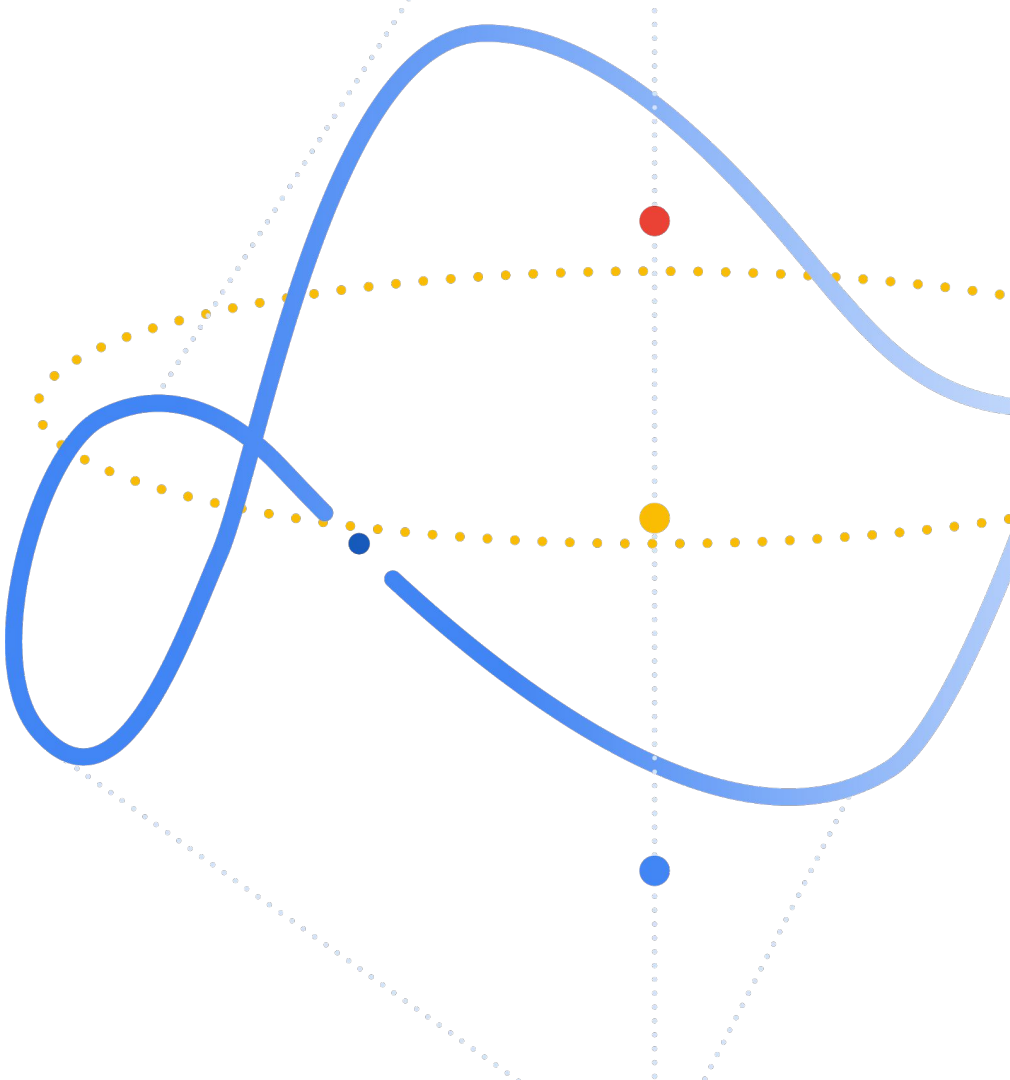
Google AI

When does label smoothing help?

Rafael Müller, Simon Kornblith, Geoffrey Hinton
Neurips 2019



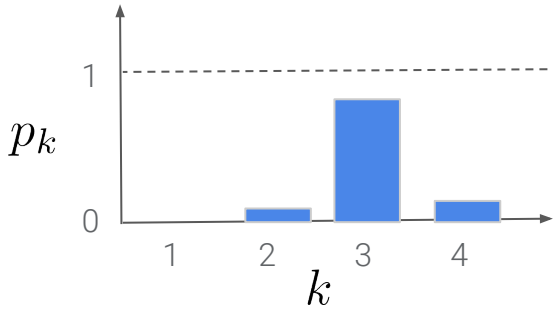
Label smoothing

Cross-entropy

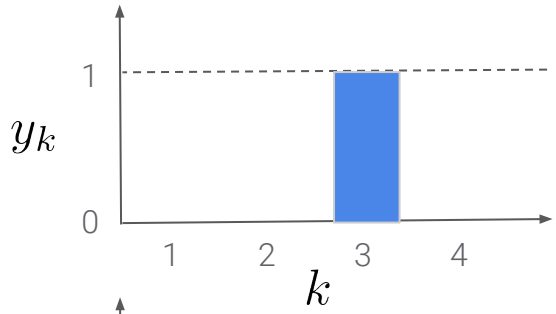
$$H(\mathbf{y}, \mathbf{p}) = \sum_{k=1}^{K} -y_k \log(p_k)$$

Modified targets with label smoothing

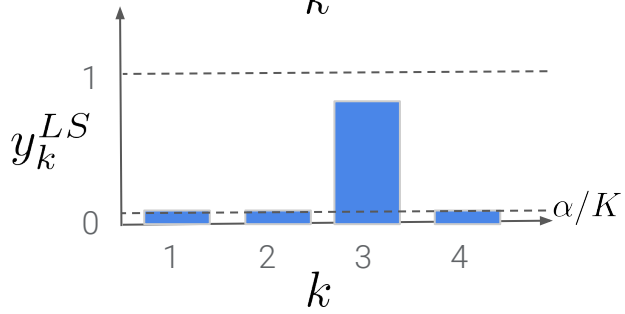
$$y_k^{LS} = y_k(1-\alpha) + \alpha/K$$



Predictions



Target



Label smoothing target

Label smoothing

Table 1: Survey of literature label smoothing results on three supervised learning tasks.

DATA SET	ARCHITECTURE	METRIC	VALUE W/O LS	VALUE W/ LS
IMAGENET	INCEPTION-V2 [6]	TOP-1 ERROR	23.1	**22.8**
		TOP-5 ERROR	6.3	**6.1**
EN-DE	TRANSFORMER [11]	BLEU	25.3	**25.8**
		PERPLEXITY	**4.67**	4.92
WSJ	BILSTM+ATT.[10]	WER	8.9	7.0/**6.7**

Widely used to improve performance across different tasks and architectures.

However, why it works is not well understood.

Outline

Penultimate layer representations

Implicit calibration

Distillation and mutual information

Future work

Penultimate layer representations

Penultimate layer representations

$$p_k = \frac{e^{\mathbf{x}^T \mathbf{w}_k}}{\sum_{l=1}^{K} e^{\mathbf{x}^T \mathbf{w}_l}}$$

activations penultimate layer

weights of last layer for k-th logit

k-th logit

Logits are approximate distance between activations of penultimate layer and class' templates

$$||\mathbf{x} - \mathbf{w}_k||^2 = \mathbf{x}^T\mathbf{x} - 2\mathbf{x}^T\mathbf{w}_k + \mathbf{w}_k^T\mathbf{w}_k$$

Logits (approximate distance to template)

With hard targets

- Can be very large (overconfident) for correct class
- Can vary a lot among incorrect classes as long as logit of correct class is very large

With label smoothing

- Limited in magnitude for correct class
- Same value for the incorrect classes

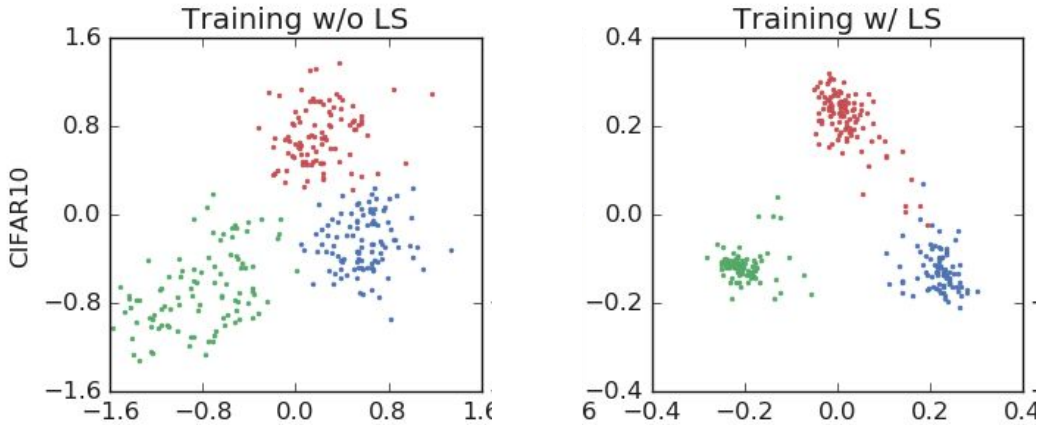
With label smoothing, activation is close to template of correct class and equally distant to templates of all remaining classes.

$$p_k = \frac{e^{\mathbf{x}^T \mathbf{w}_k}}{\sum_{l=1}^{K} e^{\mathbf{x}^T \mathbf{w}_l}}$$

Projecting penultimate layer activations in 2-D

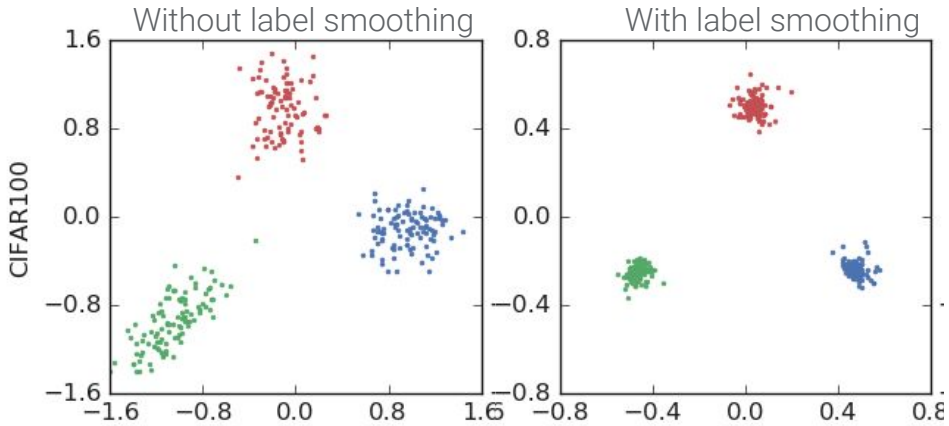
Pick 3 classes (k1, k2, k3) and corresponding templates $\mathbf{w}_{k_1}, \mathbf{w}_{k_3}, \mathbf{w}_{k_2}$

Project activations onto plane connecting the 3 templates



With label smoothing, activation is close to template of correct class and equally distant to templates of all remaining classes.

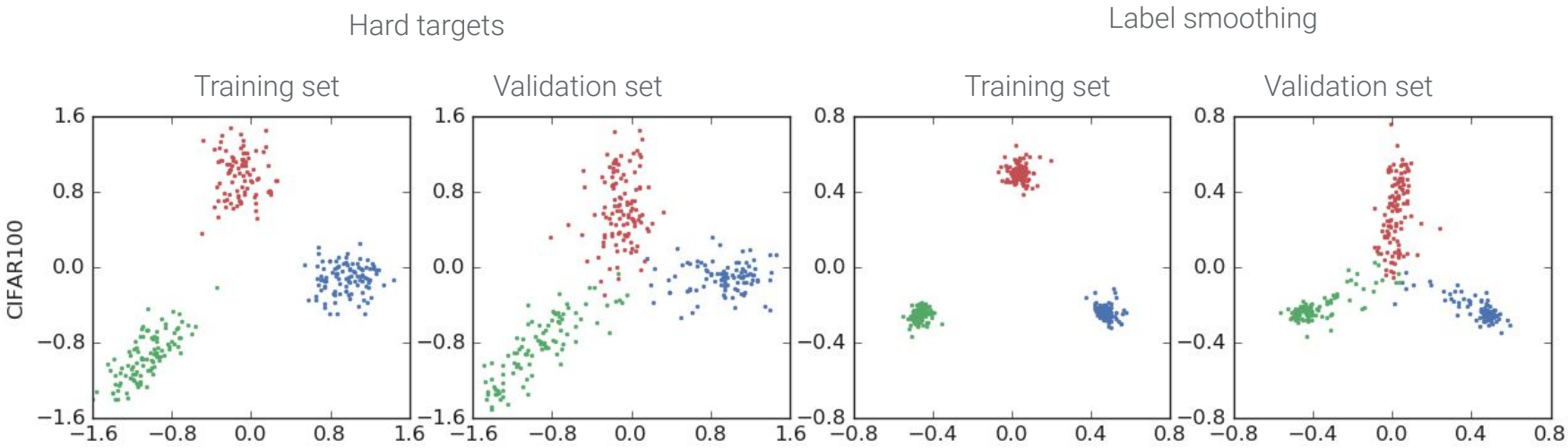
Projecting penultimate layer activations in 2-D (CIFAR100)



Information lost with label smoothing:

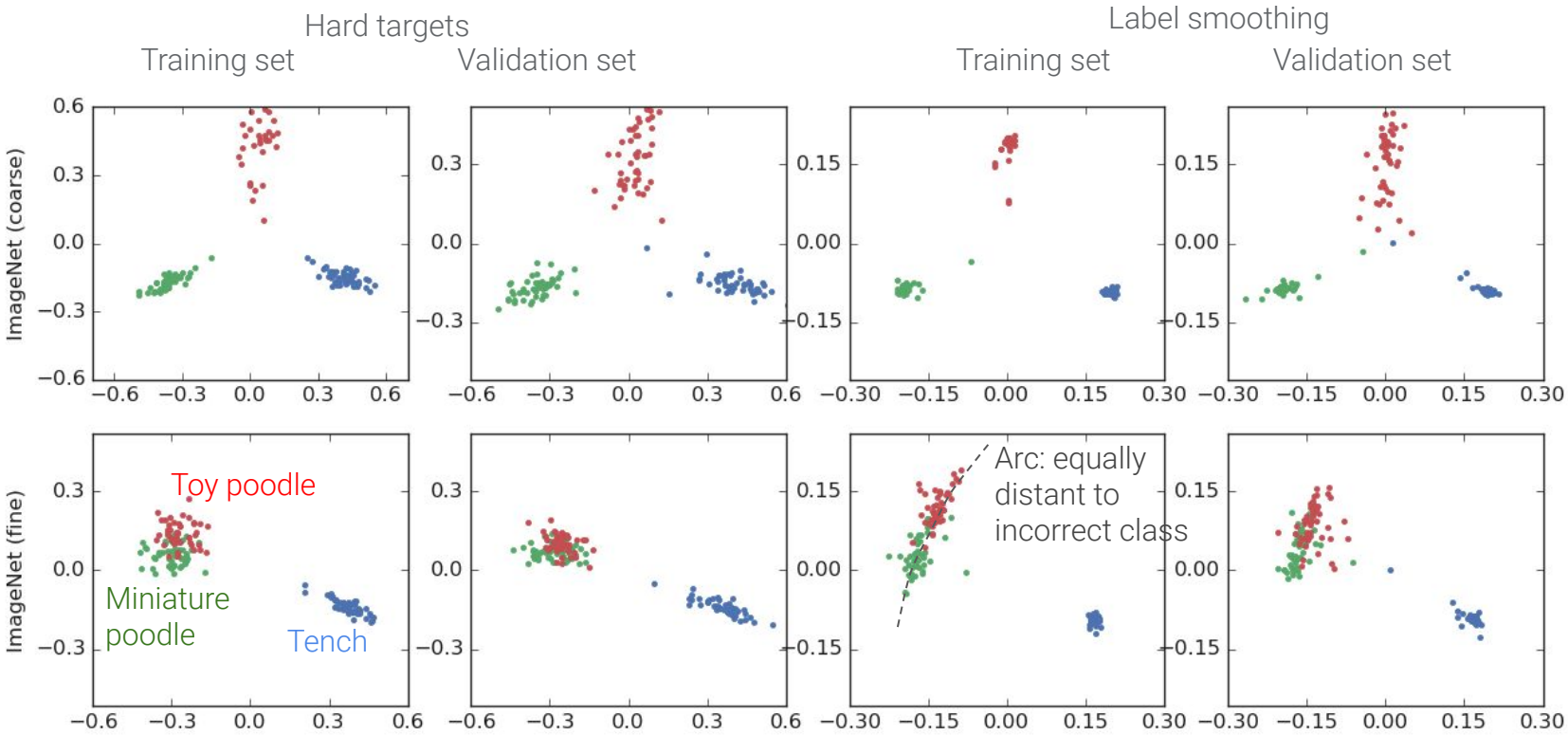
- **Confidence difference between examples of the same class**
- **Similarity between classes**

What about the validation set?



With label smoothing, there is a range of confidences for different examples on validation set.

ImageNet (semantically different vs semantically similar)



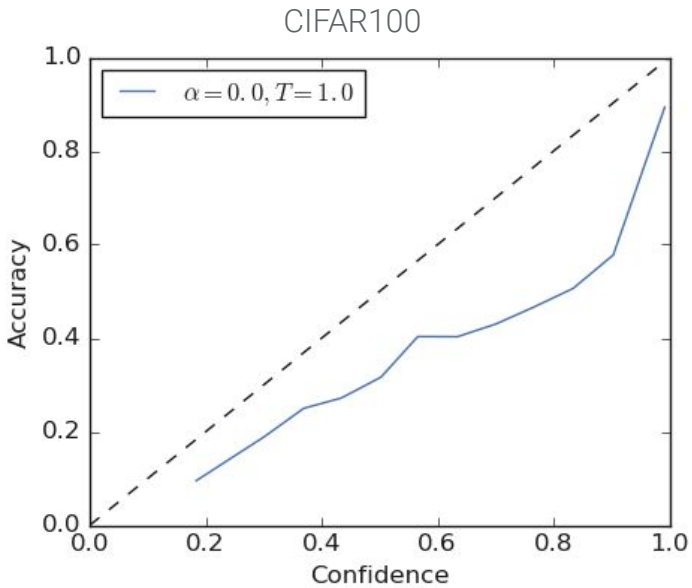
Implicit Calibration

Calibration

Network is calibrated if for a softmax value of X (confidence) the prediction is correct X*100% of time

Reliability diagram bins network's confidences for max-prediction and calculate accuracy for each bin

Modern networks are overconfident

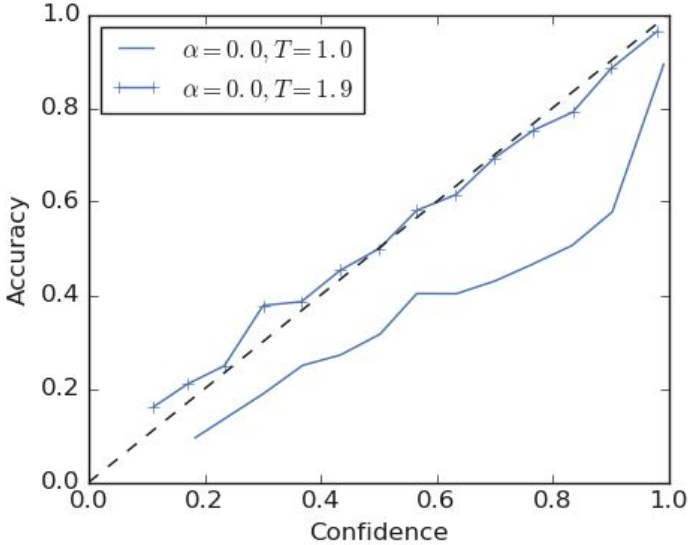


Calibration

Network is calibrated if for a softmax value of X (confidence) the prediction is correct X*100% of time

Reliability diagram bins network's confidences for max-prediction and calculate accuracy for each bin

Modern networks are overconfident **but simple logit temperature scaling is surprisingly effective**



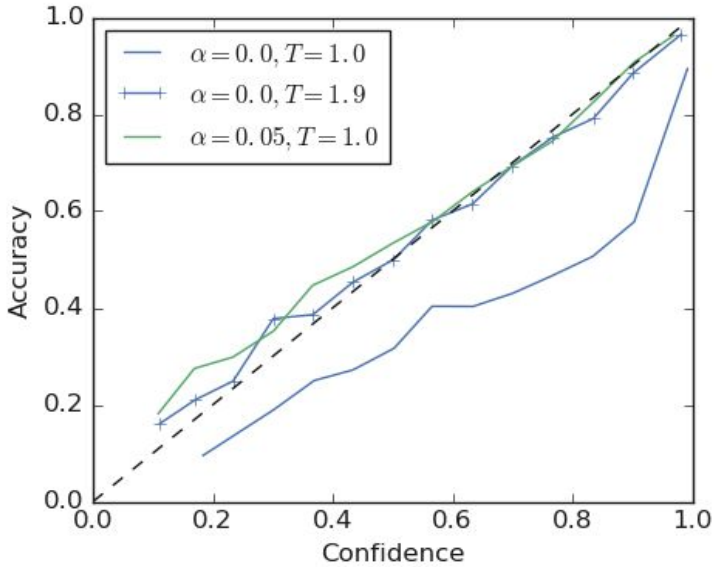
Calibration

Network is calibrated if for a softmax value of X (confidence) the prediction is correct X*100% of time

Reliability diagram bins network's confidences for max-prediction and calculate accuracy for each bin

Modern networks are overconfident but simple logit temperature scaling is surprisingly effective

And label smoothing has a similar effect to temperature scaling (green curve)



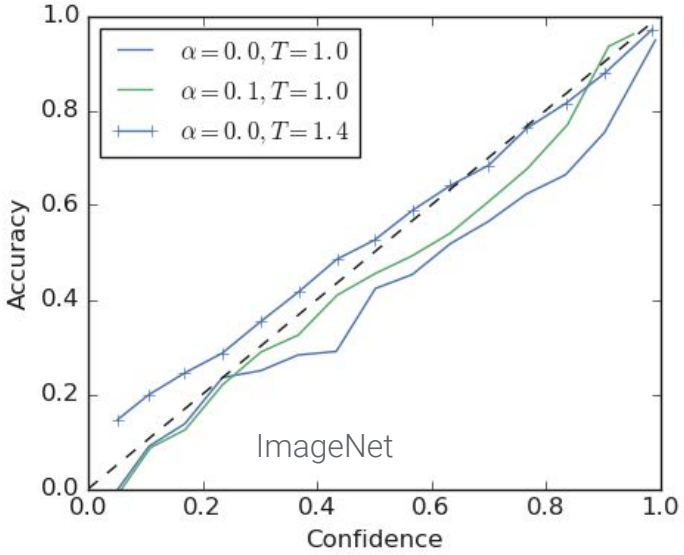
Calibration

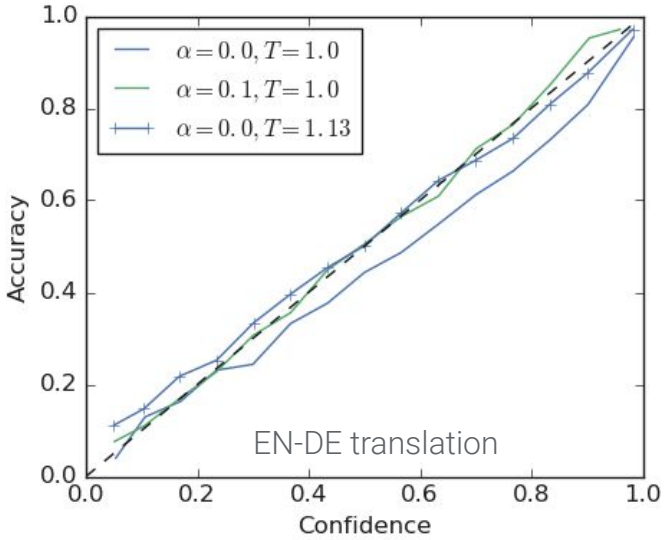
It is also effective for ImageNet (Inception-v4)

And English-German translation (Transformer)

Data set	Architecture	Baseline ECE (T=1.0, $\alpha = 0.0$)	Temp. scaling ECE / T ($\alpha = 0.0$)	Label smoothing ECE / α (T=1.0)
CIFAR-100	ResNet-56	0.150	0.021 / 1.9	0.024 / 0.05
ImageNet	Inception-v4	0.071	0.022 / 1.4	0.035 / 0.1
EN-DE	Transformer	0.056	0.018 / 1.13	0.019 / 0.1

Expected calibration error (**ECE**): average difference between confidence and accuracy for each bin (weighted by bin frequency)



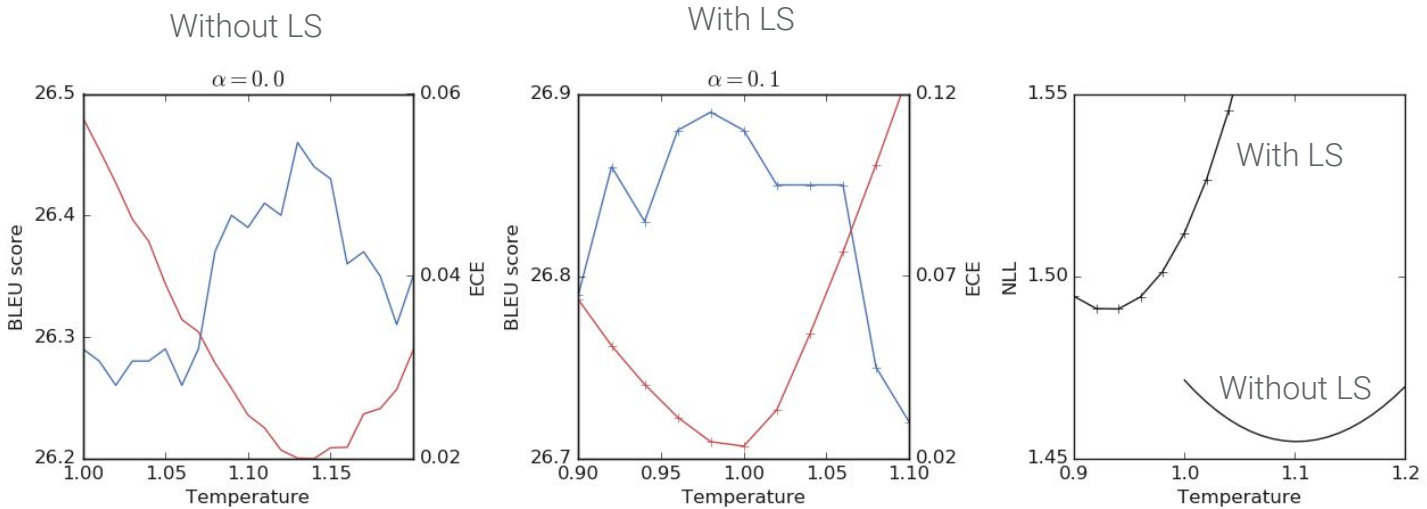


Calibration with beam-search

Calibration partly explain why label smoothing helps translation (despite hurting perplexity)

Beam-search with beam-size>1 benefits from calibrated predictions (higher BLEU score)

While other downstream tasks are invariant to calibration (classification: argmax)



Knowledge distillation

Knowledge distillation

Toy experiment on MNIST

	Test error [%]
Teacher Dropout	0.67
Distill Teacher Dropout to narrow student	0.74
Teacher Label Smoothing	0.59
Distill Teacher Label smoothing to narrow student	0.91

Something goes seriously wrong with distillation when the teacher is trained with label smoothing.

Label smoothing improves teacher's generalization but hurts knowledge transfer to student.

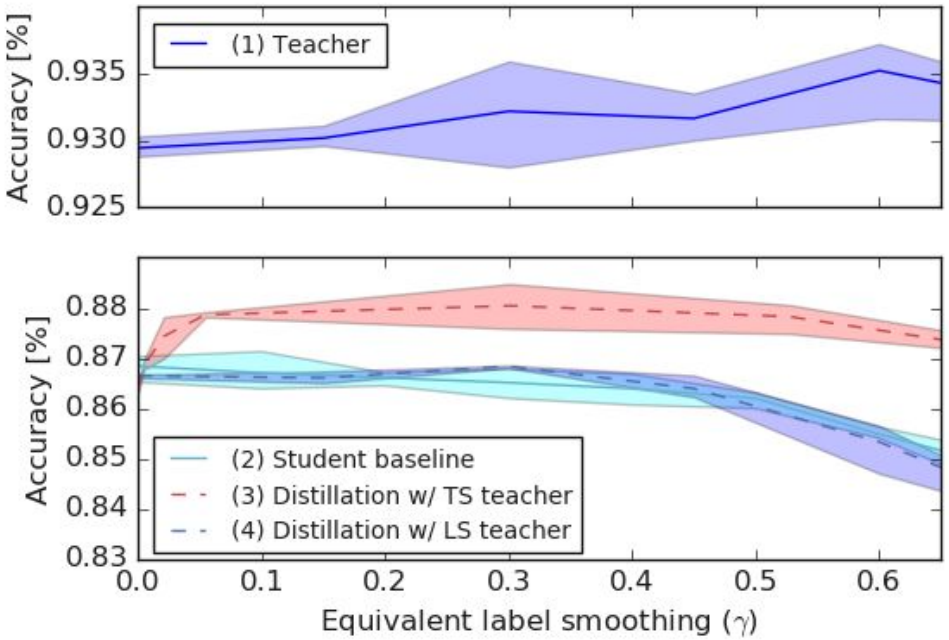
Knowledge distillation (CIFAR10)

Similar effect on CIFAR10

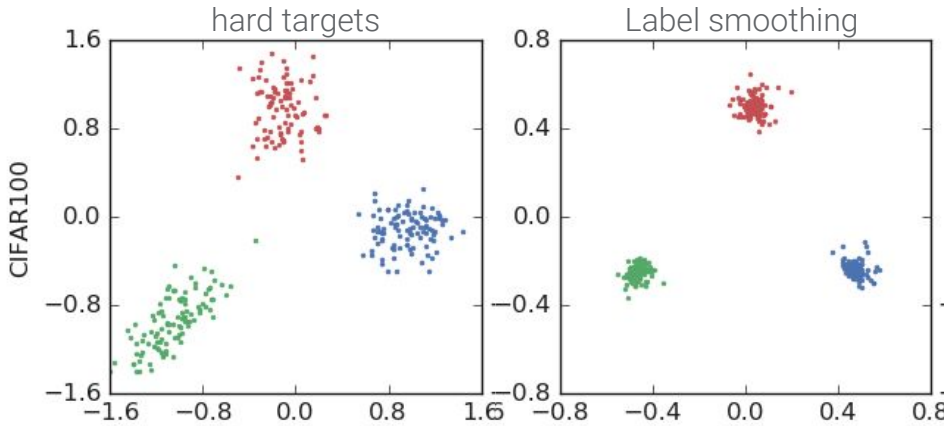
Label smoothing benefits Resnet teacher (1)

Teacher without label smoothing transfers well with temperature scaling to AlexNet student (3)

Teacher with label smoothing transfers poorly (4) and is no better than student trained with label smoothing (2)



Revisiting representations training set



Information lost with label smoothing:

- **Confidence difference between examples of the same class**
- **Similarity between classes**
- **Harder to distinguish between examples, thus less information for distillation!**

Measuring how much the logit remembers the input

$$y = f(d(\mathbf{z}_x))$$

x => index of image from training set

z => image

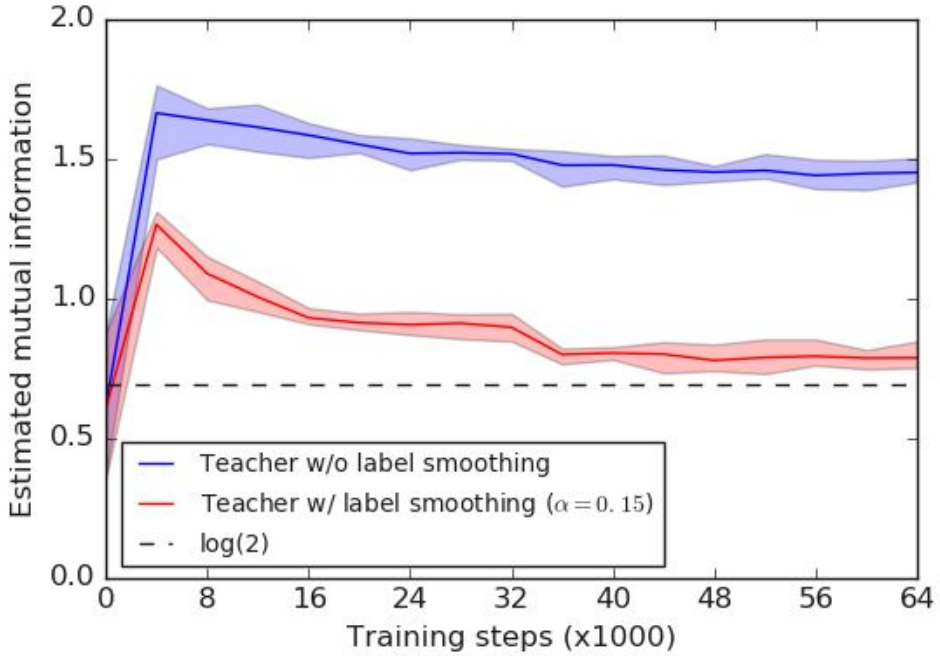
d() => random data augmentation

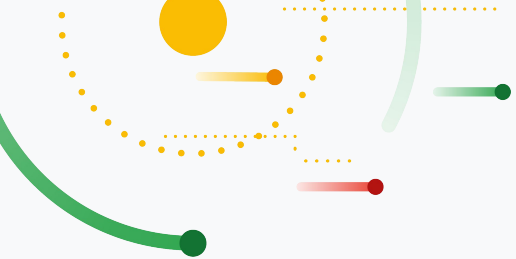
f() => image to difference between two logits (includes neural network)

y => real-valued single dimension

$$I(X;Y) = E_{X,Y}[\log(p(y|x)) - \log(\sum_x p(y|x))]$$

Approximate p(y|x) as Gaussian with mean and variance calculated via Monte Carlo





Google AI

Summary

Summary

Label smoothing changes representations learned in penultimate layer and attenuates differences between examples and classes

Label smoothing helps:

1. Works broadly across datasets architectures
2. Label smoothing implicitly calibrates model's predictions
3. Calibration improves beam-search partly explaining success of label smoothing in translation

Label smoothing does not help:

1. Better teachers may distill worse, i.e. label smoothing trained teacher distill poorly
2. Explained visually and by mutual information reduction