Research Guide: Advanced Loss Functions for Machine Learning Models

[Nearly] Everything you need to know in 2019

Derrick Mwiti

Oct 10, 2019 · 8 min read

Photo by Karim MANJRA on Unsplash

In addition to good training data and the right model architecture, loss functions are one of the most important parts of training an accurate machine learning model. For this post, I'd love to give developers an overview of some of the more advanced loss functions and how they can be used to improve the accuracy of models—or solve entirely new tasks.

For example, semantic segmentation models typically use a simple cross-categorical entropy loss function during training, but if we want to segment objects with many fine details like hair, adding a gradient loss function to the model can vastly improve results.

This is just one example—the following guide explores research centered on a variety of advanced loss functions for machine learning models.

. . .

Robust Bi-Tempered Logistic Loss Based on Bregman Divergences

Logistic loss functions don't perform very well during training when the data in question is very noisy. Such noise can be caused by outliers and mislabeled data. In this paper, Google Brain authors aim to solve the shortcomings of the logistic loss function by replacing the logarithm and exponential functions with their corresponding "tempered" versions.

Bi-Tempered Logistic Loss for Training Neural Nets with Noisy Data

The quality of models produced by machine learning (ML) algorithms directly depends on the quality of the training...

ai.googleblog.com

Google AI

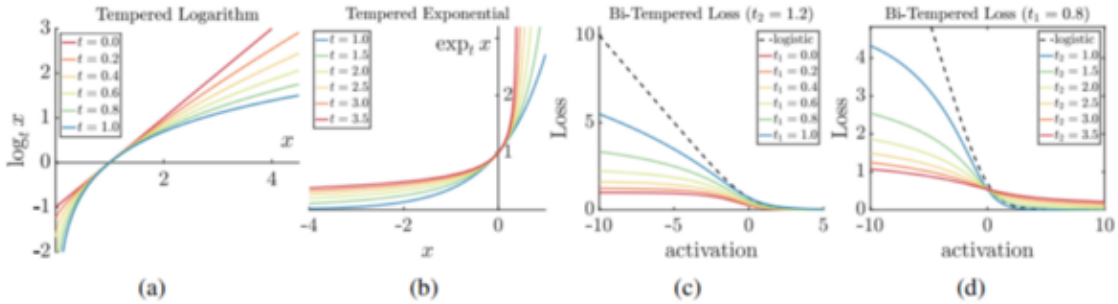


Figure 1: Tempered logarithm and exponential functions, and the bi-tempered logistic loss: (a) $\log_t$ function, (b) $\exp_t$ function, bi-tempered logistic loss when (c) $t_2 = 1.2$ fixed and $t_1 \leqslant 1$, and (d) $t_1 = 0.8$ fixed and $t_2 \geqslant 1$.

source

The authors introduce a temperature into the exponential function and replace the softmax output layer of neural nets with a high-temperature generalization. The

algorithm used in the log loss is replaced by a low-temperature logarithm. The two temperatures are tuned to create loss functions that are nonconvex.

The last neural net layer is replaced with the bi-temperature generalization of the logistic loss. This makes the training process more robust to noise. The method proposed in this paper is based on Bregman divergences. Its performance can be visualized in the figure below.

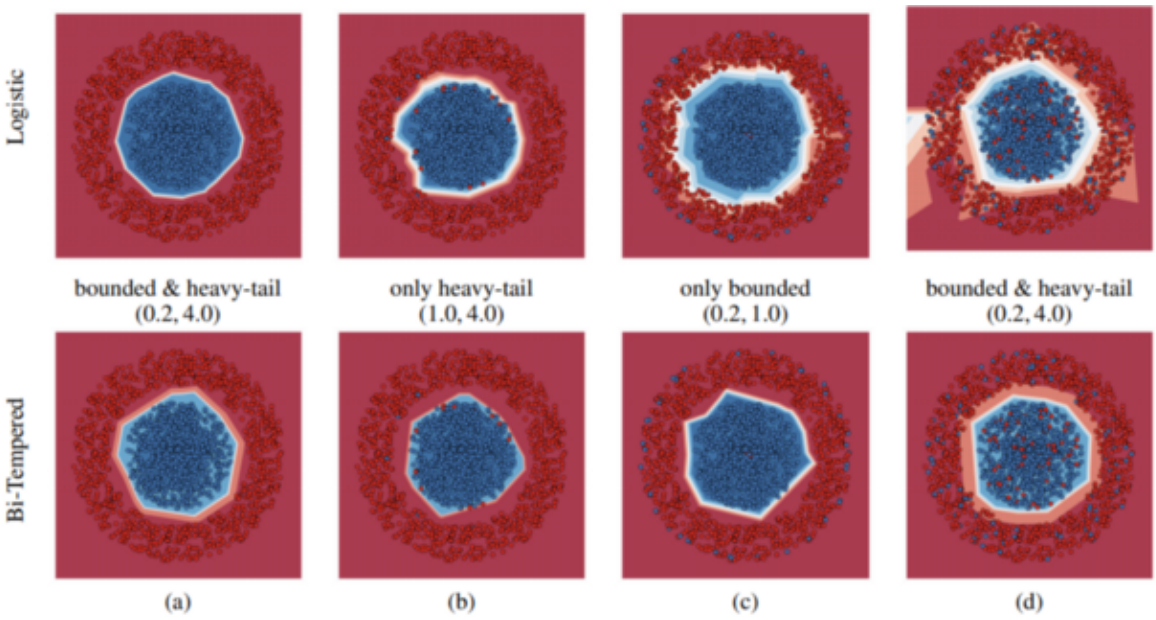


Figure 2: Logistic vs. robust bi-tempered logistic loss: (a) noise-free labels, (b) small-margin label noise, (c) large-margin label noise, and (d) random label noise. The temperature values (t_1, t_2) for the bi-tempered loss are shown above each figure.

source

For experimentation, the authors added synthetic noise to MNIST and CIFAR-100 datasets. The results obtained with their bi-temperature loss function was then compared to the vanilla logistic loss function. The bi-temperature loss obtains an accuracy of 98.56% on MNIST and 62.5% ON CIFAR-100. The figure below shows the performance in detail.

Dataset	Loss	Label Noise Level					
		0.0	0.1	0.2	0.3	0.4	0.5
MNIST	Logistic	**99.40**	98.96	98.70	98.50	97.64	96.13
	Bi-Tempered (0.5, 4.0)	99.24	**99.13**	**99.02**	**98.62**	**98.56**	**97.69**
CIFAR-100	Logistic	74.03	69.94	66.39	63.00	53.17	52.96
	Bi-Tempered (0.8, 1.2)	**75.30**	**73.30**	**70.69**	**67.45**	**62.55**	**57.80**

Table 1: Top-1 accuracy on a clean test set for MNIST and CIFAR-100 datasets where a fraction of the training labels are corrupted.

source

. . .

> Machine learning models are moving closer and closer to edge devices. Fritz AI is here to help with this transition. Explore our suite of developer tools that makes it easy to teach devices to see, hear, sense, and think.

. . .

GANs Loss Functions

Discriminator loss aims at maximizing the probability given to real and fake images. Minimax loss is used in the paper that introduced GANs. This is a strategy aimed at reducing the worst-case-scenario possible loss. It's simply minimizing the maximum loss. This loss is also used in two-player games to reduce the maximum loss for a layer.

> Are GANs Created Equal? A Large-Scale Study
>
> Generative adversarial networks (GAN) are a powerful subclass of generative models. Despite a very rich research...
>
> arxiv.org

In the case of GANs, the two players are the generator and discriminator. This involves the minimization of the generator's loss and maximization of the discriminator's loss. Modification of the discriminator loss forms the non-saturating GAN loss, whose aim is to tackle the saturation problem. This involves the generator maximizing the log of the discriminator probabilities. It is done for the generated images.

Least squares GAN loss was developed to counter the challenges of binary cross-entropy loss that resulted in the generated images being very different from the real images. This loss function is adopted for the discriminator. As a result of this, GANs

using this loss function are able to generate higher quality images than regular GANs. A comparison of the two is shown in the next figure.

> NIPS 2016 Tutorial: Generative Adversarial Networks
>
> This report summarizes the tutorial presented by the author at NIPS 2016 on generative adversarial networks (GANs). The...
>
> arxiv.org

(a) LSGANs. (b) Regular GANs.

(c) LSGANs. (d) Regular GANs.

Figure 7: Comparison experiments by excluding batch normalization (BN). (a): LSGANs without BN in G using Adam. (b): Regular GANs without BN in G using Adam. (c): LSGANs without BN in G and D using RMSProp. (d): Regular GANs without BN in G and D using RMSProp.

sources

The Wasserstein loss function is dependent on the modification of the GAN architecture, where the discriminator doesn't perform instance classification. Instead, the discriminator outputs a number for each instance. It attempts to make the number bigger for real instances than for fake ones.

In this loss function, the discriminator attempts to maximize the difference between the output on real instances and the output on fake instances. The generator, on the other hand, attempts to maximize the discriminator's output for its fake instances.

> Wasserstein GAN
>
> We introduce a new algorithm named WGAN, an alternative to traditional GAN training. In this new model, we show that we...
>
> arxiv.org

Here's an image showing the performance of the GANs using this loss.

Figure 5: Algorithms trained with a DCGAN generator. Left: WGAN algorithm. Right: standard GAN formulation. Both algorithms produce high quality samples.

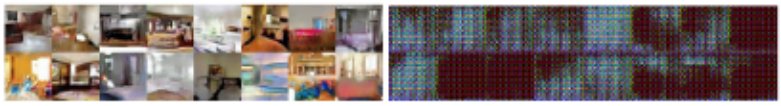


Figure 6: Algorithms trained with a generator without batch normalization and constant number of filters at every layer (as opposed to duplicating them every time as in [18]). Aside from taking out batch normalization, the number of parameters is therefore reduced by a bit more than an order of magnitude. Left: WGAN algorithm. Right: standard GAN formulation. As we can see the standard GAN failed to learn while the WGAN still was able to produce samples.

Figure 7: Algorithms trained with an MLP generator with 4 layers and 512 units with ReLU nonlinearities. The number of parameters is similar to that of a DCGAN, but it lacks a strong inductive bias for image generation. Left: WGAN algorithm. Right: standard GAN formulation. The WGAN method still was able to produce samples, lower quality than the DCGAN, and of higher quality than the MLP of the standard GAN. Note the significant degree of mode collapse in the GAN MLP.

source

. . .

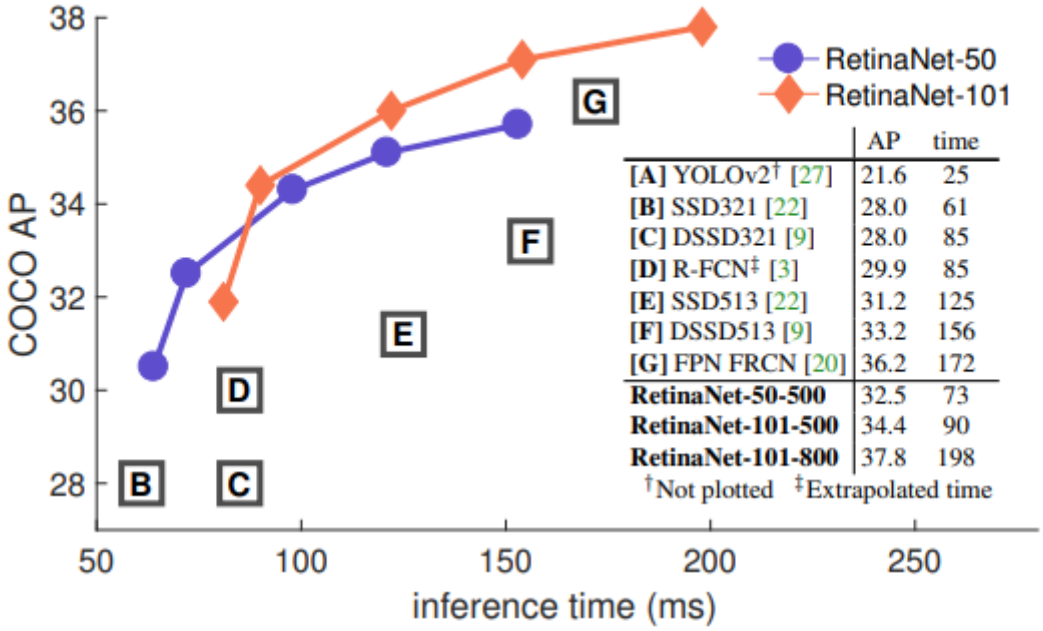
Focal Loss for Dense Object Detection

This paper proposes an improvement to the standard cross-entropy criterion by reshaping it such that it down-weights the loss assigned to the well-classified examples — focal loss. This loss function is aimed at solving the class imbalance problem.

Focal loss aims at training on a sparse set of hard examples and prevents easy negatives from trouncing the detector at training. For testing, the authors develop RetinaNet — a simple dense detector.

> Focal Loss for Dense Object Detection
>
> The highest accuracy object detectors to date are based on a two-stage approach popularized by R-CNN, where a...
>
> arxiv.org

In this loss function, the cross-entropy loss is scaled with the scaling factors decaying at zero as the confidence in the correct classes increases. The scaling factor automatically down weights the contribution of easy examples at training time and focuses on the hard ones.



	AP	time
[A] YOLOv2† [27]	21.6	25
[B] SSD321 [22]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [22]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [20]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

Figure 2. Speed (ms) versus accuracy (AP) on COCO `test-dev`. Enabled by the focal loss, our simple one-stage *RetinaNet* detector outperforms all previous one-stage and two-stage detectors, including the best reported Faster R-CNN [28] system from [20]. We show variants of RetinaNet with ResNet-50-FPN (blue circles) and ResNet-101-FPN (orange diamonds) at five scales (400-800 pixels). Ignoring the low-accuracy regime (AP<25), RetinaNet forms an upper envelope of all current detectors, and an improved variant (not shown) achieves 40.8 AP. Details are given in §5.

source

Here are the results obtained by the focal loss function on RetinaNet.

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [16]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [20]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [34]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [32]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [27]	DarkNet-19 [27]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [22, 9]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [9]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet (ours)	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet (ours)	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2

Table 2. **Object detection** *single-model* results (bounding box AP), *vs.* state-of-the-art on COCO `test-dev`. We show results for our RetinaNet-101-800 model, trained with scale jitter and for 1.5× longer than the same model from Table 1e. Our model achieves top results, outperforming both one-stage and two-stage models. For a detailed breakdown of speed versus accuracy see Table 1e and Figure 2.

source

• • •

• • •

Intersection over Union (IoU)-balanced Loss Functions for Single-stage Object Detection

Loss functions adopted by single-stage detectors perform sub-optimally in localization. This paper proposes an IoU-based loss function that consists of IoU-balanced classification and IoU-balanced localization loss.

> IoU-balanced Loss Functions for Single-stage Object Detection
>
> Single-stage detectors are efficient. However, we find that the loss functions adopted by single-stage detectors are...
>
> arxiv.org

The IoU-balanced classification loss focuses on positive scenarios with high IoU can increase the correlation between classification and the task of localization. The loss aims at decreasing the gradient of the examples with low IoU and increasing the gradient of examples with high IoU. This increases the localization accuracy of models.

The loss's performance on the COCO dataset is shown below.

Table 1: Main results. Comparison of single-stage detectors on COCO *test-dev*.

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet	ResNet50	35.9	55.8	38.4	19.9	38.8	45.0
RetinaNet	ResNet101	38.1	58.5	0.40.8	21.2	0.41.5	48.2
IoU-balanced RetinaNet	ResNet50	37.0	56.2	39.7	20.6	39.8	46.3
IoU-balanced RetinaNet	ResNet101	39.2	58.7	42.3	21.5	42.4	49.4

Table 2: Effectiveness of IoU-balanced Classification Loss and IoU-balanced Localization Loss for RetinaNet-ResNet50 on COCO val-2017.

IoU-balanced Cls	IoU-balanced Loc	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
		34.4	53.9	36.6	17.2	38.2	48

√		35.1	54.6	37.5	18.4	38.5	47.8
	√	35.2	53.7	37.6	17.9	39.3	48.5
√	√	35.7	54.3	38	17.7	39.4	48.8

Table 3: The impact of IoU-balanced Classification Loss and IoU-balanced Localization Loss on AP at different IoU threshold.

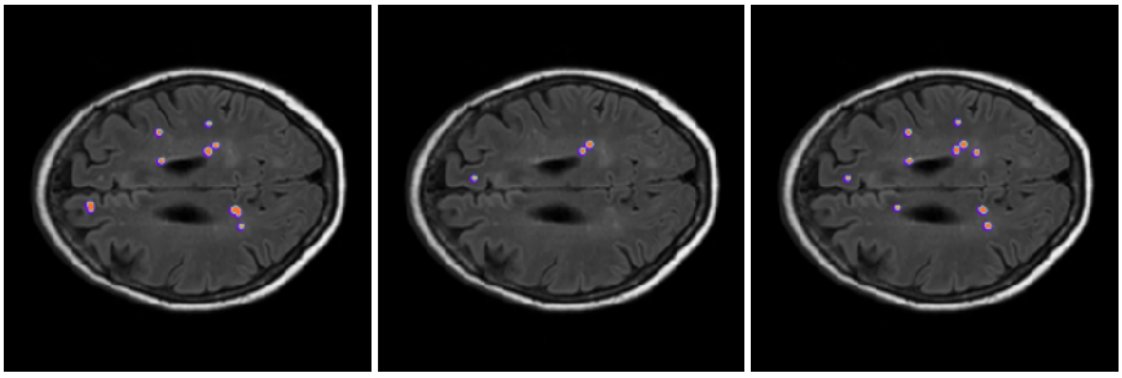
IoU-balanced Cls	IoU-balanced Loc	AP_{50}	AP_{60}	AP_{70}	AP_{80}	AP_{90}
		53.9	49.2	41.9	30.0	11.2
√		54.6(+0.7)	50.2(+1)	42.7(+0.8)	31.0(+1.0)	11.4(+0.2)
	√	53.7(-0.2)	49.3(+0.1)	42.3(+0.4)	31.8(+1.8)	13.1(+1.9)
√	√	54.3(+0.3)	50(+0.8)	43(+1.1)	32.1(+2.1)	13.5(+2.3)

source

. . .

Boundary loss for highly unbalanced segmentation

This paper proposes a boundary loss for highly unbalanced segmentations. The loss takes the form of a distance metric on the space of contours and not regions. This is done to tackle the challenges of regional losses for highly unbalanced segmentation problems. The loss is inspired by discrete optimization techniques for computing gradient flows of curve evolution.



(*a*) Ground truth (*b*) GDL (*c*) GDL w/ boundary loss

Figure 1: A visual comparison that shows the positive effect of our boundary loss on a validation data from the WMH dataset. Our boundary loss helped recovering small regions that were otherwise missed by the generalized Dice loss (GDL). Best viewed in colors.

source

The boundary loss uses integrals over the boundary between regions as opposed to using unbalanced integrals over the regions. An integral approach for computing boundary variations is used. The authors express a non-symmetric L2 distance on the

space of shapes as a regional integral. This avoids local differential computations involving contour points. This then yields a boundary loss that's expressed as the sum of the regional softmax probability outputs of the network. The loss is easily combined with regional losses and incorporated in existing deep network architectures.

The boundary loss was tested on the Ischemic Stroke Lesion (ISLES) and the White Matter Hyperintensities (WMH) benchmark datasets.

source

. . .

Perceptual Loss Function

This loss function is used when images that look similar are being compared. The loss function is primarily used for training feedforward neural networks for tasks image transformation tasks.

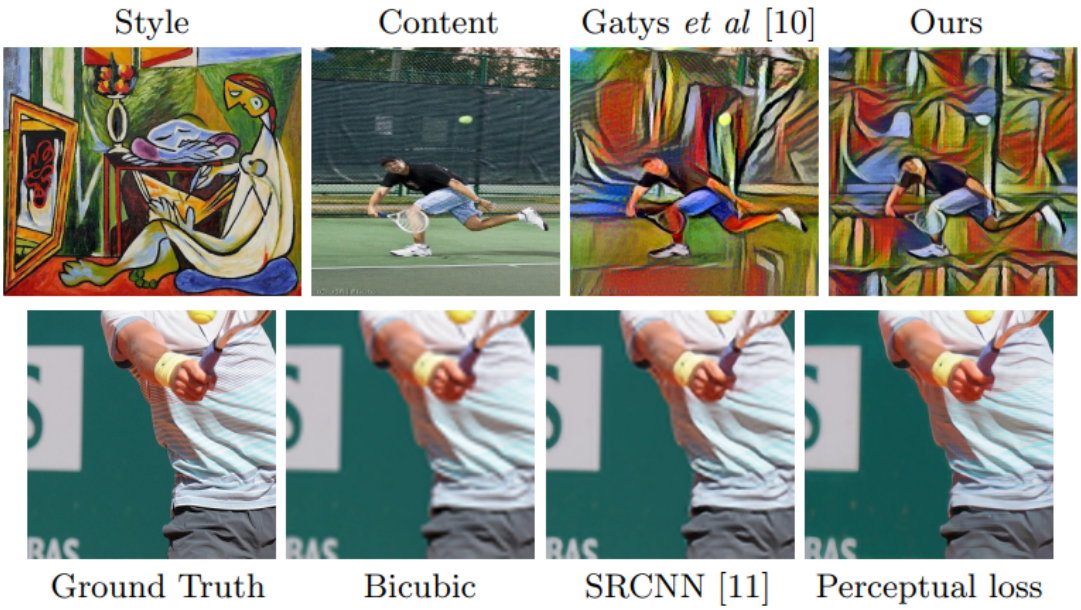


Fig. 1. Example results for style transfer (top) and ×4 super-resolution (bottom). For style transfer, we achieve similar results as Gatys *et al* [10] but are three orders of magnitude faster. For super-resolution our method trained with a perceptual loss is able to better reconstruct fine details compared to methods trained with per-pixel loss.

source

Perceptual Losses for Real-Time Style Transfer and Super-Resolution

We consider image transformation problems, where an input image is transformed into an output image. Recent methods for...

arxiv.org

The perceptual loss function works by adding the squared errors in the middle of all pixels and calculating the mean.

source

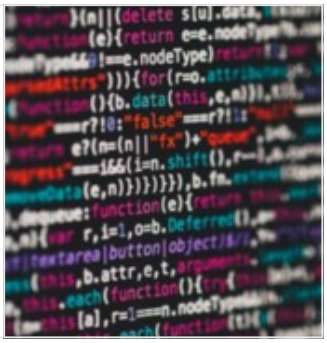
In style transfer, perceptual loss enables deep learning models to reconstruct finer details better than other loss functions. At training time, perceptual losses measure image similarities better than per-pixel loss functions. They also enable the transfer of semantic knowledge from the loss network to the transformation network.

. . .

Conclusion

We should now be up to speed on some of the most common — and a couple of very recent — advanced loss functions.

The papers/abstracts mentioned and linked to above also contain links to their code implementations. We'd be happy to see the results you obtain after testing them.



. . .

Editor's Note: ***Heartbeat*** *is a contributor-driven online publication and community dedicated to exploring the emerging intersection of mobile app development and machine learning. We're committed to supporting and inspiring developers and engineers from all walks of life.*

Editorially independent, Heartbeat is sponsored and published by ***Fritz AI****, the machine learning platform that helps developers teach devices to see, hear, sense, and think. We pay our contributors, and we don't sell ads.*

If you'd like to contribute, head on over to our ***call for contributors****. You can also sign up to receive our weekly newsletters (****Deep Learning Weekly*** *and the* ***Fritz AI Newsletter****),*

*join us on **Slack**, and follow Fritz AI on **Twitter** for all the latest in mobile machine learning.*

Machine Learning | Loss Function | Deep Learning | Heartbeat | Guides And Tutorials