Google AI

When does label smoothing help?

Rafael Müller, Simon Kornblith, Geoffrey Hinton

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**

Improves performance across different tasks and architectures.

However, why it works is not well understood.

Google AI

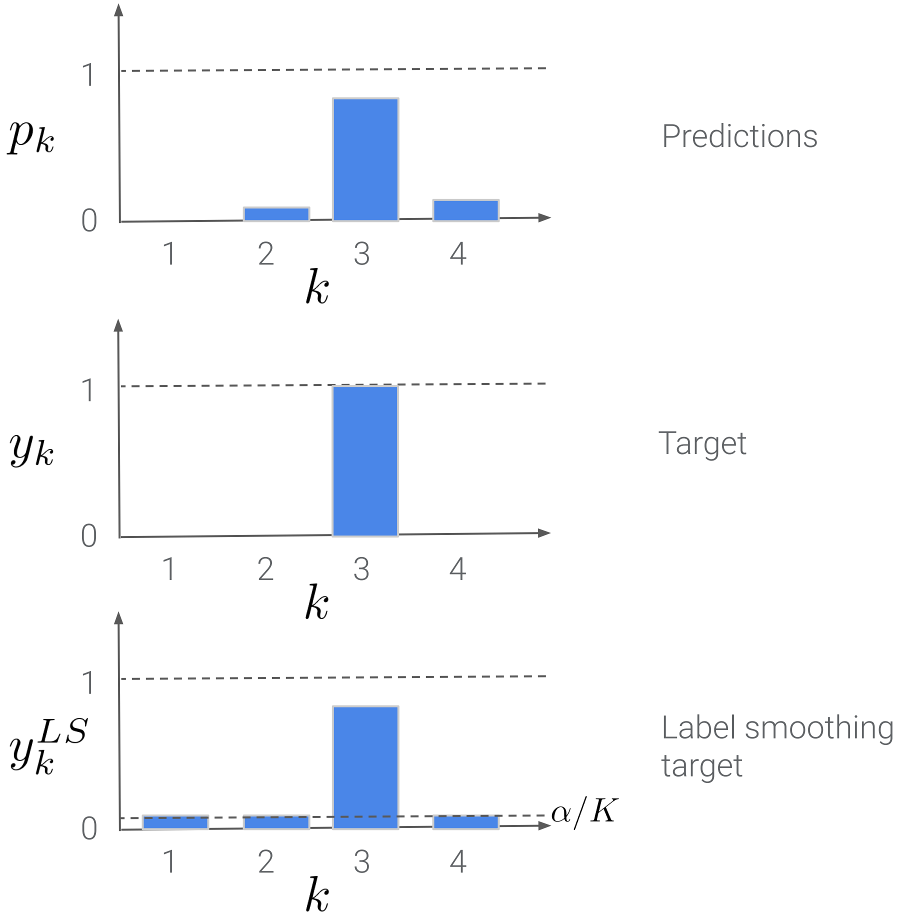
Preliminaries

Cross-entropy

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

Modified targets with label smoothing

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



Penultimate layer representations

Penultimate layer representations

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

activations penultimate layer

weights of last layer for k-th logit (class' prototype)

k-th logit

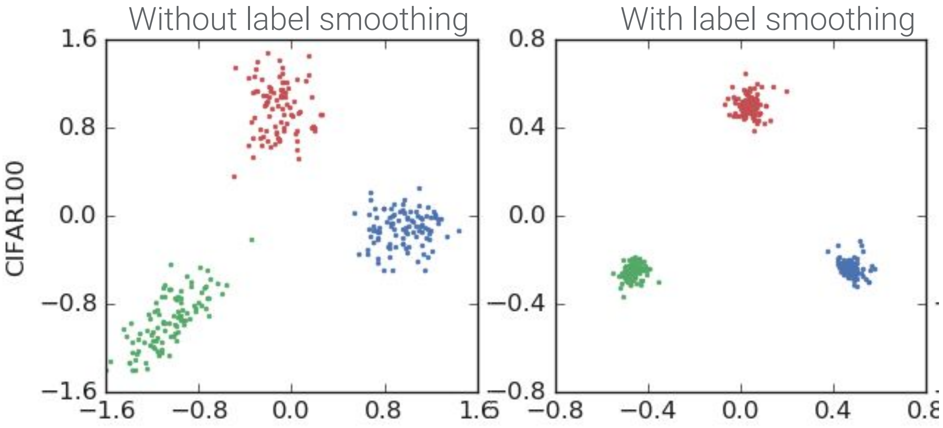
$$||\boxed{\mathbf{x}} - \boxed{\mathbf{w}_k}||^2 = \mathbf{x}^T\mathbf{x} - 2\boxed{\mathbf{x}^T\mathbf{w}_k} + \mathbf{w}_k^T\mathbf{w}_k$$

Logits are approximate distance between activations of penultimate layer and class’ prototypes

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 prototype of correct class and equally distant to prototypes of all remaining classes.

Implicit Calibration

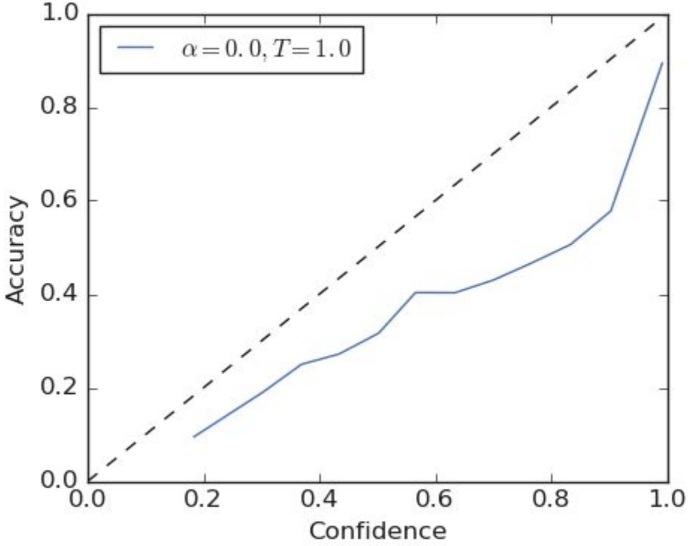
Calibration

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 neural networks are overconfident

CIFAR100

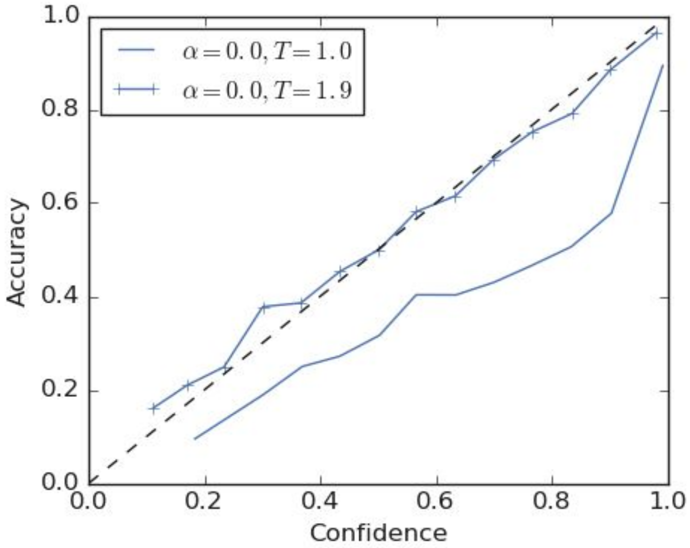


Calibration

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



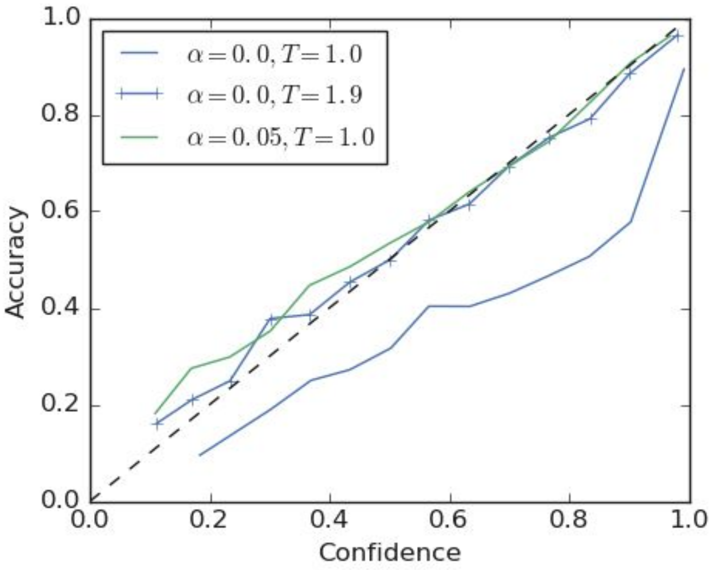
Calibration

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

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



Calibration with beam-search

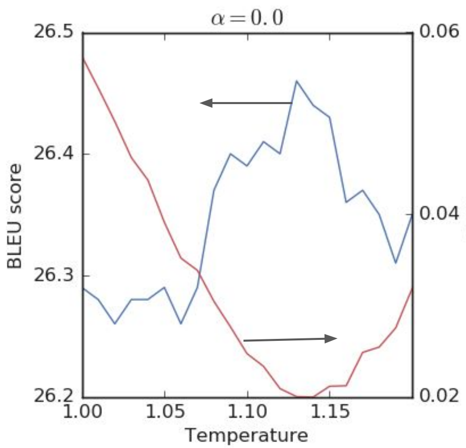
English-to-German translation using Transformer

Expected calibration error (ECE)

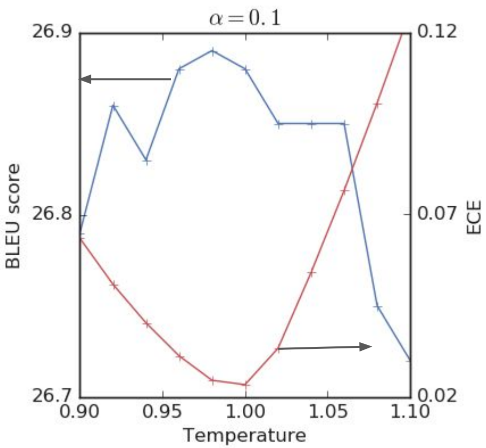
Beam-search benefits from calibrated predictions (higher BLEU score)

Calibration partly explain why LS helps translation (despite hurting perplexity)

Without LS



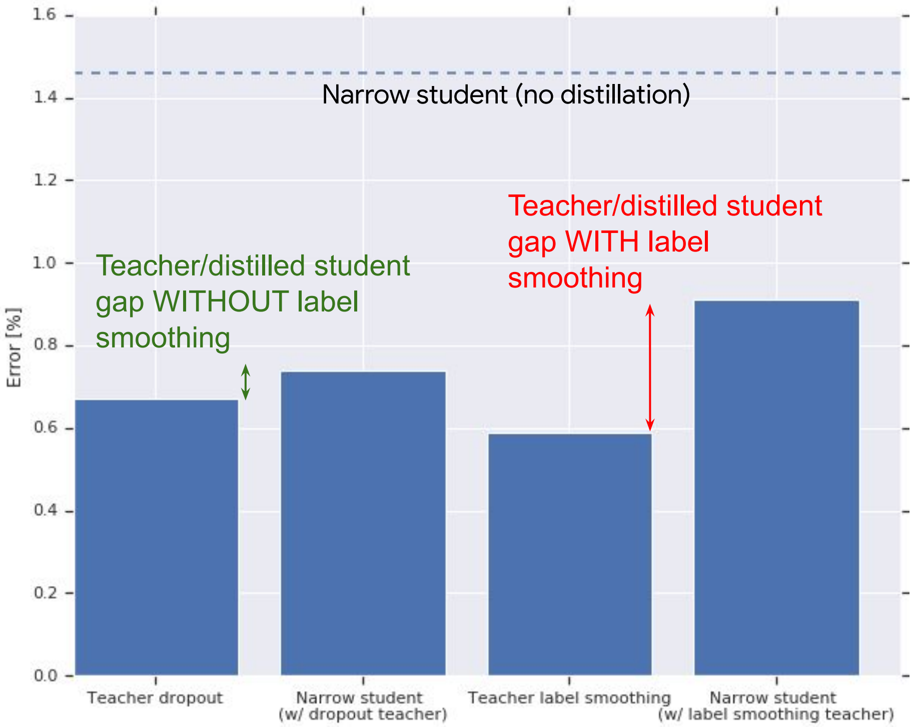
With LS



Knowledge distillation

Knowledge distillation

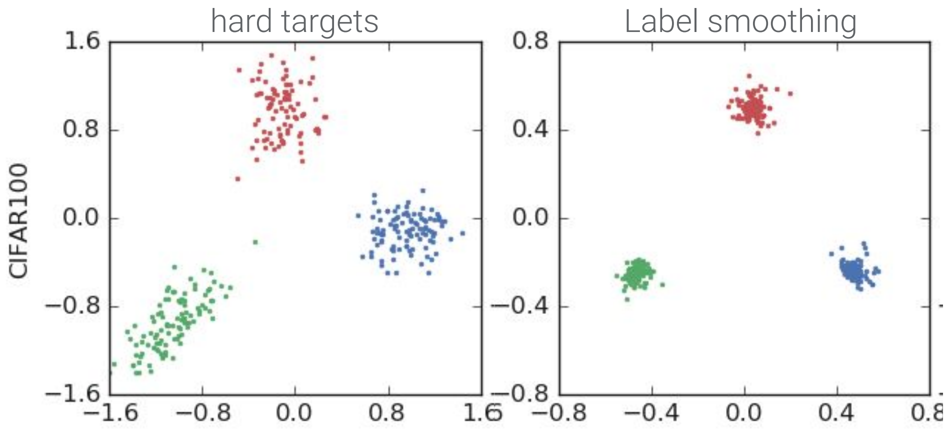
Toy experiment on MNIST



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.

Revisiting representations training set



Information lost with label smoothing:

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

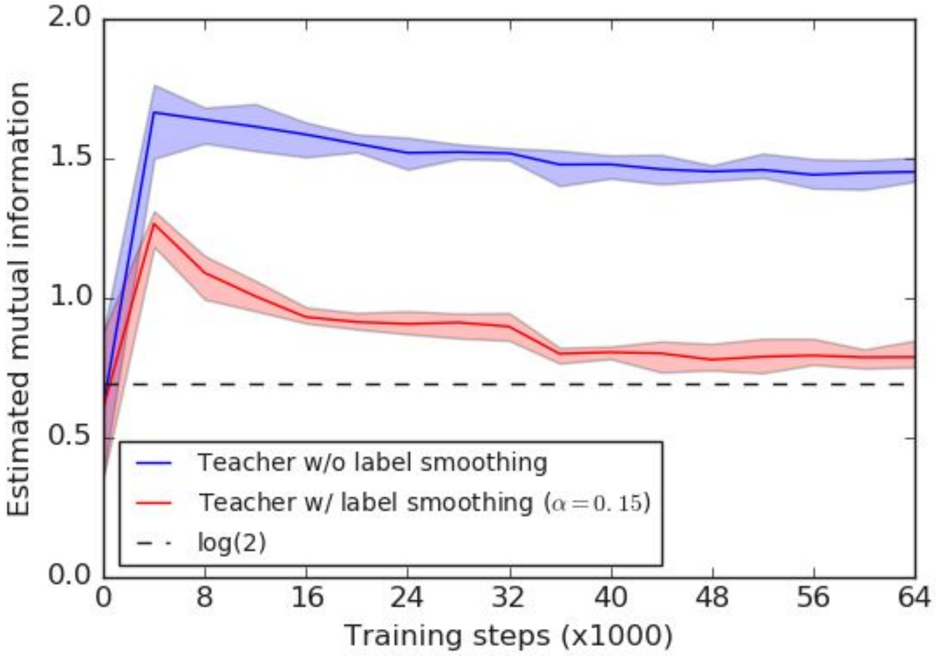
d() => random data augmentation

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



Summary

Google AI

Summary

Label smoothing attenuates differences between examples and classes

Label smoothing helps:

1. Better accuracy across datasets and architectures
2. Implicitly calibrates model's predictions
3. Calibration helps beam-search
   a. 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
   a. Explained visually and by mutual information reduction

Poster
#164