Review on The Forward-Forward Algorithm: Some Preliminary Investigations

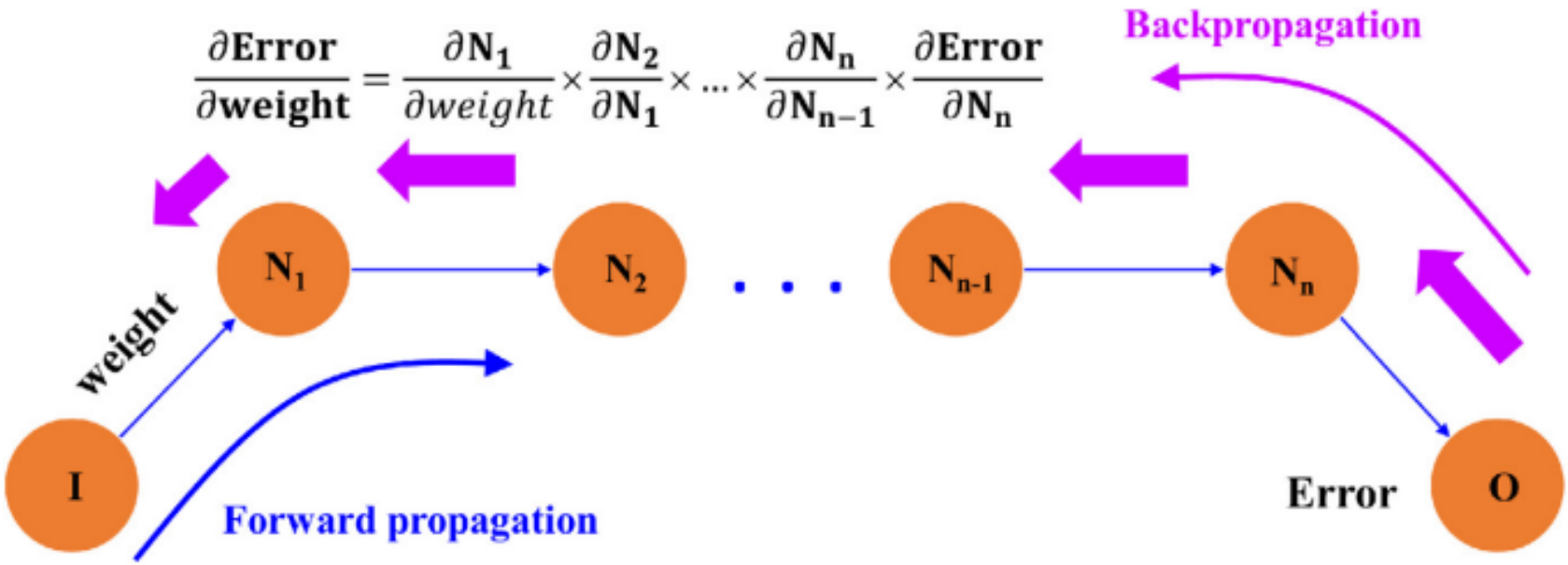
Department of Statistics, Seoul National University

March 4, 2026

Outline

Introduction

Backpropagation

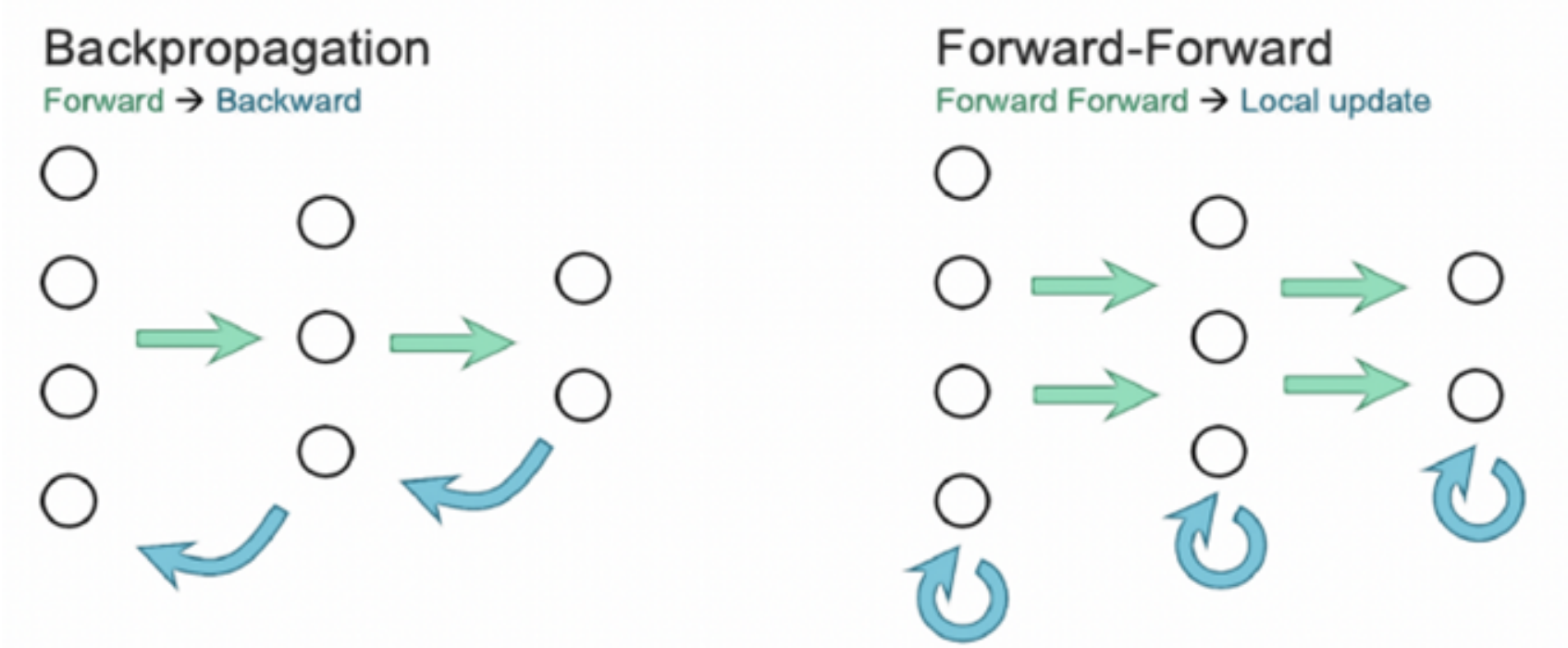


- Backpropagation (BP) is often considered biologically implausible.
  - The brain is not believed to propagate explicit error derivatives.
  - It is unclear how neural circuits could store gradient.

Outline

The Forward Forward Algorithm

FF is a greedy multi-layer learning algorithm



- Instead of forward-backward passes, use two forward passes on negative and positive data to drive the update.

The Forward-Forward Algorithm

Contrastive learning & local update rule

- **Contrastive learning**
  - Define a layer-wise measure, *Goodness*, that is **high** for positive data and **low** for negative data.
  - Let x be the data and y be the label.
  - **Positive data:** the correct pair (x, y).
  - **Negative data:** a mismatched pair $(x, \tilde{y})$ with $\tilde{y} \neq y$.
- **Local update rule (per layer)**
  - Layer-ℓ activity:

$$\mathbf{h}^{(0)} = [\mathbf{x}; \mathbf{y}], \qquad \mathbf{a}^{(\ell)} = \mathrm{ReLU}(W^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}).$$

  - Goodness (sum of squared activities):

$$G = \sum_j {a_j^{(\ell)}}^2 \tag{1}$$

$$P(\text{positive}) = \sigma(G - \theta) \tag{2}$$

where σ is sigmoid and θ is fixed threshold.

The Forward-Forward Algorithm

Inference

- **Softmax head:** Freeze FF layers, attach a classifier on top, and train it to predict y from the final-layer representation.
- **Goodness scoring:** For each candidate label $\tilde{y} \in \{1, \ldots, C\}$, form $(x, \tilde{y})$, run a forward pass, compute

$$S(\tilde{y}) = \sum_{\ell=1}^{L} G^{(\ell)}(x, \tilde{y}),$$

and predict $\hat{y} = \arg\max_{\tilde{y}} S(\tilde{y})$.

The Forward Forward Algorithm

Supervised example in image classification

- Model: network with 4 hidden layers each containing 2000 ReLUs
- Input: replace the first N pixels with the label's one-hot vector.
- Result:
  FF: 1.36% test error on MNIST after 60 epochs
  BP: similar performance after 20 epochs

The Forward Forward Algorithm

Unsupervised example in image classification

- Representation learning by using real data vectors as the positive examples and corrupted data vectors as the negative examples.
- Corrupted data has very different long range correlations but very similar short range correlations
- Result: 1.37% test error for 100 epochs

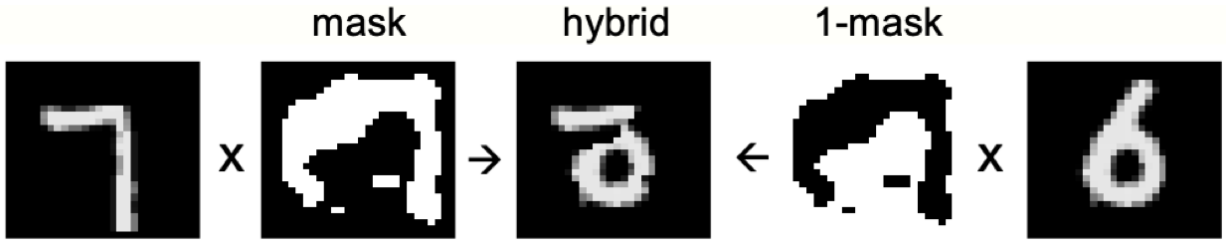


Figure: Corrupted data example

The Forward Forward Algorithm

Conclusion

- Strengths
  - Memory efficiency
  - Can learn with black-box modules inserted between layers
  - Biologically plausible
- Weaknesses
  - Dead/ Alive units persist
  - Defining negative data
- Question
  - Goodness는 도대체 무엇을 말하고 있는가 ACC와 같은 것과 어떻게 연관이 되는가

Outline

Boltzmann Machine (BM)

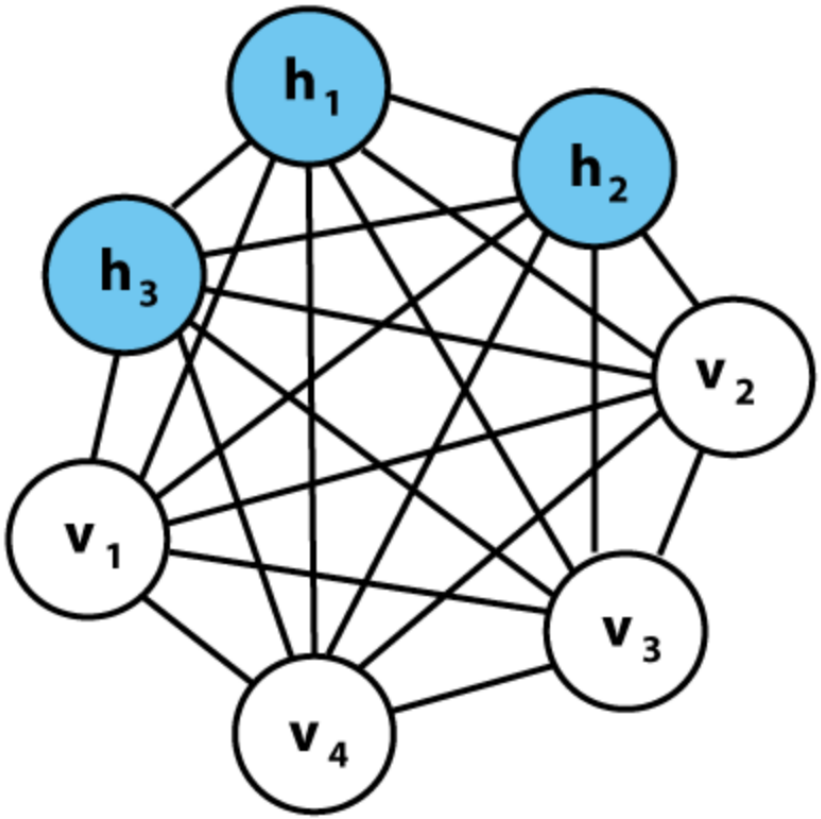


Figure: BM Architecture

- A BM is a network of binary neurons with symmetric pairwise connections.
- Neurons are categorized into **visible**(v1,.,v4) and **hidden** units(h1,h2,h3).

Objective

- Observations: $v_1, \ldots, v_n \overset{i.i.d.}{\sim} g(v)$ (unknown data distribution).
- Goal: fit an energy-based model $p_\theta(v)$ that approximates $g(v)$, i.e.,

$$\theta^* = \arg\max_\theta \mathbb{E}_g[\log p_\theta(V)] \;=\; \arg\min_\theta \mathrm{KL}(g \,\|\, p_\theta)\,.$$

- Define a joint distribution via an energy function:

$$p_\theta(v,h) = \frac{1}{Z(\theta)} \exp\{-E_\theta(v,h)\}, \qquad p_\theta(v) = \sum_h p_\theta(v,h),$$

where $Z(\theta) = \sum_{v,h} \exp\{-E_\theta(v,h)\}$.

- Energy function:

$$E_\theta(v,h) = -\theta_1^\top v - \theta_2^\top h - \frac{1}{2}\, v^\top \theta_3 v - \frac{1}{2}\, h^\top \theta_4 h - v^\top \theta_5 h,$$

where $\mathrm{diag}(\theta_3) = \mathrm{diag}(\theta_4) = 0$.

How to estimate θ^*: Positive vs. Negative Phase

- For an energy-based model

$$p_\theta(v,h) = \frac{1}{Z(\theta)} \exp\{-E_\theta(v,h)\}, \qquad p_\theta(v) = \sum_h p_\theta(v,h),$$

the log-likelihood is

$$\log p_\theta(v) = \log \sum_h \exp\{-E_\theta(v,h)\} - \log Z(\theta).$$

- The gradient decomposes into two expectations:

$$\nabla_\theta \log p_\theta(v) = -\mathbb{E}_{p_\theta(h|v)}[\nabla_\theta E_\theta(v,h)] + \mathbb{E}_{p_\theta(v,h)}[\nabla_\theta E_\theta(v,h)]\,.$$

  - **Positive phase** ($p_\theta(h \mid v)$): hidden units is explained by real data.
  - **Negative phase** ($p_\theta(v,h)$): hidden units is explained by model generated data.
- We approximate these expectations using Gibbs sampling.

Relation to Forward Forward Algorithm

- **Shared idea (contrastive learning):** increase score (decrease energy) on **positive** data and decrease score (increase energy) on **negative** data.
- **FF:** defines a layer-wise "goodness" G and trains with two forward passes: maximize G on positive data and minimize G on negative data (equivalently, $E = -G$).
- **Key difference:** FF avoids MCMC by constructing negatives externally and doing **local, layer-wise updates**.

Outline

Derivation: gradient as two expectations

- **Model:**

$$p_\theta(v,h) = \frac{1}{Z(\theta)} e^{-E_\theta(v,h)}, \qquad p_\theta(v) = \sum_h p_\theta(v,h) = \frac{\sum_h e^{-E_\theta(v,h)}}{Z(\theta)}.$$

Hence

$$\log p_\theta(v) = \log \sum_h e^{-E_\theta(v,h)} - \log Z(\theta).$$

- **Differentiate the data term (log-sum-exp):**

$$\nabla_\theta \log \sum_h e^{-E_\theta(v,h)} = \frac{\sum_h e^{-E_\theta(v,h)}(-\nabla_\theta E_\theta(v,h))}{\sum_h e^{-E_\theta(v,h)}} = -\mathbb{E}_{p_\theta(h|v)}[\nabla_\theta E_\theta(v,h)].$$

- **Differentiate the normalizer:**

$$\nabla_\theta \log Z(\theta) = \frac{\sum_{v,h} e^{-E_\theta(v,h)}(-\nabla_\theta E_\theta(v,h))}{\sum_{v,h} e^{-E_\theta(v,h)}} = -\mathbb{E}_{p_\theta(v,h)}[\nabla_\theta E_\theta(v,h)].$$

Restricted Boltzmann Machine (RBM)

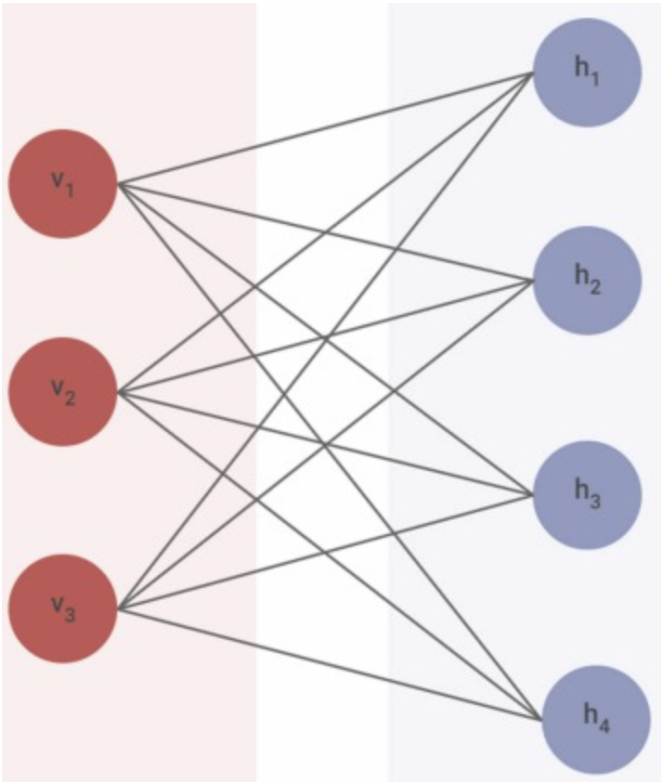


Figure: RBM Architecture

- 차이1 구조적 차이 비저블 유닛, 히든 유닛간의 연결이 없다
- 차이2 차이1로 인해, 에너지 함수가 달라진다.
- 차이3 차이1로 인해, 분포 구하는 과정이 달라진다.

Gibbs sampling

- **Settings.** Let

$$s = \begin{bmatrix} v \\ h \end{bmatrix} \in \{0,1\}^D, \qquad s_{-i} := (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_D).$$

$$p(s) = \frac{1}{Z}\exp\{-E(s)\}, \qquad E(s) = -b^\top s - \frac{1}{2}s^\top W s$$

  where $W = W^\top$, $\mathrm{diag}(W) = 0$.
- **Conditional Probability** For $i = 1, \ldots, D$,

$$p(s_i = 1 \mid s_{-i}) = \sigma\Big(b_i + \sum_{j \neq i} W_{ij} s_j\Big)$$

Gibbs Sampling

Gibbs updates (binary units)

- **General BM (single-site):**

$$p_\theta(s_i = 1 \mid s_{-i}) = \sigma\left(b_i + \sum_{j \neq i} w_{ij} s_j\right), \qquad s = (v, h).$$

- **RBM (block Gibbs; factorizes):**

$$p_\theta(h_j = 1 \mid v) = \sigma\left(c_j + \sum_i W_{ij} v_i\right),$$

So, $p(h = 1|v) = \prod_j p(h_j = 1|v)$