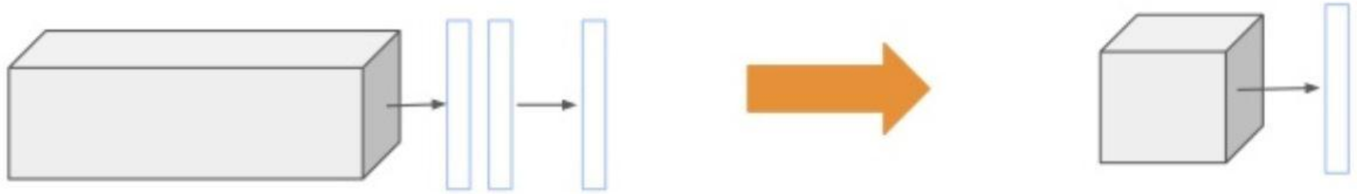
Distillation in Machine Learning

Ramesh Kumar Sah

What is Distillation?

- Distillation is a training procedure designed to train a DNN using knowledge transferred from a different DNN. [2]



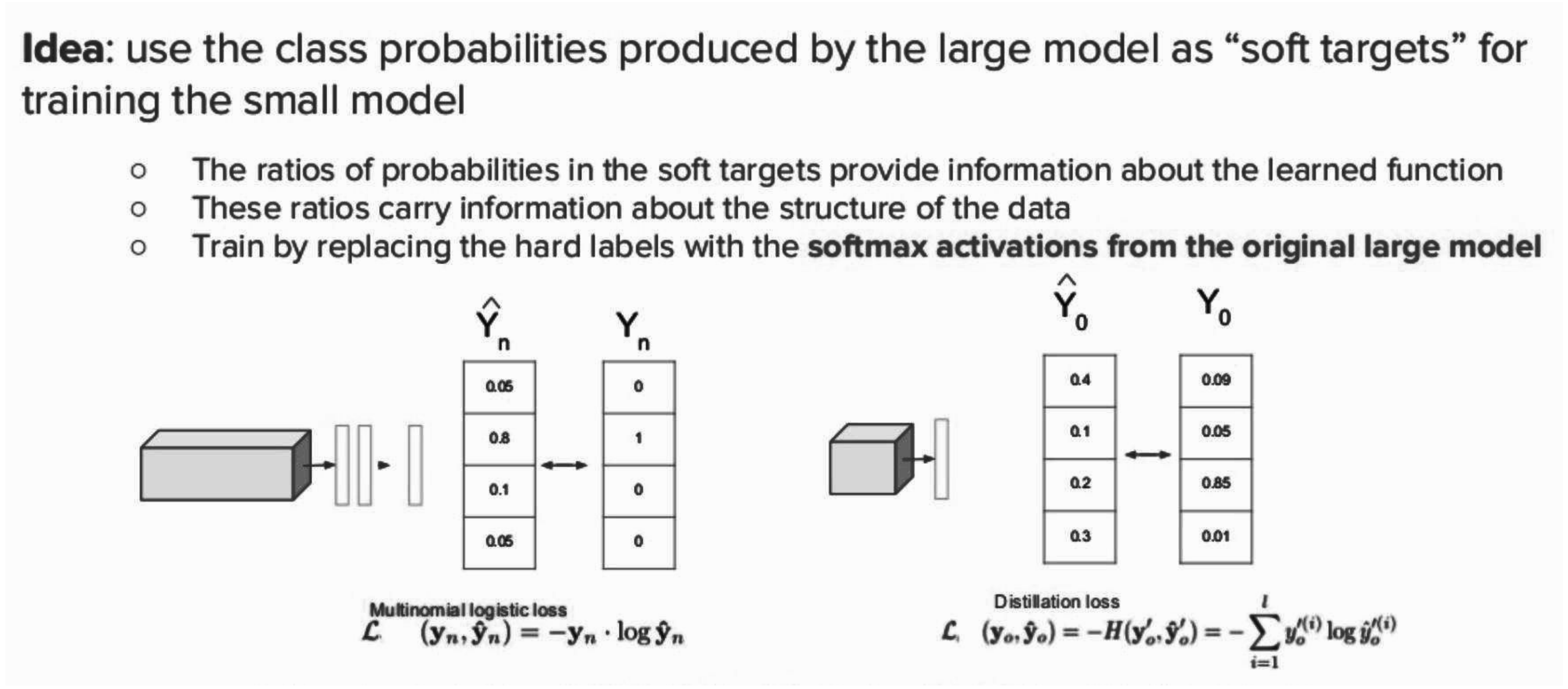
- The motivation behind the knowledge transfer is to reduce the computational complexity of DNN by transferring knowledge from larger architecture to smaller ones. This in turn facilitates the deployment of deep learning model in resource constrained devices .

[1] J. Ba and R. Caruana, “Do deep nets really need to be deep?” in Advances in Neural Information Processing Systems.

[2] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” in Deep Learning and Representation Learning Workshop at NIPS 2014

Intuition

Knowledge acquired by DNNs during training is not only encoded in weight parameters learned by the DNN but is also encoded in the probability vectors produced by the network.



Idea: use the class probabilities produced by the large model as "soft targets" for training the small model

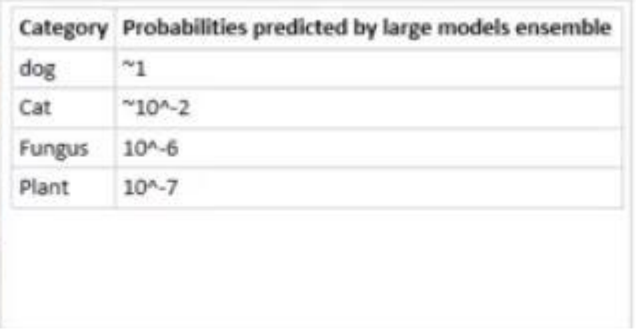
- The ratios of probabilities in the soft targets provide information about the learned function
- These ratios carry information about the structure of the data
- Train by replacing the hard labels with the **softmax activations from the original large model**

Multinomial logistic loss

$$\mathcal{L}\ (\mathbf{y}_n, \hat{\mathbf{y}}_n) = -\mathbf{y}_n \cdot \log \hat{\mathbf{y}}_n$$

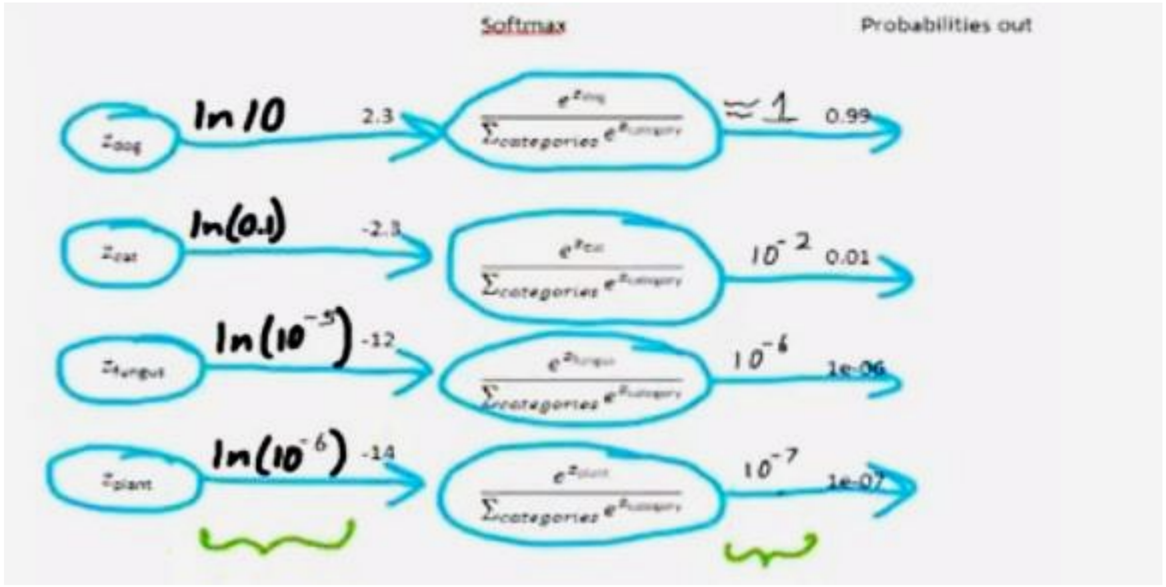
Distillation loss

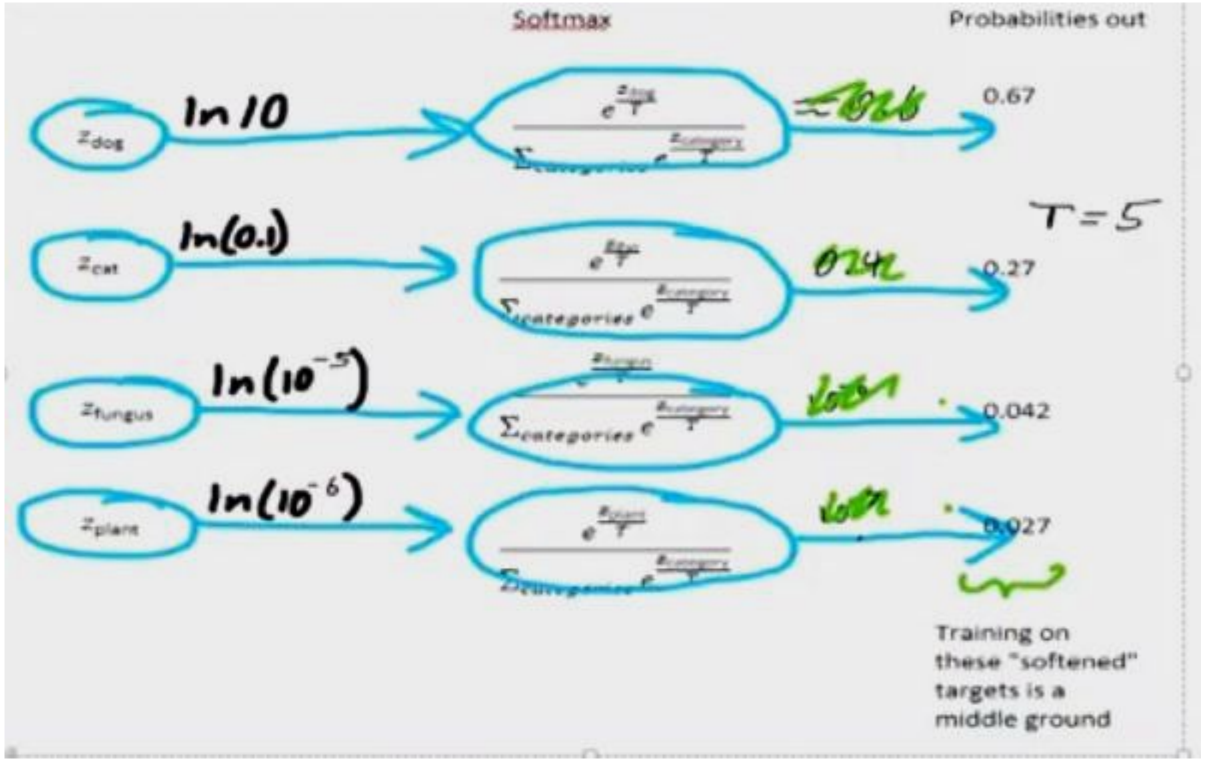
$$\mathcal{L}\ (\mathbf{y}_o, \hat{\mathbf{y}}_o) = -H(\mathbf{y}'_o, \hat{\mathbf{y}}'_o) = -\sum_{i=1}^{l} y_o'^{(i)} \log \hat{y}_o'^{(i)}$$



Category	Probabilities predicted by large models ensemble
dog	~1
Cat	~10^-2
Fungus	10^-6
Plant	10^-7

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}$$


Softmax
Probabilities out
z_{dog}
ln10
2.3
$\frac{e^{z_{dog}}}{\sum_{categories} e^{z_{category}}}$
≈1
0.99
z_{cat}
ln(0.1)
-2.3
$\frac{e^{z_{cat}}}{\sum_{categories} e^{z_{category}}}$
10^{-2}
0.01
z_{fungus}
ln(10^{-5})
-12
$\frac{e^{z_{fungus}}}{\sum_{categories} e^{z_{category}}}$
10^{-6}
1e-06
z_{plant}
ln(10^{-6})
-14
$\frac{e^{z_{plant}}}{\sum_{categories} e^{z_{category}}}$
10^{-7}
1e-07


Softmax
Probabilities out
z_{dog}
$\ln 10$
$\frac{e^{\frac{z_{dog}}{T}}}{\sum_{categories} e^{\frac{z_{category}}{T}}}$
≈0.66
0.67
$T=5$
z_{cat}
$\ln(0.1)$
$\frac{e^{\frac{z_{cat}}{T}}}{\sum_{categories} e^{\frac{z_{category}}{T}}}$
0.242
0.27
z_{fungus}
$\ln(10^{-5})$
$\frac{e^{\frac{z_{fungus}}{T}}}{\sum_{categories} e^{\frac{z_{category}}{T}}}$
0.042
z_{plant}
$\ln(10^{-6})$
$\frac{e^{\frac{z_{plant}}{T}}}{\sum_{categories} e^{\frac{z_{category}}{T}}}$
0.027

Training on these "softened" targets is a middle ground

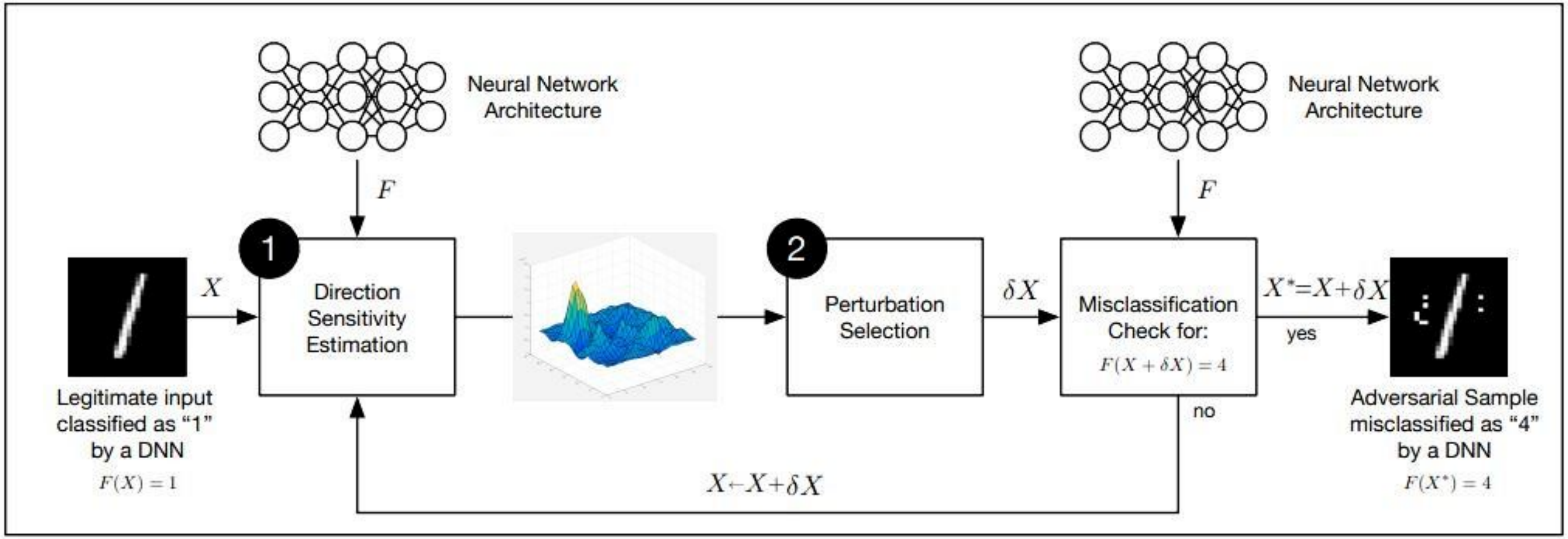

Neural Network Architecture
Neural Network Architecture
F
F
1
2
X
Direction Sensitivity Estimation
Perturbation Selection
δX
Misclassification Check for: $F(X+\delta X)=4$
$X^{*}=X+\delta X$
yes
no
Legitimate input classified as "1" by a DNN
$F(X)=1$
Adversarial Sample misclassified as "4" by a DNN
$F(X^{*})=4$
$X \leftarrow X+\delta X$

Fig. 3: **Adversarial crafting framework:** Existing algorithms for adversarial sample crafting [7], [9] are a succession of two steps: (1) *direction sensitivity estimation* and (2) *perturbation selection*. Step (1) evaluates the sensitivity of model F at the input point corresponding to sample X. Step (2) uses this knowledge to select a perturbation affecting sample X's classification. If the resulting sample $X + \delta X$ is misclassified by model F in the adversarial target class (here 4) instead of the original class (here 1), an adversarial sample X^* has been found. If not, the steps can be repeated on updated input $X \leftarrow X + \delta X$.

- Distillation as a Defense to Adversarial Perturbations against Deep Neural Network, Nicolas Papernot et.al.

Defensive Distillation

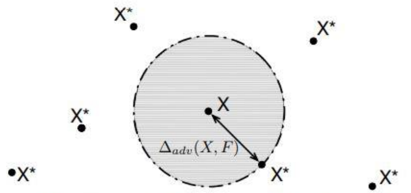


Fig. 4: **Visualizing the hardness metric:** This 2D representation illustrates the hardness metric as the radius of the disc centered at the original sample X and going through the closest adversarial sample X^* among all the possible adversarial samples crafted from sample X. Inside the disc, the class output by the classifier is constant. However, outside the disc, all samples X^* are classified differently than X.

Instead of transferring knowledge between different architecture, use the knowledge extracted from DNN to improve its own resilience to adversarial examples.

The knowledge extracted during distillation is used to reduce the amplitude of network gradients exploited to craft adversarial examples.

Defensive distillation smoothes the model learned by a DNN during training by helping the model generalize better to samples outside of its training set.

- Distillation as a Defense to Adversarial Perturbations against Deep Neural Network, Nicolas Papernot et.al.

Robustness of DNN

A robust DNN should:

1. Display good accuracy inside and outside of its training dataset
2. Model a smooth classifier function which would intuitively classify inputs relatively consistently in the neighborhood of a given sample.

Thus **the higher the average minimum perturbation required to misclassify a sample form the data manifold is, the more robust the DNN is to adversarial samples.**

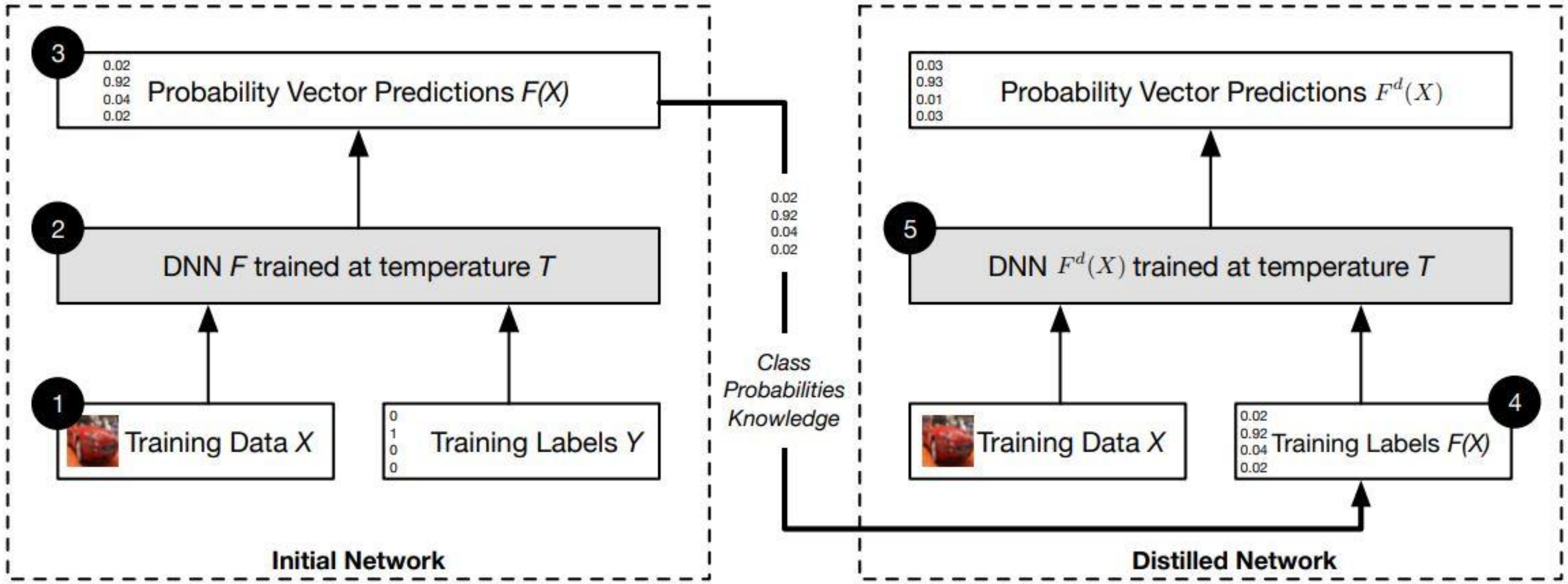


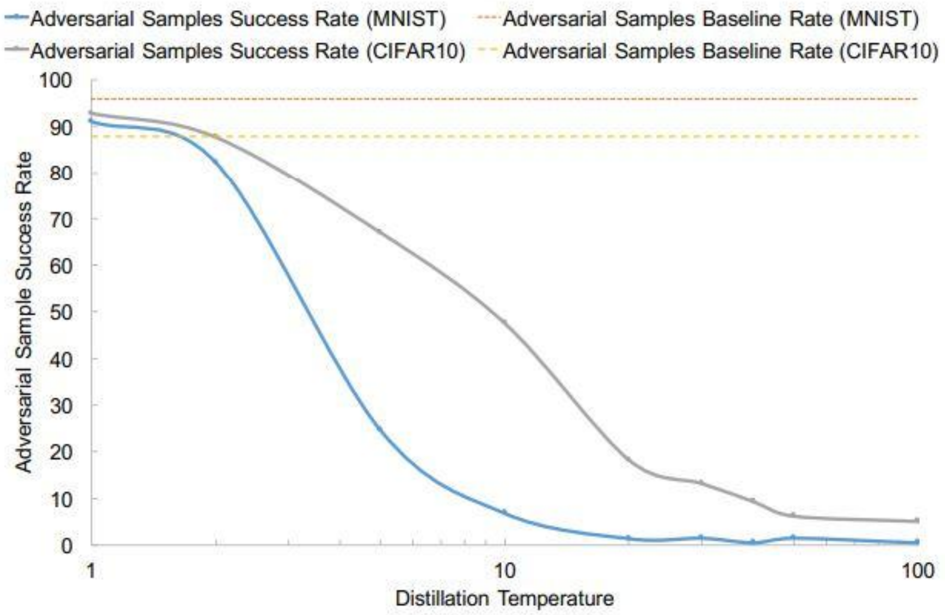
Fig. 5: **An overview of our defense mechanism based on a transfer of knowledge contained in probability vectors through distillation:** We first train an initial network F on data X with a softmax temperature of T. We then use the probability vector $F(X)$, which includes additional knowledge about classes compared to a class label, predicted by network F to train a distilled network F^d at temperature T on the same data X.

- Distillation as a Defense to Adversarial Perturbations against Deep Neural Network, Nicolas Papernot et.al.

The benefit of using the soft-targets as training labels lies in the additional knowledge found in the probability vectors compared to hard class labels.

This additional entropy encodes the relative differences between classes.

Training a network with this explicit relative information about classes prevents models from fitting too tightly to the data, and contributes to a better generalization around training points.



Distillation Temperature	MNIST Adversarial Samples Success Rate (%)	CIFAR10 Adversarial Samples Success Rate (%)
1	91	92.78
2	82.23	87.67
5	24.67	67
10	6.78	47.56
20	1.34	18.23
30	1.44	13.23
40	0.45	9.34
50	1.45	6.23
100	0.45	5.11
No distillation	95.89	87.89

Fig. 7: **An exploration of the temperature parameter space:** for 900 targets against the MNIST and CIFAR10 based models and several distillation temperatures, we plot the percentage of targets achieved by crafting an adversarial sample while altering at most 112 features. Baselines for models trained without distillation are in dashes. Note the horizontal logarithmic scale.

- Distillation as a Defense to Adversarial Perturbations against Deep Neural Network, Nicolas Papernot et.al.

Thank You!