Energy-Based Learning

with a brief history of CRFs since 1991

[Driancourt&Bottou 1991, Bottou 1991, Denker&Burges 1995, LeCun et al. 1998, Lafferty et al, 2001]

Yann LeCun

The Courant Institute of Mathematical Sciences

New York University

See: [LeCun et al. 2006]: "A Tutorial on Energy-Based Learning"

http://yann.lecun.com/exdb/publis/

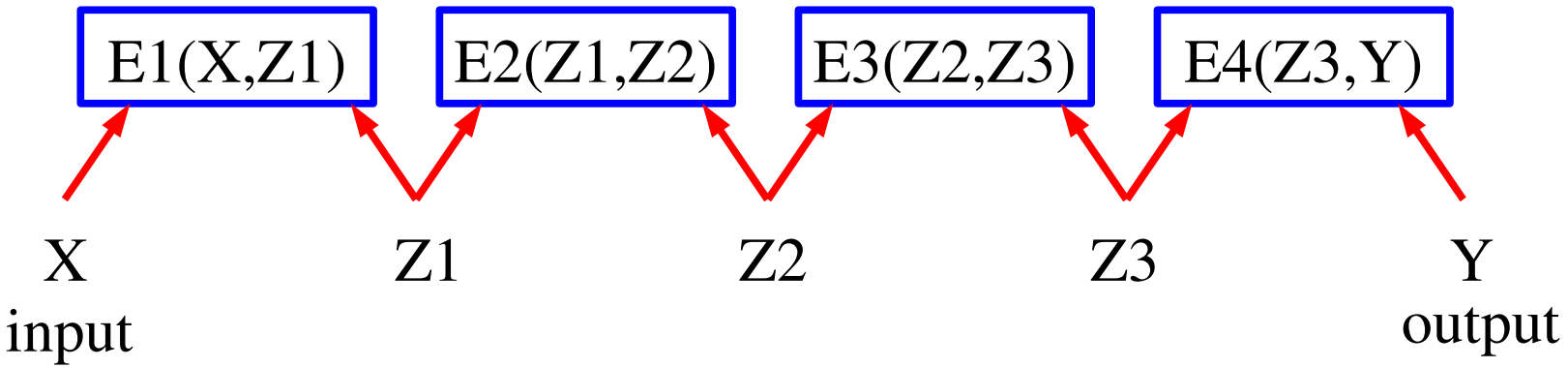
Two Problems in Machine Learning and Vision

1. The "Partition Function Problem"

- Max likelihood learning gives high probability to good answers
- The partition function ensures that undesired answers are given low probability
- For learning, we need to approximate the partition function (or its gradient with respect to the parameters)
- Problem: there are too many bad answers!

2. The "Deep Learning Problem"

- "Deep belief networks (e.g. Hierarchies of features) are a good way to handle the invariance problem in vision (and perception in general) [Selfridge 1958, Hubel&Wiesel 1961, Fukushima 1978-84, LeCun 1989,-2006, Poggio 2002-2006, Lowe 2006,.....]
- How can we train deep belief networks efficiently?



The Partition Function Problem

- **Predict Y from X**

$$P(Y|X) = \frac{e^{-\beta E(Y,X)}}{\int_{y \in \mathcal{Y}} e^{-\beta E(y,X)}},$$

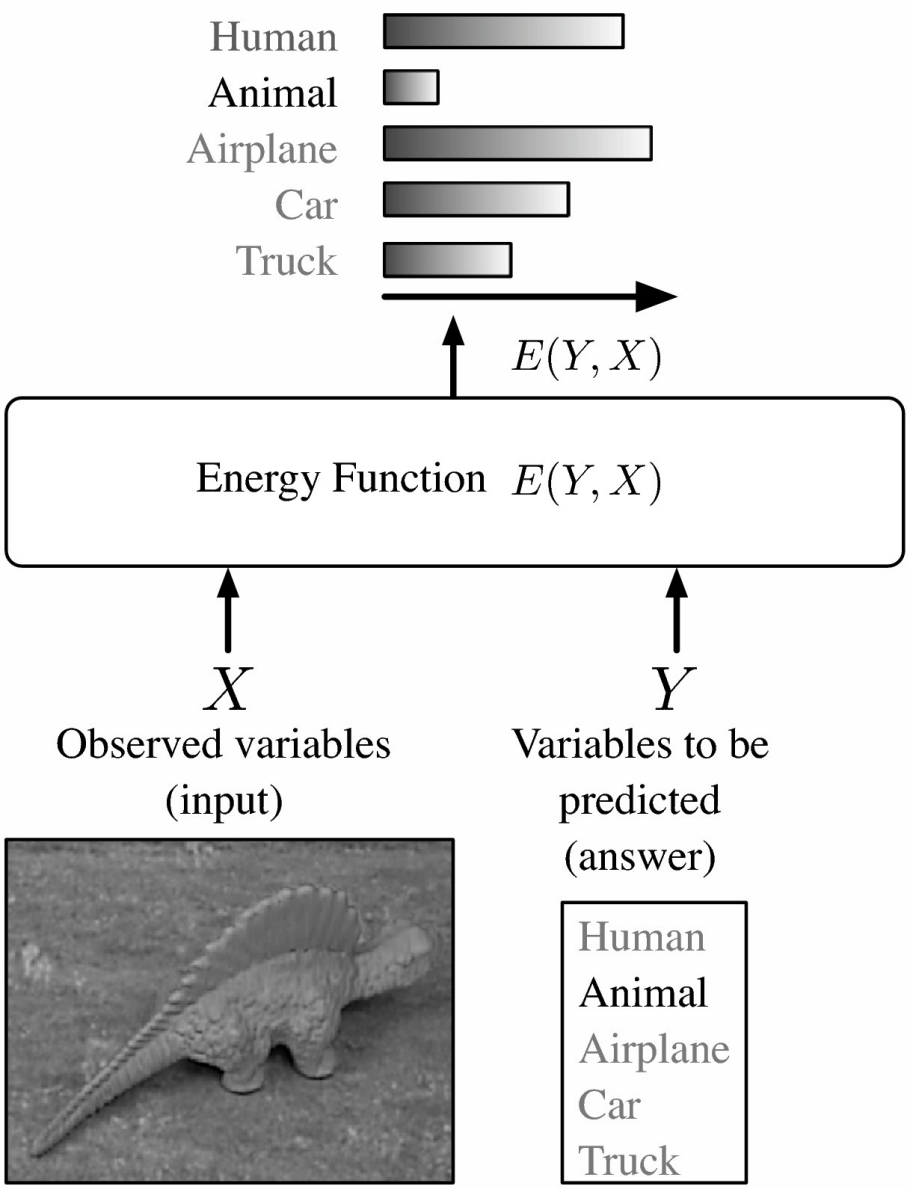
- **Negative Log Likelihood Loss function**

$$\mathcal{L}_{\text{nll}}(W, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} \left(E(W, Y^i, X^i) + \frac{1}{\beta} \log \int_{y \in \mathcal{Y}} e^{-\beta E(W,y,X^i)} \right).$$

- **Gradient of Negative Log Likelihood Loss function**

$$\frac{\partial L_{\text{nll}}(W, Y^i, X^i)}{\partial W} = \frac{\partial E(W, Y^i, X^i)}{\partial W} - \int_{Y \in \mathcal{Y}} \frac{\partial E(W, Y, X^i)}{\partial W} P(Y|X^i, W),$$

Energy-Based Model for Decision-Making

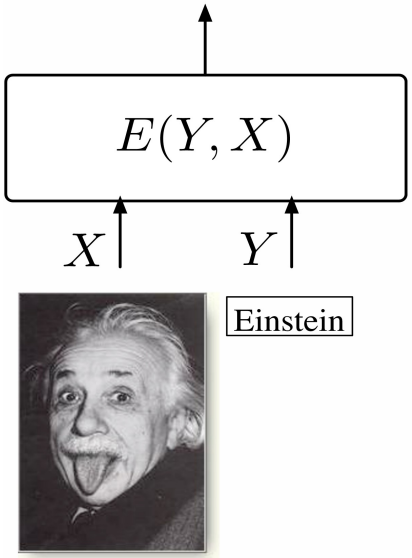


- **Model:** Measures the compatibility between an observed variable X and a variable to be predicted Y through an energy function E(Y,X).

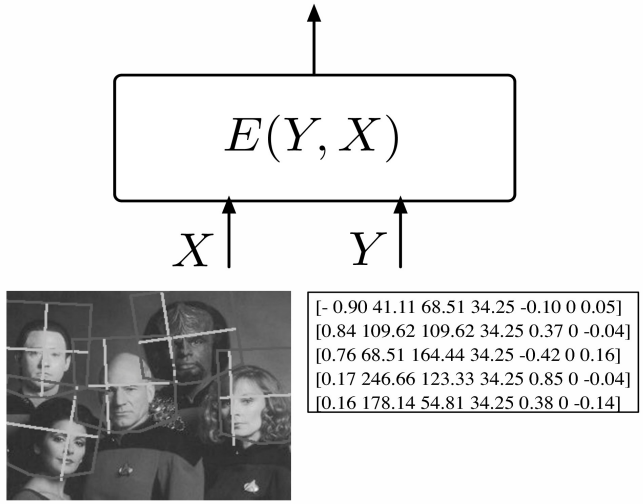
$$Y^* = \text{argmin}_{Y \in \mathcal{Y}} E(Y, X).$$

- **Inference:** Search for the Y that minimizes the energy within a set $\mathcal{Y}$
- If the set has low cardinality, we can use exhaustive search.

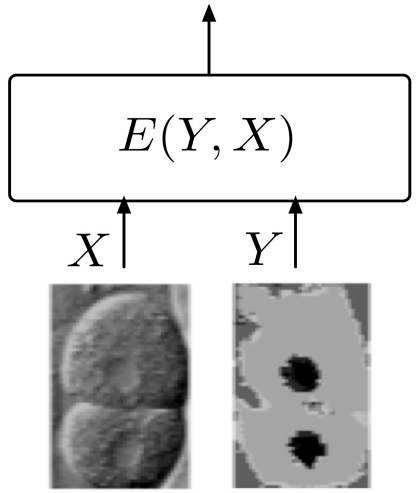
Complex Tasks: Inference is non-trivial



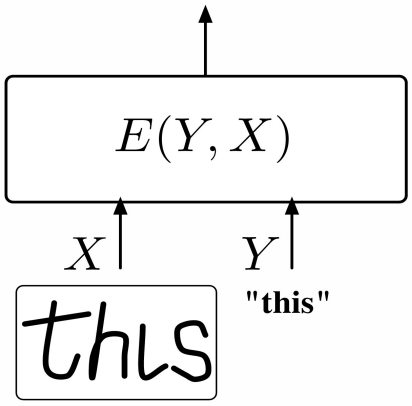
(a)



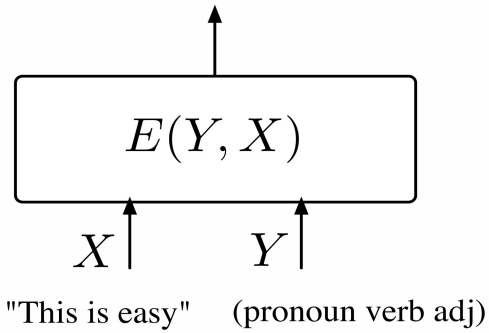
(b)



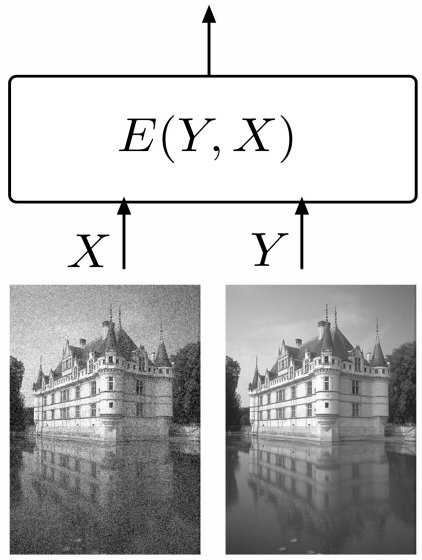
(c)



(d)



(e)



(f)

- **When the cardinality or dimension of Y is large, we must use a suitable inference procedure: Viterbi, min-sum, min cut, gradient descent.....**

Turning Energies into Probabilities

- **Simplest way: Gibbs distribution**
  - Other ways can be reduced to Gibbs by a suitable redefinition of the energy.

$$P(Y|X) = \frac{e^{-\beta E(Y,X)}}{\int_{y \in \mathcal{Y}} e^{-\beta E(y,X)}},$$

Partition function

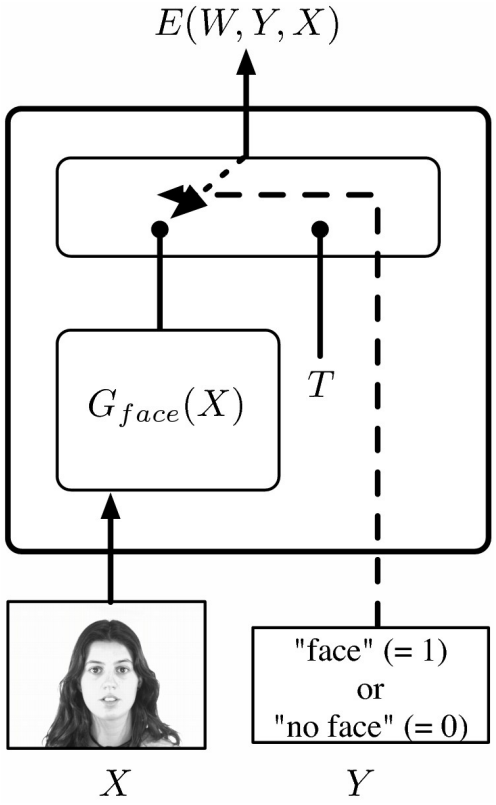
Inverse temperature

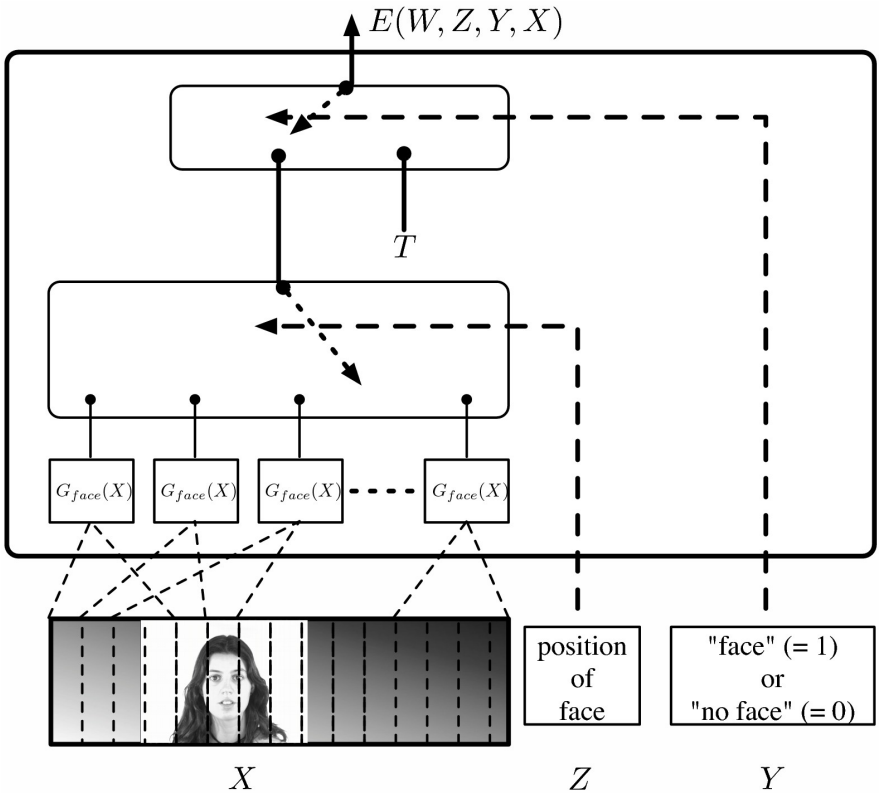
Latent Variable Models

- **The energy includes "hidden" variables Z whose value is never given to us**

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

$$Y^* = \text{argmin}_{Y \in \mathcal{Y}, Z \in \mathcal{Z}} E(Z, Y, X).$$





Probabilistic Latent Variable Models

- Marginalizing over latent variables instead of minimizing.

$$P(Z,Y|X) = \frac{e^{-\beta E(Z,Y,X)}}{\int_{y\in\mathcal{Y},\ z\in\mathcal{Z}} e^{-\beta E(y,z,X)}}.$$

$$P(Y|X) = \frac{\int_{z\in\mathcal{Z}} e^{-\beta E(Z,Y,X)}}{\int_{y\in\mathcal{Y},\ z\in\mathcal{Z}} e^{-\beta E(y,z,X)}}.$$

- Equivalent to traditional energy-based inference with a redefined energy function (the free energy):

$$Y^* = \text{argmin}_{Y\in\mathcal{Y}} -\frac{1}{\beta}\log\int_{z\in\mathcal{Z}} e^{-\beta E(z,Y,X)}.$$

- Reduces to traditional minimization when Beta->infinity

Uncertainty and Inference

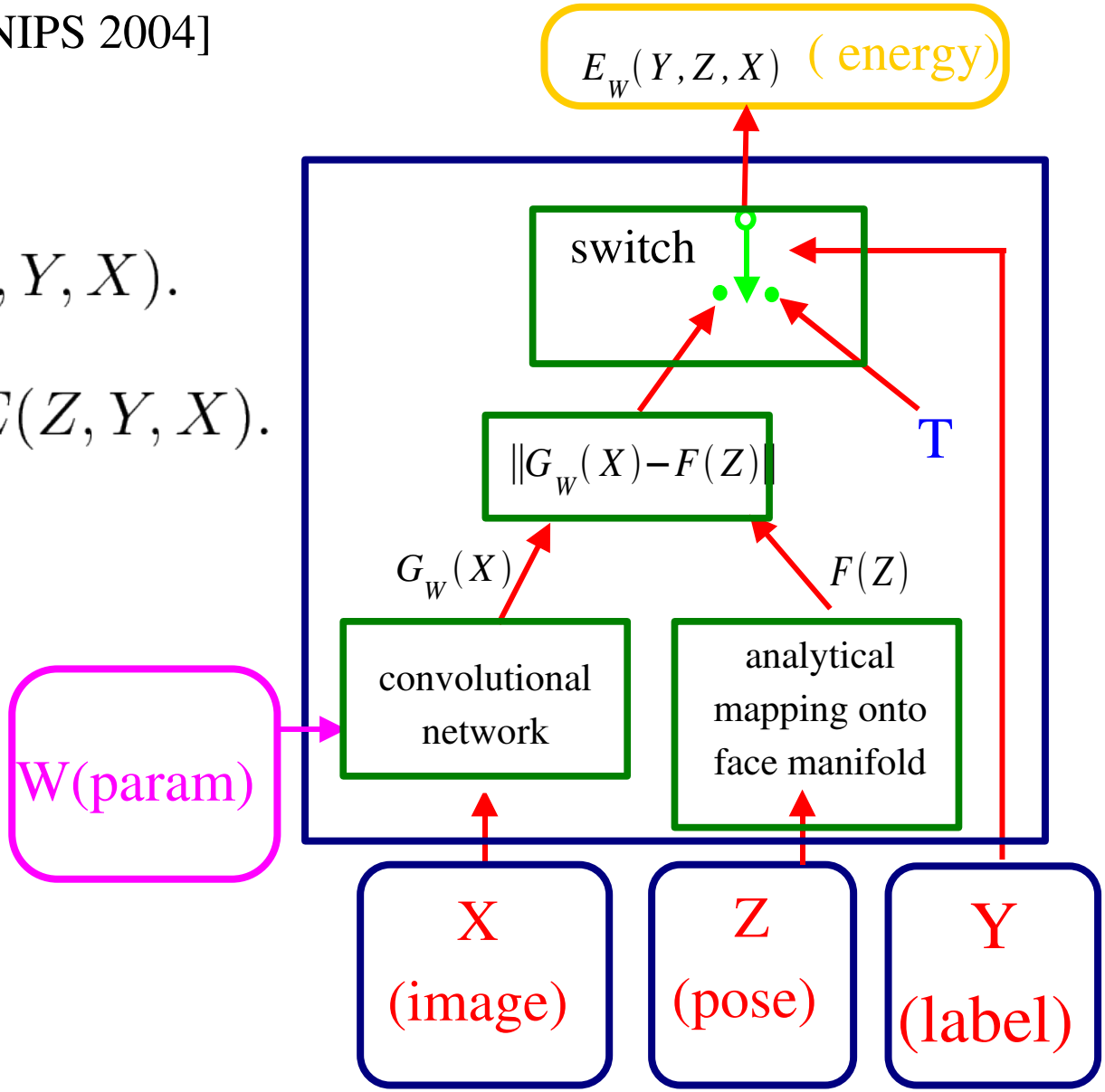
- **Global energy minimization gives the same result as MAP probabilistic inference (if trained with negative lof likelihood!)**
  - Normalization plays nor role whatsoever in inference
  - We do not need normalized probability distributions for handling uncertainty during inference

Simultaneous Face Detection and Pose Estimation

[Osadchy, Miller, LeCun, NIPS 2004]

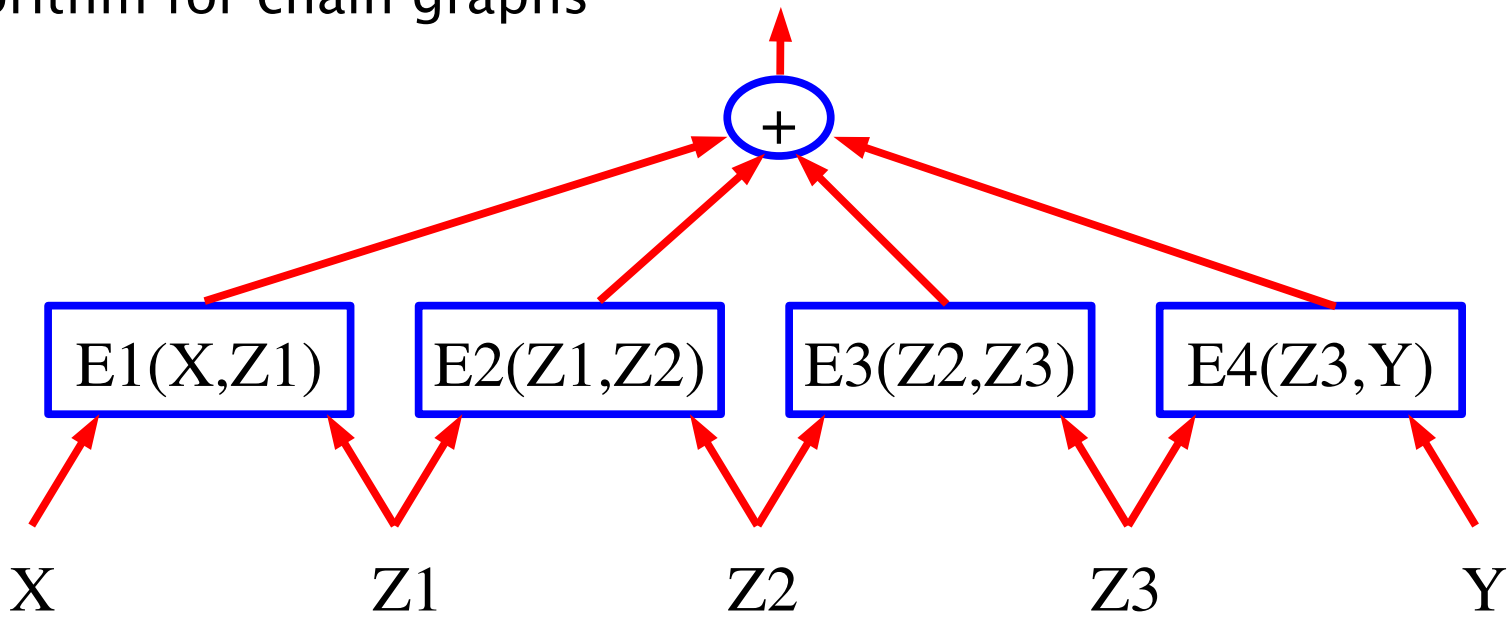
$$E(Y, X) = \min_{Z \in \mathcal{Z}} E(Z, Y, X).$$

$$Y^* = \text{argmin}_{Y \in \mathcal{Y}, Z \in \mathcal{Z}} E(Z, Y, X).$$



Energy-Based Factor Graphs

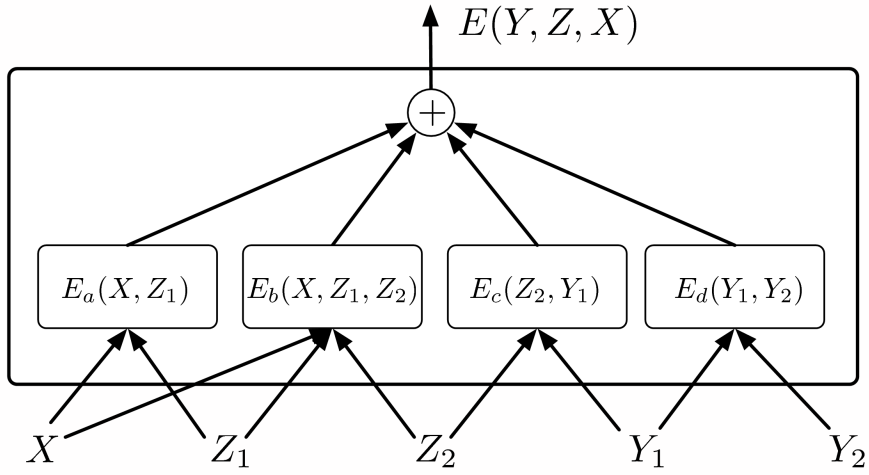
- **When the energy is a sum of partial energy functions (or when the probability is a product of factors):**
  - An EBM can be seen as a factor graph in the log domain
  - Our favorite efficient inference algorithms can be used for inference (without the normalization step).
  - Min-sum algorithm (instead of max-product), Viterbi for chain graphs
  - Log/sum/exp-sum algorithm (instead of sum-product), Forward algorithm for chain graphs



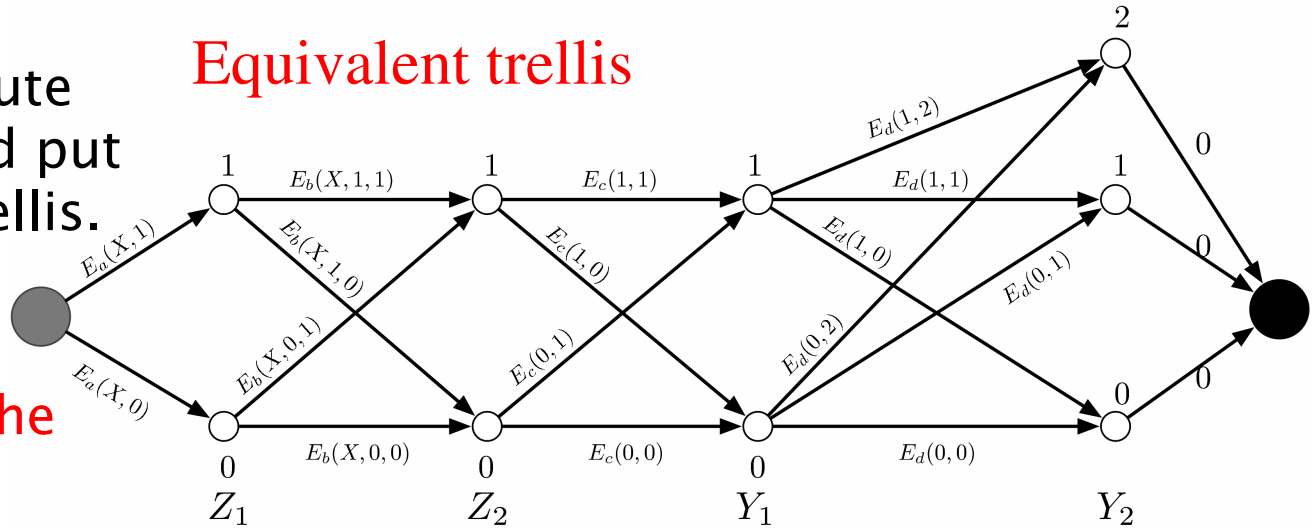
Efficient Inference: Energy-Based Factor Graphs

- **The energy is a sum of "factor" functions**
- **Example:**
  - Z1, Z2, Y1 are binary
  - Z2 is ternary
  - A naïve exhaustive inference would require 2x2x2x3=24 energy evaluations (= 96 factor evaluations)
  - BUT: Ea only has 2 possible input configurations, Eb and Ec have 4, and Ed 6.
  - Hence, we can precompute the 16 factor values, and put them on the arcs in a trellis.
  - A path in the trellis is a config of variable
  - The cost of the path is the energy of the config

Factor graph

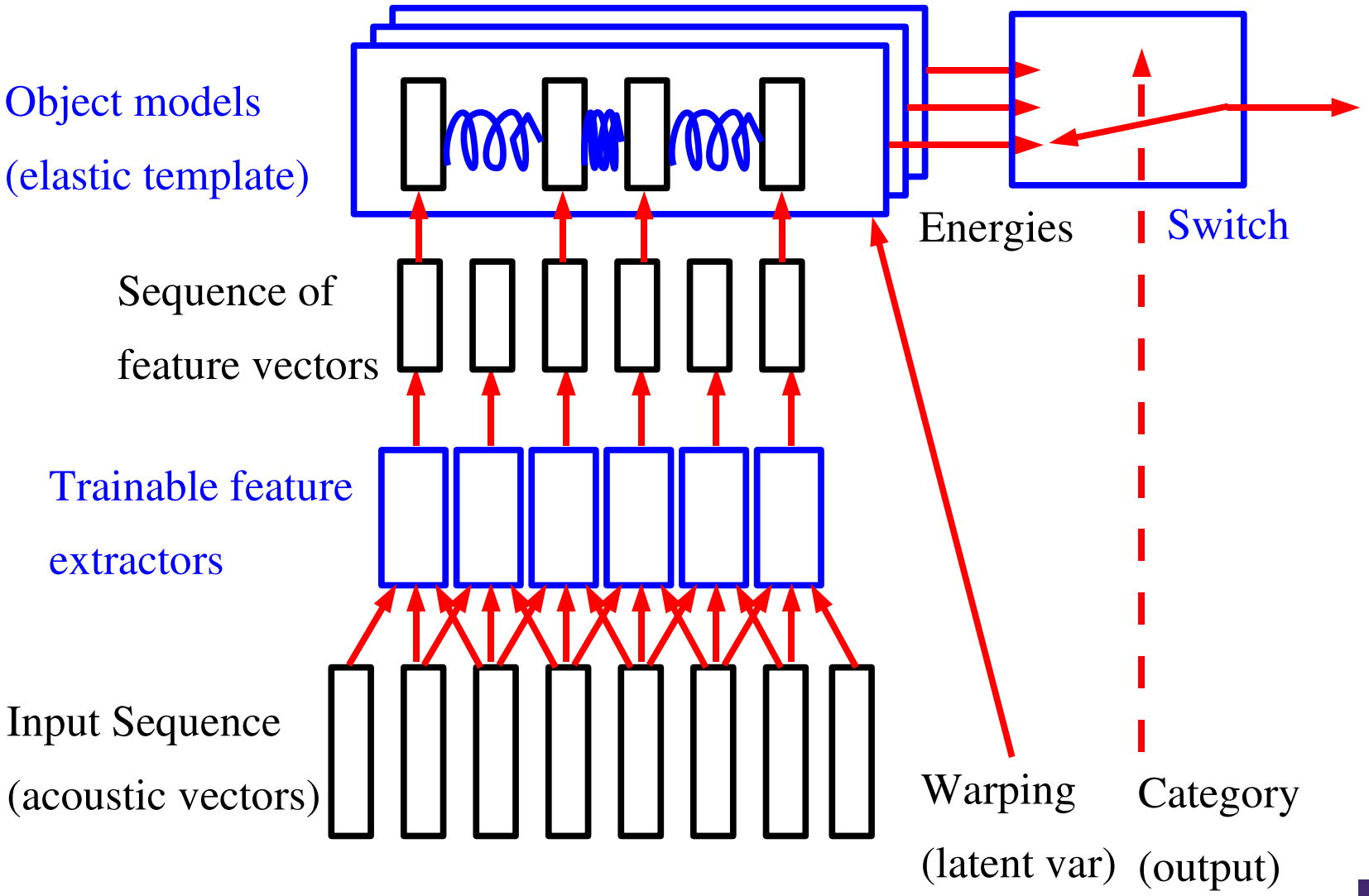


Equivalent trellis



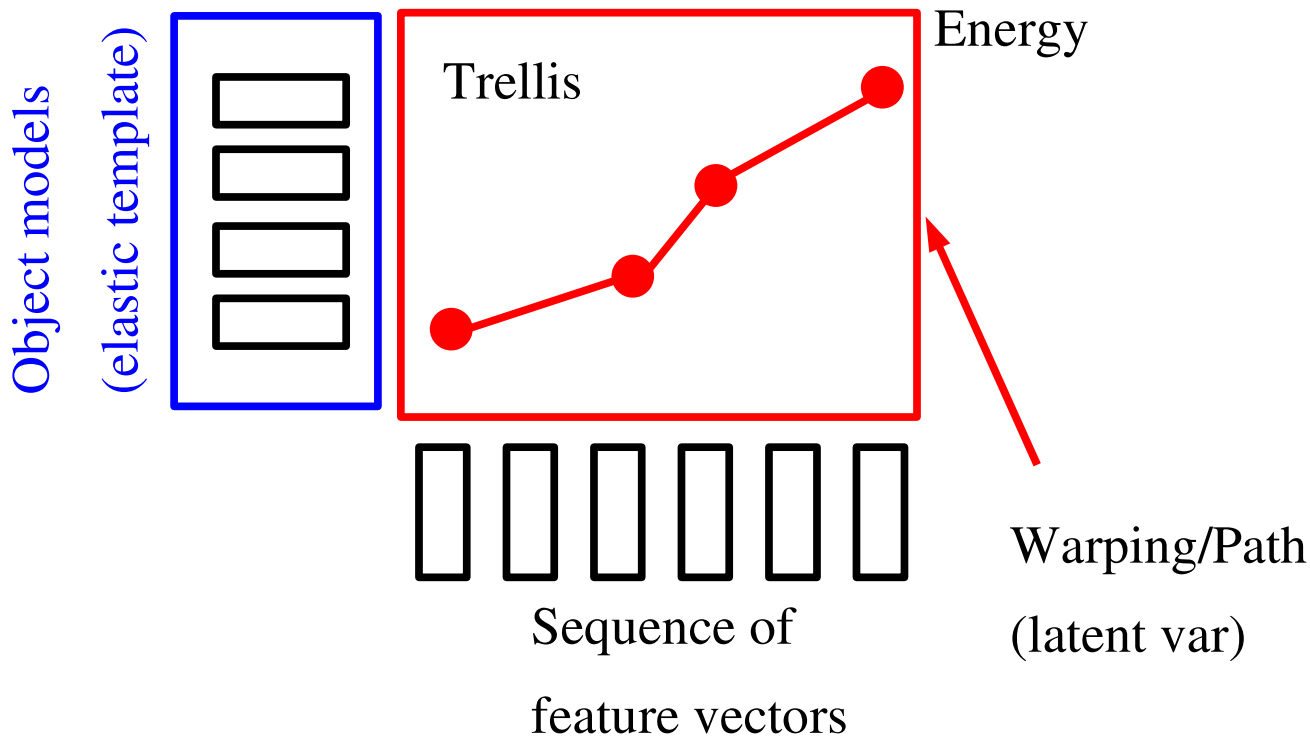
Example 1: 1-D Constellation Model (a.k.a. Dynamic Time Warping)

- **Spoken word recognition with trainable elastic templates and trainable feature extraction [Driancourt&Bottou 1991, Bottou 1991, Driancourt 1994]**



Example: 1-D Constellation Model (a.k.a. Dynamic Time Warping)

- Spoken word recognition with trainable elastic templates and trainable feature extraction [Driancourt&Bottou 1991, Bottou 1991, Driancourt 1994]
- Elastic matching using dynamic time warping (Viterbi algorithm on a trellis).

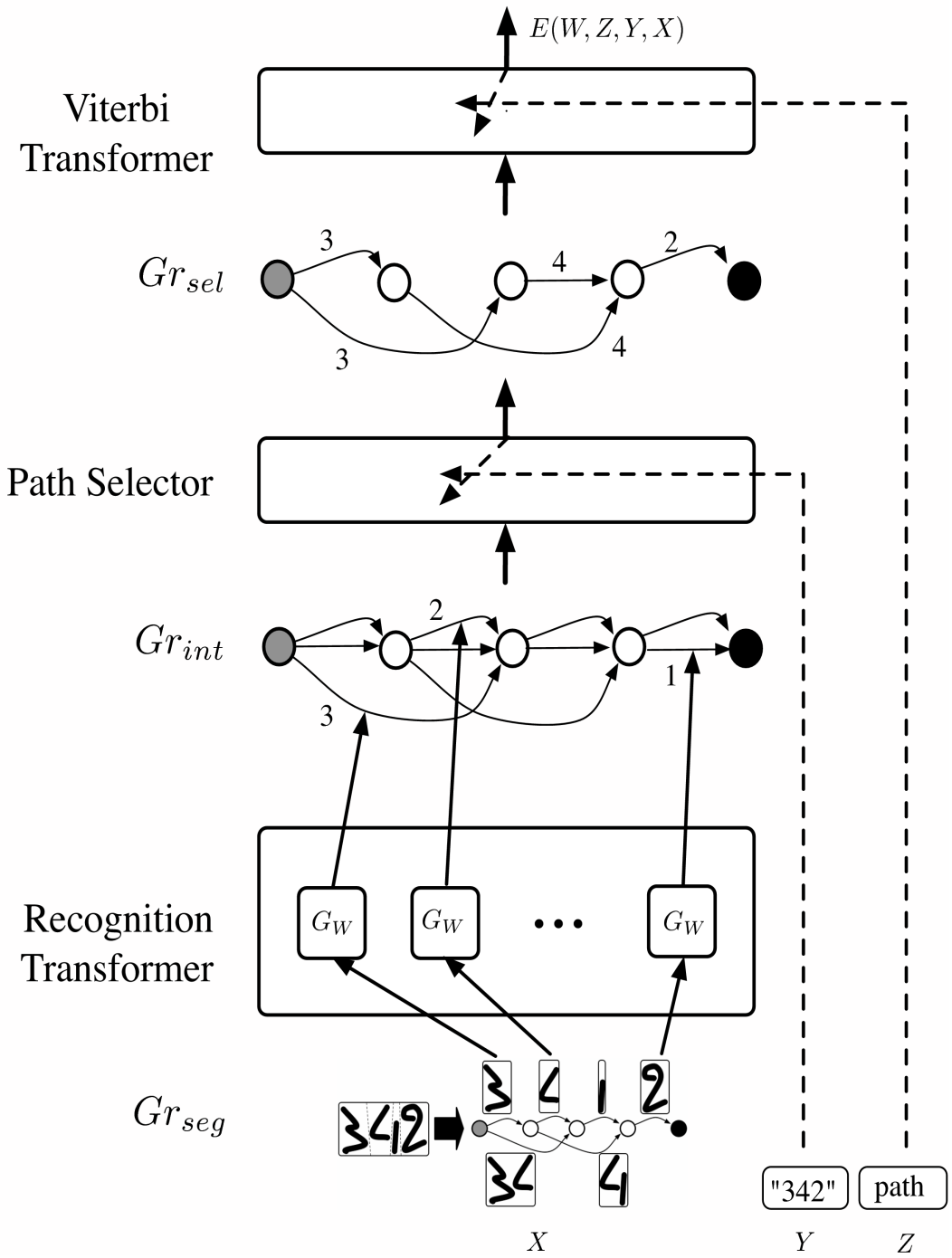


Example 2: Parts-Based Recognition

- **Handwriting word recognition**
  - Integrated training of word models, character recognizer, and segmentation.
- **Answer = sequence of symbols**
- **Latent variable = segmentation**

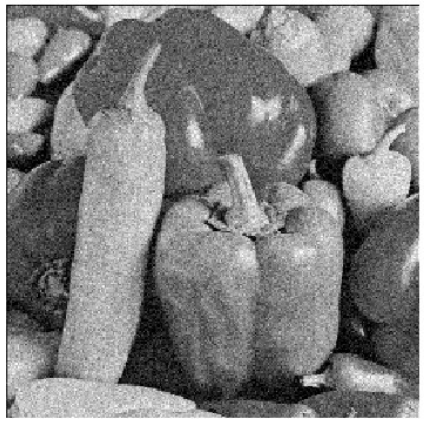
[Bengio, LeCun 1994]

[LeCun et al. 1998]



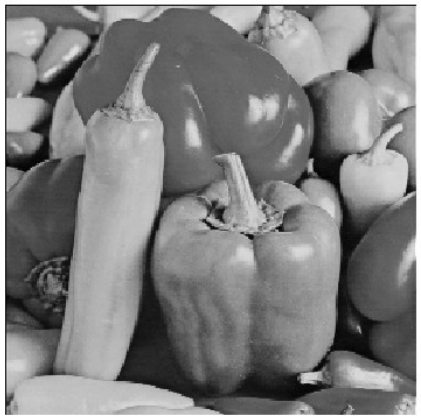
Example 3: Convolutional Product of Experts

- Local consistency through local energy functions replicated across the image [Feng et al, IEEE Trans. Image Processing, 2005],
- [Roth & Black, CVPR 2005] "Field of Experts" is essentially identical, but is trained generatively, not discriminatively.



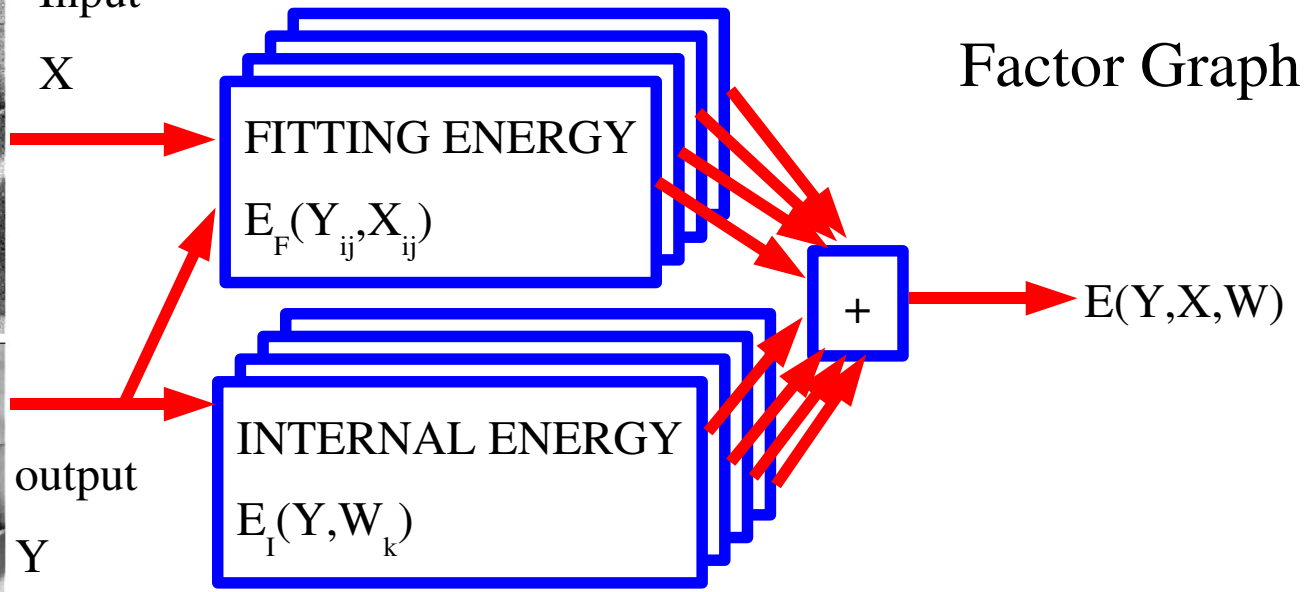
Input

X



output

Y

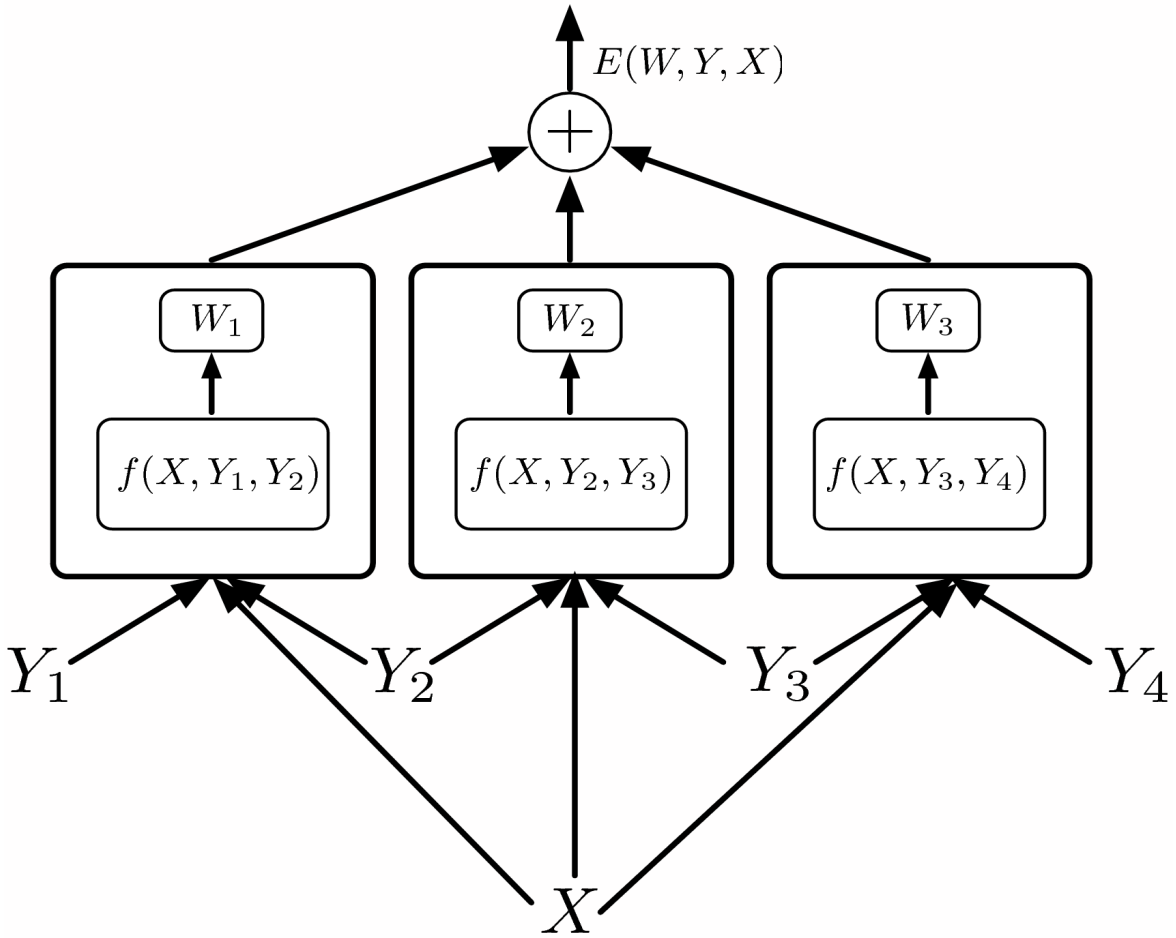


Factor Graph

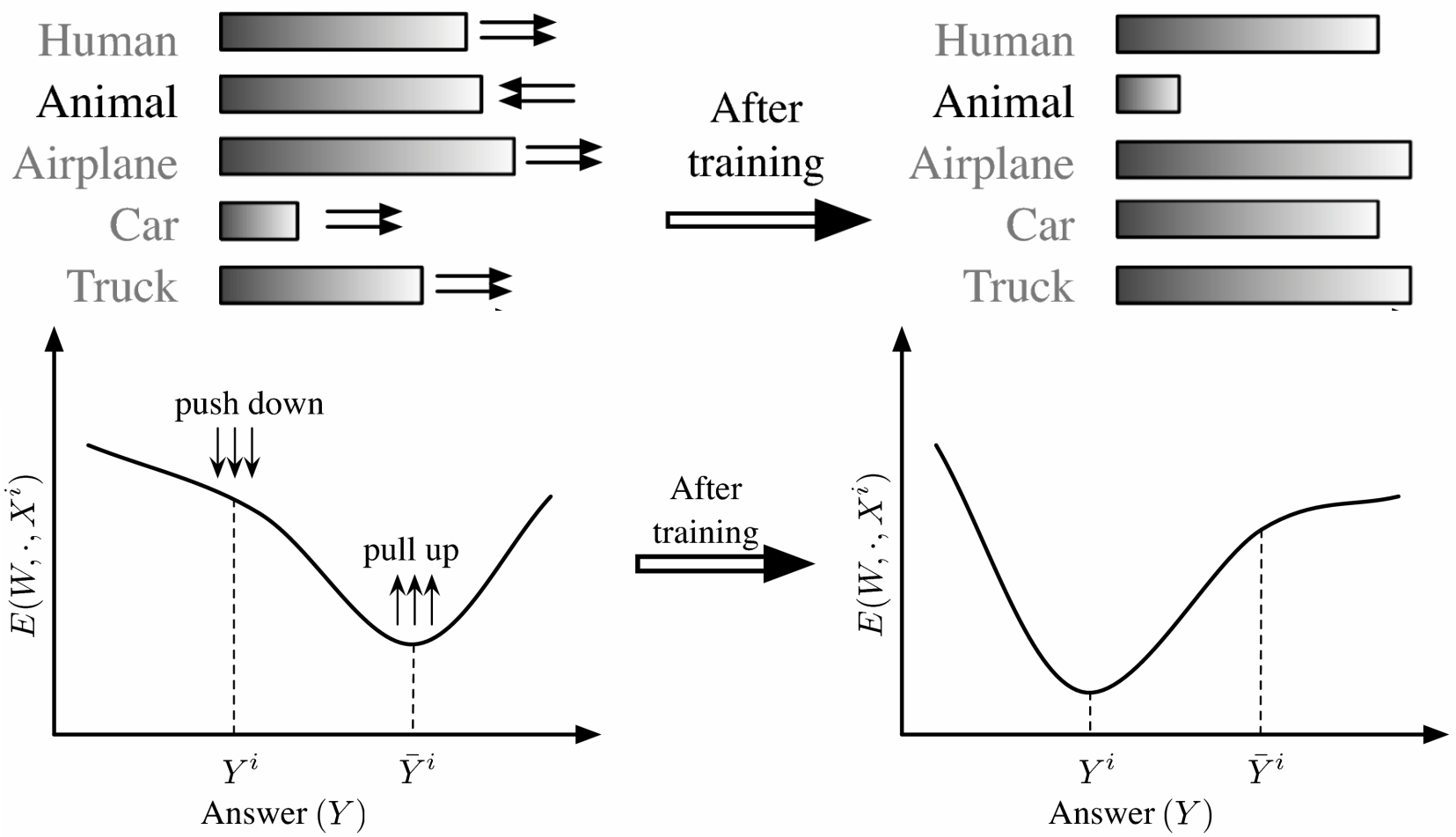
Each factor takes a local patch as input

Example 4: Traditional CRF Architecture

- **A traditional CRF is an energy-based factor graph in which:**
  - the factors are **linear in the parameters** **(shallow factors)**
  - The factors take neighboring output variables as inputs
  - The factors are often all identical



How do we Train and Energy-Based Model?



- **Push down on the energy of the correct answer**
- **Pull up on the energies of the incorrect answers, particularly if their energies are lower than that of the correct answer.**

Architecture and Loss Function

- **Family of energy functions** $\mathcal{E} = \{E(W, Y, X) : \quad W \in \mathcal{W}\}.$
- **Training set** $\mathcal{S} = \{(X^i, Y^i) : i = 1 \ldots P\}$
- **Loss functional / Loss function** $\mathcal{L}(E, \mathcal{S}) \qquad \mathcal{L}(W, \mathcal{S})$
  - Measures the quality of an energy function
- **Training** $W^* = \min_{W \in \mathcal{W}} \mathcal{L}(W, \mathcal{S}).$
- **Form of the loss functional**
  - invariant under permutations and repetitions of the samples

$$\mathcal{L}(E, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} L(Y^i, E(W, \mathcal{Y}, X^i)) + R(W).$$

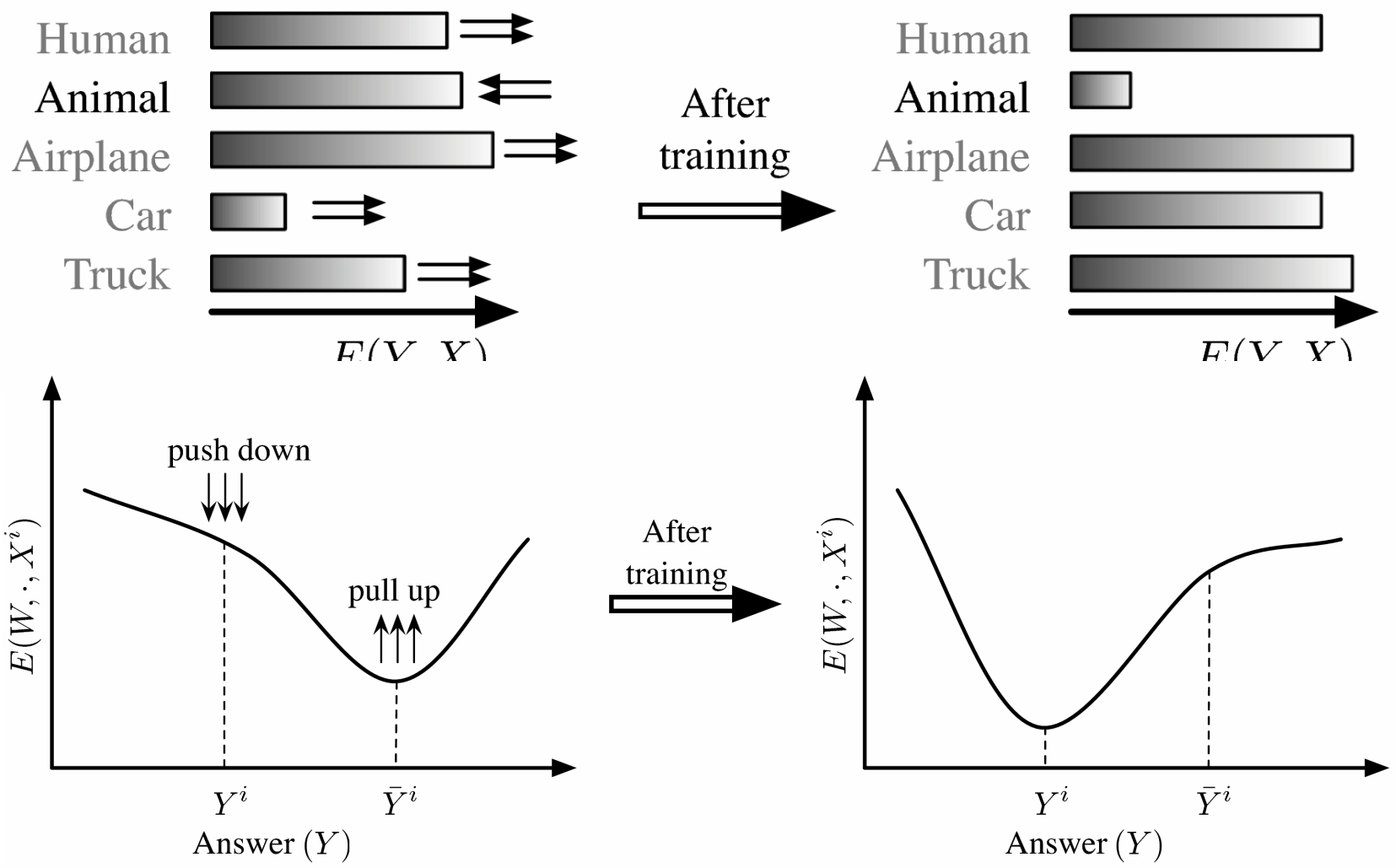
Per-sample loss

Desired answer

Energy surface for a given Xi as Y varies

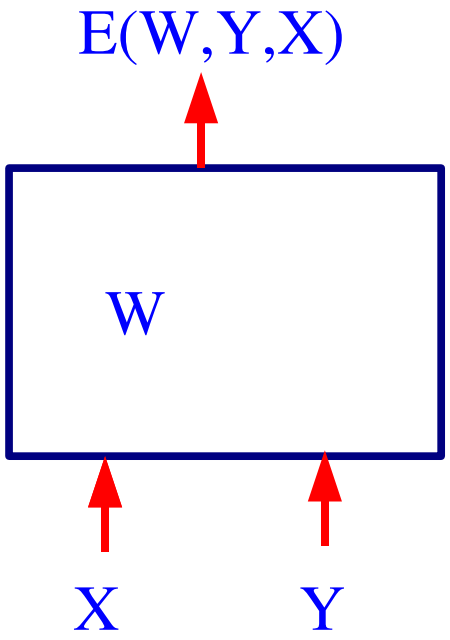
Regularizer

Designing a Loss Functional



- **Correct answer has the lowest energy -> LOW LOSS**
- **Lowest energy is not for the correct answer -> HIGH LOSS**

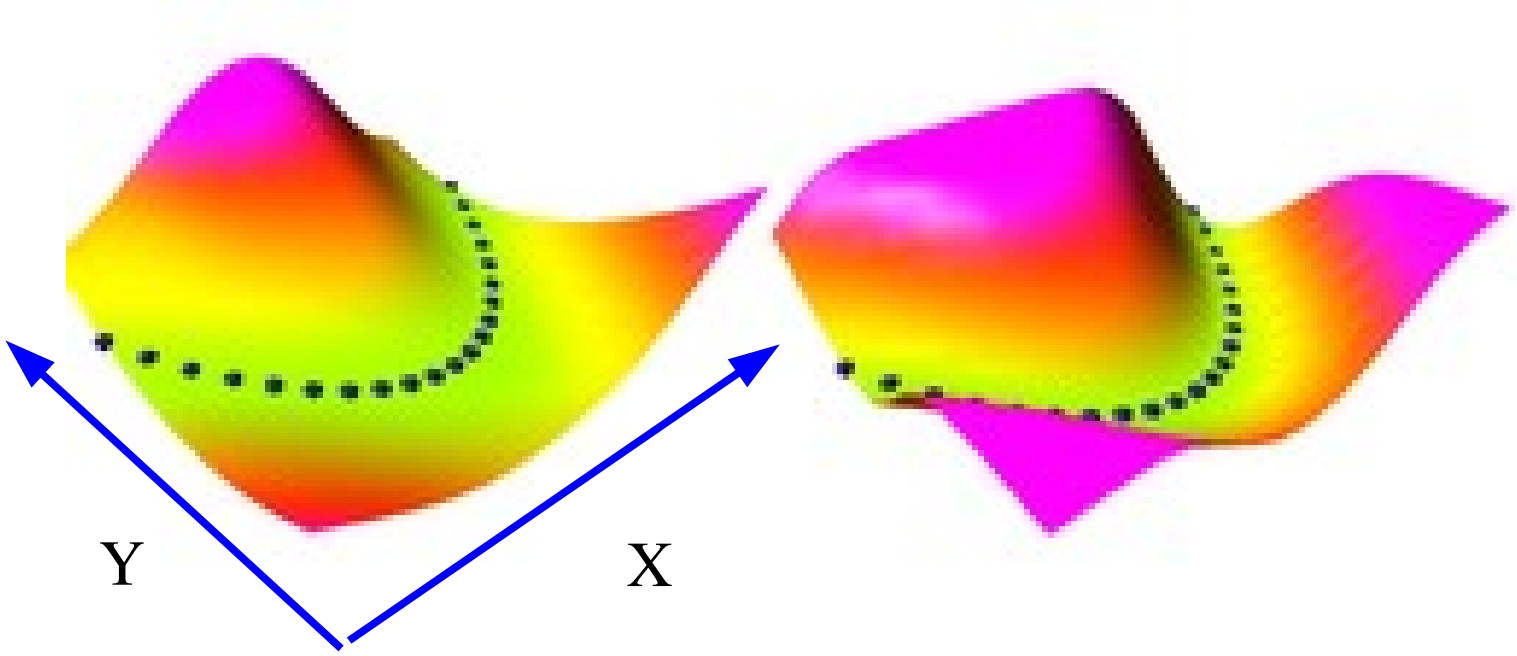
Architecture + Inference Algo + Loss Function = Model



- **1. Design an architecture:** a particular form for E(W,Y,X).
- **2. Pick an inference algorithm for Y:** MAP or conditional distribution, belief prop, min cut, variational methods, gradient descent, MCMC, HMC.....
- **3. Pick a loss function:** in such a way that minimizing it with respect to W over a training set will make the inference algorithm find the correct Y for a given X.
- **4. Pick an optimization method.**

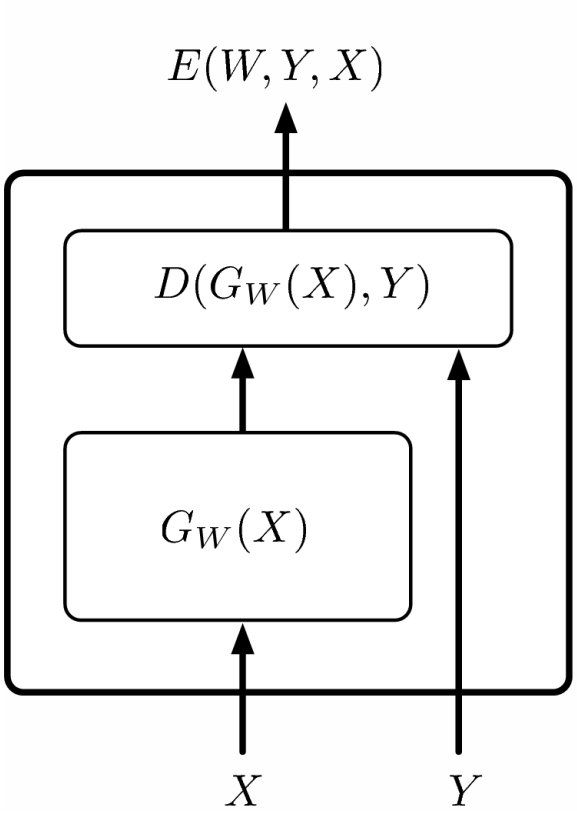
- **PROBLEM: What loss functions will make the machine approach the desired behavior?**

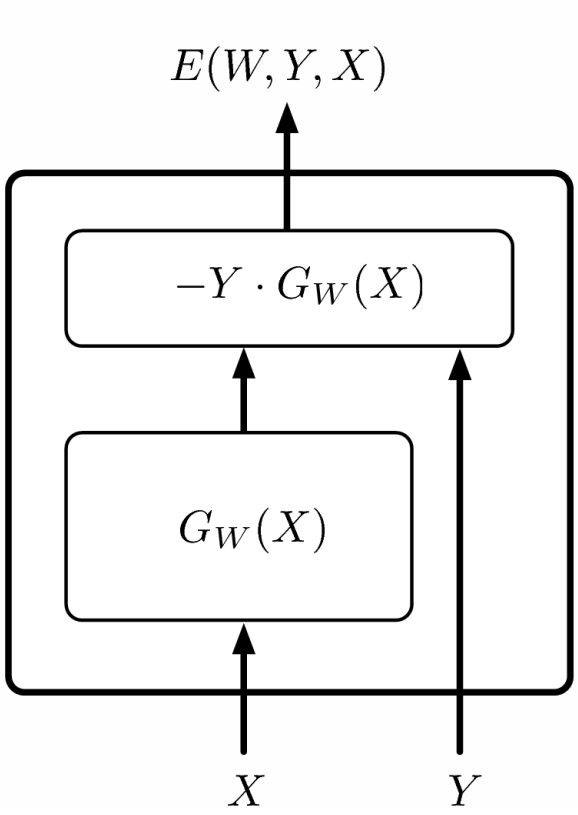
Several Energy Surfaces can give the same answers

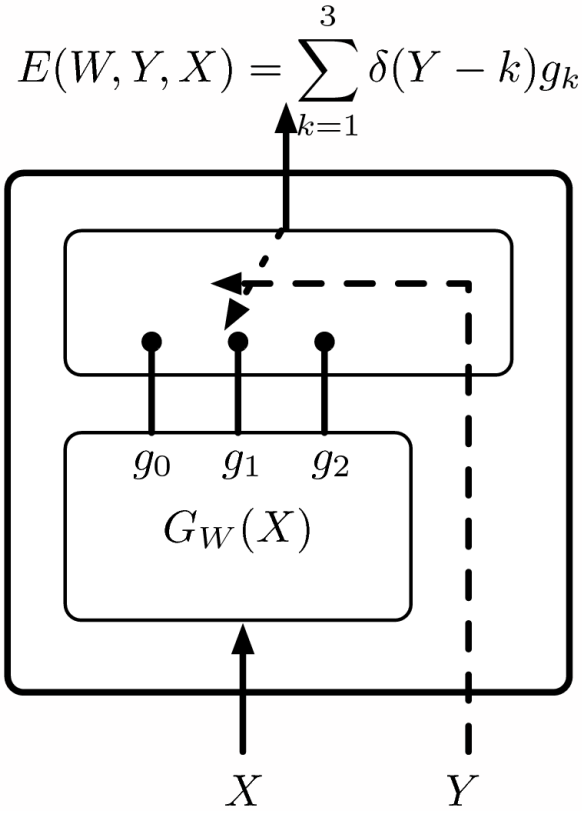


- Both surfaces compute Y=X^2
- MINy E(Y,X) = X^2
- Minimum-energy inference gives us the same answer

Simple Architectures







- **Regression**

$$E(W,Y,X) = \frac{1}{2}||G_W(X) - Y||^2.$$

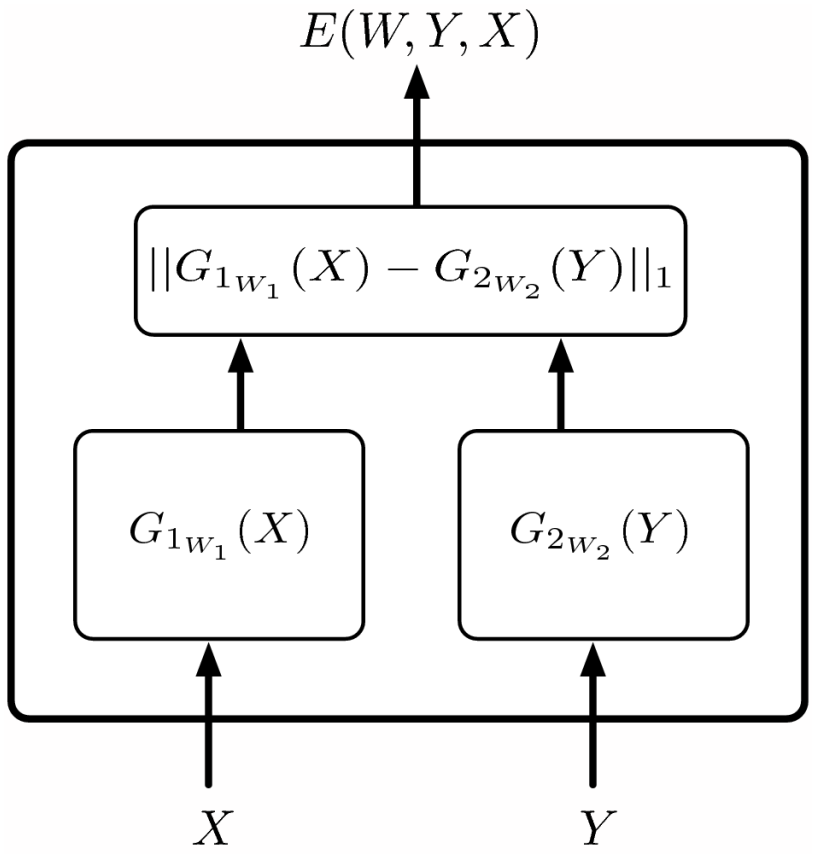
- **Binary Classification**

$$E(W,Y,X) = -Y G_W(X),$$

- **Multi-class Classification**

Simple Architecture: Implicit Regression

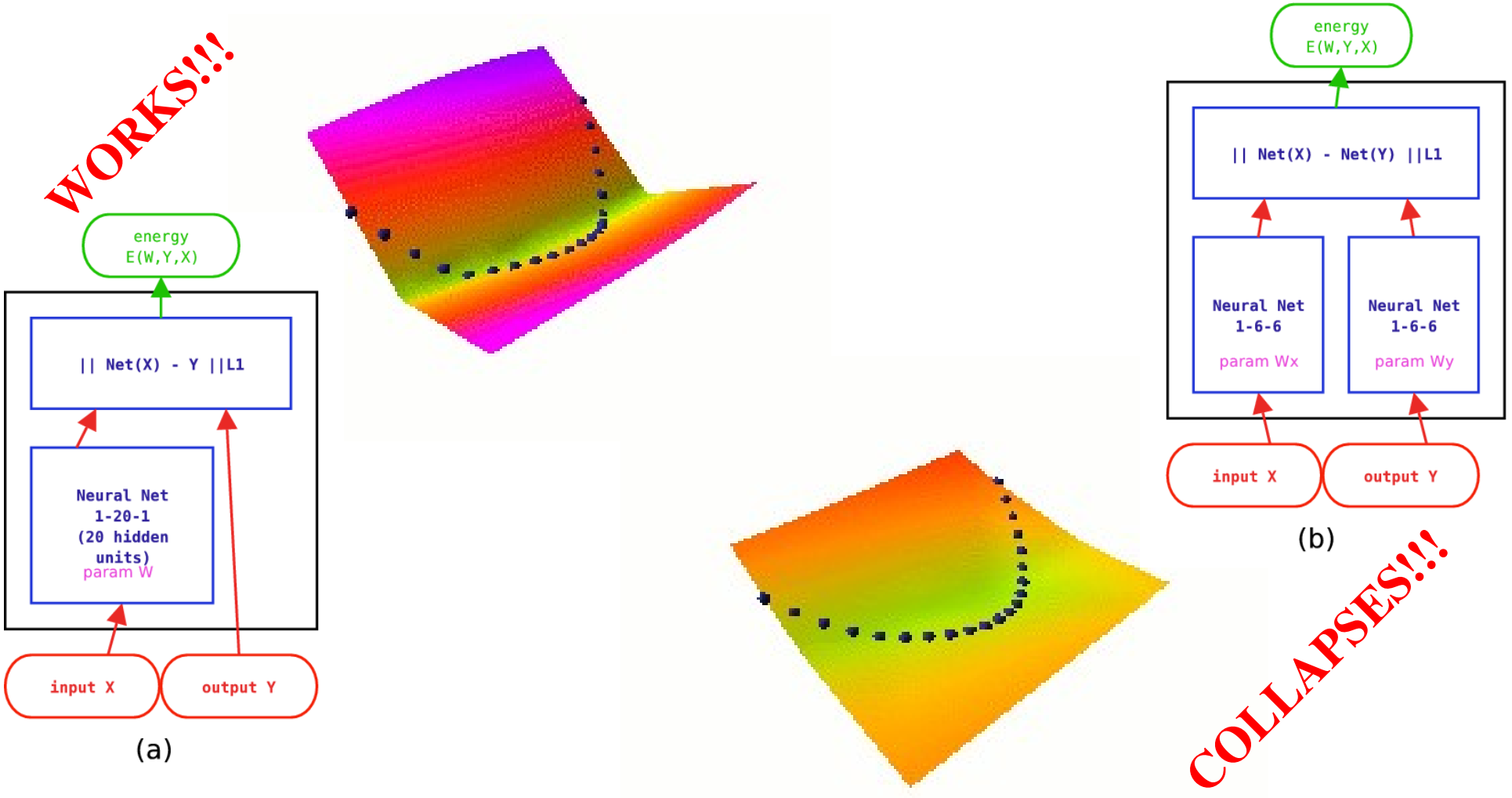
$$E(W, X, Y) = ||G_{1_{W_1}}(X) - G_{2_{W_2}}(Y)||_1,$$



- **The Implicit Regression architecture**
  - allows multiple answers to have low energy.
  - Encodes a constraint between X and Y rather than an explicit functional relationship
  - This is useful for many applications
  - Example: sentence completion: "The cat ate the {mouse,bird,homework,...}"
  - [Bengio et al. 2003]
  - But, inference may be difficult.

Examples of Loss Functions: Energy Loss

- **Energy Loss** $L_{energy}(Y^i, E(W, \mathcal{Y}, X^i)) = E(W, Y^i, X^i).$
  - Simply pushes down on the energy of the correct answer



Examples of Loss Functions:Perceptron Loss

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

- **Perceptron Loss**
  - [LeCun et al. 1998] for handwriting
  - [Collins 2002] for parts of speech tagging
  - Pushes down on the energy of the correct answer
  - Pulls up on the energy of the machine's answer
  - Always positive. Zero when answer is correct
  - No "margin": technically does not prevent the energy surface from being almost flat.
  - Works pretty well in practice, particularly if the energy parameterization does not allow flat surfaces.

Perceptron Loss for Binary Classification

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

- **Energy:** $E(W, Y, X) = -Y G_W(X),$
- **Inference:** $Y^* = \text{argmin}_{Y \in \{-1,1\}} - Y G_W(X) = \text{sign}(G_W(X)).$
- **Loss:** $\mathcal{L}_{\text{perceptron}}(W, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} \left(\text{sign}(G_W(X^i)) - Y^i\right) G_W(X^i).$
- **Learning Rule:** $W \leftarrow W + \eta \left(Y^i - \text{sign}(G_W(X^i))\right) \frac{\partial G_W(X^i)}{\partial W},$
- **If Gw(X) is linear in W:** $E(W, Y, X) = -Y W^T \Phi(X)$

$$W \leftarrow W + \eta \left(Y^i - \text{sign}(W^T \Phi(X^i))\right) \Phi(X^i)$$

Generalized Margin Losses: Most Offending Incorrect Answer

- **First, we need to define the Most Offending Incorrect Answer**

- **Most Offending Incorrect Answer: discrete case**

Definition 1 *Let Y be a discrete variable. Then for a training sample (X^i, Y^i), the* ***most offending incorrect answer*** *$\bar{Y}^i$ is the answer that has the lowest energy among all answers that are incorrect:*

$$\bar{Y}^i = \text{argmin}_{Y \in \mathcal{Y} and Y \neq Y^i} E(W, Y, X^i). \quad (8)$$

- **Most Offending Incorrect Answer: continuous case**

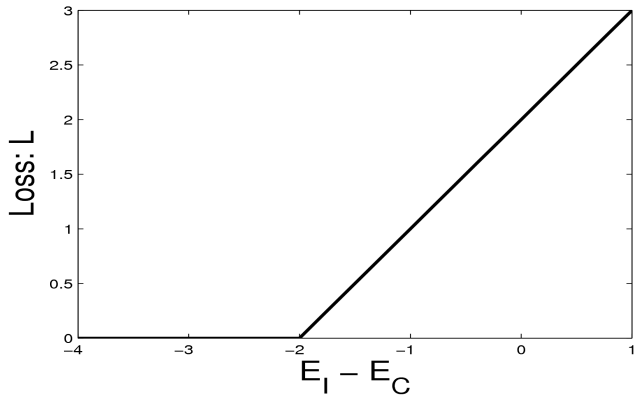
Definition 2 *Let Y be a continuous variable. Then for a training sample (X^i, Y^i), the* ***most offending incorrect answer*** *$\bar{Y}^i$ is the answer that has the lowest energy among all answers that are at least ϵ away from the correct answer:*

$$\bar{Y}^i = \text{argmin}_{Y \in \mathcal{Y}, \|Y - Y^i\| > \epsilon} E(W, Y, X^i). \quad (9)$$

Examples of Generalized Margin Losses

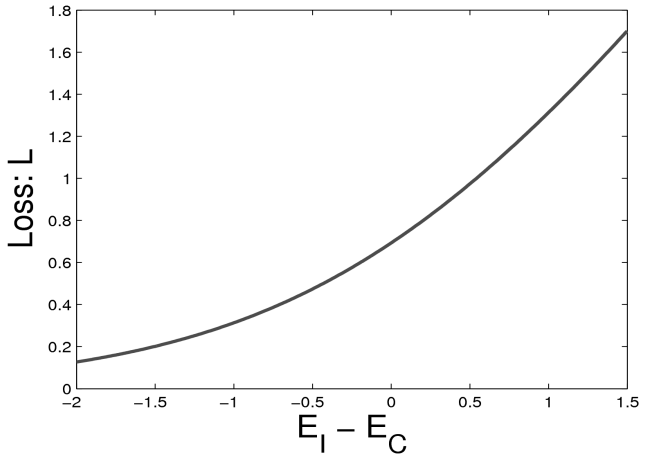
$$L_{\text{hinge}}(W, Y^i, X^i) = \max\left(0, m + E(W, Y^i, X^i) - E(W, \bar{Y}^i, X^i)\right),$$

- **Hinge Loss**
  - [Altun et al. 2003], [Taskar et al. 2003]
  - With the linearly-parameterized binary classifier architecture, we get linear SVMs



$$L_{\log}(W, Y^i, X^i) = \log\left(1 + e^{E(W,Y^i,X^i) - E(W,\bar{Y}^i,X^i)}\right).$$

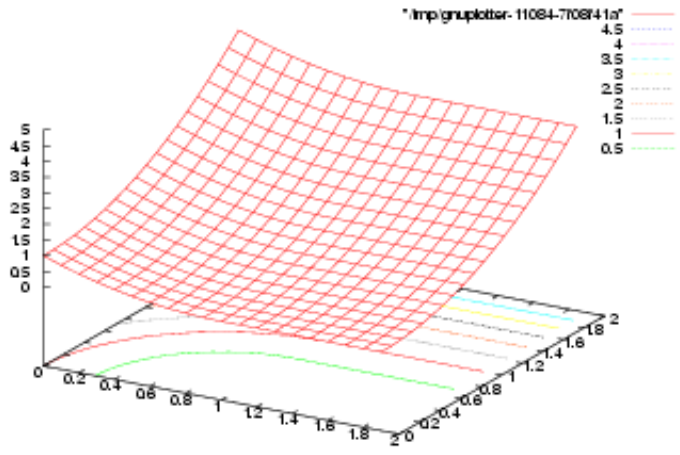
- **Log Loss**
  - "soft hinge" loss
  - With the linearly-parameterized binary classifier architecture, we get linear Logistic Regression

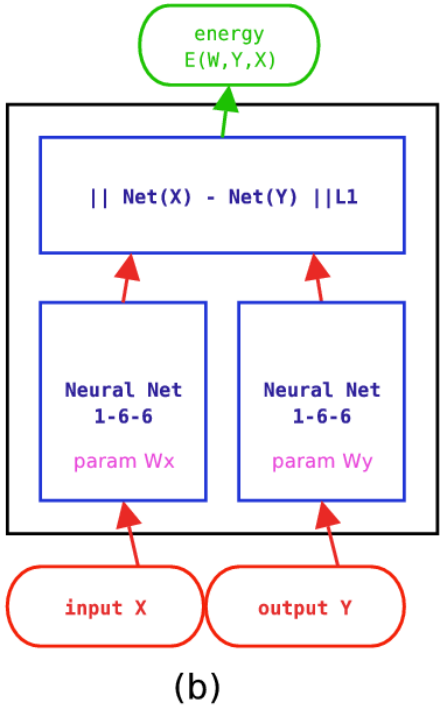


Examples of Margin Losses: Square-Square Loss

$$L_{\text{sq-sq}}(W, Y^i, X^i) = E(W, Y^i, X^i)^2 + \left(\max(0, m - E(W, \bar{Y}^i, X^i))\right)^2 .$$

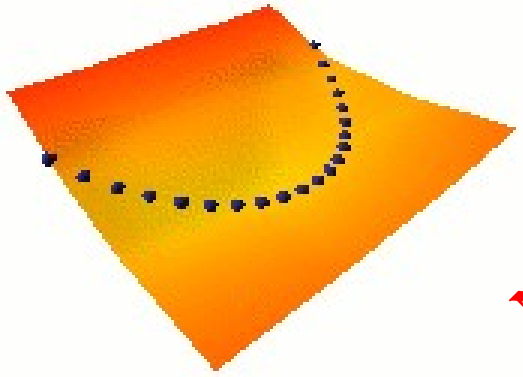
- **Square-Square Loss**
  - [LeCun-Huang 2005]
  - Appropriate for positive energy functions





(b)

Learning Y = X^2



NO COLLAPSE!!!

Other Margin-Like Losses

- **LVQ2 Loss** [Driancourt-Bottou 1991] for spoken word recognition

$$L_{\text{lvq2}}(W, Y^i, X^i) = \min\left(1, \max\left(0, \frac{E(W, Y^i, X^i) - E(W, \bar{Y}^i, X^i)}{\delta E(W, \bar{Y}^i, X^i)}\right)\right),$$

- **Minimum Classification Error Loss** [Juang, Chou, Lee 1997] for ASR

$$L_{\text{mce}}(W, Y^i, X^i) = \sigma\left(E(W, Y^i, X^i) - E(W, \bar{Y}^i, X^i)\right),$$

$$\sigma(x) = (1 + e^{-x})^{-1}$$

- **Square-Exponential Loss** [Osadchy, Miller, LeCun 2004] for face detection and pose estimation

$$L_{\text{sq-exp}}(W, Y^i, X^i) = E(W, Y^i, X^i)^2 + \gamma e^{-E(W, \bar{Y}^i, X^i)},$$

Negative Log-Likelihood Loss

- **Conditional probability of the samples (assuming independence)**

$$P(Y^1, \ldots, Y^P | X^1, \ldots, X^P, W) = \prod_{i=1}^{P} P(Y^i | X^i, W).$$

$$-\log \prod_{i=1}^{P} P(Y^i | X^i, W) = \sum_{i=1}^{P} -\log P(Y^i | X^i, W).$$

- **Gibbs distribution:** $$P(Y | X^i, W) = \frac{e^{-\beta E(W, Y, X^i)}}{\int_{y \in \mathcal{Y}} e^{-\beta E(W, y, X^i)}}.$$

$$-\log \prod_{i=1}^{P} P(Y^i | X^i, W) = \sum_{i=1}^{P} \beta E(W, Y^i, X^i) + \log \int_{y \in \mathcal{Y}} e^{-\beta E(W, y, X^i)}.$$

- **We get the NLL loss by dividing by P and Beta:**

$$\mathcal{L}_{\text{nll}}(W, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} \left(E(W, Y^i, X^i) + \frac{1}{\beta} \log \int_{y \in \mathcal{Y}} e^{-\beta E(W, y, X^i)} \right).$$

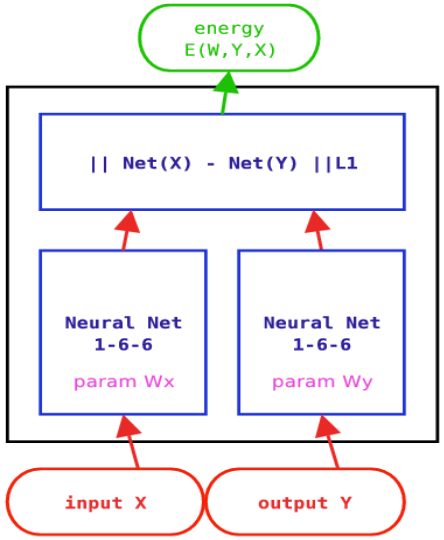
- **Reduces to the perceptron loss when Beta->infinity**

Negative Log-Likelihood Loss

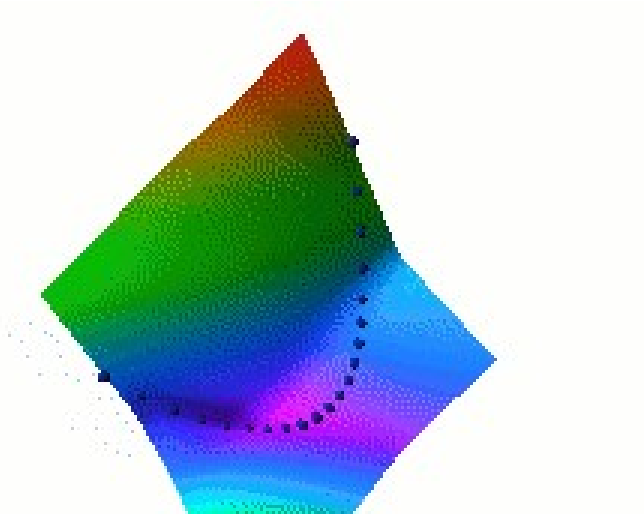
- Pushes down on the energy of the correct answer
- Pulls up on the energies of all answers in proportion to their probability

$$\mathcal{L}_{\text{nll}}(W, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} \left(E(W, Y^i, X^i) + \frac{1}{\beta} \log \int_{y \in \mathcal{Y}} e^{-\beta E(W, y, X^i)} \right).$$

$$\frac{\partial L_{\text{nll}}(W, Y^i, X^i)}{\partial W} = \frac{\partial E(W, Y^i, X^i)}{\partial W} - \int_{Y \in \mathcal{Y}} \frac{\partial E(W, Y, X^i)}{\partial W} P(Y | X^i, W),$$



(b)



Negative Log-Likelihood Loss: Binary Classification

- **Binary Classifier Architecture:**

$$\mathcal{L}_{\text{nll}}(W, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} \left[-Y^i G_W(X^i) + \log \left(e^{Y^i G_W(X^i)} + e^{-Y^i G_W(X^i)} \right) \right].$$

$$\mathcal{L}_{\text{nll}}(W, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} \log \left(1 + e^{-2Y^i G_W(X^i)} \right),$$

- **Linear Binary Classifier Architecture:**

$$\mathcal{L}_{\text{nll}}(W, \mathcal{S}) = \frac{1}{P} \sum_{i=1}^{P} \log \left(1 + e^{-2Y^i W^T \Phi(X^i)} \right).$$

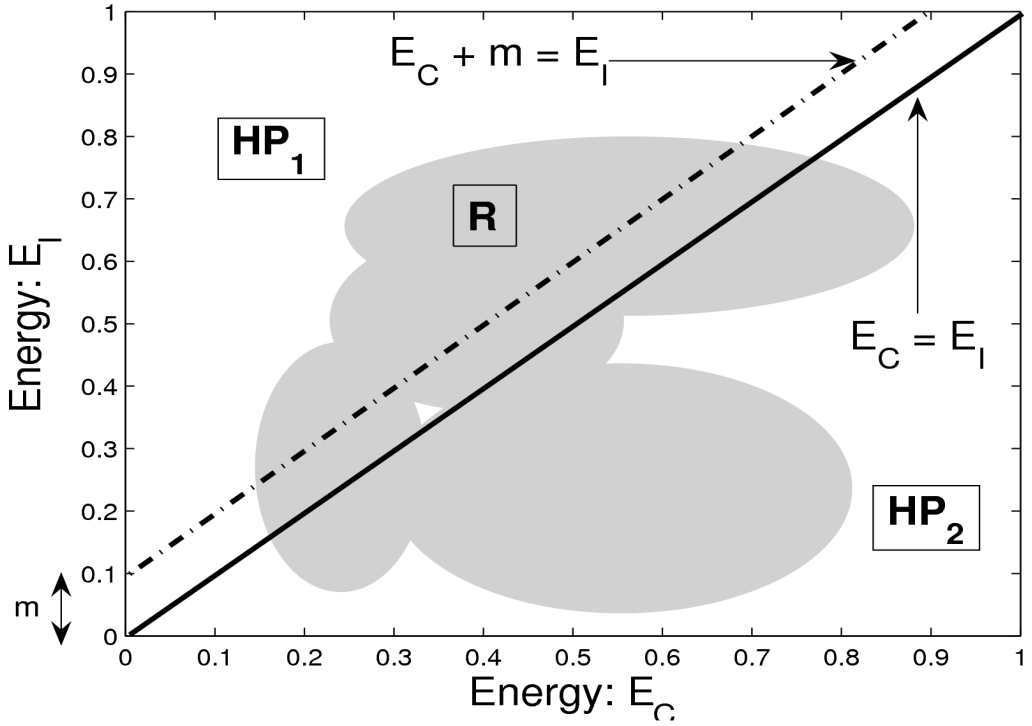
- **Learning Rule: logistic regression**

Negative Log-Likelihood Loss

- **Used extensively under different names (e.g. Maximum Mutual Information, Maximum Entropy.....).**
  - [Bengio et al 1992-1993], [Haffner 1993], [Bourlard 1994] continuous speech recognition.
  - [Bengio & LeCun 1994] on-line handwriting recognition
  - [LeCun et al. 1998] off-line handwriting recognition
  - [Lafferty et al. 2001] CRF (but used iterative scaling instead of the more efficient stochastic gradient descent method).
  - Used by almost every single discriminative training method for automatic speech recognition since the mid 90's.

What Makes a "Good" Loss Function

- **Good loss functions make the machine produce the correct answer**
  - Avoid collapses and flat energy surfaces



Sufficient Condition on the Loss

Let (X^i, Y^i) be the i^{th} training example and m be a positive margin. Minimizing the loss function L will cause the machine to satisfy $E(W, Y^i, X^i) < E(W, Y, X^i) - m$ for all $Y \neq Y^i$, if there exists at least one point (e_1, e_2) with $e_1 + m < e_2$ such that for all points (e'_1, e'_2) with $e'_1 + m \geq e'_2$, we have

$$Q_{[E_y]}(e_1, e_2) < Q_{[E_y]}(e'_1, e'_2),$$

where $Q_{[E_y]}$ is given by

$$L(W, Y^i, X^i) = Q_{[E_y]}(E(W, Y^i, X^i), E(W, \bar{Y}^i, X^i)).$$

What Make a "Good" Loss Function

- **Good and bad loss functions**

Loss (equation #)	Formula	Margin
energy loss	$E(W, Y^i, X^i)$	none
perceptron	$E(W, Y^i, X^i) - \min_{Y \in \mathcal{Y}} E(W, Y, X^i)$	0
hinge	$\max\left(0, m + E(W, Y^i, X^i) - E(W, \bar{Y}^i, X^i)\right)$	m
log	$\log\left(1 + e^{E(W,Y^i,X^i) - E(W,\bar{Y}^i,X^i)}\right)$	> 0
LVQ2	$\min\left(M, \max(0, E(W, Y^i, X^i) - E(W, \bar{Y}^i, X^i)\right)$	0
MCE	$\left(1 + e^{-\left(E(W,Y^i,X^i) - E(W,\bar{Y}^i,X^i)\right)}\right)^{-1}$	> 0
square-square	$E(W, Y^i, X^i)^2 - \left(\max(0, m - E(W, \bar{Y}^i, X^i))\right)^2$	m
square-exp	$E(W, Y^i, X^i)^2 + \beta e^{-E(W,\bar{Y}^i,X^i)}$	> 0
NLL/MMI	$E(W, Y^i, X^i) + \frac{1}{\beta} \log \int_{y \in \mathcal{Y}} e^{-\beta E(W,y,X^i)}$	> 0
MEE	$1 - e^{-\beta E(W,Y^i,X^i)} / \int_{y \in \mathcal{Y}} e^{-\beta E(W,y,X^i)}$	> 0

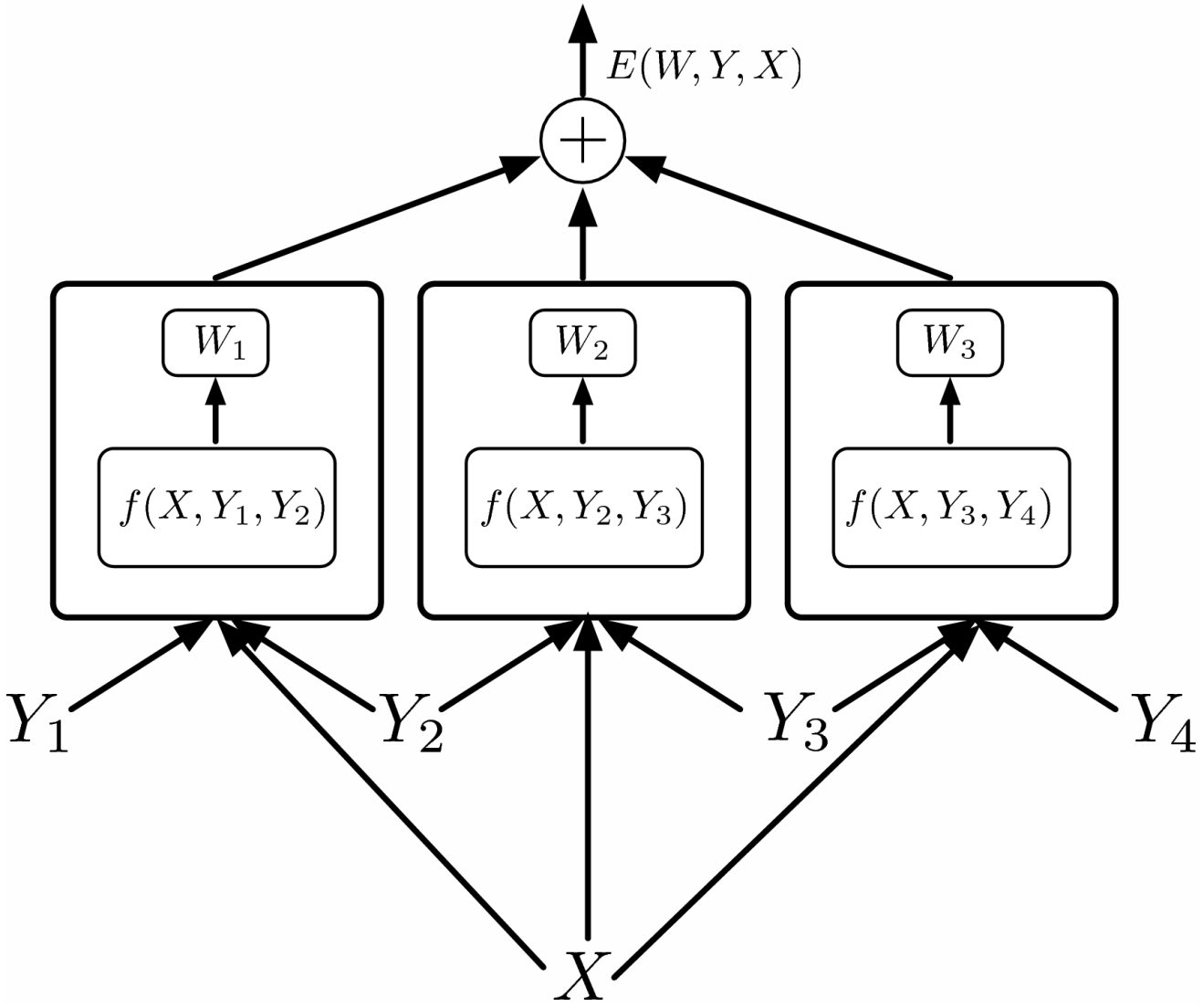
Advantages/Disadvantages of various losses

- **Loss functions differ in how they pick the point(s) whose energy is pulled up, and how much they pull them up**
- **Losses with a log partition function in the contrastive term pull up all the bad answers simultaneously.**
  - This may be good if the gradient of the contrastive term can be computed efficiently
  - This may be bad if it cannot, in which case we might as well use a loss with a single point in the contrastive term
- **Variational methods pull up many points, but not as many as with the full log partition function.**
- **Efficiency of a loss/architecture: how many energies are pulled up for a given amount of computation?**
  - The theory for this is to be developed

Shallow Factors / Deep Graph

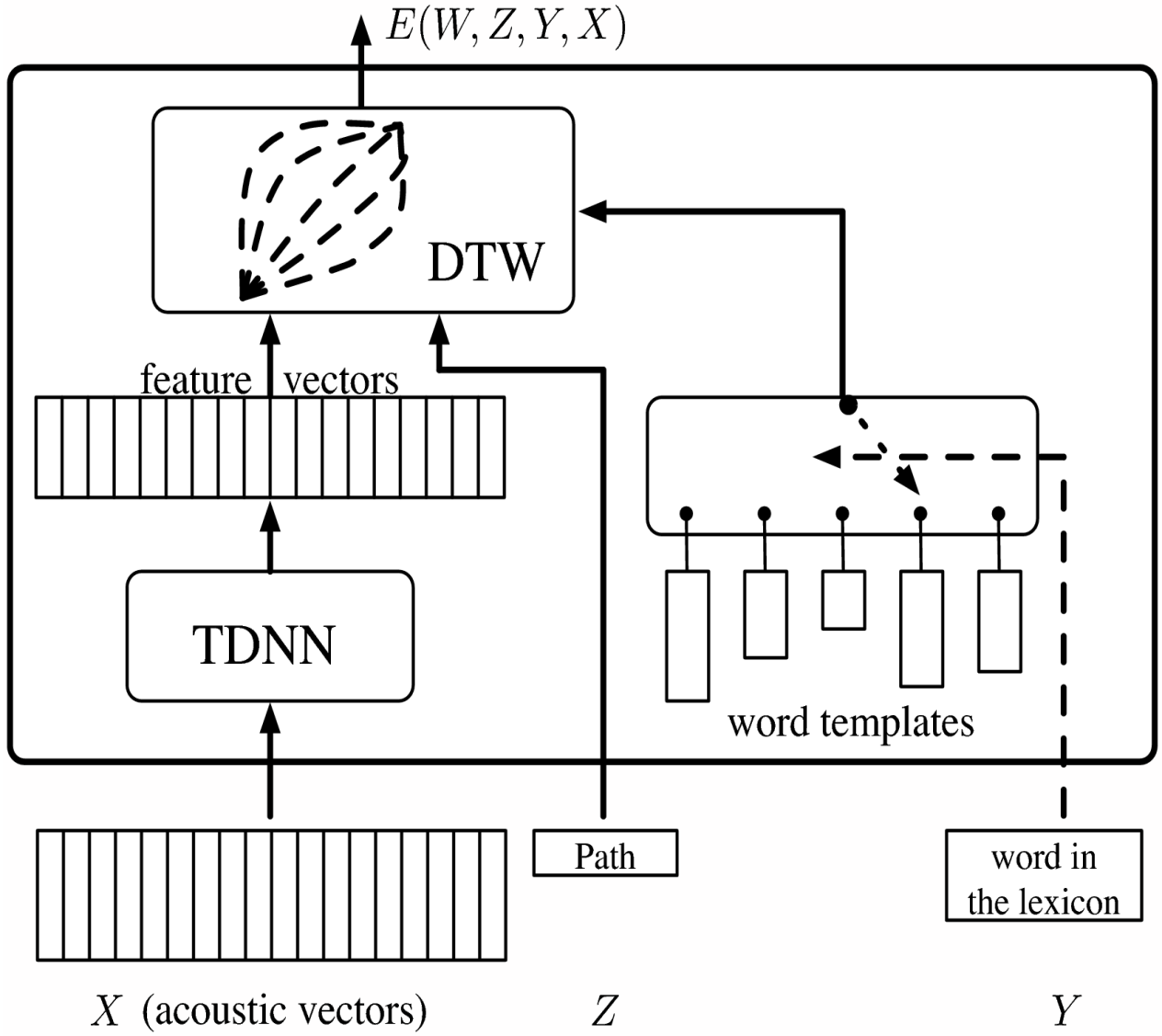
- **Linearly Parameterized Factors (shallow factors)**

- **with the NLL Loss :**
  - Lafferty's Conditional Random Field
  - Kumar&Hebert's DRF.
- **with Hinge Loss:**
  - Taskar's Max Margin Markov Nets
- **with Perceptron Loss**
  - Collins's sequence labeling model
- **With Log Loss:**
  - Altun/Hofmann sequence labeling model



Deep Factors / Deep Graph: ASR with TDNN/DTW

- **Trainable Automatic Speech Recognition system with convolutional nets (TDNN) and dynamic time warping (DTW)**
- **Training the feature extractor as part of the whole process.**
- **with the LVQ2 Loss :**
  - Driancourt and Bottou's speech recognizer (1991)
- **with NLL:**
  - Bengio's speech recognizer (1992)
  - Haffner's speech recognizer (1993)

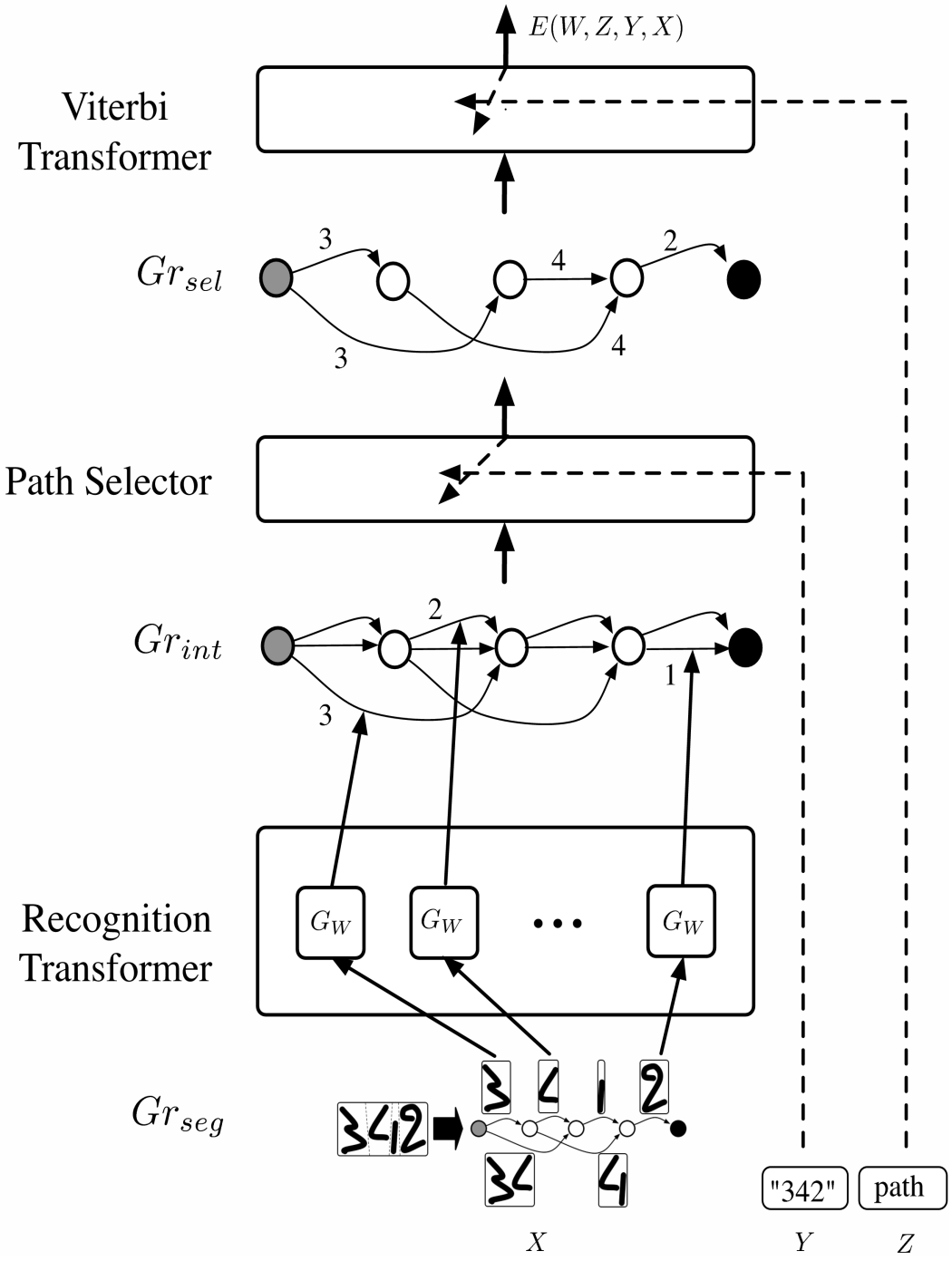


Deep Factors / Deep Graph: ASR with TDNN/HMM

- **Discriminative Automatic Speech Recognition system with HMM and various acoustic models**
  - Training the acoustic model (feature extractor) and a (normalized) HMM in an integrated fashion.
- **With Minimum Empirical Error loss**
  - Ljolje and Rabiner (1990)
- **with NLL:**
  - Bengio (1992)
  - Haffner (1993)
  - Bourlard (1994)
- **With MCE**
  - Juang et al. (1997)
- **Late normalization scheme (un-normalized HMM)**
  - Bottou pointed out the **label bias problem** (1991)
  - Denker and Burges proposed a solution (1995)

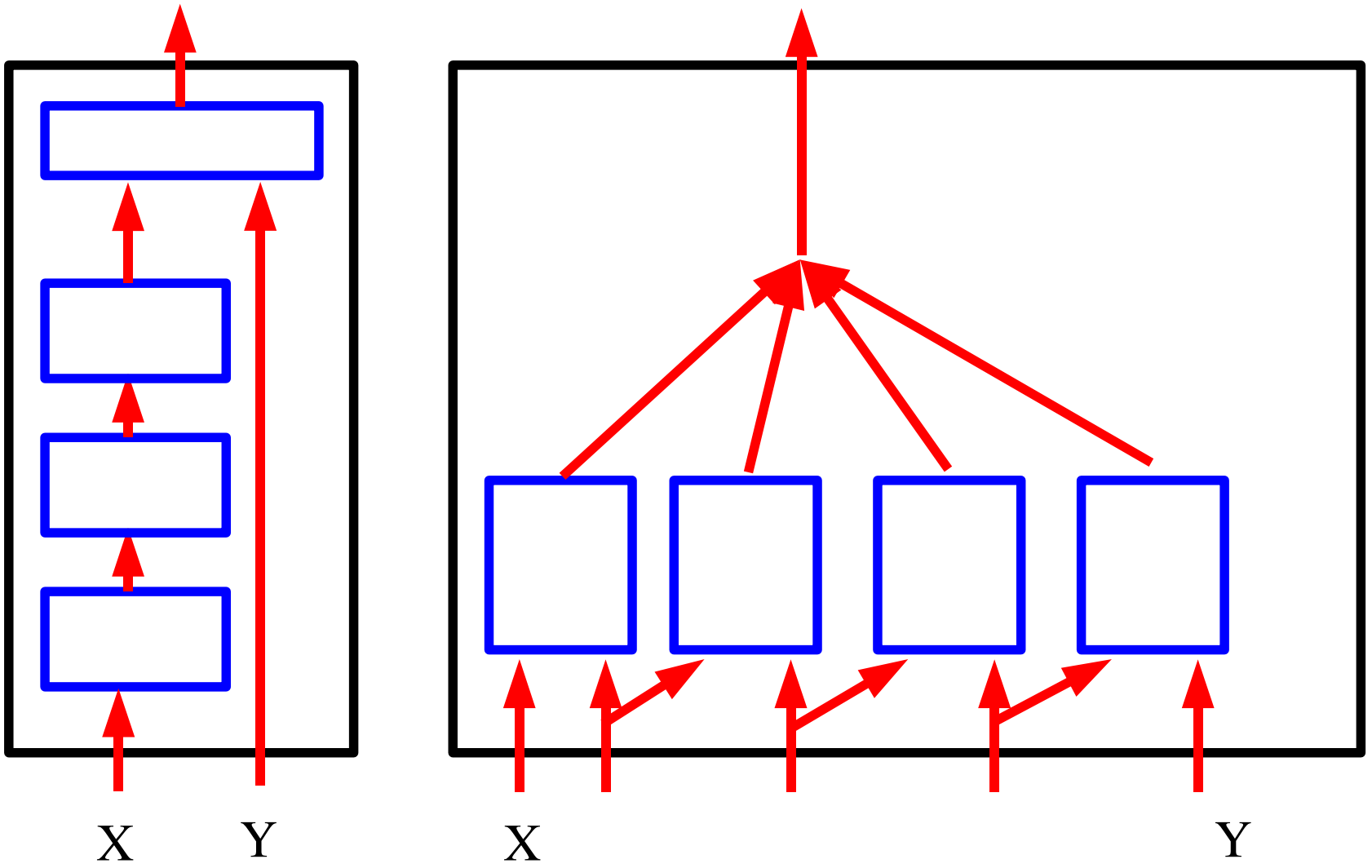
Really Deep Factors / Really Deep Graph

- **Handwriting Recognition with Graph Transformer Networks**
- **Un-normalized hierarchical HMMs**
  - Trained with Perceptron loss [LeCun, Bottou, Bengio, Haffner 1998]
  - Trained with NLL loss [Bengio, LeCun 1994], [LeCun, Bottou, Bengio, Haffner 1998]
- **Answer = sequence of symbols**
- **Latent variable = segmentation**



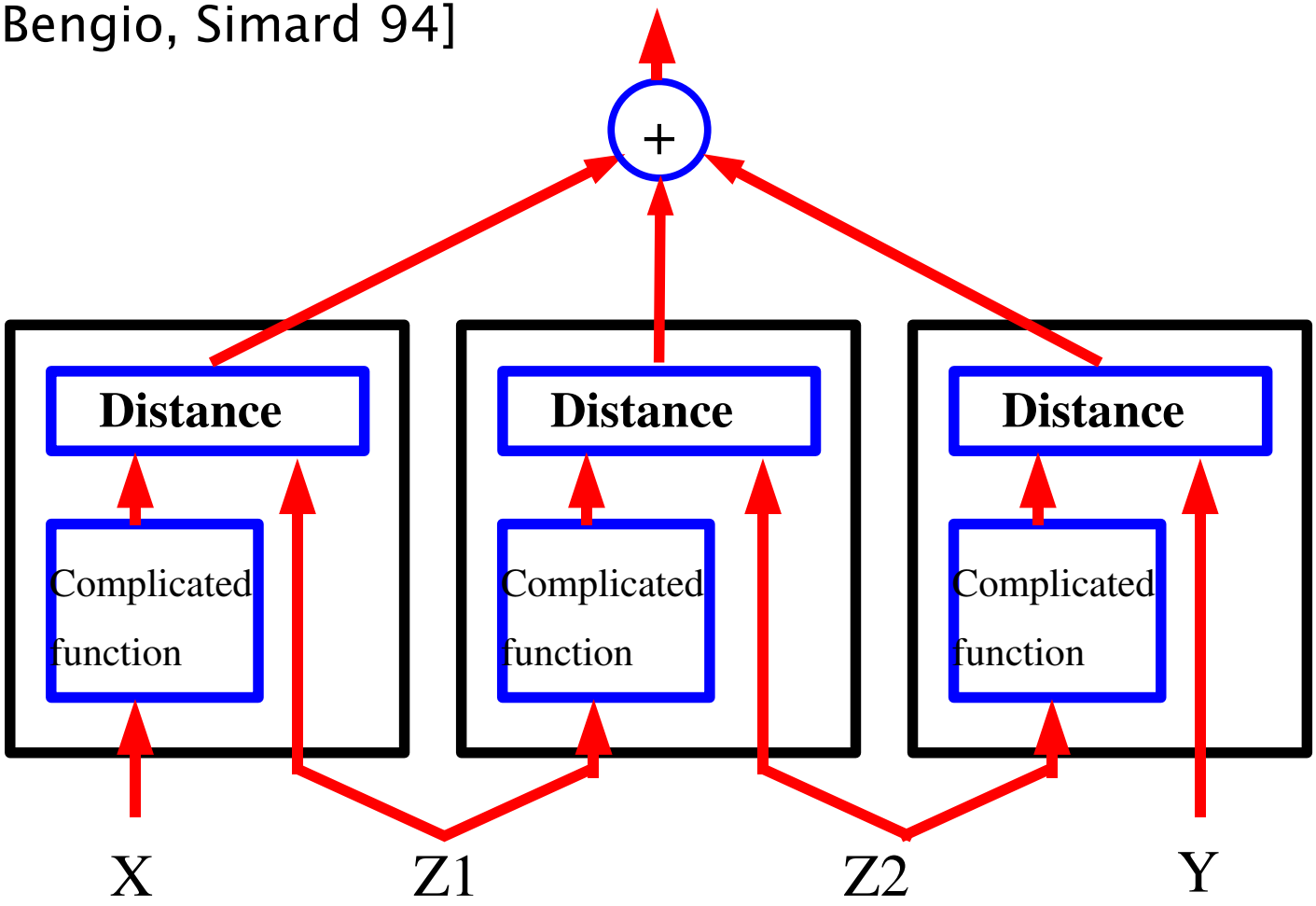
Two types of "deep" architectures

- Factors are deep / graph is deep



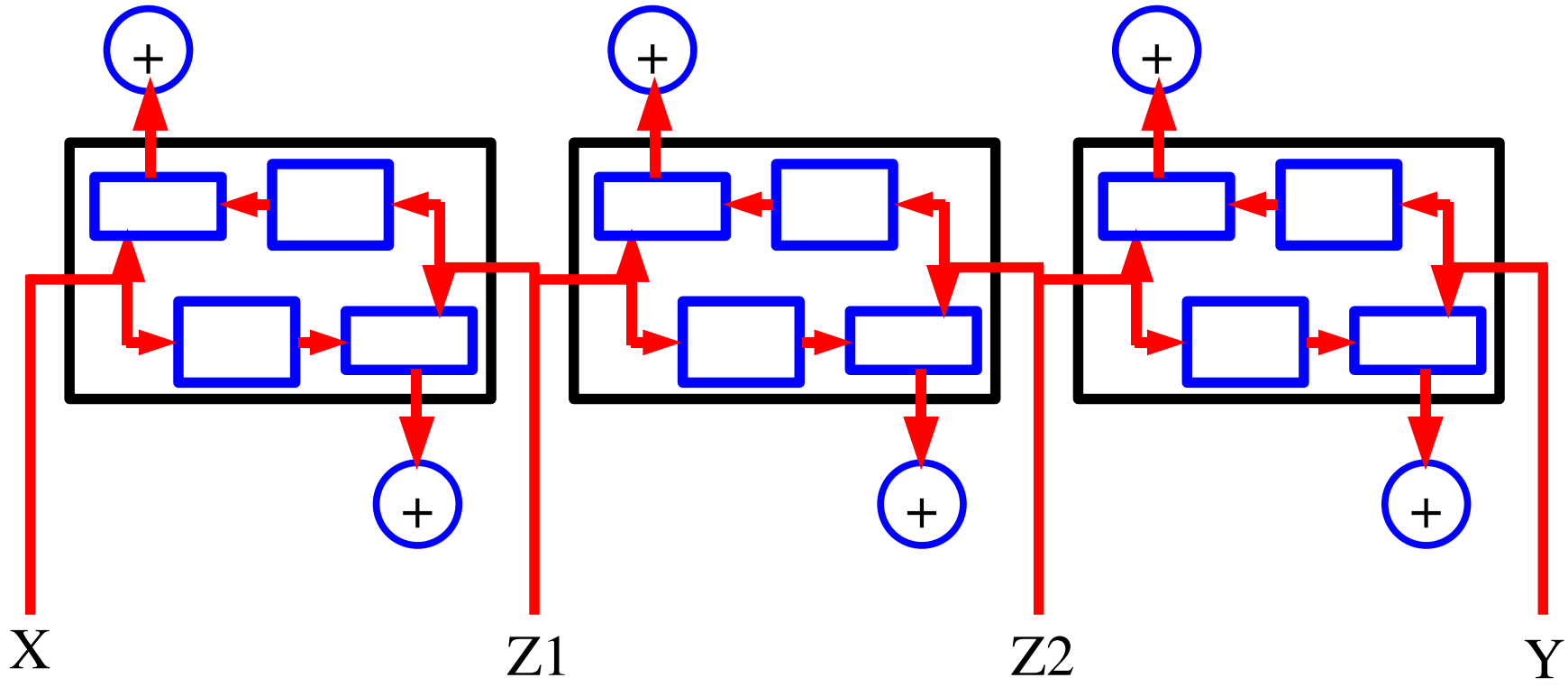
Feed-Forward Deep Belief Net

- **Equivalent to Backprop**
  - Energy-Based formulation gives an EM-like training procedure for multilayer nets.
  - "target prop" [LeCun 85], [Grossman 88], [Rohwer 89], [Krogh et al 89], [Bengio, Simard 94]



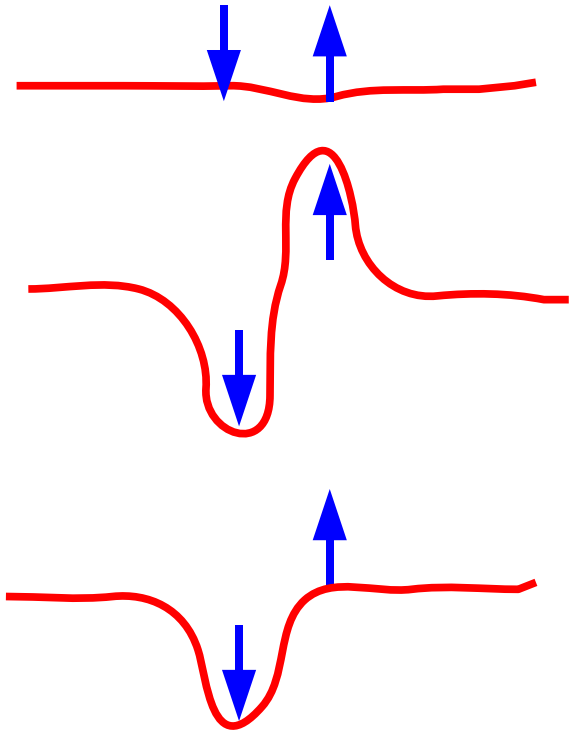
Bi-Directional Deep Belief Net

- **Easy Inference Both Ways**
  - Stacked RBMs belong to this family



Efficiency of Energy-Based Learning

- **How many points must we pull up for the energy surface to take the right shape?**
  - Highly malleable energy surfaces require a lot of pulling-up
  - More rigid surface may require less pulling



Convolutional Conditional PoE for Image Denoising

- Somewhat similar to the Field of Experts [Roth & Black CVPR 2005]

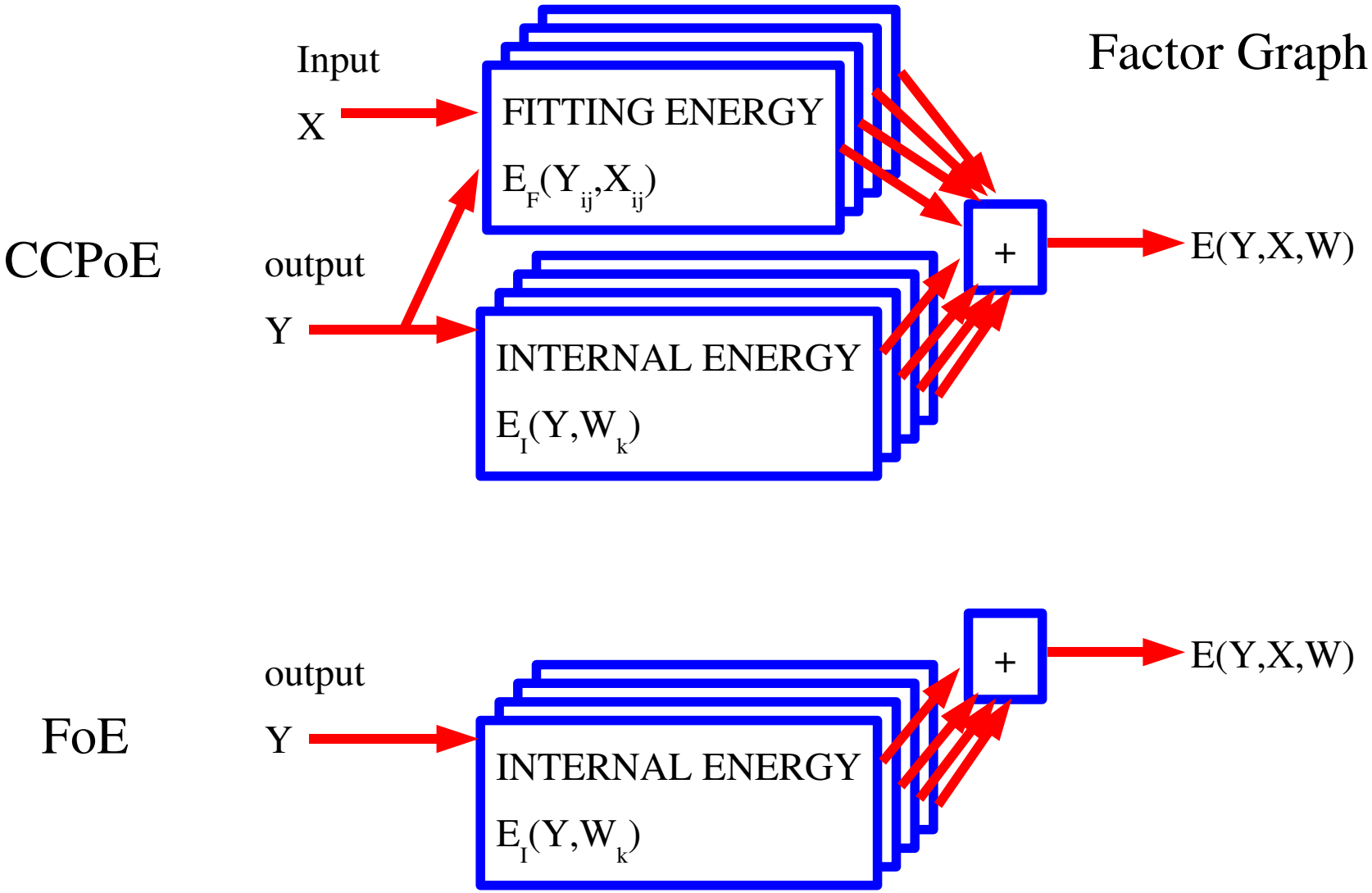
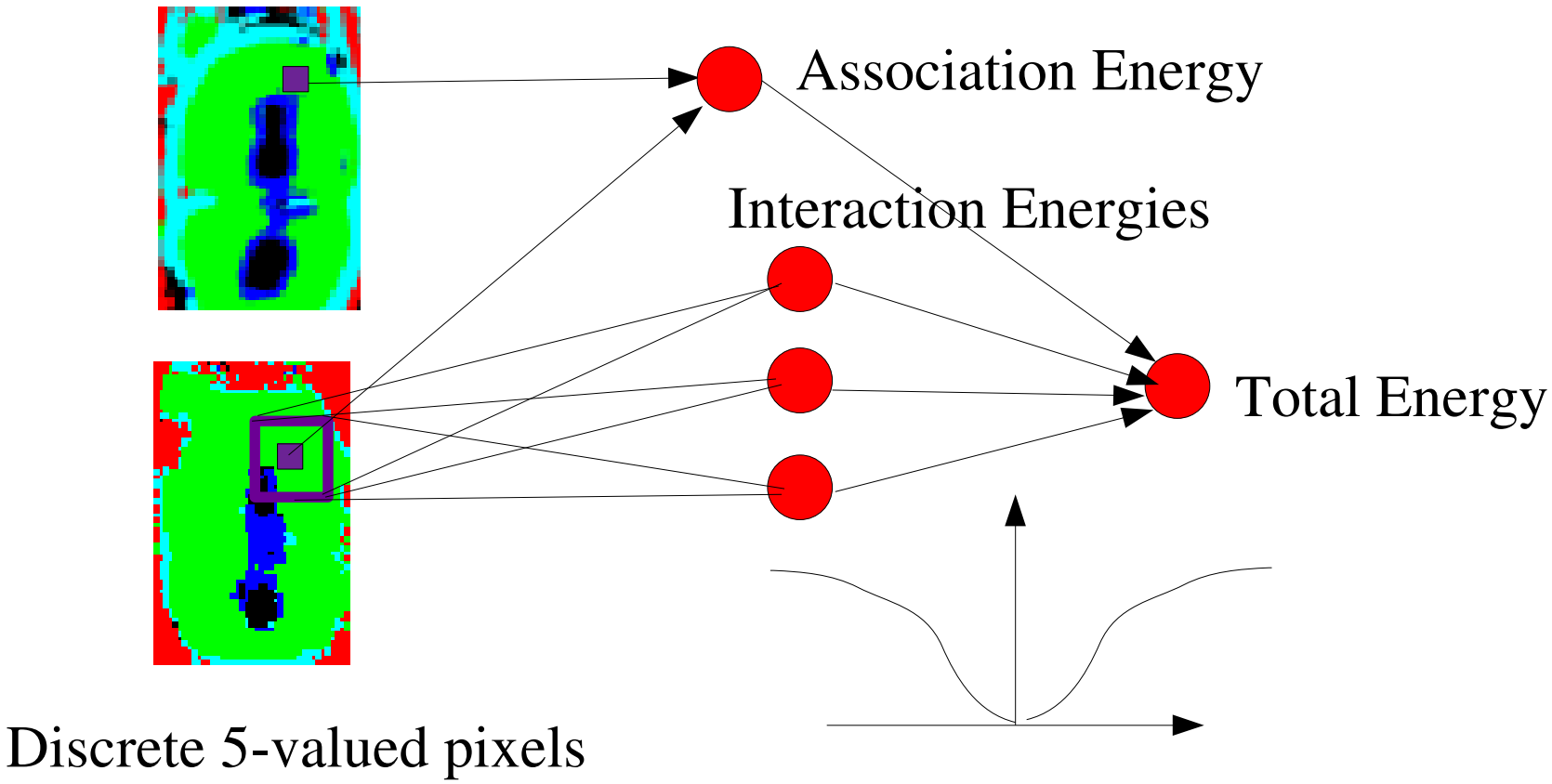


Image Segmentation with Local Consistency Constraints

[Teh, Welling, Osindero, Hinton, 2001], [Kumar, Hebert 2003], [Zemel 2004]

- **Learn local consistency constraints with an Energy-Based Model so as to clean up images produced by the segmentor.**



C. Elegans Embryo Phenotyping

- **Analyzing results for Gene Knock-Out Experiments**

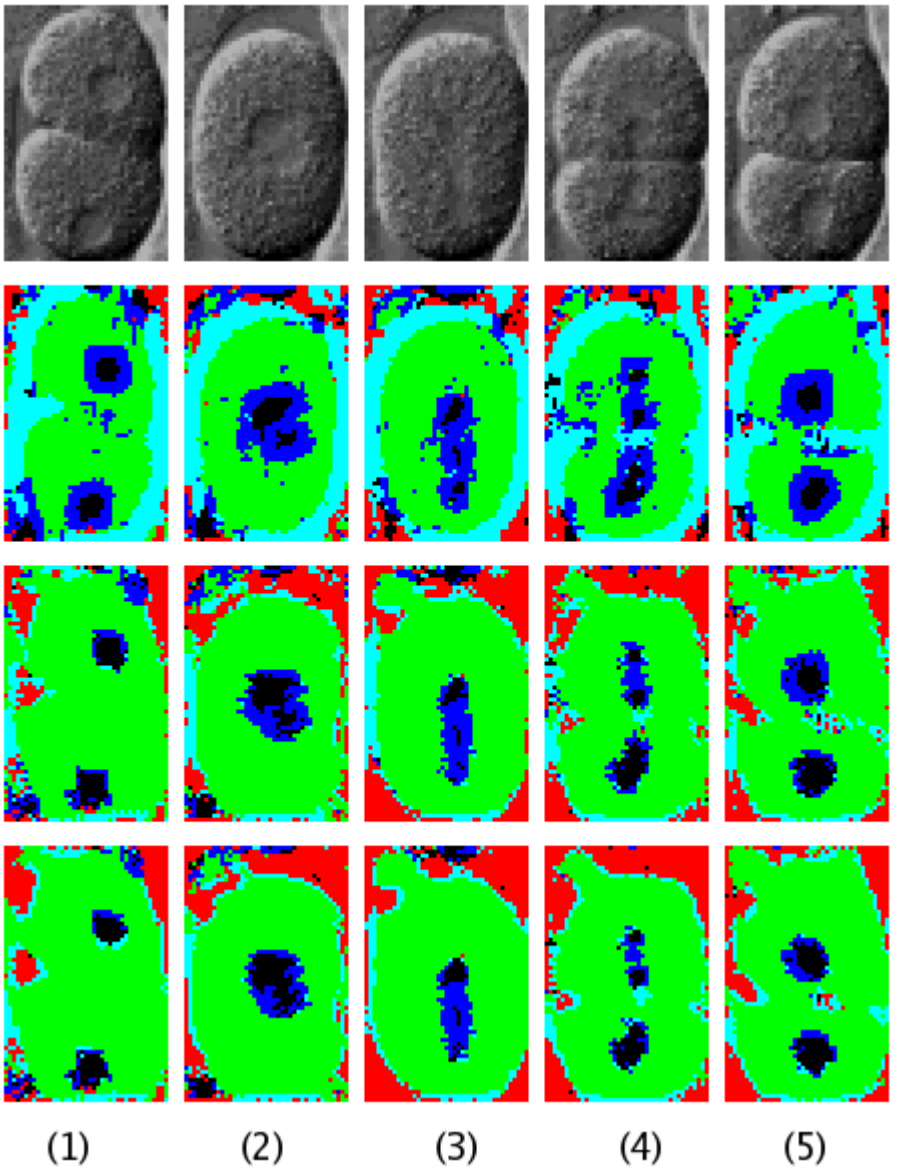
Original Images

Segmentation #1

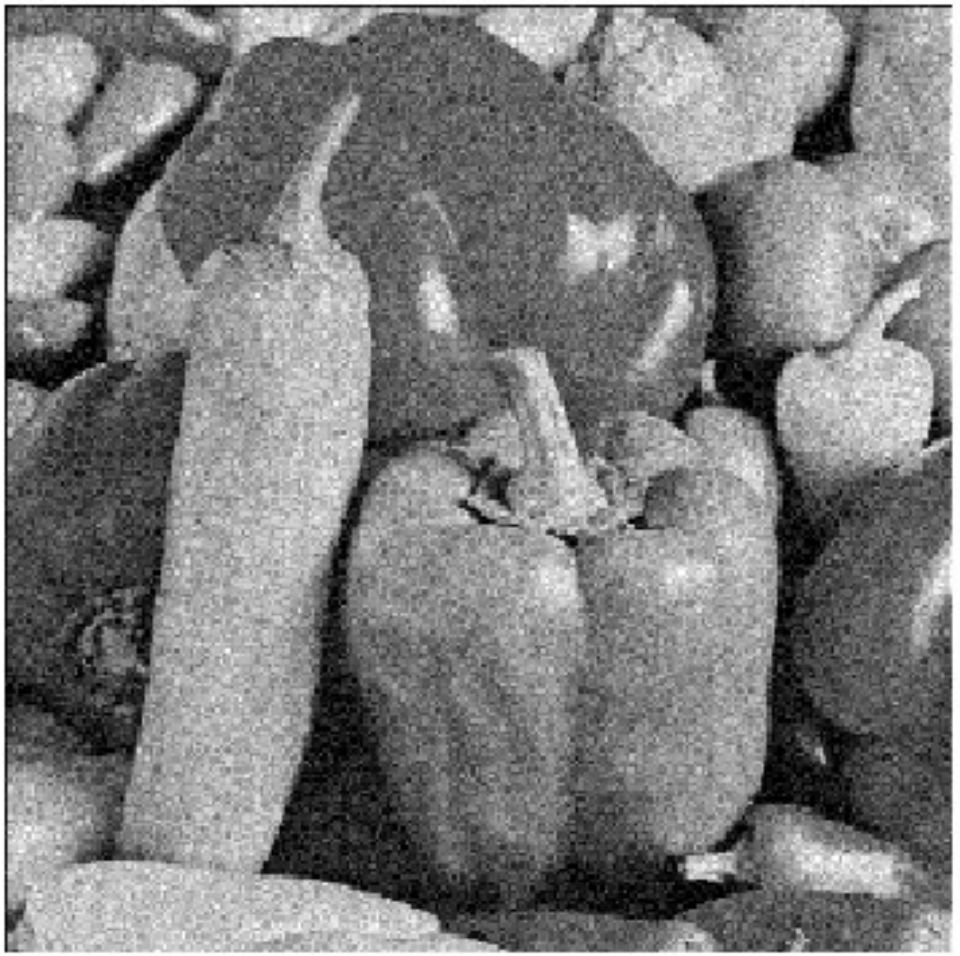
Segmentation #2

CCPoE

Cleanup



Convolutional Conditional PoE for Image Denoising



Noisy peppers PSNR=22.10

CCPoE PSNR=30.40

Convolutional Conditional PoE for Image Denoising

FoE PSNR=30.41

(Roth & Black report 30.58)

CCPoE PSNR=30.40

Conclusion

- CRF-like things have been around since 1991 in the speech and handwriting communities
- Un-normalized energy-based models were proposed in 1995 and 1998, but no characterization of "good" loss functions were available until now.
- Tutorial paper on energy-based learning available at http://yann.lecun.com/exdb/publis/
- We don't need no stinkin' partition function