Robust Bi-Tempered Logistic Loss Based on Bregman Divergences

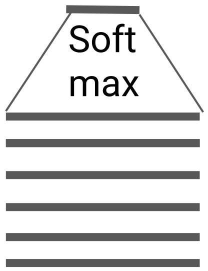
Ehsan Amid
Google Brain & UC Santa Cruz

{eamid, manfred, rohananil, tkoren}@google.com

Nov 12, 2019 - Google NYC

Logistic Loss

Logistic Loss = relative entropy (KL) divergence + softmax probabilities



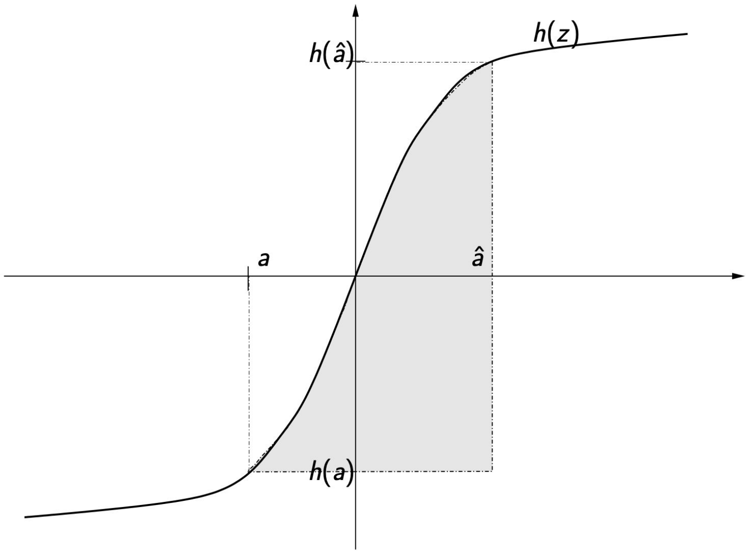
- The most commonly used loss for classification in NN
  - Always convex in the activations/weights of the last layer
  - However, anyway non-convex in weights of lower layers
  - We replace last layer by non-convex loss that makes NN robust to outliers

Matching Loss

[HKW95]

Softmax $\hat{y}_i = \frac{\exp(\hat{a}_i)}{\sum_{j=1}^{k} \exp(\hat{a}_j)}$

Matching loss



Logistic loss = area under sigmoid

$$\Delta_{\log}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_i y_i \log \frac{y_i}{\hat{y}_i}$$

In general

$$\Delta_h(\hat{\boldsymbol{a}}, \boldsymbol{a}) = \int_{\boldsymbol{a}}^{\hat{\boldsymbol{a}}} (h(\mathbf{z}) - h(\boldsymbol{a}))\, d\,\mathbf{z}$$
$$= \Delta_{h^{-1}}(\underbrace{h(\boldsymbol{a})}_{\boldsymbol{y}}, \underbrace{h(\hat{\boldsymbol{a}})}_{\hat{\boldsymbol{y}}})$$

The simplicity of the matching loss

- $$\Delta_h(\hat{\boldsymbol{a}}, \boldsymbol{a}) = \int_{\boldsymbol{a}}^{\hat{\boldsymbol{a}}} (h(\mathbf{z}) - h(\boldsymbol{a}))\, d\,\mathbf{z}$$

- **Convex for any increasing transfer function**

$$\frac{\partial}{\partial \hat{\boldsymbol{a}}} \Delta_h(\hat{\boldsymbol{a}}, \boldsymbol{a}) = h(\hat{\boldsymbol{a}}) - h(\boldsymbol{a}) = \underbrace{\hat{\boldsymbol{y}} - \boldsymbol{y}}_{\text{delta rule}}$$

- Examples

h = id

$$\Delta_{id}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = {}^1\!/_2 \sum_i (y_i - \hat{y_i})^2$$

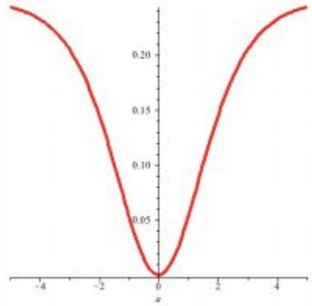
h = softmax

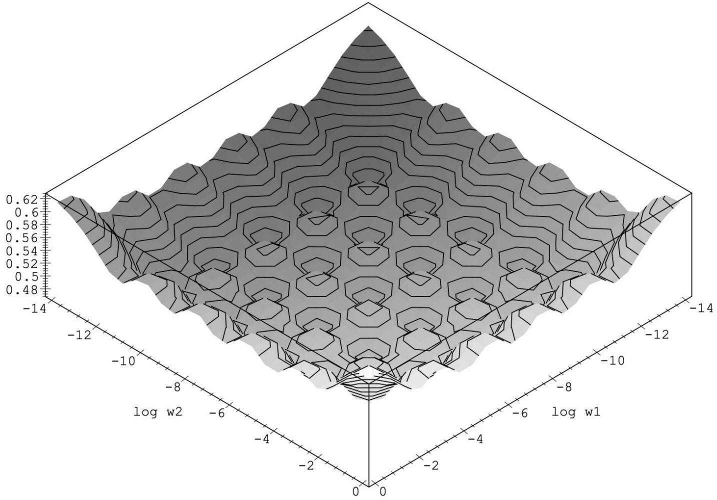
$$\Delta_{\log}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_i y_i \log \frac{y_i}{\hat{y}_i}$$

(logistic loss)

Canonical mismatched case: sigmoid with square loss

$$\frac{\partial}{\partial \hat{a}}(\sigma(\hat{a}) - y)^2 = (\sigma(\hat{a}) - y) \underbrace{\sigma'(\hat{a})}_{\sigma(\hat{a})(1-\sigma(\hat{a}))}$$

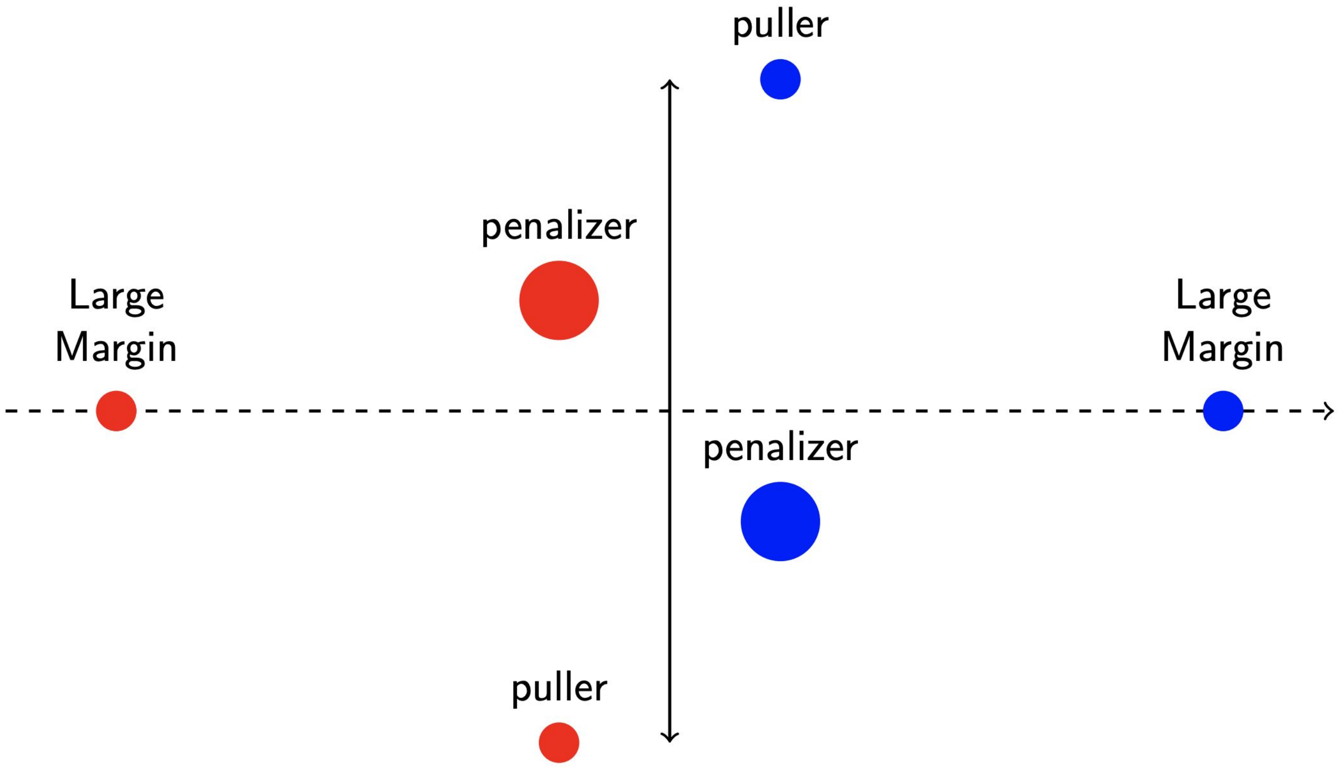




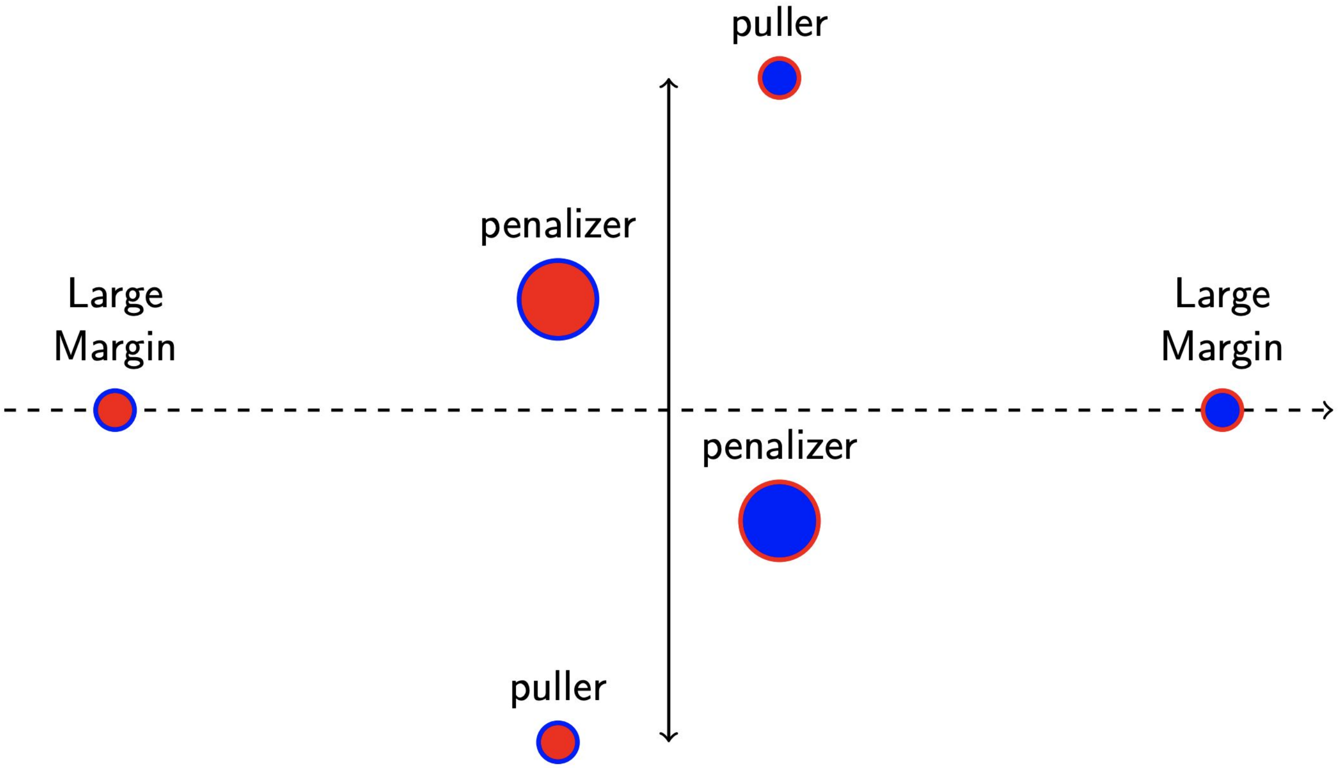
Can lead to exponent. many minimas

[AHW95]

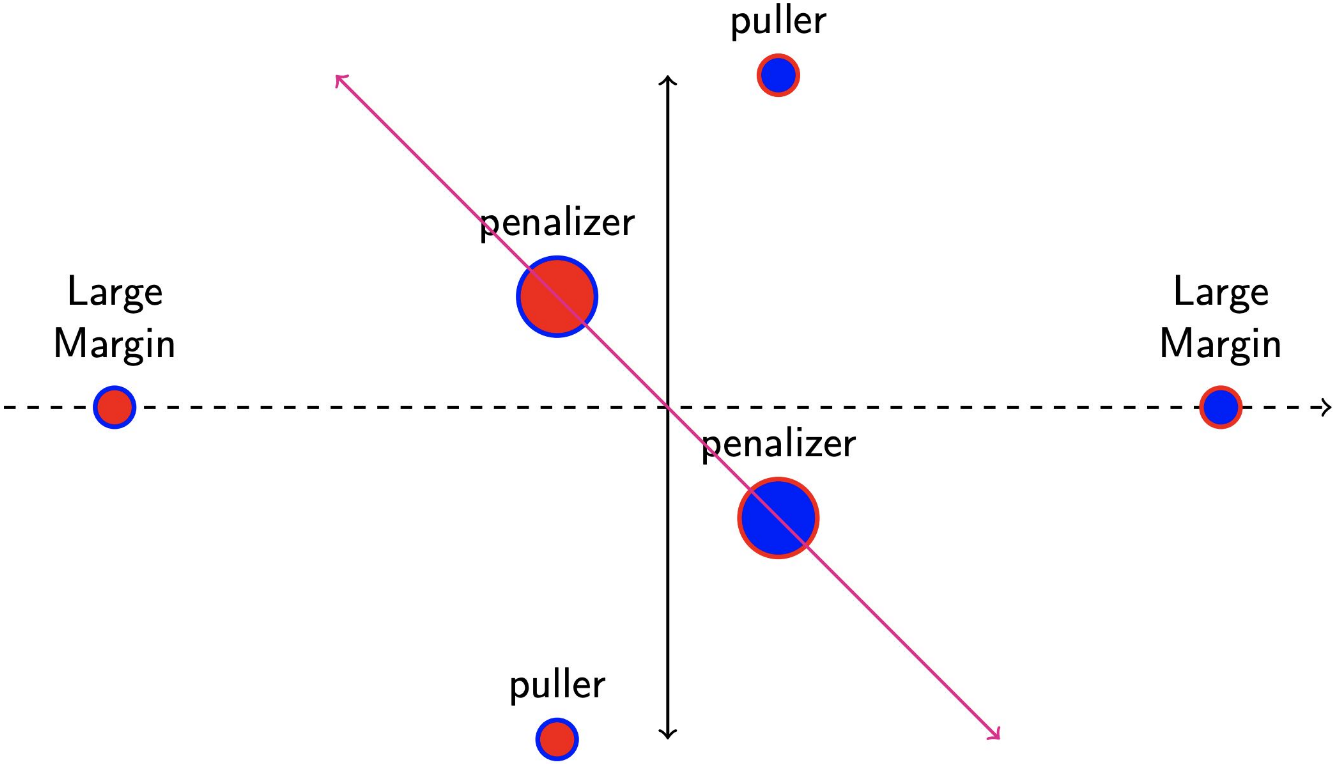
Key example justifying non-convexity [LS08]



10% label noise



Solution determined by the outliers



Logistic Loss as a Matching Loss

Start: Logistic Loss = relative entropy divergence + softmax probabilities

- Introduce temperatures into links, i.e. $\log_{t_1}$ and $\exp_{t_2}$
- When the 2 temperatures are **equal**, we again obtain a **convex** loss
- By increasing the temperature in the exponential,

  loss becomes non-convex
- Tuning the two temperatures will be crucial

Tempered Logarithm and Exponential [N02]

Generalization of log and exp functions endowed with a **temperature** $t \geqslant 0$

$$\log_t(x) := \frac{1}{1-t}(x^{1-t} - 1)$$

Bounded by $-1/(1-t)$ at 0 for $0 \leqslant t < 1$

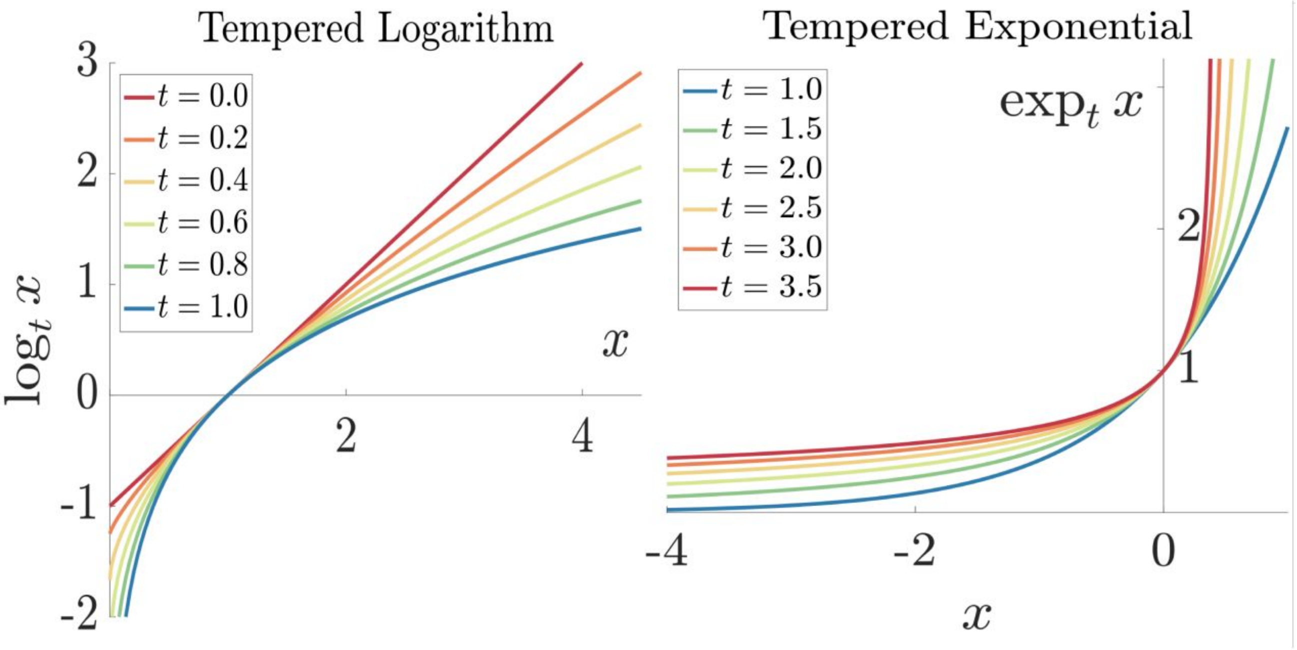
$$\exp_t(x) := [1 + (1-t)\,x]_+^{1/(1-t)}$$

Heavier tail for $x < 0$ for $t > 1$

Standard log and exp are recovered at the limit $t \to 1$

Tempered Logarithm and Exponential

Generalization of log and exp functions endowed with a **temperature** $t \geqslant 0$



Two Drawbacks of Logistic Loss

1. Convex losses are not robust to noise [LS08]
    - Convex losses increase unboundedly
    - A single bad example can dominate the cumulative loss
    - Extreme examples can cause large gradients
    - Non-convex (bending down) losses have been shown to perform significantly better

2. Softmax probabilities have exponentially decaying tail
    - The margin becomes small for mislabeled examples near the boundary
    - Heavy-tailed alternatives yield better margins and improved results [DV,10]

1. Replacing the Relative Entropy Divergence

Tempered relative entropy divergence ($0 \leqslant t_1 < 1$):

$$\sum_{i=1}^{k} \left(y_i \left(\log_{t_1} y_i - \log_{t_1} \hat{y}_i\right) - \tfrac{1}{2-t_1} \left(y_i^{2-t_1} - \hat{y}_i^{2-t_1}\right)\right) \stackrel{\text{if } \boldsymbol{y} \text{ one-hot}}{=} -\log_{t_1} \hat{y}_c - \tfrac{1}{2-t_1} \left(1 - \sum_{i=1}^{k} \hat{y}_i^{2-t_1}\right)$$

bounded by $^1/_{(1-t_1)}$

where $c = \operatorname{argmax}_i y_i$ is the index of the one-hot class

2. Replacing the Softmax Probabilities

$\boldsymbol{z}$: vector of inputs to softmax layer
$\boldsymbol{w}_i$: trainable weight vector for class *i*
$\boldsymbol{y}$: target vector

Softmax:

$$\hat{y}_i = \frac{\exp(\hat{a}_i)}{\sum_{j=1}^{k} \exp(\hat{a}_j)} = \exp\left(\hat{a}_i - \log \sum_{j=1}^{k} \exp(\hat{a}_j)\right), \text{ for linear activation } \hat{a}_i = \boldsymbol{w}_i \cdot \boldsymbol{z} \text{ for class } i.$$

Tempered Softmax ($t_2 > 1$):

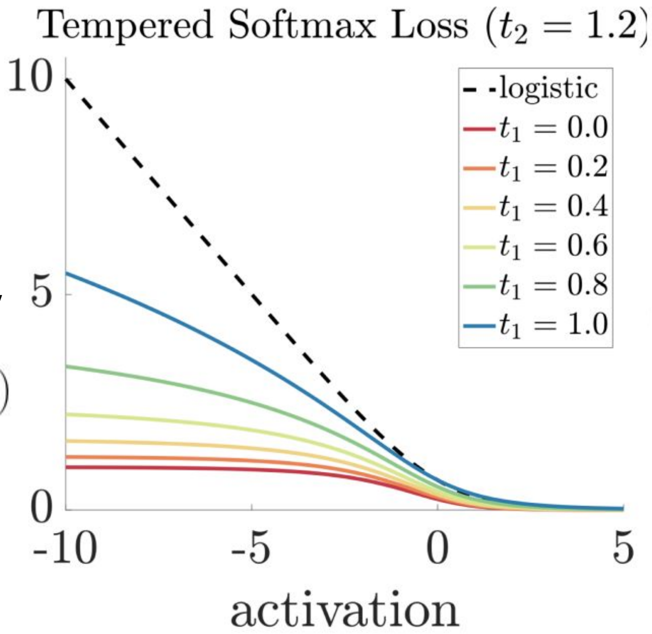
$$\hat{y}_i = \exp_{t_2}\left(\hat{a}_i - \lambda_{t_2}(\hat{\boldsymbol{a}})\right), \quad \text{where} \quad \lambda_{t_2}(\hat{\boldsymbol{a}}) \in \mathbb{R} \quad \text{is s.t.} \quad \sum_{j=1}^{k} \exp_{t_2}\left(\hat{a}_j - \lambda_{t_2}(\hat{\boldsymbol{a}})\right) = 1$$

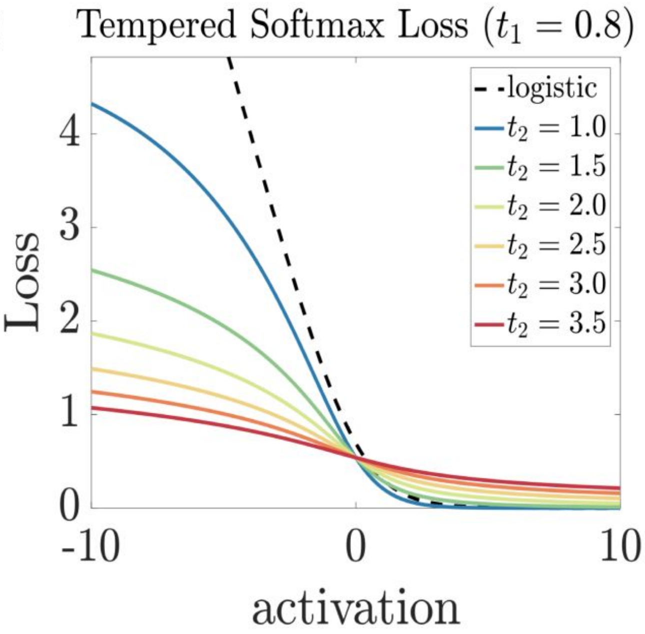
Tail-heavy for t2 > 1!

Examples of the Bi-Tempered Logistic Loss (y = 1.0)

A binary classification task with true label = 1.0

Bounded by $1/(1-t_1)$





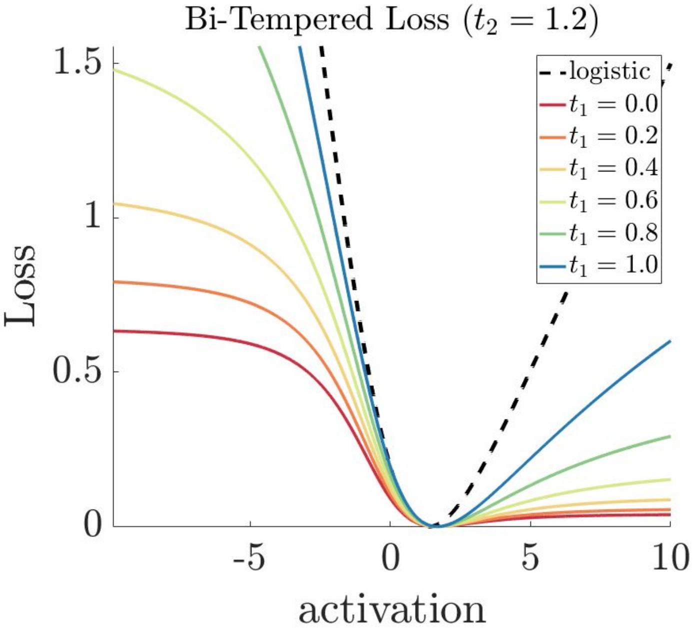
Examples of the Bi-Tempered Logistic Loss (y = 0.8)

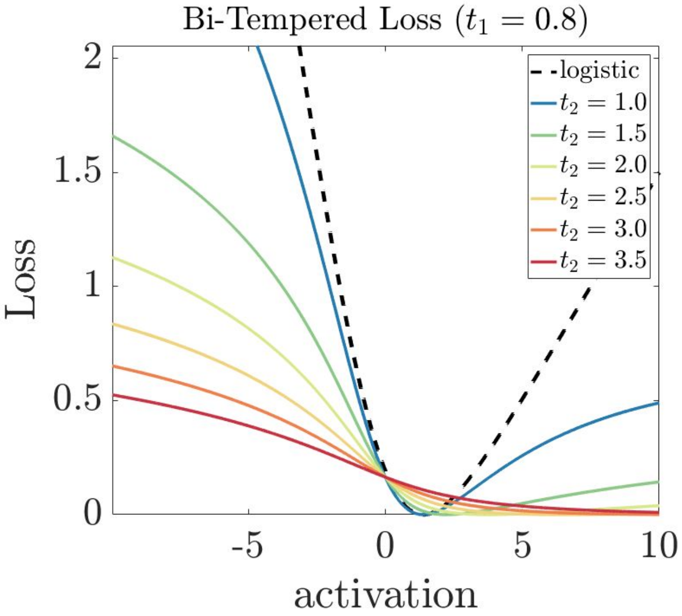
A binary classification task with true label = 0.8

Heavier-tail requires higher activations to produce the same probability

Bounded by

$1/(1 - t_1)$

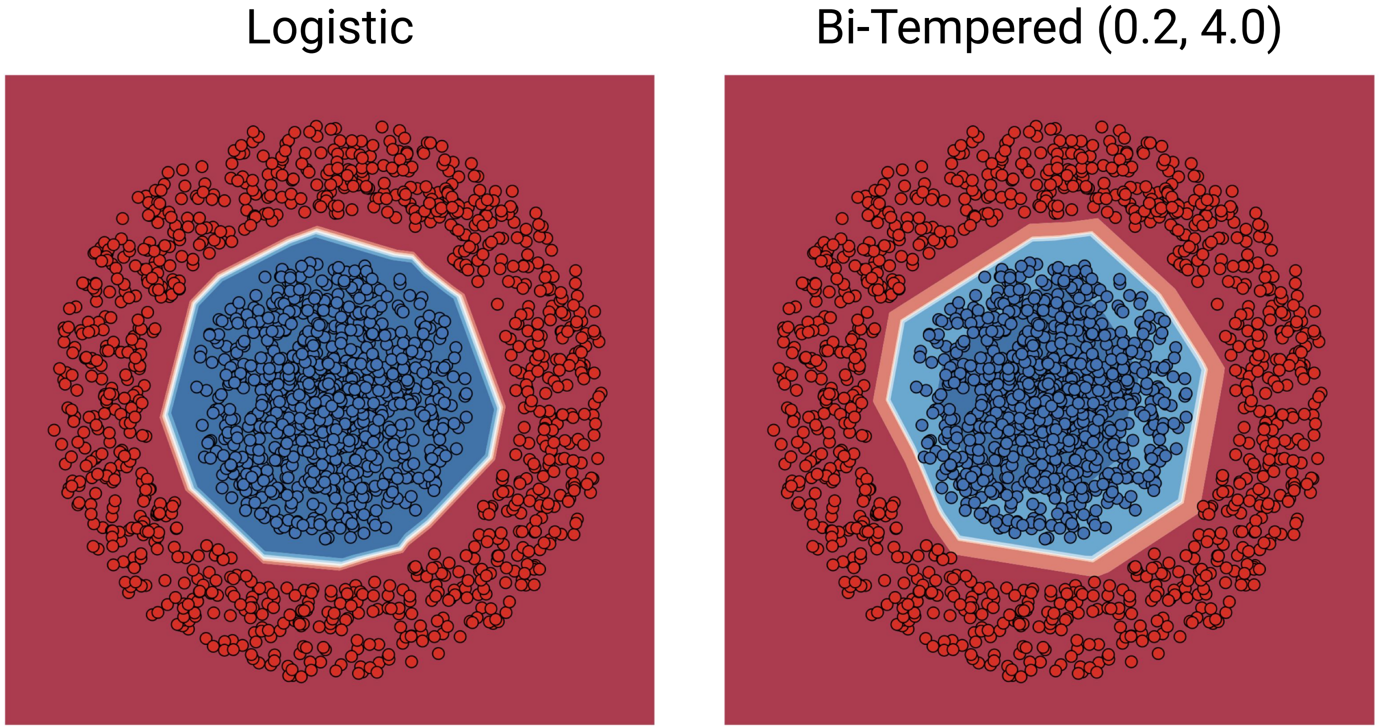




An Illustration

- A small two layer feed-forward neural net on a synthetic binary classification problem in two dimension
  - 10 and 5 units in the first and second layer, respectively
  - Trained using logistic and our bi-tempered logistic loss
  - We add synthetic label noise by flipping the labels
  - Four cases:
    1. Noise-free
    2. Small-margin noise (targeting point near the boundary)
    3. Large-margin noise (targeting points far away from the boundary)
    4. Random noise (points are selected uniformly at random)

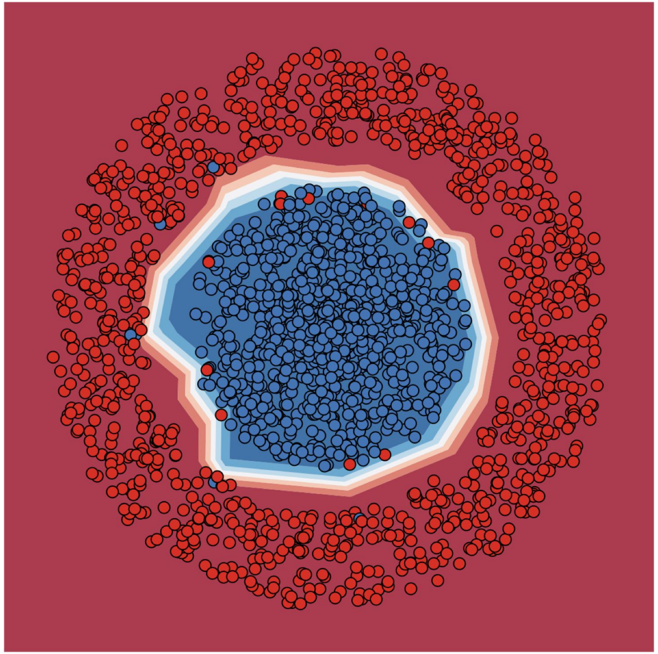
Noise-free Case



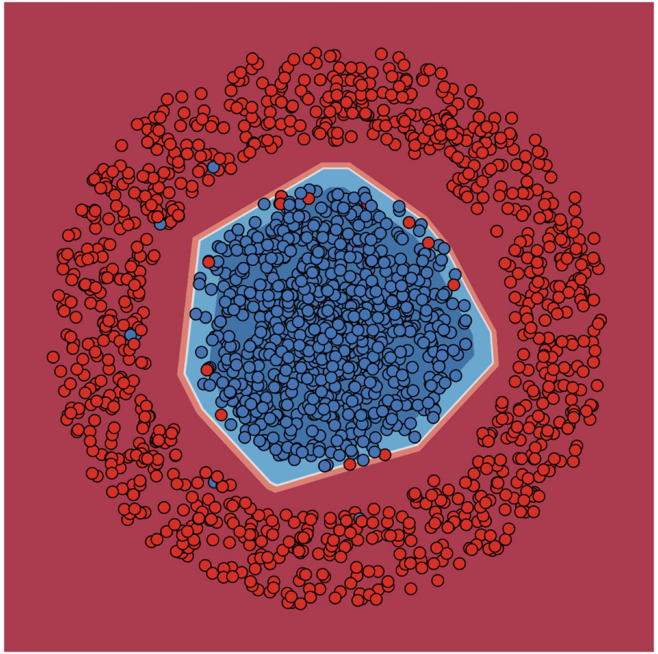
tuned for bounded & tail-heavy

Small-margin Noise

Logistic



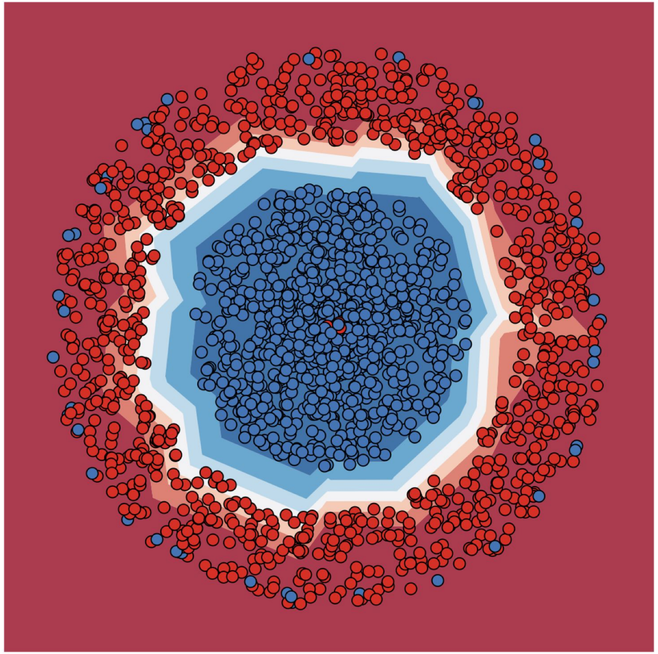
Bi-Tempered (1.0, 4.0)



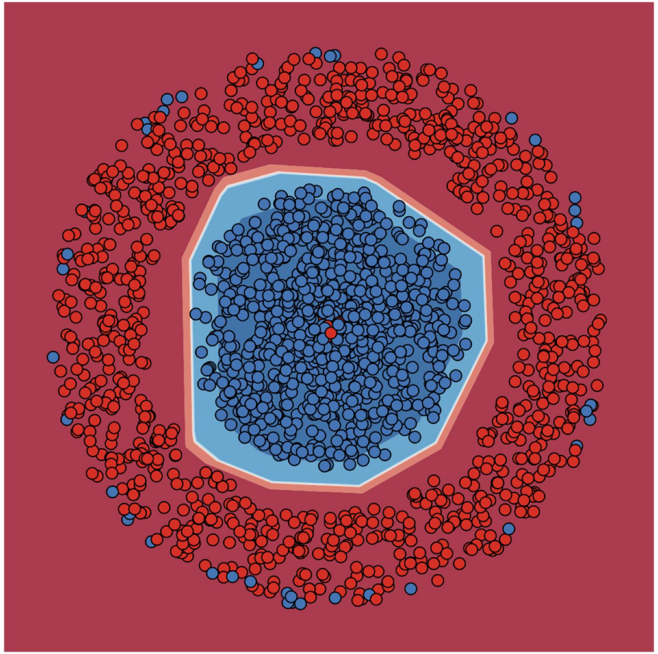
tuned for tail-heavy

Large-margin Noise

Logistic



Bi-Tempered (0.2, 1.0)



tuned for bounded

Experiments

Synthetic Label Noise

For MNIST: 2 convolution layers: 32, 64. Followed by 2 FFN of size 1024 and 10, trained for 500 epochs

For CIFAR-100: a Resnet56 architecture with SGD + momentum optimizer trained for 50k steps with batch size of 128

We search over range [0.5, 1) and (1, 4] for t_1 and t_2, respectively

Synthetic Label Noise

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.

Large-scale Experiments

On the Imagenet-2012 dataset with state-of-the-art Resnet-18 and Resnet-50 models

Trained on a 4x4 CloudTPU-v2 device with a batch size of 4096

180 epochs, SGD + momentum optimizer with staircase learning rate decay schedule

Imagenet Results

Model	Logistic	Bi-tempered (0.9,1.05)
Resnet18	71.333 ± 0.069	$\mathbf{71.618} \pm 0.163$
Resnet50	76.332 ± 0.105	$\mathbf{76.748} \pm 0.164$

Top-1 accuracy

Theoretical Preliminaries

Convex Duality and Bregman Divergences

For a continuously-differentiable strictly convex function $F : \mathcal{D} \to \mathbb{R}$

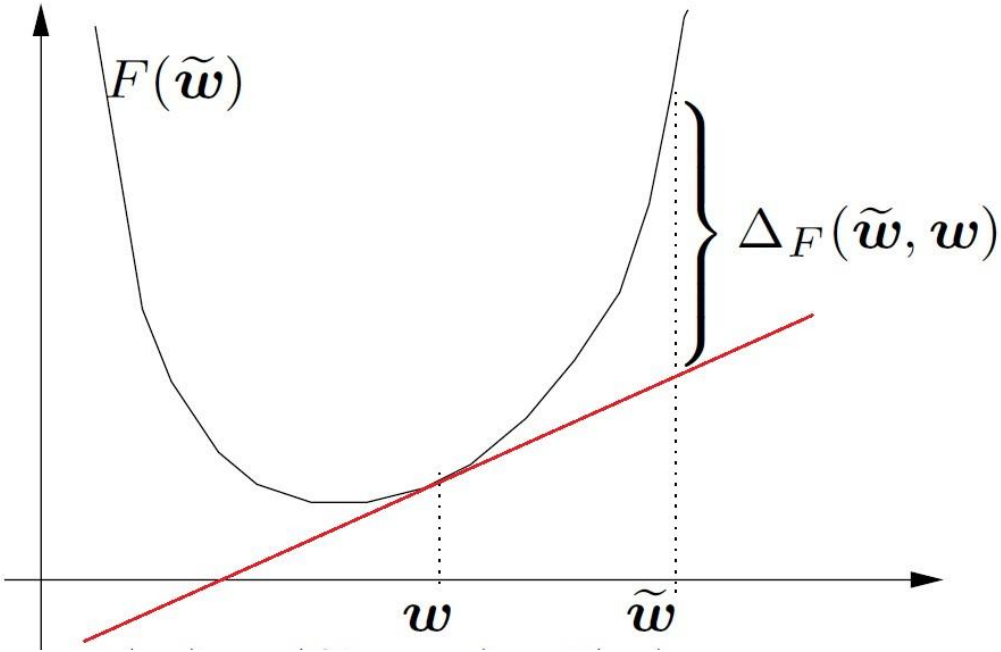
Bregman divergence between $\boldsymbol{y}, \hat{\boldsymbol{y}} \in \mathcal{D}$

$$\Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = F(\boldsymbol{y}) - F(\hat{\boldsymbol{y}}) - (\boldsymbol{y} - \hat{\boldsymbol{y}}) \cdot f(\hat{\boldsymbol{y}})$$

where $f(\hat{\boldsymbol{y}}) \coloneqq \nabla F(\hat{\boldsymbol{y}})$ denotes the gradient

Convex Duality and Bregman Divergences

$F : \mathcal{D} \to \mathbb{R}$



$\Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = F(\boldsymbol{y}) - F(\hat{\boldsymbol{y}}) - (\boldsymbol{y} - \hat{\boldsymbol{y}}) \cdot f(\hat{\boldsymbol{y}})$, where $f(\hat{\boldsymbol{y}}) \coloneqq \nabla F(\hat{\boldsymbol{y}})$

Properties of Bregman Divergence

- **Convexity:** always in the first argument (not necessarily the second)
- **Non-negativity:** $\Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) \geqslant 0$ and $\Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = 0$ iff $\boldsymbol{y} = \hat{\boldsymbol{y}}$
- **Gradient:** $\nabla_{\boldsymbol{y}}\, \Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = f(\boldsymbol{y}) - f(\hat{\boldsymbol{y}})$
- **Invariance to adding affine functions:**

$$\Delta_{F+A}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}), \text{ where } A(\boldsymbol{y}) = b + \boldsymbol{c} \cdot \boldsymbol{y}$$

- **Many well-known cases:**
  - Squared Euclidean: $\Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{2} \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2$ (with $F(\boldsymbol{y}) = \frac{1}{2} \|\boldsymbol{y}\|_2^2$)
  - Relative entropy: $\Delta_F(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_i (y_i \log \frac{y_i}{\hat{y}_i} - y_i + \hat{y}_i)$ (with $F(\boldsymbol{y}) = \sum_i (y_i \log y_i - y_i)$)

Tempered Entropy Function

convex function $F_t : \mathbb{R}_+^k \to \mathbb{R}$ with a temperature parameter $t \geqslant 0$

$$F_t(\boldsymbol{y}) = \sum_{i=1}^{k} \left(y_i \log_t y_i + \tfrac{1}{2-t}(1 - y_i^{2-t}) \right)$$

Gradient: $f_t(\boldsymbol{y}) \coloneqq \nabla F_t(\boldsymbol{y}) = \log_t \boldsymbol{y}$

Lemma 1. *The function F_t, with $0 \leqslant t \leqslant 1$, is B^{-t}–strongly convex over the set $\{\boldsymbol{y} \in \mathbb{R}_+^k : \|\boldsymbol{y}\|_{2-t} \leqslant B\}$ w.r.t. the L_{2-t}-norm.*

Tempered Relative Entropy Divergence

The Bregman divergence induced by F_t

$$\Delta_{F_t}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{i=1}^{k} \left(y_i \log_t y_i - y_i \log_t \hat{y}_i - \tfrac{1}{2-t} y_i^{2-t} + \tfrac{1}{2-t} \hat{y}_i^{2-t} \right)$$

$$= \sum_{i=1}^{k} \left(\tfrac{1}{(1-t)(2-t)} \, y_i^{2-t} - \tfrac{1}{1-t} \, y_i \hat{y}_i^{1-t} + \tfrac{1}{2-t} \, \hat{y}_i^{2-t} \right).$$

Also known as β-divergence with $\beta = 2 - t$

Some Special Cases of the Tempered RE

t	$F_t(\boldsymbol{y})$	$\Delta_{F_t}(\boldsymbol{y}, \hat{\boldsymbol{y}})$	Name
0	$\frac{1}{2}\|\boldsymbol{y}\|_2^2$	$\frac{1}{2}\|\boldsymbol{y}-\hat{\boldsymbol{y}}\|_2^2$	Euclidean
$\frac{1}{2}$	$\frac{1}{3}\sum_i(4\,y_i^{\frac{4}{3}}-6\,y_i+2)$	$\sum_i(\frac{4}{3}y_i^{\frac{3}{2}}-2y_i\sqrt{\hat{y}_i}+\frac{3}{2}\hat{y}_i^{\frac{3}{2}})$	
1	$\sum_i(y_i\log y_i-y_i+1)$	$\sum_i(y_i\log\frac{y_i}{\hat{y}_i}-y_i+\hat{y}_i)$	KL-divergence
$\frac{3}{2}$	$\sum_i(-4\,y_i^{\frac{3}{2}}+2\,y_i+2)$	$2\sum_i\frac{(\sqrt{y_i}-\sqrt{\hat{y}_i})^2}{\sqrt{\hat{y}_i}}$	Squared Xi on roots
2	$\sum_i(-\log y_i+y_i)$	$\sum_i(\frac{y_i}{\hat{y}_i}-\log\frac{y_i}{\hat{y}_i}-1)$	Itakura-Saito
3	$\frac{1}{2}\sum_i(-\frac{1}{y_i}+y_i-2)$	$\frac{1}{2}\sum_i(\frac{1}{y_i}-\frac{2}{\hat{y}_i}+\frac{y_i}{\hat{y}_i^2})$	Inverse

Tempered Transfer Function

Using the duality argument

$$\check{F}_t^*(\boldsymbol{a}) = \sup_{\boldsymbol{y}' \in S^k} \left(\boldsymbol{y}' \cdot \boldsymbol{a} - F_t(\boldsymbol{y}')\right) = \sup_{\boldsymbol{y}' \in \mathbb{R}_+^k} \inf_{\lambda_t \in \mathbb{R}} \left(\boldsymbol{y}' \cdot \boldsymbol{a} - F_t(\boldsymbol{y}') + \lambda_t \left(1 - \sum_{i=1}^{k} y_i'\right)\right)$$

The tempered (softmax) transfer function becomes

$$\boldsymbol{y} = \exp_t\left(\boldsymbol{a} - \lambda_t(\boldsymbol{a})\,\mathbf{1}\right), \quad \text{with } \lambda_t(\boldsymbol{a}) \text{ s.t. } \sum_{i=1}^{k} \exp_t\left(a_i - \lambda_t(\boldsymbol{a})\right) = 1$$

Robust Bi-Tempered Logistic Loss

Bi-tempered logistic = tempered relative entropy divergence + tempered softmax

$$\forall\, 0 \leqslant t_1 < 1 < t_2 :\; L_{t_1}^{t_2}(\hat{\boldsymbol{a}} \mid \boldsymbol{y}) := \Delta_{F_{t_1}}\left(\boldsymbol{y}, \exp_{t_2}(\hat{\boldsymbol{a}} - \lambda_{t_2}(\hat{\boldsymbol{a}}))\right), \;\text{ with } \lambda_t(\hat{\boldsymbol{a}}) \;\text{ s.t. } \sum_{i=1}^{k} \exp_t\left(a_i - \lambda_t(\boldsymbol{a})\right) = 1$$

$0 \leqslant t_1 < 1$: controls boundedness of relative entropy

$1 < t_2$: controls tail-heaviness of softmax

Two Important Properties

1. Properness
    - Ensures that we have an unbiased estimator of the expected loss
2. Bayes-risk Consistency
    - Implies inference can be done via the argmax operation over the activations

Properness

Model fitting:

Unknown data distribution: $P_{\mathrm{UK}}(y \mid \boldsymbol{x})$

Model distribution: $P(y \mid \boldsymbol{x}; \Theta)$ parameterized by Θ

Minimize:

$$\mathbb{E}_{P_{\mathrm{UK}}(\boldsymbol{x})}\left[\Delta\left(P_{\mathrm{UK}}(y \mid \boldsymbol{x}), P(y \mid \boldsymbol{x}; \Theta)\right)\right]$$

We would like to get an unbiased estimator

Properness of Bi-Tempered Loss

Ignoring the constant terms w.r.t. Θ

$$\mathbb{E}_{P_{\mathrm{UK}}(\boldsymbol{x})}\Big[\sum_i \big(-P_{\mathrm{UK}}(i \mid \boldsymbol{x}) \log_t P(i \mid \boldsymbol{x}; \Theta) + \frac{1}{2-t} P(i \mid \boldsymbol{x}; \Theta)^{2-t}\big)\Big]$$

$$\approx \frac{1}{N} \sum_n \sum_i \big(-P_{\mathrm{UK}}(i \mid \boldsymbol{x}_n) \log_t P(i \mid \boldsymbol{x}_n; \Theta) + \frac{1}{2-t} P(i \mid \boldsymbol{x}_n; \Theta)^{2-t}\big)$$

$$\approx \frac{1}{N} \sum_n \big(-\log_t P(y_n \mid \boldsymbol{x}_n; \Theta) + \sum_i \frac{1}{2-t} P(i \mid \boldsymbol{x}_n; \Theta)^{2-t}\big) ,$$

Thus, it is a proper loss

Tsallis Divergence is not Proper

The approximation using the Tsallis divergence is not proper **[AWS19]**

$$\mathbb{E}_{P_{\mathrm{UK}}(\boldsymbol{x})}\Bigg[\underbrace{-\sum_i P_{\mathrm{UK}}(i \mid \boldsymbol{x}) \log_t \frac{P(i \mid \boldsymbol{x}; \Theta)}{P_{\mathrm{UK}}(i \mid \boldsymbol{x})}}_{\Delta_t^{\mathrm{Tsallis}}\left(P_{\mathrm{UK}}(y|\boldsymbol{x}),\, P(y|\boldsymbol{x};\Theta)\right)}\Bigg] \approx -\frac{1}{N}\sum_n \log_t \frac{P(y_n \mid \boldsymbol{x}_n; \Theta)}{P_{\mathrm{UK}}(y_n \mid \boldsymbol{x}_n)}$$

But in practice, $P_{\mathrm{UK}}(y_n \mid \boldsymbol{x}_n)$ is unknown and the loss is approximated by

$$-\tfrac{1}{N}\sum_n \log_t P(y_n \mid \boldsymbol{x}_n; \Theta)$$

Bayes-risk Consistency

The conditional risk of the multiclass loss $\boldsymbol{l}(\hat{\boldsymbol{a}})$ with $l_i := \ell(\hat{\boldsymbol{a}} | y = i)$, $i \in [k]$ is defined as

$$R(\boldsymbol{\eta}, \boldsymbol{l}(\hat{\boldsymbol{a}})) = \sum_i \eta_i \, l_i \,,$$

where $\eta_i := P_{\mathrm{UK}}(y = i | \boldsymbol{x})$.

Definition 4 (Bayes-risk Consistency). A Bayes-risk consistent loss for multiclass classification is the class of loss functions ℓ for which $\hat{\boldsymbol{a}}^\star$, the minimizer of $R(\boldsymbol{\eta}, \boldsymbol{l}(\hat{\boldsymbol{a}}))$, satisfies

$$\arg\min_i \ \ell(\hat{\boldsymbol{a}}^\star | \, y = i) \subseteq \mathrm{argmax}_i \, \eta_i \,.$$

Proposition 2. *The multiclass bi-tempered logistic loss* $L_{t_1}^{t_2}(\hat{\boldsymbol{a}} \,|\, y)$ *is Bayes-risk consistent.*

Implementation and Future Work

Implementation

Current version of the paper accepted to NeurIPS 2019:
https://arxiv.org/pdf/1906.03361.pdf

Open source TF implementation available at **Google research Github**:

https://github.com/google-research/google-research/tree/master/bitempered_loss

Replace one line:

~~`Softmax_cross_entropy_with_logits(activations,labels)`~~

`bi_tempered_logistic_loss(activation,labels,t1,t2)`

Future Work

- Better tuning of the temps (possibly dynamic tuning during training)
- Reoptimize all other variables:

  regularization, batch normalization, structure, dropout, ...
- Use Bi-Tempered loss for language models, ad placement, ...
- Can we avoid model blow-ups w. new loss
- Design asymmetric losses
- Long-term: generalization of the matching loss for deep NNs

References

[HKW95] David P. Helmbold, Jyrki Kivinen and Manfred K. Warmuth. Worst-case loss bounds for single neurons. NIPS ‘95, pp. 309–315.

[AHW95] Peter Auer, Mark Herbster, and Manfred K. Warmuth. Exponentially Many Local Minima for Single Neurons. NIPS’95. pp. 315–322.

[KW01] Jyrki Kivinen and Manfred K. Warmuth. Relative loss bounds for multidimensional regression problems. Journal of Machine Learning, Vol. 45(3), pp. 301-329.

[LS08] Philip M Long and Rocco A Servedio. Random classification noise defeats all convex potential boosters. ICML’08. pp. 608–615.

[DV10] Nan Ding and S. V. N. Vishwanathan. t-logistic regression. NIPS’10, pp. 514–522.

[AWS19] Ehsan Amid, Manfred K. Warmuth, and Sriram Srinivasan. Two-temperature logistic regression based on the Tsallis divergence. AISTATS’19.