니다.
결국 background에 대하여 학습이 훨씬 많이 될 것입니다.

정리하면 one-stage-method는 background samples가 필터링되지 않기 때문에 background케이스가 너무 많고, 일반적인 CE를 사용하면 케이스 비율을 고려하지 않으므로 background만 많이 학습하는 문제가 생긴다는 것입니다.

[우리가 구하고자 하는 문제는 객체가 있는 경우(foreground)입니다.
그러므로 앞으로 foreground는 positive라고 표현하겠습니다.
반대로 관심이 적은 background는 negative라고 표현하겠습니다.
다만 관심이 적다고 해도 성능은 보장되어야 합니다!]

이번에 설명할 CE는 **Balanced Cross Entropy**입니다.
이것은 단순히 positive 문제에 대해 더 학습하고자 만들어진 loss function입니다.

$$\mathrm{CE}(p_t) = -\alpha_t \log(p_t).$$

$$\alpha \in [0, 1]$$

앞에 a라는 가중치가 포함되면 어떤 결과가 발생하는지 예시를 보이겠습니다.

positive 케이스가 0.5, negative 케이스가 0.5라고 가정하겠습니다. 그리고 가중치 a를 0.75만큼 부여하겠습니다.

그럼 CE(positive)=-0.75*log(0.5)=0.22가 될 것이고 CE(negative)=-(1-0.75)*log(1-0.5)=0.07이 됩니다. 즉 positive에 비해 loss값이 현저히 낮으므로 negative loss가 누적된다고 하더라도 낮은 값을 갖게 될 것입니다.

하지만 이와 같은 방법에도 문제점이 있습니다.

positive 케이스가 0.98, negative 케이스가 0.2라고 가정하겠습니다.

그럼 CE(positive)=-0.75*log(0.98)=0.038이 될 것이고 CE(negative)=-(1-0.75)*log(1-0.2)=0.024가 될 것입니다.

이런 경우 negative 케이스에 대한 학습이 더 필요하지만 positive loss가 더 크므로 negative쪽으로는 학습이 잘 안 될 수밖에 없습니다.

즉 positive문제는 어느정도 잘 푸니 easy해졌고 negative문제는 아직 잘 못 풀어서 hard한 상태인데 이 loss function이 easy, hard까지 고려하지는 못 한다는 의미입니다.

[easy positive: positive케이스에 대해 학습이 잘 된 경우 (positive case의 p가 높음, p_t가 높음)
easy negative: negative케이스에 대해 학습이 잘 된 경우 (negative case의 p가 낮음, p_t가 높음)
hard positive: positive케이스에 대해 학습이 잘 안 된 경우 (positive case의 p가 낮음, p_t가 낮음)
hard negative: negative케이스에 대해 학습이 잘 안 된 경우 (negative case의 p가 높음, p_t가 낮음)]

사실 negative case-p = 0.2도 학습이 잘 됐다고 볼 수 있지만 positive case에 비해 상대적으로 학습이 잘 안 됐으므로 hard negative라고 간주함

정리하면 one-stage-method에서 일반적인 CE를 사용하면 negative에 대해 학습이 많이 되기 때문에 positive에 큰 가중치를 주는 Balanced CE가 제시되었고, 이 CE를 사용하려고 하였으나 easy, hard를 고려하지 못해서 hard negative에 대한 학습이 잘 안 되었다는 의미입니다.

이번에 설명할 loss function는 **Focal Loss**입니다.
이것은 easy와 hard한 케이스를 고려하는 loss function입니다.

$$\mathrm{FL}(p_\mathrm{t}) = -(1-p_\mathrm{t})^\gamma \log(p_\mathrm{t}).$$

focusing parameter: $\gamma \geq 0$

moderating factor: $(1-p_\mathrm{t})^\gamma$

마찬가지로 positive 케이스가 0.98, negative 케이스가 0.2라고 가정하겠습니다. r은 2를 부여하겠습니다.

그럼
CE(positive)=-(1-0.98)^2*log(0.98) = 3.50956972e-7

CE(negative)=-(1-(1-0.2))^2*log(1-0.2) = -(0.2)^2*log(0.8) = 0.003

가 되면서 hard negative문제에 대한 학습을 더 할 수 있습니다!

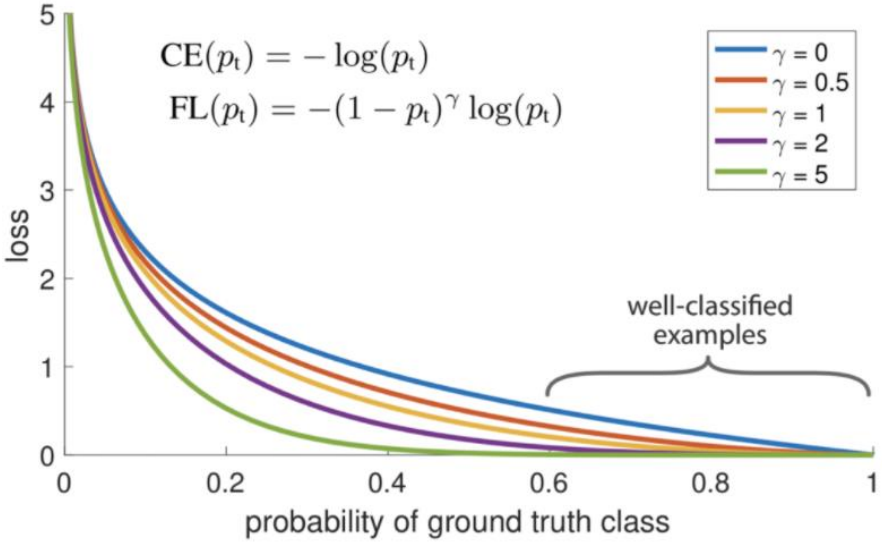
focal loss의 특징은 moderating factor을 사용하는 것인데
hard할수록(p_t가 낮을수록) 값이 커지고
easy할수록(p_t가 높을수록) 값이 작아집니다.

이 moderating factor은 가중치 역할을 하게 되는데
hard할수록 가중치를 많이 줘서 loss를 크게 하고
easy할수록 가중치를 적게 줘서 loss를 작게 한다는 의미입니다.

다음은 **CE와 FL을 비교하는 그래프**입니다.
x축은 p_t, y축은 loss라고 생각해봅시다.

hard할수록(p_t가 낮을수록) CE와 FL의 차이는 작고 (적게 떨어뜨림)
easy할수록(p_t가 높을수록) CE와 FL의 차이는 큰 것을 확인할 수 있습니다. (많이 떨어뜨림)

참고로 r이 클수록 loss가 크게 줄어듭니다. (차이가 큽니다)



1page 설명에서 CE는 easy하면(p_t가 높으면) 패널티는 없고, hard하면(p_t가 낮으면) 패널티를 많이 부여한다고 언급했습니다.

FL은 easy하면(p_t가 높으면) loss를 훨씬 낮추는 보상을 부여하고, hard하면(p_t가 낮으면) 패널티를 많이 부여한다고 생각하면 됩니다.

추가적으로 전체적인 Loss 값을 조절하는 a값 또한 논문에서 사용되어 a,r 값을 조절하여 어떤 값이 좋은 성능을 가졌는지 보여주었습니다. 식은 아래와 같고 논문에서는 a=0.25, r=2를 최종적으로 사용하였습니다.

$$\mathrm{FL}(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t).$$

<u>본 논문을 정리하면 다음과 같습니다.</u>

1. one-stage-method는 background samples가 필터링되지 않기 때문에 background(negative)케이스가 너무 많다.

2. 일반적인 CE를 사용하면 케이스 비율을 고려하지 않으므로 negative에 대해 학습이 많이 된다.
3. positive에 큰 가중치를 주는 Balanced CE가 제시되었고, 이 CE를 사용하려고 하였으나 easy, hard를 고려하지 못해서 hard negative에 대한 학습이 잘 안 되었다. (positive는 학습 잘 됨)
4. positive에 가중치를 더 부여하는 개념보다는 easy, hard에 따라 가중치가 달라지는 Focal Loss가 제안되었고 이 loss를 사용하였을 때 hard한 케이스들(hard positive, hard negative)까지 학습이 잘 되었다.