Autoencoder - Derivations & Proofs

Paul F. Roysdon, Ph.D.

Contents

1 Mathematical Derivations & Proofs

1.1 Introduction

An autoencoder (AE) is a neural architecture that learns an *encoder* mapping inputs to a low-dimensional representation (latent code) and a *decoder* that reconstructs the inputs from that code. Training minimizes a reconstruction loss, optionally augmented with regularizers that shape the representation (sparsity, contraction, denoising). We derive: (i) the AE objective from empirical risk minimization, (ii) complete backpropagation gradients for a one-hidden-layer AE (generalizes to deep AEs by recursion), and (iii) the connection between the *linear* AE and PCA, giving a proof (via projection and Eckart–Young) that the optimal linear AE spans the principal subspace. We also state variants (denoising, sparse, contractive) with their objectives and gradients.

1.2 Data and Notation

Given training data

$$\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{n}, \qquad \mathbf{x}_i \in \mathbb{R}^d,$$

we learn encoder/decoder functions $f_{\boldsymbol{\phi}} : \mathbb{R}^d \to \mathbb{R}^r$ and $g_{\boldsymbol{\theta}} : \mathbb{R}^r \to \mathbb{R}^d$ with $r \ll d$ (bottleneck). For the canonical one-hidden-layer AE.

The Encoder is defined

$$\mathbf{h} = f_{\boldsymbol{\phi}}(\mathbf{x}) = \sigma_e(\mathbf{W}_e\mathbf{x} + \mathbf{b}_e) \in \mathbb{R}^r,$$

the Decoder

$$\widehat{\mathbf{x}} = g_{\boldsymbol{\theta}}(\mathbf{h}) = \sigma_d(\mathbf{W}_d\mathbf{h} + \mathbf{b}_d) \in \mathbb{R}^d,$$

with parameters:

$$\boldsymbol{\phi} = \{\mathbf{W}_e \in \mathbb{R}^{r\times d},\ \mathbf{b}_e \in \mathbb{R}^r\}, \qquad \boldsymbol{\theta} = \{\mathbf{W}_d \in \mathbb{R}^{d\times r},\ \mathbf{b}_d \in \mathbb{R}^d\},$$

where the weights are $\mathbf{W}_e, \mathbf{W}_d$ and biases $\mathbf{b}_e, \mathbf{b}_d$.

For a mini-batch $\mathcal{B}$ we stack columns in matrices $\mathbf{X} \in \mathbb{R}^{d\times|\mathcal{B}|}$, $\mathbf{H} \in \mathbb{R}^{r\times|\mathcal{B}|}$, $\widehat{\mathbf{X}} \in \mathbb{R}^{d\times|\mathcal{B}|}$.

1.3 Model Formulation and Objective

Let $\ell(\mathbf{x}, \widehat{\mathbf{x}})$ measure reconstruction error. The empirical risk (with optional weight decay) is

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \ \widehat{\mathcal{L}}(\boldsymbol{\phi},\boldsymbol{\theta}) \ = \ \frac{1}{n}\sum_{i=1}^{n} \ell\big(\mathbf{x}_i,\ g_{\boldsymbol{\theta}}(f_{\boldsymbol{\phi}}(\mathbf{x}_i))\big) \ + \ \frac{\lambda}{2}\Big(\|\mathbf{W}_e\|_F^2 + \|\mathbf{W}_d\|_F^2\Big). \tag{1}$$

Common choices:

$$\ell_{\mathrm{MSE}}(\mathbf{x}, \widehat{\mathbf{x}}) = \frac{1}{2}\|\widehat{\mathbf{x}} - \mathbf{x}\|_2^2, \qquad \ell_{\mathrm{CE}}(\mathbf{x}, \widehat{\mathbf{x}}) = -\sum_{j=1}^{d} \Big(x_j \log \widehat{x}_j + (1 - x_j)\log(1 - \widehat{x}_j)\Big)$$

(MSE for real-valued inputs with linear decoder; cross-entropy for binary inputs with sigmoid decoder).

1.4 Backpropagation Derivation (One-Hidden-Layer AE)

For a single example, define pre-activations

$$\mathbf{a}^{(e)} = \mathbf{W}_e\mathbf{x} + \mathbf{b}_e, \qquad \mathbf{h} = \sigma_e(\mathbf{a}^{(e)}), \qquad \mathbf{a}^{(d)} = \mathbf{W}_d\mathbf{h} + \mathbf{b}_d, \qquad \widehat{\mathbf{x}} = \sigma_d(\mathbf{a}^{(d)}).$$

Gradients follow by chain rule. Let $\odot$ denote Hadamard product and σ_e', σ_d' the elementwise derivatives.

Decoder layer. For general ℓ,

$$\frac{\partial \ell}{\partial \mathbf{a}^{(d)}} = \Big(\nabla_{\widehat{\mathbf{x}}}\ell\Big) \odot \sigma_d'(\mathbf{a}^{(d)}).$$

For MSE: $\nabla_{\widehat{\mathbf{x}}}\ell = \widehat{\mathbf{x}} - \mathbf{x}$, thus

$$\boldsymbol{\delta}^{(d)} \ \triangleq \ \frac{\partial \ell}{\partial \mathbf{a}^{(d)}} = (\widehat{\mathbf{x}} - \mathbf{x}) \odot \sigma_d'(\mathbf{a}^{(d)}). \tag{2}$$

Gradients:

$$\nabla_{\mathbf{W}_d}\ell = \boldsymbol{\delta}^{(d)}\,\mathbf{h}^\top + \lambda\mathbf{W}_d, \qquad \nabla_{\mathbf{b}_d}\ell = \boldsymbol{\delta}^{(d)}. \tag{3}$$

Encoder layer. Backpropagate through decoder:

$$\frac{\partial \ell}{\partial \mathbf{h}} = \mathbf{W}_d^\top \boldsymbol{\delta}^{(d)}, \qquad \boldsymbol{\delta}^{(e)} \ \triangleq \ \frac{\partial \ell}{\partial \mathbf{a}^{(e)}} = \Big(\mathbf{W}_d^\top \boldsymbol{\delta}^{(d)}\Big) \odot \sigma_e'(\mathbf{a}^{(e)}).$$

Gradients:

$$\nabla_{\mathbf{W}_e}\ell = \boldsymbol{\delta}^{(e)}\,\mathbf{x}^\top + \lambda\mathbf{W}_e, \qquad \nabla_{\mathbf{b}_e}\ell = \boldsymbol{\delta}^{(e)}. \tag{4}$$

Proof (chain rule). *Proof.* For MSE, $\ell = \frac{1}{2}\|\widehat{\mathbf{x}} - \mathbf{x}\|^2$ so $d\ell = (\widehat{\mathbf{x}} - \mathbf{x})^\top d\widehat{\mathbf{x}}$. With $\widehat{\mathbf{x}} = \sigma_d(\mathbf{a}^{(d)})$ and $\mathbf{a}^{(d)} = \mathbf{W}_d\mathbf{h} + \mathbf{b}_d$,

$$d\ell = \underbrace{\big((\widehat{\mathbf{x}} - \mathbf{x}) \odot \sigma_d'(\mathbf{a}^{(d)})\big)}_{\boldsymbol{\delta}^{(d)}}{}^\top \big(d\mathbf{W}_d\,\mathbf{h} + \mathbf{W}_d\,d\mathbf{h} + d\mathbf{b}_d\big).$$

Collect coefficients of $d\mathbf{W}_d$ and $d\mathbf{b}_d$ to obtain Eqn. (3). Then propagate $d\mathbf{h}$ back through $\mathbf{h} = \sigma_e(\mathbf{a}^{(e)})$ with $\mathbf{a}^{(e)} = \mathbf{W}_e\mathbf{x} + \mathbf{b}_e$ to obtain Eqn. (4). ■

Mini-batch form. Stacking columns for batch $\mathcal{B}$,

$$\boldsymbol{\Delta}^{(d)} = (\widehat{\mathbf{X}} - \mathbf{X}) \odot \sigma_d'(\mathbf{A}^{(d)}), \quad \nabla_{\mathbf{W}_d}\widehat{\mathcal{L}}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|}\boldsymbol{\Delta}^{(d)}\mathbf{H}^\top + \lambda\mathbf{W}_d,$$

$$\boldsymbol{\Delta}^{(e)} = \big(\mathbf{W}_d^\top \boldsymbol{\Delta}^{(d)}\big) \odot \sigma_e'(\mathbf{A}^{(e)}), \quad \nabla_{\mathbf{W}_e}\widehat{\mathcal{L}}_{\mathcal{B}} = \frac{1}{|\mathcal{B}|}\boldsymbol{\Delta}^{(e)}\mathbf{X}^\top + \lambda\mathbf{W}_e.$$

Tied weights (optional). If we enforce $\mathbf{W}_d = \mathbf{W}_e^\top$ (parameter sharing), then the total derivative w.r.t. $\mathbf{W}_e$ includes both paths:

$$\nabla_{\mathbf{W}_e}\ell\Big|_{\text{tied}} \;=\; \boldsymbol{\delta}^{(e)}\mathbf{x}^\top \;+\; \Big(\nabla_{\mathbf{W}_d}\ell\Big)^\top \;+\; \lambda\mathbf{W}_e.$$

1.5 Linear Autoencoder and PCA (Equivalence Proof)

Assume centered data ($\frac{1}{n}\sum_i \mathbf{x}_i = \mathbf{0}$), linear activations ($\sigma_e, \sigma_d$ identity), no biases, and tied weights with orthonormal rows:

$$\mathbf{h} = \mathbf{W}\mathbf{x}, \quad \widehat{\mathbf{x}} = \mathbf{W}^\top\mathbf{h} = \mathbf{W}^\top\mathbf{W}\mathbf{x}, \quad \mathbf{W} \in \mathbb{R}^{r\times d}, \;\; \mathbf{W}\mathbf{W}^\top = \mathbf{I}_r.$$

The reconstruction error is

$$\frac{1}{n}\sum_{i=1}^{n} \tfrac{1}{2}\|\mathbf{x}_i - \widehat{\mathbf{x}}_i\|^2 = \frac{1}{2}\operatorname{tr}\Big(\mathbf{S} - 2\mathbf{S}\mathbf{P} + \mathbf{P}\mathbf{S}\mathbf{P}\Big), \quad \mathbf{S} = \frac{1}{n}\sum_i \mathbf{x}_i\mathbf{x}_i^\top, \;\; \mathbf{P} = \mathbf{W}^\top\mathbf{W}. \tag{5}$$

Since $\mathbf{P}$ is an orthogonal projector onto the row-space of $\mathbf{W}$, it satisfies $\mathbf{P} = \mathbf{P}^\top = \mathbf{P}^2$. Then $\mathbf{P}\mathbf{S}\mathbf{P} = \mathbf{P}\mathbf{S}$, and Eqn. (5) reduces to

$$\frac{1}{2}\operatorname{tr}(\mathbf{S}) - \operatorname{tr}(\mathbf{S}\mathbf{P}).$$

The first term is constant; minimizing reconstruction error is equivalent to maximizing $\operatorname{tr}(\mathbf{S}\mathbf{P})$ over rank-r projectors $\mathbf{P}$.

Eckart–Young/Principal subspace argument. *Proof.* Let $\mathbf{S} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ with eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$. Any rank-$r$ projector can be written $\mathbf{P} = \mathbf{U}\begin{bmatrix}\mathbf{I}_r & \\ & \mathbf{0}\end{bmatrix}\mathbf{U}^\top$ after choosing an r-dimensional subspace. Then

$$\operatorname{tr}(\mathbf{S}\mathbf{P}) = \sum_{j=1}^{r}\lambda_{i_j} \leq \sum_{j=1}^{r}\lambda_j,$$

with equality when the projector spans the top-r eigenvectors. Therefore the linear AE with orthonormal-tied weights learns the PCA principal subspace, and the reconstruction $\widehat{\mathbf{x}} = \mathbf{P}\mathbf{x}$ is the orthogonal projection onto that subspace. ■

Untied weights, no orthonormal constraint. Even without explicit orthonormality, at the optimum (with linear decoder/encoder and centered data), the column space of $\mathbf{W}_d$ equals the row space of $\mathbf{W}_e$ and spans the top-r eigenvectors; the factorization $\mathbf{W}_d\mathbf{W}_e$ is the best rank-r approximation to the identity in the metric induced by $\mathbf{S}$, again by Eckart–Young for low-rank approximation of the data matrix.

1.6 Regularized Variants

Denoising Autoencoder (DAE). Corrupt inputs by $\widetilde{\mathbf{x}} \sim q(\widetilde{\mathbf{x}} \mid \mathbf{x})$ (e.g., Gaussian noise or masking) and train to reconstruct the clean $\mathbf{x}$:

$$\min_{\boldsymbol{\phi},\boldsymbol{\theta}} \;\; \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{\widetilde{\mathbf{x}}_i\sim q(\cdot|\mathbf{x}_i)}\ell\big(\mathbf{x}_i,\; g_{\boldsymbol{\theta}}(f_{\boldsymbol{\phi}}(\widetilde{\mathbf{x}}_i))\big) \;+\; \frac{\lambda}{2}\big(\|\mathbf{W}_e\|_F^2 + \|\mathbf{W}_d\|_F^2\big). \tag{6}$$

Gradients are as in Section 1.4 (replace $\mathbf{x}$ by $\widetilde{\mathbf{x}}$ in the forward pass, keep the target $\mathbf{x}$). For small additive Gaussian noise of variance σ^2 and MSE loss, $g_{\boldsymbol{\theta}}(f_{\boldsymbol{\phi}}(\mathbf{x})) - \mathbf{x} \approx \sigma^2\nabla_{\mathbf{x}}\log p(\mathbf{x})$ (score-matching connection).

Sparse Autoencoder. Encourage low average activation in the code by a KL penalty

$$\Omega_{\text{sparse}}(\mathbf{H}) = \beta \sum_{j=1}^{r} \mathrm{KL}\big(\rho \parallel \widehat{\rho}_j\big), \quad \widehat{\rho}_j = \frac{1}{n}\sum_{i=1}^{n} h_{ij}, \quad \mathrm{KL}(\rho \parallel \widehat{\rho}) = \rho \log \frac{\rho}{\widehat{\rho}} + (1-\rho)\log \frac{1-\rho}{1-\widehat{\rho}}. \tag{7}$$

The gradient contribution to the encoder pre-activation for unit j adds

$$\frac{\partial \Omega_{\text{sparse}}}{\partial a_j^{(e)}} \approx \beta\Big(-\frac{\rho}{\widehat{\rho}_j} + \frac{1-\rho}{1-\widehat{\rho}_j} \Big)\, \sigma_e'(a_j^{(e)}),$$

broadcast to all samples; in practice one uses mini-batch estimates of $\widehat{\rho}_j$.

Contractive Autoencoder (CAE). Penalize the encoder Jacobian to enforce local robustness:

$$\Omega_{\text{contr}}(\boldsymbol{\phi}) = \frac{\gamma}{n}\sum_{i=1}^{n} \Big\| \nabla_{\mathbf{x}} f_{\boldsymbol{\phi}}(\mathbf{x}_i) \Big\|_F^2. \tag{8}$$

For elementwise σ_e with rows $\boldsymbol{w}_{e,j}^\top$ of $\mathbf{W}_e$,

$$\Big\| \nabla_{\mathbf{x}} f_{\boldsymbol{\phi}}(\mathbf{x}) \Big\|_F^2 = \sum_{j=1}^{r} \big(\sigma_e'(a_j^{(e)})\big)^2 \|\boldsymbol{w}_{e,j}\|_2^2,$$

so CAE adds a data-dependent weight decay scaled by $\big(\sigma_e'(a_j^{(e)})\big)^2$.

1.7 Algorithm (Autoencoder Training)

1. **Input:** data $\{\mathbf{x}_i\}$, code dimension r, activations (σ_e, σ_d), loss ℓ, regularization hyperparameters (λ, β, γ).
2. **Initialize:** parameters $\boldsymbol{\phi}, \boldsymbol{\theta}$ (e.g., Xavier/He).
3. **Iterate (SGD/Adam on mini-batches):**
   (a) (Optional corruption) draw $\widetilde{\mathbf{x}}$ for DAE.
   (b) Forward: $\mathbf{h} = \sigma_e(\mathbf{W}_e\mathbf{x} + \mathbf{b}_e)$; $\widehat{\mathbf{x}} = \sigma_d(\mathbf{W}_d\mathbf{h} + \mathbf{b}_d)$.
   (c) Compute loss $\widehat{\mathcal{L}}_{\mathcal{B}}$ (ERM + regularizers).
   (d) Backprop: compute $\boldsymbol{\Delta}^{(d)}$, $\boldsymbol{\Delta}^{(e)}$; form gradients Eqns. (3)–(4) plus regularizer terms.
   (e) Update parameters.
4. **Output:** trained encoder $f_{\boldsymbol{\phi}}$ (features) and decoder $g_{\boldsymbol{\theta}}$ (reconstruction).

1.8 Summary of Variables and Their Dimensions

- $\mathbf{x}_i \in \mathbb{R}^d$: input vector (column).
- r: code dimension (bottleneck), $1 \le r \le d$.
- Encoder parameters: $\mathbf{W}_e \in \mathbb{R}^{r\times d}$, $\mathbf{b}_e \in \mathbb{R}^r$; pre-activation $\mathbf{a}^{(e)} \in \mathbb{R}^r$; code $\mathbf{h} \in \mathbb{R}^r$.
- Decoder parameters: $\mathbf{W}_d \in \mathbb{R}^{d\times r}$, $\mathbf{b}_d \in \mathbb{R}^d$; pre-activation $\mathbf{a}^{(d)} \in \mathbb{R}^d$; reconstruction $\widehat{\mathbf{x}} \in \mathbb{R}^d$.
- Loss $\ell(\mathbf{x}, \widehat{\mathbf{x}}) \in \mathbb{R}_{\ge 0}$; regularization $\lambda, \beta, \gamma \in \mathbb{R}_{\ge 0}$.
- (Linear AE/PCA) $\mathbf{S} \in \mathbb{R}^{d\times d}$ covariance; projector $\mathbf{P} = \mathbf{W}^\top\mathbf{W}$ with $\mathbf{W} \in \mathbb{R}^{r\times d}$, $\mathbf{W}\mathbf{W}^\top = \mathbf{I}_r$.

1.9 Summary

From first principles, an autoencoder minimizes empirical reconstruction error Eqn. (1) under an encoder–decoder factorization. Exact gradients are obtained by the chain rule Eqns. (2)–(4), yielding standard backprop updates (with tied-weights handled by parameter sharing). In the linear, centered, tied, orthonormal case, the optimal AE computes the orthogonal projection onto the PCA principal subspace, as shown by maximizing tr($\mathbf{SP}$) via Eckart–Young. Regularized variants (denoising, sparse, contractive) add principled penalties or noise models that shape the learned representation while preserving the same backpropagation structure.