A Fast Learning Algorithm for Deep Belief Nets

Hinton, Osindero, Teh
Conditional Learning is Hard
Conditional Learning is Hard
Conditional Learning is Hard

-10

earthquake

+20

house jumps

-20

truck hits house

+20

-10
Conditional Learning is Hard

\[ P(y|x) = \prod_j P(y_j|x) \]
WHAT IF THERE WERE A PRIOR

WHERE $P(Y|X) = \prod(Y_i|X)$
Complementary Priors

\[ P(x|y) = \frac{1}{\Omega(y)} \exp \left( \sum_i \Phi_j(x, y_j) + \beta(x) \right) \]

\[ P(y) = \frac{1}{C} \exp \left( \log \Omega(y) + \sum_j \alpha_j(y_j) \right), \]

\[ P(x, y) = \frac{1}{C} \exp \left( \sum_j \Phi_j(x, y_j) + \beta(x) + \sum_j \alpha_j(y_j) \right). \]
A specially structured deep network
Training our deep network

\[
\frac{\partial \log p(v^0)}{\partial w_{ij}^{00}} = (h^0_j (v^0_i - \hat{v}^0_i)),
\]

\[
\frac{\partial \log p(v^0)}{\partial w_{ij}^{00}} = (h^0_j (v^0_i - v^1_i)).
\]

\[
\frac{\partial \log p(v^0)}{\partial w_{ij}} = (h^0_j (v^0_i - v^1_i)) + (v^1_i (h^0_j - h^1_j)) + (h^1_j (v^1_i - v^2_i)) + \cdots
\]
Training our deep network

\[
\frac{\partial \log p(v^0)}{\partial w_{ij}} = \langle h_j^0 (v_i^0 - v_i^1) \rangle + \langle v_i^1 (h_j^0 - h_j^1) \rangle + \langle h_j^1 (v_i^1 - v_i^2) \rangle + \cdots
\]

\[
\frac{\partial \log p(v^0)}{\partial w_{ij}} = \langle v_i^0 h_j^0 \rangle - \langle v_i^\infty h_j^\infty \rangle.
\]
Training our deep network

\[ \frac{\partial \log p(v^0)}{\partial w_{ij}} = \langle h^0_j (v^0_i - v^1_i) \rangle + \langle v^1_i (h^0_j - h^1_j) \rangle + \langle h^1_j (v^1_i - v^2_i) \rangle + \cdots \]

\[ \frac{\partial \log p(v^0)}{\partial w_{ij}} = \langle v^0_i h^0_j \rangle - \langle v^\infty_i h^\infty_j \rangle. \]

\[ KL(P^0 \| P^\infty_\theta) - KL(P^n_\theta \| P^\infty_\theta). \]

- This is the update for a restricted Boltzmann Machine
RBM training
Let’s relax the assumptions

WE NEED TO GO

DERPER
Greedy Training
Greedy Training
Greedy Training

\[ W_0, W_1, W_2, W_3 \]

Input
Fine tuning
Fine tuning

\[ W_0^T \uparrow \quad \downarrow W_0 \]
\[ W_1^T \uparrow \quad \downarrow W_1 \]
\[ W_2^T \uparrow \quad \downarrow W_2 \]
\[ W_3 \quad \uparrow \quad \downarrow \]

Input
Fine tuning

\[ W_0, W_1, W_2, W_3 \]

Input
Fine tuning
Yes, this actually works

THIS NEURAL NETWORK HAS BETTER HANDWRITING THAN I DO
All Done!