

Formulas PDF

UNIT 1

Name (When to Use)	Formula
Sparse Autoencoder (SAE) (Encourages sparsity in hidden units)	$L = \ X - \hat{X}\ ^2 + \lambda \ W\ ^2 + \beta KL(\rho \ \hat{\rho})$
KL Divergence for SAE (Penalty Term)	$KL(\rho \ \hat{\rho}) = \sum_j \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j}$
Denosing Autoencoder (DAE) (Use when input is noisy, to improve robustness)	$L = \ X - \hat{X}\ ^2 + \lambda \ W\ ^2$
Gaussian Noise for DAE	$X_{\text{noisy}} = X + \mathcal{N}(0, \sigma^2)$
Contractive Autoencoder (CAE) (Use when you want robustness to small input variations)	$L = \ X - \hat{X}\ ^2 + \lambda \ J_h\ ^2$
Jacobian Norm (Penalty Term for CAE)	$\ J_h\ ^2 = \sum_i \sum_j \left(\frac{\partial h_i}{\partial x_j} \right)^2$
KL Divergence for VAE (Posterior vs Prior) (Used in Variational Autoencoders to regularize latent space)	$D_{KL}(q(z x) \ p(z)) = \frac{1}{2} \sum_{i=1}^d (\sigma_i^2 + \mu_i^2 - 1 - \log \sigma_i^2)$
Compute σ from log-variance (Used in VAEs where network outputs $\log \sigma^2$)	$\sigma = \exp \left(\frac{\log \sigma^2}{2} \right)$
Reparameterization Trick (Used in VAEs to allow backpropagation through stochastic sampling)	$z = \mu + \sigma \odot \epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$
Li Penalty for Sparsity (Use to encourage sparse connections)	$\lambda \sum_i w_i$

UNIT 2

Name (When to Use)	Formula
Hopfield Network Energy Function (Used to find energy associated with a binary state configuration)	$E = - \sum_i b_i s_i - \sum_{i,j; i < j} w_{ij} s_i s_j$
Hopfield Energy Gap (Used to decide whether flipping s_i is beneficial)	$\Delta E_i = E_{s_i=0} - E_{s_i=1} = b_i + \sum_{j \neq i} w_{ij} s_j$
Hopfield Update Rule (State update based on energy gap)	$s_i = \begin{cases} 1 & \text{if } \sum_{j \neq i} w_{ij} s_j + b_i \geq 0 \\ 0 & \text{otherwise} \end{cases}$
Hebbian Learning Rule for Hopfield Networks (Used to compute weights from training data)	$w_{ij} = \frac{4}{n} \sum_{k=1}^n (x_{ki} - 0.5)(x_{kj} - 0.5)$
Boltzmann Machine Energy Function (Defines energy for visible and hidden states)	$E(s) = - \sum_i b_i s_i - \sum_{i,j; i < j} w_{ij} s_i s_j$
Boltzmann Probability of a State (Defines probability distribution over states)	$P(s) = \frac{e^{-E(s)}}{Z}$
Partition Function (Boltzmann Machine) (Normalization factor for probability)	$Z = \sum_s e^{-E(s)}$

Name (When to Use)	Formula
Probability of Unit Activation (Boltzmann) <i>(Used in Gibbs Sampling)</i>	$P(s_i = 1) = \frac{1}{1 + e^{-\Delta E_i}}$
Gradient of Log-Likelihood w.r.t Weight <i>(Used in weight updates for Boltzmann Machines)</i>	$\frac{\partial \log P(s)}{\partial w_{ij}} = \langle s_i s_j \rangle_{\text{data}} - \langle s_i s_j \rangle_{\text{model}}$
Weight Update Rule (Boltzmann Machine)	$w_{ij} \leftarrow w_{ij} + \alpha (\langle s_i s_j \rangle_{\text{data}} - \langle s_i s_j \rangle_{\text{model}})$
Bias Update Rule (Boltzmann Machine)	$b_i \leftarrow b_i + \alpha (\langle s_i \rangle_{\text{data}} - \langle s_i \rangle_{\text{model}})$
RBM Hidden Unit Activation Probability <i>(Given visible units)</i>	$P(h_j = 1 \mid v) = \frac{1}{1 + \exp(-b_j^{(h)} - \sum_i v_i w_{ij})}$
RBM Visible Unit Activation Probability <i>(Given hidden units)</i>	$P(v_i = 1 \mid h) = \frac{1}{1 + \exp(-b_i^{(v)} - \sum_j h_j w_{ij})}$
RBM Weight Update Rule	$w_{ij} \leftarrow w_{ij} + \alpha (\langle v_i h_j \rangle_{\text{pos}} - \langle v_i h_j \rangle_{\text{neg}})$
RBM Bias Update for Visible Unit	$b_i^{(v)} \leftarrow b_i^{(v)} + \alpha (\langle v_i \rangle_{\text{pos}} - \langle v_i \rangle_{\text{neg}})$
RBM Bias Update for Hidden Unit	$b_j^{(h)} \leftarrow b_j^{(h)} + \alpha (\langle h_j \rangle_{\text{pos}} - \langle h_j \rangle_{\text{neg}})$
Gaussian RBM Energy Function <i>(For real-valued data)</i>	$E(v, h) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_j b_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j w_{ij}$

UNIT 3

Name (When to Use)	Formula
Joint Distribution of DBN <i>(Stacked RBMs, top RBM undirected, others directed)</i>	$P(v, h^{(1)}, \dots, h^{(L)}) = P(h^{(L-1)}, h^{(L)}) \prod_{l=1}^{L-2} P(h^{(l)} \mid h^{(l+1)})$
Energy Function of RBM	$E(v, h) = -b^\top v - c^\top h - v^\top W h$
RBM Joint Distribution	$P(v, h) = \frac{1}{Z} \exp(-E(v, h))$
Conditional Probabilities in RBM	$P(h_i = 1 \mid v) = \sigma \left(\sum_j W_{ji} v_j + c_i \right)$ $P(v_j = 1 \mid h) = \sigma \left(\sum_i W_{ji} h_i + b_j \right)$
Energy Function of DBM (with bias)	$E(v, h^{(1)}, h^{(2)}, h^{(3)}) = -v^\top W^{(1)} h^{(1)} - h^{(1)\top} W^{(2)} h^{(2)} - h^{(2)\top} W^{(3)} h^{(3)} - b^\top v - c^{(1)\top} h$
DBM Probability Distribution	$P(v, h^{(1)}, h^{(2)}, h^{(3)}) = \frac{1}{Z} \exp(-E(v, h^{(1)}, h^{(2)}, h^{(3)}))$
Gaussian-Bernoulli RBM Energy (unit variance)	$E(v, h) = \sum_i \frac{(v_i - b_i)^2}{2} - \sum_i \sum_j v_i W_{ij} h_j - \sum_j c_j h_j$
Gaussian-Bernoulli	$E(v, h) = \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i,j} \frac{v_i}{\sigma_i} W_{ij} h_j - \sum_j c_j h_j$

Name (When to Use)	Formula
RBM Energy (non-unit variance)	
Mean and Covariance RBM Energy (mcRBM)	$E_{mc}(v, h^{(m)}, h^{(c)}) = E_m(v, h^{(m)}) + \sum_j h_j^{(c)} \left(\frac{1}{2} v^\top r_j r_j^\top v - b_j^{(c)} \right)$
mPoT Energy (Mean-Product of Student's t)	$E_{mPoT}(v, h) = \frac{1}{2} \ v\ ^2 - \sum_j h_j \left(\frac{1}{2} v^\top r_j r_j^\top v - b_j \right)$
Spike and Slab Mean	$\mathbb{E}[v \mid h, s] = (h \odot s) W^\top$
Spike and Slab Marginal	$p(v \mid h) = \mathcal{N}(v \mid \mu(h), \Sigma(h))$
MMD Loss (GMMN)	$\text{MMD}(p, q) = \ \mu_p - \mu_q\ $, where $\mu_p = \mathbb{E}_{x \sim p}[\phi(x)]$ and $\mu_q = \mathbb{E}_{x \sim q}[\phi(x)]$
Kernel Function (Gaussian)	$k(x, y) = \exp \left(-\frac{\ x-y\ ^2}{2\sigma^2} \right)$
Change of Variables (Flow models)	$p(x) = p(z) \cdot \det \left(\frac{\partial z}{\partial x} \right)$
Variational Lower Bound (ELBO)	$\log p(v) \geq \mathbb{E}_{Q(h v)}[\log P(v, h)] - \mathbb{E}_{Q(h v)}[\log Q(h \mid v)]$
REINFORCE Estimator	$\nabla_\theta \mathbb{E}_{z \sim p(z; \theta)}[J(f(z))] = \mathbb{E}_{z \sim p(z; \theta)}[J(f(z)) \nabla_\theta \log p(z; \theta)]$
REINFORCE with Baseline	$\nabla_\theta \approx (J(f(z)) - b) \nabla_\theta \log p(z; \theta)$
Gumbel-Softmax Sampling	$y_i = \frac{\exp((\log p_i + g_i)/\tau)}{\sum_j \exp((\log p_j + g_j)/\tau)}, \quad g_i = -\log(-\log(\epsilon)), \epsilon \sim U(0, 1)$
Sigmoid Belief Network (SBN) Conditional	$P(h_i^{(l)} = 1 \mid h^{(l+1)}) = \sigma \left(\sum_j w_{ij} h_j^{(l+1)} + b_i \right)$
NADE Factorization	$P(x) = \prod_{i=1}^d P(x_i \mid x_1, \dots, x_{i-1})$
GAN Objective	$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))]$
KL Divergence (GSN example)	$\text{KL}(q(zx) \parallel p(z)) = \log \left(\frac{\sigma_{\text{prior}}}{\sigma_{\text{posterior}}} \right) + \frac{\sigma_{\text{posterior}}^2 + (\mu_{\text{posterior}} - \mu_{\text{prior}})^2}{2\sigma_{\text{prior}}^2} - \frac{1}{2}$
Expected Log Likelihood	$\log p(xz) = -\frac{(x-z)^2}{2} - \frac{1}{2} \log(2\pi)$
2-Hidden Layer DBM	$E(v, h^{(1)}, h^{(2)}) = -v^\top W^{(1)} h^{(1)} - h^{(1)\top} W^{(2)} h^{(2)}$

Name (When to Use)	Formula
Energy Function	
Expected Energy under Posterior	$\mathbb{E}_Q[-E(v, h^{(1)}, h^{(2)})] = (\text{sum of weighted terms under } Q)$
Entropy of Posterior	$H(Q) \approx 3.0$ (approximated)
Lower Bound Estimate (ELBO for DBM)	$L(Q, \theta) = \mathbb{E}_Q[-E] + H(Q) = 0.042 + 3.0 = 3.042$

UNIT 4

Formula Name (Usage)	Formula
Forward Process (General)	$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t I)$
Forward Process (Sampling x_t from x_0)	$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \epsilon \sim \mathcal{N}(0, I)$
Reverse Process (General)	$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t))$
Reverse Mean $\mu_\theta(x_t, t)$	$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right)$
Reverse Variance (Fixed)	$\Sigma_\theta(x_t) = \beta_t I$
Sampling x_{t-1} from Reverse Distribution	$x_{t-1} = \mu_\theta + \sqrt{\beta_t} \cdot \epsilon, \epsilon \sim \mathcal{N}(0, I)$
Variance Schedule (Linear)	$\beta_t = \beta_1 + \left(\frac{t-1}{T-1}\right)(\beta_T - \beta_1)$
Compute $\bar{\alpha}_t$ (Alpha Bar)	$\bar{\alpha}_t = \prod_{s=1}^t (1 - \beta_s)$
Compute Noise ϵ from x_t and x_0	$\epsilon = \frac{x_t - \sqrt{\bar{\alpha}_t} x_0}{\sqrt{1 - \bar{\alpha}_t}}$
Compute x_0 from x_t and ϵ_θ	$\hat{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta \right)$

UNIT 5

Formula Name (Usage)	Formula
Binary Cross-Entropy Loss (General Form)	$L(y, p) = -[y \log(p) + (1 - y) \log(1 - p)]$
Discriminator Loss	$L_D = -\frac{1}{N} \sum_{i=1}^N \log D(x_{(i)}) - \frac{1}{M} \sum_{j=1}^M \log(1 - D(G(z_{(j)})))$
Generator Loss (Minimax)	$L_G = -\frac{1}{M} \sum_{j=1}^M \log D(G(z_{(j)}))$
Combined GAN Minimax Loss	$\min_G \max_D V(D, G) = \frac{1}{N} \sum_{i=1}^N \log D(x^{(i)}) + \frac{1}{M} \sum_{j=1}^M \log(1 - D(G(z^{(j)})))$
Gradient Penalty Term	$GP = \lambda \cdot \frac{1}{K} \sum_{k=1}^K \left(\left\ \nabla_{\hat{x}_{(k)}} D(\hat{x}_{(k)}) \right\ _2 - c \right)^2$

Formula Name (Usage)	Formula
Discriminator Loss with Gradient Penalty	$L_D = -\frac{1}{N} \sum_{i=1}^N \log D(x_{(i)}) - \frac{1}{M} \sum_{j=1}^M \log(1 - D(G(z_{(j)}))) + GP$