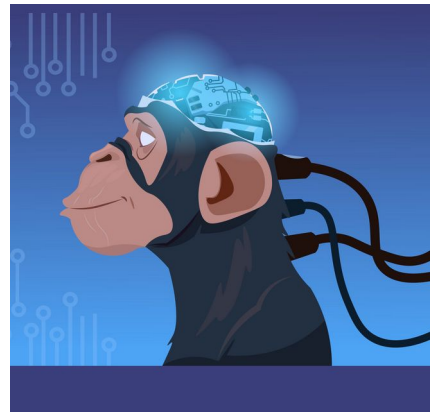
"Statistical mechanics approach for deep-belief neural networks exploration."

Rudamenko Roman

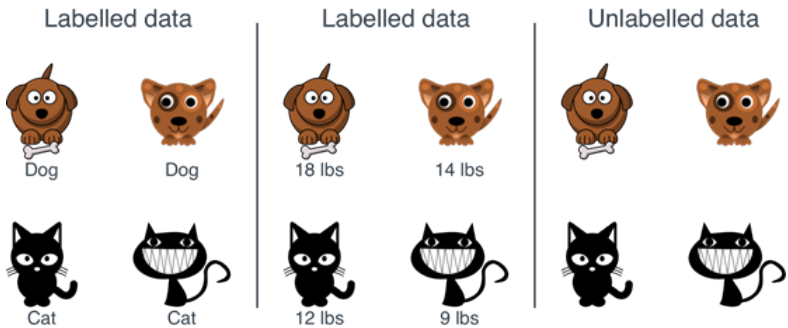
MSU named after M.V. Lomonosov

Use of neural networks

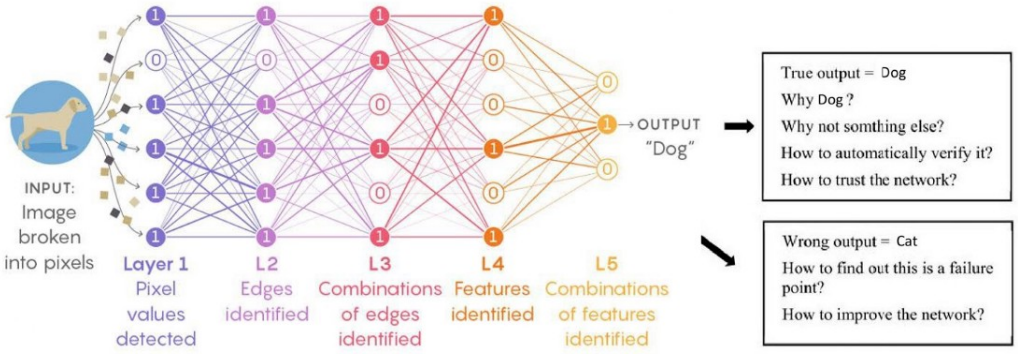
- Synthesis from input data
  - Voice assistants
  - Time-series forecasting
- Pattern recognition
  - Computer vision
  - Anomaly detection
- Data preprocessing
  - Escaping from dimensional curse



Data representation

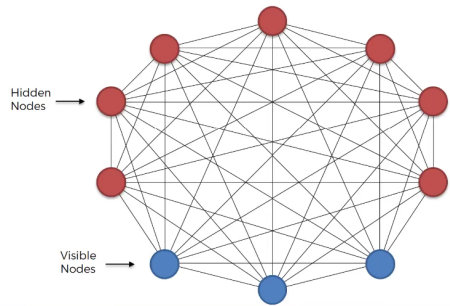


Solution steps



Boltzmann Machine

- Energy-based stochastic neural network
- Base unit for deep-belief network
  - Infer hidden representations
  - Solve combinatorial problems
- Identical to spin-glasses

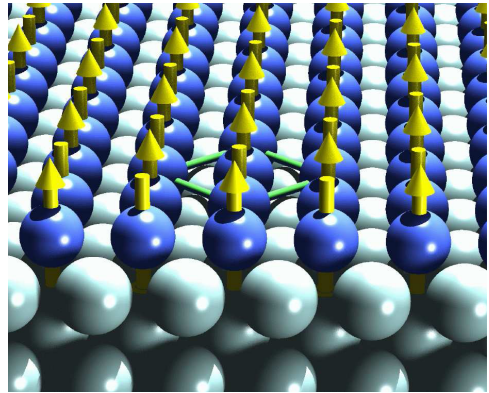


Spin Glasses

- Edwards-Anderson model

$$H = -\sum_{<ij>} J_{ij}\sigma_i\sigma_j$$

- $\sigma_i = \{\pm 1\}$, J_{ij}- gaussian real values



Spin glass visualization

Quenched disorder

- $J = const$, describes system's physical properties
- Free energy:

$$F_N(J) = -\frac{1}{\beta N} \log \int D\sigma e^{-\beta H(\sigma;J)}$$

- Must be free from J J

$$F = -\lim_{N\to\infty} \frac{1}{\beta N} \overline{\log Z(J)} = F_\infty(\beta)$$

Self-averaging

- $N \to \infty$ - self-averaging
- Replica trick:

$$\overline{\log Z} = \lim_{n \to 0} \frac{1}{n} \log \overline{Z^n}$$

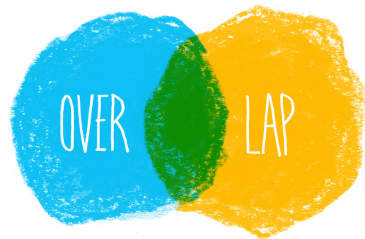
where n - amount of system copies.

Self-averaging over chaos

Similarity measure

- Overlap

$$q_{\sigma\tau} = \frac{1}{N}\sum_{i=1}^{N}\sigma_i\tau_i$$



- Ergodicity breaking — pure states

$$q_{\alpha\beta} = \frac{1}{N}\sum_{i=1}^{N}\langle\sigma_i\rangle_\alpha\langle\sigma_i\rangle_\beta = \frac{1}{Z_\alpha Z_\beta}\int_{\sigma\in\alpha}\int_{\tau\in\beta} e^{-\beta H(\sigma)}e^{-\beta H(\tau)}q_{\sigma\tau}D\sigma D\tau$$

Pure states

Clustering - the main property :

$$\langle \sigma_i \sigma_j \rangle \to \langle \sigma_i \rangle \langle \, \sigma_j \rangle$$
$$|i - j| \to \infty$$

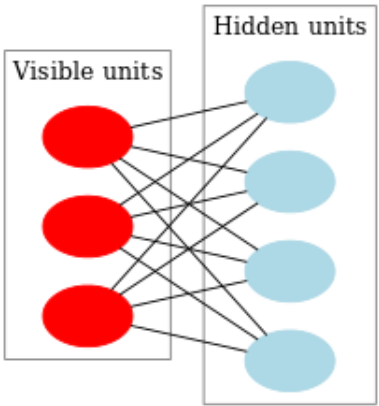
Thouless-Anderson-Palmer free energy: $F(m_i)$.
Infinity-dimensional Ising model: $m = \tanh(\beta m)$

Essence identity

- $S = -\sum_{i=1}^{n} P_i \ln P_i$ in case $E = const$
- Each spin linked to each other - full-dense graph
- Iteraction -only between hidden and visible full-dense bi-partie graph

$$H = -\sum_{i=1}^{N}\sum_{j=1}^{P} \xi_{ij}\sigma_i h_j - \sum_{i=1}^{N} b_i^{(v)}\sigma_i - \sum_{j=1}^{P} b_i^{(h)} h_j$$

Restricted Boltzmann Machine



- Excluding external fields for revealing hidden distribution ξ_i . Obtained configuration energy: $E = -\sum_{i=1}^{N} \xi_i \sigma_i h.$

- From inferred vector ξ_i data with joint probability σ and h is created: $P(\sigma; h) \propto e^{\frac{-\beta E(\sigma;h)}{\sqrt{N}}}$.

K-step contrastive divergence

Algorithm 1. k-step contrastive divergence

Input: RBM $(V_1, \ldots, V_m, H_1, \ldots, H_n)$, training batch S

Output: gradient approximation Δw_{ij}, Δb_j and Δc_i for $i = 1, \ldots, n$, $j = 1, \ldots, m$

1 init $\Delta w_{ij} = \Delta b_j = \Delta c_i = 0$ for $i = 1, \ldots, n$, $j = 1, \ldots, m$
2 **forall the** $\boldsymbol{v} \in S$ **do**
3 $\boldsymbol{v}^{(0)} \leftarrow \boldsymbol{v}$
4 **for** $t = 0, \ldots, k-1$ **do**
5 **for** $i = 1, \ldots, n$ **do** sample $h_i^{(t)} \sim p(h_i \,|\, \boldsymbol{v}^{(t)})$
6 **for** $j = 1, \ldots, m$ **do** sample $v_j^{(t+1)} \sim p(v_j \,|\, \boldsymbol{h}^{(t)})$
7 **for** $i = 1, \ldots, n$, $j = 1, \ldots, m$ **do**
8 $\Delta w_{ij} \leftarrow \Delta w_{ij} + p(H_i = 1 \,|\, \boldsymbol{v}^{(0)}) \cdot v_j^{(0)} - p(H_i = 1 \,|\, \boldsymbol{v}^{(k)}) \cdot v_j^{(k)}$
9 $\Delta b_j \leftarrow \Delta b_j + v_j^{(0)} - v_j^{(k)}$
10 $\Delta c_i \leftarrow \Delta c_i + p(H_i = 1 \,|\, \boldsymbol{v}^{(0)}) - p(H_i = 1 \,|\, \boldsymbol{v}^{(k)})$

Bayessian inference in case of one neuron

Feature importance:

$$\beta = T^{-1}.$$

Distribution σ with Bayes theorem:

$$P(\sigma|\xi) = \frac{\cosh \frac{\beta}{\sqrt{N}} \xi^T \sigma}{\sum_\sigma \cosh \frac{\beta}{\sqrt{N}} \xi^T \sigma}$$

Aposteriorial distribution $\{\sigma_a\}_{a=1}^M$,:

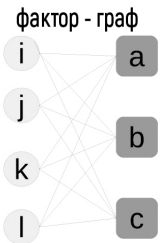
$$P(\xi|\{\sigma^a\}) = \frac{\prod_a P(\{\sigma^a\}|\xi)}{\sum_\xi \prod_a P(\{\sigma^a\}|\xi)} = \frac{1}{Z} \prod_a \cosh \frac{\beta}{\sqrt{N}} \xi^T \sigma^a$$

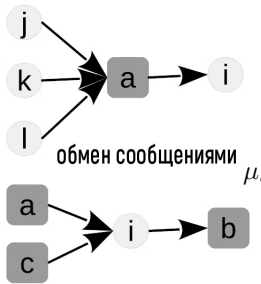
Learning algorithm

- EM - algorithm
  - **E - step**: calculate value of h from approximation ξ at current step.
  - **M - step**: maximize overlap $q = \frac{1}{N}\sum_i \xi_i^{true}\hat{\xi}_i$, $\hat{\xi}_i = argmax_{\xi_i} P_i(\xi_i)$ and find next approximation $\hat{\xi}_i$

- Overlap interpretation
  - $q = 0$: initial vector doesn't contain targer
  - $q = 1$: purely describes

Message passing algorithm

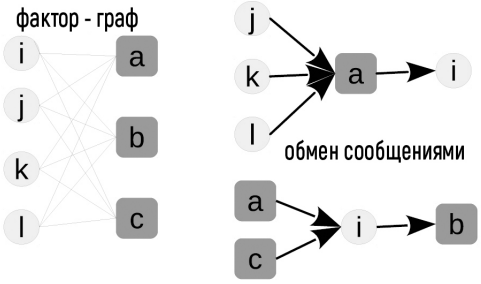




$$P_{i\to a}(\xi_i) \propto \prod_{b\in\partial\setminus a} \mu_{b\to i}(\xi_i)$$

$$\mu_{b\to i}(\xi_i) = \sum_{\{\xi_j | j\in\partial b\setminus i\}} \cosh\left(\frac{\beta}{\sqrt{N}}\xi^T\sigma^b\right) \prod_{j\in\partial b\setminus i} P_{j\to b}(\xi_j)$$

Iteratively message passing



$$m_{i\to a} = \tanh\left(\sum_{b\in\partial i\setminus a} u_{b\to i}\right)$$

$$u_{b\to i} = \tanh^{-1}\left(\tanh(\beta G_{b\to i})\tanh(\beta\sigma_i^b/\sqrt{N})\right)$$

$$G_{b\to i} = \frac{1}{N}\sum_{j\in\partial b\setminus i}\sigma_j^b m_{j\to b}$$

$$m_{j\to b} = \sum_{\xi_j}\xi_j P_{j\to b}(\xi_j)$$

$$P_i(\xi_i) = \frac{1+m_i\xi_i}{2}$$

Simplification message passing

Thouless-Anderson-Palmer equation:

$$Q \equiv \frac{1}{N}\sum_i m_i^2$$

$$G_a^{t-1} = \frac{1}{\sqrt{N}}\sum_{i\in\partial a}\sigma_i^a m_i^{t-1} - \beta(1-Q^{t-1})\tanh\beta G_a^{t-2}$$

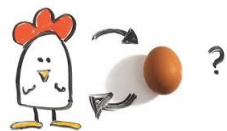
$$m_i^t \simeq \tanh\left(\sum_{b\in\partial i}\frac{\beta\sigma_i^b}{\sqrt{N}}\tanh\beta G_b^{t-1} - \frac{\beta^2 m_i^{t-1}}{N}\sum_{b\in\partial i}(1-\tanh^2\beta G_b^{t-1}\right).$$

Calculation complexity $O(M+N)$.

Learning subset creation

- Approx partition function with means:

$$s = \sum_x p(x) f(x) = E_p[f(x)]$$



- Energy model:
  - x_0 — random choice
  - $T(x^{'} \mid x)$ — transiton distribution

For choice a sampling from из $p(a \mid b)$,
For choice b – sampling from $p(b \mid a)$

Formalization

- Distribution parametrization:

$$g(x = i) = v_i.$$

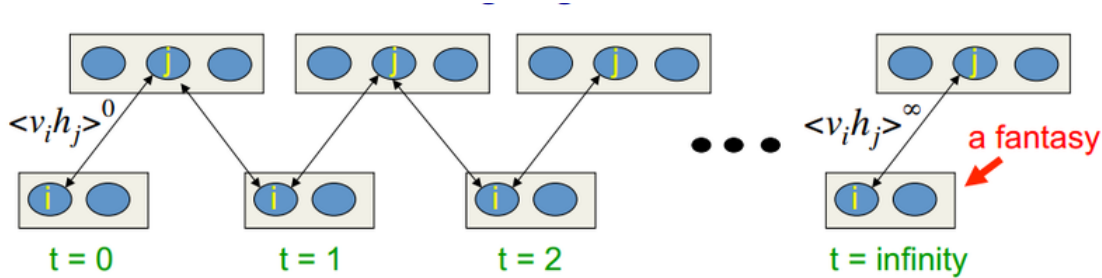
- New state probability x' with current x:

$$g^{(t+1)}(x') = \sum_x q^{(t)}(x) T(x \mid x)$$

- Matrix representation:

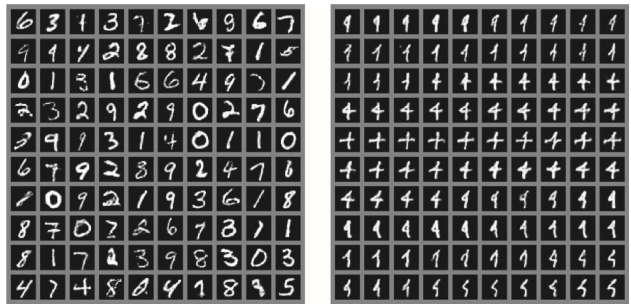
$$A_{i,j} = T(x = i \mid x = j), v^{(t)} = Av^{(t-1)},$$
$$v^{(t)} = A^t v^{(0)}.$$

Gibbs sampling



Tempering

- Fast Mode mixing with $\beta < 1$.
- Efficently in area of slow temperature transition.



Nearest neighbours in RBM and GAN

Data temperature measuring

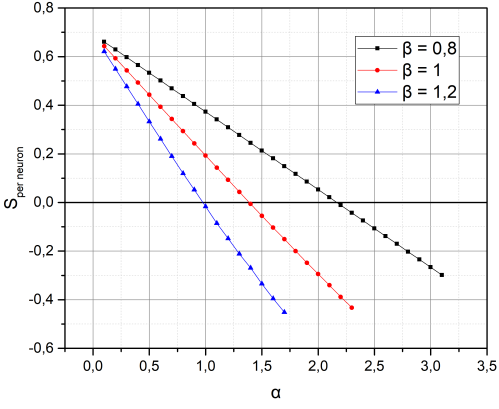
- Bayes theorem once again

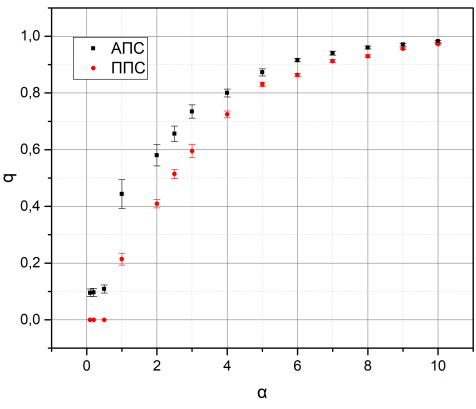
$$P(\beta|\{\sigma^a\}) = \sum_{\xi} P(\beta, \xi|\{\sigma^a\}) \propto e^{-M\frac{\beta^2}{2}} Z(\beta, \{\sigma^a\})$$

- Equation for β should satisfy

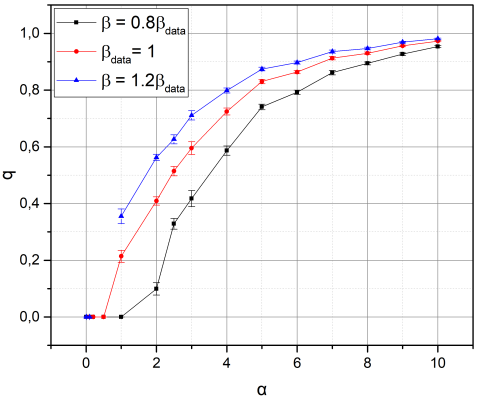
$$\frac{\partial \ln Z(\beta|\{\sigma^a\})}{\partial \beta} = N\alpha\beta$$

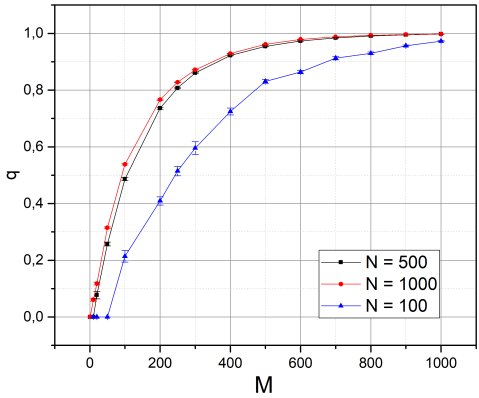
Entropy and perfomance





Choosing size and temperature





Temperature measuring and minima search