

---

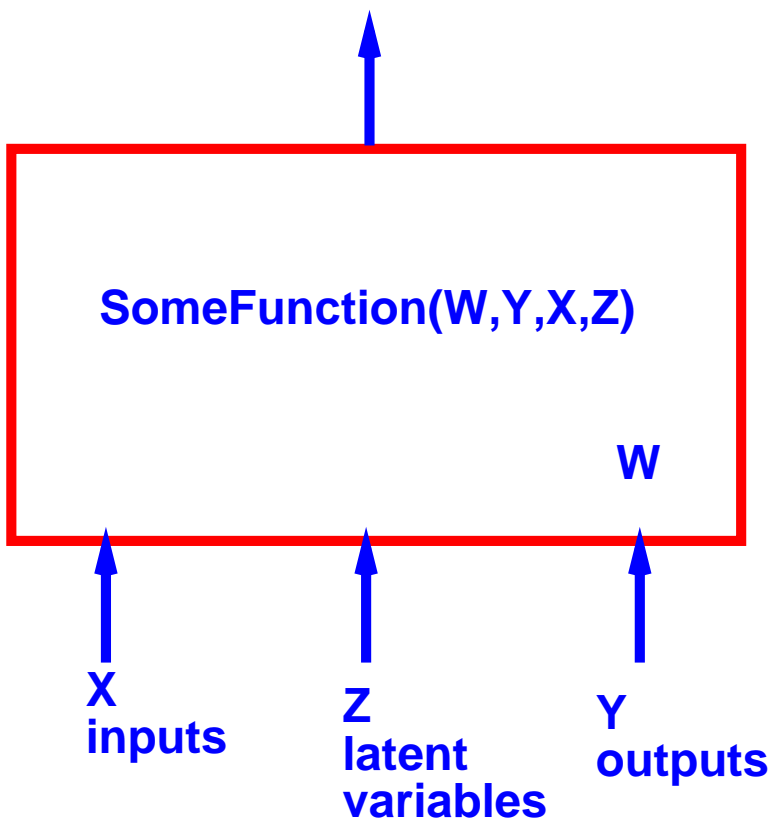
# Loss Functions for Energy-Based Models With Applications to Object Recognition.

Yann LeCun, Fu Jie Huang

The Courant Institute,  
New York University  
<http://yann.lecun.com>

# What is a Model?

---



- **X**: input variables. Always observed
- **Y**: output variables. Not observed, except on training samples.
- **Z**: latent variables. Never observed
- **SomeFunction(Y, X, Z)**: model. Measures the compatibility between the values of  $X$ ,  $Y$ , and  $Z$ .

Inference: find  $Y$  (and  $Z$ ) that are most compatible with an observed  $X$ .

# What is a Probabilistic Model?

many probabilistic models parameterize the joint distribution over  $X, Y$ , and  $Z$  (e.g. graphical models).

- “Causal” generative models parameterize  $P(X|Y)$

- Cond. prob. (without latent vars):

$$P(Y|X, W) = \frac{P(W, Y, X)}{\int_y P(W, y, X)}$$

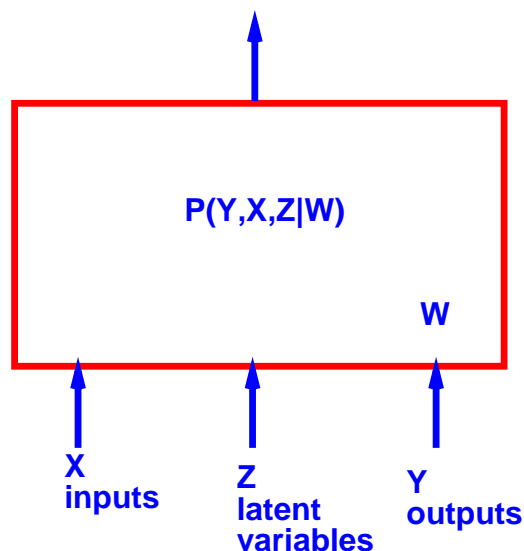
- Cond. prob. (with latent vars):

$$P(Y|X, W) = \frac{\int_z P(W, Y, z, X)}{\int_{yz} P(W, y, z, X)}$$

- $P(Y|X, W)$  must be normalized over  $Y$ .

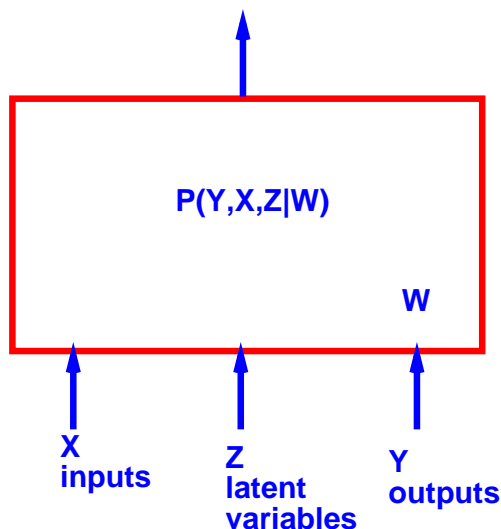
**Inference/Decision Making.** find  $Y$  with largest prob:  $\check{Y} = \max_{y \in \{Y\}} P(Y, X, W)$   
(note: we don't need normalization for decision making).

$$P(Y|X, W) = \text{SUM}_z P(Y, X, z|W) / \text{SUM}_{yz} P(y, X, z|W)$$



# Training Probabilistic Models

$$P(Y|X,W) = \frac{\sum_z P(Y,X,z|W)}{\sum_{yz} P(y,X,z|W)}$$



Training set:  $\mathcal{S} = \{(X^1, Y^1), \dots, (X^p, Y^p)\}$ .

**Criterion:** Max Likelihood

$$\prod_i P(Y^i | X^i, W) = \prod_i \frac{\int_z P(W, Y, z, X)}{\int_{yz} P(W, y, z, X)}$$

**Loss Function:** Minimum Negative Log Likelihood

$$\mathcal{L}(W, \mathcal{S}) = -\log \prod_i P(Y^i | X^i, W)$$

$$\mathcal{L}(W, \mathcal{S}) = \sum_i -\log \left( \int_z P(W, Y, z, X) \right) + \log \left( \int_{yz} P(W, y, z, X) \right)$$

# What's good about probabilistic models?

---

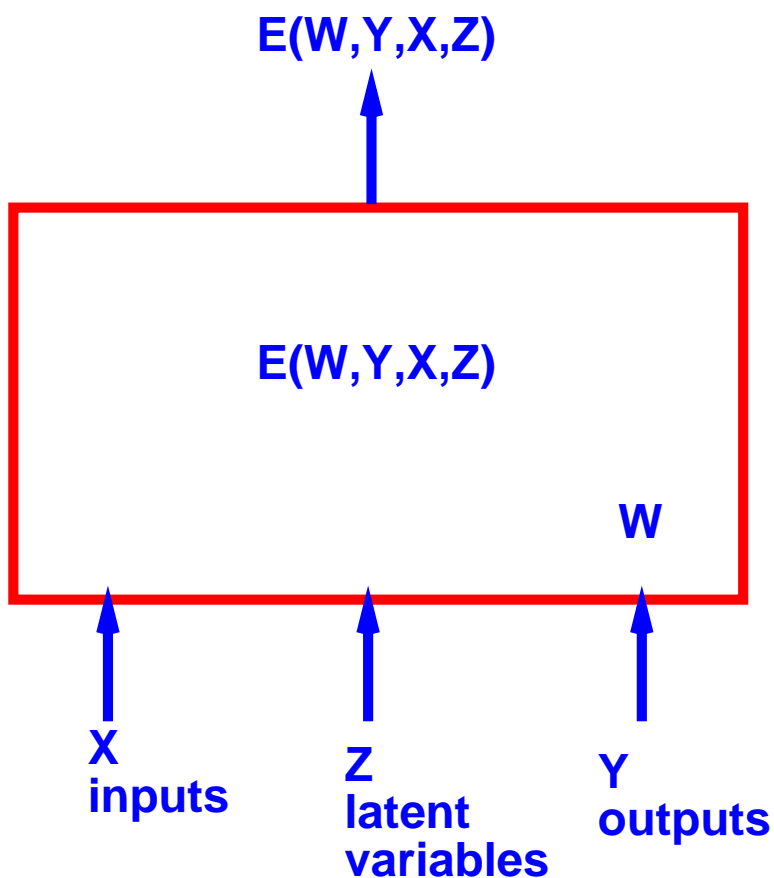
- Compositionality
- A well-justified loss function: the Negative Log Likelihood Loss.
- Neat Tricks (e.g. EM)

# What's bad about probabilistic models?

---

- with generative models, normalization over  $X$  is useless, difficult, and restrictive. we should never, *ever*, **EVER** have to normalize anything over  $X$  (normalization in high dimensional spaces is silly).
- only a **tiny number of models** are pre-normalized (e.g. Gaussians).
- only a **very small number of models** have tractable partition functions (easily normalizable).
- **many models** have intractable partition functions.
- **most models** are not even normalizable.
- If we only care about making good decisions (picking the best  $Y$ ), why should we have to estimate the correct  $P(Y|X)$  over the full range of  $Y$ ? We merely need  $P(Y|X)$  to have maxima at the right places.
- We have to come up with embarrassing justifications for fudge factors that make things work, but break the normalization. For example  $P_{\text{appearance}}^\alpha \times P_{\text{shape}}$  in image recognition, or  $P_{\text{transition}}^\alpha \times P_{\text{emission}}$  in speech recognition.
- **Learning by maximizing the likelihood solves a more complex problem than we have to.**

# Energy-Based Models



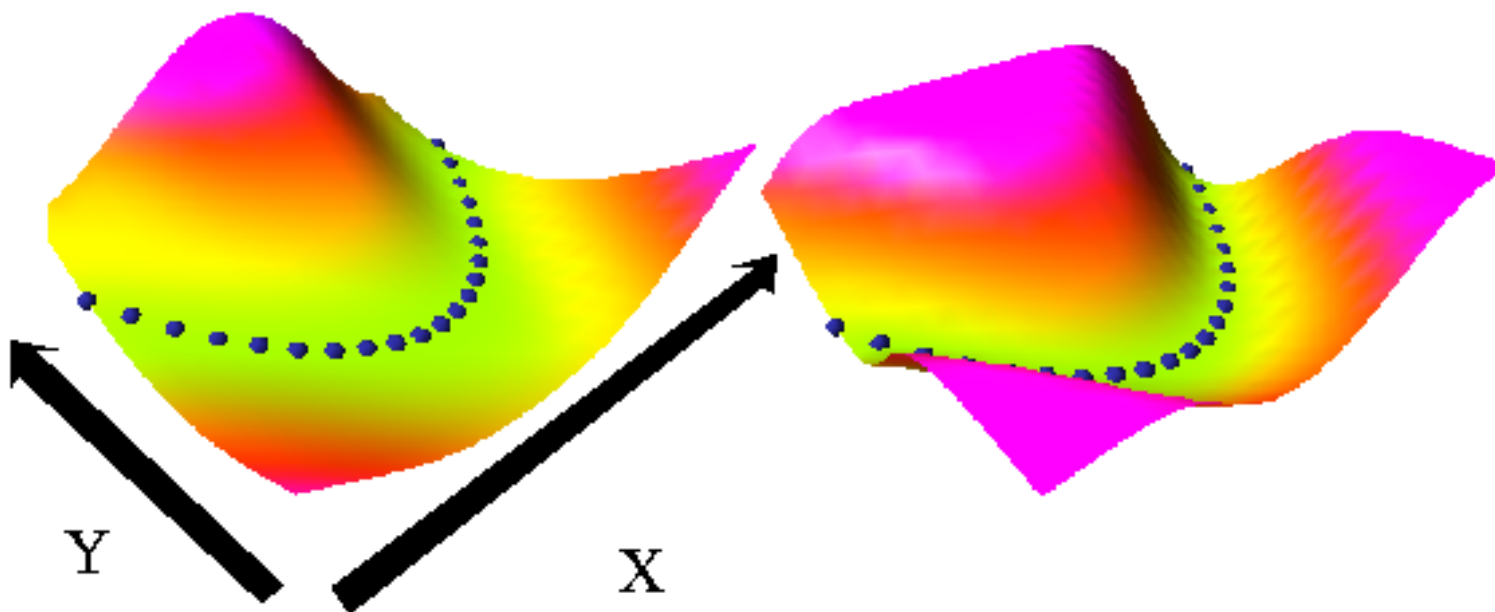
- Associate a scalar **energy**  $E(W, Y, X, Z)$  to configurations of  $(Y, X, Z)$
- $W$  is the parameter vector to be learned.
- **Inference (without latent vars)** consists in comparing energies:  $Y$  is better than  $Y'$  if  $E(W, Y, X, Z) < E(W, Y', X, Z)$ .
- **Inference (with latent vars)**:  $Y$  is better than  $Y'$  if  $\min_z E(W, Y, X, z) < \min_z E(W, Y', X, z)$ .

**Decision making (without latent vars):**  $\check{Y} = \min_{y \in \{Y\}} E(W, y, X)$

**Decision making (with latent vars):**  $\check{Y} = \min_{y \in \{Y\}, z \in \{Z\}} E(W, y, X, z)$

# EBM Energy Surfaces

---



Examples: An EBM that computes  $Y = X^2$ .

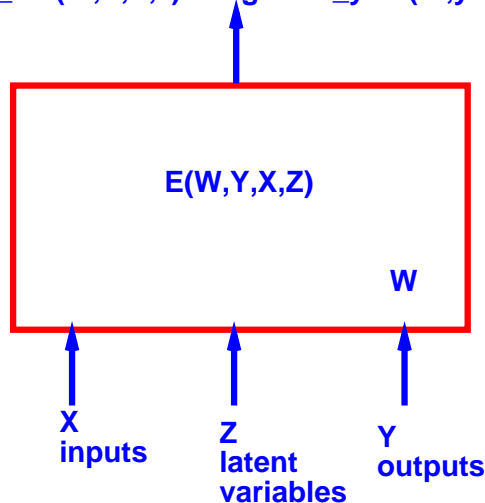
**On the left:**  $E(Y, X)$  is quadratic in  $Y$ . It corresponds to a Gaussian model of  $P(Y|X)$ .

**On the right:**  $E(Y, X)$  is saturated. Although it gives the same answers as the EBM on the left, it has no probabilistic equivalent because  $\int_y \exp(-E(Y, X))$  does not converge.



# Probabilistic Models from Energy-Based Models

$$-\log P(Y|X,W) = \text{SUM}_z E(W,Y,X,z) + \log \text{SUM}_{y,z} E(W,yX,z)$$



- Any joint probability model can be approached as close as we want by an equivalent EBM. If  $P(Y, X, Z)$  is non-zero everywhere:  
 $E(Y, X, Z) = C - \frac{1}{\beta} \log P(Y, X, Z)$  where  $C$  is an arbitrary constant and  $\beta$  a strictly positive constant.
- not all EBMs can be turned into a probabilistic model. Only those for which  $\int_y \exp(-\beta E(W, y, X))$  converges:

$$P(Y|X) = \frac{\exp(-\beta E(W, Y, X))}{\int_y \exp(-\beta E(W, y, X))}$$

Any single probabilistic model will have many equivalent EBMs when it comes to comparison-based inference or decision. *Because many energy surfaces have minima at the same places.*

**We have a lot more flexibility when building EBMs**

# What's good/bad about EBM?

---

## What's bad about EBMs:

- There is no simple compositionality...
- ... but we don't care because we are going to train our whole system *end-to-end*.  
With *end-to-end learning*, we do not need compositionality.

## What's good about EBMs:

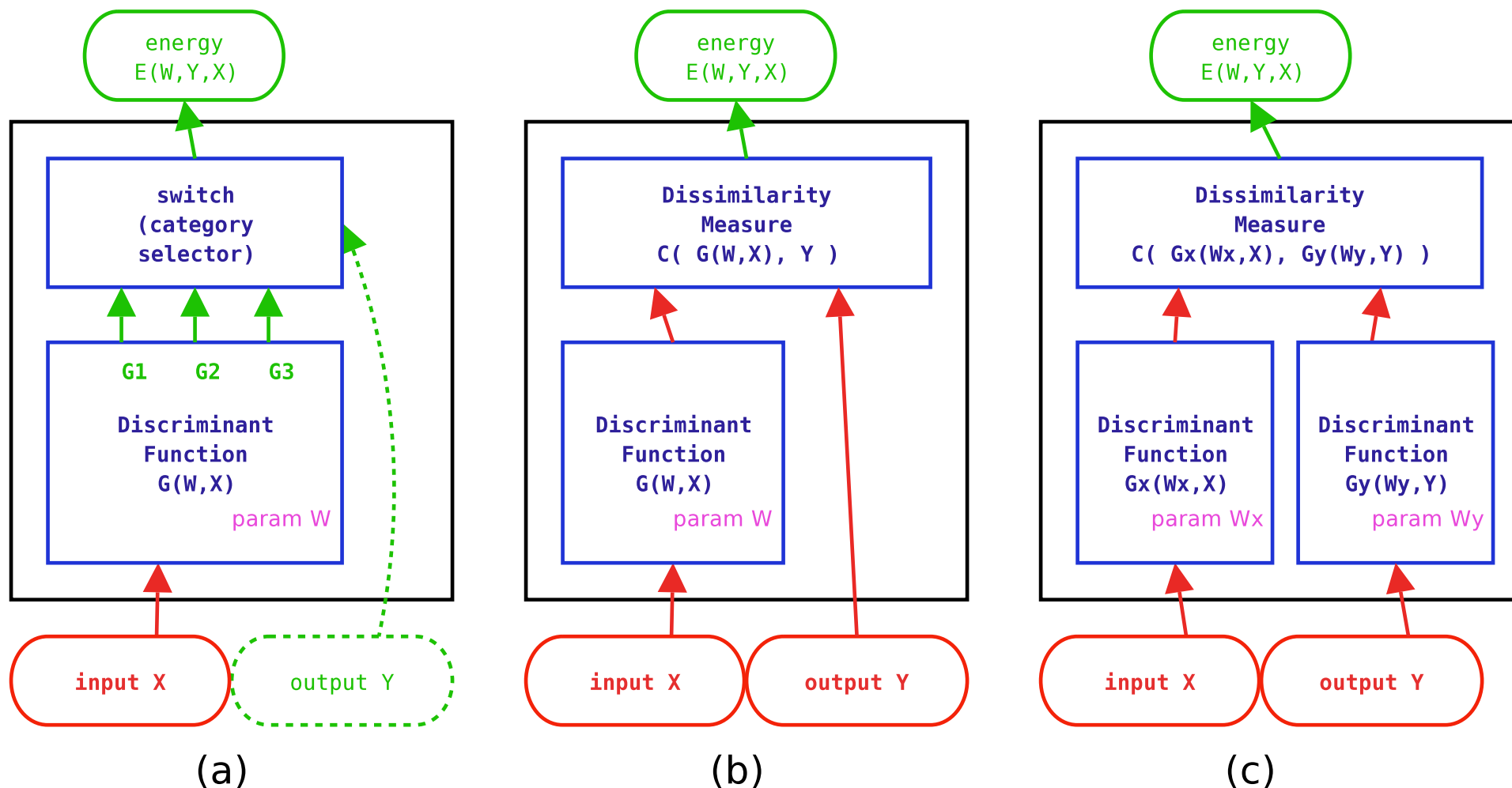
- We have complete freedom for the form and parameterization of the energy function (including things that can't be normalized).
- because we don't need to normalize, we can use a much larger repertoire of model architectures.
- No need to find excuses for fudge factors: your energy function is your prerogative. The Probabilist Police can't tell you you are wrong because you are outside of their jurisdiction.
- No need for computing (intractable) partition functions
- No need to find excuses as to why your favorite approximation of the partition function is legitimate.

Pretty much every model we know is some form of EBM.

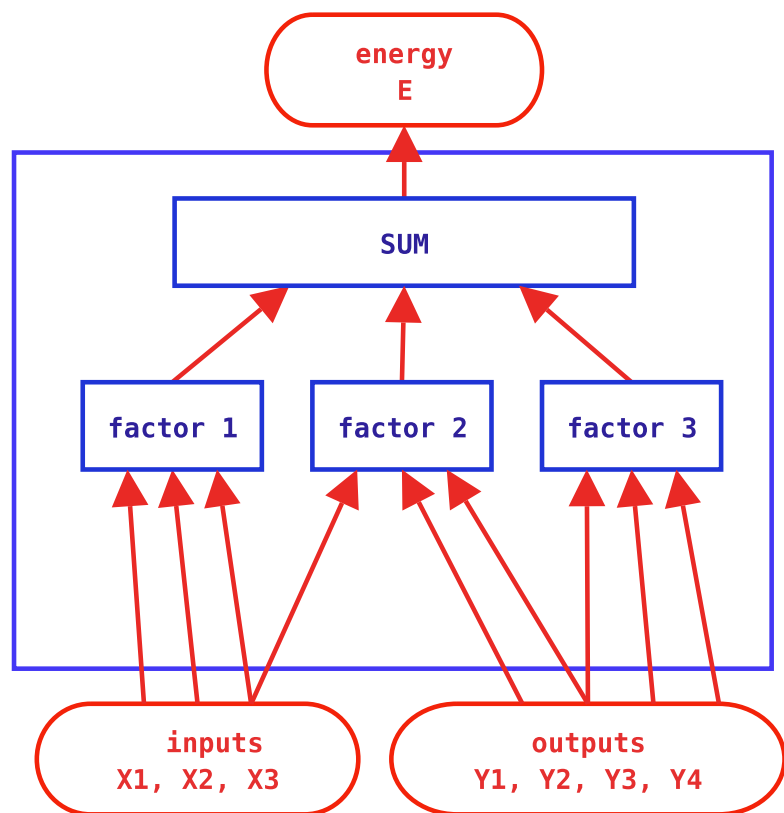
**QUESTION:** what loss functions can we use for training?

# Examples of EBM

Almost every type of model we know is some form of EBM. It all depends on how  $E(W, Y, X, Z)$  is parameterized.



# EBMs as Factor Graphs



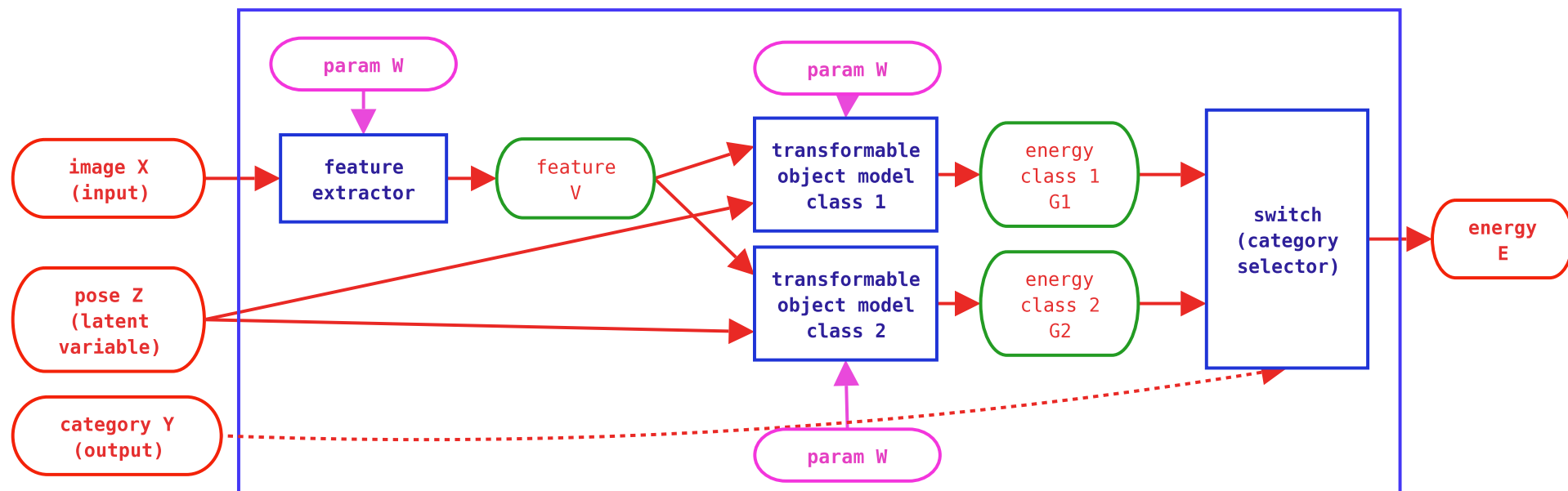
An EBM whose energy function can be “factorized” as a sum of individual functions (factors) is equivalent to a **graphical model** represented as a **factor graph**.

Any traditional graphical model can be formulated as a factor graph, but the converse is not true. Each factor is akin to  $-\log$  of the potential functions of a clique of variable nodes.

Efficient inference algorithms such as **(loopy) belief propagation** can be used to compute the marginals of  $Y$ , or the lowest energy configuration [Kschischang, Frey, Loeliger, 2001].

# EBM for Invariant Recognition

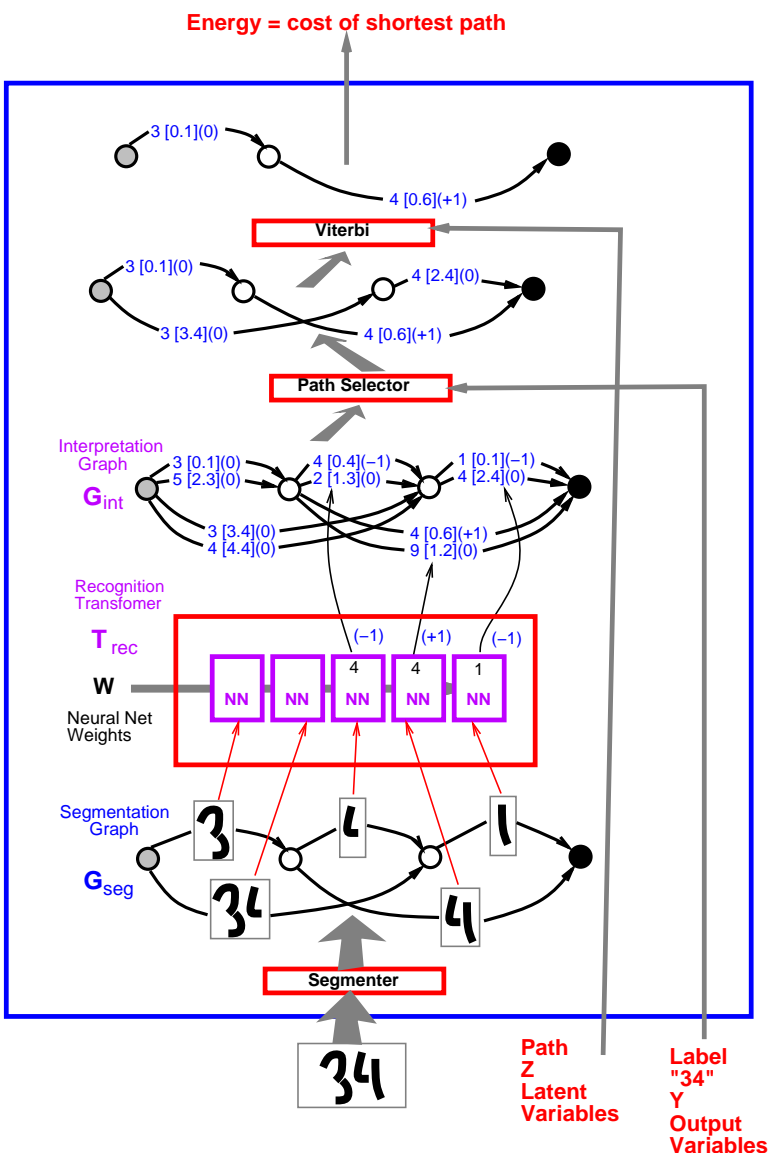
EBM Architecture for invariant object recognition



Each object model matches the output of the feature extractor to a reference representation that is transformed by the pose parameters.

**Inference** finds the category and the pose that minimize the energy.

# EBM for Sequence Labeling



- Word recognition, Speech recognition, natural language processing.
- Looking for the shortest path in a trellis is like minimizing an energy where the latent variable  $Z$  is the path, and the output  $Y$  is the labeling along the path.

# Training EBMs

---

- Training will consist in finding a  $W$  that minimizes a **loss function**  $\mathcal{L}(W, \mathcal{S})$ , over the training set  $\mathcal{S}$ .
- We must devise loss functions that “**carve**” the energy landscape so that the energy is **small around training samples** and **high everywhere else**..
- We seek loss functions that **do not require** evaluating intractable integrals, but which, nevertheless, drive the machine to approach the desired behavior.
- Basic idea: “**dig holes**” at  $(X, Y)$  locations near training samples, while “**building hills**” at un-desired locations, particularly the ones that are erroneously picked by the inference algorithm.
- Whereas probabilistic models trained with max likelihood shape the entire energy surface, our EBM loss function will merely dig holes at the right places and build hills only where needed to avoid erroneous inferences.

# Loss Functions for EBMs

---

■ training set  $\mathcal{S} = \{(X^i, Y^i), i = 1..P\}$

■ Loss:

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

■  $L(W, Y^i, X^i)$  is the per-sample loss function for sample  $(X^i, Y^i)$ .  $L$  is assumed to have a lower bound.

■  $R$  is a monotonically increasing function. In the following we assume  $R$ =identity

■ the loss is invariant under permutations of the samples, and under multiple repetitions of the same training set.

■ What form can  $L(W, Y, X)$  take?



# Condition on the Energy

---

- Condition for correct output on sample  $(X^i, Y^i)$ : there is a margin  $m > 0$ , such that:

$$E(W, Y^i, X^i) < E(W, Y, X^i) - m, \quad \forall Y \in \{Y\}, Y \neq Y^i$$

- **Assumption:**  $L$  depends on  $X^i$  only through the set of energies  $\{E(W, Y, X^i), Y \in \{Y\}\}$ .

- For example, if  $\{Y\} = \{0, 1, \dots, k - 1\}$

$$L(W, Y^i, X^i) = L(Y^i, E(W, 0, X^i), \dots, E(W, k - 1, X^i))$$

- We want to design  $L$  so that making an update of  $W$  to decrease  $L(W, Y^i, X^i)$  will automatically decrease the difference  $E(W, Y^i, X^i) - E(W, Y, X^i)$  for values of  $Y$  such that  $E(W, Y^i, X^i) < E(W, Y, X^i) - m$ .

# Examples of Loss Functions

---

- **Energy Loss:**  $L_{\text{energy}}(W, Y^i, X^i) = E(W, Y^i, X^i)$ .

Only works if the architecture is such that decreasing  $E(W, Y^i, X^i)$  will automatically increase  $E(W, Y, X^i)$  for  $y \neq Y^i$ .

- **Generalized Perceptron Loss:**

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

Does not work because the margin is zero. This reduces to the traditional linear perceptron loss when  $E(W, Y, X) = -YW.X$ .

- **Generalized Margin Loss:**

$$L_{\text{gmargin}}(W, Y^i, X^i) = Q[E(W, Y^i, X^i), E(W, \bar{Y}, X^i)]$$

Where  $Q$  is an increasing function of  $E(W, Y^i, X^i)$  and a decreasing function of  $E(W, \bar{Y}, X^i)$ .

- **Negative Log Likelihood Loss:**  $L_{\text{nll}}(W, Y^i, X^i) = E(W, Y^i, X^i) - F_{\beta}(W, X^i)$

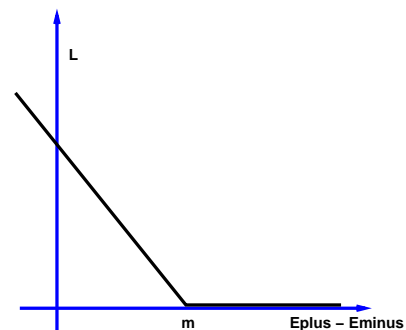
$$\text{with: } F_{\beta}(W, X^i) = -\frac{1}{\beta} \log \left( \int_{Y \in \{Y\}} \exp[-\beta E(W, Y, X^i)] \right)$$

# Special Cases of the Generalized Margin Loss

$$L_{\text{gmargin}}(W, Y^i, X^i) = Q[E(W, Y^i, X^i), E(W, \bar{Y}, X^i)]$$

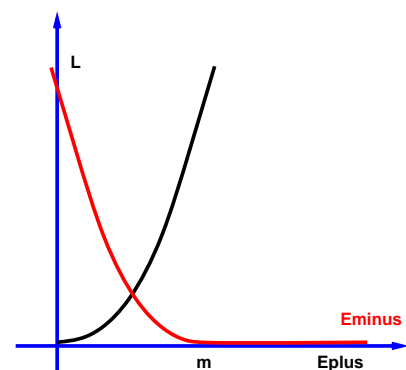
**Hinge Loss:**

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



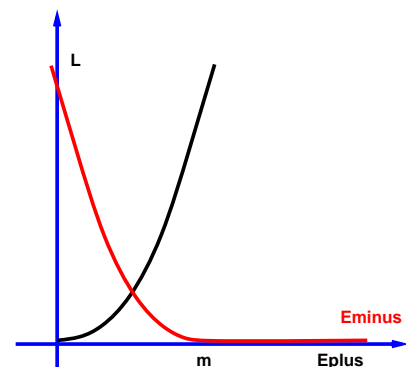
**Square-Square Loss:**

$$L_{\text{sqsq}}(W, Y^i, X^i) = E(W, Y^i, X^i)^2 + (m - E(W, \bar{Y}, X^i))^2$$



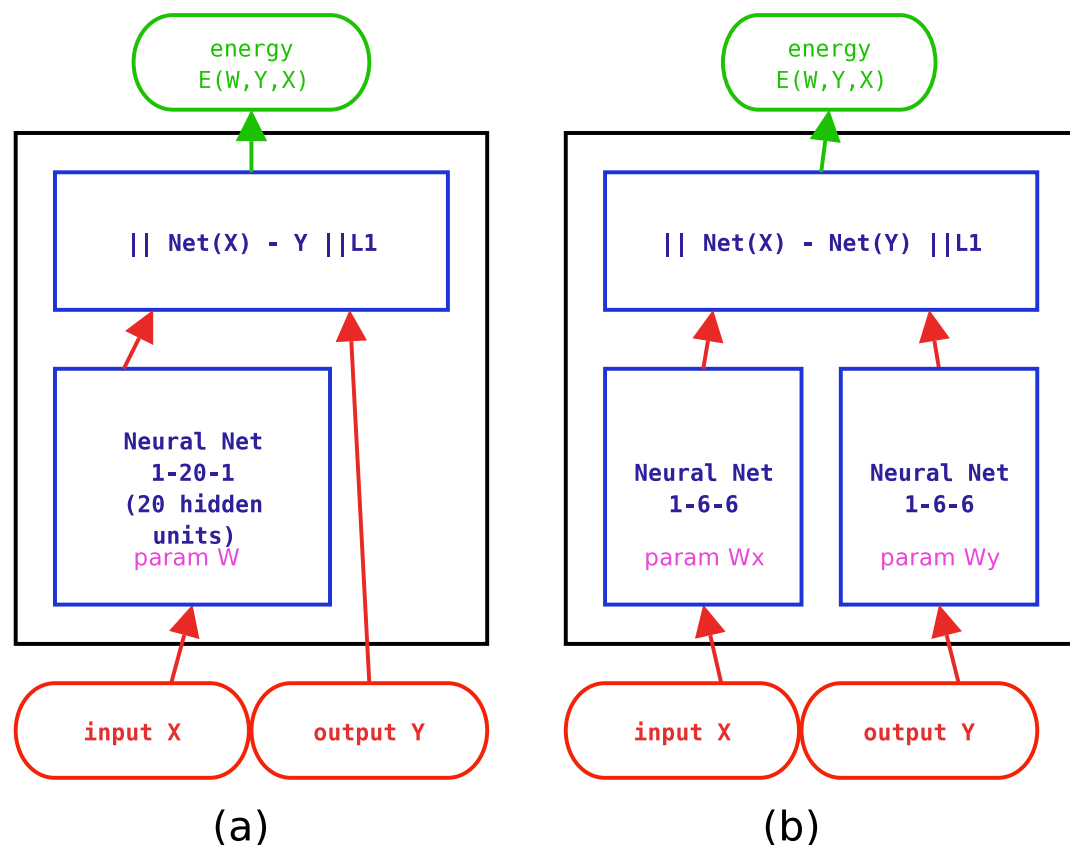
**Square-Exp Loss:**

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



# EBM Demos

Initially, the forbidden sphere around  $Y^i$  is 0.2, then 0.1.



- Demo 1:  $Y = X^2$ , Architecture A, Square Energy Loss. It works because  $E(Y, X)$  is a fixed quadratic function of  $Y$ .
- Demo 2:  $Y = X^2$ , Architecture B, Square Energy. It collapses.
- Demo 3:  $Y = X^2$ , Architecture B, Square-Square Margin Loss
- Demo 4:  $Y = X^2$ , Architecture B, Negative Log Likelihood Loss. Few iterations, but each iteration is expensive
- Demo 5: eye pattern, Architecture B, Negative Log Likelihood Loss

# EBM

---

- The normalization of probabilistic models is an unnecessary aggravation
- Energy-based models with appropriate loss functions avoid the estimation of intractable partition functions and their derivative.
- EBMs give us complete freedom in the choice of architecture that model the joint “compatibility” (energy) between variables.
- We can use building blocks that are not normally allowed in probabilistic models (like neural nets).