High-frequency Component Helps Explain the Generalization of Convolutional Neural Networks

Haohan Wang, Xindi Wu, Zeyi Huang, Eric P. Xing
School of Computer Science
Carnegie Mellon University
Outline

- Motivation, Introduction
- HFC, LFC, definition
- Heuristics (Dropout, mix-up, BatchNorm, Adversarial Training)
- Are HFC just noises?
- A hypothesis on Batch Normalization
- examples CIFAR, ImageNet, Detection
- Discussion
(a) A sample of frog
High/Low-Frequency Component (HFC, LFC)

Fourier Transform

$z = \mathcal{F}(x)$

Filter radius

$LFC$

$z_l, z_h = t(z; r)$

$x_h = \mathcal{F}^{-1}(z_h)$

$HFC$

$x_l = \mathcal{F}^{-1}(z_l)$
Figure 2. Eight testing samples selected from CIFAR10 that help explain that CNN can capture the high-frequency image: the model (ResNet18) correctly predicts the original image (1\textsuperscript{st} column in each panel) and the high-frequency reconstructed image (3\textsuperscript{rd} column in each panel), but incorrectly predict the low-frequency reconstructed image (2\textsuperscript{nd} column in each panel). The prediction confidences are also shown. The frequency components are split with $r = 12$. Details of the experiment will be introduced later.
What is it about

- Accuracy vs. Robustness (adversarial examples)
- Generalization vs. Overfitting
- Shuffled label memorization paradox (overfitting)

Figure 3. Training curves of the original label case (100 epoches) and shuffled label case (300 epoches), together plotted with the low-frequent counterpart of the images. All curves in this figure are from train samples.
Generalization power of LFC and HFC

- CIFAR 10, ResNet-18, ADAM, LR=1e-4, batch size=100, epochs=100

Table 1. We test the generalization power of LFC and HFC by training the model with $x_l$ or $x_h$ and test on the original test set.

<table>
<thead>
<tr>
<th></th>
<th>LFC</th>
<th></th>
<th>HFC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$</td>
<td>train acc.</td>
<td>test acc.</td>
<td>train acc.</td>
<td>test acc.</td>
</tr>
<tr>
<td>4</td>
<td>0.9668</td>
<td>0.6167</td>
<td>4</td>
<td>0.9885</td>
</tr>
<tr>
<td>8</td>
<td>0.9786</td>
<td>0.7154</td>
<td>8</td>
<td>0.9768</td>
</tr>
<tr>
<td>12</td>
<td>0.9786</td>
<td>0.7516</td>
<td>12</td>
<td>0.9797</td>
</tr>
<tr>
<td>16</td>
<td>0.9839</td>
<td>0.7714</td>
<td>16</td>
<td>0.9384</td>
</tr>
</tbody>
</table>
High/Low-Frequency Component (HFC, LFC)

An image is decomposed \[ \mathbf{x} = \{ \mathbf{x}_l, \mathbf{x}_h \} \]

\[ \mathbf{z} = \mathcal{F}(\mathbf{x}), \quad \mathbf{z}_l, \mathbf{z}_h = t(\mathbf{z}; r), \]

\[ \mathbf{x}_l = \mathcal{F}^{-1}(\mathbf{z}_l), \quad \mathbf{x}_h = \mathcal{F}^{-1}(\mathbf{z}_h), \]

\[ \mathbf{z}_l(i, j) = \begin{cases} \mathbf{z}(i, j), & \text{if } d((i, j), (c_i, c_j)) \leq r, \\ 0, & \text{otherwise} \end{cases} \]

\[ \mathbf{z}_h(i, j) = \begin{cases} 0, & \text{if } d((i, j), (c_i, c_j)) \leq r \\ \mathbf{z}(i, j), & \text{otherwise} \end{cases} \]

- this process is channel-wise
- imaginary part of \( \mathcal{F}^{-1}(.) \) is discarded.
Are HFC just Noise?

- An Experiment:
  - “LFC and HFC separation” is done using SVD decomposition
  - The first image is reconstructed based on dominant singular values
  - The second is reconstructed with trailing singular values
  - With this set-up, much fewer images supporting the story in Figure 2. (where CNN is classifying HFC image correctly) was observed.

![Reconstructed Images](image1)

Figure 2. Eight testing samples selected from CIFAR10 that help explain that CNN can capture the high-frequency image; the model (ResNet18) correctly predicts the original image (1\textsuperscript{st} column in each panel) and the high-frequency reconstructed image (3\textsuperscript{rd} column in each panel), but incorrectly predict the low-frequency reconstructed image (2\textsuperscript{nd} column in each panel). The prediction confidences are also shown. The frequency components are split with $r = 12$. Details of the experiment will be introduced later.
Vanilla

Dropout

Mix-up

\[
\hat{x} = \lambda x_i + (1 - \lambda) x_j, \\
\hat{y} = \lambda y_i + (1 - \lambda) y_j,
\]

where \( \lambda \in [0, 1] \) is a random number.

Image

Label

[1.0, 0.0]

[0.0, 1.0]

[0.7, 0.3]

image credit: https://www.kaggle.com/kaushal2896/data-augmentation-tutorial-basic-cutout-mixup
Figure 6. Comparison of models with vs. without BatchNorm trained with LFC data.
Bonus: JPEG compression

- CIFAR 100, ResNet-18
- We investigated effect of JPEG compression on the inference accuracy.
- We tried to use JPEG compression as an augmentation

- Same = NN was trained and evaluated with the same compression
- RandomTrain = random compression was picked during training
- 60Train = image quality was fixed during training to 60
- NoCompressionTrain = the data for training was taken as provided in CIFAR-100

Credits: Jan Pokorný, Milan Šulc, Lukáš Picek
comparison of various methods

Credits: Jan Pokorný, Milan Šulc, Lukáš Picek
Robust Models Have Smooth Kernels

Figure 7. Visualization of convolutional kernels (16 kernels each channel × 3 channels at the first layer) of models.
Robust Models Have Smooth Kernels

<table>
<thead>
<tr>
<th></th>
<th>Clean</th>
<th>FGSM $\epsilon = 0.03$</th>
<th>FGSM $\epsilon = 0.06$</th>
<th>FGSM $\epsilon = 0.09$</th>
<th>PGD $\epsilon = 0.03$</th>
<th>PGD $\epsilon = 0.06$</th>
<th>PGD $\epsilon = 0.09$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_{\text{natural}}$</td>
<td>0.856</td>
<td>0.107</td>
<td>0.069</td>
<td>0.044</td>
<td>0.003</td>
<td>0.002</td>
<td>0.002</td>
</tr>
<tr>
<td>$M_{\text{natural}}(\rho = 0.10)$</td>
<td>0.815</td>
<td>0.149</td>
<td>0.105</td>
<td>0.073</td>
<td>0.009</td>
<td>0.002</td>
<td>0.001</td>
</tr>
<tr>
<td>$M_{\text{natural}}(\rho = 0.25)$</td>
<td>0.743</td>
<td>0.16</td>
<td>0.11</td>
<td>0.079</td>
<td>0.021</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>$M_{\text{natural}}(\rho = 0.50)$</td>
<td>0.674</td>
<td>0.17</td>
<td>0.11</td>
<td>0.083</td>
<td>0.031</td>
<td>0.016</td>
<td>0.014</td>
</tr>
<tr>
<td>$M_{\text{natural}}(\rho = 1.0)$</td>
<td>0.631</td>
<td>0.171</td>
<td>0.14</td>
<td>0.127</td>
<td>0.086</td>
<td>0.078</td>
<td>0.078</td>
</tr>
<tr>
<td>$M_{\text{adversarial}}$</td>
<td>0.707</td>
<td>0.435</td>
<td>0.232</td>
<td>0.137</td>
<td>0.403</td>
<td>0.138</td>
<td>0.038</td>
</tr>
<tr>
<td>$M_{\text{adversarial}}(\rho = 0.10)$</td>
<td>0.691</td>
<td>0.412</td>
<td>0.192</td>
<td>0.109</td>
<td>0.379</td>
<td>0.13</td>
<td>0.047</td>
</tr>
<tr>
<td>$M_{\text{adversarial}}(\rho = 0.25)$</td>
<td>0.667</td>
<td>0.385</td>
<td>0.176</td>
<td>0.097</td>
<td>0.352</td>
<td>0.116</td>
<td>0.04</td>
</tr>
<tr>
<td>$M_{\text{adversarial}}(\rho = 0.50)$</td>
<td>0.653</td>
<td>0.365</td>
<td>0.18</td>
<td>0.106</td>
<td>0.334</td>
<td>0.121</td>
<td>0.062</td>
</tr>
<tr>
<td>$M_{\text{adversarial}}(\rho = 1.0)$</td>
<td>0.638</td>
<td>0.356</td>
<td>0.223</td>
<td>0.186</td>
<td>0.337</td>
<td>0.175</td>
<td>0.131</td>
</tr>
</tbody>
</table>

Table 2. Prediction performance of models against different adversarial attacks with different $\epsilon$. 
Beyond Image Classification

- RetinaNet ResNet50 + FPN on MS COCO -> mAP 35.6%
- Use r=128 and map imgs to LFC and HFC and test the same model
  - LFC 27.5% mAP
  - HFC 10.7% mAP
- 1684 where LFC actually performs better (Fig 9)
Figure 8. Some objects are recognized worse (lower MAP scores) when the experiments are repeated with low-frequent images. Marked objects are the ones that induce differences.

Figure 9. Some objects are recognized better (higher MAP scores) when the experiments are repeated with low-frequent images. Marked objects are the ones that induce differences.
Critique

- Hard-to-read plots.
- Not justified relation between human labeled data and LHF (e.g. what if the task is to detect cracks in a metal plate).
- Missing appendix even though it is mentioned multiple times.
- They do not mention how many times they run experiments.
- A table with values accompanying Figure 5 would be nice.
A1: “only $x_l$ is perceivable to human, but both $x_l$ and $x_h$ are perceivable to a CNN,” we have:

$$y := f(x; \mathcal{H}) = f(x_l; \mathcal{H}),$$

but when a CNN is trained with

$$\arg\min_{\theta} l(f(\theta), y),$$

which is equivalent to

$$\arg\min_{\theta} l(f(\{x_l, x_h\}; \theta), y),$$

CNN may learn to exploit $x_h$ to minimize the loss. As a result, CNN’s generalization behavior appears unintuitive to a human.
Trade-off between Robustness and Accuracy

the accuracy of $\theta$

$$\mathbb{E}_{(x,y)} \alpha(f(x; \theta), y) \quad (1)$$

and the adversarial robustness of $\theta$ as in e.g., [8]:

$$\mathbb{E}_{(x,y)} \min_{x' : d(x', x) \leq \epsilon} \alpha(f(x'; \theta), y) \quad (2)$$

where $\epsilon$ is the upper bound of the perturbation allowed.
Trade-off between Robustness and Accuracy

A2: “for model $\theta$, there exists a sample $\langle x, y \rangle$ such that:

$$f(x; \theta) \neq f(x_l; \theta),$$

**Corollary 1.** With assumptions A1 and A2, there exists a sample $\langle x, y \rangle$ that the model $\theta$ cannot predict both accurately (evaluated to be 1.0 by Equation 1) and robustly (evaluated to be 1.0 by Equation 2) under any distance metric $d(\cdot, \cdot)$ and bound $\epsilon$ as long as $\epsilon \geq d(x, x_l)$. 
Critique

If a model can exploit multiple different sets of signals, then why $M_{\text{natural}}$ prefers to learn LFC that happens to align well with the human perceptual preference? While there are explanations suggesting neural networks’ tendency towards simpler functions [48], we conjecture that this is simply because, since the data sets are organized and annotated by human, the LFC-label association is more “generalizable” than the one of HFC: picking up LFC-label association will lead to the steepest descent of the loss surface, especially at the early stage of the training.
Conclusion

- CNN may capture HFC that are misaligned with human visual preference.
- Heuristics like Mix-up and BatchNorm may encourage capturing HFC
- Bonus: If you can’t control input quality during inference, add JPEG compression as an augmentation.
Thanks Giorgos for comments and organization! Thank you for your attention!

Discussion