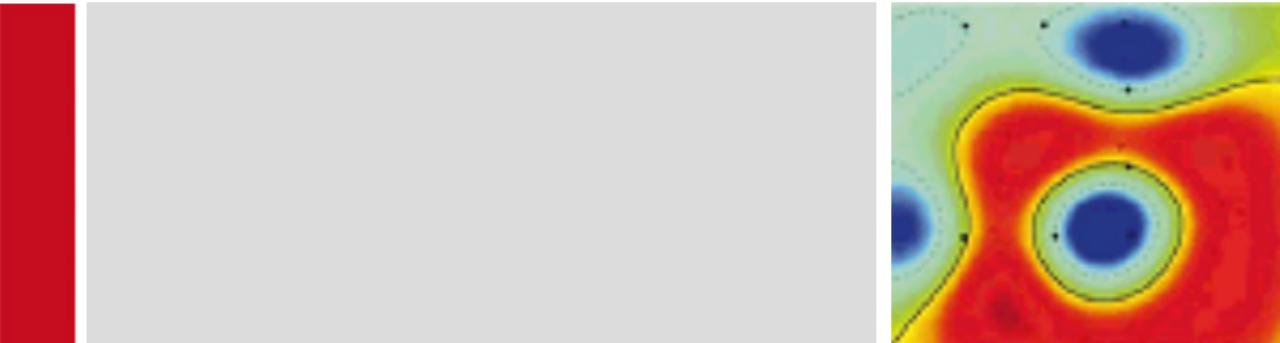


WiSe 2024/25

Deep Learning 1



Lecture 11

Structured Prediction and EBMs

Outline

Practical Motivations

- ▶ Limitation of standard neural networks

Mixture Density Networks

- ▶ Formulation
- ▶ Advantages and limitations

Conditional Restricted Boltzmann Machines

- ▶ Formulation
- ▶ Advantages and limitations

Energy-Based Learning

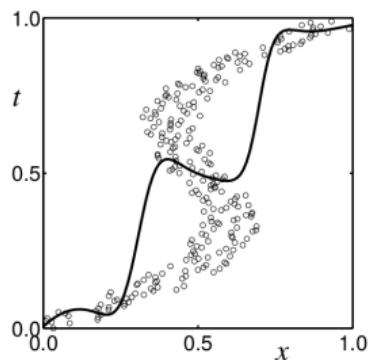
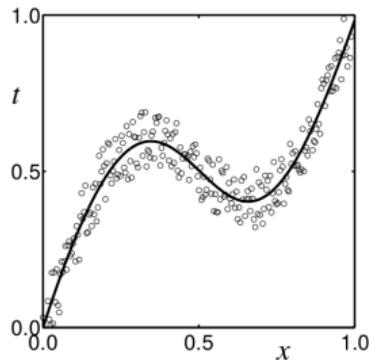
- ▶ Training Procedure and Loss Functions
- ▶ Obtaining Contrastive Examples

Learning with Structured Output

- ▶ Neural networks are generally trained to predict simple quantities such as real values, or class membership.
 - ▶ In some cases, e.g. machine translation and question-answering systems, it is necessary to predict more complex structures, e.g. a sequence.
 - ▶ These structured objects are subject to constraints (e.g. local consistency in the sequence, or guided by some underlying dynamics).
 - ▶ Many methods have been proposed for structured output in the context of neural networks (e.g. sequence-to-sequence models, mixture density networks, energy-based learning).
1. 神经网络通常被训练来预测简单的量，例如实数值或类别标签。
2. 在某些情况下（例如机器翻译和问答系统），需要预测更为复杂的结构，例如一个序列。
3. 这些结构化对象通常受到约束（例如在序列中需要保持局部一致性，或受到某种潜在动力学的引导）。
4. 在神经网络背景下，为了进行结构化输出，已经提出了许多方法（例如序列到序列模型、混合密度网络、基于能量的学习等）。

Forward vs. Inverse Problems

Consider now another aspect of structured output which is how to deal with cases where two or more distinct predictions are equally plausible. This often occurs when learning inverse problems.



- ▶ Structure of the target labels
 $t = f(x) + \text{noise}$.
- ▶ Standard neural networks can be used.

现在再来看结构化输出的另一个方面：当两种或多种不同预测同样合理时如何处理。这种情况在学习逆问题时经常出现。

- 目标标签的结构: $t = f(x) + \text{噪声}$ 。
- 可以使用标准的神经网络。
- 问题: f 并不可逆。
- 对于给定输入, 我们需要产生多个输出 (或最可能的输出)。

- ▶ Problem: f is not invertible.
- ▶ For a given input, we need to produce several outputs (or the most likely output).

当我们说「 f 不可逆」 (f not invertible) 时, 指的是给定输出值时无法唯一地确定对应的输入值。换言之, 多个不同的输入 x 可能会映射到同一个输出 t , 导致无法从 t 唯一地「反推」出 x 。这类函数也常被称为「多对一」映射 (many-to-one mapping)。

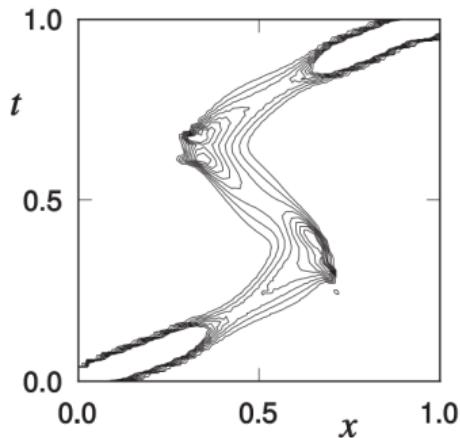
Learning Output Models

1. 学习条件概率模型 $p_\theta(t | x)$

当面对“多对一”或有多种可能输出的情形时，仅输出一个确定值往往无法真实反映数据的分布。

为此，我们希望直接去学习“给定输入 x 时，输出 t 的概率分布”——即条件概率分布 $p_\theta(t | x)$ 。

通过最小化 $D(p(t|x) \parallel p_\theta(t|x))$ ，我们就能使模型产生的分布尽量与真实分布接近。



- ▶ View the task of prediction as that of learning a conditional probability model $p_\theta(t | x)$.
- ▶ Minimize the objective

$$\min_{\theta} D(p(t|x) \parallel p_\theta(t|x))$$

where D is some divergence measure between two distributions (e.g. Kullback-Leibler Divergence or Wasserstein distance)

D 是衡量两个分布之间差异的散度（例如 KL 散度或者 Wasserstein 距离）。

Mixture Density Networks (MDNs) [2]

1. Model output as a Gaussian mixture

$$p(\mathbf{t} \mid \mathbf{x}) = \sum_{i=1}^m \alpha_i(\mathbf{x}) \phi_i(\mathbf{t} \mid \mathbf{x})$$

$$\phi_i(\mathbf{t} \mid \mathbf{x}) = \frac{1}{(2\pi)^{c/2} \sigma_i(\mathbf{x})^c} \exp \left\{ -\frac{\|\mathbf{t} - \boldsymbol{\mu}_i(\mathbf{x})\|^2}{2\sigma_i(\mathbf{x})^2} \right\}$$

where parameters of the mixture are the output of the neural network.

2. Optimize Gaussian mixture's likelihood

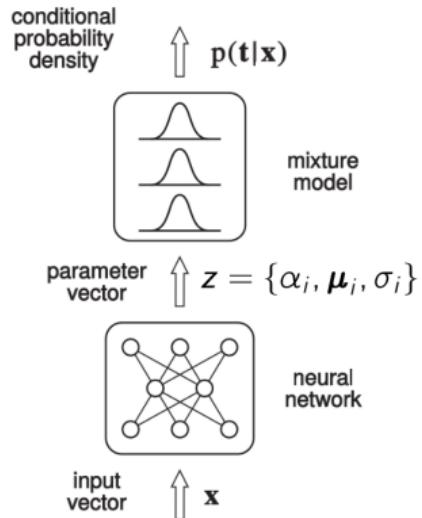
$$E^q = -\ln \left\{ \sum_{i=1}^m \alpha_i(\mathbf{x}^q) \phi_i(\mathbf{t}^q \mid \mathbf{x}^q) \right\}$$

2. 为什么使用混合密度网络 (MDN)

- 很多真实场景下, $p(\mathbf{t} \mid \mathbf{x})$ 可能是多峰 (多模态) 的, 或者形状比较复杂。单一的高斯分布无法满足需求, 而高斯混合模型 (GMM) 可以用多个高斯的加权和去近似任意复杂分布。
- MDN 通过神经网络输出混合模型中的所有参数 (包括每个高斯的均值、方差以及混合权重 α_i 等), 从而可以灵活地根据输入 x 动态调整这些高斯分布的形状与位置。

3. 公式与含义

- $\alpha_i(\mathbf{x})$ 表示混合中第 i 个高斯成分的权重, 总和为 1;
- $\mu_i(\mathbf{x})$ 和 $\sigma_i(\mathbf{x})$ 分别是第 i 个高斯成分的均值和标准差;
- $\phi_i(\mathbf{t} \mid \mathbf{x})$ 则是高斯密度函数本身, 用来描述“对于给定的输入 x , 输出为 t 的概率密度”。这些参数全部由神经网络在前向传播中生成, 网络训练时则通过最大似然 (或最小化散度)



Mixture Density Networks (MDNs) [2]

参数需要满足一定约束，可以通过恰当选择输出层来强制这些约束：

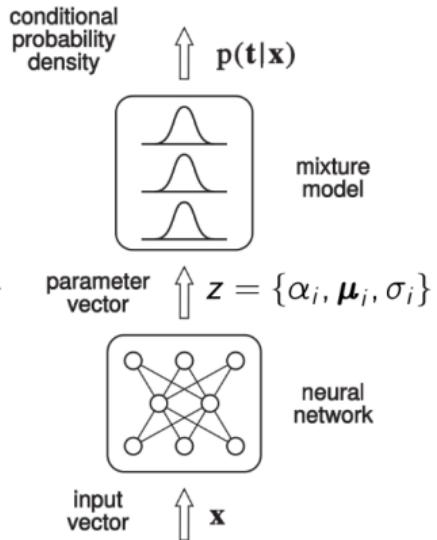
Parameters are subject to constraints.

These constraints can be enforced with an appropriate choice of output layer.

$$\alpha_i = \frac{\exp(z_i^\alpha)}{\sum_{j=1}^M \exp(z_j^\alpha)}$$

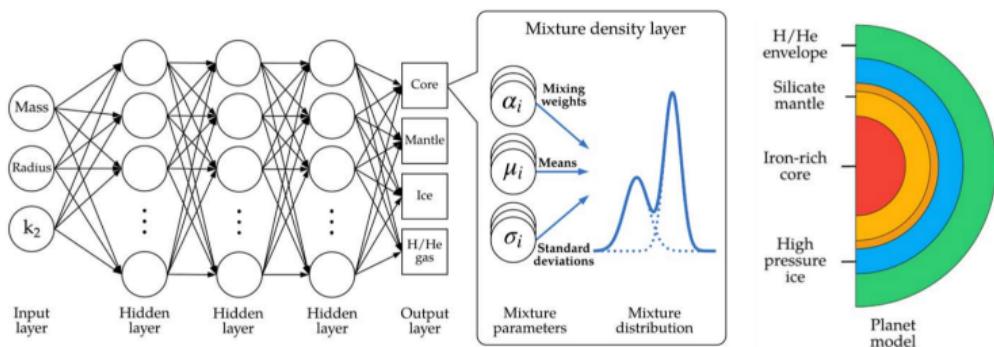
$$\sigma_i = \exp(z_i^\sigma)$$

$$\mu_{ik} = z_{ik}^\mu$$



MDNs for Inferring Planets Composition

- ▶ A dataset of artificial planets is built from physics-based simulations.
- ▶ A MDN model mapping ‘observed’ variables (mass/radius/ k_2) to ‘hidden’ variables (composition) is trained on the dataset [1].



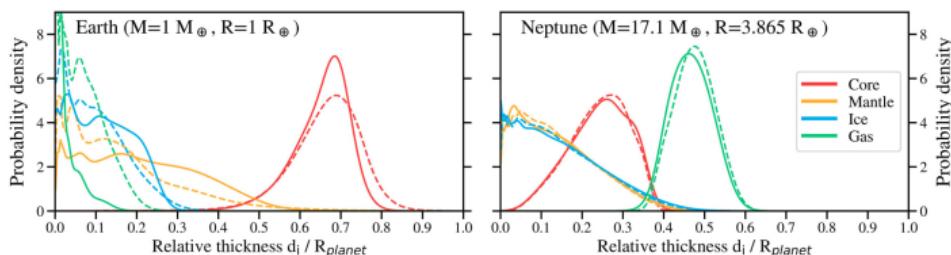
Source: Baumeister et al. (2020), ML Inference of the Interior Structure of Low-mass Exoplanets

用于推断行星成分的混合密度网络 (MDNs)

- 一个由「人造行星」组成的数据集由基于物理的模拟生成。
- 训练了一个 MDN 模型，将“观测”变量（质量 / 半径 / k_2 ）映射到“隐藏”变量（成分），并在该数据集上进行了训练 [1]。

MDNs for Inferring Planets Composition

- ▶ The MDN model can be verified on planets for which the interior structure is known (e.g. planets in the solar system).



Source: Baumeister et al. (2020), ML Inference of the Interior Structure of Low-mass Exoplanets

- ▶ The MDN model can be used to infer the composition of less known exoplanets from a limited number of observations.

- 若行星的内部结构已知（例如太阳系行星），则可用来验证 MDN 模型。
(如图所示，左图是地球 ($M = 1 M_\oplus$, $R = 1 R_\oplus$)，右图是海王星 ($M = 17.1 M_\oplus$, $R = 3.865 R_\oplus$)，分别显示了各个层（核心、地幔、冰层、气体）的相对厚度分布。)
- 该 MDN 模型也可用于在仅有少量观测数据的情况下，推断未知系外行星的成分。

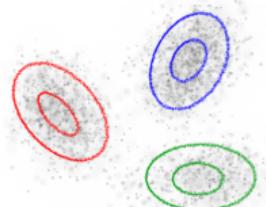
From Mixtures to Boltzmann Machines

- ▶ Gaussian mixture models (GMMs) capture local regions of high density but is not efficient for distributions that are shaped by global effects.
- ▶ Boltzmann machines (BMs) describe the probability function as a product of factors. Each factor captures a global effect in the distribution.

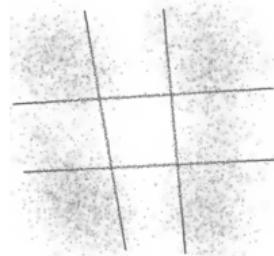
从混合到玻尔兹曼机

- 高斯混合模型（GMM）能捕捉局部的高密度区域，但对依赖全局效应的分布而言效率不高。
- 玻尔兹曼机（BM）通过将概率分布描述为多个因子的乘积来进行刻画，每个因子都能捕捉分布中的全局影响。

GMM



RBM



GMM 局限：高斯混合模型常用于描述局部密度峰分布，但若行星或天体分布存在更复杂的（全局）相互关联或物理规律，GMM 可能难以有效捕捉。

玻尔兹曼机（BM）：通过因子分解的方式对整体概率进行表示；多个因子可以共同刻画分布的全局特性，适合更复杂的物理或相互作用过程。

Conditional RBM

条件受限玻尔兹曼机 (CRBM) 是什么?

- RBM (受限玻尔兹曼机) 本身是一种能量基模型, 通常由可见层和隐藏层组成, 能对联合分布进行建模。
- CRBM 在此基础上加入了对「输入 x 」和「输出 y 」的建模, 把 x 和 y 一并视为可见变量, 并用额外的隐藏变量 h 来捕捉它们之间的依赖关系。
- 通过“条件”一词强调: 我们往往想得到 $p(y | x)$, 即在给定输入 x 后, 对输出 y 的分布进行推断。

能量函数与概率分布

- CRBM 通过一个能量函数 $E(x, h, y)$ 来为三者的组合分配分数: 分数越低 (能量越小) 就意味着出现的概率越高。
- 该概率由 softmax 型的归一化公式 $\exp(-E)$ 给出, 并用 Z 归一化, 保证所有可能组合的概率总和为 1。

A conditional restricted Boltzmann machine [6] is a system of binary variables composed of an input x , and output y and a latent state h

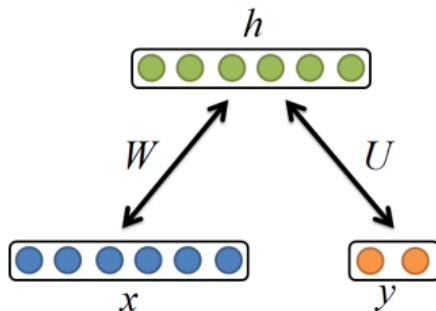


Image from Jing et al. (2014) Ensemble of ML algorithms for cognitive and physical speaker load detection

- ▶ It is governed by an energy function

$$E(\mathbf{x}, \mathbf{h}, \mathbf{y}) = -\mathbf{x}^\top \mathbf{W} \mathbf{h} - \mathbf{y}^\top \mathbf{U} \mathbf{h}$$

- ▶ The system associates to each joint configuration the probability

$$p(\mathbf{x}, \mathbf{h}, \mathbf{y}) = Z^{-1} \exp(-E(\mathbf{x}, \mathbf{h}, \mathbf{y}))$$

where Z is a normalization term.

Conditional RBM, Marginalization

Marginalization can be easily achieved in a CRBM:

$$p(\mathbf{x}, \mathbf{y}) = Z^{-1} \exp(-F(\mathbf{x}, \mathbf{y}))$$

where

$$F(\mathbf{x}, \mathbf{y}) = -\sum_{k=1}^K \log(1 + \exp(W_{:k}^\top \mathbf{x} + U_{:k}^\top \mathbf{y}))$$

is called the free energy and can be interpreted as a two-layer neural network. The lower the free energy the more likely the joint configuration (\mathbf{x}, \mathbf{y}) . Hence, structured prediction can be performed as:

$$\mathbf{y}|\mathbf{x} = \arg \min_{\mathbf{y}} F(\mathbf{x}, \mathbf{y})$$

自由能（Free Energy）与边缘化

- 在 CRBM 中，将隐藏变量 h 积分（或求和）掉就能得到边缘分布 $p(x, y)$ 。
- 自由能 $F(x, y)$ 就是执行这一边缘化过程后得到的对数分布的负值（带负号），可以理解为“把隐藏变量综合考虑进来之后，对 (x, y) 的整体测度”。
- 自由能越低意味着 (x, y) 的匹配程度越高，因此在结构化预测中，直接对 $F(x, y)$ 进行最优化就可以找到最可能的 y 。

Training a Conditional RBM

A common training procedure is to maximize the data likelihood

$$\ell = \frac{1}{N} \sum_{n=1}^N \log p(\mathbf{x}^n, \mathbf{y}^n).$$

Its gradient is given by:

$$\begin{aligned}\frac{\partial \ell}{\partial W_{ik}} &= \mathbb{E}_{\hat{p}(\mathbf{x}, \mathbf{y})} [x_i \cdot \text{sigm}(W_{:k}^\top \mathbf{x})] - \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [x_i \cdot \text{sigm}(W_{:k}^\top \mathbf{x})] \\ \frac{\partial \ell}{\partial U_{jk}} &= \mathbb{E}_{\hat{p}(\mathbf{x}, \mathbf{y})} [y_i \cdot \text{sigm}(U_{:k}^\top \mathbf{y})] - \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [y_i \cdot \text{sigm}(U_{:k}^\top \mathbf{y})]\end{aligned}$$

左侧是相对于真实数据分布 ($\hat{p}(x, y)$) 的期望，右侧是相对于 CRBM 实现的模型分布 ($p(x, y)$) 的期望。

- ▶ The left hand side is a data expectation.
- ▶ The right hand side is an expectation over the probability distribution $p(\mathbf{x}, \mathbf{y})$ implemented by the CRBM. The latter can be approximated using alternate Gibbs sampling (i.e. iteratively sampling from $p(\mathbf{x}, \mathbf{y} | \mathbf{h})$ and $p(\mathbf{h} | \mathbf{x}, \mathbf{y})$).

后者可以通过交替吉布斯采样来近似计算（即在给定 \mathbf{h} 时从 $p(x, y | \mathbf{h})$ 采样，以及在给定 (x, y) 时从 $p(\mathbf{h} | x, y)$ 采样的循环过程中）。

CRBM Variant: Product Interactions

- 在标准 CRBM 中，输入 x 与隐藏层 h 的交互项、输出 y 与隐藏层 h 的交互项是独立相加的，也就是

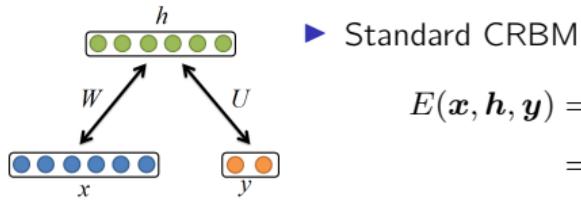
$$E(x, h, y) = -x^\top Wh - y^\top Uh.$$

这种形式只考虑到输入和输出分别与隐藏层的两重关系，却并未直接把输入与输出“耦合”在一起。

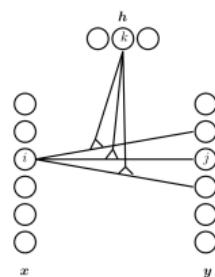
- 在高阶 CRBM 中，则增加了输入、输出与隐藏层三者的**乘积**项，通过一个额外的因素（或者说张量分解）来实现。如果用符号表示就是

$$E(x, h, y) = - \sum_{ijklf} x_i y_j h_k W_{if} U_{jf} V_{kf}.$$

这样能捕捉到输入与输出在同一隐藏单元里更深层次的联合依赖（例如“输入的某些模式与输出的某些模式一同出现时，会显著影响隐藏层”），从而可能得到更灵活的模型。



$$\begin{aligned} E(\mathbf{x}, \mathbf{h}, \mathbf{y}) &= -\mathbf{x}^\top \mathbf{W} \mathbf{h} - \mathbf{y}^\top \mathbf{U} \mathbf{h} \\ &= -\sum_{ik} x_i h_k W_{ik} - \sum_{jk} y_j h_k U_{jk} \end{aligned}$$



Input and output contribute additively to the energy.

- Higher-order factored CRBM [5]

$$E(\mathbf{x}, \mathbf{h}, \mathbf{y}) = - \sum_{ijkf} x_i y_j h_k W_{if} U_{jf} V_{kf}$$

Input and output interact in a multiplicative manner.

Inspecting the CRBM (input filters W)

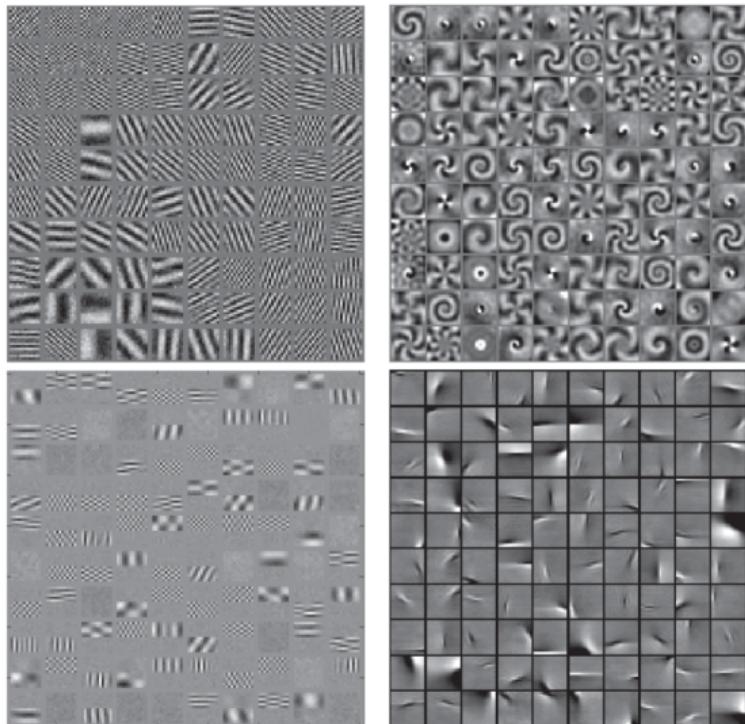


Image from Memisevic et al. (2013) Learning to Relate Images

Inspecting the CRBM (output filters U)

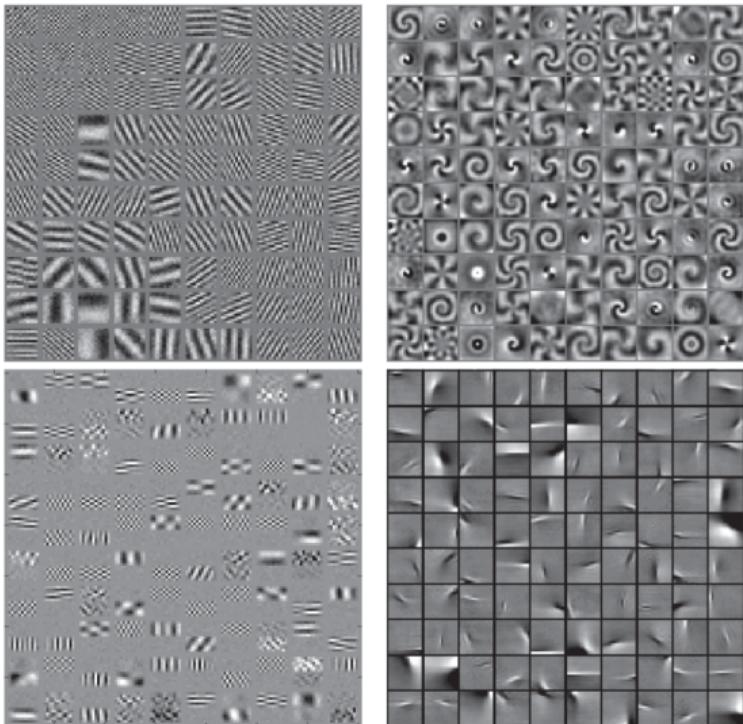
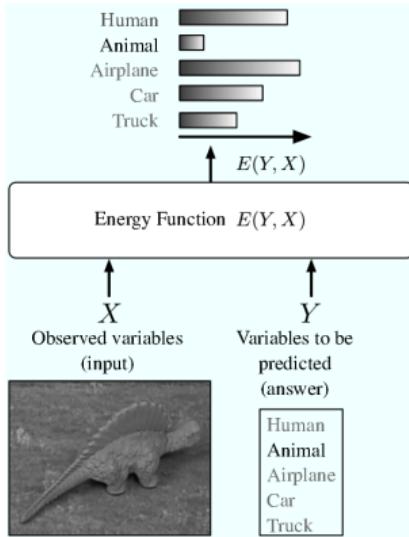


Image from Memisevic et al. (2013) Learning to Relate Images

General Framework: Energy-Based Learning

与基于核的结构化预测方法（如 SVM 结构化预测）类似，基于能量的学习同样对输入与输出对赋予一个标量表示“好坏程度”。不同之处在于，能量基模型通常依赖可微函数（如神经网络）来定义能量，同时可能采用各种训练准则（如对数似然或最大间隔等）。

最终目的都是相同：找到使能量最低（或打分最高）的输出。



- 因为能量函数本质上是对“模型如何评估 (X, Y) 这对组合”进行量化。如果能量低，表示模型认为输入和输出之间匹配度高；若能量高，则模型认为它们不太匹配。
- 在训练中，可以通过对已有的真实数据 (X^n, Y^n) 施加约束或优化目标（如最小化真实对的能量，最大化错误对的能量），来让模型学到合适的能量分配。

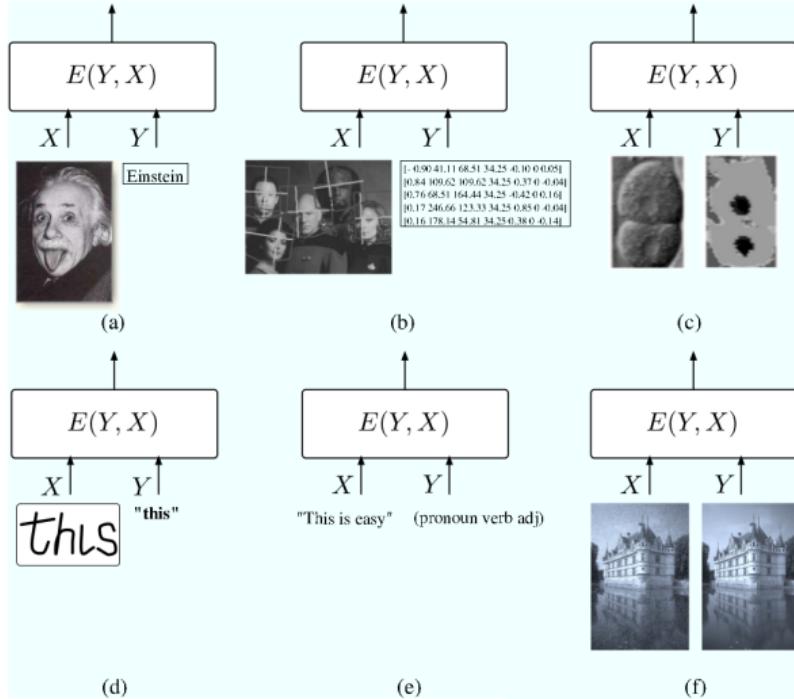
- ▶ Energy-based learning [4] is a general framework for performing structured predictions with neural networks.
- ▶ Like for kernel-based structured prediction, it associates to each input-output pair a score.
- ▶ The predicted output is the one that yields the highest score (lowest energy).
- ▶ Most of the images in the following slides are from the paper:

LeCun et al. (2006) A Tutorial on Energy-Based Learning

通用框架：基于能量的学习 (Energy-Based Learning)

- 基于能量的学习 [4] 是一种通用的框架，可用于在神经网络中执行结构化预测。
- 类似于基于核的方法，它为每个输入-输出对分配一个评分（分数）。
- 预测输出则是得到最高评分（即最低能量）的那个候选。

Applications of Energy-Based Learning



语义分割：输入是图像像素，输出是各像素所属的语义类别（如图 (c) 所示前景/背景）。

OCR/文本纠错：输入是手写文本图像，输出是识别到的文本（图 (d)）。

自然语言处理：给出文本片段，输出可以是句子结构（词性标注）或翻译结果（图 (e)）。

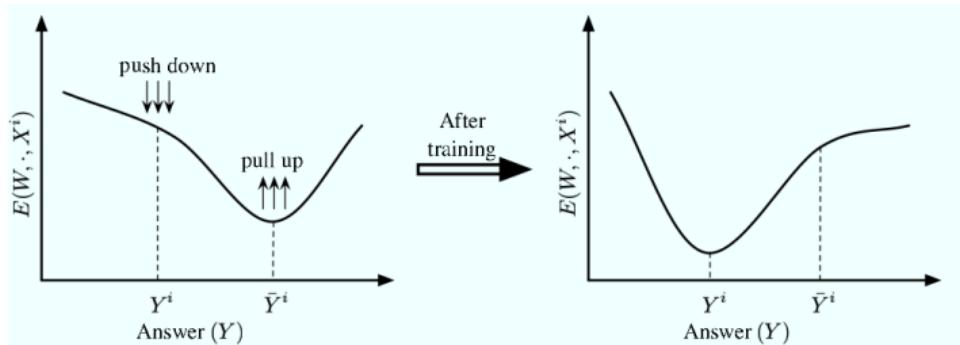
图像去噪/修复：输入是受损的图像，输出是修复后的图像（图 (f)）。

Energy-Based Learning

- 模型可通过“推/拉” (push/pull) 策略来训练：

- “push down”: 降低真实答案 y^i 的能量 $E(W, y^i, X^i)$, 即让正确输出的能量更低;
- “pull up”: 提升对比样本 \bar{y}^i 的能量, 使其高于真实答案, 以便区分正确与错误答案。
- 训练完成后, 理想状态是: y^i (真实标签) 能量较低、 \bar{y}^i (错误标签) 能量较高。

The model can be trained using a push/pull approach



Example: Generalized perceptron loss

$$L_{\text{perceptron}}(Y^i, E(W, \mathcal{Y}, X^i)) = E(W, Y^i, X^i) - \min_{Y \in \mathcal{Y}} E(W, Y, X^i)$$

Energy-Based Learning

Examples of Loss Functions:

Loss (equation #)	Formula	Margin
energy loss (6)	$E(W, Y^i, X^i)$	none
perceptron (7)	$E(W, Y^i, X^i) - \min_{Y \in \mathcal{Y}} E(W, Y, X^i)$	0
hinge (11)	$\max(0, m + E(W, Y^i, X^i) - E(W, \bar{Y}^i, X^i))$	m
log (12)	$\log(1 + e^{E(W, Y^i, X^i) - E(W, \bar{Y}^i, X^i)})$	> 0

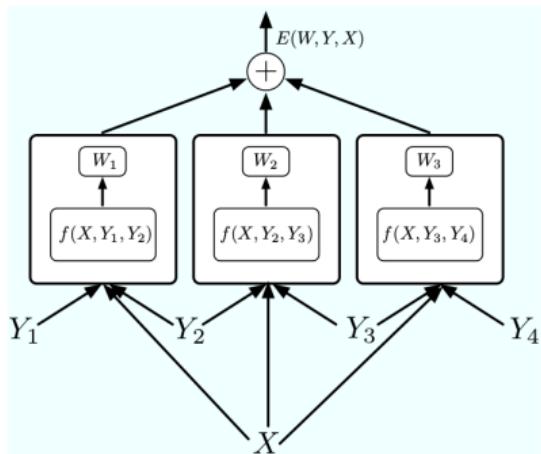
- ▶ Margin is an important component to ensure that the learned model generalizes well to test data.

Question: How to find the contrastive examples \bar{Y}^i for which the loss function is high, i.e. examples with low energy?

- 边界 (Margin) : 在一些损失函数 (如 hinge 损失) 中引入 m 等边界, 使得模型在区分正确和错误输出时有一定的“安全间隔”, 帮助模型在测试集上具有更好的泛化。
 - 问题: 如何找到那些会使损失函数高的对比样本 \bar{y}^i (即能量较低的“错误输出”) ?
 - 也就是说, 在训练过程中, 需要寻找 (或采样) 那些能量比真实样本相近或更低的错误输出 (难例), 来让模型更有针对性地提高对它们的区分能力。

Finding Contrastive Examples

For certain tasks where the output is sequential (e.g. segmentation) the model can be structured in a way that makes finding the best \bar{Y}^i easy.



Key steps:

- ▶ Define a set of functions involving only adjacent items in the sequence.
- ▶ The best sequences \bar{Y}^i can then be obtained using dynamic programming (cf. lecture on kernel-based structured prediction).

标题：找到对比示例

在某些输出是序列的任务（例如分割任务）中，模型可以设计成使找到最佳序列 \bar{Y}^i 变得容易。

关键步骤：

1. 定义一组仅涉及序列中相邻项的函数。
2. 最佳序列 \bar{Y}^i 可以通过动态规划（例如内核结构预测中的相关课程）来获得。

图示说明：

- 图中展示了模型结构：
 - 每个模块 W_1, W_2, W_3 分别与相邻的 Y 和 X 进行交互，通过函数 $f(X, Y_i, Y_{i+1})$ 进行计算。
 - 通过组合能量函数 $E(W, Y, X)$ 对序列的整体进行建模。

Finding Contrastive Examples

标题: 找到对比示例

- 对于动态规划不可用的问题, 可以训练一个生成网络 G_θ 来生成对比示例。

- 生成网络 G_θ 与结构化输出目标进行对抗训练, 例如, 对于对数损失:

$$\min_W \max_\theta \log \left(1 + \exp \left(E_W(Y, X) - E_W(G_\theta(X), X) \right) \right)$$

实现技巧: 通过单一优化器求解:

$$\min_{W, \theta} \log \left(1 + \exp \left(E_W(Y, X) - E_W(2G_\theta(X) - G_\theta(X), X) \right) \right)$$

并从微分图中分离棕色标记的项。

- 我们将在编程作业中使用这种方法。

- For problems where dynamic programming is not applicable, we can still train a ‘generator network’ [3] to produce good contrastive examples.
- The generator network G_θ is trained in competition with the structured output objective, e.g. for the log-loss:

$$\min_W \max_\theta \log \left(1 + \exp \left(E_W(Y, X) - E_W(G_\theta(X), X) \right) \right)$$

Implementation trick: solve with a single optimizer

$$\min_{W, \theta} \log \left(1 + \exp \left(E_W(Y, X) - E_W(2G_\theta(X) - G_\theta(X), X) \right) \right)$$

and detach the term in brown from the differentiation graph.

- We will use this method in the programming homework.

Adding Latent Variables

标题：添加潜在变量

为了处理未知因素（例如人脸的位置），可以使用潜在变量。

左侧：

- 使用潜在变量 T 对人脸检测建模，区分“有脸”(1) 和“无脸”(0)。

右侧：

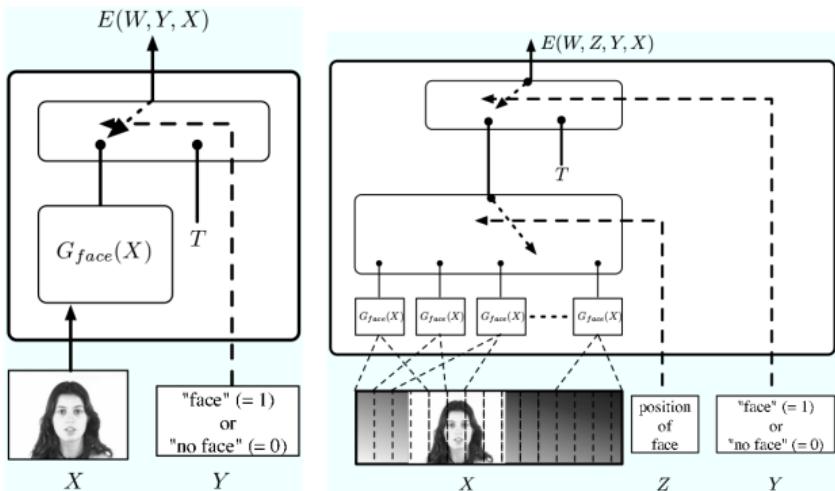
- 使用潜在变量 Z 来表示人脸的位置。

- 通过对所有可能的 Z 进行边缘化处理。公式为：

$$E(W, X, Y) = \min_{Z \in \mathcal{Z}} E(W, Z, X, Y)$$

- 动态规划可以用于计算此最小化。

Useful to account for unknown factors (e.g. position of the face).



Latent variable can be marginalized as $E(W, X, Y) = \min_{Z \in \mathcal{Z}} E(W, Z, X, Y)$, potentially using dynamic programming.

Summary

- ▶ Neural networks need to be adapted to be able to perform structured predictions.
- ▶ Simple methods for structured output learning include mixture density networks and conditional RBMs.
- ▶ More general structured predictions (e.g. sequences, trees, etc.) can be achieved within the flexible framework of energy-based learning.

References I

-  P. Baumeister, S. Padovan, N. Tosi, G. Montavon, N. Nettelmann, J. MacKenzie, and M. Godolt.
Machine-learning inference of the interior structure of low-mass exoplanets.
The Astrophysical Journal, 889(1):42, Jan. 2020.
-  C. M. Bishop.
Neural Networks for Pattern Recognition.
Oxford University Press, Inc., USA, 1995.
-  I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio.
Generative adversarial nets.
In *NIPS*, pages 2672–2680, 2014.
-  Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang.
A tutorial on energy-based learning.
MIT Press, 2006.
-  R. Memisevic and G. E. Hinton.
Learning to represent spatial transformations with factored higher-order boltzmann machines.
Neural Computation, 22(6):1473–1492, 2010.
-  V. Mnih, H. Larochelle, and G. E. Hinton.
Conditional restricted boltzmann machines for structured output prediction.
In *UAI*, pages 514–522. AUAI Press, 2011.