The Local Elasticity of Neural Networks

Weijie Su

University of Pennsylvania

Joint work with Hangfeng He (Penn CS)
Some are similar, some are different

- $D(\text{tabby cat, tiger cat}) = 853.6$
- $D(\text{tabby cat, fighter aircraft}) = 510.5$
Do neural nets know similarities?
Influence of one image on another

How does the SGD update at tabby cat impact on the prediction at tiger cat or fighter aircraft?
Influence of one image on another

How does the SGD update at *tabby cat* impact on the prediction at *tiger cat* or *fighter aircraft*?

- Will learning French affect my English?
Influence of one image on another

How does the SGD update at *tabby cat* impact on the prediction at *tiger cat* or *fighter aircraft*?

- Will learning French affect my English? I would say YES
Influence of one image on another

How does the SGD update at *tabby cat* impact on the prediction at *tiger cat* or *fighter aircraft*?

- Will learning French affect my English? I would say *YES*
- Will learning French affect my Chinese?
Influence of one image on another

How does the SGD update at *tabby cat* impact on the prediction at *tiger cat* or *fighter aircraft*?

- Will learning French affect my English? I would say YES
- Will learning French affect my Chinese? I would say NO
Large relative change
Small relative change

Weijie Su on Local Elasticity
Which case for neural nets?
Hypothesis of *local elasticity* in neural networks

- **Locality**: relative change is large when \( x \) and \( x' \) are close/similar. Akin to nearest neighbors algorithm
- **Elasticity**: relative change decreases *gradually* and *smoothly* (as opposed to abruptly) when \( x' \) moves away from \( x \). Adaptivity to data structure?
Definition of relative change

Let $f(x, w)$ denote neural networks, with weights $w$, and loss $L(f, y)$ SGD update at $w$ using the example $(x, y)$:

$$w^+ = w - \gamma \frac{dL(f(x, w), y)}{dw} = w - \gamma \left( \frac{\partial L(f(x, w), y)}{\partial f} \cdot \frac{\partial f(x, w)}{\partial w} \right)$$

Relative Change

$$S_{\text{rel}}(x, x') := \frac{|f(x', w^+) - f(x', w)|}{|f(x, w^+) - f(x, w)|}$$

- Near optimal $w$
- $S_{\text{rel}}(\text{tabby cat, tiger cat}) \gg S_{\text{rel}}(\text{tabby cat, fighter aircraft})$
Local elasticity shown in experiments

Torus

Two-layer nets fitting the torus

Two-layer linear nets fitting the torus

Two folded boxes

Three-layer nets fitting the boxes

Three-layer linear nets fitting the boxes

Weijie Su on Local Elasticity
Connection with neural tangent kernel (Jacot et al ’18, Arora et al ’19)

\[ f(x', w^+) - f(x', w) = f\left(x', w - \gamma \frac{\partial L}{\partial f} \cdot \frac{\partial f(x, w)}{\partial w}\right) - f(x', w) \approx f(x', w) - \left\langle \frac{\partial f(x', w)}{\partial w}, \gamma \frac{\partial L}{\partial f} \cdot \frac{\partial f(x, w)}{\partial w} \right\rangle - f(x', w) \]

\[ = -\gamma \frac{\partial L}{\partial f} \left\langle \frac{\partial f(x', w)}{\partial w}, \frac{\partial f(x, w)}{\partial w} \right\rangle \]
Connection with neural tangent kernel (Jacot et al ’18, Arora et al ’19)

\[
f(x', w^+) - f(x', w) = f\left(x', w - \gamma \frac{\partial L}{\partial f} \cdot \frac{\partial f(x, w)}{\partial w}\right) - f(x', w)
\]

\[
\approx f(x', w) - \left\langle \frac{\partial f(x', w)}{\partial w}, \gamma \frac{\partial L}{\partial f} \cdot \frac{\partial f(x, w)}{\partial w} \right\rangle - f(x', w)
\]

\[
= -\gamma \frac{\partial L}{\partial f} \left\langle \frac{\partial f(x', w)}{\partial w}, \frac{\partial f(x, w)}{\partial w} \right\rangle
\]

**Kernelized Change**

\[
S_{\text{ker}}(x, x') := \frac{f(x', w) - f(x', w^+)}{\gamma \frac{\partial L(f(x, w), y)}{\partial f}} \approx \left\langle \frac{\partial f(x', w)}{\partial w}, \frac{\partial f(x, w)}{\partial w} \right\rangle
\]
The NTK interpretation

- Improvement at tabby cat leads to improvement at tiger cat
- Improvement at tabby cat does not affect fighter plane
Clustering via local elasticity

- primary dataset $\mathcal{P} = \{x_i\}_{i=1}^n$
- auxiliary dataset $\mathcal{A} = \{\tilde{x}_j\}_{j=1}^m$
- classifier $f(x, w)$, loss function $\mathcal{L}$
- initial weights $w_0$, learning rate $\eta_t$, option $o \in \{\text{relative, kernelized}\}$
The algorithm

Algorithm 1 The Local Elasticity Based Clustering Algorithm

combine $\mathcal{D} = \{(x_i, y_i = 1) \text{ for } x_i \in \mathcal{P}\} \cup \{(x_i, y_i = -1) \text{ for } x_i \in \mathcal{A}\}$
set $S$ to $n \times n$ matrix of all zeros

for $t = 1$ to $n + m$ do
    sample $(x, y)$ from $\mathcal{D}$ w/o replacement
    $w_t = \text{SGD}(w_{t-1}, x, y, f, \mathcal{L}, \eta_t)$
    if $y = 1$ then
        $p_t = \text{Predict}(w_t, \mathcal{P}, f)$
        find $1 \leq i \leq n$ such that $x = x_i \in \mathcal{P}$
        if $o = \text{relative}$ then
            $s_t = \frac{|p_t - p_{t-1}|}{|p_t(i) - p_{t-1}(i)|}$
        else
            $g_t = \text{GetGradient}(w_{t-1}, x, y, f, \mathcal{L})$
            $s_t = \frac{p_t - p_{t-1}}{-\eta_t \times g_t}$
        end if
    end if
    set the $i$th row $S(i, :) = s_t$
end for

Weijie Su on Local Elasticity
Experiments on MNIST

<table>
<thead>
<tr>
<th>Primary Examples</th>
<th>5 vs 8</th>
<th>4 vs 9</th>
<th>7 vs 9</th>
<th>5 vs 9</th>
<th>3 vs 5</th>
<th>3 vs 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auxiliary Examples</td>
<td>3, 9</td>
<td>5, 7</td>
<td>4, 5</td>
<td>4, 8</td>
<td>8, 9</td>
<td>5</td>
</tr>
<tr>
<td>$\ell_2$-relative (FNN)</td>
<td><strong>75.9</strong></td>
<td>55.6</td>
<td>62.5</td>
<td>89.3</td>
<td>50.3</td>
<td>74.7</td>
</tr>
<tr>
<td>$\ell_2$-kernelized (FNN)</td>
<td>71.0</td>
<td>63.8</td>
<td>64.6</td>
<td>67.8</td>
<td>71.5</td>
<td>78.8</td>
</tr>
<tr>
<td>$\ell_2$-relative (CNN)</td>
<td>54.2</td>
<td>53.7</td>
<td>89.1</td>
<td>50.1</td>
<td>50.1</td>
<td>83.0</td>
</tr>
<tr>
<td>$\ell_2$-kernelized (CNN)</td>
<td>64.1</td>
<td><strong>69.5</strong></td>
<td><strong>91.3</strong></td>
<td><strong>97.6</strong></td>
<td><strong>75.3</strong></td>
<td><strong>87.4</strong></td>
</tr>
<tr>
<td>$\ell_2$-relative (ResNet)</td>
<td>50.7</td>
<td>55.0</td>
<td>55.5</td>
<td>78.3</td>
<td>52.3</td>
<td>52.3</td>
</tr>
<tr>
<td>$\ell_2$-kernelized (ResNet)</td>
<td>50.2</td>
<td>60.4</td>
<td>54.8</td>
<td>76.3</td>
<td>66.9</td>
<td>68.8</td>
</tr>
</tbody>
</table>

- Corroborates the hypothesis of local elasticity
What else is locally elastic?
What else is locally elastic?
What else is locally elastic?
What else is locally elastic?
What else is locally elastic?
What else is locally elastic?
What else is locally elastic?
What else is locally elastic?
What else is locally elastic?
What else is locally elastic?
Take-home messages

Neural networks are *locally elastic* classifiers!
Take-home messages

Neural networks are *locally elastic* classifiers!

- Learning at tabby cat improves understanding of another cat, but not plane
Take-home messages

Neural networks are *locally elastic* classifiers!

- Learning at tabby cat improves understanding of another cat, but not plane
- Need a firm mathematical foundation (low-dimensional topology, dynamical systems, differential geometry?)
Neural networks are *locally elastic* classifiers!

- Learning at tabby cat improves understanding of another cat, but not plane
- Need a firm mathematical foundation (low-dimensional topology, dynamical systems, differential geometry?)

**Implications**

- Memorization
- Stability and generalization
- Data normalization and batch normalization