Differential Operators for Structured Adversarial Examples

Hossein Mobahi

Joint work with Calvin Luo Samy Bengio

Google Research

April 22, 2020
Data Augmentation

- Data augmentation is an important ingredient for achieving SOTA on image classification tasks.
- Currently incorporated into training by sampling; may not scale to larger space of transformations.

From [https://github.com/aleju/imgaug](https://github.com/aleju/imgaug)
Adversarial Examples

- **Original Image**
- **Adversarial Perturbation**

Very rich set of perturbations, that often fall outside of natural transformations.
Goal

Model space of transformations, then use adversarial perturbation

Original Image

Raw Adversarial Perturbation

Restricted Perturbation

Recolorized Image
Notation

• Given a loss function $L(w; z, l)$ that takes as its input the model parameters $w$, input image $z$, and label $l$.

• Suppose components of $z$ (i.e. pixels) are sampled from a continuous image flow $Z(x, y, t)$ at a specific $t$,

\[ z \triangleq [Z(x_1, y_1, t), \ldots, Z(x_n, y_m, t)]. \quad (1) \]

• Thus loss can be written as,

\[ L(Z(x_1, y_1, t), \ldots, Z(x_n, y_m, t)), \quad (2) \]

($w$ and $l$ dropped for brevity).
Seek a perturbation in the image that maximally increases the loss.

\[
\frac{\partial}{\partial t} L = \langle \nabla_z L(z), [\dot{Z}(x_1, y_1, t), \ldots, \dot{Z}(x_n, y_m, t)] \rangle. \tag{3}
\]

We will demonstrate a few examples to construct \( \dot{Z} \).
In a geometric transform, points may move, but cannot change color/brightness.

Image from Michael Black’s PhD thesis
Consider \textit{brightness constancy} assumption,

\[
\frac{d}{dt} Z(x(t), y(t), t) = 0 ,
\]  

which is equivalent to,

\[
Z_x(x(t), y(t), t) \dot{x}(t) \\
+ Z_y(x(t), y(t), t) \dot{y}(t) \\
+ \dot{Z}(x(t), y(t), t)
\]  

\[
= 0 .
\]

Applying this yields,

\[
\frac{\partial}{\partial t} L = -\langle \nabla_z L(z) , [ Z_x(x_1, y_1, t) \dot{x}(x_1, y_1, t) \\
+ Z_y(x_1, y_1, t) \dot{y}(x_1, y_1, t) \\
\dot{Z}(x_1, y_1, t) \\
\vdots \\
\vdots \\
Z_x(x_n, y_m, t) \dot{x}(x_n, y_m, t) \\
+ Z_y(x_n, y_m, t) \dot{y}(x_n, y_m, t) \rangle \rangle .
\]
We are interested in this dynamics at the reference point $t = 0$. For brevity, denote $Z_x(x_i, y_j, t = 0)$ by $Z_{x,i,j}$ and $\dot{x}((x_i, y_j), t = 0)$ by $\dot{x}_{i,j}$. Also denote each component of $\nabla_x L(x)$ by $\ell_{i,j}$. Thus in matrix form,

$$\frac{\partial}{\partial t} L = - \begin{pmatrix} d_x \\ d_y \end{pmatrix}^T \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \quad (10)$$

where,

$$N \triangleq n m \quad (11)$$

$$d_{x} N \times 1 \triangleq \begin{pmatrix} \ell_{1,1} Z_{x,1,1} & \ldots & \ell_{m,n} Z_{x,m,n} \end{pmatrix} \quad (12)$$

$$d_{y} N \times 1 \triangleq \begin{pmatrix} \ell_{1,1} Z_{y,1,1} & \ldots & \ell_{m,n} Z_{y,m,n} \end{pmatrix} \quad (13)$$

$$\dot{x} N \times 1 \triangleq \begin{pmatrix} \dot{x}_{1,1} & \ldots & \dot{x}_{m,n} \end{pmatrix} \quad (14)$$

$$\dot{y} N \times 1 \triangleq \begin{pmatrix} \dot{y}_{1,1} & \ldots & \dot{y}_{m,n} \end{pmatrix} \quad (15)$$
We seek motion field \((\dot{x} \text{ and } \dot{y})\) that maximizes \(\frac{\partial}{\partial t} L\). To keep velocity bounded, restrict \(\|\dot{x}\|\) and \(\|\dot{y}\|\). Also to have coherency in the field impose smoothness.

\[
(\dot{x}^*, \dot{y}^*) = \arg\min_{(\dot{x}, \dot{y})} \left( d_x \right)^T \left( \dot{x} \right) + \frac{1}{2} \alpha (\|\dot{x}\|^2 + \|\dot{y}\|^2)
\]

\[
+ \frac{1}{2} \beta (\|D_x \dot{x}\|^2 + \|D_y \dot{x}\|^2)
\]

\[
+ \frac{1}{2} \beta (\|D_x \dot{y}\|^2 + \|D_y \dot{y}\|^2),
\]

where \(D_x\) and \(D_y\) are differential operators along \(x\) and \(y\) direction, represented in matrix form. For brevity, let \(P \triangleq D_x^T D_x + D_y^T D_y\). We obtain,

\[
(\dot{x}^*, \dot{y}^*) = \arg\min_{(\dot{x}, \dot{y})} \left( d_x \right)^T \left( \dot{x} \right)
\]

\[
\left( d_y \right)^T \left( \dot{y} \right) + \frac{1}{2} \alpha (\dot{x}^T I \dot{x} + \dot{y}^T I \dot{y})
\]

\[
+ \frac{1}{2} \beta (\dot{x}^T P \dot{x} + \dot{y}^T P \dot{y}).
\]

Hossein Mobahi
Differential Operators for Structured Adversarial Examples
Letting $\gamma \triangleq \frac{\beta}{\alpha}$, we can express

$$(\dot{x}^*, \dot{y}^*) = \arg\min_{(\dot{x}, \dot{y})} \begin{pmatrix} d_x \\ d_y \end{pmatrix}^T \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}$$

$$+ \frac{1}{2} \alpha (\dot{x}^T \dot{x} + \dot{y}^T \dot{y})$$

$$+ \frac{1}{2} \beta (\dot{x}^T \dot{P_x} + \dot{y}^T \dot{P_y})$$

(19)

(20)

(21)

as,

$$(\dot{x}^*, \dot{y}^*) = \arg\min_{(\dot{x}, \dot{y})} \begin{pmatrix} d_x \\ d_y \end{pmatrix}^T \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}$$

$$+ \frac{1}{2} \alpha ((\dot{x}^T \dot{x} + \dot{y}^T \dot{y})$$

$$+ \gamma (\dot{x}^T \dot{P_x} + \dot{y}^T \dot{P_y})$$

(22)

(23)

(24)

This is a convex objective function in $(\dot{x}, \dot{y})$ and its minimizer can be obtained by zero crossing its gradient,

$$d_x + \alpha (I + \gamma P) \dot{x} = 0$$

(25)

$$d_y + \alpha (I + \gamma P) \dot{y} = 0$$

(26)
Therefore,

\[
\dot{x} = -\frac{1}{\alpha} (I + \gamma P)^{-1} d_x 
\] (27)

\[
\dot{y} = -\frac{1}{\alpha} (I + \gamma P)^{-1} d_y .
\] (28)

Note that the matrix \((I + \gamma P)^{-1}\) solely depends on regularizer, and does not depend on data (data being \(d_x\) and \(d_y\)). Thus, one can compute \((I + \gamma P)^{-1}\) offline given image dimensions and trade off parameter \(\gamma\).

Once the motion field \((\dot{x}, \dot{y})\) is computed, it can be used to warp \(I\) to a new form via the motion field.
Examples of ResNet trained on ImageNet:

- Garbage truck. Confidence 60.95%
- Minibus. Confidence 33.90%
- Redbone. Confidence of 44.95
- Boxer. Confidence 19.34%

The flow warped the back of the truck to be raised - a trait common in minibuses. Similarly, for the dog image, it warps the white cloth onto the dog’s leg (redbones are normally entirely red, boxers are commonly red dogs that feature patches of white fur).
Photometric

Goal is to recolorize the image; preserve contours, object boundaries, edges, etc., but okay to change colors in the image. Consider some notion of edge strength, such as,

\[
E(x, y, t) \triangleq \sum_{i=A, B, C} \left[ (Z_{Ax}(x, y, t))^2 + (Z_{Ay}(x, y, t))^2 + (Z_{Bx}(x, y, t))^2 + (Z_{By}(x, y, t))^2 + (Z_{Cx}(x, y, t))^2 + (Z_{Cy}(x, y, t))^2 \right].
\]  

(29)

(30)

(31)

We penalize changes in edge strength anywhere that has already weak edges.

\[
P(t) \triangleq \int_X \int_Y \left( \frac{1}{E(x, y, 0)} \frac{\partial}{\partial t} E(x, y, t) \right)^2 dx dy
\]  

(32)

\[
\int_X \int_Y \left( \frac{1}{E(x, y, 0)} \right)^2 \left( Z_{Ax}(x, y, t) \dot{Z}_{Ax} + Z_{Ay}(x, y, t) \dot{Z}_{Ay} + Z_{Bx}(x, y, t) \dot{Z}_{Bx} + Z_{By}(x, y, t) \dot{Z}_{By} + Z_{Cx}(x, y, t) \dot{Z}_{Cx} + Z_{Cy}(x, y, t) \dot{Z}_{Cy} \right)^2 dx dy.
\]  

(33)

(34)

(35)

(36)
Photometric

We are interested in $P(t)$ at $t = 0$ and thus drop dependency of the on $t$ from the notation,

$$P = \int_X \int_Y \left( \frac{1}{E(x, y)} \right) \left( Z_{A,x}(x, y) Z_{A,x} + Z_{A,y}(x, y) Z_{A,y} \right) \right) \right)^2 \, dx \, dy . \quad (37)$$

$$+ Z_{B,x}(x, y) Z_{B,x} + Z_{B,y}(x, y) Z_{B,y} \right) \right) \right)^2 \, dx \, dy . \quad (38)$$

$$+ Z_{C,x}(x, y) Z_{C,x} + Z_{C,y}(x, y) Z_{C,y} \right) \right) \right)^2 \, dx \, dy . \quad (39)$$

This integral can be approximated numerically, e.g. sampling over \textit{evenly spaced} grid of steps $\delta_x$ and $\delta_y$,

$$P \approx \hat{P} \triangleq \delta_x \delta_y \| M_A \dot{z}_A + M_B \dot{z}_B + M_C \dot{z}_C \|^2 , \quad (41)$$

where,

$$M_{\alpha N \times N} \triangleq \text{diag}(z_{\alpha x} \div e) D_x + \text{diag}(z_{\alpha y} \div e) D_y$$

$$e_{N \times 1} \triangleq \left\| \begin{bmatrix} z_{A,x} \mid z_{A,y} \mid z_{B,x} \mid z_{B,y} \mid z_{C,x} \mid z_{C,y} \end{bmatrix} \right\|_{N \times 6} ,$$

and $\div$ denotes element-wise division.
Photometric

Let \( z \triangleq [z^T_A \mid z^T_B \mid z^T_C]^T \). The goal is to find \( \dot{z} \) that maximally increases the loss under this penalty.

\[
\dot{z}_{3N \times 1}^* \triangleq \arg \min_{\dot{z}} -\langle \nabla_z L(z), \dot{z} \rangle + \frac{1}{2} \lambda \hat{P}.
\] (42)

This is a convex objective function in \( \dot{z} \) and its solution can be obtained by zero crossing the gradient,

\[
-\nabla z_A L(z) + \lambda M^T_A(M_A \dot{z}_A + M_B \dot{z}_B + M_C \dot{z}_C) = 0
\] (43)

\[
-\nabla z_B L(z) + \lambda M^T_B(M_A \dot{z}_A + M_B \dot{z}_B + M_C \dot{z}_C) = 0
\] (44)

\[
-\nabla z_C L(z) + \lambda M^T_C(M_A \dot{z}_A + M_B \dot{z}_B + M_C \dot{z}_C) = 0.
\] (45)

This can be written more compactly as,

\[
M_{3N \times 3N} \dot{z}_{3N \times 1} = \frac{1}{\lambda} \nabla z L(z),\]
(46)

where,

\[
M_{3N \times 3N} \triangleq \begin{pmatrix}
M^T_A M_A & M^T_A M_B & M^T_A M_C \\
M^T_B M_A & M^T_B M_B & M^T_B M_C \\
M^T_C M_A & M^T_C M_B & M^T_C M_C
\end{pmatrix}.
\] (47)
Thus,

\[
\hat{z}^{3N \times 1} = \frac{1}{\lambda} M^{-1}_{3N \times 3N} (\nabla_z L(z))_{3N \times 1},
\]  

(48)

While the matrix $M$ depends on the data (images), it has nothing to do with the loss function $L$ (thus independent of learning model). Thus for each image $z$, the associated matrix $M^{-1}$ can be computed once. Then the computed $M^{-1}$ can the be used for any learning model in the future. Clearly $M^{-1}$ captures some intrinsic properties of the data that are useful later for photometric transforms.
• There are still challenges using this in practice: although $M^{-1}$ can be computed once, applying such giant matrix to the gradient vector can be expensive.
• We use SVD to extract leading eigenvectors of $M^{-1}$ and only use those to achieve an efficient approximation.
Photometric Examples

Original  Recolorized  Big Eigenvector.
### Experiments

<table>
<thead>
<tr>
<th>Augmentation</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
<th>STL-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>BasicAug</td>
<td>95.56%</td>
<td>77.41%</td>
<td>84.60%</td>
</tr>
<tr>
<td>AutoAug</td>
<td>95.70%</td>
<td>77.41%</td>
<td>84.46%</td>
</tr>
<tr>
<td>RandAug</td>
<td>94.95%</td>
<td>74.56%</td>
<td>88.75%</td>
</tr>
<tr>
<td>Flow + BasicAug</td>
<td>95.79%</td>
<td>76.82%</td>
<td>85.41%</td>
</tr>
<tr>
<td>Flow + AutoAug</td>
<td>95.77%</td>
<td>77.26%</td>
<td>85.39%</td>
</tr>
<tr>
<td>Flow + RandAug</td>
<td>93.22%</td>
<td>70.14%</td>
<td>87.96%</td>
</tr>
<tr>
<td>Recolor + BasicAug</td>
<td>95.80%</td>
<td><strong>77.45%</strong></td>
<td>86.11%</td>
</tr>
<tr>
<td>Recolor + AutoAug</td>
<td><strong>95.86%</strong></td>
<td>76.97%</td>
<td>85.21%</td>
</tr>
<tr>
<td>Recolor + RandAug</td>
<td>94.78%</td>
<td>74.18%</td>
<td><strong>88.84%</strong></td>
</tr>
</tbody>
</table>

**Table:** Reported accuracy of a Wide-ResNet-32-10 model trained from scratch using the listed augmentations.
## Experiments

<table>
<thead>
<tr>
<th>Method</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
<th>STL-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Augmentation</td>
<td>97.52%</td>
<td>85.68%</td>
<td>97.55%</td>
</tr>
<tr>
<td>AutoAugment</td>
<td>97.44%</td>
<td>84.09%</td>
<td>97.32%</td>
</tr>
<tr>
<td>RandAugment</td>
<td>96.39%</td>
<td>80.38%</td>
<td>97.14%</td>
</tr>
<tr>
<td>Flow</td>
<td>97.60%</td>
<td>85.87%</td>
<td>97.61%</td>
</tr>
<tr>
<td>Recolorization</td>
<td>97.34%</td>
<td>85.25%</td>
<td>97.61%</td>
</tr>
</tbody>
</table>

Table: Reported accuracy after finetuning on a EfficientNet-B0 checkpoint for 350 epochs.
Directions for Future Work

Most important directions for future Work

• Additional transformations that can be modeled in similar way.
• Learning the linear operators from another training set.
Thank You!