Sketch: Accuracy vs. Corruption Probability in Pixel Replacement

Setup

Let $x \in [0,1]^d$ be an input image and $y \in \{1,\ldots,10\}$ the label. We define a corruption operator $\mathcal{C}_p$ that independently replaces each pixel with probability p:

$$(\mathcal{C}_p(x))_i = \begin{cases} u_i, & \text{with probability } p, \\ x_i, & \text{with probability } 1-p, \end{cases} \qquad u_i \sim \mathrm{Uniform}(0,1).$$

Training uses corrupted inputs $\tilde{x} = \mathcal{C}_p(x)$ and clean labels y. We study the test accuracy $A(p)$ of a network trained at corruption strength p.

Corruption as Attenuation + Noise

Let $M_i \sim \mathrm{Bernoulli}(p)$ and $u_i \sim \mathrm{Uniform}(0,1)$, independent. Then

$$\tilde{x} = (1-M) \odot x + M \odot u.$$

Conditioned on x,

$$\mathbb{E}[\tilde{x} \mid x] = (1-p)x + \tfrac{p}{2}\mathbf{1}, \quad \mathrm{Var}(\tilde{x}_i \mid x) = p(1-p)(x_i - \tfrac{1}{2})^2 + \tfrac{p}{12}.$$

So corruption both shrinks the signal by $(1-p)$ and injects noise of scale $\sqrt{p}$.

For a generic scalar score function $s(x)$ (e.g., a logit margin), a first-order expansion gives

$$s(\tilde{x}) \approx s(x) + \nabla_x s(x)^\top (\tilde{x} - x).$$

This is a local but exact linearization of the trained network. The gradient $g(x) = \nabla_x s(x)$ is a measurable sensitivity vector.

Margin Criterion for a Sharp Drop

Define the margin for example (x, y) as

$$\gamma(x) = f_y(x) - \max_{k \neq y} f_k(x),$$

where f_k is the logit for class k. A sufficient condition for label flip is $\gamma(\tilde{x}) < 0$. Using the linearization above,

$$\Delta\gamma \approx g(x)^\top (\tilde{x} - x).$$

Because the corruption is iid across pixels, $\Delta\gamma$ concentrates with variance

$$\mathrm{Var}(\Delta\gamma \mid x) \approx \sum_i g_i(x)^2 \, \mathrm{Var}(\tilde{x}_i \mid x).$$

This suggests a threshold when typical fluctuations match the clean margin:

$$\gamma(x) \approx c\sqrt{\mathrm{Var}(\Delta\gamma \mid x)}.$$

Aggregating over the data distribution yields a population-level crossover $p^\star$. As model size increases, the margin distribution can shift and sharpen, causing a steeper drop in accuracy as p passes $p^\star$.

Finite-Size Scaling Hypothesis

Let $A(p)$ be the test accuracy when training with corruption p. We hypothesize:

- **Shift:** the midpoint $p^\star$ of the accuracy drop increases with model size or data size, reflecting improved margins.
- **Sharpening:** the slope $|A'(p^\star)|$ increases with size, producing an apparently "critical" knee.
- **Collapse:** when plotted against an effective SNR proxy (e.g., $(1-p)/\sqrt{p}$ or a measured margin-to-noise ratio), curves for different sizes align.

This is a finite-size crossover that can mimic a phase transition in the large-system limit.

Testable Predictions

1. Fit $A(p)$ with a sigmoid to estimate $p^\star$ and the slope; study scaling vs. width.
2. Compute $g(x) = \nabla_x \gamma(x)$ on a held-out set; test whether $\gamma(x)/\sqrt{\mathrm{Var}(\Delta\gamma \mid x)}$ predicts failures.
3. Compare MLP vs. CNN: CNNs should tolerate larger $p^\star$ due to inductive bias.
4. Test curve collapse using an empirical SNR proxy derived from margins and gradients.

RBM Baseline (Practical Reference)

As an additional comparison point, one can train a Bernoulli RBM on corrupted inputs and stack a logistic regression classifier on the learned features. This provides a classical unsupervised baseline to contrast with modern supervised networks under the same corruption channel.

RBM Baseline (Mathematical Detail)

Let $v \in \{0,1\}^D$ denote a visible binary vector (e.g., a binarized or rescaled image) and $h \in \{0,1\}^H$ the hidden units. A Bernoulli RBM defines an energy:

$$E(v,h) = -b^\top v - c^\top h - v^\top W h,$$

with parameters (W, b, c). The joint distribution is

$$p(v,h) = \frac{1}{Z}\exp(-E(v,h)), \quad Z = \sum_{v,h}\exp(-E(v,h)).$$

The conditional distributions factorize:

$$p(h_j = 1 \mid v) = \sigma\left(c_j + \sum_i W_{ij} v_i\right), \quad p(v_i = 1 \mid h) = \sigma\left(b_i + \sum_j W_{ij} h_j\right),$$

where $\sigma(x) = (1 + e^{-x})^{-1}$.

Maximum likelihood. For a dataset $\{v^{(n)}\}_{n=1}^{N}$, the log-likelihood gradient is

$$\frac{\partial \log p(v)}{\partial W} = \mathbb{E}_{p(h|v)}[vh^\top] - \mathbb{E}_{p(v,h)}[vh^\top],$$

with analogous expressions for b, c. The second term (model expectation) is intractable.

Contrastive divergence (CD-k). CD approximates the model term by running a short Gibbs chain:

$$v^{(0)} = v, \quad h^{(t)} \sim p(h \mid v^{(t)}), \quad v^{(t+1)} \sim p(v \mid h^{(t)}), \quad t = 0, \ldots, k-1.$$

The update uses

$$\Delta W \propto v^{(0)} {h^{(0)}}^\top - v^{(k)} {h^{(k)}}^\top.$$

The *epoch* analogue for the RBM is the number of full passes over the dataset, denoted by `rbm_n_iter` in our implementation. The RBM "width" is the number of hidden units H (called `n_components` in scikit-learn).

Classification via logistic regression. After unsupervised training, one can map inputs to hidden activations $\phi(v) = \mathbb{E}[h \mid v]$ and train a linear classifier:

$$p_\theta(y \mid v) = \mathrm{softmax}(A\,\phi(v) + d),$$

with cross-entropy loss

$$L(\theta) = -\frac{1}{N} \sum_{n=1}^{N} \log p_\theta(y^{(n)} \mid v^{(n)}).$$

The classifier training iterations are governed by `rbm_classifier_max_iter`.

Corruption channel. When the input is corrupted by the replacement or additive channel, the RBM is trained on $\tilde{v} = \mathcal{C}(v)$; the same corruption is applied at test time for evaluating accuracy vs. corruption strength.