Fault Attacks on Neural Networks

Chester Rebeiro
IIT Madras

Image Classification with Neural Networks

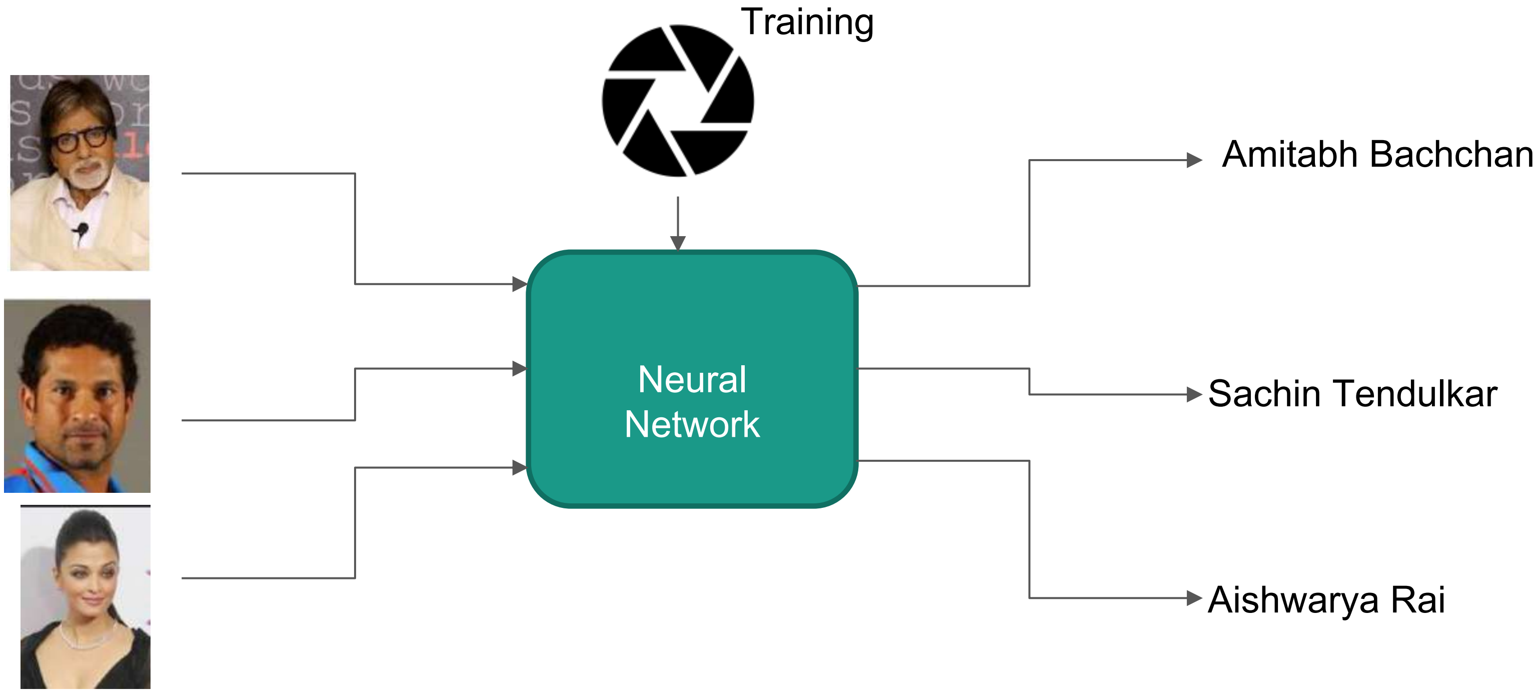
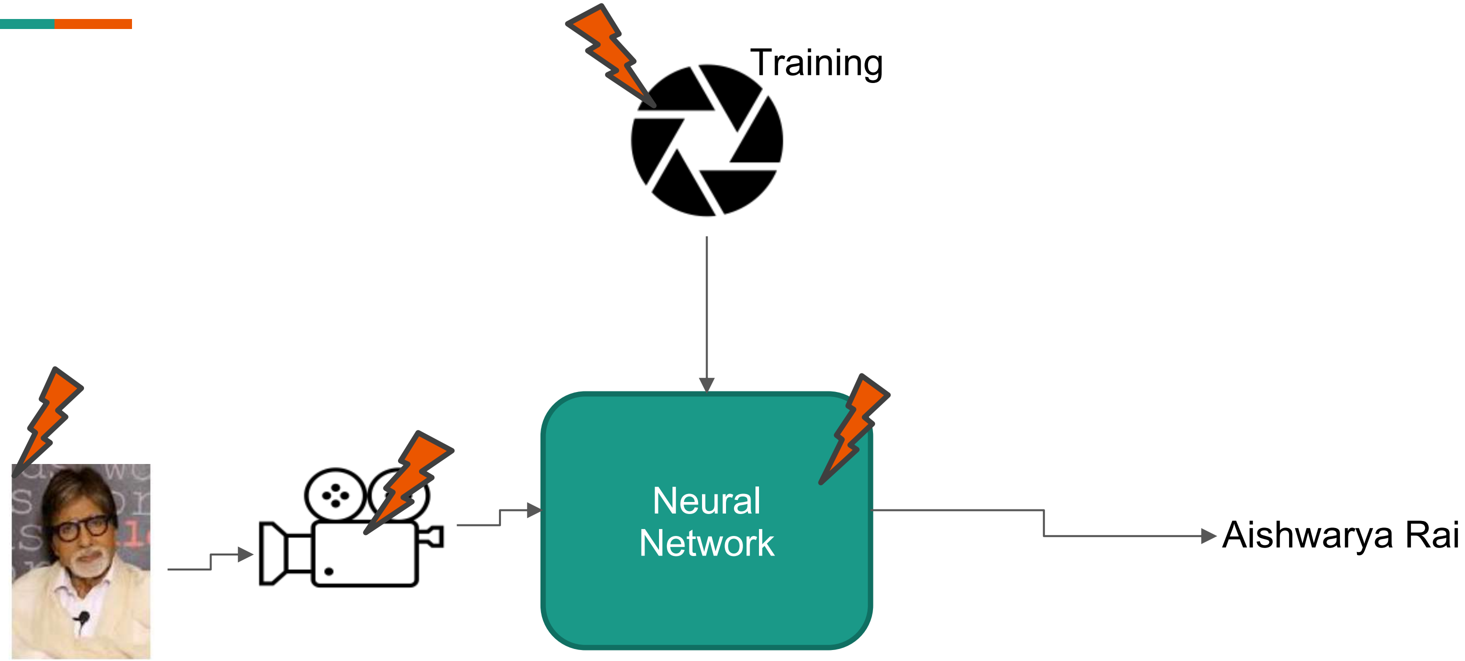


Image source : Wikipedia

Faults can cause misclassification



Attack categories and assumptions

Impersonation:

This is Aishwarya Rai

Dodging:

This is NOT Amitabh Bachchan

Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition, *Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, Michael K. Reiter, CCS 2016*

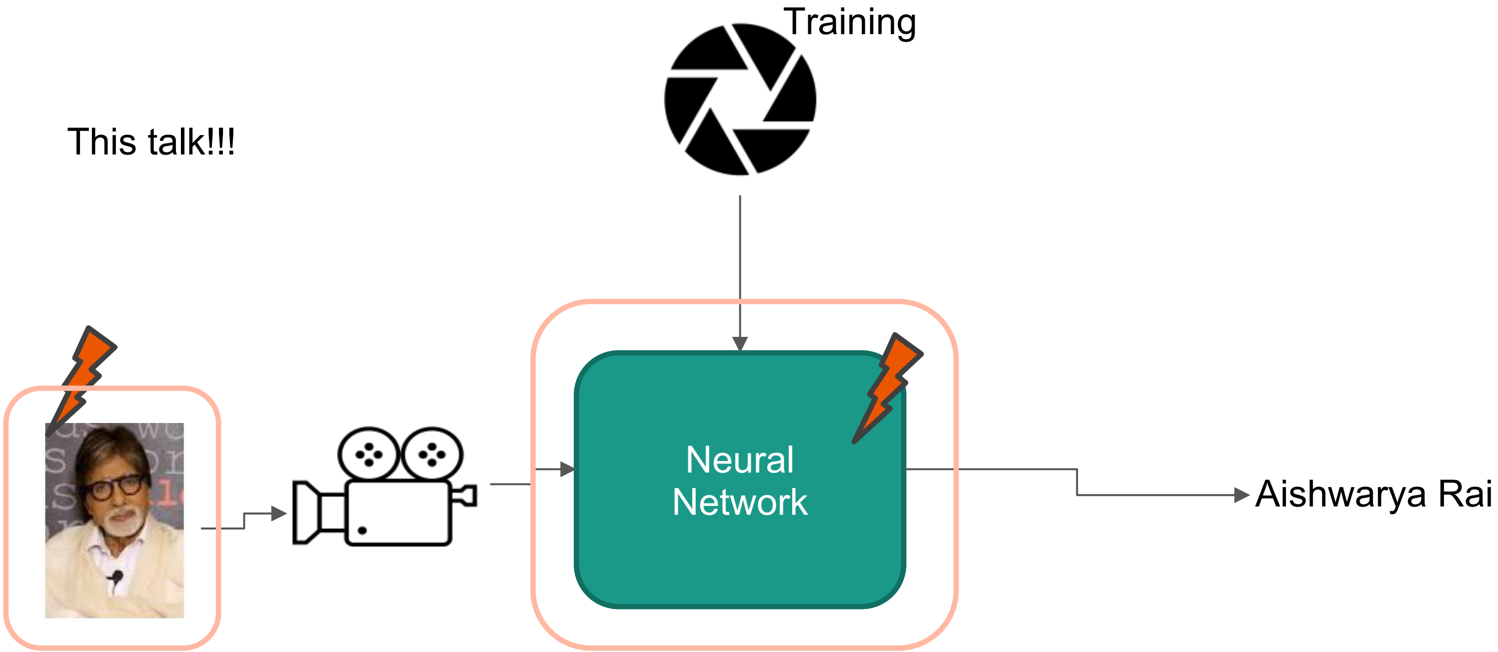
Image source : https://www.cs.cmu.edu/~sbhagava/papers/face-rec-ccs16.pdf

Attack Requirements

Requirements for a successful attack:

- Physically realizable.
- Inconspicuous (changes not easily noticed by observers)

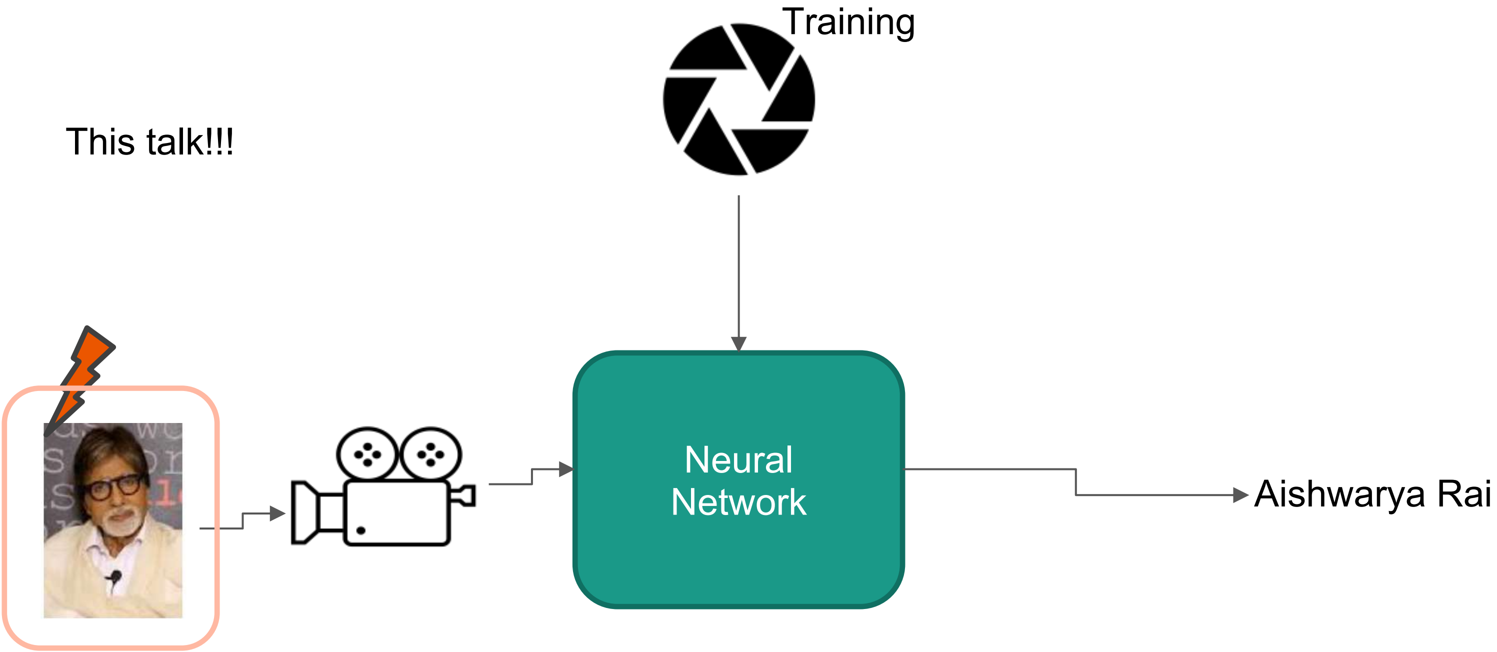
Faults can cause misclassification



Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition, *Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, Michael K. Reiter, CCS 2016*

Fault Injection on Deep Neural Networks, *-Yannan Liu, Lingxiao Wei, Bo Luo, Qiang Xu,ICCAD 2017*

Faults can cause misclassification



Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition, *Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, Michael K. Reiter, CCS 2016*

Fault Injection on Deep Neural Networks, -*Yannan Liu, Lingxiao Wei, Bo Luo, Qiang Xu,ICCAD 2017*

Formal definition

$$x \longrightarrow f(x)$$

Amitabh Bachchan (probability 0.75)
Sachin Tendulkar (probability 0.15)
Aishwarya Rai (probability 0.1)

Measurement of correctness

$$softmaxloss(f(x), c_x) = -\log\left(\frac{e^{\langle h_{c_x}, f(x)\rangle}}{\sum_{c=1}^{N} e^{\langle h_c, f(x)\rangle}}\right)$$

class corresponding to x

one hot encoding of c_x

(eg. 001, 010, 100)

$<*, *>$ inner product

Typically, softmaxloss is minimum for the correct predictions:

Amitabh Bachchan: 0.72
Sachin Tendulkar: 1.32
Aishwarya Rai: 1.37

Formalizing Attacks

- Impersonation $x \longrightarrow c_t \quad (\text{target class})$

$$\underset{r}{argmin}\,(softmaxloss(f(x+r), c_t))$$

minimum change to r so that softmaxloss is minimized

- Dodging

$$\underset{r}{argmin}\,(-softmaxloss(f(x+r), c_x))$$

minimum change to r so that softmaxloss is maximized

solve using Gradiant Descent

First Results

- Dodging: 100% success
- Impersonation: 100% success

original image
classified correctly

modified image
classified incorrectly
(dodged)

however, we are far from done….

Far from done...

- Not all perturbations are practical

original image
classified correctly

modified image
classified incorrectly
(dodged)

Making the attacks more practical

- **Utilize facial accessories**
  - easily implemented (example using an Inkjet printer)
  - Inconspicuous (many people wear glasses)

Making the attacks more practical

- **Utilize facial accessories**
  - easily implemented (example using an Inkjet printer)
  - Inconspicuous (many people wear glasses)

- **Enhancing Perturbations’ Robustness**

Aishwarya Rai

$$\underset{r}{argmin}\,(softmaxloss(f(x+r),c_t))$$

change to

$$\boxed{\underset{r}{argmin}\sum_{x\in X}(softmaxloss(f(x+r),c_t))}$$

minimum change to r so that
softmaxloss is minimized over a set of images

Making the attacks more practical

- **Utilize facial accessories**
  - easily implemented (example using an Inkjet printer)
  - Inconspicuous (many people wear glasses)

- **Enhancing Perturbations' Robustness**

$$\underset{r}{argmin} \sum_{x \in X} (softmaxloss(f(x+r), c_t))$$

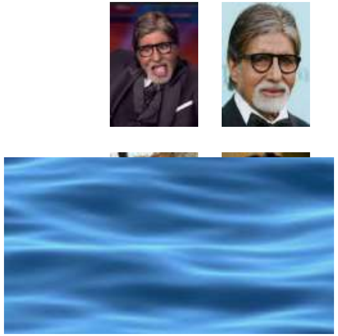
→ Aishwarya Rai

Making the attacks more practical

- **Utilize facial accessories**
  - easily implemented (example using an Inkjet printer)
  - Inconspicuous (many people wear glasses)

- **Enhancing Perturbations' Robustness**

$$\underset{r}{argmin} \sum_{x \in X} (softmaxloss(f(x+r), c_t))$$



Aishwarya Rai

- **Enhancing Perturbations' Smoothness**

$$\boxed{TV(r) = \sum_{i,j} ((r_{i,j} - r_{i+1,j})^2 + (r_{i,j} - r_{i,j+1})^2)^{1/2}}$$

difference between adjacent perturbations is minimized

Making the attacks more practical

- **Utilize facial accessories**
  - easily implemented (example using an Inkjet printer)
  - Inconspicuous (many people wear glasses)
- **Enhancing Perturbations' Robustness**

$$\underset{r}{argmin} \sum_{x \in X} (softmaxloss(f(x+r), c_t))$$

Aishwarya Rai

- **Enhancing Perturbations' Smoothness**

$$TV(r) = \sum_{i,j} ((r_{i,j} - r_{i+1,j})^2 + (r_{i,j} - r_{i,j+1})^2)^{1/2}$$

- **Enhance printability**

$$\boxed{NPS(\hat{p}) = \prod_{p \in P} |\hat{p} - p|}$$

Non-printability score

RGB printable colors

Making the attacks more practical

- Utilize facial accessories
  - easily implemented (example using an Inkjet printer)
  - Inconspicuous (many people wear glasses)
- Enh

ar

- Enh

TV

i,j

$$\underset{r}{\operatorname{argmin}} \Bigg(\Big(\sum_{x \in X} softmaxloss(x + r, c_t) \Big) + \kappa_1 \cdot TV(r) + \kappa_2 \cdot NPS(r) \Bigg)$$

- Enhance printability

$$NPS(\hat{p}) = \prod_{p \in P} |\hat{p} - p|$$

Making the attacks more practical

- Utilize facial accessories
  - easily implemented (example using an Inkjet printer)
  - Inconspicuous (many people wear glasses)

- Enhancing Perturbations' Robustness

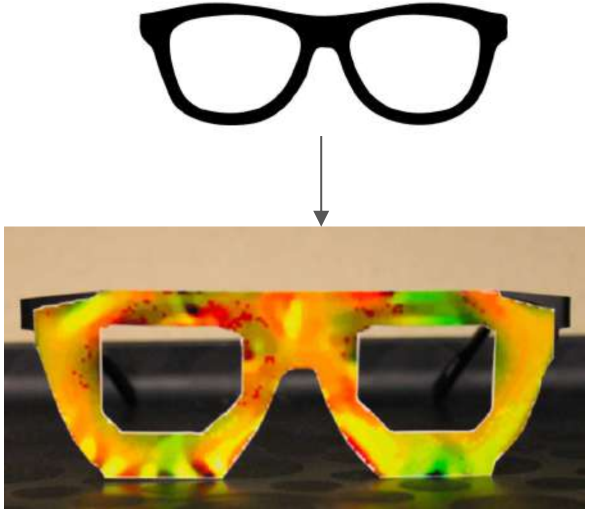
$$\underset{r}{argmin} \sum_{x \in X} (softmaxloss(f(x+r), c_t))$$

- Enhancing Perturbations' Smoothness

$$TV(r) = \sum_{i,j} ((r_{i,j} - r_{i+1,j})^2 + (r_{i,j} - r_{i,j+1})^2)^{1/2}$$

- Enhance printability

$$NPS(\hat{p}) = \prod_{p \in P} |\hat{p} - p|$$



DNNs used for the experiments

1. DNN_A trained to recognize celebrities with an accuracy of 98.95%.
2. DNN_B is trained to recognize 10 subjects: 5 people from author’s lab and 5 celebrities.
3. DNN_c was trained to recognize a larger set of people: 140 celebrities + 3 people from author’s lab.

DNN reference : https://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/parkhi15.pdf

Dodging Attacks

Dodging

DNN	Subject (attacker) info		Dodging results	
	Subject	*Identity*	*SR*	$E(p(\text{correct-class}))$
DNN_B	S_A	3rd author	100.00%	0.01
	S_B	2nd author	97.22%	0.03
	S_C	1st author	80.00%	0.35
DNN_C	S_A	3rd author	100.00%	0.03
	S_B	2nd author	100.00%	<0.01
	S_C	1st author	100.00%	<0.01

success rate

Expected probability of the correct class
Prior to dodging, this was at-least 0.85

Image source : https://www.cs.cmu.edu/~sbhagava/papers/face-rec-ccs16.pdf

Impersonation Attacks

	Subject (attacker) info		Impersonation results		
DNN	*Subject*	*Identity*	*Target*	*SR*	*SRT*
DNN_B	S_A	3rd author	Milla Jovovich	87.87%	48.48%
	S_B	2nd author	S_C	88.00%	75.00%
	S_C	1st author	Clive Owen	16.13%	0.00%
DNN_C	S_A	3rd author	John Malkovich	100.00%	100.00%
	S_B	2nd author	Colin Powell	16.22%	0.00%
	S_C	1st author	Carson Daly	100.00%	100.00%

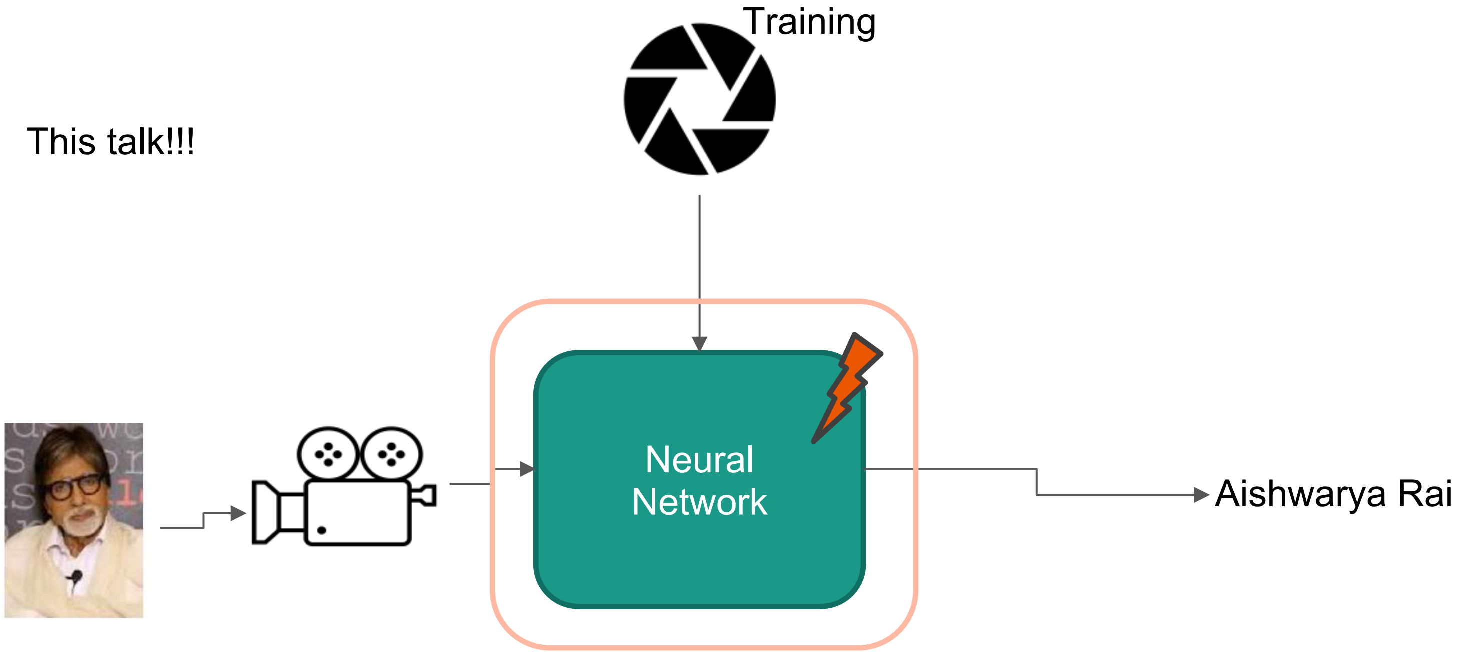
Success Rate

Success Rate with Threshold

Image source : https://www.cs.cmu.edu/~sbhagava/papers/face-rec-ccs16.pdf

Faults during the neural network

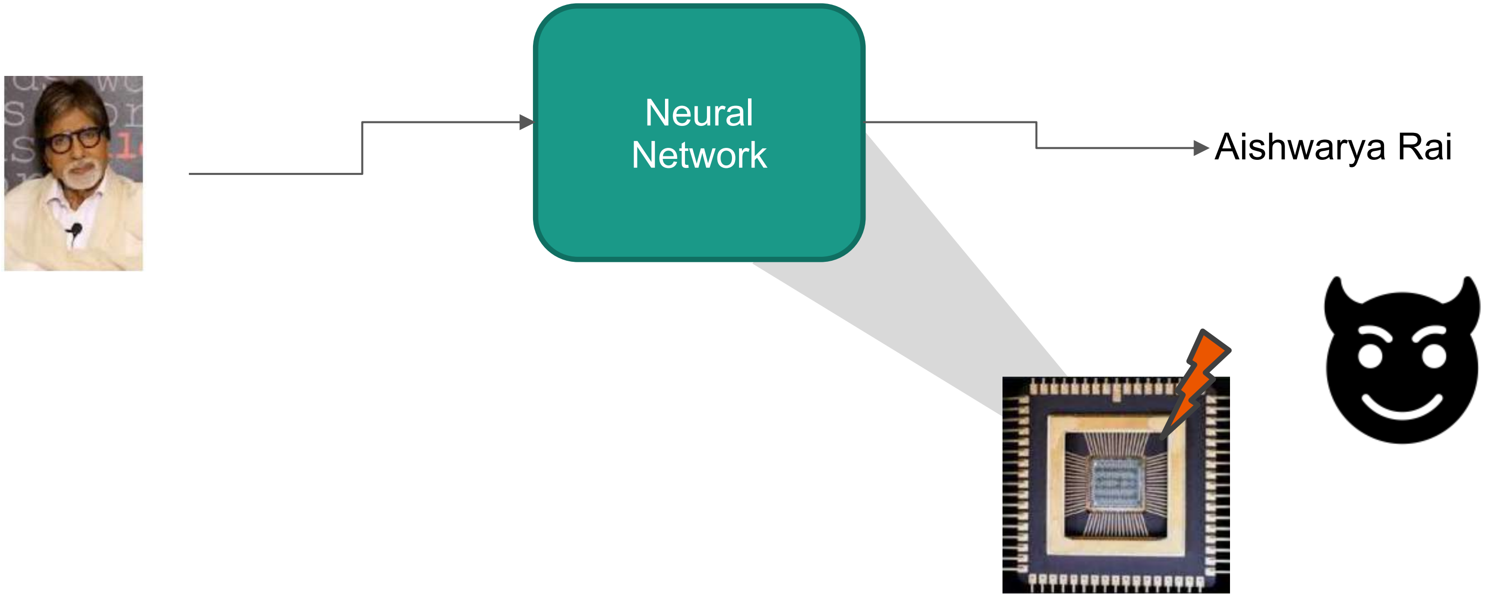
This talk!!!



Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition, *Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, Michael K. Reiter, CCS 2016*

Fault Injection on Deep Neural Networks, ***-Yannan Liu, Lingxiao Wei, Bo Luo, Qiang Xu,ICCAD 2017***

Faults in the neural network

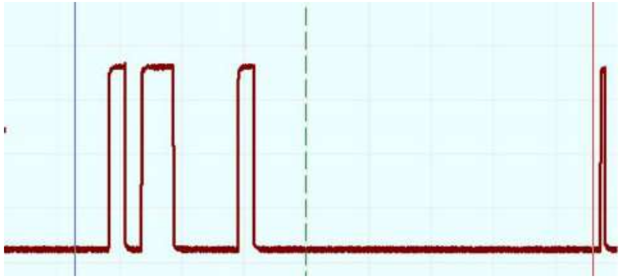


Fault Injection on Deep Neural Networks, ***-Yannan Liu, Lingxiao Wei, Bo Luo, Qiang Xu,ICCAD 2017***

Injecting Faults in Semiconductor Devices

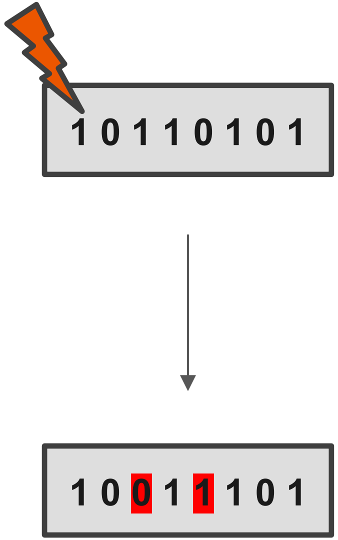
Laser fault injection

Row hammer

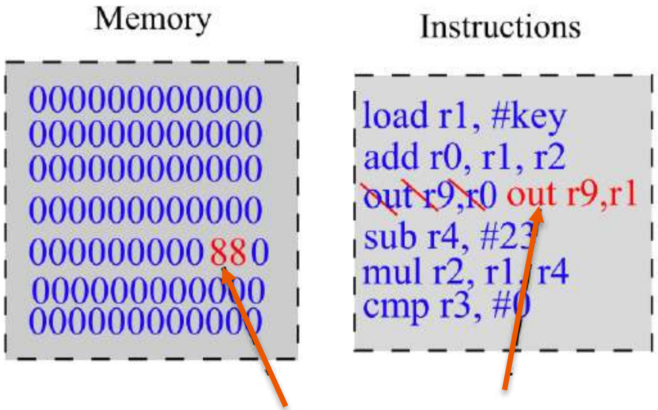


Glitches in clock or power lines

Fault Injection

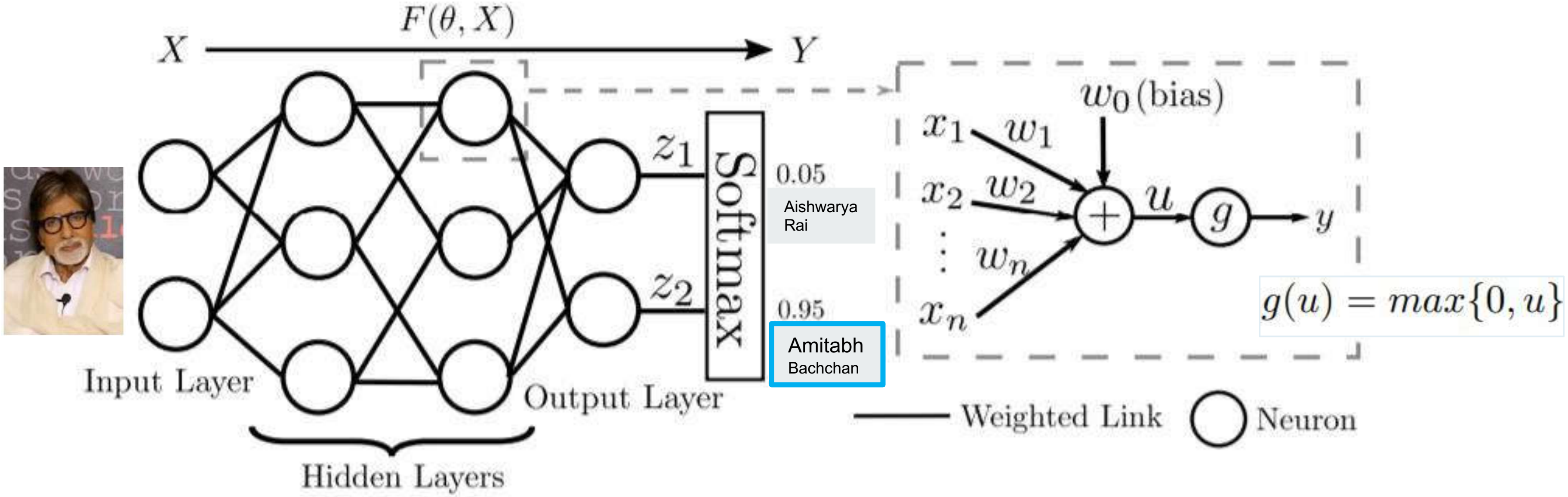


perturbs memory or registers

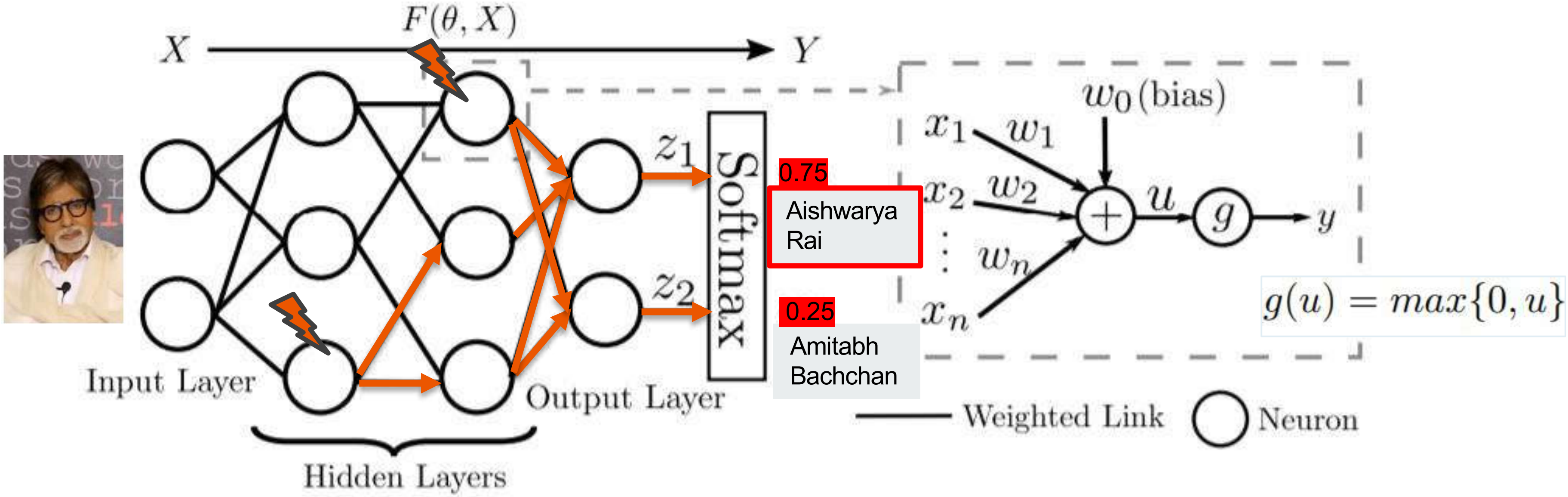


causes faults in data, modifies instructions or skips instructions

Neural Network Architecture



Faults in Neural Network



Inject faults in one or more neurons so that dodging or impersonation can be achieved.
Faults injected by changing the weights/bias in the neuron

Properties of an attack

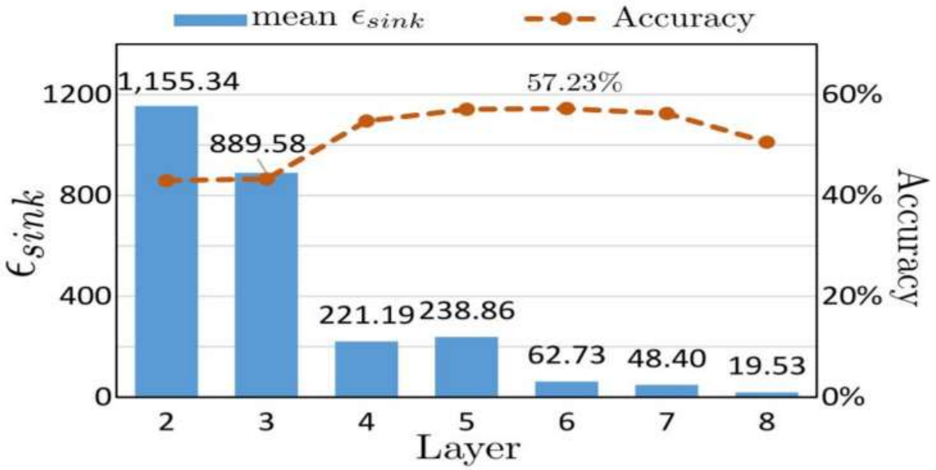
Efficiency: The misclassification should be efficient

Stealthiness:Need to make minimum changes to the Neural Network to achieve the desired impersonation.

Efficiency

Efficiency: The misclassification should be efficient

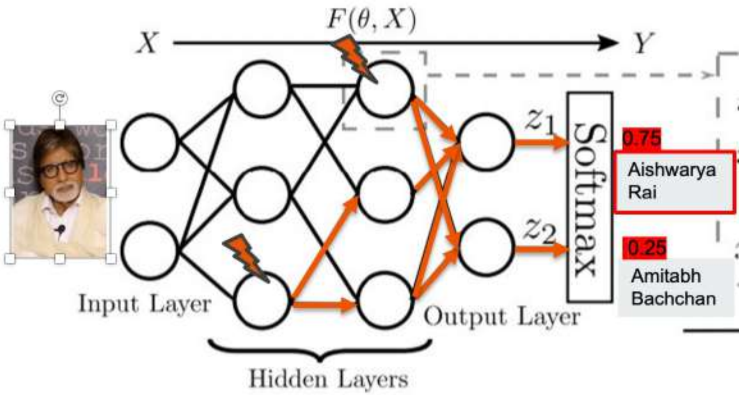
Stealthiness: Need to make minimum changes to the Neural Network to achieve the desired impersonation.



Change in bias to achieve impersonation depends on the layer

Stealthiness

Efficiency: The misclassification should be efficient



Need to make minimum changes to the Neural Network to achieve the desired impersonation.

Stealthiness: Need to make minimum changes to the Neural Network to achieve the desired impersonation.

Achieving Stealthiness with Gradiant Descent

Efficiency: The misclassification should be efficient

Stealthiness:Need to make minimum changes to the Neural Network to achieve the desired impersonation.

$$y_1 = F(x_1, \theta)$$

output probability of class

input image

state of the NN

Achieving Stealthiness with Gradiant Descent

Efficiency: minimizing the number of changes required to achieve the needed misclassification

$$y_1 = F(x_1, \theta)$$

$$\Big\downarrow \text{fault}$$

$$y_2 = F(x_1, \theta')$$

Stealthiness:Need to make minimum changes to the Neural Network to achieve the desired impersonation.

optimization function $\boxed{y_2 - \lambda|\theta - \theta'|}$

increase probability

decrease changes in model

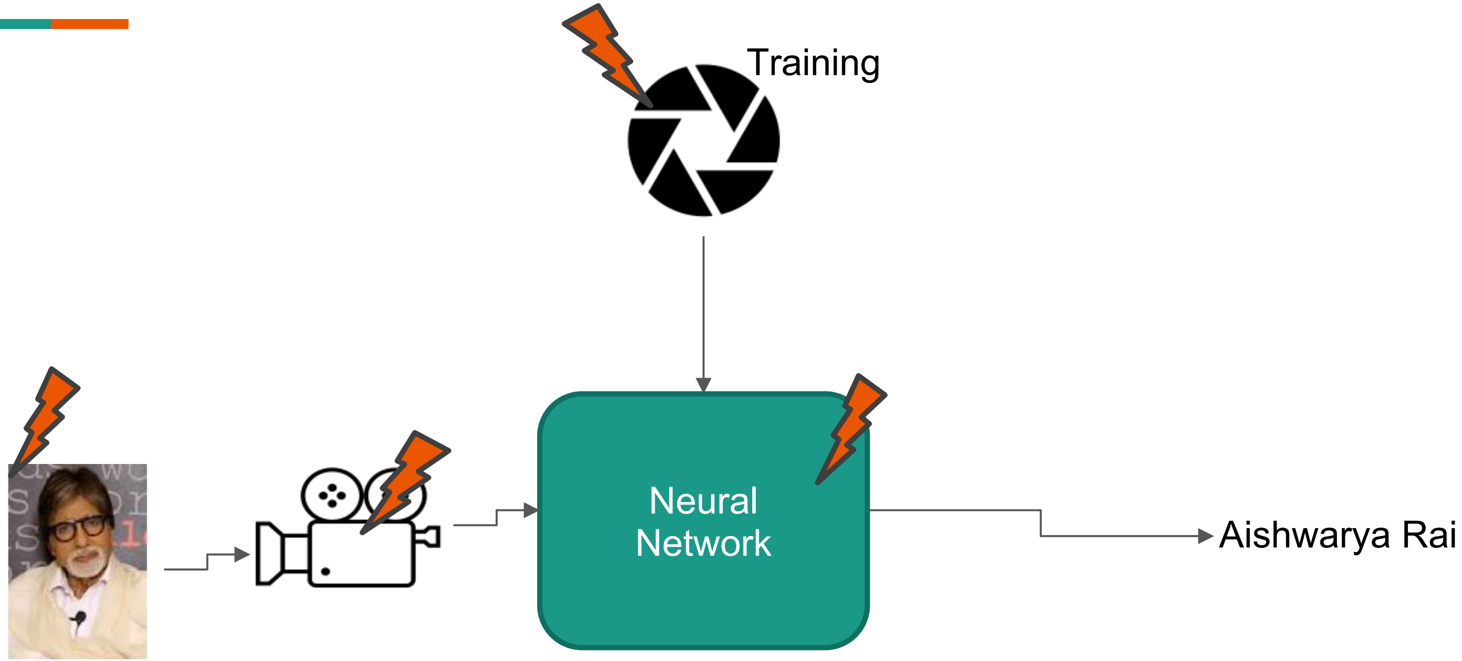
Classification accuracy and the number of modified parameters after attack

	MNIST				CIFAR			
	CA		# of MP		CA		# of MP	
	w/o MC	MC	w/o MC	MC	w/o MC	MC	w/o MC	MC
LW 2	46.38%	59.89%	200	19	12.98%	25.06%	2334	283
LW 3	56.22%	68.62%	7240	221	12.98%	54.54%	57009	1354
LW 4	58.80%	84.93%	21660	1077	25.34%	76.45%	129759	697
LW 5	46.07%	90.44%	43280	1215	23.39%	73.73%	195502	2321
LW 6	65.23%	95.20%	86520	2345	11.68%	81.66%	115127	198
LW 7	89.88%	97.01%	72150	5734	13.87%	80.57%	19109	43
LW 8	95.12%	96.86%	1439	125	13.02%	80.32%	1147	2
Global-wise	26.68%(§)	63.70%	232559(§)	1170	10.00%(§)	50.97%	519691(§)	425
	Accuracy		Modifications		Accuracy		Modifications	

MC: modification compression

Image source : https://ieeexplore.ieee.org/document/8203770

Summary



Open research problems

Unlike cryptographic attacks, adversarial attacks on ML models are relatively new. Our aim is to do the following in the field of adversarial machine learning:

- formal models for the adversarial attacks on a given implementation
- frameworks that automatically identify hot-spots of vulnerabilities
- tools that automatically fix vulnerabilities
- relationships between various forms of adversarial attack possible

Thank You