A Re-Balancing Strategy for Class-Imbalanced Classification Based on Instance Difficulty

CVPR 2022

Paper Reading by Erin

智能多媒体内容计算实验室
Intelligent Multimedia Content Computing Lab

- 作者介绍
- 研究背景
- 本文方法
- 实验效果
- 总结反思

作者介绍

A Re-Balancing Strategy for Class-Imbalanced Classification Based on Instance Difficulty

Sihao Yu, Jiafeng Guo*, Ruqing Zhang, Yixing Fan, Zizhen Wang and Xueqi Cheng
University of Chinese Academy of Sciences, Beijing, China
CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China

- 作者介绍
- 研究背景
- 本文方法
- 实验效果
- 总结反思

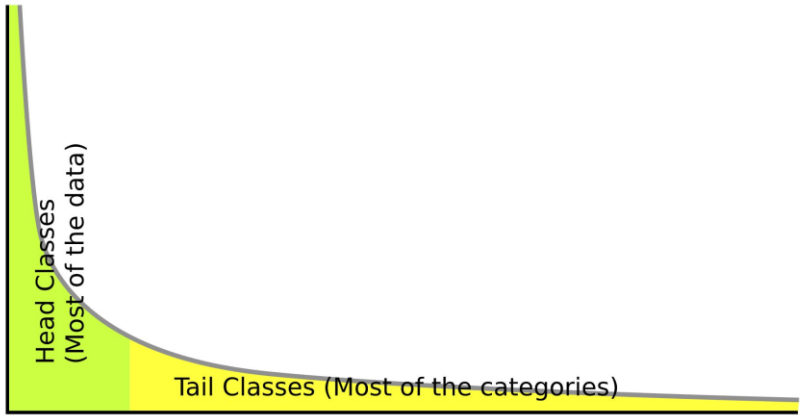
Long-Tailed Distribution

任务：分类、
目标检测、
实例分割

特点：少数类别样本数量庞大，
多数类别样本数量稀少

问题：数据分布先验影响分类精度

偏向头部类别的误分类
尾部类别样本量过少，难以建模，精度很低

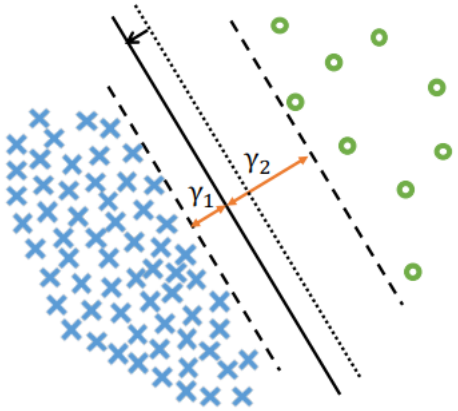


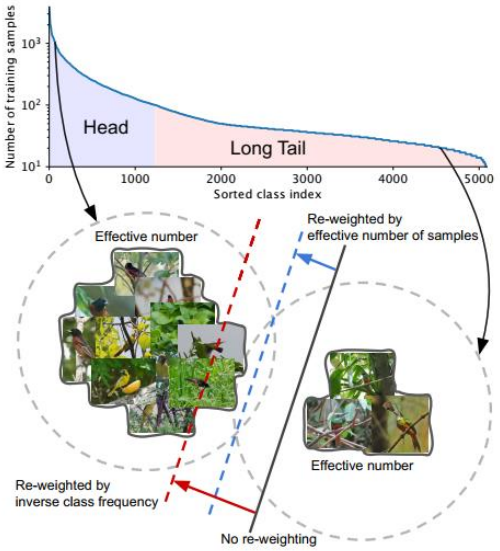
Re-balancing

重采样

简单复制、类别均衡复制
根据样本的统计特征生成合成样本

重加权

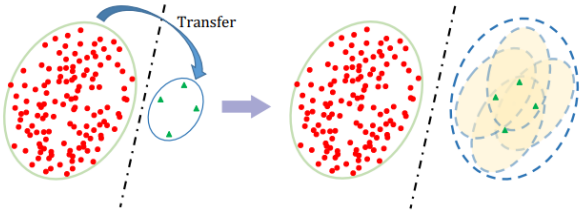


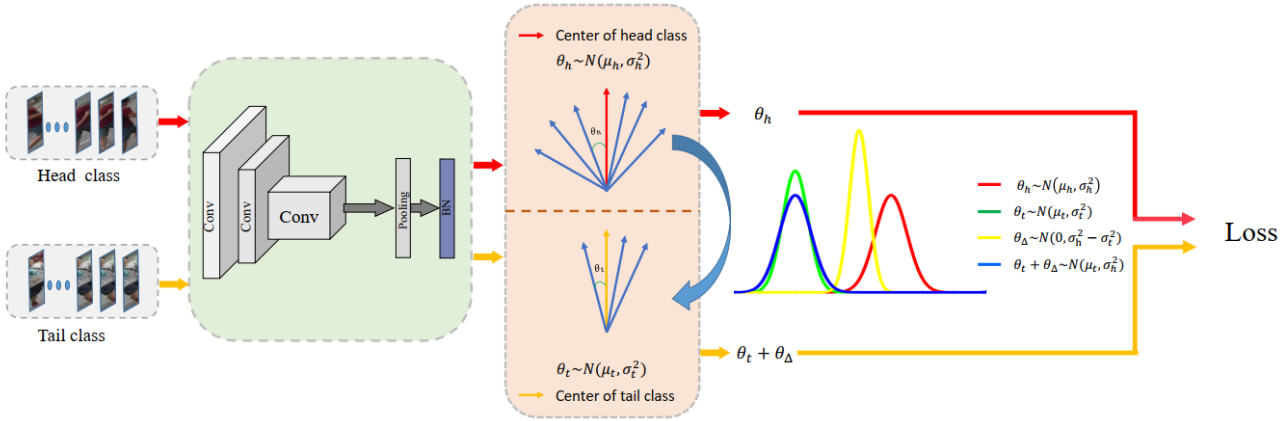


Class-level

Transfer Learning

将头部类别的性质迁移到尾部类别
直接在特征空间里操作
特征点→特征云



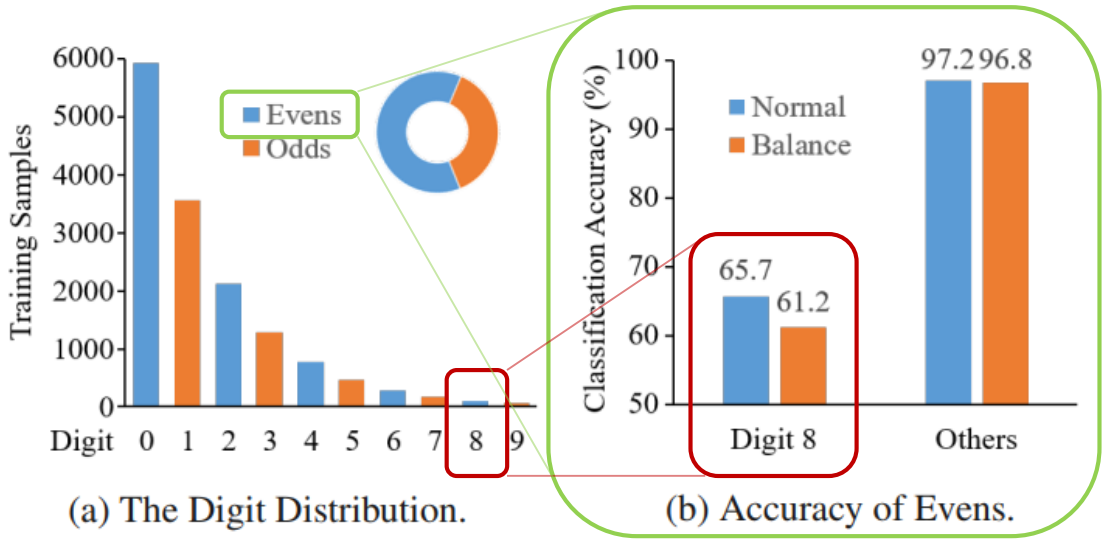


Class-level

Motivation

Class-level

多数类存在难样本吗？
一个栗子👇



(a) The Digit Distribution. (b) Accuracy of Evens.

Figure 1. Binary Classification on the Long-Taied MNIST. Fig. 1a

Class-level Method：

牺牲一定多数类性能
对少数类精度有较高提升

多数情况：

子类分布未知
甚至难以定义子类

Motivation

Class-level

多数类存在难样本吗？
Class-level：忽略类内差异

Instance-level

Instance-level：对实例自适应
训练时模型对不同样本的学习程度不同
学习程度⇔样本难易

Hard Mining

如何衡量样本难易？

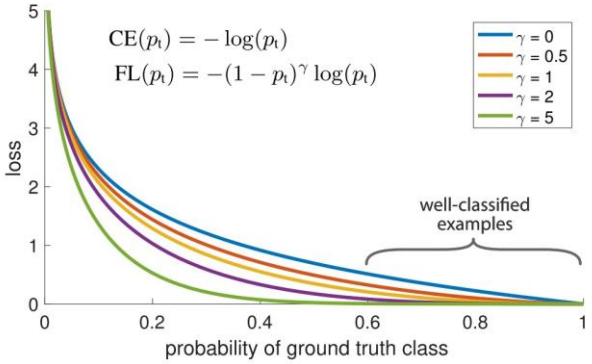
- Loss Magnitude（FOCAL、ACSL…）
- Gradient Magnitude（GHM）
- Meta-Learning

如何调整模型？

- 重加权：调整损失函数权重
- 重采样：调整训练数据分布
- 课程学习、自主学习…

Hard Mining

FOCAL



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

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

ACSL

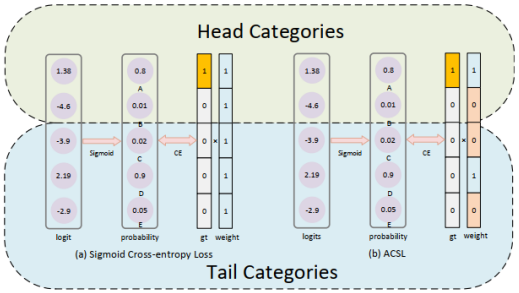
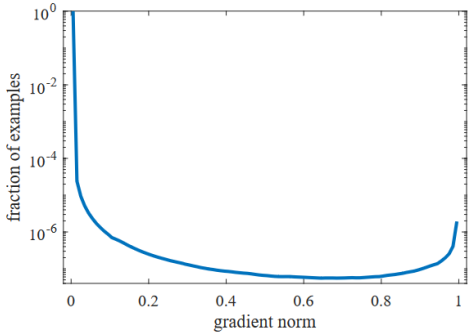


Figure 3: An illustration of Sigmoid Cross-entropy Loss and our proposed ACSL. The top two classes belong to head categories and the bottom three classes belong to tail categories. For ACSL, the hyper-parameter ξ is 0.7.

$$L_{ACSL}(x_s) = -\sum_{i=1}^{C} w_i log(\hat{p}_i)$$

$$w_i = \begin{cases} 1, & \text{if } i = k \\ 1, & \text{if } i \neq k \text{ and } p_i \geq \xi \\ 0, & \text{if } i \neq k \text{ and } p_i < \xi \end{cases}$$

GHM



$$GD(g) = \frac{1}{l_\epsilon(g)} \sum_{k=1}^{N} \delta_\epsilon(g_k, g)$$

$$\beta_i = \frac{N}{GD(g_i)}$$

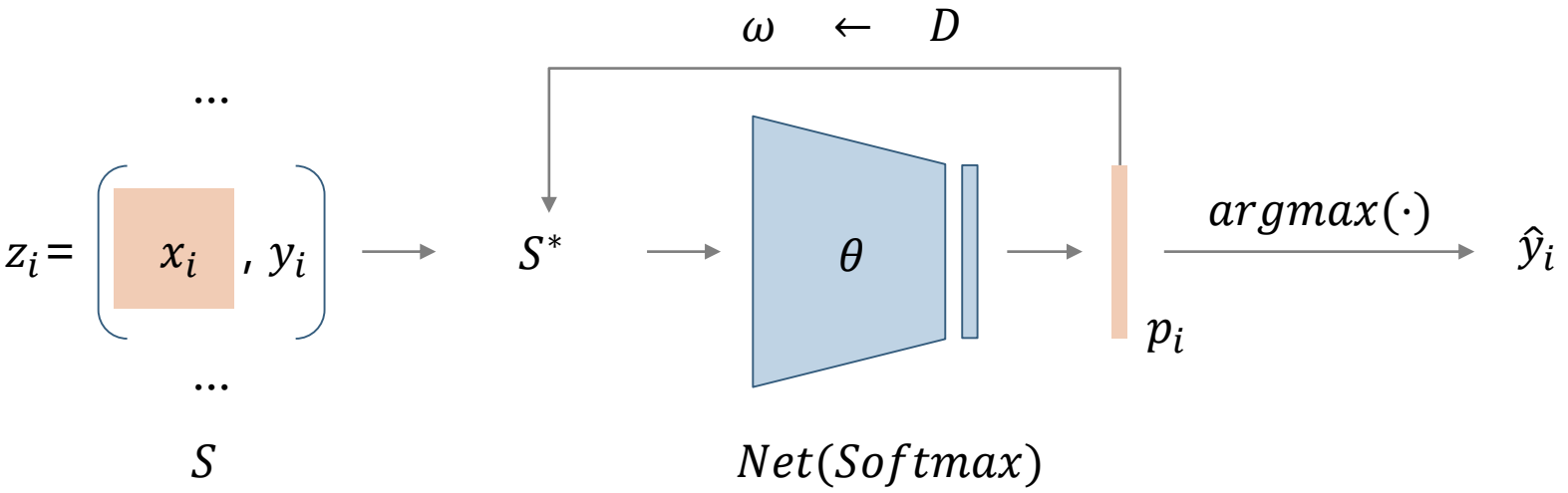
- 作者介绍
- 研究背景
- 本文方法
- 实验效果
- 总结反思

Instance Difficulty Modeling

- Overview
  - 如何衡量样本难易？
    - 训练过程中，模型对实例的**学习速度**
  - 如何调整模型？
    - 重采样：调整训练数据分布

Instance Difficulty Modeling

- Task Formulation



$$\mathcal{L}(\theta, \mathcal{S}) = \sum_{i=1}^{N} \mathcal{L}(\theta, z_i)$$ 二阶可导

Instance Difficulty Modeling

- Re-sample for training

Algorithm 1 Re-Sampling

Input: dataset $\mathcal{S}$, network Net, training times T
Initialize sampling weight (probability) $\omega \leftarrow \{\frac{1}{|\mathcal{S}|}\}^{|\mathcal{S}|}$
Initialize $p_{i,0} \leftarrow \{\frac{1}{k}, \dots\}$ for each $\boldsymbol{x}_i$ in $\mathcal{S}$
for t **in** 1 **to** T **do**
  $\mathcal{S}^* \leftarrow$ Sample from $\mathcal{S}$ according to ω
  Train Net by using $\mathcal{S}^*$
  for $\boldsymbol{x}_i$ **in** $\mathcal{S}$ **do**
    $p_{i,t} \leftarrow Net(\boldsymbol{x}_i)$
    $D_{i,t} \leftarrow Difficulty(p_{i,0}, \dots, p_{i,t})$
  end for
  calculate new ω by D
end for

$$w_{i,t} = \frac{D_{i,t}}{\sum_{j=1}^{N} D_{j,t}}$$

Instance Difficulty Modeling

- General Model

$$\mathcal{L}(\theta, \mathcal{S}) = \mathcal{L}(\theta_0, \mathcal{S}) + \mathcal{L}'(\theta_0, \mathcal{S})(\theta - \theta_0)$$

$$\triangle\theta = (\theta - \theta_0) = -\eta\mathcal{L}'(\theta_0, \mathcal{S})$$

$$\mathcal{L}(\theta_1, z) - \mathcal{L}(\theta_0, z) = \triangle\mathcal{L}_z = -\eta\langle\mathcal{L}'(\theta_0, z), \mathcal{L}'(\theta_0, \mathcal{S})\rangle$$

要学习的实例　数据集

辅助集　$\mathcal{A}_z := \{a : a \in \mathcal{S}, \langle\mathcal{L}'(\theta_0, z), \mathcal{L}'(\theta_0, a)\rangle > 0\}$

阻碍集　$\mathcal{H}_z := \{r : r \in \mathcal{S}, \langle\mathcal{L}'(\theta_0, z), \mathcal{L}'(\theta_0, r)\rangle < 0\}$

→ **样本难易**

内积计算消耗大
→相邻两次迭代，Δθ较小时，模型预测变化趋势相似
→使用预测变化量估计实例倾向

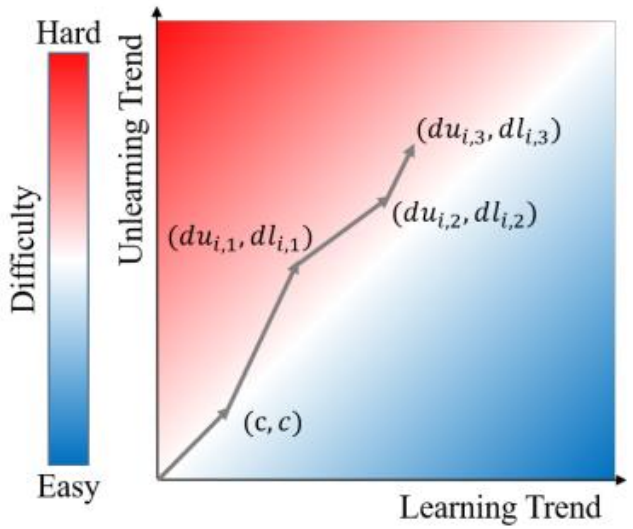
Instance Difficulty Modeling

- General Model

$$\vec{D}_{i,T} = \vec{c} + \sum_{t=1}^{T} \vec{d}_{i,t}$$

$$\vec{c} = (c, c)$$

$$\vec{d}_{i,t} = (du_{i,t}, dl_{i,t})$$



Unlearning方向的预测变化量

$$D_{i,T} = \frac{c + \sum_{t=1}^{T} du_{i,t}}{c + \sum_{t=1}^{T} dl_{i,t}}$$

斜率

learning方向的预测变化量

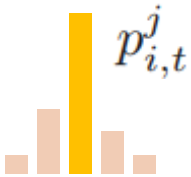
$D_{i,t} > D_{j,t}$ 表示Zi比Zj更难学习

$$||\vec{D}_{i,t-1}|| = ||\vec{D}_{i,t}|| \text{ when } t \to \infty$$

Instance Difficulty Modeling

- Specific Instance

$$d_{i,t} = \sum_{j=1}^{k} (p_{i,t}^{j} - p_{i,t-1}^{j}) \ln(\frac{p_{i,t}^{j}}{p_{i,t-1}^{j}})$$



稳定度指标(population stability index ,PSI)，
衡量分布差异，常用于评估模型稳定度

$$d_{i,t} = du_{i,t} + dl_{i,t}$$

$p_{i,t}^{y_i} - p_{i,t-1}^{y_i} > 0$ learning

$$dl_{i,t} = \max(p_{i,t}^{y_i} - p_{i,t-1}^{y_i}, 0) \ln(\frac{p_{i,t}^{y_i}}{p_{i,t-1}^{y_i}}) + \sum_{j=1,j\neq y_i}^{k} \min(p_{i,t}^{j} - p_{i,t-1}^{j}, 0) \ln(\frac{p_{i,t}^{j}}{p_{i,t-1}^{j}})$$

$p_{i,t}^{y_i} - p_{i,t-1}^{y_i} < 0$ unlearning

$$du_{i,t} = \min(p_{i,t}^{y_i} - p_{i,t-1}^{y_i}, 0) \ln(\frac{p_{i,t}^{y_i}}{p_{i,t-1}^{y_i}}) + \sum_{j=1,j\neq y_i}^{k} \max(p_{i,t}^{j} - p_{i,t-1}^{j}, 0) \ln(\frac{p_{i,t}^{j}}{p_{i,t-1}^{j}})$$

- 作者介绍
- 研究背景
- 本文方法
- 实验效果
- 总结反思

Results

- CIFAR

Table 1. Accuracy % on Long-Tailed CIFAR-10/-100 with Different Imbalance Ratios. All methods use the same network structure (the ResNet-32 backbone and a multi-head decision classifier).

Dataset Name	Long-tailed CIFAR 10				Long-tailed CIFAR 100			
Imbalance Ratio	1	20	50	100	1	20	50	100
Base Model	92.1	83.9	78.3	72.2	70.6	53.0	45.0	40.6
Focal Loss	92.2	83.8	78.2	72.6	70.8	53.1	45.6	41.0
Class-Balance Loss	92.1	84.1	79.1	74.3	70.6	54.9	46.1	41.1
Our Method	**93.8**	85.5	80.2	75.0	**71.5**	54.5	48.0	42.3
TDE	91.1	84.7	82.1	79.1	67.8	54.5	48.5	43.5
TDE + Our Method	93.5	**87.2**	**84.5**	**79.6**	70.5	**55.9**	**50.3**	**44.9**

Results

- Different Groups

Table 2. Accuracy % on Long-Tailed CIFAR100 with Imbalance Ratio 100. The network is the same as that in Tab. 1. Class-Balance(More Minority) is another instance of Class-Balance Loss, which assigns much more weights for minority classes.

Methods	Majority	Minority	Overall
Base Model	54.1	9.0	40.6
Focal Loss	54.7	9.1	41.0
Class-Balance Loss	53.5	11.0	41.1
Class-Balance(More Minority)	49.3	**12.2**	38.2
Our Method	**56.2**	9.9	**42.3**

Results

- General Datasets

Table 4. Accuracy % on 10 Datasets. All methods use the same network structure. The "Base" here is the Logistic Regression. "CB" denotes the class re-balance loss [7].

Methods	Base	CB	Ours
Sonar	83.3	83.3	**85.7**
Balance	91.2	92.0	**92.8**
CMC	59.0	60.0	**62.4**
Ecoli	82.4	**85.3**	**85.3**
Glass	34.9	44.2	**53.5**
Heart	**72.2**	**72.2**	**72.2**
Iris	93.3	93.3	**96.7**
Robot	93.5	94.0	**94.1**
Seeds	**97.6**	**97.6**	**97.6**
Wine	**41.7**	**41.7**	**41.7**
Average	74.9	76.4	**78.2**

Simulation

Table 3. Accuracy % on Long-Tailed MNIST with Imbalance Ratio 100 for Simulation Binary Classification. All methods use the same network structure. The Base Model is the Multilayer Perceptron. Two ratio columns present the class imbalance ratio(*i.e.*, "Class Ratio") and the sub-class imbalance ratio(*i.e.* "Sub-Class Ratio"). Especially, the values in these two columns are the imbalance ratios that calculated after re-balancing. The "Major" and "Minor" represent the majority sub-classes and the minority sub-classes within a class.

Methods	Class Ratio	Sub-Class Ratio	Overall	Even (Majority Class)			Odd (Minority Class)		
				Overall	Major	Minor	Overall	Major	Minor
Base Model	1.7	100.0	86.04	90.95	98.88	85.54	81.28	96.92	70.09
Class-Balance Loss	**1.2**	69.6	86.19	89.77	98.73	83.66	82.72	97.11	72.44
Our Approach	1.4	**37.3**	**87.99**	**92.33**	**99.08**	**87.73**	**83.78**	**97.41**	**77.43**

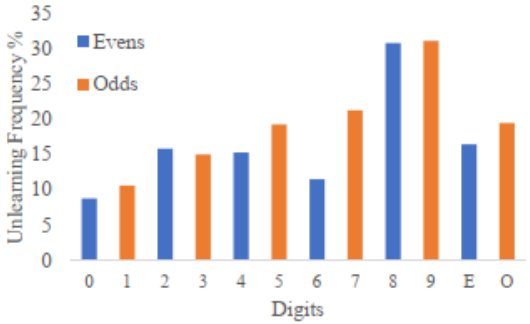
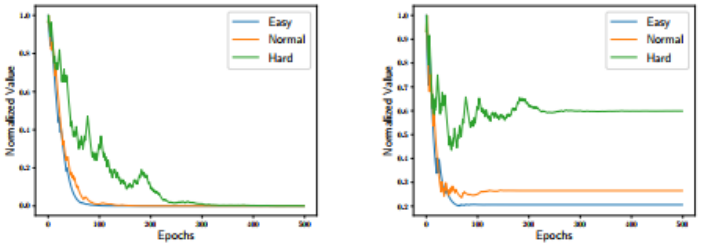


Figure 3. Unlearning Frequency of Classes and Sub-Classes. E denotes the even class. O denotes the odd class.



(a) Loss (b) Difficulty

Figure 4. Loss and Difficulty of Instances in Training. "Easy" is unlearned with 10% probability, "Normal" is unlearned with 20% probability, "Hard" is unlearned with 40% probability.

- 作者介绍
- 研究背景
- 本文方法
- 实验效果
- 总结反思

总结

- 总结
  - 实例级（难例挖掘）
  - 重采样

- 反思
  - 难度定义（vs.损失幅值/梯度幅值及其统计量）

TDE

因果分析，优化器动量项带入数据分布的影响
那么直接从logits中去掉数据分布带来的影响就可以啦

$$v_t = \underbrace{\mu \cdot v_{t-1}}_{momentum} + g_t, \qquad \theta_t = \theta_{t-1} - lr \cdot v_t,$$

$$TDE(Y_i) = \frac{\tau}{K}\sum_{k=1}^{K}\left(\frac{(w_i^k)^T x^k}{(||w_i^k|| + \gamma)||x^k||} - \alpha \cdot \frac{cos(x^k, \hat{d}^k) \cdot (w_i^k)^T \hat{d}^k}{||w_i^k|| + \gamma}\right)$$

Thanks for Attention!