Measuring Invariances in Deep Networks

Goodfellow et al., 2009, NIPS

Presented by Ilsang Ohn

December 3, 2016

Invariance measure

- A hidden unit i, $h^{(i)}$, is said to **fire** for the given input x when

$$s_i h^{(i)}(x) > t_i$$

  where t_i is a threshold to be determined and $s_i \in \{-1, 1\}$ gives the sign of $h^{(i)}(x)$.
- The **global firing rate** is the firing rate of a hidden unit:

$$G(i) = \mathbb{P}_x(s_i h^{(i)}(x) > t_i)$$

  where $\mathbb{P}_x$ is a distribution over the possible inputs x.

Invariance measure

The **local firing rate** is the firing rate of hidden unit when it is applied to local trajectories surrounding inputs $z \in Z_i$:

$$L(i, T_{\tau,\Gamma}) = \frac{1}{|Z_i|} \sum_{z \in Z_i} \frac{1}{|T_{\tau,\Gamma}(z)|} \sum_{x \in T_{\tau,\Gamma}(z)} 1\{s_i h^{(i)}(x) > t_i\}.$$

where

- the set Z_i as a set of inputs that activate $h^{(i)}$ near maximally
- a **local trajectory** $T_{\tau,\Gamma}(x)$ is a set of stimuli that are semantically similar to some reference stimulus x, that is

$$T_{\tau,\Gamma}(x) = \{\tau(x,\gamma) | \gamma \in \Gamma\}$$

- a **transformation function** $\tau(x, \gamma)$ transforms x into a new, related input, where the degree of transformation is parametrized by $\gamma \in \mathbb{R}$
- Γ is a set of transformation amounts of limited size

Invariance measure

The **invariance score** for a hidden unit i and a local trajectory $T_{\tau,\Gamma}(x)$ is given by

$$S(i, T_{\tau,\Gamma}) = \frac{L(i, T_{\tau,\Gamma})}{G(i)}.$$

- The numerator is a measure of the hidden unit's **robustness** to transformation τ near the unit's optimal inputs
- The denominator ensure that the neurone is **selective** and not simply always active.
- In tests, select the threshold t_i so that $G(i) = 0.01$.

The invariance score $Inv_p(N)$ of a network N is given by the mean of $S(i, T_{\tau,\Gamma})$ over the top-scoring proportion p of hidden units in the network.

Experiment setup: Network architecture

1. Stacked Autoencoder (SAE)
   - Several single layer AEs are trauned with various target mean activations and amounts of weight decay.
   - Three-layer AEs to invastigate the effect of depth on invariance.
   - successively train up layers (by minimizing reconstruction error) of the network in a greed layer wise fashion.
2. Convolutional Deep Belief Network (CBDN)
   - trained using two hidden layers consisting of a collection of convolution units as well as a collection of max-pooling units
   - Because the convolution units share weights and their outputs are combined in the max-pooling units, the CDBN is explicitly designed to be invariant to small amounts of image translation.

Regulizaion for SAEs

Given a input $x \in \mathbb{R}^n$, the activtion of each neuran, $h^{(i)}$, $i = 1, \ldots, m$ is

$$h^{(i)}(x) = \sigma(W_1^{(i)} x + b_1^{(i)})$$

and the network output is

$$\hat{x}^{(i)} = \sigma(W_2^{(i)} h(x) + b_2^{(i)})$$

Given a set of inputs $x^{(i)}$, $i = 1, \ldots, N$, the parameters are trained by minimizing

$$\frac{1}{N} \sum_{i=1}^{N} \|x^{(i)} - \hat{x}^{(i)}\|^2 + \frac{\lambda}{2} \sum_{k=1}^{2} \sum_{i=1}^{N} \|W_k^{(i)}\|_F^2 + \beta \sum_{j=1}^{m} \mathrm{KL}(\rho \| \hat{\rho}_j)$$

where $\mathrm{KL}(\rho \| \hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$ and

$$\hat{\rho}_j = \frac{1}{N} \sum_{i=1}^{N} h^{(j)}(x^{(i)})$$

- ρ: target mean activation, sparsity parameter
- λ: weight decay

Experiment setup: Data

Grating test

- An input image I of a grating, with image pixel intensities given by

$$I(x_1, x_2) = b + a\sin(\omega(x_1 \cos\theta + x_2 \sin\theta - \phi)),$$

  where ω is the spatial frequency, θ is the orientation of the grating, and ϕ is the phase.
- $\mathbb{P}_x$ is defined as a uniform distribution over patches produced by varying $\omega \in \{2, 4, 6, 8\}$, $\theta \in \{0, \pi/20, \cdots, \pi\}$, and $\phi \in \{0, \pi/20, \cdots, \pi\}$.
- Local tragectories $T_{\tau,\Gamma}$ are generated by
  - (translation) $\tau(x, \gamma)$ changing ϕ_x to $\phi_x + \gamma$ with

$$\gamma = \left\{-\pi, -\frac{19}{20}\pi, \ldots, \frac{19}{20}\pi, \pi\right\}$$

  - (rotation) $\tau(x, \gamma)$ changing ω_x to $\omega_x + \gamma$ with

$$\gamma = \left\{-\pi, -\frac{39}{40}\pi, \ldots, \frac{39}{40}\pi, \pi\right\}$$

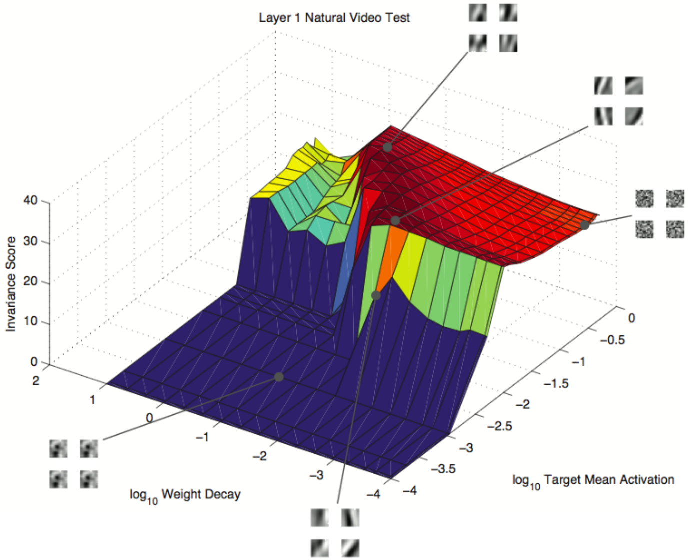
Experiment setup: Data

Natural video test

- consists of natural videos containing common image transformations such as translations, 2-D (in-plane) rotations, and 3-D (out-of-plane) rotations.
- $\mathbb{P}_x$ is defined as a uniform distribution over image patches contained in the test videos.
- $\tau(x, \gamma)$ is defined to be the image patch at the same image location as x but occurring γ video frames later in time.
- To measure invariance to different types of transformation, simply use videos that involve each type of transformation.

Result: SAE

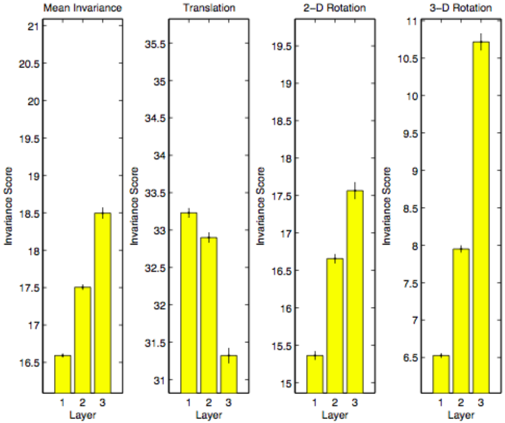
Pronounced effect of sparsity and weight decay



- $p = 1$
- A network with no regularization obtains a score of 35.88 and the best scoring netwotk receives a score of 32.41

Result: SAE

Modest improvements with depth



- $p = 0.2$, no sparsity regularization.
- The magnitude of the increase in invariance is limited compared to the increase that can be gained with the correct sparsity and weight decay.
- While depth is valuable, mere stacking of shallow architectures may not be sufficient to exploit the full potential of deep architectures to learn invariant features.

Result: CDBN

Test	Layer 1	Layer 2	% change
Grating phase	68.7	95.3	38.2
Grating orientation	52.3	77.8	48.7
Natural translation	15.2	23.0	51.0
Natural 3-D rotation	10.7	19.3	79.5

Table 1: Results of the CDBN invariance tests.

- CDBN does enjoy dramatically increasing invariance with depth.
- The single test with the greatest relative improvement is the 3-D (out-of-plane) rotation test. This is the most complex transformation included in our tests, and it is where depth provides the greatest percentagewise increase.