Visualization and Adversarial Examples

Jyoti Aneja, Ralf Gunter Correa Carvalho, Jiahui Yu

CS-598LAZ
Today’s Talk

Visualization

1. What is Visualization?
2. Visualize patches that maximally activate neurons
3. Visualize the weights
4. Gradient based approaches
5. Optimization based approach

Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack
Outline - Visualization

• What is Visualization?
• Visualize patches that maximally activate neurons
• Visualize the weights
• Gradient based approaches
• Optimization based approach
What is visualization?

Mapping between a neuron in a layer to the features in the image.
Background Check!

Input Image
(227 x 227 x 3)

Filters/Weights/Kernels
(eg: 96, 11 x 11 x 3)

Feature Maps
(227 x 227 x 96)

Neuron (Each small square)

Activations

Max Pool Layer
What is visualization?

Mapping between a neuron in a layer to the features in the original image.

Backpropagation: How does the loss change with weights?

Visualization: How does the activation of a particular neuron change when we change a part in the image?
Why visualization?

• Understand how and why neural networks work
• Observe the evolution of features during training
• Aid the development of better models (rather than just trial-and-error)
• Diagnose potential problems with the model
Outline - Visualization

- What is Visualization?
- Visualize patches that maximally activate neurons
- Visualize the weights
- Gradient based approaches
- Optimization based approach
Visualize patches that maximally activate neurons
Visualize patches that maximally activate neurons
Visualize patches that maximally activate neurons
Outline - Visualization

- What is Visualization?
- Visualize patches that maximally activate neurons
- Visualize the weights
- Gradient based approaches
- Optimization based approach
Visualize the weights
Visualize the weights

Only possible for the first layer 😞
Outline - Visualization

- What is Visualization?
- Visualize patches that maximally activate neurons
- Visualize the weights
- Gradient based approaches
- Optimization based approach
Gradient based approaches

Q: How can we compute the gradient of an arbitrary neuron w.r.t the image?
Gradient based approaches

1. Input the image into the net
2. Pick a layer, set the gradient there to be all 0 except for one 1 for some neuron of interest
3. “Map it” back to the image
Gradient based approaches - “Map back”

Gradient based approaches - “Map back”

Gradient based approaches - “Map back”

a) Input image $f^0$ → Forward pass $f^1$, $f^2$, ..., $f^{l-1}$ → Feature map $f^l$

Reconstructed image $R^l$ → Backward pass $R^1$, $R^2$, ..., $R^{l-1}$ → Backward pass: backpropagation

Activation: $f_i^{l+1} = \text{relu}(f_i^l) = \max(f_i^l, 0)$

Backpropagation: $R_i^l = (f_i^l > 0) \cdot R_i^{l+1}$, where $R_i^{l+1} = \frac{\partial f_{\text{out}}}{\partial f_i^{l+1}}$

b) Forward pass

Forward pass

Backward pass

Backward pass: backpropagation

Gradient based approaches - “Map back”

Visualizing the neurons along the way to the top

1. Choose a target neuron
2. Input the images 1 by 1
3. Select the top 9 images that have the highest activation for that neuron
4. Cluster those images together
5. Map back from that neuron and create a “back-pass map”
Visualizing the neurons along the way to the top
Matthew D. Zeiler, Rob Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014
What features are being captured from these pictures?
Outline - Visualization

• What is Visualization?
• Visualize patches that maximally activate neurons
• Visualize the weights
• Gradient based approaches
• Optimization based approach
Optimization Approach

Can we find an image that increases some class score?

\[ \arg \max_I S_c(I) - \lambda \|I\|_2^2 \]

Score for class c before softmax  

Regularization term
Optimization Approach - Algorithm

Start with zero image
Repeat:
  Feed image forward
  Set the gradient of the scores’ vector to be [0,0,...1,...,0]
  Backward pass the gradients to the image
  Update image (add regularization to avoid large updates)

$$\arg \max_{I} S_c(I) - \lambda \|I\|_2^2$$
Optimization Approach - Examples

- dumbbell
- cup
- dalmatian
- bell pepper
- lemon
- husky

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps
Karen Simonyan et al. 2014
Optimization Approach - Examples

washing machine  computer keyboard  kit fox

goose  ostrich  limousine
Visualizing Intermediate Layers

Smaller receptive field

Layer 4

Layer 2

Understanding Neural Networks Through Deep Visualization, Yosinski et al. - 2015
Visualizing Intermediate Layers

Layer 8

- Pirate Ship
- Rocking Chair
- Teddy Bear
- Windsor Tie
- Pitcher

Large receptive field

Layer 7

Understanding Neural Networks Through Deep Visualization, Yosinski et al. - 2015]
What if we map back the gradients onto the original image?
What if we map back the gradients onto the original image?
What if we map back the gradients onto the original image?

"Admiral Dog!"  "The Pig-Snail"  "The Camel-Bird"  "The Dog-Fish"
What if we map back the gradients onto the original image?

Deep Dream Grocery Store

"Admiral Dog!"  "The Pig-Snail"  "The Camel-Bird"  "The Dog-Fish"
Q: What is the difference between the gradient approach and the optimization approach for visualization?
Adversarial Examples

\[ K(X + v) \neq K(X), \]
where \( K \) is a classifier, \( X \) is input image, \( v \) is perturbation.

Intriguing properties of neural networks, Szegedy et al. - 2013
Why care about adversarial examples?
Why care about adversarial examples?
Why care about adversarial examples?

"Build safe, widely distributed AI."
-- OpenAI

Biometrics

Autonomous Driving

Speech Recognition

Security Guard Robot
Outline – Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack
Outline – Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack
Adversarial and Rubbish examples

**Adversarial**
- corrupt an existing natural image

**Rubbish**
- noisy meaningless pictures that achieve high confidence classification

Intriguing properties of neural networks, Szegedy et al. - 2013
Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images - Nguyen, et al - 2014
Outline – Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack
Evolutionary Approach

1. State-of-the-art DNNs can recognize real images with high confidence

2. But DNNs are also easily fooled: images can be produced that are unrecognizable to humans, but DNNs believe with 99.99% certainty are natural objects

Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images - Nguyen, et al - 2014
Rubbish examples by evolutionary approach

Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images - Nguyen, et al - 2014
Q: How can we change the image to fool the classifier?
Outline – Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack
Gradient-based approaches for visualization

\[
\arg \max \limits_I S_c(I) - \lambda \|I\|_2^2
\]

Input image

Score of class c given input image

Regularization term

Deep Inside Convolutional Networks Visualising Image Classification Models and Saliency Maps – Simonyan et al - 2013
Gradient-based approaches for visualization

**Adversarial examples**

**Visualization:**

$$\arg \max_I S_c(I) - \lambda \|I\|_2^2$$

**Adversarial examples:**

1. Let $S_c(I)$ have high score for input $I$
2. We maximize the $-S_c(I + \text{noise})$ w.r.t noise
3. and penalize the L2-norm of noise.
4. We get a new image $X = (I + \text{noise})$

Deep Inside Convolutional Networks Visualising Image Classification Models and Saliency Maps – Simonyan et al - 2013
Fast Gradient Sign Method

Score of label $y_{true}$, given input image $X$

$$X^{adv} = X + \epsilon \text{sign} \left( \nabla_X J(X, y_{true}) \right)$$

Adversarial examples in the physical world - Kurakin, et al - 2016
Explaining and Harnessing Adversarial Examples - Goodfellow, et al - 2014
Fast Gradient Sign Method

\[ X^{adv} = X + \varepsilon \text{sign}(\nabla_X J(X, y_{true})) \]

“gibbon”

“panda”

adversarial perturbation

Adversarial examples in the physical world - Kurakin, et al - 2016
Explaining and Harnessing Adversarial Examples - Goodfellow, et al - 2014
Fast Gradient Sign Method

\[ \mathbf{X}^{adv} = \mathbf{X} + \epsilon \text{sign}(\nabla_{\mathbf{X}} J(\mathbf{X}, y_{true})) \]

“gibbon”

“panda”

adversarial perturbation

Adversarial examples in the physical world - Kurakin, et al - 2016
Explaining and Harnessing Adversarial Examples - Goodfellow, et al - 2014
Gradients-based Methods

- Fast Gradient Sign Method:

\[
X^{adv} = X + \epsilon \text{sign}(\nabla_X J(X, y_{true}))
\]

- Iterative Gradient Sign Method

\[
X_0^{adv} = X, \quad X_{N+1}^{adv} = \text{Clip}_{X, \epsilon} \left\{ X_N^{adv} + \alpha \text{sign}(\nabla_X J(X_N^{adv}, y_{true})) \right\}
\]

Iteratively repeat

Adversarial examples in the physical world - Kurakin, et al - 2016
Gradients-based Methods

- **Fast Gradient Sign Method:**

\[ X_{adv} = X + \epsilon \text{sign}(\nabla_X J(X, y_{true})) \]

- **Iterative Gradient Sign Method**

\[ X_0^{adv} = X, \quad X_{N+1}^{adv} = \text{Clip}_{X,\epsilon}\left\{ X_N^{adv} + \alpha \text{sign}(\nabla_X J(X_N^{adv}, y_{true})) \right\} \]

- **Iterative Least-likely Class Method**

\[ y_{LL} = \arg \min_y \{ p(y|X) \} \]

\[ X_0^{adv} = X, \quad X_{N+1}^{adv} = \text{Clip}_{X,\epsilon}\left\{ X_N^{adv} - \alpha \text{sign}(\nabla_X J(X_N^{adv}, y_{LL})) \right\} \]

Adversarial examples in the physical world - Kurakin, et al - 2016
Visual Comparison of Gradients-based Methods

Adversarial examples in the physical world - Kurakin, et al - 2016
Outline – Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack
Adversarial Training

Q: How can we use adversarial examples to train a robust network?

A: Train it both on natural images and constructed adversarial images.

\[
\tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x + \epsilon \text{sign} (\nabla_x J(\theta, x, y)))
\]

Training Target \hspace{2cm} Adversarial regularizer

Adversarial examples in the physical world - Kurakin, et al - 2016
Adversarial Training

How can we use adversarial examples to train a robust network?

\[ \tilde{J}(\theta, x, y) = \alpha J(\theta, x, y) + (1 - \alpha) J(\theta, x + \epsilon \text{sign} (\nabla_x J(\theta, x, y))) \]

Training Target \hspace{2cm} Adversarial regularizer

For natural images, error rate drops from 0.94% to 0.84% on mnist.
For adversarial images, error rate drops from 89.4% to 17.9% on mnist.

Adversarial examples in the physical world - Kurakin, et al - 2016
Outline – Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack
How much information do we need to fool a neural net?

<table>
<thead>
<tr>
<th>Model weights</th>
<th>Have full access to model weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>Know what the model looks like</td>
</tr>
<tr>
<td>Training data</td>
<td>Know what training dataset was used</td>
</tr>
<tr>
<td>Oracle/black box</td>
<td>Query model with input $X$, get label $Y$</td>
</tr>
</tbody>
</table>
Black box example – what we hear

https://www.youtube.com/watch?v=vM5C4nHUQDs
Black box example – what we hear

Buy me a diamond ring

To order it, tell me your voice code

https://www.youtube.com/watch?v=vM5C4nHUQDs
Transferability scenarios

Cross training-set generalization

• Same architecture, different training set

Cross model generalization

• Different architecture, same training set

https://www.cs.toronto.edu/~frossard/post/vgg16/
http://johnloeber.com/docs/kmeans.html
Generalization error rates

<table>
<thead>
<tr>
<th></th>
<th>Model 1 - Dataset 1</th>
<th>Model 2 - Dataset 1</th>
<th>Model 1 - Dataset 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1 D1</td>
<td>100%</td>
<td>26.2%</td>
<td>5.9%</td>
</tr>
<tr>
<td>M2 D1</td>
<td>6.25%</td>
<td>100%</td>
<td>5.1%</td>
</tr>
<tr>
<td>M1 D2</td>
<td>8.2%</td>
<td>8.2%</td>
<td>100%</td>
</tr>
<tr>
<td>Gaussian noise</td>
<td>2.2%</td>
<td>2.6%</td>
<td>2.4%</td>
</tr>
</tbody>
</table>

Table 1: Fooling rate for average perturbation stddev = 0.06
Generalization error rates

<table>
<thead>
<tr>
<th></th>
<th>Model 1 - Dataset 1</th>
<th>Model 2 - Dataset 1</th>
<th>Model 1 - Dataset 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1 D1</td>
<td>100%</td>
<td>98%</td>
<td>43%</td>
</tr>
<tr>
<td>M2 D1</td>
<td>96%</td>
<td>100%</td>
<td>22%</td>
</tr>
<tr>
<td>M1 D2</td>
<td>27%</td>
<td>50%</td>
<td>100%</td>
</tr>
<tr>
<td>Gaussian noise</td>
<td>2.6%</td>
<td>2.8%</td>
<td>2.7%</td>
</tr>
</tbody>
</table>

Table 2: Fooling rate for average perturbation stddev = 0.1

Intriguing properties of neural networks - Szegedy et al - 2013
This is a very inefficient process
This is a very inefficient process

Q: what is the missing transferability property?
Outline – Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack
Universal Adversarial Perturbations

Universal Adversarial Perturbations – Moosavi-Dezfooli et al - 2016
Candidate universal perturbations

- Random noise
  - Easy to compute
  - Needs high norm to be effective
  - Obvious to human

- Sum of all adversarial perturbations over $X$
  - Less obvious
  - Components known to be effective
  - Very expensive (compute $|X|$ times)

- Universal Adversarial Perturbations (new method)
  - Adaptively expensive (compute for a subset of $X$)
  - Very subtle

Universal Adversarial Perturbations – Moosavi-Dezfooli et al - 2016
Algorithm

**Intuition:**
1. Start with \( v = 0 \)
2. If \((X_i + v)\) is misclassified, skip to \(X_{i+1}\)
3. Find minimum perturbation \(\Delta v\) that takes \(X_i + v + \Delta v\) to another class
4. Update \(v = v + \Delta v\)
5. Repeat with \(X_{i+1}\)

---

```python
Algorithm 1 Computation of universal perturbations.

1: input: Data points \(X\), classifier \(\hat{k}\), desired \(\ell_p\) norm of the perturbation \(\xi\), desired accuracy on perturbed samples \(\delta\).
2: output: Universal perturbation vector \(v\).
3: Initialize \(v \leftarrow 0\).
4: while \(\text{Err}(X_v) \leq 1 - \delta\) do
5:     for each datapoint \(x_i \in X\) do
6:         if \(\hat{k}(x_i + v) = \hat{k}(x_i)\) then
7:             Compute the minimal perturbation that sends \(x_i + v\) to the decision boundary:
8:                 \[ \Delta v_i \leftarrow \arg\min_{r} ||r||_2 \]
9:                 s.t. \(\hat{k}(x_i + v + r) \neq \hat{k}(x_i)\).
10:         Update the perturbation:
11:             \(v \leftarrow P_{\ell_p}(v + \Delta v_i)\).
12:     end if
13: end for
14: end while
```

Universal Adversarial Perturbations – Moosavi-Dezfooli et al - 2016
Sample universal perturbations
Cross-model universality

<table>
<thead>
<tr>
<th>Model</th>
<th>VGG-F</th>
<th>CaffeNet</th>
<th>GoogLeNet</th>
<th>VGG-16</th>
<th>VGG-19</th>
<th>ResNet-152</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-F</td>
<td>93.7%</td>
<td>71.8%</td>
<td>48.4%</td>
<td>42.1%</td>
<td>42.1%</td>
<td>47.4%</td>
</tr>
<tr>
<td>CaffeNet</td>
<td>74.0%</td>
<td>93.3%</td>
<td>47.7%</td>
<td>39.9%</td>
<td>39.9%</td>
<td>48.0%</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>46.2%</td>
<td>43.8%</td>
<td>78.9%</td>
<td>39.2%</td>
<td>39.8%</td>
<td>45.5%</td>
</tr>
<tr>
<td>VGG-16</td>
<td>63.4%</td>
<td>55.8%</td>
<td>56.5%</td>
<td>78.3%</td>
<td>73.1%</td>
<td>63.4%</td>
</tr>
<tr>
<td>VGG-19</td>
<td>64.0%</td>
<td>57.2%</td>
<td>53.6%</td>
<td>73.5%</td>
<td>77.8%</td>
<td>58.0%</td>
</tr>
<tr>
<td>ResNet-152</td>
<td>46.3%</td>
<td>46.3%</td>
<td>50.5%</td>
<td>47.0%</td>
<td>45.5%</td>
<td>84.0%</td>
</tr>
</tbody>
</table>

Fooling rate when computing a perturbation for one model (rows) and testing it on others (columns)
Outline – Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack
Models are too linear

\[ x = \begin{bmatrix} 2 & -1 & 3 & -2 & 2 & 2 & 1 & -4 & 5 & 1 \end{bmatrix} \]

\[ w = \begin{bmatrix} -1 & -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 \end{bmatrix} \]

class 1 score = dot product:
\[ = -2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3 \]
Models are too linear

$$\text{class 1 score} = \text{dot product:}$$
$$= -2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3$$
Models are too linear

<table>
<thead>
<tr>
<th>X</th>
<th>2</th>
<th>-1</th>
<th>3</th>
<th>-2</th>
<th>2</th>
<th>2</th>
<th>1</th>
<th>-4</th>
<th>5</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>adversarial X</td>
<td>1.5</td>
<td>-1.5</td>
<td>3.5</td>
<td>-2.5</td>
<td>2.5</td>
<td>1.5</td>
<td>1.5</td>
<td>-3.5</td>
<td>4.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>

class 1 score before:
-2 + 1 + 3 + 2 + 2 - 2 + 1 - 4 - 5 + 1 = -3

-1.5 + 1.5 + 3.5 + 2.5 + 2.5 - 1.5 + 1.5 - 3.5 - 4.5 + 1.5 = 2
Outline – Adversarial Examples

1. Adversarial and Rubbish examples
2. Evolutionary approach
3. Gradient based approaches
4. Adversarial training
5. Transferability
6. Universal Adversarial Perturbations
7. Why are neural networks easily fooled?
8. Proposed Solutions for adversarial attack
Proposed solution: highly non-linear models

- Use a rectified polynomial as the activation

\[ F_n(x) = \begin{cases} 
  x^n, & x \geq 0 \\
  0, & x < 0 
\end{cases} \]
Robustness against Adversarial Examples

Dense Associative Memory is Robust to Adversarial Inputs - Dmitri Kotrov, John J Hopfield - 2017
Fooling Rate

<table>
<thead>
<tr>
<th>generate</th>
<th>n=2</th>
<th>n=3</th>
<th>n=20</th>
<th>n=30</th>
</tr>
</thead>
<tbody>
<tr>
<td>n=2</td>
<td>98.9%</td>
<td>50.7%</td>
<td>9.07%</td>
<td>3.44%</td>
</tr>
<tr>
<td>n=3</td>
<td>33.9%</td>
<td>99%</td>
<td>8.71%</td>
<td>3.32%</td>
</tr>
<tr>
<td>n=20</td>
<td>45.3%</td>
<td>63.7%</td>
<td>98.9%</td>
<td>5.77%</td>
</tr>
<tr>
<td>n=30</td>
<td>37.6%</td>
<td>48.3%</td>
<td>56.9%</td>
<td>98.8%</td>
</tr>
</tbody>
</table>

clean MNIST test set:

- $\text{error}_{n=2} = 1.51\%$
- $\text{error}_{n=3} = 1.44\%$
- $\text{error}_{n=20} = 1.61\%$
- $\text{error}_{n=30} = 1.80\%$

Dense Associative Memory is Robust to Adversarial Inputs - Dmitri Kotrov, John J Hopfield - 2017
Adversarial Examples

- Adversarial and Rubbish examples
- Evolutionary approach
- Gradient based approaches
- Adversarial training
- Transferability
- Universal Adversarial Perturbations
- Why are neural networks easily fooled?
- Proposed Solutions for adversarial attack

Visualization

- What is Visualization?
- Visualize patches that maximally activate neurons
- Visualize the weights
- Gradient based approaches
- Optimization based approach
Reading list

• Matthew D. Zeiler, Rob Fergus *Visualizing and Understanding Convolutional Networks*, ECCV 2014
• Anh Nguyen, Jason Yosinski, Jeff Clune. *Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images*, CVPR 2015
• Dmitry Krotov, et al, *Dense Associative Memory is Robust to Adversarial Inputs*, arXiv preprint arXiv:1701.00939
• Ian J. Goodfellow et al, *Attacking machine learning with adversarial examples*, OpenAI blog post
Conclusion

Visualization

Adversarial Examples

Future of DL/AI