<u>Focal Loss for Dense Object Detection (RerinaNet):</u>

This paper mainly talks about solving class imbalance and hard examples results in single shot detectors using the proposed Focal Loss. They used Feature Pyramid Network (FPN) for feature extraction for detection of objects in different scales and sizes.

Class imbalance

In singe shot detectors a large set of candidate objects is sampled across every image (~100k in this model), which densely cover the image in all locations with different scales and aspect ratios. This can impair training results since easily classified background examples dominate the training phase.

Focal Loss:

Focal loss is a modification to Cross Entropy Loss

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

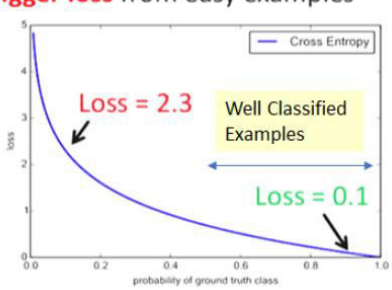
For convenience, it is redefined as:

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

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

When summed over many easy examples, rare classes can be overwhelmed

- 100000 easy : 100 hard examples
- **40x bigger loss** from easy examples



To address these problems, Focal Loss is suggested:

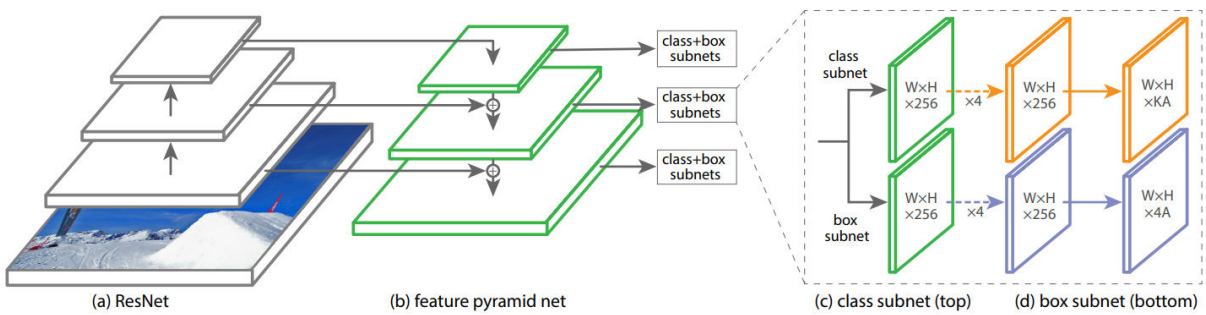
$$\mathrm{FL}(p_t) = -(1-p_t)^{\gamma} \log(p_t).$$

A modulation factor is added (1-pt)^γ to the Cross Entropy Loss, this has 2 affects:

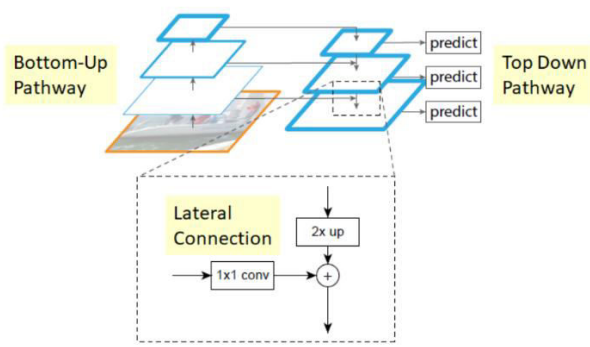
1. When an example is misclassified and pt is small, the modulating factor approaches 1 and the loss function is unaffected. As pt approaches 1, the modulating factor approaches 0, thus the loss for well-classified examples is down-weighted.

2. The focusing parameter γ smoothly adjusts the rate at which easy examples are down-weighted. For instance, with $\gamma = 2$, an example classified with pt = 0.9 would have 100 lower loss compared with CE and with pt = 0.968 it would have 1000 lower loss. This in turn increases the importance of correcting misclassified examples.

Model initialization:
Binary classifiers are usually set to have a 50% chance of outputting every class as default. This initialization, in the presence of class imbalance, can impair training results. To solve this problem, a "prior" value (e.g. 0.01) is set as a default estimate for rare classes. They found this to improve training stability for both the cross entropy and focal loss in the case of heavy class imbalance.



Feature pyramid network:



FPN uses standard convolutional networks to construct feature maps in different scales and then using nearest neighbor for up sampling and lateral connections, it constructs a feature pyramid. This combines low resolution, semantically strong features with high resolution, semantically weak features, this way every feature map contains strong semantic information, which was found to improve prediction results. 1x1 convolution is used on the feature maps from the bottom-up pathway to reduce the numbers of channels so it matches the number of channels in the feature maps in the top down pathway. FPN uses ResNet for the bottom up pathway, and uses the last residual blocks ({C2, C3, C4, C5} which have stride of {4, 8, 16, 32}) respectively for the top down pathway to create the feature pyramid layers {P2, P3, P4, P5}.
In this paper they make small changes to FPN:

- Uses feature pyramid levels P2 to P7, P2 to P5 are the same, P6 is obtained via 3x3 convolution stride = 2 and P7 is obtained and by applying ReLu and using the same convolution.

- P2 is not used for computational reasons.
- P7 and P6 are introduced to improve large object detection.

Anchors:
Anchors of sizes 32^2 – 512^2 are attached to every feature cell in levels P3 – P7 respectively and anchors of size {1, 2^(1/3), 2^(2/3)} of the original sizes are added to every layer with 3 different scales {1:2, 1:1, 2:1}. This means there are 9 anchor boxes per feature cell in every feature map.

The classification subnet:
This subnet predicts the probability of presence of an object at each spatial location for each of the A anchors and K object classes. This subnet is attached to each FPN lever, and parameters are shared across all levels. Given a feature map with C channels, four 3X3 convolutions are applied, each with C filters followed with a ReLU activation function, followed by a 3X3 convolution with KA filters. Finally, the FCN a sigmoid function is used to produce KA binary predictions per spatial location.

Box regression subnet:
In parallel to the classification subnet (weights are not shared), a box regression subnet is attached to each pyramid level in the purpose of regressing bounding boxes to a nearby ground-truth object. The design of the box regression subnet is identical to the classification subnet except that it terminates in 4A linear outputs per spatial location. For each of the A anchor boxes per spatial location, 4 offset values are predicted.