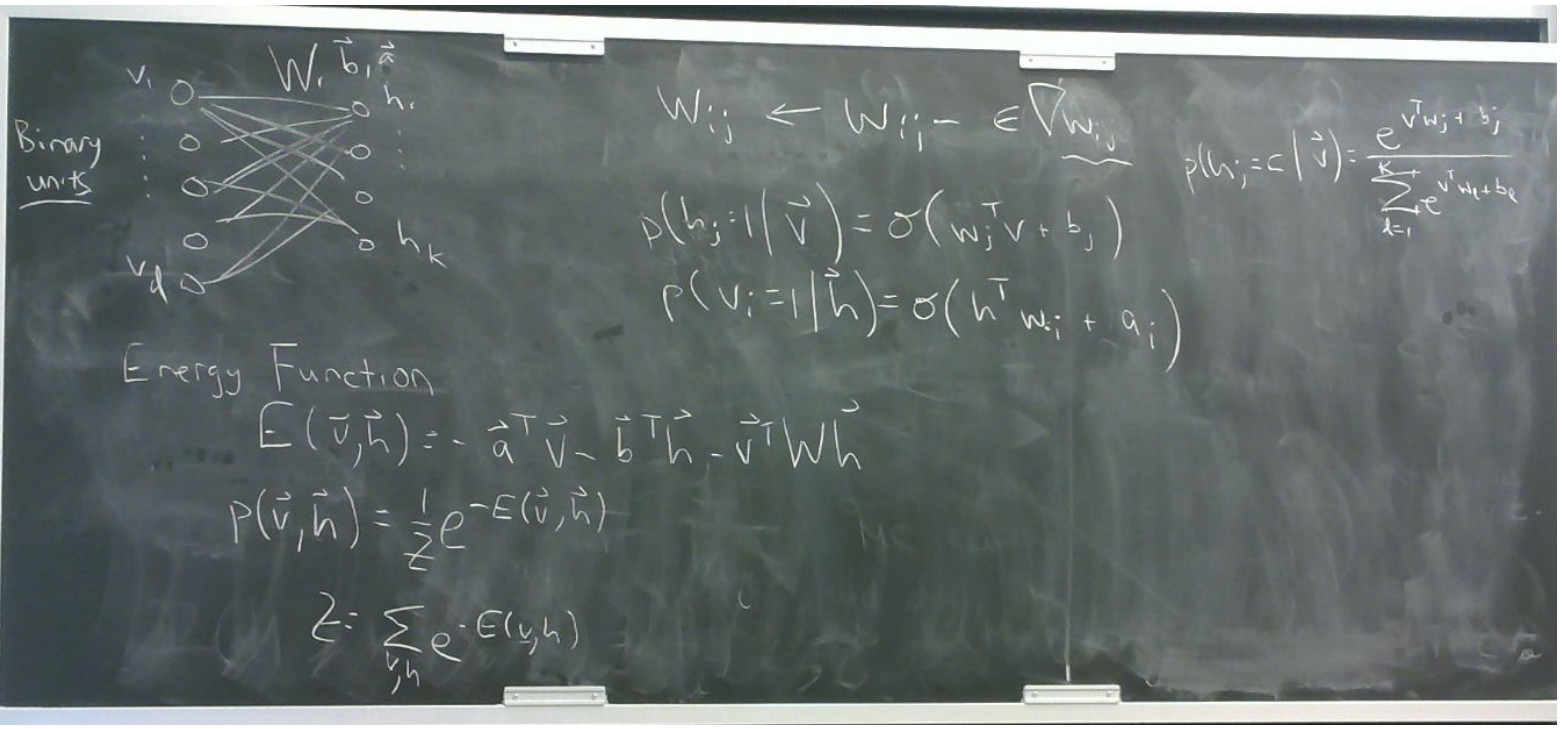
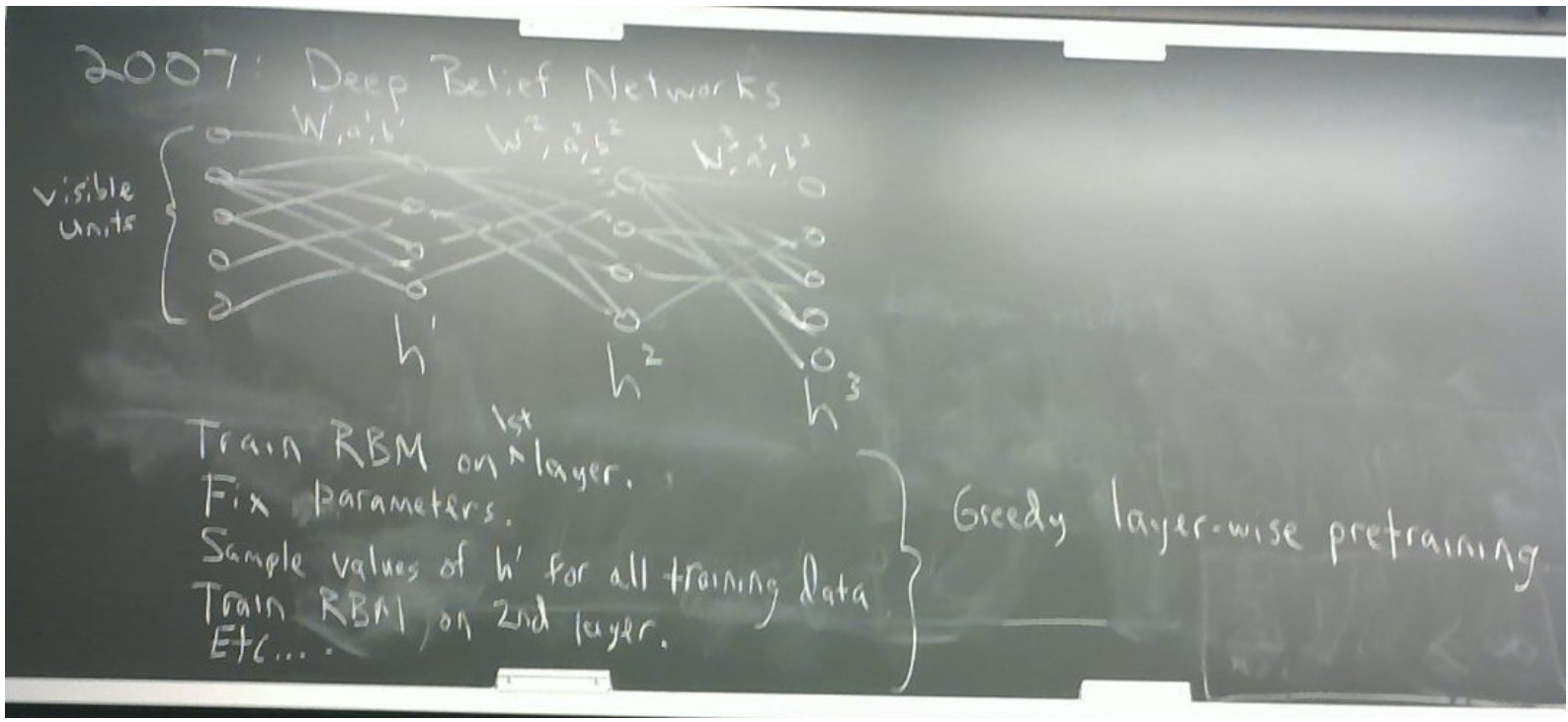
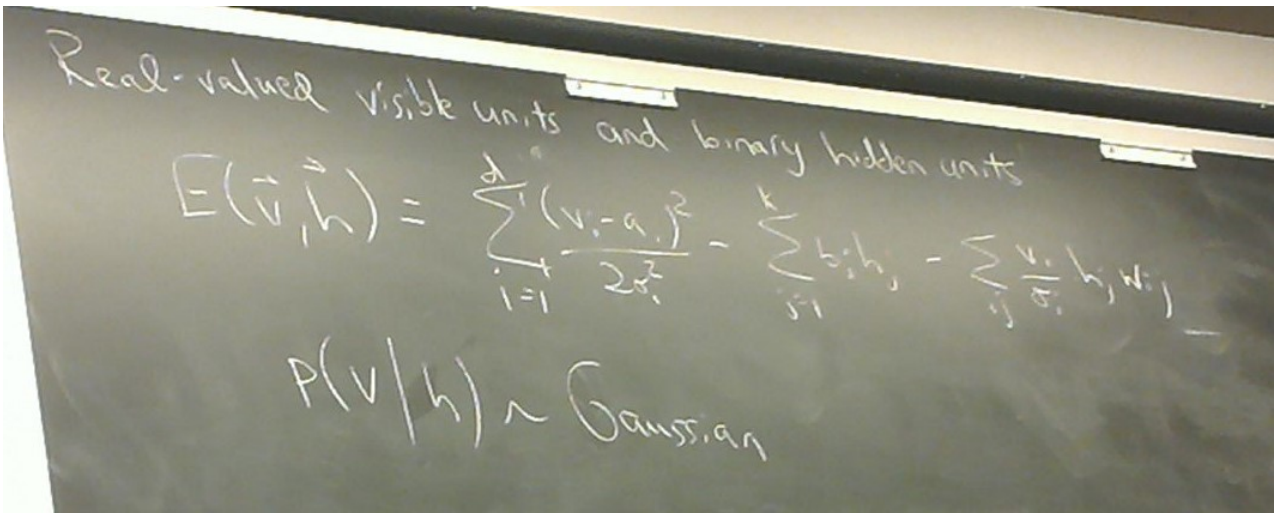

Binary units
v_1
v_d
$W, \vec{b}, \vec{a}$
h_1
h_k
$W_{ij} \leftarrow W_{ij} - \epsilon \nabla_{W_{ij}}$
$p(h_j = 1 \mid \vec{v}) = \sigma(w_j^T v + b_j)$
$p(v_i = 1 \mid \vec{h}) = \sigma(h^T w_i + a_i)$
$p(h_j = c \mid \vec{v}) = \frac{e^{v^T w_j + b_j}}{\sum_{k=1}^{K} e^{v^T w_k + b_k}}$
Energy Function
$E(\vec{v}, \vec{h}) = -\vec{a}^T \vec{v} - \vec{b}^T \vec{h} - \vec{v}^T W \vec{h}$
$p(\vec{v}, \vec{h}) = \frac{1}{Z} e^{-E(\vec{v}, \vec{h})}$
$Z = \sum_{v,h} e^{-E(v,h)}$


2007: Deep Belief Networks
Visible units
W_1, a^1, b^1
W^2, a^2, b^2
W^3, a^3, b^3
h^1
h^2
h^3
Train RBM on 1st layer.
Fix parameters.
Sample values of h^1 for all training data
Train RBM on 2nd layer.
Etc...
Greedy layer-wise pretraining


Real-valued visible units and binary hidden units
$E(\vec{v}, \vec{h}) = \sum_{i=1}^{d} \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{j=1}^{k} b_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j w_{ij}$
$p(v \mid h) \sim$ Gaussian