Neural networks

Autoencoder - contractive autoencoder
Topics: overcomplete representation

- Hidden layer is overcomplete if greater than the input layer
  - No compression in hidden layer
  - Each hidden unit could copy a different input component
- No guarantee that the hidden units will extract meaningful structure
Topics: contractive autoencoder

• Alternative approach to avoid uninteresting solutions
  ‣ add an explicit term in the loss that penalizes that solution

• We wish to extract features that only reflect variations observed in the training set
  ‣ we’d like to be invariant to the other variations
**CONTRACTIVE AUTOENCODER**

**Topics:** contractive autoencoder

• New loss function:

\[
\underbrace{l(f(x^{(t)}))}_\text{autoencoder reconstruction} + \lambda \underbrace{\|\nabla_x h(x^{(t)})\|_F^2}_\text{Jacobian of encoder}
\]

• where, for binary observations:

\[
l(f(x^{(t)})) = - \sum_k \left( x_k^{(t)} \log(h_k^{(t)}) + (1 - x_k^{(t)}) \log(1 - h_k^{(t)}) \right)
\]

\[
\|\nabla_{x^{(t)}} h(x^{(t)})\|_F^2 = \sum_j \sum_k \left( \frac{\partial h(x^{(t)})}{\partial x_k^{(t)}} \right)^2
\]
CONTRACTIVE AUTOENCODER

**Topics:** contractive autoencoder

- New loss function:

$$ l(f(x^{(t)})) + \lambda \| \nabla_{x^{(t)}} h(x^{(t)}) \|_F^2 $$

- where, for binary observations:

$$ l(f(x^{(t)})) = - \sum_k \left( x_k^{(t)} \log(\hat{x}_k^{(t)}) + (1 - x_k^{(t)}) \log(1 - \hat{x}_k^{(t)}) \right) $$

$$ \| \nabla_{x^{(t)}} h(x^{(t)}) \|_F^2 = \sum_j \sum_k \left( \frac{\partial h(x^{(t)})}{\partial x_k^{(t)}} \right)_j^2 $$

encoder keeps good information
CONTRACTIVE AUTOENCODER

**Topics:** contractive autoencoder

- New loss function:

\[
\ell(f(x^{(t)})) + \lambda \| \nabla_{x^{(t)}} h(x^{(t)}) \|^2_F
\]

\[
\text{autoencoder reconstruction} \quad \text{Jacobian of encoder}
\]

- where, for binary observations:

\[
\ell(f(x^{(t)})) = - \sum_k \left( x_k^{(t)} \log(\hat{x}_k^{(t)}) + (1 - x_k^{(t)}) \log(1 - \hat{x}_k^{(t)}) \right) \]

\[
\text{good information}
\]

\[
\| \nabla_{x^{(t)}} h(x^{(t)}) \|^2_F = \sum_j \sum_k \left( \frac{\partial h(x^{(t)})}{\partial x_k^{(t)}} \right)^2 \]

\[
\text{encoder keeps away all information}
\]
CONTRACTIVE AUTOENCODER

**Topics:** contractive autoencoder

- New loss function:

\[
\begin{align*}
    l(f(x^{(t)})) + \lambda \| \nabla_{x^{(t)}} h(x^{(t)}) \|_F^2
\end{align*}
\]

- autoencoder reconstruction
- Jacobian of encoder

- where, for binary observations:

\[
    l(f(x^{(t)})) = - \sum_k \left( x_k^{(t)} \log(h_k^{(t)}) + (1 - x_k^{(t)}) \log(1 - \hat{x}_k^{(t)}) \right)
\]

\[
    \| \nabla_{x^{(t)}} h(x^{(t)}) \|_F^2 = \sum_j \sum_k \left( \frac{\partial h(x^{(t)})}{\partial x_k^{(t)}} \right)^2
\]

- encoder keeps good information
- encoder throws away all information
- encoder keeps only good information
CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

• Illustration:
CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

• Illustration:
CONTRACTIVE AUTOENCODER

**Topics:** contractive autoencoder

- Illustration:
CONTRACTIVE AUTOENCODER

Topics: contractive autoencoder

• Illustration:

encoder must be sensitive to this variation to reconstruct well
**Topics:** contractive autoencoder

- Illustration:
  
  encoder must be sensitive to this variation (not observed in training set)

  encoder doesn’t need to be sensitive to this variation to reconstruct well
WHICH AUTOENCODER?

**Topics:** denoising autoencoder, contractive autoencoder

- Both the denoising and contractive autoencoder perform well
  - Advantage of denoising autoencoder: simpler to implement
    - requires adding one or two lines of code to regular autoencoder
    - no need to compute Jacobian of hidden layer
  - Advantage of contractive autoencoder: gradient is deterministic
    - can use second order optimizers (conjugate gradient, LBFGS, etc.)
    - might be more stable than denoising autoencoder, which uses a sampled gradient

- To learn more on contractive autoencoders:
  - Contractive Auto-Encoders: Explicit Invariance During Feature Extraction.
    Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot et Yoshua Bengio, 2011.