Using Soft Labeling to Defend Against Adversarial Attacks

SANNA TYRVÄINEN, KEEGAN LENSINK, OPHIR GREIF, and ELDAD HABER, University of British Columbia, Canada

LUIS TENORIO, Colorado School of Mines, USA

Adversarial attacks on deep neural networks and the instability of such networks with respect to noise are important problems in practical machine learning applications. In recent years, much attention has been given to the design of architectures and procedures that help circumvent or at least reduce the effect of adversarial attacks. In this work, we take a different route and present a novel approach for dealing with such attacks by soft-labeling the data. We show that appropriately designed soft labels can help reduce the effects of adversarial attacks. We provide examples using a newly collected soft-label CIFAR-10 data set with images that are not intuitively ambiguous. Using this data set, we analyze hard- and soft-labeled data. Results show that models trained with soft-labeled data are more resilient to targeted adversarial attacks than models trained with hard-labeled data.

Additional Key Words and Phrases: data sets, neural networks, classification, soft labels

1 INTRODUCTION

Image classification is one of the core tasks of computer vision problems with many practical applications that range from face recognition to autonomous vehicles, [1, 8, 24]. As a result of its wide use, many academic as well as industrial data sets have been proposed to test different algorithms, [5, 13, 14].[1]

Most image learning tasks are designed around so-called hard labels where there is either a single true class or multiple equally true classes for each image. These labels can be represented as vectors whose length is the number of classes, where the true classes have the value 1 and the others 0. When using a single class for each image, it is assumed that the classes are disjoint, well defined, and exhaustive.

Another type of data for image-related tasks uses soft labels. In these applications, the class of the image can be thought of as ambivalent. Soft labels can be defined as vectors of length equal to the number of classes K: $\boldsymbol{y}_i = \left(y_i^1, \dots y_i^K\right)$, where each value y_i^j represents how much the j^{th} class describes the image $\boldsymbol{x}_i$. We assume that $0 \leq y_i^j \leq 1$, for all $j = 1, \dots K$ and require that $\sum_{j=1}^{K} y_i^j = 1$. This way the soft labels can be interpreted as label distributions such as in [6], and the hard labels can be seen as special cases of these distributions. Examples for this class of problems are the classification of multilabel *Nature Scenes* [6], emotion description [6, 22], and age estimation [4].

Beyond the natural setting for the problems above, the use of soft labels has many advantages, such as their connection to robustness [17, 25] and ability to store and present information. An application of this is the teacher-student models, where the soft labels are generated by the teacher model that distills information about the data and the task for the smaller student model [9].

Another very important advantage of soft labels that we explore in this work is the ability of soft-labeled training to be more resilient to adversarial attacks. As adversarial attacks have been in the center of robust training in the last few

[1] 1,073 results when searching with "image classification" in Kaggle data sets, www.kaggle.com/datasets?search=image+classification (Accessed 20 May 2021)

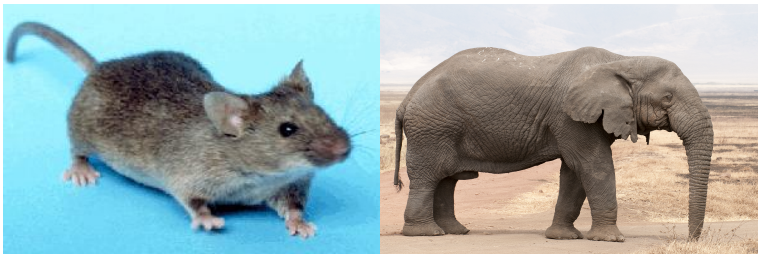


Fig. 1. Images used for image classification, with two known labels. (a) House mouse, from National Institute of Health (NIH), USA [Public domain], via Wikimedia Commons (https://commons.wikimedia.org/wiki/File:House_mouse.jpg) (b) Loxodonta africana - old bull, Ngorongoro, 2020. Photograph by Yathin S Krishnappa (CC BY-SA 3.0), vai Wikimedia Commons (https://commons.wikimedia.org/wiki/File:Loxodonta_africana_-_old_bull_(Ngorongoro,_2009).jpg).

years, developing data collection strategies that reduce the effect of such attacks is crucial. The methodology developed in this paper and the newly collected data set are designed to show that soft-labeled data collection and training aid in the defence of such attacks, even when the data do not seem to be ambivalent.

In our methodology we seek a way to define the label of an image, any image, in a way that encodes some of its ambivalent nature. We propose a novel way to define the label of an image by asking the participants to assess the content of the label classes in the image. In this way every image has multiple labels and can benefit from the properties of soft-labeled images. In a way, this representation is close in spirit to Bayesian labeling, where the content of the image represents the amount of information about a particular label potentially present in the image.

To give a concrete example, consider Figure 1 where we show images of an elephant and a mouse.IN figure it is clear which class, mouse or elephant, each image belongs to, that is, they are not ambivalent if the only question we ask is "what is in the image". However, a different question that can be asked is the amount of information that defines a mouse or an elephant in each of these images. The answer to this question can be quite different. The long trunk of the elephant and the long tail of the mouse can be similar. The color of the animals is also similar, and therefore one may say that the elephant's image has some "mouse content" and vice versa. Collecting such data allows us to include soft-labeled information, even for images that seem to belong to one class, and to exploit the benefits of soft labels.

We thus present a new image data set that is not naturally soft-labeled and a labeling strategy that allows us to achieve this goal. The images belong to one true class, but the survey participants were asked to estimate the information content for each label for each image. In this way we collect image-specific labels that follow human-like labeling processes and add new information about the taxonomy of the classes. When training a deep neural network (DNN) with an appropriate loss, such an enriched data set improves the model's robustness against targeted adversarial attacks.

The rest of the paper is organized as follows. We review some related work in Section 2. To further motivate our study and collection of data sets with soft labels, in Section 4 we present a simple example that illustrates the differences between hard and soft labels when learning a machine learning model. In Section 5 we present the graphical user interface (GUI) that was designed to collect participant estimates on how the available labels fit for each image. In Section 6 we show how the use of soft labels differs from hard labels in training a network, and illustrate how soft labels perform under adversarial attacks.

2 RELATED WORK

Soft labels have been a topic of interest in machine learning (ML) since the early days. Keller and Hunt [12] studied fuzzy labels, and Waegeman *et al.* [27] proposed multiple supervised learning algorithms for soft labels. These methods were mainly one-versus-one and one-versus-all classification methods that were not considered deep learning. A general Label Distribution Learning (LDL) framework for data learning with ambiguous labels was proposed [6]. This work introduces soft-labeled LDL data sets and ML algorithms to use with the data sets, and compare different ways to assess the performance of these algorithms.

In [18], a combination of hard and soft labels were used to add auxiliary information into a binary classification problem. The training was done with a data set of three components: an input, a class label, and a confidence of the class, $(\boldsymbol{x}_i, \boldsymbol{y}_i, p_i)$. This approach was further studied in [19, 30, 31].

Because data sets with smooth labels can be difficult to obtain, many label-smoothing methods have been proposed. Label smoothing regularization (LSR) [23] is an output-regularization method for training DNN. For the case with K categories, a smoothing variable ϵ is chosen and the true hard label $\boldsymbol{y}$ is softened by reducing its weight by the factor $1-\epsilon$ and adding the same fraction ϵ/K to each category:

$$\widehat{\boldsymbol{y}} = (1-\epsilon)\boldsymbol{y} + \epsilon/K. \tag{1}$$

Related to LSR is a method proposed in [20] that penalizes outputs with low entropy. The motivation is similar to LSR, hard labels have zero entropy and the regularization prevents the outputs from getting too close to hard labels and overfitting the model.

Soft labels are also used to store and present information. For example, in knowledge distilling, hard-labeled data are used to train large DNN's [9]. The soft labels from this network are then used with the original data to train a smaller model. The authors argue that the probabilities assigned by the model to incorrect categories provide information about the model that is just as valuable as the probabilities of correct categories. Incorrect and correct here refer to the ground truth of the data set. Soft labels can transfer this information about the data, the model, and the task. Another method known as Label Refinery has been applied to the object labeling problem. Multiple networks are trained sequentially, and soft labels provided by a network are used to train the next network [2].

Label softening has recently received a lot of attention in medical image segmentation, [10, 11, 15, 26]. The increased interest comes from the need to overcome the variability introduced by experts' opinions and the need to use methods that provide uncertainty for medical professionals [3]. The softening can be applied on the boundaries of segmentations used in the training of the model, [15, 26]. Kats *et al.* [11] also discusses collecting data sets that include uncertainty estimates for the labels.

2.1 Our contribution

In this paper we develop a new labeling strategy that asks the users to provide content information beyond the label. We develop a procedure to collect the data and use it for training. We further demonstrate that such data yield more robust models that closely mimic human intuition about the classification problem and are therefore much more robust against adversarial attacks.

3 FORMULATION OF ADVERSARIAL ATTACKS

There are different ways to perform an adversarial attack. For a comprehensive review, see work by Wiyatno *et al.* [28]. Here we assume that a ML model h_θ has been chosen, where θ is a vector of parameters of the model. The model is trained to determine a label $\boldsymbol{y}$ for each input $\boldsymbol{x}$.

In an adversarial attack, the attacker tries to sabotage the model's performance in the pairing task. We focus on attacks, where the attacker ads a perturbation $\delta(\boldsymbol{x})$ to the input $\boldsymbol{x}$ chosen by solving the maximizing problem

$$\max_{\delta(\mathbf{x})\in\Delta} \ \ell\left(h_\theta\left(\boldsymbol{x}+\delta(\mathbf{x})\right),\boldsymbol{y}\right), \tag{2}$$

where ℓ is an appropriate loss function and Δ is a set of acceptable perturbations. In this paper we focus on perturbations in the L^∞-ball: $\|\delta(\boldsymbol{x})\|_\infty \leq \epsilon$ for some chosen $\epsilon > 0$.

In targeted adversarial attacks, the goal is to get the model to classify the perturbed input as a specific class y_{target}. This time we maximize the loss between the output label and the perturbed image while also minimizing the loss between the perturbed image and the target class label:

$$\max_{\delta(\mathbf{x})\in\Delta} \ \ell\left(h_\theta\left(\boldsymbol{x}+\delta(\mathbf{x})\right),\boldsymbol{y}\right) - \ell\left(h_\theta\left(\boldsymbol{x}+\delta(\mathbf{x})\right),\boldsymbol{y}_{\text{target}}\right). \tag{3}$$

The optimization (2) is done iteratively with projected gradient descent (PGD):

$$\delta(\mathbf{x}) := P\left(\delta(\mathbf{x}) + \alpha\nabla_\delta \ell\left(h_{\theta(\mathbf{x})}\left(\boldsymbol{x}+\delta(\mathbf{x})\right),\boldsymbol{y}\right)\right), \tag{4}$$

where $\alpha \in \mathbb{R}$ is the step length and P projects the new $\delta(\mathbf{x})$ into the required domain after the gradient step. The optimization problem related to the targeted attack defined in (3) can be solved using the same algorithm with a small adjustment [16].

4 SIMPLE SOFT LABEL EXAMPLE

In the mouse and elephant example above, the labeling was based on a subjective estimate, and it can be hard to estimate how soft labels would impact the training of an ML model that uses high dimensional image data set. To visualize how the extra information on the labels can impact the training of DNN, here we present a simpler classification problem with soft and hard labels.

Consider a disk of radius R on the plane and a random sample of points $\boldsymbol{x}_i = (x_{i1}, x_{i2})$, $i = 1, \ldots, N$ on the disk of radius R, so $\|\boldsymbol{x}_i\| = (x_{i1}^2 + x_{i2}^2)^{1/2} \leq R$. The datapoints are labeled according to color, which is defined as a function of the distance to the origin (see Figure 2). We define soft labels

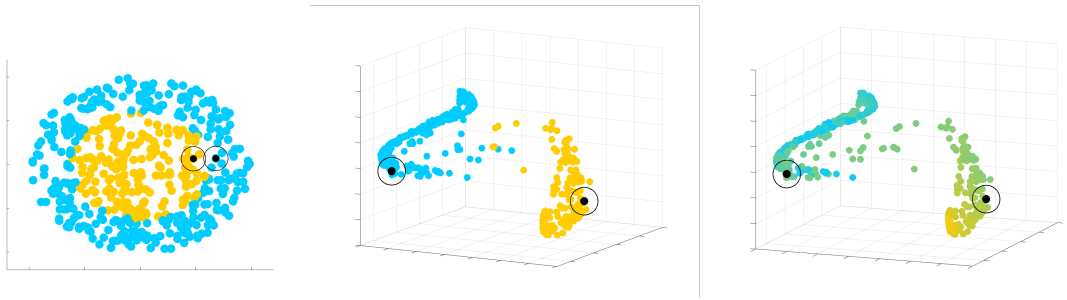
$$\begin{bmatrix} p(\boldsymbol{x}) \\ 1-p(\boldsymbol{x}) \end{bmatrix} \tag{5}$$

where $0 \leq p(\boldsymbol{x}) \leq 1$ is given by

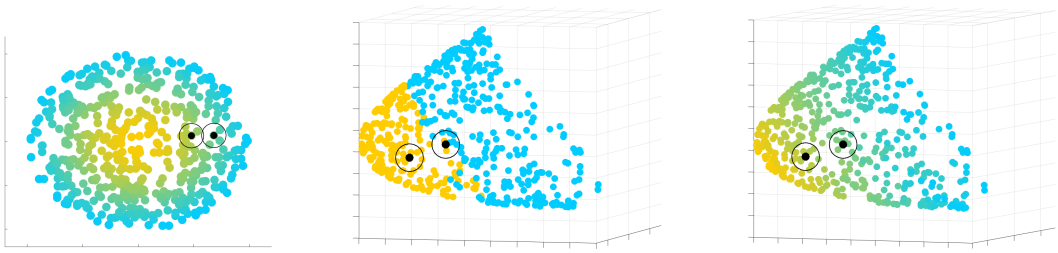
$$p(\boldsymbol{x}) = \frac{e - e^{1-\|\boldsymbol{x}\|/R}}{e-1}. \tag{6}$$

One-hot hard labels are defined by rounding the soft labels:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ for blue and } \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ for yellow.} \tag{7}$$



(a) Hard-labeled data (left) and trained model output (middle and right).



(b) Soft-labeled data (left) and trained model output (middle and right).

Fig. 2. Left: Data on $\mathbb{R}^2$ disk. Middle: Model outputs on $\mathbb{R}^3$ before a classifying layer presented with hard labels. Right: Same model outputs presented with soft labels. Two selected points from the disk (marked with black) where they are transformed in the two models.

The soft labels are more yellow or blue the further they are from the decision boundary, where the labels are close to $[0.5, 0.5]$. Note that the decision boundary is in the same location with soft and hard labeled data in $\mathbb{R}^2$.

To classify the points we trained ResNet [7] with six hidden layers. The network transforms the data from $\mathbb{R}^2$ into $\mathbb{R}^3$, so it can be linearly separated and classified. For the training process the network weights are initialized as normally distributed random matrices. The optimization is done using Newton conjugate gradient with a maximum step length of 0.1 and 50 iterations. We use a weight decay of 10^{-5}.

When training DNN for image classification it is common to use cross-entropy as loss function. With soft labels we compare two distributions; one defined by the subjective probabilities provided by the soft label, and the other defined by probabilities obtained using the DNN. We therefore use Kullback-Leibler divergence (relative entropy). Figure 2 shows the training sets and the outputs from the last layer of the DNNs before classification. The first plot in Figure 2a shows the hard-labeled data and show the output of the last layer. The last plot is the same as the second but colored using the function $p(\boldsymbol{x})$, where $\boldsymbol{x}$ is the original location of the point in $\mathbb{R}^2$. We see that the datapoints have moved to two separate locations with only few points left in between: yellow and blue points are separated. The third plot shows that the transition points are mainly decision boundary points that are both yellow and blue.

Figure 2b is analogous to Figure 2a but using soft labels in the model's training. We see that the trained network stretched the original cloud of points but preserved the basic topology (the interpoint distances, and relative distances to the decision boundary). Instead of classifying the points into two separate classes, the model has learned a continuum

Table 1. 2D models under L^∞ adversarial attack. Percentage that survived the attack.

model	0.5	0.25	0.1	0.01	0.001
Soft model	**42.68%**	**59.98%**	**85.24%**	98.10%	**99.82%**
Hard model	26.16%	55.08%	82.62%	**98.46%**	99.70%

Fig. 3. GUI of the data survey. The corresponding data-label is $[0, 0, 3, 2, 0, 2, 3, 0, 0, 0]$.

from yellow to blue. To better illustrate this behavior, the figures show two selected neighboring points (in black) that are near the decision boundary on the $\mathbb{R}^2$ disk. The model trained with hard labels transforms these points to be in opposite concentration regions, yellow and blue. The soft model keeps the points close to each other. If a new data point were added between the marked points, it is not clear how it would be transformed by the hard model. This makes the soft model more robust to noise and adversarial attacks: Two points with similar features have similar outputs. The soft labels provide more information than the hard labels, and the model can then learn and use such information.

To test the robustness of the two models, we performed an adversarial attack to have the models classify blue points as yellow and yellow points as blue. We used PGD with a linesearch. The percentage of samples that did not cross the decision boundary and survived the attack is shown in Table 1 with five different values of ϵ: 0.5, 0.25, 0.1, 0.01, and 0.001. The table shows that the soft model resists attacks better than the hard model with larger values of ϵ, and that the models are equally good with the smaller values. This means that under a large perturbation, the soft model is less likely to move the sample over the decision boundary.

Of course, in this simple example the radial distance was easy and natural information to use for the soft labels. More complex data can contain very complex information that may not be as easy to visualize and distance between the data samples is hard to define. This kind of information may not be easy to formulate as soft labels, but it may be worth the effort as the benefits of adding relevant and data sample-specific information are evident.

5 THE DATA SET

Since we were not able to find a data set with soft labels where the labels did not originate from hard labels or from an ML scheme based on hard-labeled data, we conducted a survey to create a data set with soft labels. As a source of

images, we used the well-known image data set CIFAR-10 [13]. CIFAR-10 is a collection of 60,000 32×32 color images separated into 10 classes: airplane, car, bird, cat, deer, dog, frog, horse, ship, and truck. Each image belongs to one of these categories. These classes are considered disjoint and exhaustive. This allowed us to compare the features and performance of the collected labels to the original hard labels. Here hard and soft labels are independent from each other, which is uncommon for soft-labeled data sets.

The survey was executed in 2019 using a Jupyter Notebook[2]. There were in total 28 participants, mainly university students. A typical participant labeled a batch of 200 images in an hour. Participants were instructed to trust their first impressions while labeling the images and to rely on their individual interpretation rather than seeking input from others. No image was labeled more than once, and both the original and new labels were saved. The final data set comprises 10,000 images.

The survey participants were shown a blurry image but were not given the original labels. Each of the participants worked alone on their computers. They rated the information content on a five-level scale from zero (meaning that there was no information about the category in the image) to four (meaning that the image has very high information content for the category in question). The participants were asked to consider each of the categories and were encouraged to mark at least three of them with something other than zero. This may have introduced a bias to the labels, but it was considered insignificant in the overall fluctuation of the answers. Note that a participant could choose an image to have very high information content for more than one category. Because the CIFAR-10 images are small they can be hard to interpret. See Figure 3 for an illustration of the GUI used in the survey and a CIFAR-10 image of a frog.

For better understand the results the results, survey answers were separated into the CIFAR-10 categories based on the corresponding CIFAR-10 hard labels of the images. Figure 5 shows an average matrix of the survey answers. Each row of the matrix represents an average answer for a group of images that have a hard label marked on the left side of the matrix. For example, the last value on the second row from the bottom is 1.16. This means that when all the survey answers on CIFAR-10 images from category 'ship' (row) are averaged, 1.16 is the answer for how much there is 'truck' (column) in those images. In Figure 4 these average answers with their standard deviations are presented as bar plots. The barplots show a variability in the survey answers, but overall answers for each category follow similar trends. Figures 5 and 4 show how the responses can be separated into responses about images of nature and images of man-made subjects. This can be explained by similarities of color and patterns inside the two categories, and by the general taxonomy: the man-made images were images of vehicles, and with nature nature images, four out of six categories represented land mammals (cat, deer, dog, horse). Frog and bird images seem to be outliers within the nature categories.

6 NUMERICAL EXPERIMENTS WITH CIFAR-10

We now use the new data set described in Section 5 to compare the performance of models trained with the soft labels, the hard labels, and the hard labels with LSR (1).

6.1 Learning

The soft data set was randomly divided into a training set of 7,000 images and a validation set of 2,000 images. The remaining 1,000 images were used for testing. Hard and soft labels were tested using the same images. We train a standard ResNet-18 with ReLu activation and Softmax function [7]. Softmax function ensures that the output label of

[2]https://jupyter.org/

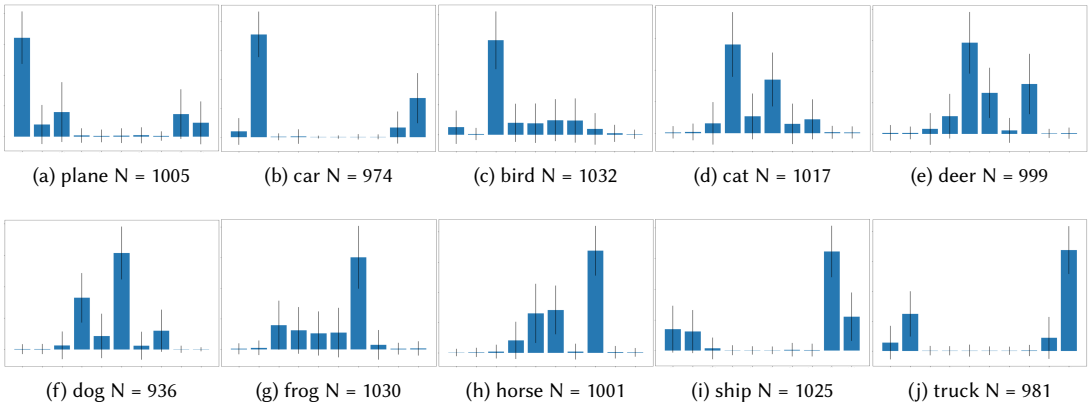


Fig. 4. Survey answers averages and standard deviations separated into the images' original CIFAR-10 categories with number of new labels N. Total number of new labels = 10,000. In each plot the bars represent the labels plane, car, bird, cat, deer, dog, frog, horse, ship and truck, in that order.

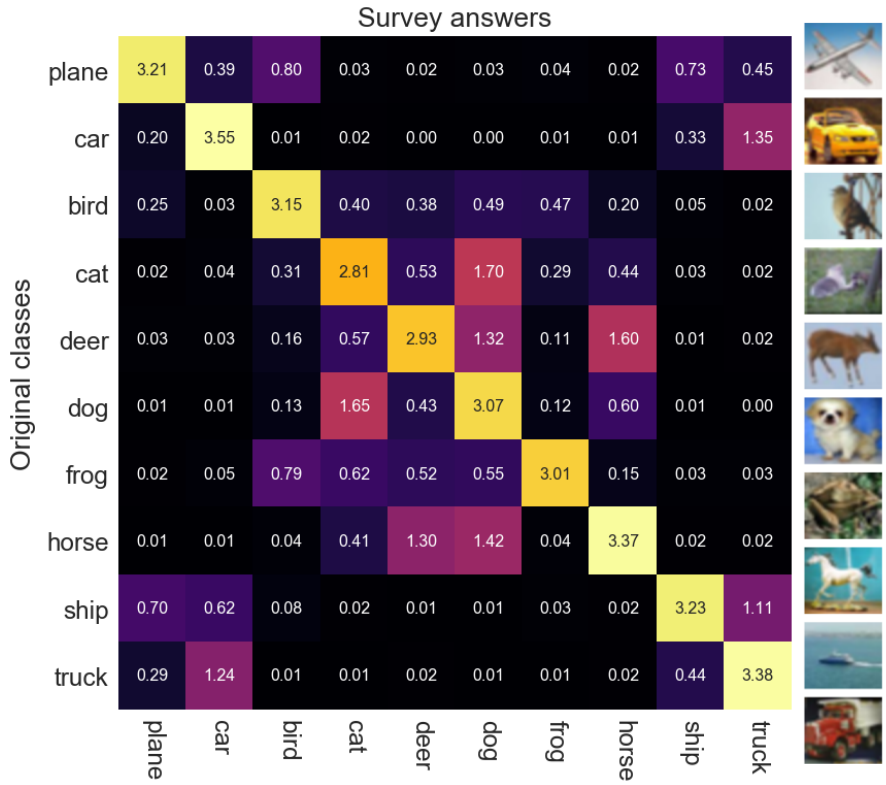


Fig. 5. Averaged survey answers. The answers (columns) are separated according to the CIFAR-10 hard labels (rows). For example, the last value on the second row from the bottom is 1.16. This means that when all the survey answers on CIFAR-10 images of 'ships' (row) are averaged, 1.11 is the answer for how much there is 'truck' (column) in those images.

Table 2. ResNet-18 model training results for test and validation sets.

model	Test loss	Val loss	Test acc	Val acc	Test l^1 err	Val l^1 err	Test l^2 err	Val l^2 err	Test l^∞ err	Val l^∞ err
Hard model	0.0559	0.0533	83.87%	84.66%	0.9046	0.8876	0.5294	0.5197	0.4155	0.4075
Soft model	0.0501	0.0515	74.29%	73.11%	**0.6316**	0.6377	**0.3149**	0.3185	**0.2385**	0.2403
LSR model	**0.0432**	0.0393	**84.48%**	85.12%	0.9147	0.8994	0.4945	0.4866	0.3855	0.3768

the model follows the requirements of soft labels, the k^{th} element of an output label $\boldsymbol{y_i}$ is defined as:

$$\hat{y}_i^k = \sigma^k(z_i) = \exp z_i^k / \sum_{j=1}^{K} \exp z_i^j, \tag{8}$$

where z_i is the output of the previous layer.

The loss of all three models was measured with Kullback-Leibler divergence, with added smoothing regularization (LSR) in one model. All models were optimized using stochastic gradient descent with weight decay of 5×10^{-4}. The learning rate was kept constant 0.1 and momentum was used with a weight 0.9. The optimization was run for 300 epochs with a batch size of 32. From now on we shall refer to ResNet-18 models trained with the soft-labeled data, the hard-labeled data, and the hard-labeled data with LSR, as soft model, hard model, and LSR model, respectively. The training results and some testing measures are shown in Table 2.

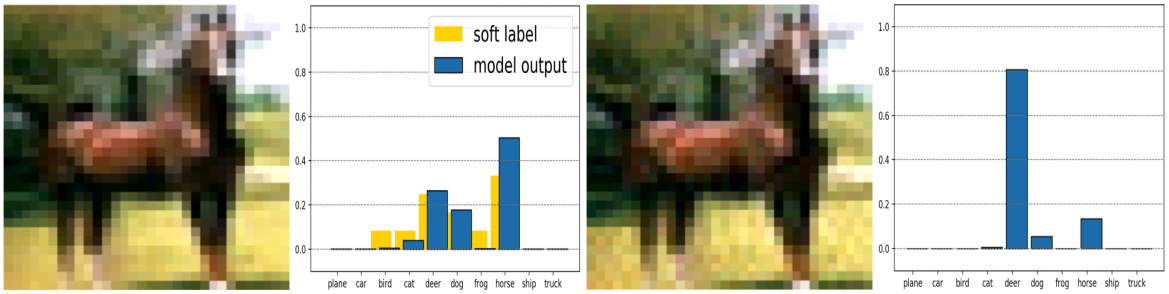
To compare the models we measure accuracy, where the output label's class with the largest value is compared to the expected labels class with the largest value. This is standard measure with hard labels and image classification but it does not take account all the information in soft labels where images that are close to a decision boundary can have two or more significant classes.

All three models have similar architecture and are trained with same methods, it is important to note that the models are still trained to do different tasks. The hard model is optimized to categorize the images in ten separate categories and takes only account the category with the largest value for each image. The LSR model and soft model have a larger output space with infinity many possible labels. The LSR model does not differentiate between the incorrect classes, where as with the soft model each element of label vector has some significance, including the ones with zero value because those tells us what the image is far from. Roughly expressed the soft model does not classify by categories but by the features of the images.

6.2 Adversarial Attack

The three models from Section 6.1 are here tested against untargeted and targeted white box (that is, the trained ML model is known to the attacker) adversarial attacks using PGD for 100 iterations with a learning rate 10 and momentum of 0.9.

For untargeted attacks, 500 random images from the CIFAR-10 test set were used. The resistance to the attacks is measured by observing the dominant class of each label, that is, the class that has the highest weight for each label. The measure is not optimal for the soft model for there may be several equal or close to equal classes, whereas with hard



(a) (b) CIFAR-10 image with added perturbation and the resulted label.

Fig. 6. Example of an untargeted adversarial attack towards the soft model with$\epsilon = 0.1$. (a) CIFAR-10 image and the soft model output with original soft label. (b) CIFAR-10 image with added perturbation and the resulted label.

labels there is only one true class. This measure still provides interesting results. The attack survival rates are presented in Table 3. A higher percentage means a higher proportion of samples retained its main label class. A range of ϵ values was used: 0.1, 0.05, 0.01, 0.005, and 0.001.

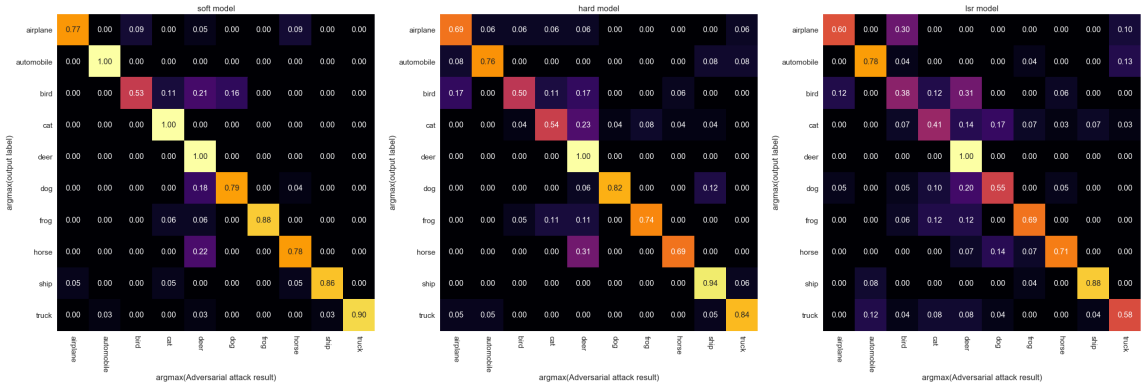
The top section in Table 3 shows that the soft model is more vulnerable against untargeted attacks with $\epsilon \geq 0.01$ than both the hard and the LRS models. These results can be partly explained by the different shapes of the output labels. For example, Figure 6 presents an image and its label from the soft model and shows how the attack appears in the image and changes the label. The original label has three dominant classes: horse, deer and dog and dog, with values 0.33, 0.25 and 0.17. In an adversarial attack, the new label becomes mostly deer and horse with values 0.81 and 0.13. This type of failing against adversarial attacks was typical with smooth labels. So the soft model might be weaker against targeted attacks when considering only the dominating class, but its failure might be more predictable than with other models.

In targeted attacks, samples from the target class were removed so that their unequal amount in the random sets would not skew the survival rates. The survival rates from the attacks with different target classes are presented in Table 3 under the rates of untargeted attacks. With most targets, the soft model performs better with most of the chosen values of ϵ. For example, average confusion matrices for targeted attacks with a target class 4 (deer) against all three models, are shown in Figure 7 and with target 8 (ship) in Figure 8.

The vertical axis in Figure 7 corresponds to the model outputs while the horizontal axis shows the results of the adversarial attack. Off-diagonal values are ratios of how much the row-class changed to the column-class category. A closer look to Figure 7 shows higher ratios in the fifth column from the left: the deer that is the target class. These matrices reveal which classes were more resistant to the attack. We can compare which classes kept 80% or more of their original label: In the hard model these classes were dog, ship, and truck, in the LSR model only ship, and in the soft model car, cat, frog, ship, and truck. These can be explained with the different relations of the classes in the data sets and in the network's training. The soft data set has a very little overlap between man-made and natural classes, whereas in the LSR model all labels are smoothed uniformly and thus classes are less separated than in other models.

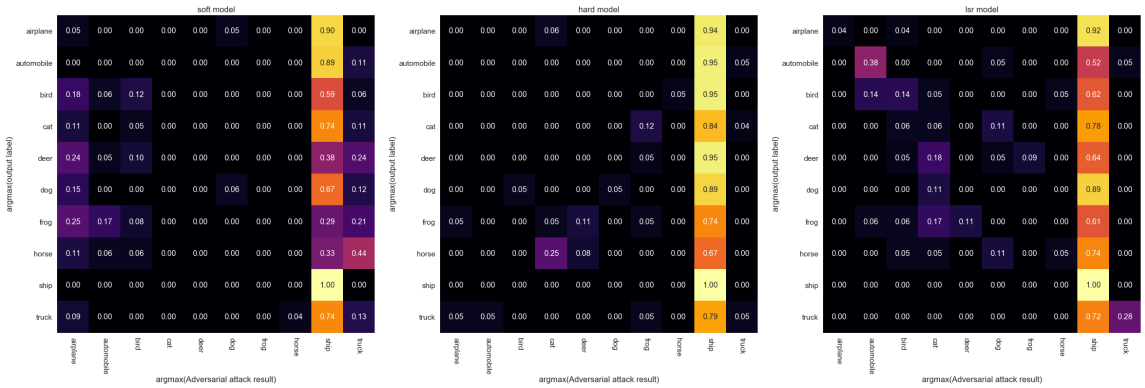
7 DISCUSSION

The image-specific soft labels stored more information about the data than traditional hard labels, including information about which images were unclear to the survey participants. These labels were then used to teach this information to



(a) Soft model, resistance 82.56%. (b) Hard model, resistance 71.48 %. (c) LSR model, resistance 61.93%.

Fig. 7. Confusion matrices for a set of 500 targeted adversarial attacks with a target class 4: deer and $\epsilon = 0.01$. For the matrices, the model output vectors and the adversarial attack output vectors have been turned to one-hot vectors and then averaged. The averages of hard vectors (columns) are separated according to the CIFAR-10 hard labels (rows). Labels on the x-axis from left to right: plane, car, bird, cat, deer, dog, frog, horse, ship, truck. Same labels on y-axis from up to down.



(a) Soft model, resistance 4.35 %. (b) Hard model, resistance 1.69 %. (c) LSR model, resistance 10.55 %.

Fig. 8. Confusion matrices for a set of 500 targeted adversarial attacks with target class 8: ship and $\epsilon = 0.05$. For the matrices, the model output vectors and the adversarial attack output vectors have been turned to one-hot vectors and then averaged. The averages of hard vectors (columns) are separated according to the CIFAR-10 hard labels (rows). Labels on the x-axis from left to right: plane, car, bird, cat, deer, dog, frog, horse, ship, truck. Same labels on the y-axis from up to down.

the model. For comparison, often in ML literature label smoothing is used to add uncertainty to the model by adding noise or removing information. There are plenty of results that adding noise or errors in the data helps to train more robust models, [25, 29]. In this work we added information to the labels.

Table 2 show how the soft model learned from the data although it didn't perform well when measuring accuracy. The hard and LSR models trained with hard classes outperformed our soft-trained model in accuracy. This might be because the hard labels train a model to categorize images in disjoint classes, whereas soft labels promote labels with multiple significant classes. Some of the training of the soft model is spent learning the continuum between the classes.

Table 3. CIFAR-10 models under untargeted and targeted L^∞ adversarial attack with multiple values of ϵ and 500 random samples. Percentage that survived the targeted attack without accounting samples from the target class.

target	**model**	0.1	0.05	0.01	0.005	0.001
no target	**Soft model**	33.6%	35.1%	76.1%	**89.1%**	**97.9%**
	Hard model	**46.1%**	**46.9%**	**77.1%**	88.7%	96.6%
	LSR model	34.9%	39.9%	76.3%	87.6%	95.7%
0: plane	**Soft model**	0.0%	**5.62%**	**81.87%**	**93.82%**	98.88%
	Hard model	0.0%	2.25%	76.54%	85.96%	**99.44%**
	LSR model	0.0%	3.35%	64.80%	81.11%	98.33%
1: car	**Soft model**	0.0%	2.21%	**80.66%**	**94.48%**	98.90%
	Hard model	0.0%	0.54%	65.05%	88.17%	**98.92%**
	LSR model	0.0%	**8.60%**	62.37%	76.34%	95.61%
2: bird	**Soft model**	0.0%	**6.37%**	**79.78%**	**92.13%**	98.50%
	Hard model	0.0%	1.14%	69.58%	85.93%	97.34%
	LSR model	0.0%	2.99%	67.16%	80.22%	**98.51%**
3: cat	**Soft model**	0.0%	**8.89%**	**79.60%**	**89.59%**	98.05%
	Hard model	0.0%	2.40%	70.31%	88.65%	97.82%
	LSR model	0.0%	6.34%	68.09%	87.99%	**98.23%**
4: deer	**Soft model**	0.0%	**7.56%**	**82.56%**	87.79%	97.67%
	Hard model	0.0%	2.87%	71.84%	**89.08%**	**99.43%**
	LSR model	0.0%	4.55%	61.93%	79.55%	96.02%
5: dog	**Soft model**	0.0%	**7.60%**	**74.27%**	**91.81%**	**1.00%**
	Hard model	0.0%	1.10%	68.00%	85.64%	98.34%
	LSR model	0.0%	3.26%	66.30%	83.15%	97.28%
6: frog	**Soft model**	0.0%	**8.89%**	**81.67%**	**91.67%**	**98.33%**
	Hard model	0.0%	1.16%	69.77%	85.47%	98.26%
	LSR model	0.0%	1.16%	60.47%	80.81%	95.93%
7: horse	**Soft model**	0.0%	**11.17%**	**81.56%**	88.83%	97.21%
	Hard model	0.0%	3.01%	71.80%	**88.72%**	**99.25%**
	LSR model	0.0%	3.93%	61.24%	78.09%	96.63%
8: ship	**Soft model**	0.0%	4.35%	**83.70%**	**90.22%**	97.83%
	Hard model	0.0%	1.69%	68.36%	88.70%	**99.44%**
	LSR model	**1.67%**	**10.55%**	67.22%	83.89%	98.33%
9: truck	**Soft model**	0.0%	3.93%	**83.15%**	**93.82%**	**98.88%**
	Hard model	0.0%	1.09%	65.57%	86.89%	98.36%
	LSR model	**0.59%**	**4.05%**	64.74%	80.92%	97.69%

The size of our novel data set is a limiting element in analyzing the impact of the different labels in training, and the soft labels are time-consuming to collect: most of the survey participants were able to label around 200 images per hour, so it took over 50 labor hours in total to collect our data set. This is not unusual for human-labeled data sets. There are tools for outsourcing the labeling, but with these, there is often a risk of receiving low-quality answers. There are also limits on how complex data and labels can be collected with human annotators. With CIFAR-10 we have only 10 classes but data sets such as ImageNet [14] can have hundreds of classes to consider. It is not realistic for human participants to consider every available class in a timely manner. For data with a large number of classes, the survey design should

be developed further. One way of managing the workload would be to relabel only a subset of hard labeled data. The subset could be chosen random or using some kind of active learning design [21].

One natural future research area is expanding the soft CIFAR-10 data set. The data set and the survey can be found online at https://github.com/sannatti/softcifar.

8 CONCLUSIONS

We have discussed the importance of data labels in machine learning, illustrated the use of soft labels in image classification, and experimented with adversarial attacks. We introduced a novel data set with instance-specific soft labels. We based the data set on CIFAR-10, and therefore it does not have the natural ambiguity that many other soft-labeled data sets have. To collect the data, we built a GUI that can be adapted easily into an online survey setting to enlarge the data set. The survey and the data set are made available online.

Our survey setting and data set have similar limitations of scalability and cost as many existing human labeled data sets. For data sets with a large number of classes, the scheme could be developed further. For example, by assigning a subset of the classes for each labeler and having multiple people label the same images. The scheme could also be used for a well-chosen subset of hard-labeled data, and the training with the mixed data might improve the robustness of the ML model.

We trained a DNN with the novel data set and showed it to be more robust against targeted adversarial attacks than the networks trained with original CIFAR-10 and with original CIFAR-10 with uniformly smoothed labels. With untargeted attacks, both DNNs with soft data under-performed compared with the model with the original CIFAR-10. The stiffness of the hard model can explain this. Analysis of these examples showed that the soft-trained model was more stable and robust against adversarial attacks, supporting the conclusion that the soft-modeled data define a continuum between the classes storing information about properties of the data. This helps the model learn finer features for the task and provide soft labels that can model uncertainty, thus leading to more robust models.

We also compared the performance of our soft model to a model that uses smoothing regularization. The smoothing combines some good features of both hard and soft data. As a big differences, smoothing is usually class- not sample-specific like our soft labels, and it does not add new knowledge to the system but uses the features of the hard data to learn the smoothness. The data set and some related codes are provided at https://github.com/sannatti/softcifar.

REFERENCES

[1] Raju Anand, T. Shanthi, Muthuchamy Selvaraj Nithish, and S. Lakshman. 2020. Face recognition and classification using GoogleNET architecture. In *Soft Computing for Problem Solving*. Springer, 261–269.

[2] Hessam Bagherinezhad, Maxwell Horton, Mohammad Rastegari, and Ali Farhadi. 2018. Label Refinery: Improving ImageNet Classification through Label Progression. *arXiv preprint arXiv:1805.02641* (2018). arXiv:1805.02641 [arXiv]

[3] Joann G. Elmore, Gary M. Longton, Patricia A. Carney, Berta M. Geller, Tracy Onega, Anna N.A. Tosteson, Heidi D. Nelson, Margaret S. Pepe, Kimberly H. Allison, Stuart J. Schnitt, et al. 2015. Diagnostic concordance among pathologists interpreting breast biopsy specimens. *Jama* 313, 11 (2015), 1122–1132.

[4] Sergio Escalera, Junior Fabian, Pablo Pardo, Xavier Baró, Jordi Gonzàlez, Hugo J. Escalante, Dusan Misevic, Ulrich Steiner, and Isabelle Guyon. 2015. ChaLearn Looking at People 2015: Apparent Age and Cultural Event Recognition Datasets and Results. In *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*. 243–251.

[5] Mark Everingham, Luc van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. 2010. The PASCAL Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* 88, 2 (June 2010), 303–338. https://doi.org/10.1007/s11263-009-0275-4

[6] Xin Geng. 2016. Label distribution learning. *IEEE Transactions on Knowledge and Data Engineering* 28, 7 (2016), 1734–1748.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[8] Yanzhang He, Tara N Sainath, Rohit Prabhavalkar, Ian McGraw, Raziel Alvarez, Ding Zhao, David Rybach, Anjuli Kannan, Yonghui Wu, Ruoming Pang, et al. 2019. Streaming end-to-end speech recognition for mobile devices. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6381–6385.

[9] Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. In *NIPS Deep Learning and Representation Learning Workshop*. arXiv:1512.03385 [arXiv]

[10] Eytan Kats, Jacob Goldberger, and Hayit Greenspan. 2019. Soft labeling by distilling anatomical knowledge for improved MS lesion segmentation. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 1563–1566.

[11] Eytan Kats, Jacob Goldberger, and Hayit Greenspan. 2019. A soft STAPLE algorithm combined with anatomical knowledge. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 510–517.

[12] James M. Keller and Douglas J. Hunt. 1985. Incorporating Fuzzy Membership Functions into the Perceptron Algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-7, 6 (1985), 693–699.

[13] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. 2009. CIFAR-10 (Canadian Institute for Advanced Research). (2009). http://www.cs.toronto.edu/~kriz/cifar.html

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (Lake Tahoe, Nevada) *(NIPS'12)*. Curran Associates Inc., USA, 1097–1105.

[15] Hang Li, Dong Wei, Shilei Cao, Kai Ma, Liansheng Wang, and Yefeng Zheng. 2020. Superpixel-Guided Label Softening for Medical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 227–237.

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[17] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in Neural Information Processing Systems* 32 (2019), 4694–4703. arXiv:1906.02629 [arXiv]

[18] Quang Nguyen, Hamed Valizadegan, and Milos Hauskrecht. 2011. Learning Classification with Auxiliary Probabilistic Information. In *2011 IEEE 11th International Conference on Data Mining*. 477–486.

[19] Quang Nguyen, Hamed Valizadegan, and Milos Hauskrecht. 2013. Learning classification models with soft-label information. *Journal of the American Medical Informatics Association : JAMIA* 21, 3 (2013), 501–508.

[20] Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548* (2017). arXiv:1701.06548 [arXiv]

[21] Burr Settles. 2012. Active Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6, 1 (2012), 1–114. arXiv:https://doi.org/10.2200/S00429ED1V01Y201207AIM018

[22] Ian Sneddon, Margaret McRorie, Gary McKeown, and Jennifer Hanratty. 2011. The belfast induced natural emotion database. *IEEE Transactions on Affective Computing* 3, 1 (2011), 32–41.

[23] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

[24] Araz Taeihagh and Hazel Si Min Lim. 2019. Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks. *Transport reviews* 39, 1 (2019), 103–128.

[25] Christian Thiel. 2008. Classification on Soft Labels Is Robust against Label Noise. In *Knowledge-Based Intelligent Information and Engineering Systems*, Ignac Lovrek, Robert J. Howlett, and Lakhmi C. Jain (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 65–73.

[26] Arna van Engelen, Wiro J Niessen, Stefan Klein, Harald C Groen, Hence JM Verhagen, Jolanda J Wentzel, Aad van der Lugt, and Marleen de Bruijne. 2012. Supervised in-vivo plaque characterization incorporating class label uncertainty. In *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*. IEEE, 246–249.

[27] Willem Waegeman, Jan Verwaeren, Bram Slabbinck, and Bernard De Baets. 2011. Supervised learning algorithms for multi-class classification problems with partial class memberships. *Fuzzy Sets and Systems* 184 (2011), 106–125. Issue 1.

[28] Rey Reza Wiyatno, Anqi Xu, Ousmane Dia, and Archy de Berker. 2019. Adversarial Examples in Modern Machine Learning: A Review. *arXiv preprint arXiv:1911.05268* (2019).

[29] Lingxi Xie, Jingdong Wang, Z. Wei, M. Wang, and Q. Tian. 2016. DisturbLabel: Regularizing CNN on the Loss Layer. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4753–4762.

[30] Yanbing Xue and Milos Hauskrecht. 2017. Efficient Learning of Classification Models from Soft-label Information by Binning and Ranking. *Proceedings of the International Florida AI Research Society Conference. Florida AI Research Symposium* 2017 (2017), 164–169.

[31] Yanbing Xue and Milos Hauskrecht. 2018. Active learning of multi-class classifiers with auxiliary probabilistic information. In *Proceedings of the... International Florida AI Research Society Conference. Florida AI Research Symposium*, Vol. 2018. NIH Public Access, 158–163.

A FIGURE DESCRIPTIONS

Table 4. Survey answers separated into the images' original CIFAR-10 categories and averaged. The number of new labels N. Total number of new labels = 10, 000.

	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
plane N=1005	3.21	0.39	0.80	0.03	0.02	0.03	0.04	0.02	0.73	0.45
car N=974	0.20	3.55	0.01	0.02	0.00	0.00	0.01	0.01	0.33	1.35
bird N=1032	0.25	0.03	3.15	0.40	0.38	0.49	0.47	0.20	0.05	0.02
cat N=1017	0.02	0.04	0.31	2.81	0.53	1.70	0.29	0.44	0.03	0.02
deer N=999	0.03	0.03	0.16	0.57	2.93	1.32	0.11	1.60	0.01	0.02
dog N=936	0.01	0.01	0.13	1.65	0.43	3.07	0.12	0.60	0.01	0.00
frog N=1030	0.02	0.05	0.79	0.62	0.52	0.55	3.01	0.15	0.03	0.03
horse N=1001	0.01	0.01	0.04	0.41	1.30	1.42	0.04	3.37	0.02	0.02
ship N=1025	0.70	0.62	0.06	0.02	0.01	0.01	0.03	0.02	3.23	1.11
truck N=981	0.29	1.24	0.01	0.01	0.02	0.01	0.01	0.02	0.44	3.38

Table 5. Table of confusion matrix for targeted adversarial attacks against the soft model with target class 4: deer and $\epsilon = 0.01$. Resistance 82.56%.

	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
plane	0.77	0.00	0.09	0.00	0.05	0.00	0.00	0.09	0.00	0.00
car	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
bird	0.00	0.00	0.53	0.11	0.21	0.16	0.00	0.00	0.00	0.00
cat	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
deer	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
dog	0.00	0.00	0.00	0.00	0.18	0.79	0.00	0.04	0.00	0.00
frog	0.00	0.00	0.00	0.06	0.06	0.00	0.88	0.00	0.00	0.00
horse	0.00	0.00	0.00	0.00	0.22	0.00	0.00	0.78	0.00	0.00
ship	0.05	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.86	0.00
truck	0.00	0.03	0.00	0.00	0.03	0.00	0.00	0.00	0.03	0.90

Table 6. Table of confusion matrix for targeted adversarial attacks against the hard model with target class 4: deer and $\epsilon = 0.01$. Resistance 71.48%.

	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
plane	0.69	0.06	0.06	0.06	0.06	0.00	0.00	0.00	0.00	0.06
car	0.08	0.79	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.08
bird	0.17	0.00	0.50	0.11	0.17	0.00	0.00	0.06	0.00	0.00
cat	0.00	0.00	0.04	0.54	0.23	0.04	0.08	0.04	0.04	0.00
deer	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
dog	0.00	0.00	0.00	0.00	0.06	0.82	0.00	0.00	0.12	0.00
frog	0.00	0.00	0.05	0.11	0.11	0.00	0.74	0.00	0.00	0.00
horse	0.00	0.00	0.00	0.00	0.31	0.00	0.00	0.69	0.00	0.00
ship	0.05	0.00	0.00	0.05	0.00	0.00	0.00	0.00	0.94	0.06
truck	0.05	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.84

Table 7. Table of confusion matrix for targeted adversarial attacks against the LSR model with target class 4: deer and $\epsilon = 0.01$. Resistance 61.93%

	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
plane	0.60	0.00	0.30	0.00	0.05	0.00	0.00	0.00	0.00	0.10
car	0.00	0.78	0.04	0.00	0.00	0.00	0.04	0.00	0.00	0.13
bird	0.12	0.00	0.38	0.12	0.31	0.00	0.00	0.06	0.00	0.00
cat	0.00	0.00	0.07	0.41	0.14	0.17	0.07	0.03	0.07	0.03
deer	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
dog	0.05	0.00	0.05	0.10	0.20	0.55	0.00	0.05	0.00	0.00
frog	0.00	0.00	0.06	0.12	0.12	0.00	0.69	0.00	0.00	0.00
horse	0.00	0.00	0.00	0.00	0.07	0.14	0.07	0.71	0.00	0.00
ship	0.00	0.08	0.00	0.00	0.00	0.00	0.04	0.00	0.88	0.00
truck	0.00	0.12	0.04	0.08	0.08	0.04	0.00	0.00	0.04	0.58

Table 8. Table of confusion matrix for targeted adversarial attacks against the soft model with target class 8: bird and $\epsilon = 0.05$. Resistance 4.35%

	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
plane	0.05	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.90	0.00
car	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.89	0.11
bird	0.18	0.06	0.12	0.00	0.00	0.00	0.00	0.00	0.59	0.06
cat	0.11	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.74	0.11
deer	0.24	0.05	0.10	0.00	0.00	0.00	0.00	0.00	0.38	0.24
dog	0.15	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.67	0.12
frog	0.25	0.17	0.08	0.00	0.00	0.00	0.00	0.00	0.29	0.21
horse	0.11	0.06	0.06	0.00	0.00	0.00	0.00	0.00	0.33	0.44
ship	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
truck	0.09	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.74	0.13

Table 9. Table of confusion matrix for targeted adversarial attacks against the hard model with target class 8: bird and $\epsilon = 0.05$. Resistance 1.69%

	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
plane	0.00	0.00	0.00	0.06	0.00	0.05	0.00	0.00	0.94	0.00
car	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.95	0.05
bird	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.95	0.00
cat	0.00	0.00	0.00	0.00	0.00	0.00	0.12	0.00	0.84	0.04
deer	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.95	0.00
dog	0.00	0.00	0.05	0.00	0.00	0.05	0.00	0.00	0.89	0.00
frog	0.05	0.00	0.00	0.05	0.11	0.00	0.05	0.00	0.74	0.00
horse	0.00	0.00	0.00	0.25	0.08	0.00	0.00	0.00	0.67	0.00
ship	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
truck	0.05	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.79	0.05

Table 10. Table of confusion matrix for targeted adversarial attacks against the LSR model with target class 8: bird and $\epsilon = 0.05$. Resistance 10.55%

	plane	car	bird	cat	deer	dog	frog	horse	ship	truck
plane	0.04	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.92	0.00
car	0.00	0.38	0.00	0.00	0.00	0.05	0.00	0.00	0.52	0.05
bird	0.00	0.14	0.14	0.05	0.00	0.00	0.00	0.05	0.62	0.00
cat	0.00	0.00	0.06	0.06	0.00	0.11	0.00	0.00	0.78	0.00
deer	0.00	0.00	0.05	0.18	0.00	0.05	0.09	0.00	0.64	0.00
dog	0.00	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.89	0.00
frog	0.00	0.06	0.06	0.17	0.11	0.00	0.00	0.00	0.61	0.00
horse	0.00	0.00	0.05	0.05	0.00	0.11	0.00	0.05	0.74	0.00
ship	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
truck	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.72	0.28