ADVERSARIAL EXAMPLES

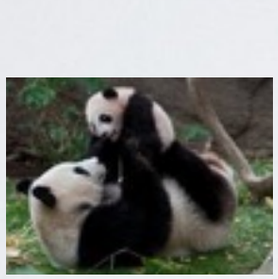
(In 15 minutes or less)

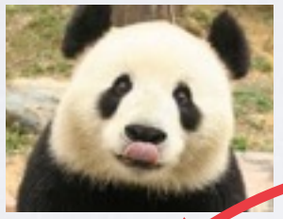
Neill Patterson, MscAC

PART I - BASIC CONCEPTS

WE TRAIN MODELS BY TAKING GRADIENTS W.R.T. WEIGHTS

$$w \leftarrow w - \eta \nabla J_w$$





"Panda"

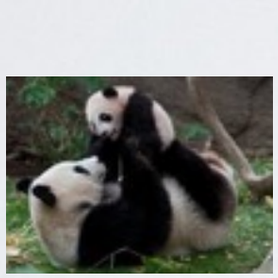
Change weights via
gradient descent

WE'RE GOING TO TAKE GRADIENTS W.R.T. PIXELS INSTEAD

$$x \leftarrow x \pm \eta \nabla J_x$$

WE ARE GOING TO TAKE GRADIENTS W.R.T. PIXELS INSTEAD

$$x \leftarrow x \pm \eta \nabla J_x$$



"Panda"

Change pixels via

gradient descent

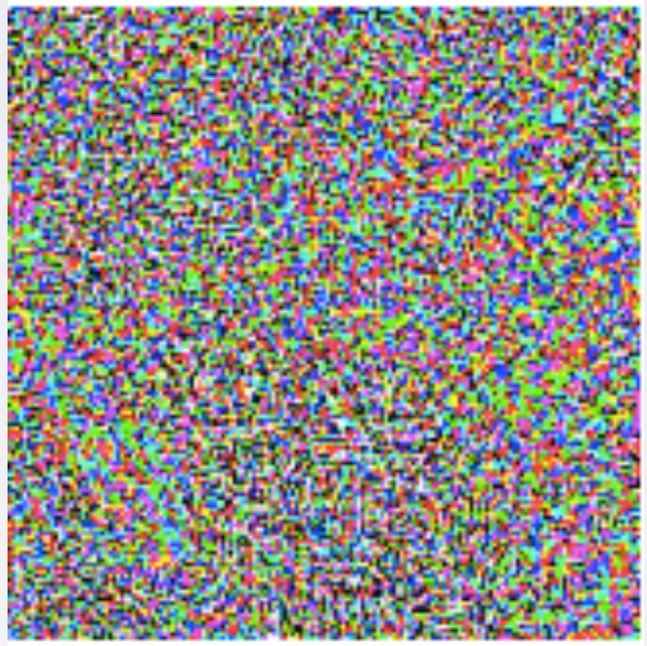
"Vulture"

KEY IDEA: ADD SMALL, WORST-CASE PIXEL DISTORTION TO CAUSE MISCLASSIFICATIONS

"Panda"

58% confidence

+



=

"Gibbon"

99% confidence

THINK OF ADVERSARIAL EXAMPLES AS WORST-CASE DOPPLEGÄNGERS

DEMO

Sanja Fidler

Fiddler Crab

PART II - HARNESSING ADVERSARIAL EXAMPLES

KEY IDEA: MAKE TRAINING MORE DIFFICULT TO GET STRONGER MODELS

(DROPOUT, RANDOM NOISE, ETC)

TRAIN WITH ADVERSARIAL EXAMPLES FOR BETTER GENERALIZATION

THE FAST GRADIENT SIGN METHOD OF IAN GOODFELLOW

QUICKLY GENERATING ADVERSARIAL EXAMPLES

WHAT DIRECTION SHOULD YOU MOVE TOWARDS?

INSTEAD OF MOVING TOWARDS A SPECIFIC TYPE OF ERROR, MOVE AWAY FROM THE CORRECT LABEL

"House"

"Panda"

"Vulture"

"Truck"

HOW BIG A STEP SHOULD YOU TAKE IF YOU WANT IMPERCEPTIBLE DISTORTION?

PIXELS ARE STORED AS SIGNED 8-BIT INTEGERS. ADD JUST LESS THAN 1-BIT OF DISTORTION TO EACH PIXEL

$$0.07 < \frac{1}{2^7} \approx 0.08$$

WE WANT PRECISELY THIS AMOUNT OF DISTORTION, SO NO MATTER HOW SMALL (OR BIG) THE GRADIENT, JUST TAKE THE SIGN OF IT AND MULTIPLY BY 0.07

$$x + 0.07 \times sign(\nabla J_x)$$

INCORPORATING ADVERSARIAL EXAMPLES INTO YOUR COST FUNCTION

GENERATE ADVERSARIAL EXAMPLES AT EACH ITERATION OF TRAINING, BUT DON'T WANT TO KEEP THEM AROUND IN MEMORY FOREVER

INSTEAD, MODIFY THE COST FUNCTION TO BE A COMBINATION OF ORIGINAL AND ADVERSARIAL INPUTS

New cost function

$$J(\boldsymbol{\theta}, \mathbf{x}, y) =$$

Parameters

inputs

labels

Old cost function

$$J(\boldsymbol{\theta}, \mathbf{x}, y) = \quad J(\boldsymbol{\theta}, \mathbf{x}, y) \quad +$$

Old cost function

$$J(\boldsymbol{\theta}, \mathbf{x}, y) = J(\boldsymbol{\theta}, \mathbf{x}, y) + J(\boldsymbol{\theta}, \underbrace{\mathbf{x} + \epsilon sign \nabla_{\mathbf{x}} J}_{\text{Adversarial example}}, y)$$

$$J(\boldsymbol{\theta}, \mathbf{x}, y) = \alpha\, J(\boldsymbol{\theta}, \mathbf{x}, y) + (1-\alpha) J(\boldsymbol{\theta}, \mathbf{x} + \epsilon sign \nabla_{\mathbf{x}} J, y)$$

mixing components

$$J(\boldsymbol{\theta}, \mathbf{x}, y) = \alpha J(\boldsymbol{\theta}, \mathbf{x}, y) + (1 - \alpha) J(\boldsymbol{\theta}, \mathbf{x} + \epsilon sign \nabla_{\mathbf{x}} J, y)$$

"Train with a mix of original and adversarial examples"

NOW DO S.G.D. ON THIS NEW COST FUNCTION, BY TAKING GRADIENTS W.R.T. WEIGHTS

$$w \leftarrow w - \eta \nabla \tilde{J}_w$$

PART III - MISCELLANEOUS TIPS FOR TRAINING

YOU NEED MORE MODEL CAPACITY

(ADVERSARIAL EXAMPLES DO NOT LIE ON THE MANIFOLD OF REALISTIC IMAGES)

FOR EARLY STOPPING, BASE YOUR DECISION ON THE VALIDATION ERROR OF ADVERSARIAL EXAMPLES ONLY

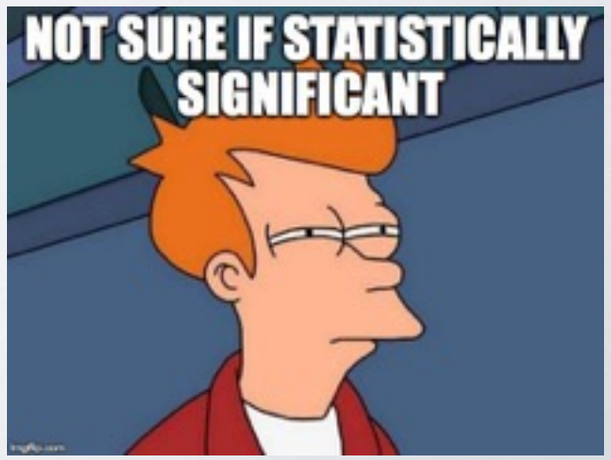
RESULTS

BETTER GENERALIZATION ABOVE AND BEYOND DROPOUT

0.94% error (MNIST) → 0.84% error

BETTER GENERALIZATION ABOVE AND BEYOND DROPOUT

0.94% error (MNIST) → 0.84% error



RESISTANCE TO ADVERSARIAL EXAMPLES

89.4% error
(97.6% confidence)

17.9% error

MATHEMATICAL PROPERTIES OF ADVERSARIAL EXAMPLES

MATHEMATICAL PROPERTIES OF ADVERSARIAL EXAMPLES

(Ain't nobody got time for that)

THANK YOU FOR YOUR TIME!