A Method to Reveal Speaker Identity in Distributed ASR Training, and How to Counter It

Trung Dang, Om Thakkar[+], Swaroop Ramaswamy[+], Rajiv Mathews[+], Peter Chin*, Françoise Beaufays[+]*

**Boston University*
[+]Google LLC

Outline

1. Motivation & Background
    - Prior Work: Gradients Matching (GM)
2. Challenges Applying GM to Speech
3. Proposed Method
4. Experiments

Motivation & Background

Motivation

- In distributed frameworks such as Federated Learning [1]
  - Model training involves transmitting gradients/updates over a network
  - Ensure user's data remains on-device

[1] Federated Learning: Collaborative Machine Learning without Centralized Training Data [Google AI Blog]

Motivation

- In distributed frameworks such as Federated Learning [1]
  - Model training involves transmitting gradients/updates over a network
  - Ensure user's data remains on-device
- But, privacy can still be leaked from gradients!
  - it is possible to obtain the private training data from the publicly shared gradients [2]

[1] Federated Learning: Collaborative Machine Learning without Centralized Training Data [Google AI Blog]
[2] Deep Leakage from Gradients [Zhu et. al., 2019]

Prior Work: Gradients Matching [2]

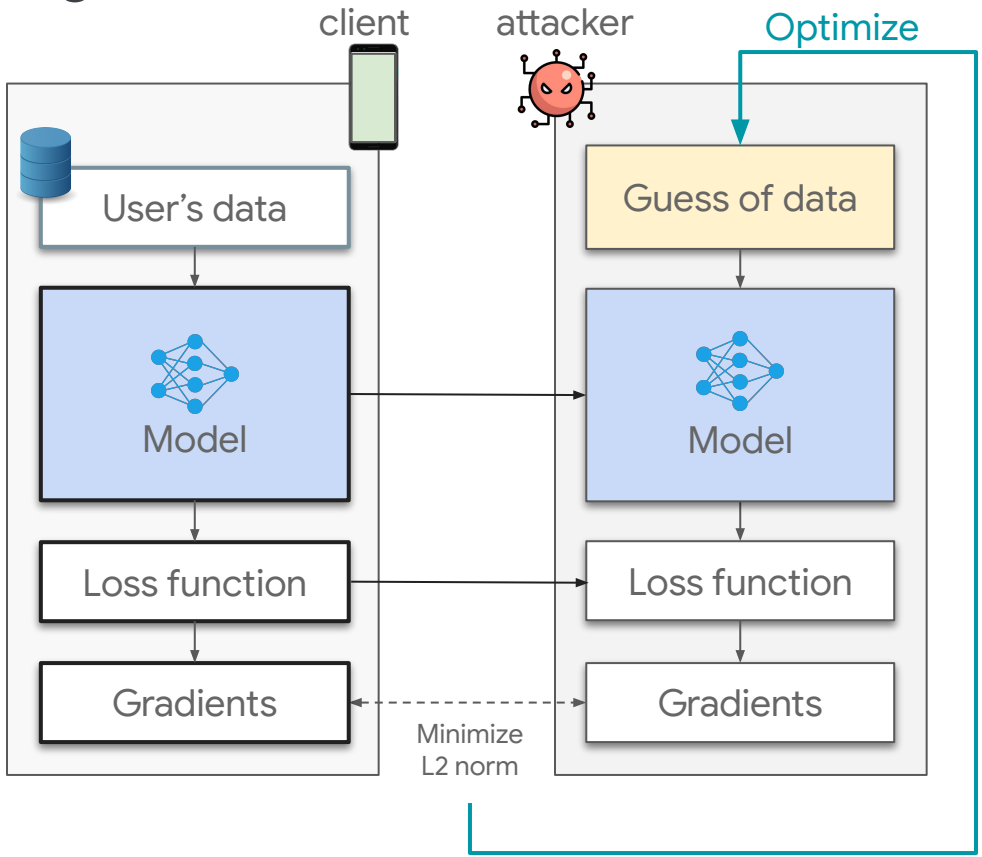
White box attack on the ***gradient***

Knowns: **Model, Loss function, Gradients**

Unknown: **User's data**

Attack strategy

- Make an initial guess of the data
- Compute gradients with guess
- Minimize difference b/w gradients from user's data and guess



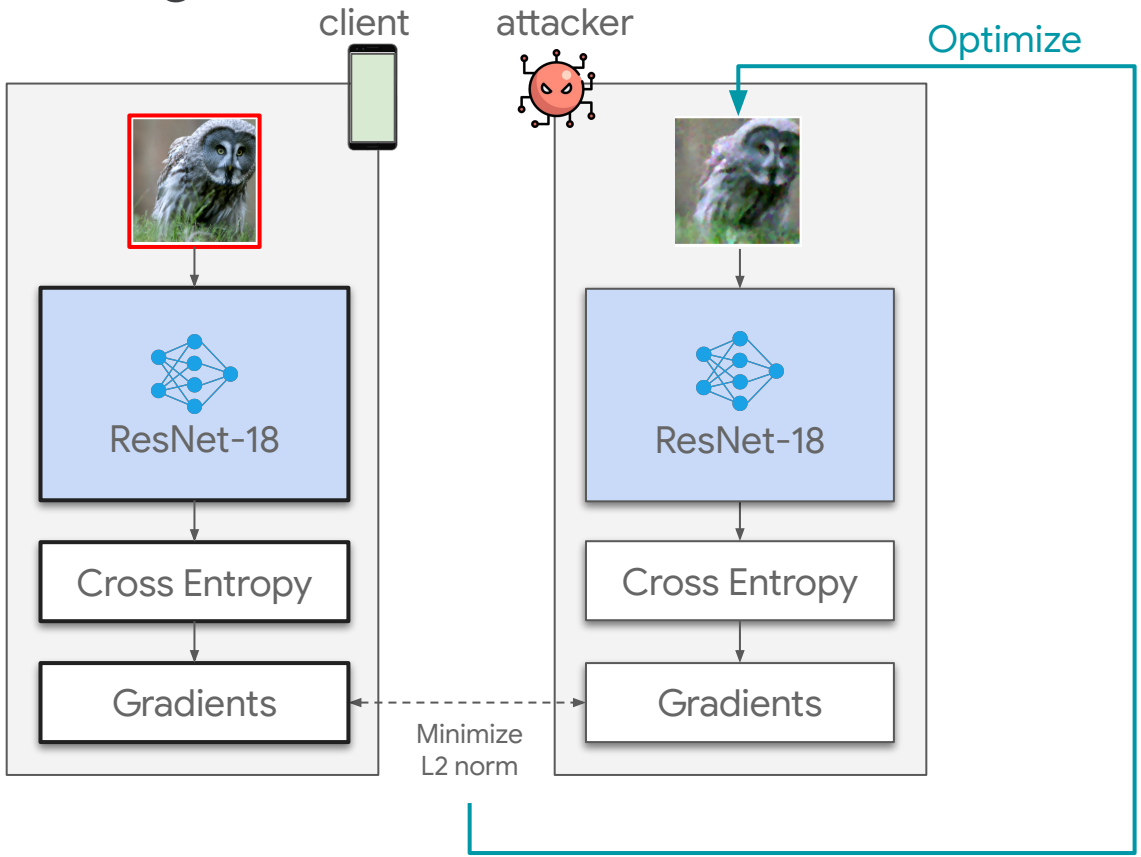
[2] Deep Leakage from Gradients [Zhu et. al., 2019]

Gradients Matching for Images

White box attack on the ***gradient***

Knowns: **Model, Loss function, Gradients, class label**

Unknown: **User's image**

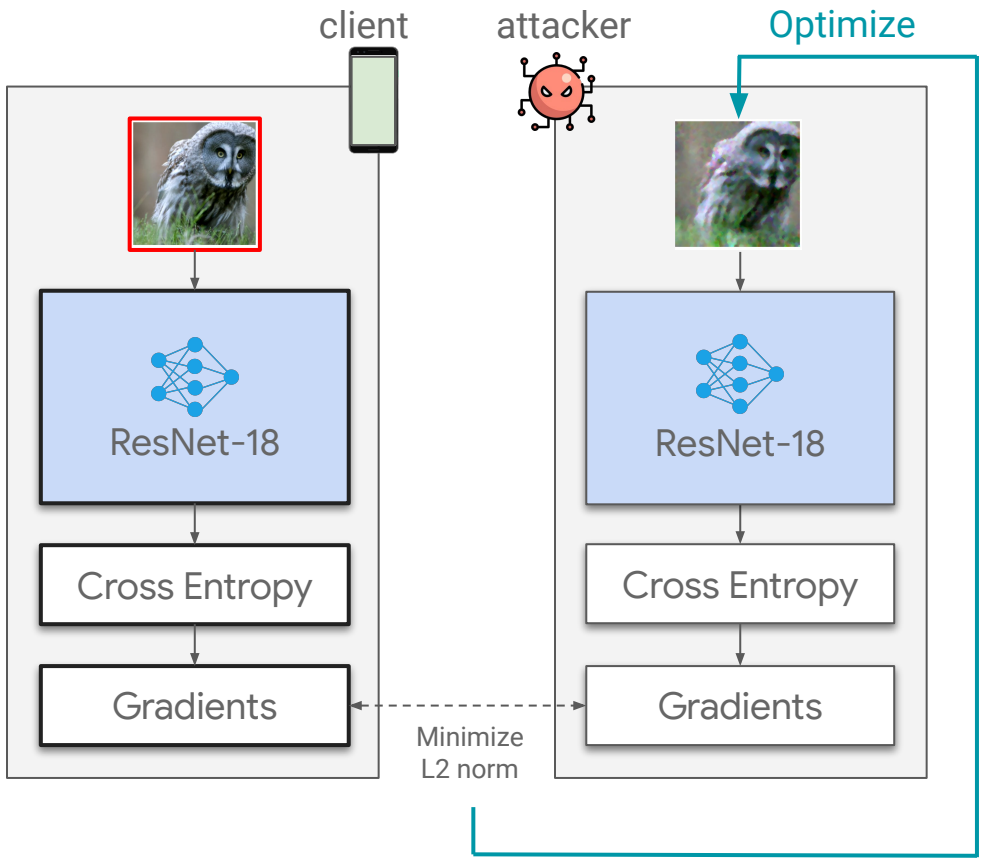


Contributions

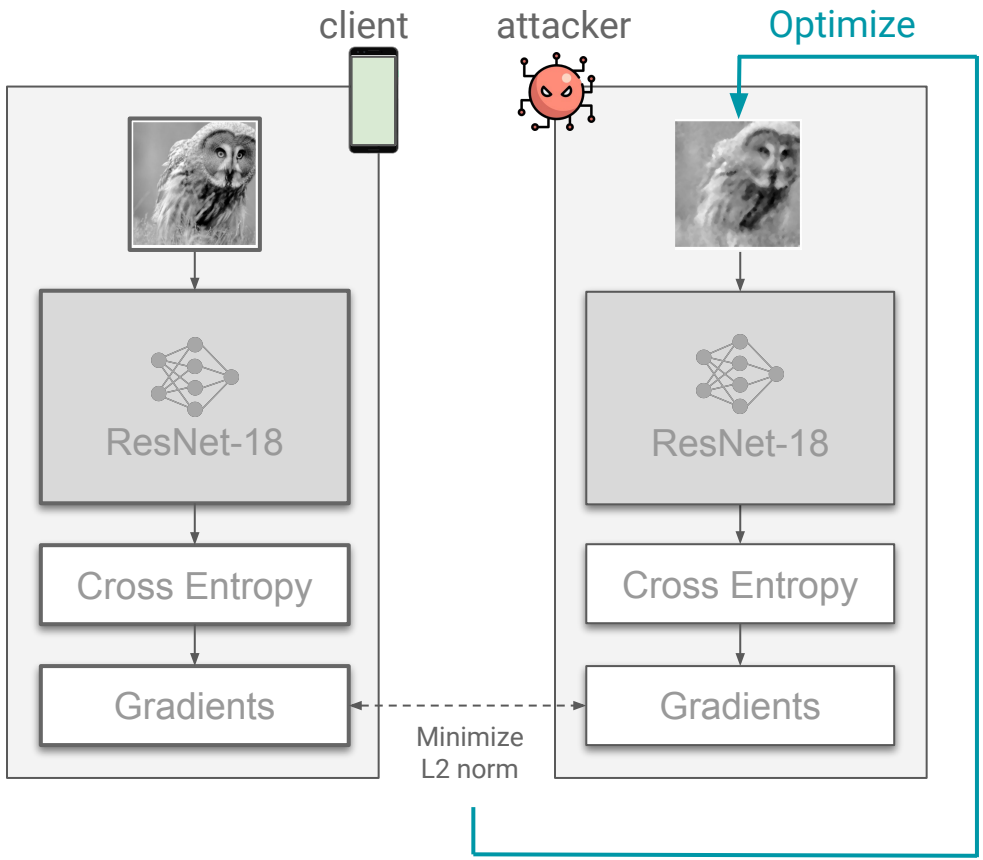
- First work to study information leakage from gradients in ASR training
  - Reveal speaker identity (SI) of an utterance from gradient
  - Propose Hessian-Free Gradients Matching
    - Input reconstruction without 2nd derivatives of the loss
- Demonstrate success using DeepSpeech training on LibriSpeech
  - Reveal SI with 34% top-1 accuracy (51% top-5 accuracy)
- Demonstrate that dropout can mitigate the success of our method
- Demonstrate our method in two complex regimes

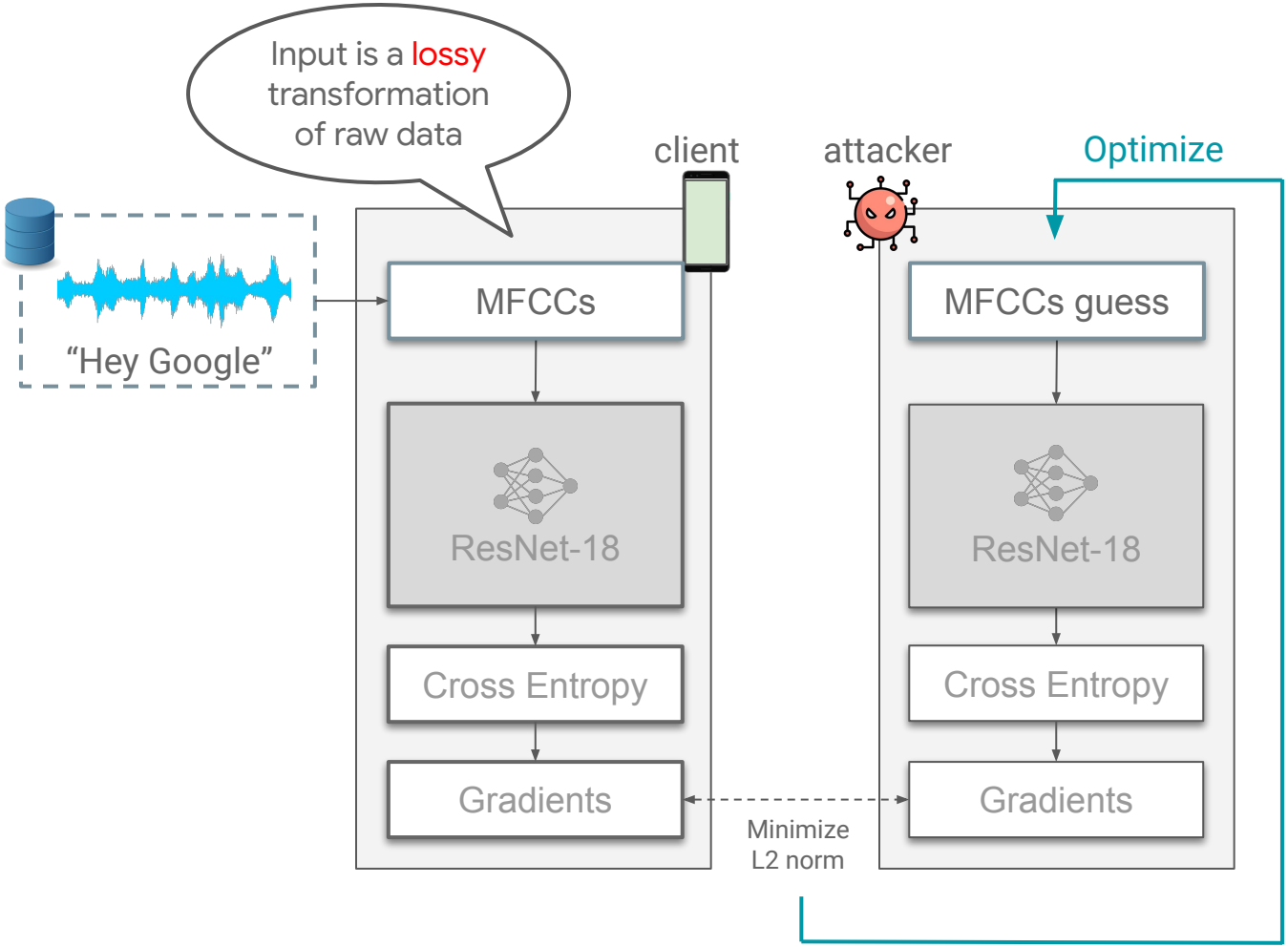
Challenges Applying GM to Speech

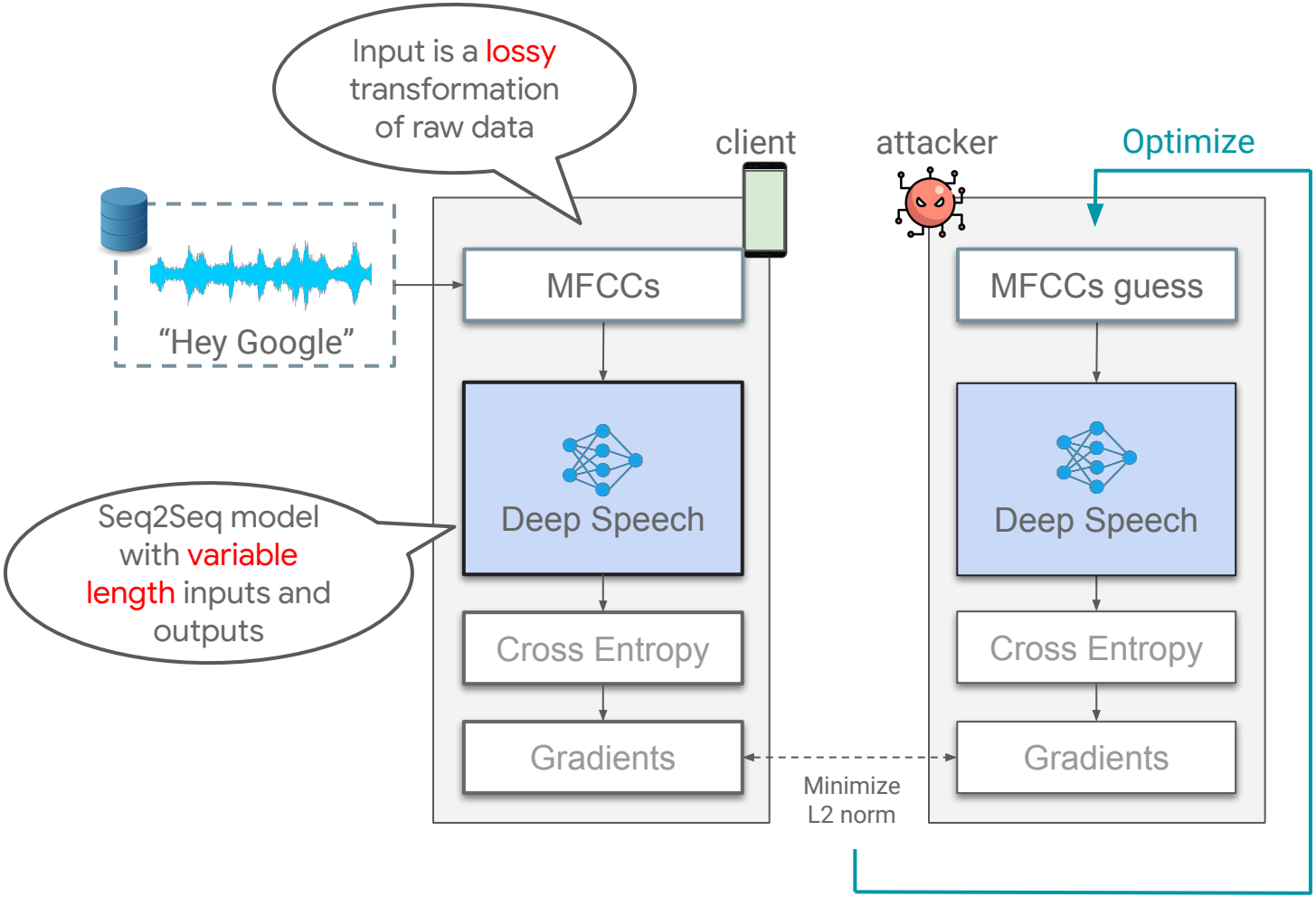
Just Apply Gradients Matching to Speech?

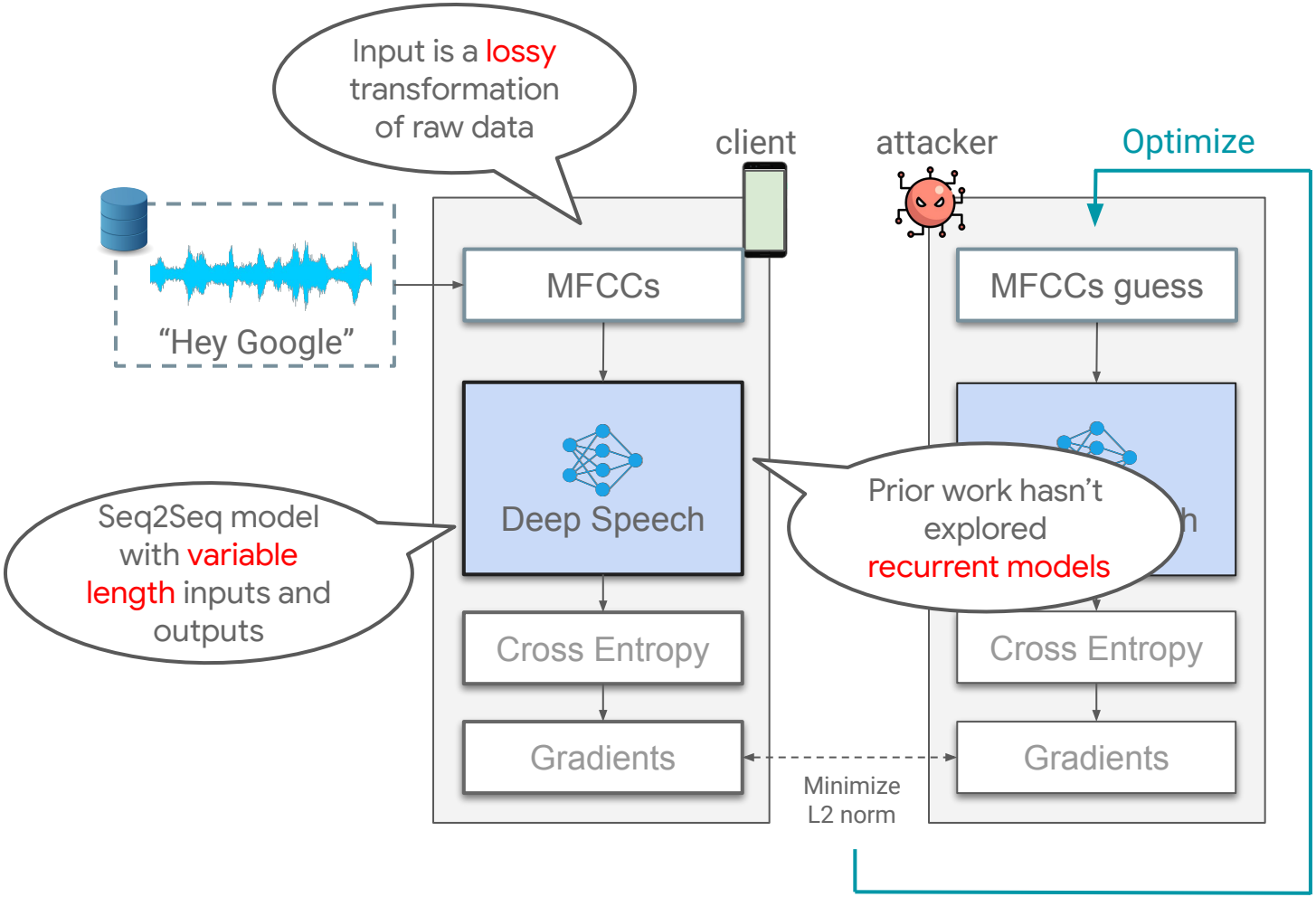


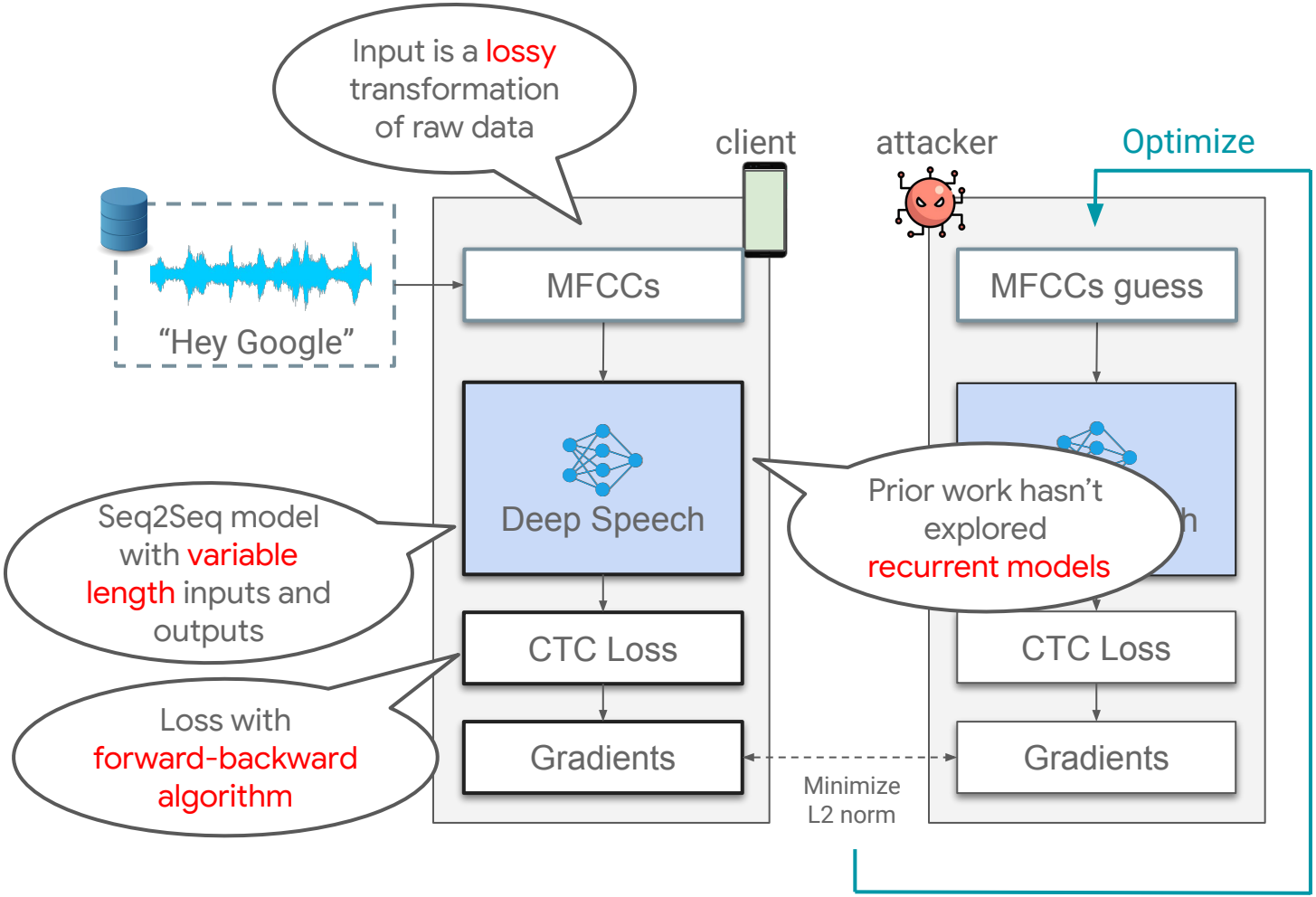
Many Differences!

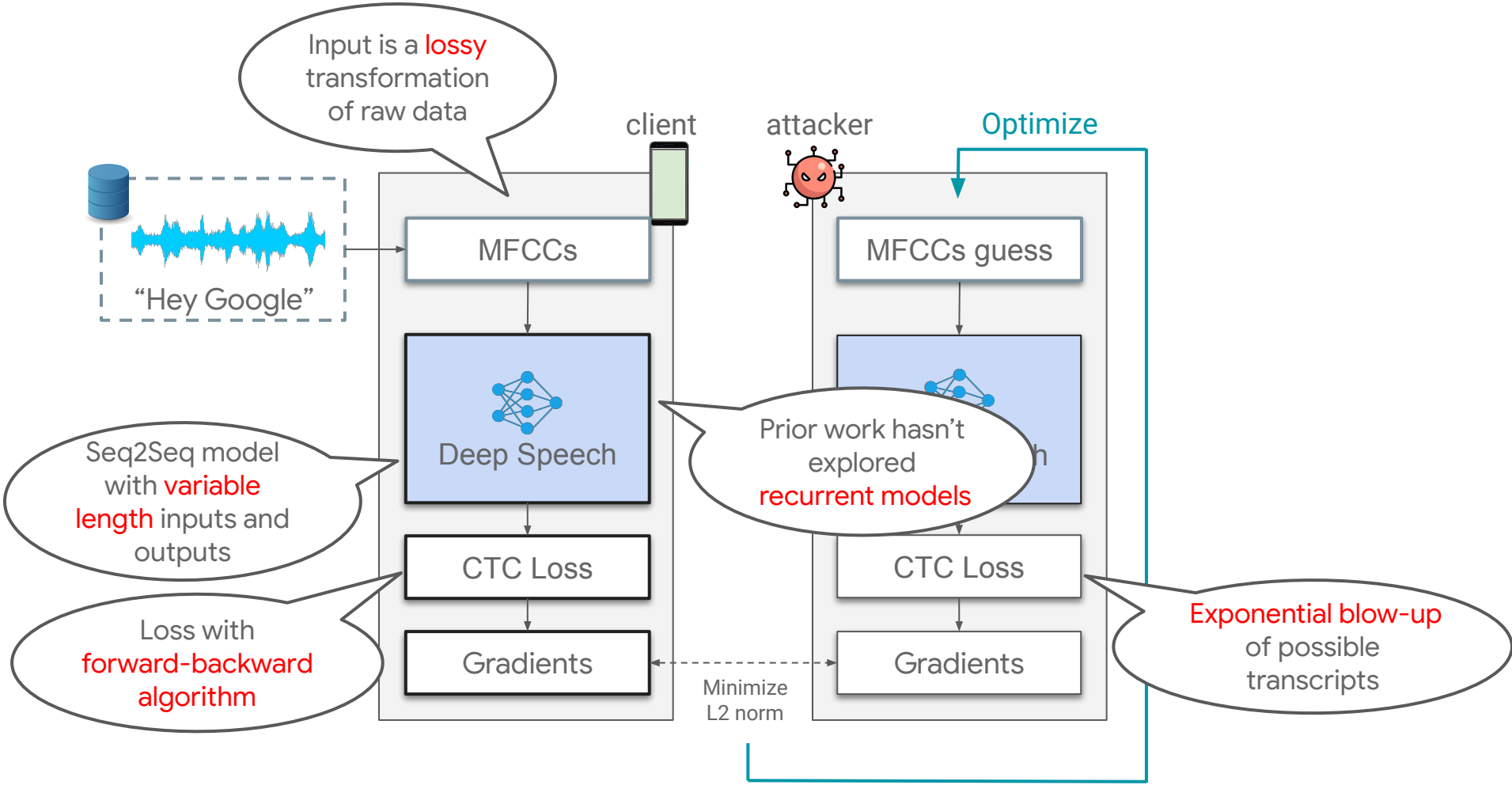


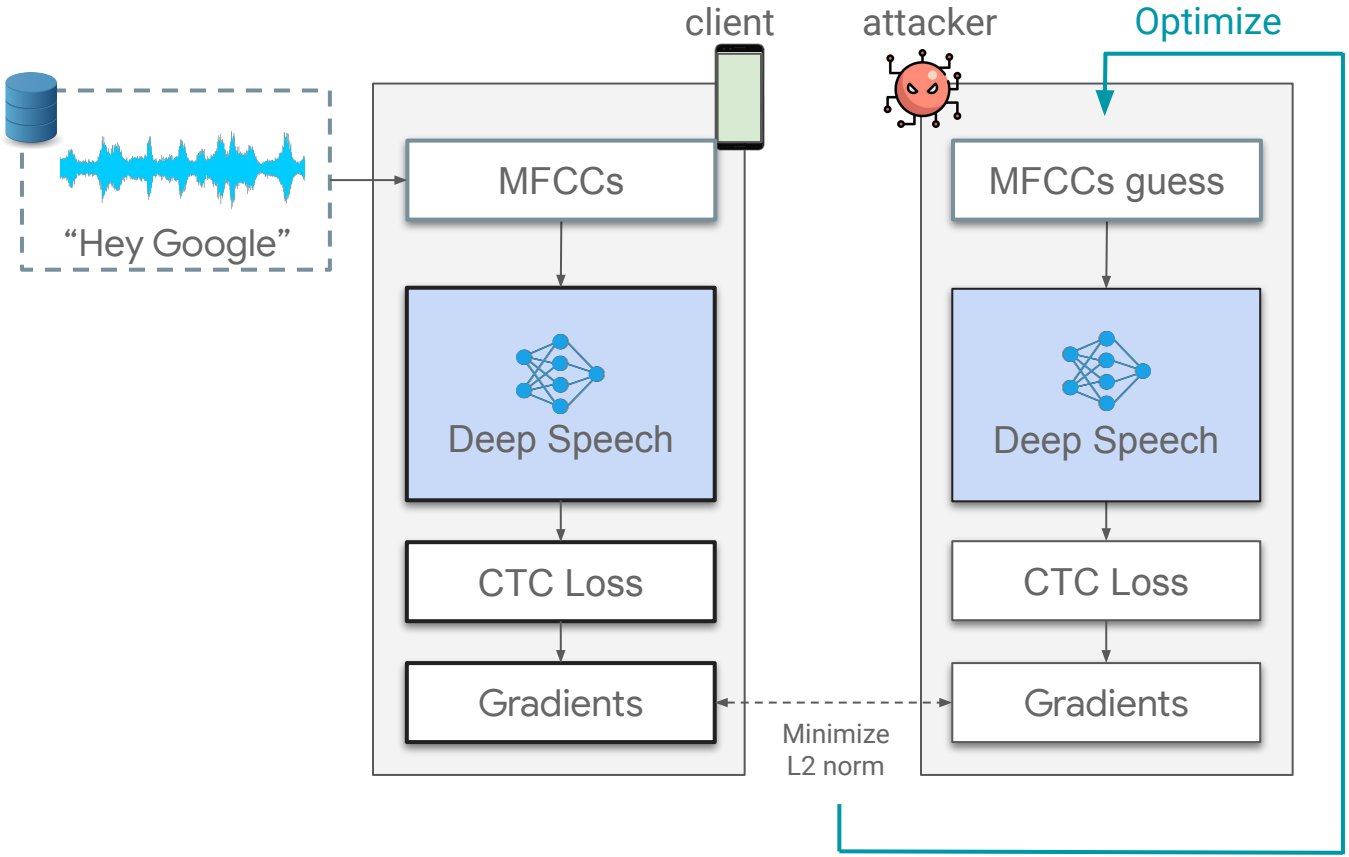

Input is a lossy transformation of raw data
client
attacker
Optimize
"Hey Google"
MFCCs
ResNet-18
Cross Entropy
Gradients
MFCCs guess
ResNet-18
Cross Entropy
Gradients
Minimize L2 norm


Input is a lossy transformation of raw data
client
attacker
Optimize
"Hey Google"
MFCCs
MFCCs guess
Deep Speech
Deep Speech
Seq2Seq model with variable length inputs and outputs
Cross Entropy
Cross Entropy
Gradients
Gradients
Minimize L2 norm


Input is a lossy transformation of raw data
client
attacker
Optimize
"Hey Google"
MFCCs
MFCCs guess
Deep Speech
Seq2Seq model with variable length inputs and outputs
Prior work hasn't explored recurrent models
Cross Entropy
Cross Entropy
Gradients
Gradients
Minimize L2 norm


Input is a lossy transformation of raw data
client
attacker
Optimize
"Hey Google"
MFCCs
MFCCs guess
Deep Speech
Prior work hasn't explored recurrent models
Seq2Seq model with variable length inputs and outputs
CTC Loss
CTC Loss
Loss with forward-backward algorithm
Gradients
Gradients
Minimize L2 norm


Input is a lossy transformation of raw data
client
attacker
Optimize
"Hey Google"
MFCCs
MFCCs guess
Deep Speech
Seq2Seq model with variable length inputs and outputs
Prior work hasn't explored recurrent models
CTC Loss
CTC Loss
Loss with forward-backward algorithm
Gradients
Gradients
Minimize L2 norm
Exponential blow-up of possible transcripts


client
attacker
Optimize
"Hey Google"
MFCCs
Deep Speech
CTC Loss
Gradients
MFCCs guess
Deep Speech
CTC Loss
Gradients
Minimize
L2 norm

Revisiting Gradients Matching

Standard Optimization: Find model parameters θ

- Loss: $L(x, \theta)$
- Objective: $\min_{\theta} L(x, \theta)$
- Use first-order methods, e.g., SGD
  - Uses $\nabla_{\theta} L(x, \theta)$

Revisiting Gradients Matching

Standard Optimization: Find model parameters θ

- Loss: $L(x, \theta)$
- Objective: $\min_{\theta} L(x, \theta)$
- Use first-order methods, e.g., SGD
  - Uses $\nabla_{\theta} L(x, \theta)$

Gradient Matching: Find model input x

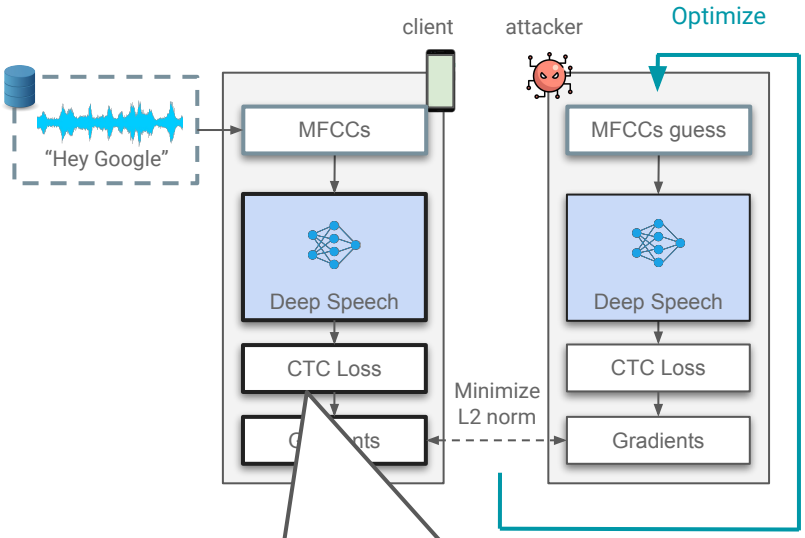
- Gradient Loss:

$$f(x) = \|\nabla_{\theta} L(x) - \nabla_{\theta} L(x')\|_2^2$$

  - $\nabla_{\theta} L(x')$: Client update (constant)

- Objective: $\min_{x} f(x)$

Revisiting Gradients Matching



2nd derivatives of CTC loss, RNN loops **not directly available** in common deep learning frameworks!

Gradient Matching: Find model input x

- Gradient Loss:

$$f(x) = \|\nabla_\theta L(x) - \nabla_\theta L(x')\|_2^2$$

  - $\nabla_\theta L(x')$: Client update (constant)
- Objective: $\min_x f(x)$
- For using first-order methods, we need $\nabla_x f(x)$
  - $f(x)$ requires $\nabla_\theta L(x)$
  - Thus, $\nabla_x f(x)$ requires $\nabla_x(\nabla_\theta L(x))$

Gradientless Descent (see e.g., [6])

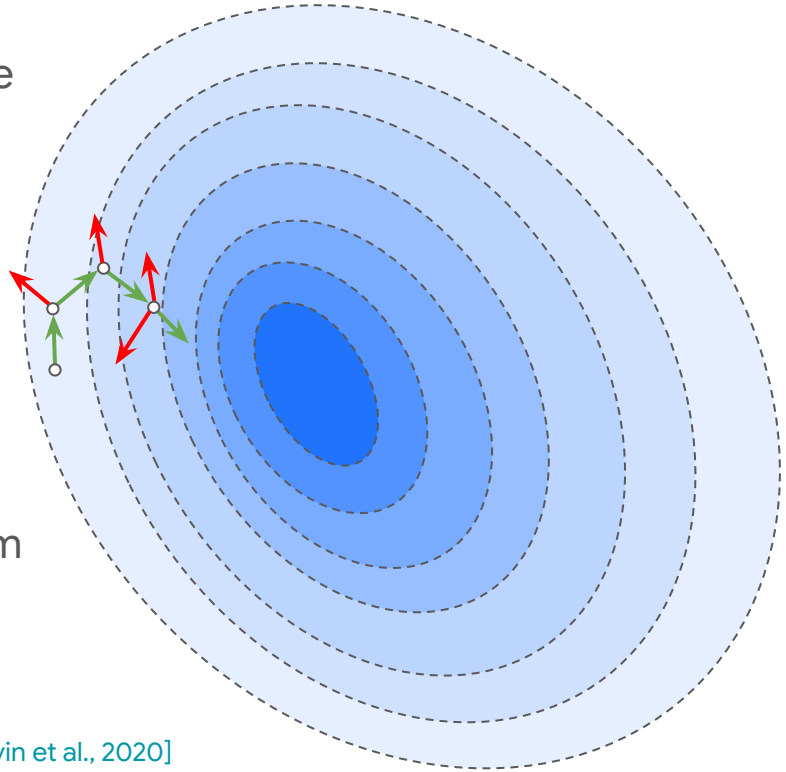
Consider a **random unit vector** in the param space

Does the loss reduce along this vector?

- If yes, take a step in that direction
- If no, do nothing

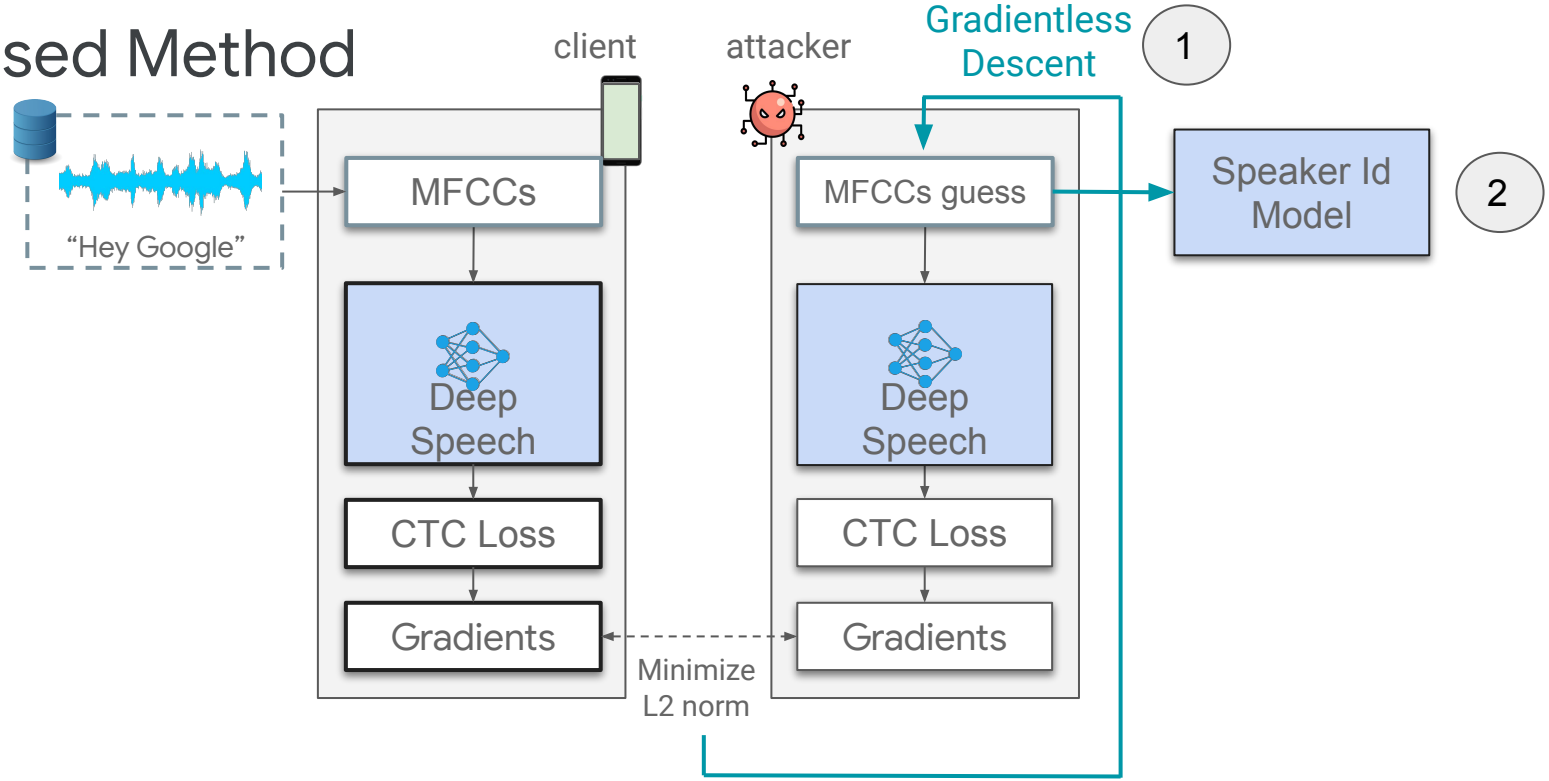
This coarse method turns out to be quite effective

- Used in, e.g., Reinforcement Learning
- Comes with convergence analysis ([6])
- In our expts, loss reduces for ~40% of “random vectors”



[6] Gradientless Descent: High-Dimensional Zeroth-Order Optimization [Golovin et al., 2020]

Proposed Method



A two-phase method to reveal speaker id:

1. Using Hessian-Free Gradients Matching (HFGM, based on Gradientless Descent) to reconstruct the input speech features.
2. Use a Speaker Id model to identify the speaker.

Experiments

Setup

- Model Architecture:
  - DeepSpeech: For the attack target
  - Deep Speaker [7]: To reveal speaker id
- Dataset:
  - LibriSpeech ASR corpus:
    - 300k utterances, 2.5k speakers
- For training Deep Speaker
  - use 5 utts for each speaker
- For reconstruction:
  - randomly sample 600 utts (not seen by Deep Speaker)

[7] Deep Speaker: an End-to-End Neural Speaker Embedding System [Li et. al., 2017]

Setup

- Phase 1 (Reconstruction):
  - Use untrained DeepSpeech model
  - Match only the last layer (~60k parameters) for each gradient
  - Sample 128 unit vectors per iteration of HFGM
- Phase 2 (Reveal Speaker Id):
  - Train Deep Speaker, obtain embeddings for each speaker
  - Identify the speaker of reconstructed utterance
- Evaluation Metrics:
  - Top-1 (Top-5) accuracy (%)
  - MAE, MRR (in the paper)

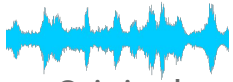
Example of Reconstruction

Original utterance

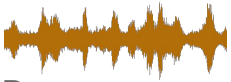
- “*you can’t stay here*”
- 59 frames, 1.2s of audio

Convergence

- Steps to reach 0.05 MAE: ~20k
- Time: ~3h



Original



Reconstructed

(you can’t stay here)

Speaker Id: Overall Results

- For reconstructed utterances:
  - Top-1: 34%, Top-5: 51%

Speaker Id: Overall Results

- For reconstructed utterances:
  - Top-1: 34%, Top-5: 51%
- For original utterances (upper bound)
  - Top-1: 42%, Top-5: 57%

Speaker id from reconstructed is close to original

Defense Methods: Training with Dropout

- Apply dropout to all layers except the projection layer (d: dropout rate)

d	TOP-1	TOP-5
0	34.0	51.0
0.1	0.8	2.0
0.2	0.0	0.5
0.3	0.1	0.3

- Dropout prevents the attacker from matching gradients

Defense Methods: Training with Dropout

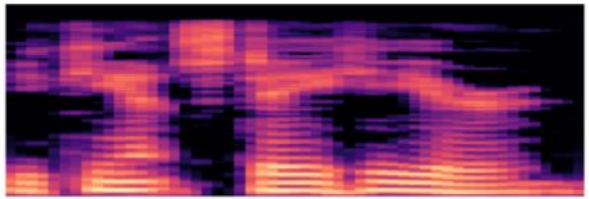
- Apply dropout to all layers except the projection layer (d: dropout rate)

d	TOP-1	TOP-5	WER (CLEAN)	WER (OTHER)
0	34.0	51.0	10.5	28.4
0.1	0.8	2.0	11.9	28.2
0.2	0.0	0.5	9.2	25.6
0.3	0.1	0.3	9.5	27.1

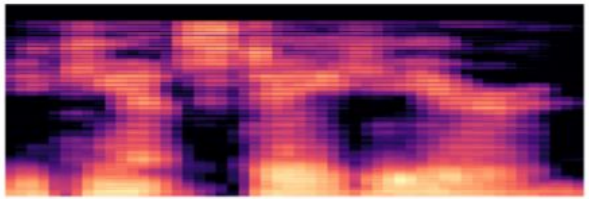
- Dropout prevents the attacker from matching gradients
- Does not hurt utility

Visualization of Speech Features

Original

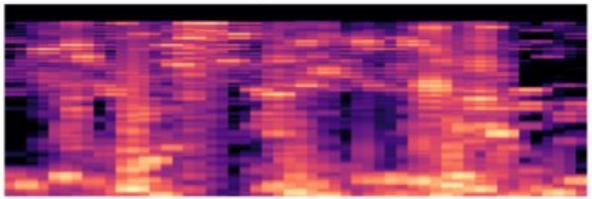


Reconstructed



Reconstructed utt looks similar to the original

Dropout 0.1



Defense methods significantly degrade signal quality

Transcript:
“where is my husband”

Additional Experiments: Average Gradients from Batches

- Reveal speaker identity from the batch of size 2/4/8

	TOP-1	TOP-5
ORIGINAL	42.0	57.0
BATCH SIZE 1	40.0	55.0
BATCH SIZE 2	37.0	54.0
BATCH SIZE 4	19.0	31.0
BATCH SIZE 8	5.0	11.0

(Results with 200 utts)

Additional Experiments: Multi-Step Updates from a Sample

- Reveal speaker identity from 2-step and 8-step model update

	TOP-1	TOP-5
ORIGINAL	42.0	57.0
1-STEP	40.0	55.0
2-STEP	26.5	39.5
8-STEP	24.5	39.0

(Results with 200 utts)

Summary

- First work to study information leakage from gradients in ASR training
  - Reveal speaker identity (SI) of an utterance from gradient
  - Proposed Hessian-Free Gradients Matching
- Demonstrated success using DeepSpeech training on LibriSpeech
- Demonstrated that dropout can mitigate the success of our method
- Demonstrated our method in two complex regimes