Machine Learning

Prof. Sergei Gleyzer

Lecture

PH451, PH551

April 8, 2025

Announcements

- **Midterm on Thursday**

Mini-Hackathon #2

- **Overall, a great job**
  - **Assignment scores reflect this**
- **Teams tried a variety of ideas for (mostly CNN) image models:**
  - **Dropout, Batch Normalization, ResNet, different activation functions**

Hackathon

- **Special team prize:**
  - **1st place overall +3 grade % points**
  - **2nd place overall +2 grade % points**
  - **3rd place overall +1 grade % points**

Group Presentations

15 minutes total (12 minutes talk + 3 min Q/A)

Scoring Rubric:

- Introduce the topic (10 pts)
  - What is the question you are trying to answer
  - Why is it important
  - Previous approaches/Why ML would help
- Machine Learning (10 pts)
  - Which technique and why
  - Model, training, hyperparameters
  - Results and conclusions
- Overall Impression (5pts) + Time Management (5pts)

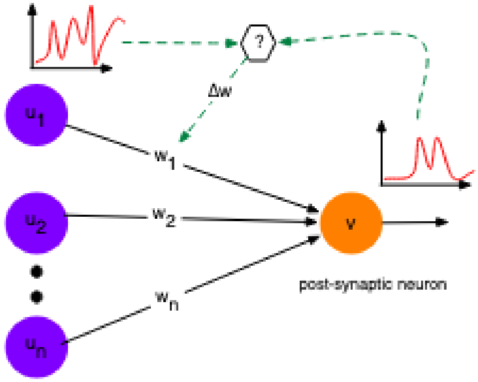
Energy Models

Energy Models

Key idea: minimize **energy** instead of **error**

- Compute **energy gradients**
- Learning = adjust **weights** to reduce energy

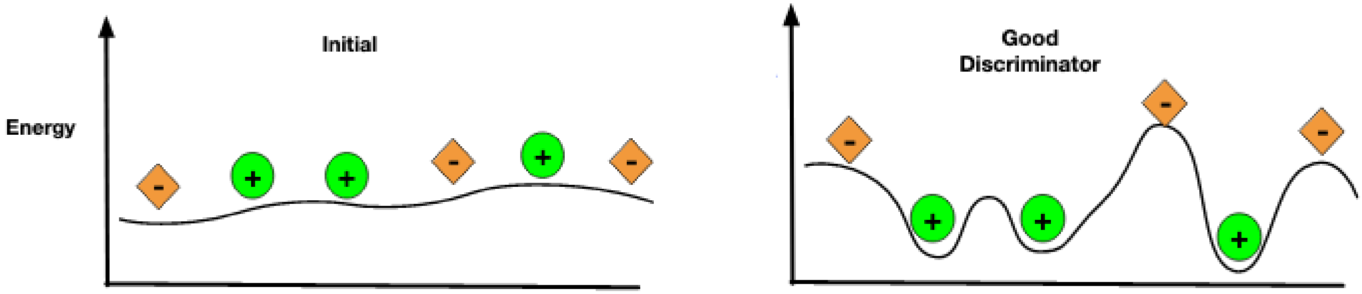
- “**Energy**” can take many forms
  - One common idea - learning is **Hebbian**
  - **Hebb Rule:** “Fire together, wire together”



Energy Models

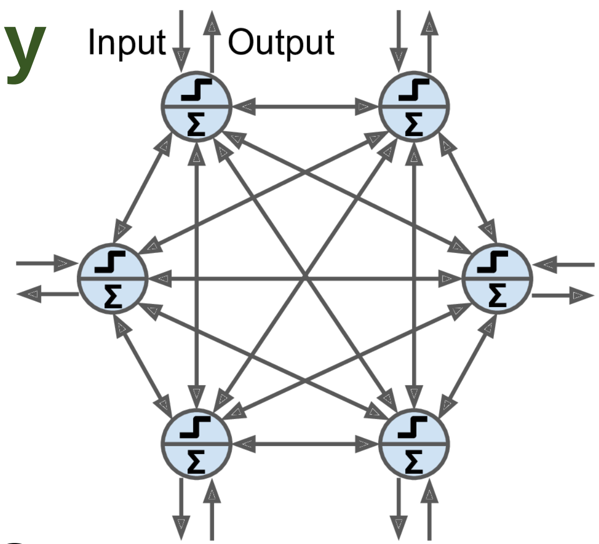
Every instance has an **energy**

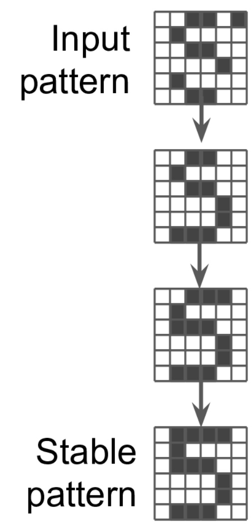
- based on its induced “neural activity”
- Learning = adjusting **weights** such that target data produces energy minima



Hopfield Networks

- Little, 1974 (Hopfield 1982)
- **Associative memory** networks
  - Developed for character recognition
  - **Fully connected** graph
  - Decrease **energy function**
    - Spin-glass theory

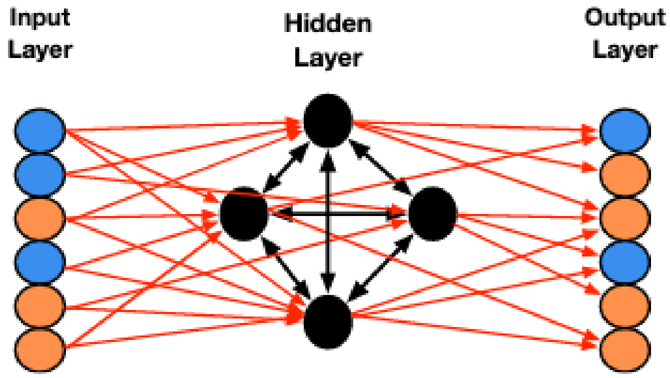




  - **Issues:** scaling, spurious patterns

Boltzmann Machines

- Hinton, Sejnowski (1985)
- **Fully-connected** networks
  - Hidden Layer
  - **Stochastic neurons**
    - Output 0 or 1 with probability based on the **Boltzmann distribution** from statistical mechanics



Boltzmann Machines

- Probability based on the **Boltzmann distribution**:

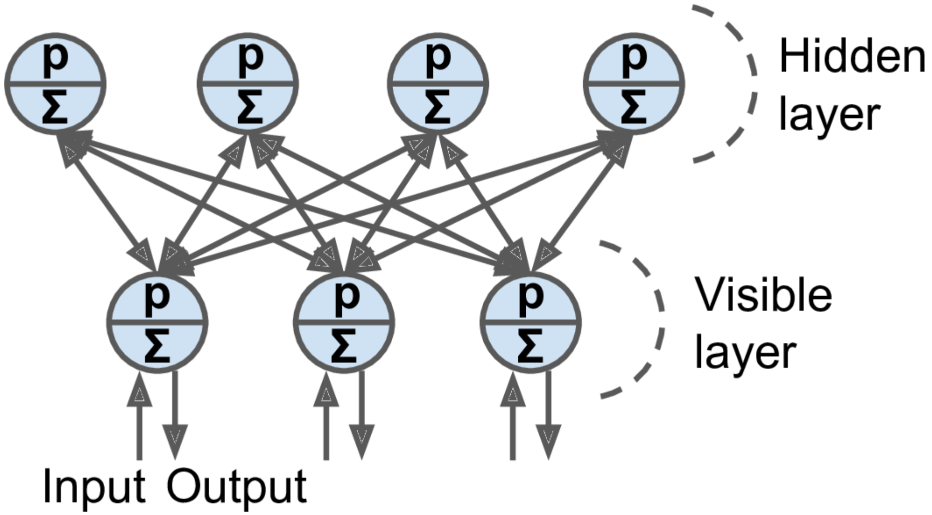
$$p\left(s_i^{(\text{next step})} = 1\right) = \sigma\left(\frac{\sum_{j=1}^{N} w_{i,j}s_j + b_i}{T}\right)$$

  - s_j = j^{th} neuron's **next state**
  - w_{ij} = **connection** weight between i,j
  - b_i = i^{th} neuron's **bias** term
  - N = number of neurons
  - T = **temperature** (higher → more random the output)
  - σ = logistic fn

Reach **Thermal Equilibrium**: **Generative model**
many possible probability distributions

Restrictive Boltzmann Machines

- Smolensky (1986)
- **Boltzmann Machines** with no connections between **visible-visible** or **hidden-hidden units**



- Don't have to wait for **thermal equilibrium**, can be stacked (**Deep Belief Networks**)