Universal Adversarial Perturbations

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, Pascal Frossard

Presented by Maximilian Reith

December 3, 2024

Contents

Introduction

Introduction

- Universal adversarial perturbations are quasi-imperceptible perturbations designed to fool deep neural networks.

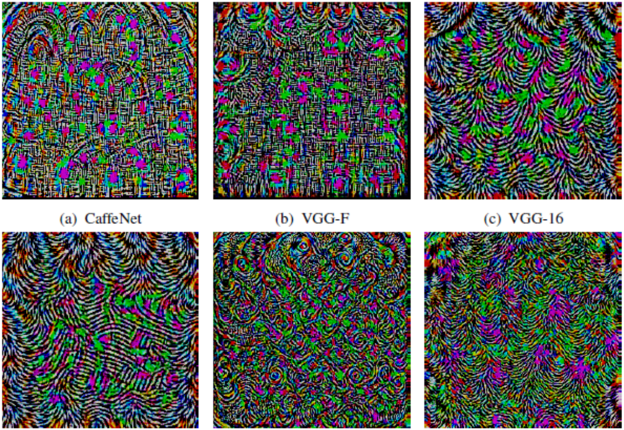
Introduction

- Universal adversarial perturbations are quasi-imperceptible perturbations designed to fool deep neural networks.
- These perturbations are image-agnostic, meaning a single perturbation can cause misclassification across a wide range of images.

Perturbed Images

Figure 3: Examples of perturbed images and their corresponding labels. The first 8 images belong to the ILSVRC 2012 validation set, and the last 4 are images taken by a mobile phone camera. See supp. material for the original images.

Perturbations



(a) CaffeNet (b) VGG-F (c) VGG-16

(d) VGG-19 (e) GoogLeNet (f) ResNet-152

Figure 4: Universal perturbations computed for different deep neural network architectures. Images generated with $p = \infty$ $\xi = 10$. The pixel values are scaled for visibility.

Perturbation Algorithm

Perturbation Vector v

- Given:
  - μ: Distribution of images in $\mathbb{R}^d$.
  - $\hat{k}$: Classification function providing labels for each $x \in \mathbb{R}^d$.

Perturbation Vector v

- Given:
  - μ: Distribution of images in $\mathbb{R}^d$.
  - $\hat{k}$: Classification function providing labels for each $x \in \mathbb{R}^d$.
- Seek vector v such that:

$$\hat{k}(x + v) \neq \hat{k}(x) \quad \text{for most } x \sim \mu$$

Perturbation Vector v

- Given:
  - μ: Distribution of images in $\mathbb{R}^d$.
  - $\hat{k}$: Classification function providing labels for each $x \in \mathbb{R}^d$.
- Seek vector v such that:

$$\hat{k}(x+v) \neq \hat{k}(x) \quad \text{for most } x \sim \mu$$

- Formally, find v that satisfies the constraints

$$\|v\|_p \leq \xi, \quad P_{x\sim\mu}\left(\hat{k}(x+v) \neq \hat{k}(x)\right) \geq 1-\delta$$

Perturbation Vector v

- Given:
  - μ: Distribution of images in $\mathbb{R}^d$.
  - $\hat{k}$: Classification function providing labels for each $x \in \mathbb{R}^d$.
- Seek vector v such that:

$$\hat{k}(x+v) \neq \hat{k}(x) \quad \text{for most } x \sim \mu$$

- Formally, find v that satisfies the constraints

$$\|v\|_p \leq \xi, \quad P_{x\sim\mu}\left(\hat{k}(x+v) \neq \hat{k}(x)\right) \geq 1-\delta$$

- Parameters:
  - ξ: Magnitude of perturbation v.
  - δ: Fooling rate threshold.

Perturbation Algorithm - Optimization

- If current universal perturbation v does not fool data point x_i, seek the extra perturbation Δv_i

$$\Delta v_i \leftarrow \arg\min_r \|r\|_2 \quad \text{s.t.} \quad \hat{k}(x_i + v + r) \neq \hat{k}(x_i)$$

Perturbation Algorithm - Optimization

- If current universal perturbation v does not fool data point x_i, seek the extra perturbation Δv_i

$$\Delta v_i \leftarrow \arg\min_r \|r\|_2 \quad \text{s.t. } \hat{k}(x_i + v + r) \neq \hat{k}(x_i)$$

- Update rule:

$$v \leftarrow P_{p,\xi}(v + \Delta v_i)$$

Perturbation Algorithm - Optimization

- If current universal perturbation v does not fool data point x_i, seek the extra perturbation Δv_i

$$\Delta v_i \leftarrow \arg\min_r \|r\|_2 \quad \text{s.t. } \hat{k}(x_i + v + r) \neq \hat{k}(x_i)$$

- Update rule:

$$v \leftarrow P_{p,\xi}(v + \Delta v_i)$$

- Projection operator $P_{p,\xi}$:

$$P_{p,\xi}(v) = \arg\min_{v'} \|v - v'\|_2 \quad \text{s.t. } \|v'\|_p \leq \xi$$

Algorithm End

- Algorithm iterates until:

$$\mathrm{Err}(X_v) = \frac{1}{m}\sum_{i=1}^{m} 1_{\hat{k}(x_i+v)\neq\hat{k}(x_i)} \geq 1-\delta$$

Algorithm End

- Algorithm iterates until:

$$\text{Err}(X_v) = \frac{1}{m}\sum_{i=1}^{m} 1_{\hat{k}(x_i+v)\neq\hat{k}(x_i)} \geq 1-\delta$$

- The number of datapoints m in X need not be large to compute a perturbation that is valid for the whole distribution!

Results

Generalization across Data Points

		CaffeNet [8]	VGG-F [2]	VGG-16 [17]	VGG-19 [17]	GoogLeNet [18]	ResNet-152 [6]
ℓ_2	X	85.4%	85.9%	90.7%	86.9%	82.9%	89.7%
	Val.	85.6	87.0%	90.3%	84.5%	82.0%	88.5%
ℓ_∞	X	93.1%	93.8%	78.5%	77.8%	80.8%	85.4%
	Val.	93.3%	93.7%	78.3%	77.8%	78.9%	84.0%

Table 1: Fooling ratios on the set X, and the validation set.

- High fooling rates on set X as well as validation set (not used for computing v).

Generalization across Architectures

	VGG-F	CaffeNet	GoogLeNet	VGG-16	VGG-19	ResNet-152
VGG-F	**93.7%**	71.8%	48.4%	42.1%	42.1%	47.4 %
CaffeNet	74.0%	**93.3%**	47.7%	39.9%	39.9%	48.0%
GoogLeNet	46.2%	43.8%	**78.9%**	39.2%	39.8%	45.5%
VGG-16	63.4%	55.8%	56.5%	**78.3%**	73.1%	63.4%
VGG-19	64.0%	57.2%	53.6%	73.5%	**77.8%**	58.0%
ResNet-152	46.3%	46.3%	50.5%	47.0%	45.5%	**84.0%**

Table 2: Generalizability of the universal perturbations across different networks. The percentages indicate the fooling rates. The rows indicate the architecture for which the universal perturbations is computed, and the columns indicate the architecture for which the fooling rate is reported.

- Cross-model universality: Perturbations generalize well across different architectures!

Explaining the Vulnerability to Universal Perturbations

Comparing Perturbations

- Random perturbations.

Comparing Perturbations

- Random perturbations.
- Adversarial perturbations computed for a randomly picked sample

Comparing Perturbations

- Random perturbations.
- Adversarial perturbations computed for a randomly picked sample
- Sum of adversarial perturbations over X.

Comparing Perturbations

- Random perturbations.
- Adversarial perturbations computed for a randomly picked sample
- Sum of adversarial perturbations over X.
- Mean of the images (ImageNet bias).

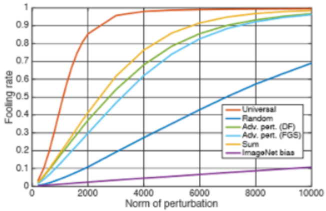


Figure 8: Comparison between fooling rates of different perturbations. Experiments performed on the CaffeNet architecture.

- Suggests that decision boundaries of deep networks exhibit geometric correlations.

Matrix N

- For each image x in the validation set, compute the adversarial perturbation vector:

$$r(x) = \arg\min_{r} \|r\|_2 \quad \text{s.t. } \hat{k}(x+r) \neq \hat{k}(x)$$

Matrix N

- For each image x in the validation set, compute the adversarial perturbation vector:

$$r(x) = \arg\min_r \|r\|_2 \quad \text{s.t. } \hat{k}(x+r) \neq \hat{k}(x)$$

- $r(x)$ is normal to the decision boundary of the classifier at $x + r(x)$.

Matrix N

- For each image x in the validation set, compute the adversarial perturbation vector:

$$r(x) = \arg\min_r \|r\|_2 \quad \text{s.t. } \hat{k}(x+r) \neq \hat{k}(x)$$

- $r(x)$ is normal to the decision boundary of the classifier at $x + r(x)$.
- Define the matrix N, containing normalized vectors $r(x_i)$, as:

$$N = \left[\frac{r(x_1)}{\|r(x_1)\|_2} \quad \frac{r(x_2)}{\|r(x_2)\|_2} \quad \ldots \quad \frac{r(x_n)}{\|r(x_n)\|_2} \right]$$

Matrix N

- For each image x in the validation set, compute the adversarial perturbation vector:

$$r(x) = \arg\min_r \|r\|_2 \quad \text{s.t. } \hat{k}(x + r) \neq \hat{k}(x)$$

- $r(x)$ is normal to the decision boundary of the classifier at $x + r(x)$.
- Define the matrix N, containing normalized vectors $r(x_i)$, as:

$$N = \begin{bmatrix} \frac{r(x_1)}{\|r(x_1)\|_2} & \frac{r(x_2)}{\|r(x_2)\|_2} & \dots & \frac{r(x_n)}{\|r(x_n)\|_2} \end{bmatrix}$$

- Compute the singular value decomposition (SVD) of N

$$N = U\Sigma V^T$$

Singular Values

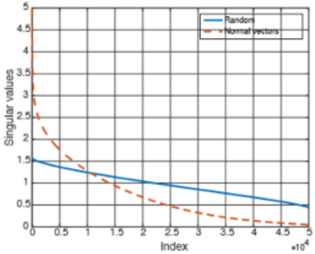


Figure 9: Singular values of matrix N containing normal vectors to the decision decision boundary.

- Singular values of N decay quickly

Singular Values

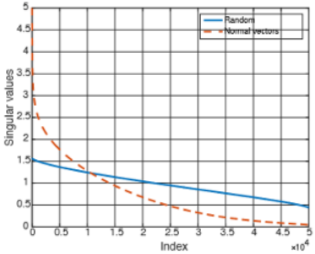


Figure 9: Singular values of matrix N containing normal vectors to the decision decision boundary.

- Singular values of N decay quickly
- Suggests that a low-dimensional subspace captures most normal vectors to decision boundaries.

Low Dimensional Subspace

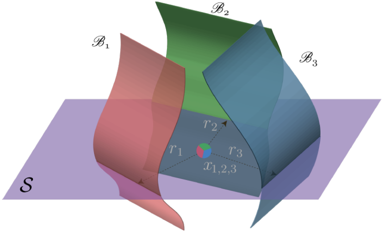


Figure 10: Illustration of the low dimensional subspace $\mathcal{S}$ containing normal vectors to the decision boundary in regions surrounding natural images. For the purpose of this illustration, we super-impose three data-points $\{x_i\}_{i=1}^3$, and the adversarial perturbations $\{r_i\}_{i=1}^3$ that send the respective datapoints to the decision boundary $\{\mathscr{B}_i\}_{i=1}^3$ are shown. Note that $\{r_i\}_{i=1}^3$ all live in the subspace $\mathcal{S}$.

Conclusion & Discussion

Conclusion

Universal adversarial perturbations

- Exist for deep neural networks.

Conclusion

Universal adversarial perturbations

- Exist for deep neural networks.
- Can generalize across images and different architectures.

Conclusion

Universal adversarial perturbations

- Exist for deep neural networks.
- Can generalize across images and different architectures.
- Decision boundaries show geometric correlations, allowing perturbations to exploit redundancies.

Future work

- A deeper theoretical analysis of the geometric properties of decision boundaries is needed to better understand these vulnerabilities.

Why?

How come adversarial perturbations exist for neural networks, and not for humans?

- Do humans have better 'training data'?

Why?

How come adversarial perturbations exist for neural networks, and not for humans?

- Do humans have better 'training data'?
- Is the human brain just bigger in scale?

Why?

How come adversarial perturbations exist for neural networks, and not for humans?

- Do humans have better 'training data'?
- Is the human brain just bigger in scale?
- Or is the way humans perceive images fundamentally different?

Bluesky

Bluesky

- There are many interesting people on Bluesky!
- Explore starter packs across Machine Learning, AI, and Economics
  - Machine Learning - Theory
  - ML & Probabilistic Stuff
  - Economists Working on AI
  - 'ML/AI People'
  - List of over 50 Econ Starter-Packs by Field