I - Introduction

- **Goal**: define and estimate the similarity of inputs, as perceived by the neural network
- **Motivation**: strong auto-denoising phenomenon in a multimodal image registration task
  
  Red: initial dataset annotations
  Blue: aligned annotations round 1
  Green: aligned annotations round 2

  ➞ accuracy far better than label noise!
  ➞ analyze noise averaging effect over labels of similar examples
- **Motivation 2**: better understand neural network decisions
  ➞ display examples considered as similar by the network

II - Building the similarity measure

- **Notations**: \( f_\theta \): trained neural network with parameters \( \theta \); \( x, x' \): possible inputs
- **Similarity def**: influence of \( x \) over \( x' \) i.e. quantify how much an additional training step for \( x \) would change the output for \( x' \) as well.
  1. \( x \) and \( x' \) very different: changing \( f_\theta(x) \) will barely affect \( f_\theta(x') \)
  2. \( x \) and \( x' \) very similar: changing \( f_\theta(x) \) will greatly affect \( f_\theta(x') \)

- Changing \( f_\theta(x) \) by a small quantity \( \varepsilon \) means updating \( \theta \) by \( \delta \theta = \varepsilon \nabla f_\theta(x) \nabla^T f_\theta(x) \)
- After update, new values for \( x \) and \( x' \):
  \[
  f_{\theta+\delta\theta}(x) = f_\theta(x) + \nabla f_\theta(x) \cdot \delta \theta + O(\|\delta \theta\|^2) = f_\theta(x) + \varepsilon + O(\varepsilon^2)
  \]
  \[
  f_{\theta+\delta\theta}(x') = f_\theta(x') + \nabla f_\theta(x') \cdot \delta \theta + O(\|\delta \theta\|^2) = f_\theta(x') + \varepsilon \frac{\nabla f_\theta(x') \cdot \nabla f_\theta(x)}{\|\nabla f_\theta(x)\|^2} + O(\varepsilon^2)
  \]
- The kernel \( \kappa_\theta^0(x, x') = \frac{\nabla f_\theta(x) \cdot \nabla f_\theta(x')}{\|\nabla f_\theta(x)\|^2} \) represents the influence of \( x \) over \( x' \)
- Symmetric kernel bounded in \([-1, 1] \):
  \[
  \kappa_\theta^0(x, x') = \frac{\nabla f_\theta(x) \cdot \nabla f_\theta(x')}{\|\nabla f_\theta(x)\| \|\nabla f_\theta(x')\|}
  \]
- Higher output dimension: \( f_\theta(x) = (f_i(x))_{i \in |d|} \in \mathbb{R}^d \) with \( d > 1 \)
- \( K^0(x', x) \) the \( d \times d \) kernel matrix defined by \( K^0_{ij}(x', x) = \nabla f_j(x') \cdot \nabla f_i(x) \)
- Unitless symmetrized normalized kernel: \( K^0_\theta(x, x') = \frac{1}{d} \text{Tr} K^0_{\theta}(x, x') \)
- Simularity in a single value: \( \kappa_\theta^0(x, x') = \frac{1}{d} \text{Tr} K^0_{\theta}(x, x') \)

III - Estimating density

How many samples \( x' \) are similar to \( x \) according to the network? Many ways to count!

<table>
<thead>
<tr>
<th>hard-thresholding with threshold ( \tau \in [0, 1] )</th>
<th>soft estimate</th>
<th>less-soft positive-only estimate</th>
<th>( \alpha &gt; 0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sum_{x'} \mathbf{1}(\kappa_\theta^0(x, x') &gt; \tau) )</td>
<td>( \sum_{x'} \kappa_\theta^0(x, x') = \frac{\nabla f_\theta(x) \cdot \nabla f_\theta(x')}{|\nabla f_\theta(x)|^2} ) + ( \frac{\nabla f_\theta(x') \cdot \nabla f_\theta(x)}{|\nabla f_\theta(x')|^2} )</td>
<td>( \sum_{x'} 2 \kappa_\theta^0(x, x') &gt; \alpha )</td>
<td>( \sum_{x'} \mathbf{1}(\kappa_\theta^0(x, x') &gt; \tau) )</td>
</tr>
</tbody>
</table>

IV - Self-denoising experiment

Source | Closest neighbor matches

Similarity: Perceptual

- true (unknown) label: \( y_i \)
- (unknown) noise: \( y_i + \varepsilon_i \) (iid, centered)
- noisy (available) label: \( y_{\tilde{i}} = y_i + \varepsilon \)
- predicted label: \( \tilde{y}_i = f_\theta(x_i) \)
- training loss: \( L(\theta) = \sum_j \|y_j - \tilde{y}_j\|^2 \)
- at convergence \( \nabla_\theta E = 0 \Rightarrow E_L[\tilde{y}_{\tilde{i}}] = E_L[y_i] \)
- \( E_L[\tilde{y}_i] := \sum_j a_j y_{\tilde{i}}(x_j, x_i) \) : mean value around \( x_i \)
- \( \sum_j (\tilde{y}_i - E_L[\tilde{y}_{\tilde{i}}] = E_L[\varepsilon] + (\tilde{y}_i - E_L[\tilde{y}_{\tilde{i}}]) \)
- Denoising factor: \( \|K_\theta \|_2 \leq 0.02 \)
- Shift: \( (\tilde{y}_i - E_L[\tilde{y}_{\tilde{i}}]) : 4.4 \text{ px} \)

V - Discussion

- Opens the door to underfit/overfit/uncertainty analyses and control during training
- Similarity enforced during training: dataset-dependent boosting effect (see paper)
- Extended Noise2Noise to the case of non-identical inputs: expressing self-denoising effects as a function of inputs similarities
- Future work: analyse and improve robustness to adversarial attacks
- Code is available on GitHub: github.com/Lydorn/netsimilarity (or scan the QR code)