NLNL: Negative Learning for Noisy Labels

Katalina Biondi

Paper by: Youngdong Kim , Junho Yim, Juseung Yun, Junmo Kim

School of Electrical Engineering, KAIST, South Korea

Overview

- Motivation
- Related Works
- Proposed Solution
- Architecture
- Experiment
- Results
- Conclusion

Motivation





“Data science becomes the art of extracting labels out of thin air”

-Eran Malach

ImageNet

- 14 Million Images
- 1,000 classes
- How long would it take to classify by hand?
- 5 sec/image
- 12 hours/day
- 162 days!!!



Atomic Visual Actions (AVA)

- 430 15-min video clips
- 1.62 M action labels
- How much would this cost to verify?
- Each action label is $0.05
- $81,000



Left: Sit, Talk to, Watch; Right: Crouch/Kneel, Listen to, Watch



Left: Stand, Carry/Hold, Listen to; Middle: Stand, Carry/Hold, Talk to; Right: Sit, Write



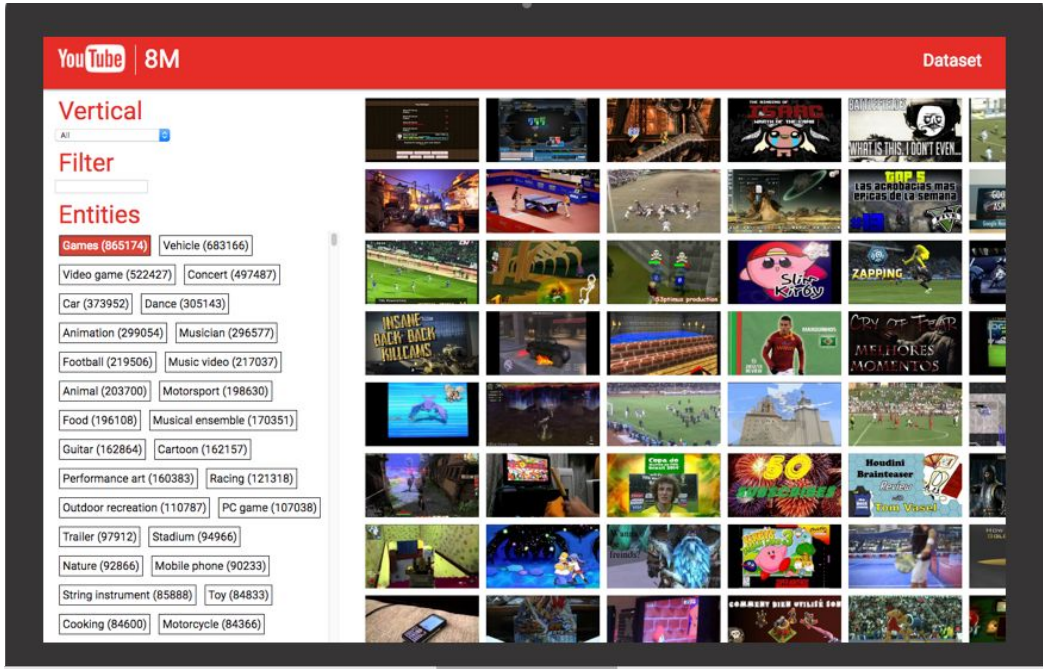
Left: Sit, Ride, Talk to; Right: Sit, Drive, Listen to



Left: Stand, Watch; Middle: Stand, Play instrument; Right: Sit, Play instrument

YouTube 8M

- 6.1 M Videos
- 350,000 hours
- 3862 Classes
- Metadata



Yahoo Flickr Creative Commons 100 Million (YFCC100m)

- 100 M media objects
- 99.2 M Photos
- 0.8 M Video
- Metadata

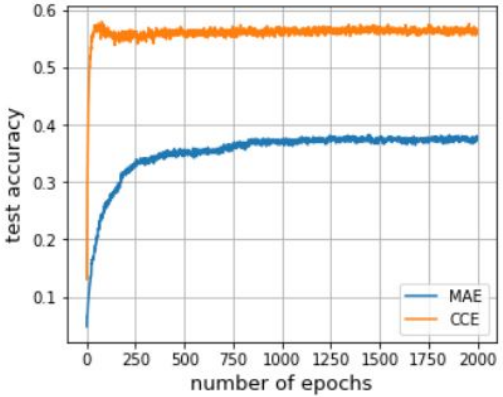


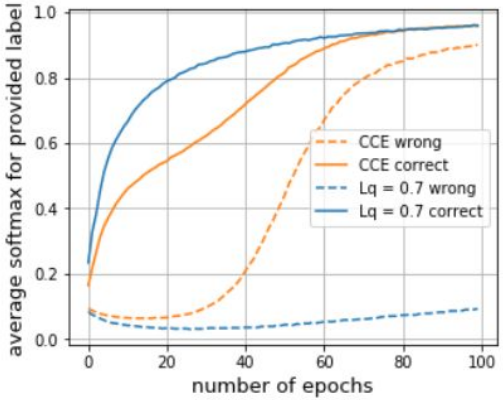
Related Works

1. Loss Functions
2. Correction Methods
3. Sample Pruning

Loss Functions

- MAE
- CCE
- L_q
- Truncated L_q





Label Correction

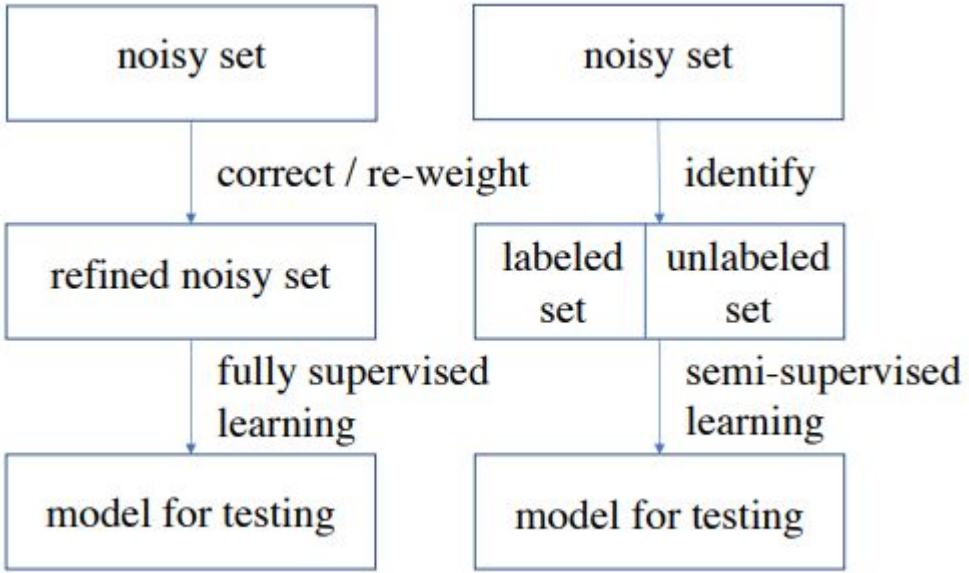
- Correct Directly
- Re-Weight
- Backwards Loss Correction

$$\mathfrak{L}^{\leftarrow}(p(y|x)) = T^{-1}\mathfrak{L}(p(y|x))$$

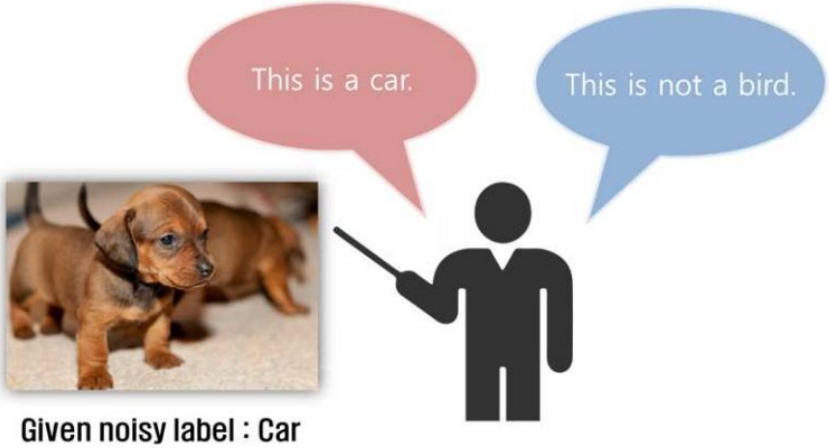
- Forward Loss Correction

$$\mathfrak{L}^{\rightarrow}(p(y|x)) = \mathfrak{L}(T^{T}p(y|x)))$$

Sample Pruning



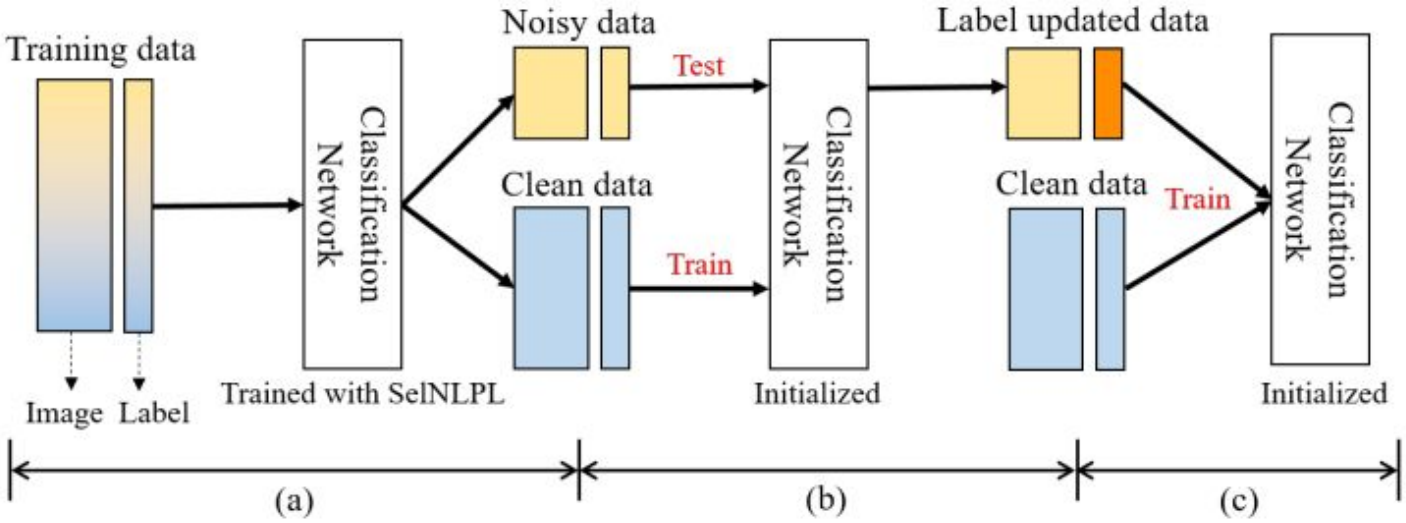
Suggested Solution - Negative Learning



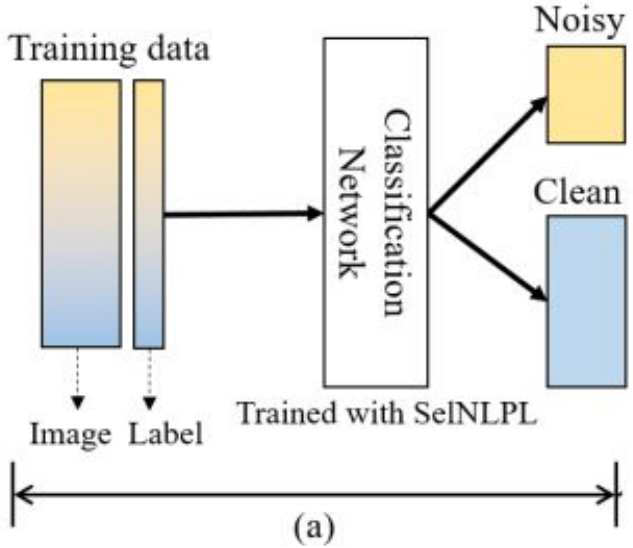
Proposed Solution

- Utilizing the proposed NL
- Selective Negative Learning and Positive Learning (SelNLPL) for filtering
- Semi-supervised learning

Architecture



SelNLPL -Division of Training Data



NL vs PL

PL - “input image belongs to this label.”
NL - “input image does not belong to this complementary label.”

Algorithm 1 Complementary label generation

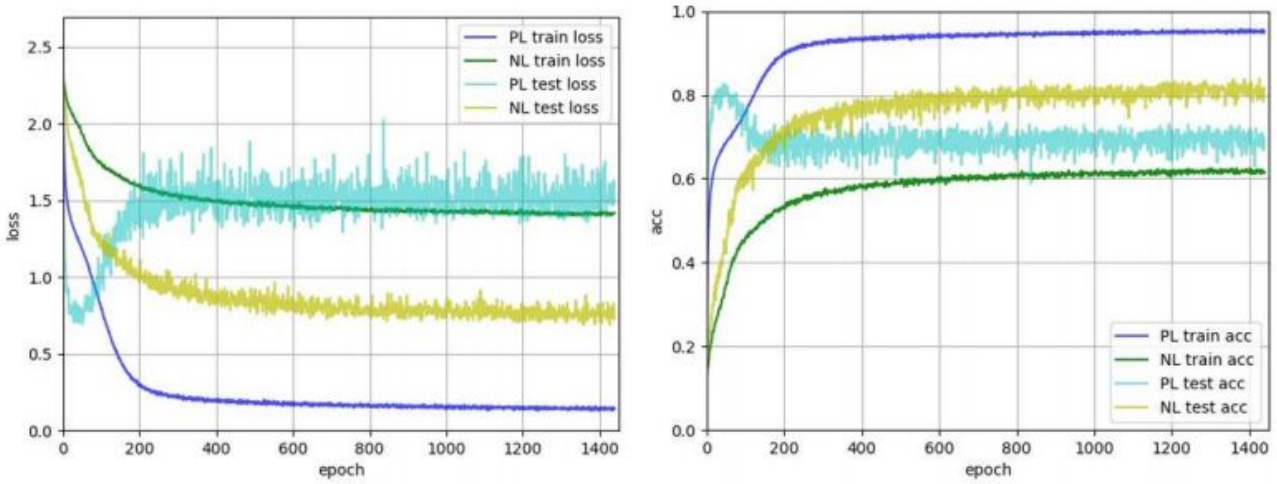
Input: Training label $y \in \mathcal{Y}$

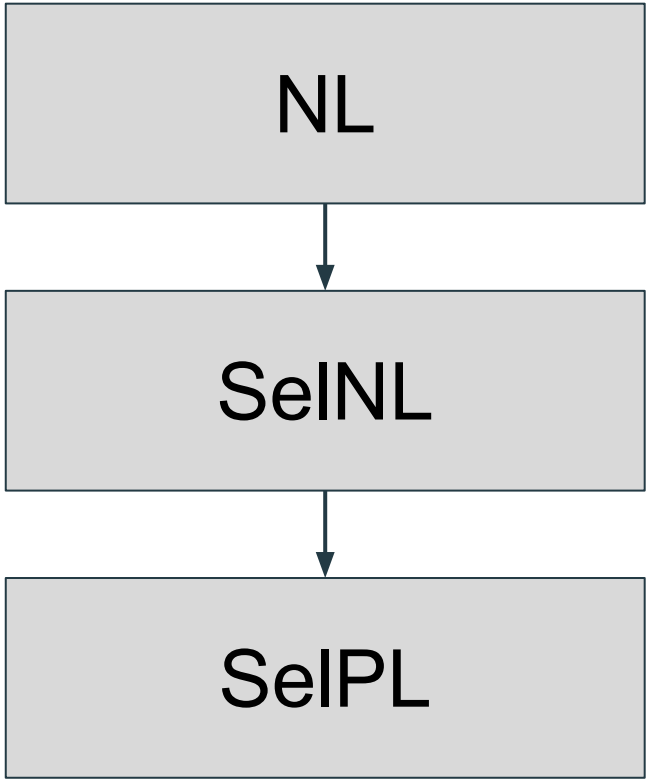
while iteration **do**

$\overline{y}$ = Randomly select from $\{1, ..., C\} \setminus \{y\}$

Output: Complementary label $\overline{y}$

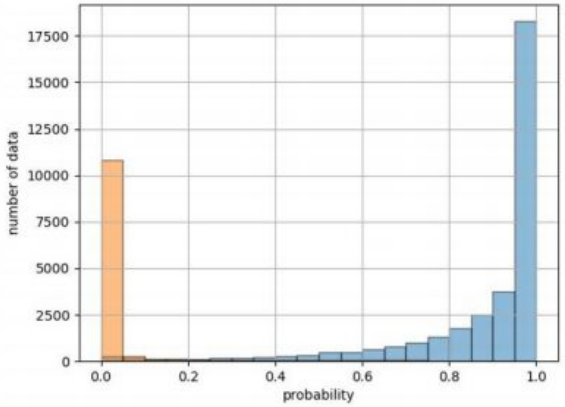
NL vs PL


PL train loss
NL train loss
PL test loss
NL test loss
loss
epoch
0.0
0.5
1.0
1.5
2.0
2.5
0
200
400
600
800
1000
1200
1400
PL train acc
NL train acc
PL test acc
NL test acc
acc
epoch
0.0
0.2
0.4
0.6
0.8
1.0

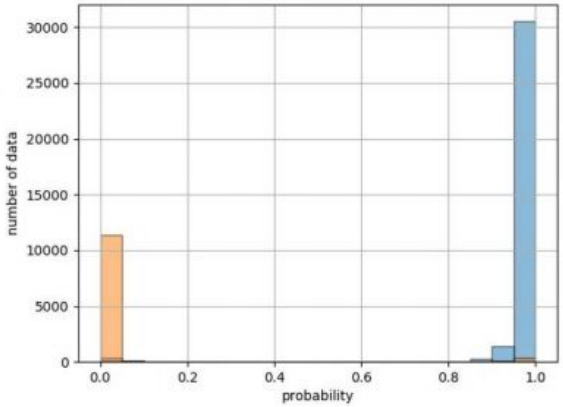

NL
SelNL
SelPL

SelNL and SelPL

- SelNL trains the CNN only on data with confidence over $\frac{1}{c}$
- SelPL trains the CNN only on data with confidence over γ

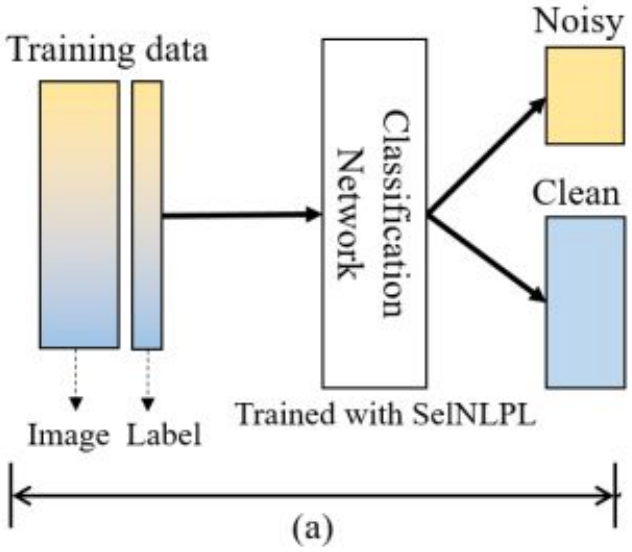


NL → SelNL



NL → SelNL→ SelPL

SelNLPL



Algorithm 2 Overall process of SelNLPL

Input: Training data $(\boldsymbol{x}, y) \in (\mathcal{X}, \mathcal{Y})$, network $f(\boldsymbol{x}; \boldsymbol{\theta})$, total epoch T

for $i \leftarrow 1$ to T **do** ▷ NL
  Batch $\leftarrow$ Sample $\boldsymbol{x}$
  Update f by minimizing Eq. 2

for $i \leftarrow 1$ to T **do** ▷ SelNL
  Batch $\leftarrow$ Sample $\boldsymbol{x}$ if $\boldsymbol{p}_y > 1/\mathrm{c}$
  Update f by minimizing Eq. 2

for $i \leftarrow 1$ to T **do** ▷ SelPL
  Batch $\leftarrow$ Sample $\boldsymbol{x}$ if $\boldsymbol{p}_y > \gamma$
  Update f by minimizing Eq. 1

Output: Network $f(\boldsymbol{x}; \boldsymbol{\theta})$

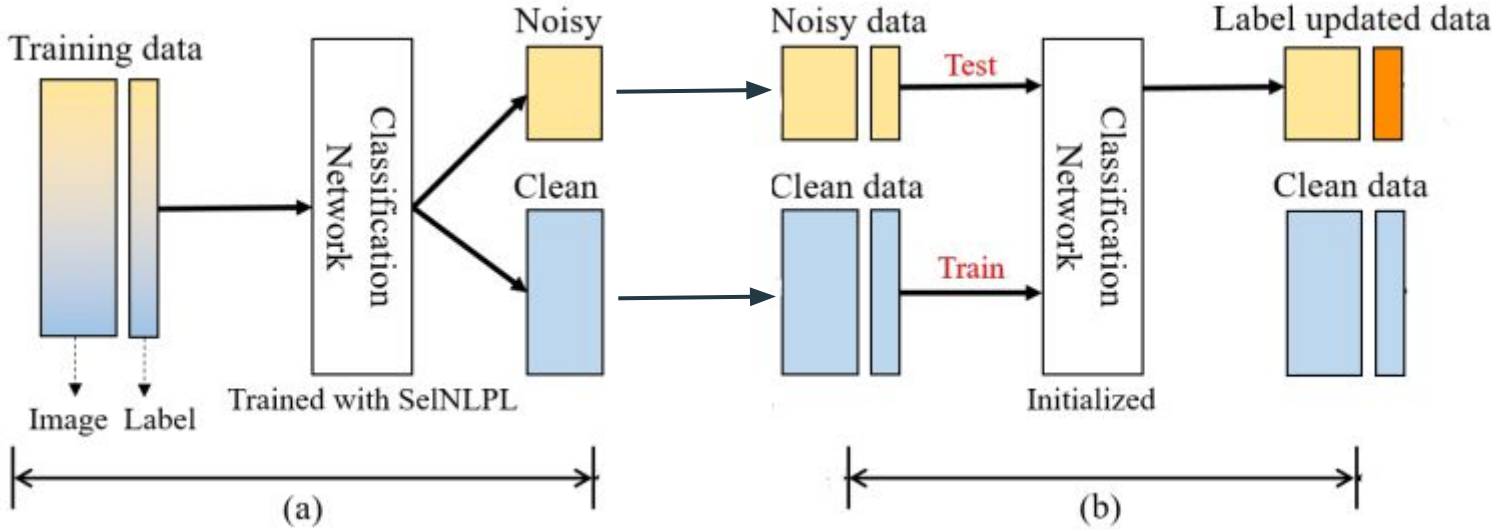
Loss

$$\mathcal{L}(f, y) = -\sum_{k=1}^{c} \boldsymbol{y}_k \log \boldsymbol{p}_k \tag{1}$$

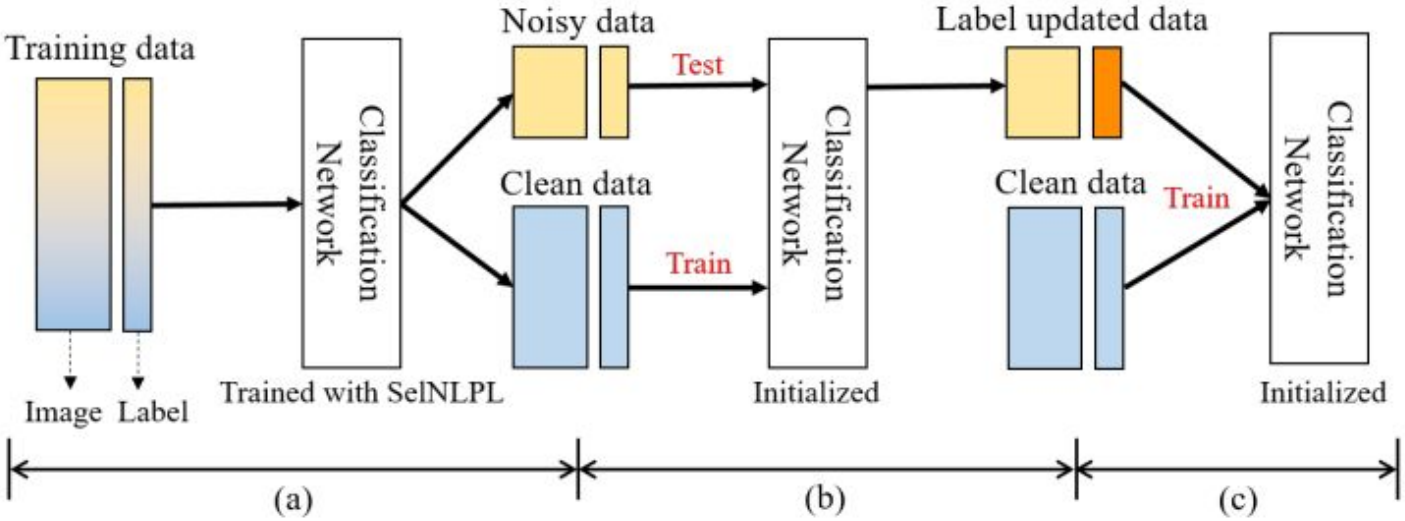
$$\mathcal{L}(f, \overline{y}) = -\sum_{k=1}^{c} \overline{\boldsymbol{y}}_k \log(1 - \boldsymbol{p}_k) \tag{2}$$

where $\boldsymbol{p}_k$ denotes the kth element of $\boldsymbol{p}$.

Pseudo labeling for semi-supervised learning



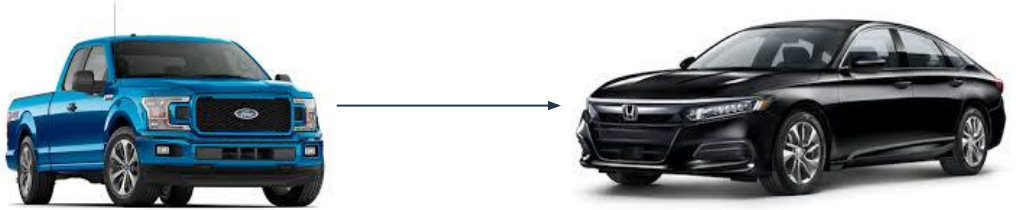
Training initialized CNN



Asymmetric Noise

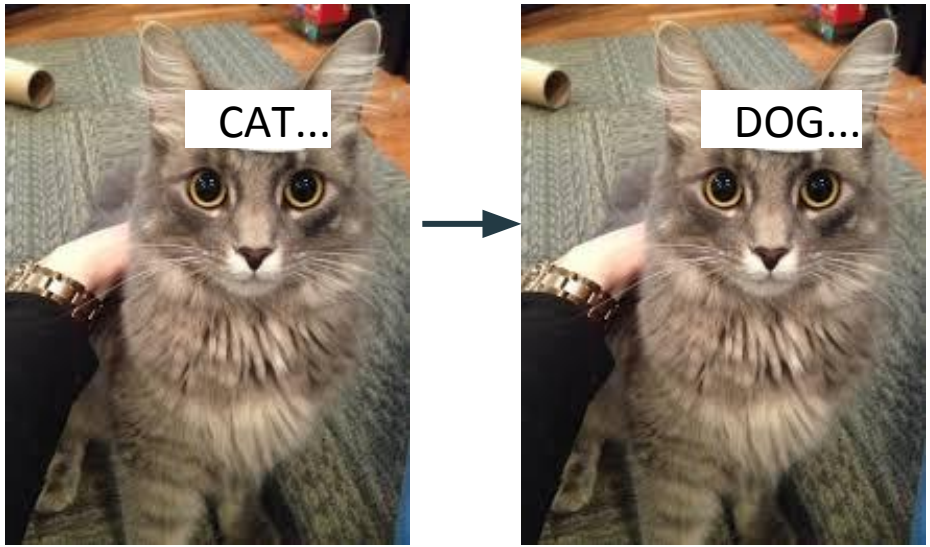
- **Asymmetric noise** - mimics real errors for similar classes





Symmetric noise

- **Symm-inc noise** - randomly selecting the label from all classes

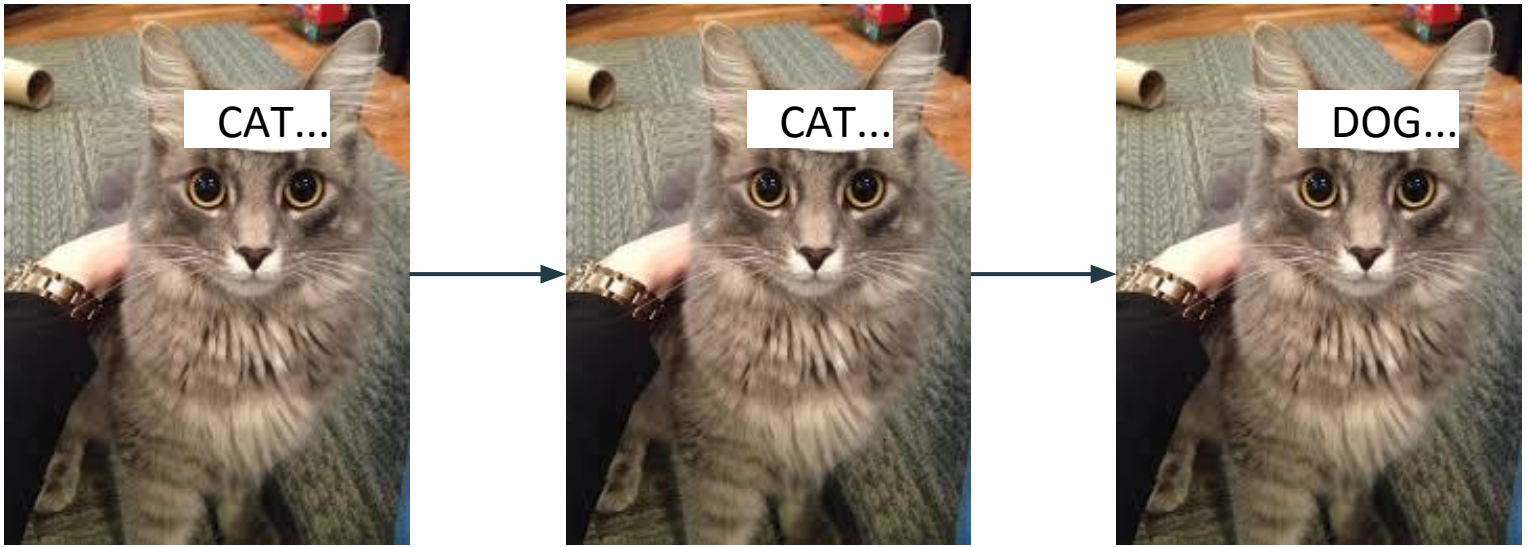


Before

After

Symmetric noise

- **Symm-exc noise** - flips the ground truth label to one of the other classes



Experiments - Datasets
MNIST

- training set of 60,000
- test set of 10,000
- 10 classes
- 28x28 B&W Images
- Trained on LeNet and FC2



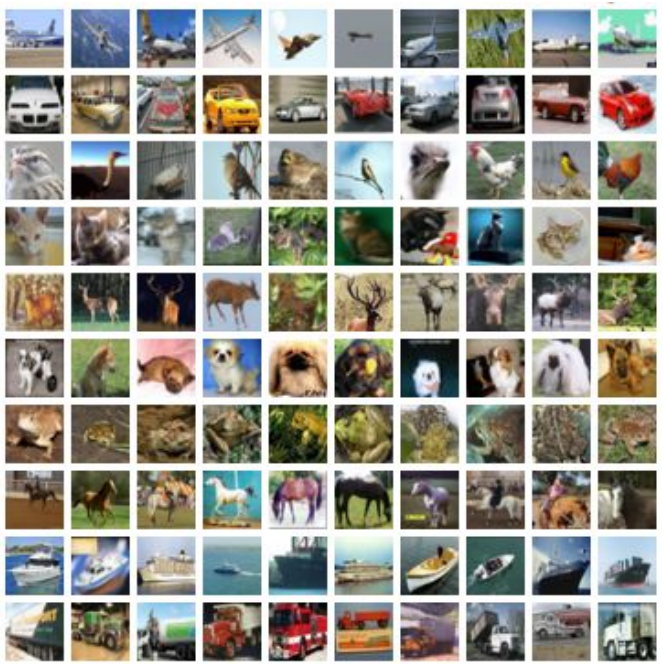
Experiments - Datasets
FashionMNIST

- Zalando's article images
- training set of 60,000
- test set of 10,000
- 10 classes
- 28x28 grayscale image
- Trained on ResNet18



Experiments - Datasets *CIFAR10 and CIFAR100*

- 50,000 training images
- 10,000 test images
- 32x32 colour images
- CIFAR 10 - 10 Classes
- CIFAR100 - 100 classes
- CIFAR100 has 20 *superclasses
- CIFAR10 Trained on ResNet34, Pre-ResNet32, ResNet14
- CIFAR100 Trained on ResNet34



* Superclass
aquatic mammals

Classes
beaver, dolphin, etc

Metrics

Metrics	
Accuracy	$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$
Estimated Noise	Amount of data that was not trained with PL.
Recall	$Recall = \frac{TP}{TP + FN}$
Precision	$Precision = \frac{TP}{TP + FP}$

Experiments

- 4 Baselines: Zang, Tanaka, Ma, Ding
- Stochastic Gradient Descent
- Momentum of 0.9
- Batch size = 128
- 720 epochs

Comparison with Results Reported by Zhang

Datasets	Model	Methods	*Symm*				*Asymm*			
			20	40	60	80	10	20	30	40
FashionMNIST	ResNet18	CE	93.24	92.09	90.29	86.20	94.06	93.72	92.72	89.82
		MAE [3]	80.39	79.30	82.41	74.73	74.03	63.03	58.14	56.04
		Forward T [25]	93.64	92.69	91.16	87.59	94.33	94.03	93.91	93.65
		Forward $\hat{T}$ [25]	93.26	92.24	90.54	85.57	94.09	93.66	93.52	88.53
		L_q [39]	93.35	92.58	91.30	88.01	93.51	93.24	92.21	89.53
		Truncated L_q [39]	93.21	92.60	91.56	88.33	93.53	93.36	92.76	91.62
		Ours	**94.82**	**94.16**	**92.78**	-	**95.10**	**94.88**	**94.66**	**93.96**

Comparison with Results Reported by Zhang

Datasets	Model	Methods	*Symm*				*Asymm*			
			20	40	60	80	10	20	30	40
CIFAR10	ResNet34	CE	86.98	81.88	74.14	53.82	90.69	88.59	86.14	80.11
		MAE [3]	83.72	67.00	64.21	38.63	82.61	52.93	50.36	45.52
		Forward T [25]	88.63	85.07	79.12	64.30	91.32	90.35	89.25	88.12
		Forward $\hat{T}$ [25]	87.99	83.25	74.96	54.64	90.52	89.09	86.79	83.55
		L_q [39]	89.83	87.13	82.54	64.07	90.91	89.33	85.45	76.74
		Truncated L_q [39]	89.70	87.62	82.70	67.92	90.43	89.45	87.10	82.28
		Ours	**94.23**	**92.43**	**88.32**	-	**94.57**	**93.35**	**91.80**	**89.86**

Comparison with Results Reported by Zhang

Datasets	Model	Methods	*Symm*				*Asymm*			
			20	40	60	80	10	20	30	40
CIFAR100	ResNet34	CE	58.72	48.20	37.41	18.10	66.54	59.20	51.40	42.74
		MAE [3]	15.80	9.03	7.74	3.76	13.38	11.50	8.91	8.20
		Forward T [25]	63.16	54.65	44.62	24.83	**71.05**	**71.08**	**70.76**	**70.82**
		Forward $\hat{T}$ [25]	39.19	31.05	19.12	8.99	45.96	42.46	38.13	34.44
		L_q [39]	66.81	61.77	53.16	29.16	68.36	66.59	61.45	47.22
		Truncated L_q [39]	67.61	62.64	54.04	29.60	68.86	66.59	61.87	47.66
		Ours	**71.52**	**66.39**	**56.51**	-	70.35	63.12	54.87	45.70

Comparison with Results Reported by Tanaka

Datasets	Model	Methods	*Symm*				*Asymm*			
			10	30	50	70	10	20	30	40
CIFAR10	Pre-ResNet32	CE	87.0	72.2	55.3	36.6	89.8	85.4	81.0	75.7
		Forward [25]	-	-	-	-	91.7	89.7	88.0	86.4
		CNN-CRF [31]	-	-	-	-	90.3	86.6	83.6	79.7
		Joint [30]	92.9	91.5	89.8	86.0	93.2	92.8	92.4	**91.7**
		Joint* [30]	-	-	-	-	93.05	92.60	91.59	89.23
		Joint** [30]	-	-	-	-	93.26	93.06	92.04	90.65
		Ours	**94.25**	**93.42**	**91.45**	**86.13**	**94.12**	**93.44**	**92.56**	90.99

Comparison with results reported by Ma

Datasets	Model	Methods	*Symm* 20	*Symm* 40	*Symm* 60
MNIST	LeNet	CE	88.02	68.46	45.51
		Forward [25]	96.45	94.90	82.88
		Backward [25]	90.12	70.89	52.83
		Boot-hard [26]	87.69	69.49	50.45
		Boot-soft [26]	88.50	70.19	46.04
		D2L [22]	98.84	98.49	94.73
		Ours	**99.35**	**99.27**	**98.91**

Comparison with Results Reported by Ding

Datasets	Model	Methods	*Symm*	*Asymm*	
			20	20	60
CIFAR10	ResNet14	CE	83.7	85.0	57.6
		Unhinged (BN) [32]	84.1	83.8	52.1
		Sigmoid (BN) [4]	66.6	71.8	57.0
		Savage [23]	77.4	76.0	50.5
		Bootstrap soft [26]	84.3	84.6	57.8
		Bootstrap hard [26]	83.6	84.7	58.3
		Backward [25]	80.4	83.8	66.7
		Forward [25]	83.4	87.0	74.8
		Semi [2]	84.5	85.6	**75.8**
		Ours	**89.85**	**90.1**	74.44

Comparison with Results Reported by Ding

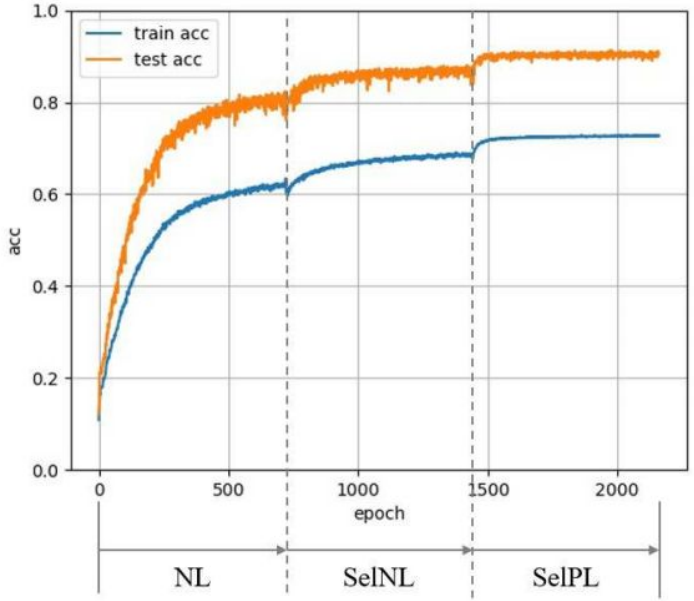
Datasets	Model	Methods	*Symm*	*Asymm*	
			20	20	60
MNIST	FC2	CE	96.9	97.5	53.0
		Unhinged (BN) [32]	96.9	97.0	71.2
		Sigmoid (BN) [4]	93.1	96.7	71.4
		Savage [23]	96.9	97.0	51.3
		Bootstrap soft [26]	96.9	97.5	53.0
		Bootstrap hard [26]	96.8	97.4	55.0
		Backward [25]	96.9	96.7	67.4
		Forward [25]	96.9	97.7	64.9
		Semi [2]	97.7	97.8	**83.4**
		Ours	**97.96**	**97.97**	79.48

Analysis of Significance of Each Step

		30% *symm-inc* noise			
		Accuracy	Estimated noise (%)	Recall	Precision
#1	NL-SelNL-SelPL	93.82	27.60	96.28	94.01
#2	NL-SelNL	92.44 (-1.38)	33.76	98.47	78.62
#3	NL-SelPL	93.41 (-0.41)	28.17	97.04	92.84
#4	NL	87.32 (-6.5)	52.24	99.80	51.49

50% *symm-inc* noise			
Accuracy	Estimated noise (%)	Recall	Precision
91.17	45.15	95.80	95.38
89.36 (-1.81)	52.75	98.56	83.99
72.91 (-18.26)	54.35	92.12	76.19
72.53 (-18.64)	89.21	99.99	50.38

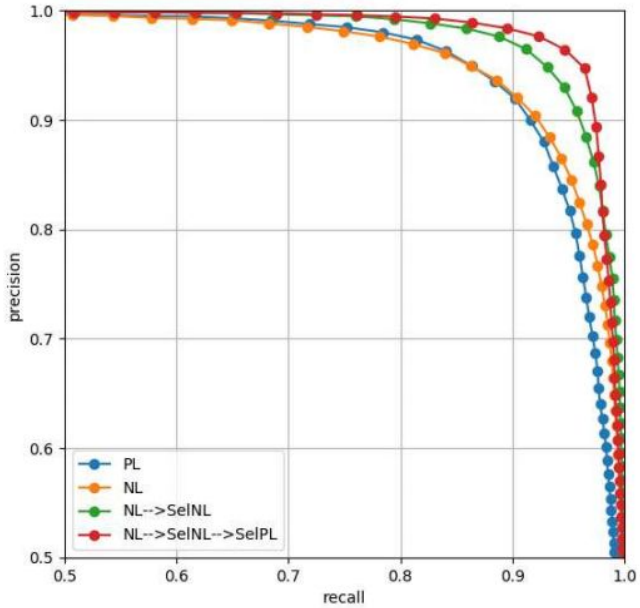
Analysis of Significance of Each Step



Filtering Ability

		Estimated noise (%)	Recall	Precision
Noise (%)	9	10.25	92.87	85.20
	27	27.60	96.28	94.01
	45	45.15	95.80	95.38

Filtering Ability



Conclusion

- performed successful noisy data classification by using semi-supervised learning (pseudo labeling) based on the filtering result of SelNLPL

Questions

