The Potential Energy of an Autoencoder

Cunxiao Du

2020-07-02

Overview

- The Potential Energy of an Autoencoder
  - *Autoencoders are popular feature learning models, that are conceptually simple, easy to train and allow for efficient inference and training. This paper shows how certain autoencoders can be associated with an energy landscape, akin to negative log-probability in a probabilistic model, which measures how well the autoencoder can represent regions in the input space.*
    - The Potential Energy of an Autoencoder(TPAMI)

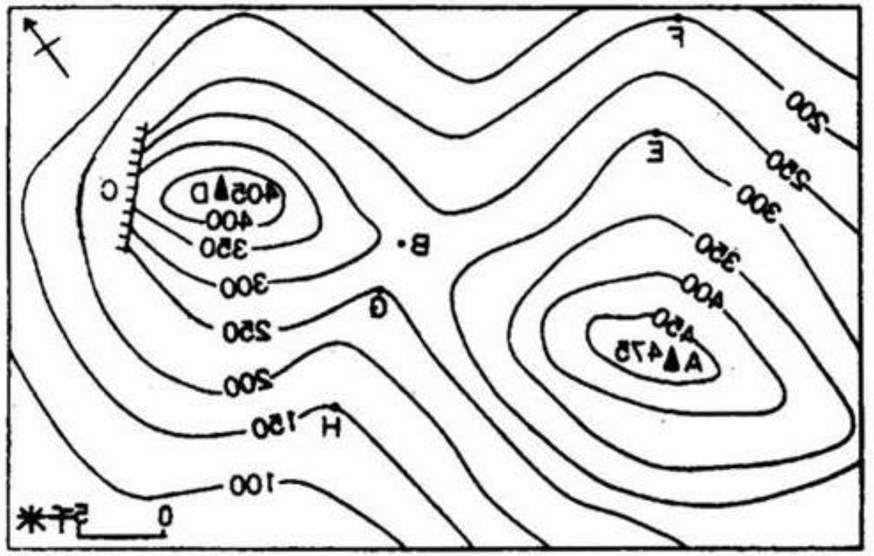
Outline

- Energy of AE (Cunxiao Du)
  - The Potential Energy of an Autoencoder(TPAMI)

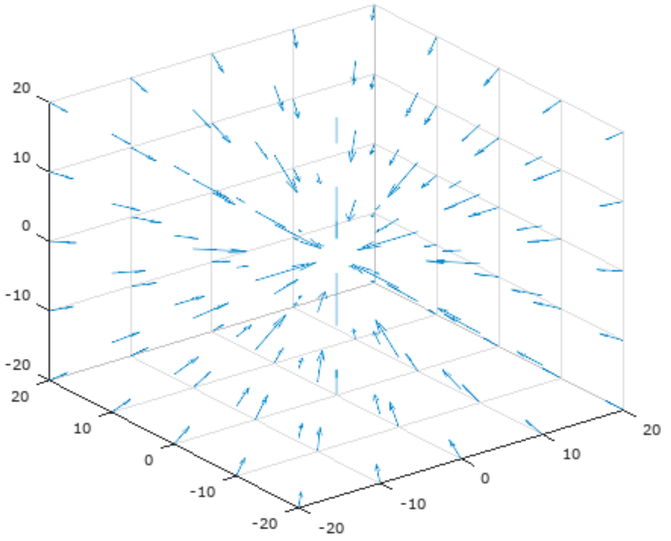
Energy Model

- $$p(x) = \frac{e^{-E(x)}}{\sum_{x'} e^{-E(x')}} = \frac{e^{-E(x)}}{Z(\theta)}$$
- Each x has an unnormalized score called energy E(x), usually $Z(\theta)$ (partition function) is hard to compute.
- Energy model satisfies the minimum entropy principle.
- Score Matching: $\nabla_x \log p(x) = \nabla_x \log \frac{e^{-E(x)}}{Z(\theta)} = -\nabla_x E(x)$

Filed Theory



- Scalar Filed
  - Each Point has a scalar value.
  - $\log p(x)$ defines a scalar filed.
- Vector Filed
  - Each point has a vector value.
  - Divergence (div): Estimate the outward flux.
  - Sink point: div P < 0
  - Source point: div P > 0
- Gradient Filed
  - The gradient of a scalar filed could construct a Vector filed.
  - Score matching is trying to learning the gradient filed of the energy model defines filed.



AutoEncoder for Score Matching

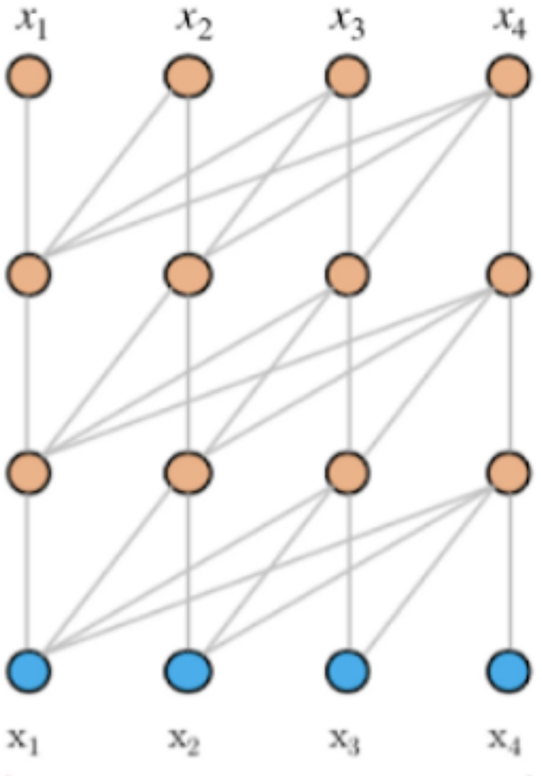
- Input = x, using a network r to reconstruct it, output = $r(x)$.
- Loss = $|r(x) - x|^2$.
- $r(x) - x$ defines a vector filed, we want it close to $\nabla_x \log p(x)$.
- Intuition of the AutoEncoder behavior: keep the training data as the fix point so that $\nabla_x \log p(x) = 0$
- So we could use gradient ascent to sampling $argmax_x \log p(x)$ from this vector filed:
  - Initialize x;
  - $x = x + \lambda * \nabla_x \log p(x)$;
- Intuition of the AutoEncoder behavior: Keep the training data as the fix point.

Restricted AutoEncoder

- Vanilla autoencoder ease to learn the identity map, turns all the data as fix point.
- Intuition: Training data not only be encouraged to be a fix point, but also to be the center of a basin of attraction (sink point), keep div P < 0.
- CAE Loss = $|r(x) - x|^2 + \lambda * |\nabla_x r(x)|_F^2$
- DAE Loss = $|r(x + \epsilon) - x|^2, \epsilon \sim N(0, \sigma^2)$
- Restrict the autoencoder less sensitive to the input.

Transformer as CAE

- Transformer is a special AutoEnocoder.
- Training: Given input, and reconstruct the input.
- Sampling:
  - Initialize x from a noise distribution.
  - $x \sim transformer(x)$
  - Until convergence.



Transformer as CAE

- AT $\begin{bmatrix} \frac{\partial r(x)_1}{\partial x_1} & \cdots & \frac{\partial r(x)_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r(x)_n}{\partial x_1} & \cdots & \frac{\partial r(x)_n}{\partial x_n} \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \frac{\partial r(x)_n}{\partial x_1} & \cdots & 0 \end{bmatrix}$

- Restrict the jacobian matrix as a lower triangular matrix.

Brain Storm

- New generative model: directly minimize the jacobian of transformer.
- New sampling process: use some intuitions from gradient descent like Adam to make sampling better.
- The property of lower triangular jacobian matrix.