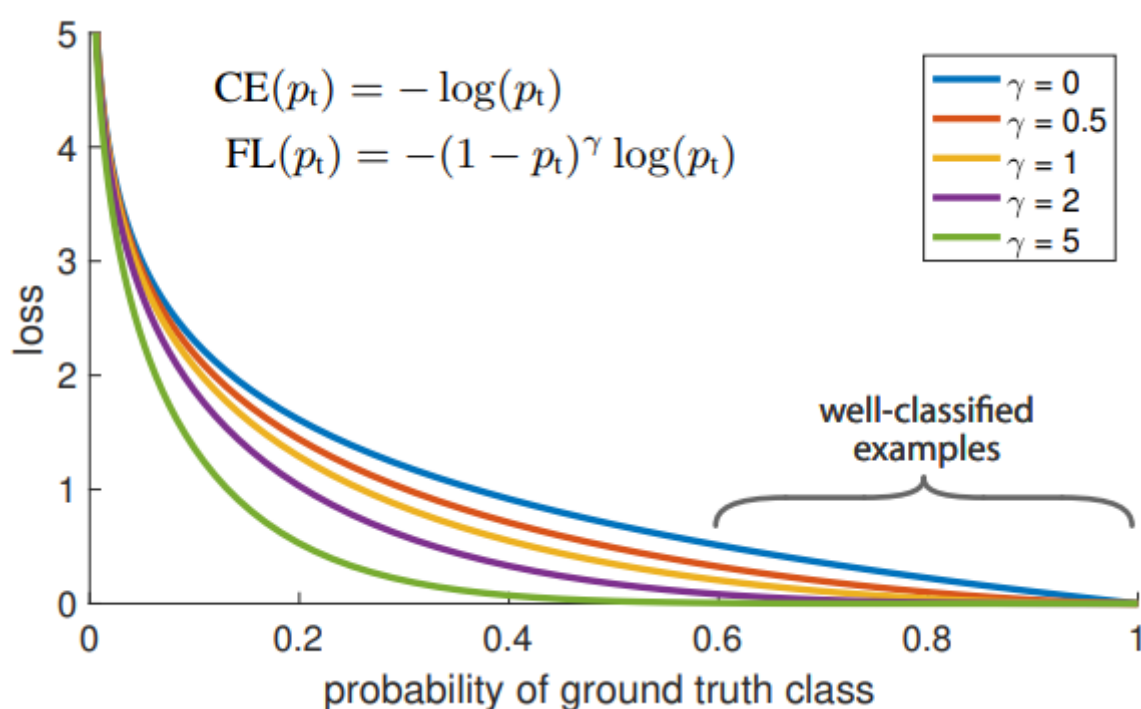


RetinaNet

Focal Loss

- **Focal Loss**는 **one-stage object detector**의 극단적인 **class 불균형 문제**(예시 -> object :background = 1:1000)를 해결하기 위해 **desing한 loss function**
- 먼저 **binary classification**(class인지 배경인지 classify)에 사용되는 **Cross Entropy Error** 설명을 함



파란색 그래프가 CE // 출처 : 원문

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

Cross Entropy Error // 출처 : 원문

- 편의성을 위해 본문에서 P_t 라는 변수를 정의
- P_t : 해당 class가 존재할 확률

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

P_t 의 정의 // 출처 : 원문

- 이제 CE를 다음과 같이 표기 가능

$$CE(p, y) = CE(p_t) = -\log(p_t).$$

CE의 단순한 표기

- Cross Entropy의 특징으로 P_t 가 0.5보다 커도(= box에 물체가 존재할 확률이 50%가 넘어도) loss 값이 꽤 있다는 점

- 이러한 특징 때문에 easy examples(P_t 가 0.5보다 큰 examples)이 많이 있다면, 이러한 loss들이 쌓이고 쌓여서 물체를 검출하지 못하는 나쁜 방향으로 학습이 될 수 있음

3.1. Balanced Cross Entropy

- α -balanced CE loss : class 불균형을 해결하기 위해 일반적으로 CE에 weighting factor α 를 적용

$$CE(p_t) = -\alpha_t \log(p_t).$$

α -balanced CE loss -> class 불균형 문제 해결 시도 // 출처 : 원문

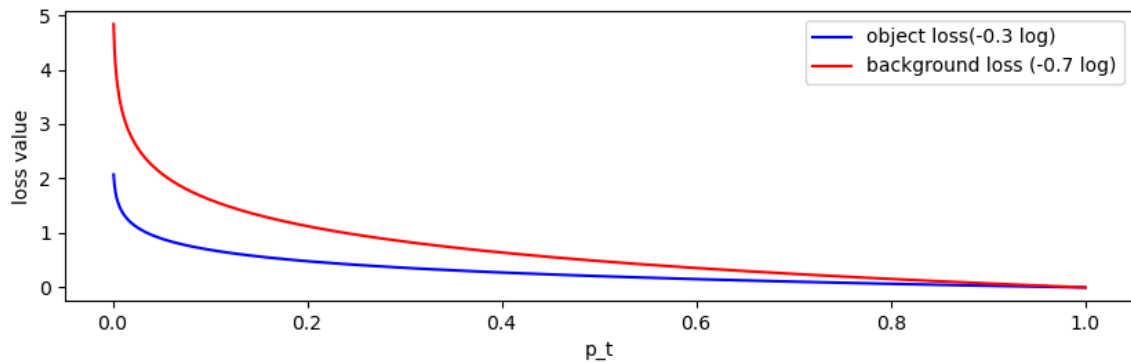
- 검출할 클래스 : α 값을 0~1 사이로 적용

- 배경 : $1-\alpha$ 를 적용

- 이는 Focal Loss의 기반이 되는 수식

- 만약 $\alpha = 0.3$ 로 설정하면, 아래와 같은 그래프가 나옴

- 이렇게 object에 대한 loss는 작게, background에 대한 loss는 크게 설정할 수 있음



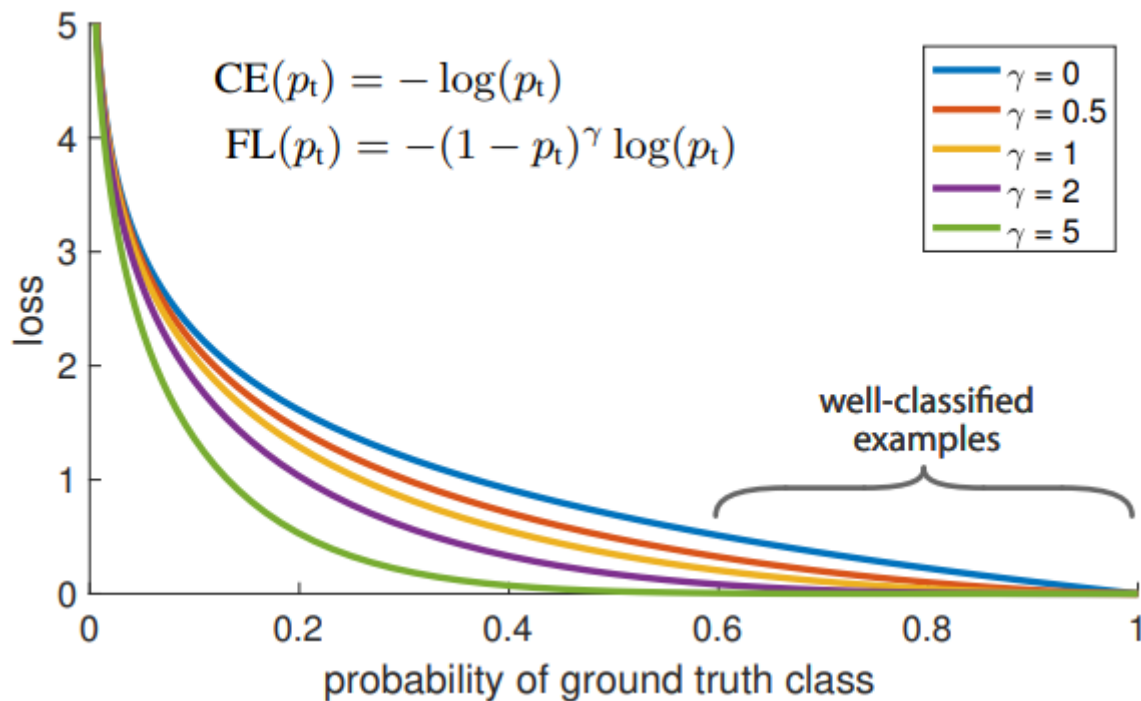
파란색 그래프 : object loss, 빨간색 그래프 : background loss // 출처 : 제가 python으로 직접 plot 했습니다 ㅎㅎ

3.2. Focal Loss Definition

- α -balanced CE loss의 장점 : positive/negative example의 차별성을 표현 가능
- α -balanced CE loss의 단점 : easy($p_t > 0.5$)/hard example($p_t < 0.5$)의 차별성을 표현 불가능
- Focal Loss : easy example에 대한 가중치를 줄이고, hard negative example의 학습에 초점을 맞추도록 α -balanced CE loss를 수정
- * easy example에 대한 가중치를 줄인다 : easy example에 대한 loss 값을 줄임
- * hard example의 학습에 초점을 맞춘다 : hard example에 대한 loss 값을 키움

$$FL(p_t) = -(1 - p_t)^\gamma \log(p_t). \quad (4)$$

Focal Loss 수식



초록색 그래프와 같이 감마 값이 커질수록 easy example와 hard example의 loss 차이가 분명해짐을 알 수 있음 // 출처 : 원문

- CE에 modulating factor $(1 - p_t)^\gamma$ 를 추가 (focusing parameter $\gamma \geq 0$)
- focal loss는 두 가지 특징이 있음
 - * 첫번째 특징 : p_t 값이 작을 때, modulating factor는 거의 1에 근접하며 loss 값이 커짐 & p_t 값이 클 때 modulating factor는 0에 근접하며 well-classified examples($p_t > 0.6$)의 loss 값이 작아짐
 - * 두번째 특징 : 감마 값이 커질수록 modulating factor의 영향이 커짐 (본 문에서는 감마 = 2가 가장 좋은 성능을 냈다고 함)
- 즉, modulating factor는 easy example의 loss 값을 더욱 더 작게 만들

< α -balanced variant of the focal loss>

- 위에서 보인 focal loss는 α -balanced 를 적용하지 않았음
- 아래의 수식은 α -balanced 를 적용한 focal loss
- > easy/hard example의 차별성뿐만 아니라 positive/negative example의 차별성도 표현 가능

$$\text{FL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t). \quad (5)$$

α -balanced 를 적용한 focal loss // 출처 : 원문

- **α -balanced 를 적용한 focal loss**가 적용하지 않은 focal loss보다 **성능이 더 좋았음**
- 아래의 부록 A에서 다양한 형태의 focal loss의 실험 결과가 적혀있음