Neural networks with recurrent generative feedback


Yujia Huang, Caltech

yjhuang@caltech.edu
Intuition

Internal Model

Cat

Feedforward

Feedback
Given a joint distribution $p(h, y, z; \theta)$ parameterized by $\theta$, $(\hat{h}, \hat{y}, \hat{z})$ are self-consistent if they satisfy the following constraints:

\[
\begin{align*}
\hat{y} &= \arg \max_y p(y|\hat{h}, \hat{z}), \\
\hat{h} &= \arg \max_h p(h|\hat{y}, \hat{z}), \\
\hat{z} &= \arg \max_z p(z|\hat{h}, \hat{y})
\end{align*}
\]
Self-Consistency

- **x**: images
- **h**: encoded features
- **z**: latent variables
- **y**: labels

**External**

- $x \rightarrow h$

**Internal**

- $\hat{h} \rightarrow \hat{z} \rightarrow \hat{y}$

**Features**

- Color
- Size
- Lightning
- ...
Self-Consistency

\[ x \overset{+0.007 \times}{\Rightarrow} \text{sign}(\nabla_x J(\theta, x, y)) \overset{8.2\% \text{ confidence}}{\Rightarrow} \text{gibbon} \]

\[ x \overset{\text{h: encoded features}}{\Rightarrow} \]

\[ \hat{h} \overset{\text{z: latent variables}}{\Rightarrow} \hat{y} \]

x: images

h: encoded features

z: latent variables

y: labels
Generative Classifier

Deconvolutional generative model (DGM)

\[ y \sim p(y) \]
\[ z_P^{(i)} \sim \text{Ber}\left(\frac{e^{b \cdot g(i)}}{e^{b \cdot g(i)} + 1}\right) \]
\[ z_R^{(i)} \sim \text{Ber}\left(\frac{e^{b \cdot g(i)}}{e^{b \cdot g(i)} + 1}\right) \]
\[ x \sim \mathcal{N}(g(0), \text{diag}(\sigma^2)) \]

Inference in the DGM

- MAP estimate of $y$: $\hat{y} = \text{CNN}(h)$
- MAP estimate of $h$: $\hat{h} = g(0)$
- MAP estimate of $z$ (informal): $\hat{z}_R = \mathbb{1}\{\sigma_{\text{AdaReLU}}(f) \neq 0\}$
  $\hat{z}_P = \mathbb{1}\{\sigma_{\text{AdaPool}}(f) \neq 0\}$

$\sigma_{\text{AdaReLU}}(f) = \begin{cases} 
\sigma_{\text{ReLU}}(f), & \text{if } g \geq 0 \\
\sigma_{\text{ReLU}}(-f), & \text{if } g < 0 
\end{cases}$

$\sigma_{\text{AdaPool}}(f) = \begin{cases} 
\sigma_{\text{MaxPool}}(f), & \text{if } g \geq 0 \\
-\sigma_{\text{MaxPool}}(-f), & \text{if } g < 0 
\end{cases}$
Iterative inference

Self-consistency

\[ \hat{y} = \arg \max_y p(y|\hat{h}, \hat{z}) , \]

\[ \hat{h} = \arg \max_h p(h|\hat{y}, \hat{z}) , \]

\[ \hat{z} = \arg \max_z p(z|\hat{h}, \hat{y}) \]
Training of CNN-F

\[ \mathcal{L}_{\text{Xentropy}}(y_0, \text{target}) \mathcal{L}_{\text{Xentropy}}(y_1, \text{target}) \mathcal{L}_{\text{Xentropy}}(y_2, \text{target}) \]

- **Logits** \( \nu \)
- **Image features** \( h \)
- **Generated features** \( g(0) \)
- **Images** \( x \)
CNN-F with adversarial training

Reconstruction loss is always between **adversarial** and **natural** features.
CNN-F on all CNN architectures

IN: Instance Normalization

VGG/Allconv/…

ResNet
CNN-F repairs distorted images

Corrupted

Shot Noise

Ground-truth

Gaussian Noise

Dotted Line
CNN-F improves adversarial robustness

- Standard training on Fashion-MNIST.
- Attack with PGD-40.
- CNN-F has higher adversarial robustness than CNN.
CNN-F improves adversarial robustness

More iterations are needed for harder images.
Adversarial training on Fashion-MNIST.
- Trained with PGD-40 (eps=0.3). Attack with PGD-40.
- CNN-F augmented with adversarial images achieves high accuracy for both clean and adversarial data.
CNN-F generalizes better to different attacks

Feedback helps when there is distribution shift between training and testing data.

Trained with FGSM (eps=0.3).  
Attack with PGD-40.

Trained with PGD-40 (eps=0.3).  
Attack with PGD-40.
Train on CIFAR-10

- CNN-F (on Wide ResNet) combined with adversarial training.
- Clean accuracy decreases over iterations.
- Adversarial accuracy increases over iterations.
Neuronal predictivity

- Used the fifth block and logits in VGG-16 to predict V4 and IT neuronal activities.
- CNN-F predicts V4 and IT neuronal activities better than CNN.
- Call for temporal neuronal data in the neuroscience community.
Conclusions and future works

**Biological inspirations**
- Recurrent feedback
  - Generative models (Bayesian brain)
  - Attention
- Lateral connections
- Sparsity

**Inspirations from other fields**
- Signal processing (Kalman filters ...)
- Control (Feedback control, dynamical system)

**Down-streaming tasks**
- Robustness
- Few shot learning
- Uncertainty quantification
Thank You!