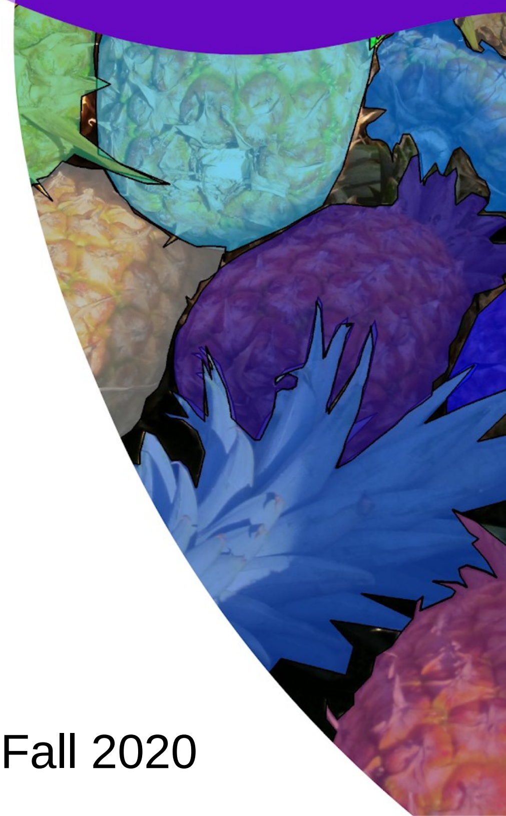
Energy-Based Models (part 1)



Yann LeCun
NYU - Courant Institute & Center for Data Science
Facebook AI Research
http://yann.lecun.com
TAs: Alfredo Canziani, Mark Goldstein

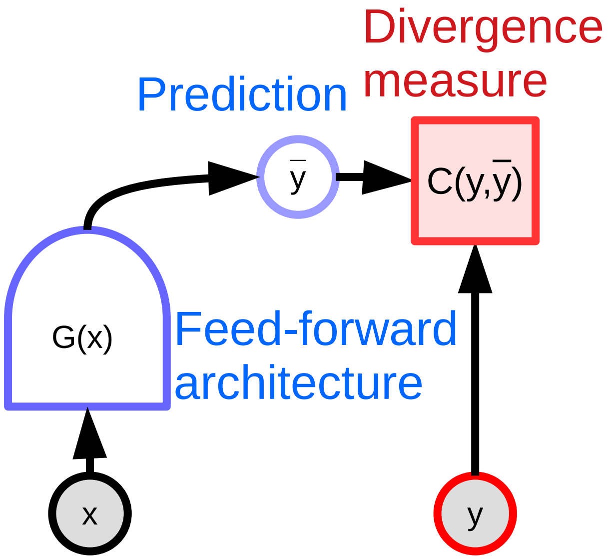
Deep Learning, NYU, Fall 2020

Plan

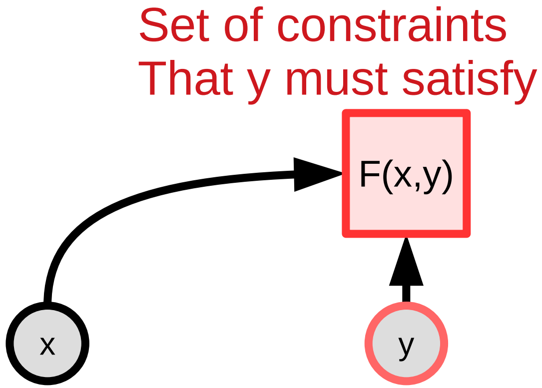
- 1. managing uncertainty / multimodality
- 2. Implicit function through energy
- 3. EBM and conditional EBM
- 4.

Energy-Based Models

- **Feed-forward nets use a finite number of steps to produce a single output.**
- **What if…**
  - The problem requires a complex computation to produce its output? (complex inference)
  - There are multiple possible outputs for a single input? (e.g. predicting future video frames)

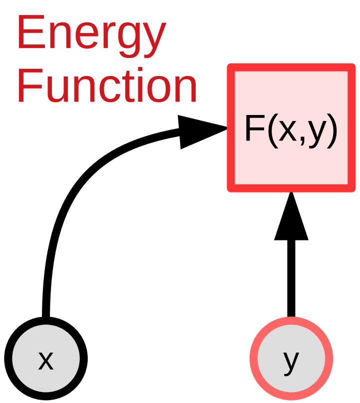


- **Inference through constraint satisfaction**
  - Finding an output that satisfies constraints: e.g a linguistically correct translation or speech transcription.
  - Maximum likelihood inference in graphical models

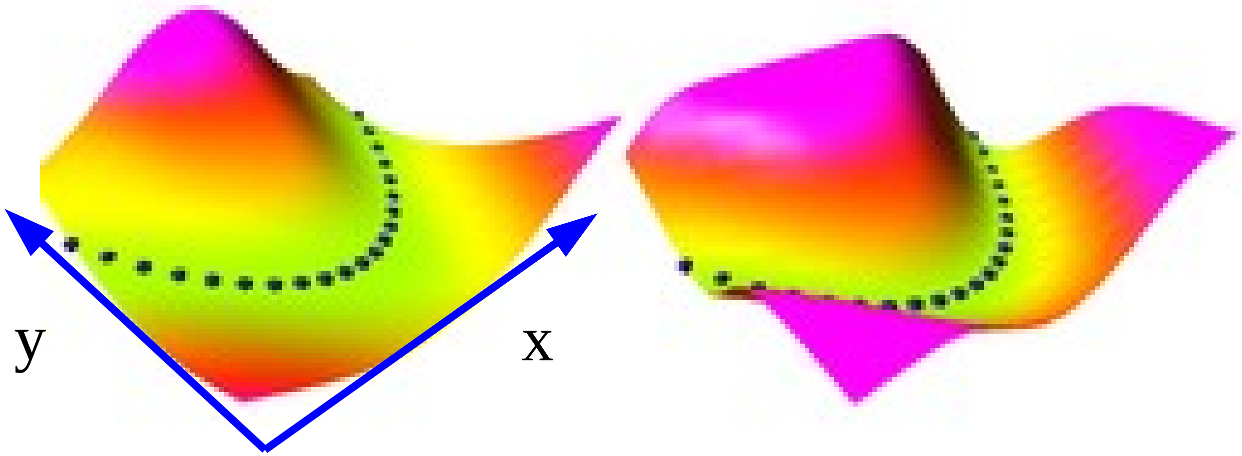


Energy-Based Models (EBM)

- **Energy function F(x,y) scalar-valued.**
  - Takes low values when y is compatible with x and higher values when y is less compatible with x
- **Inference: find values of y that make F(x,y) small.**
  - There may be multiple solutions $\check{y} = \operatorname{argmin}_y F(x, y)$

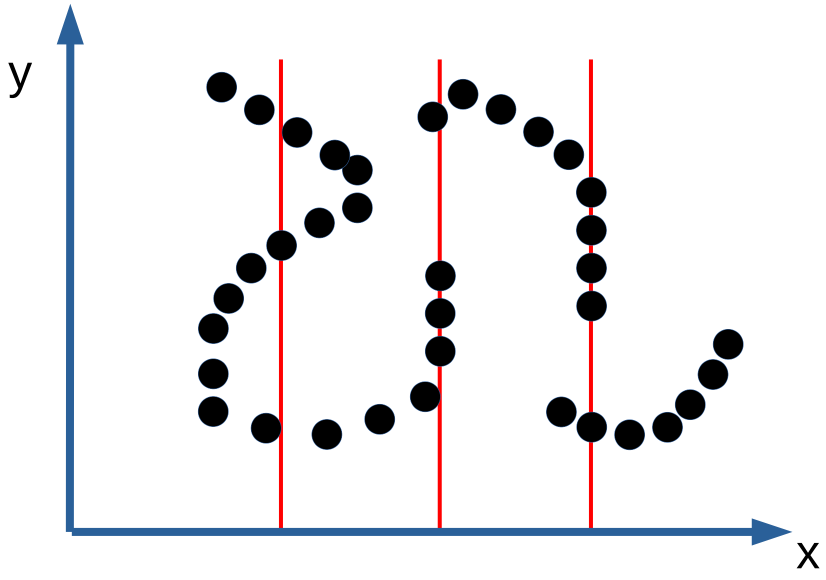


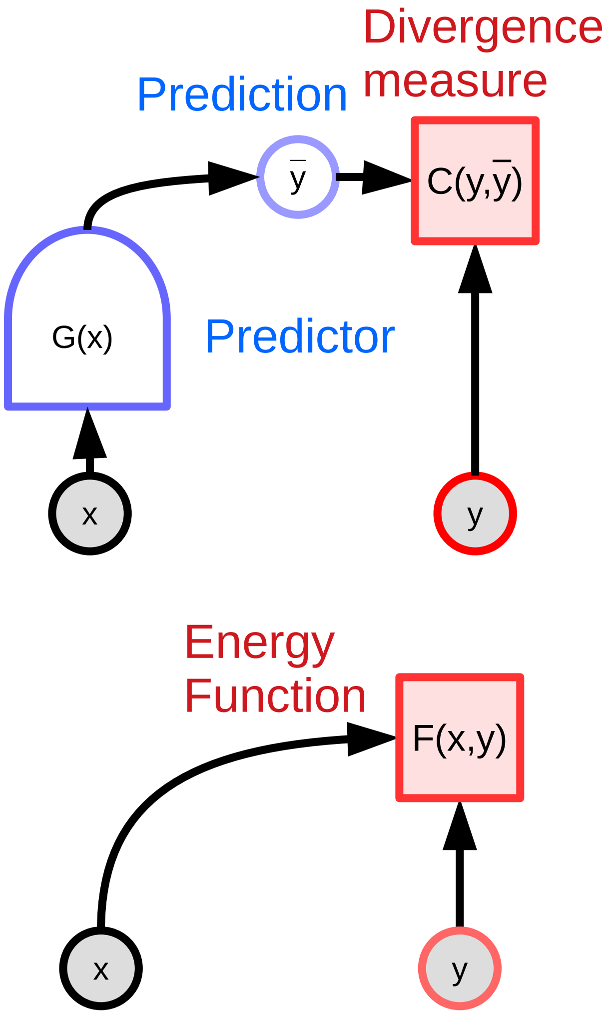
- **Note: the energy is used for inference, not for learning**
- **Example**
  - Blue dots are data points



Energy-Based Model: implicit function

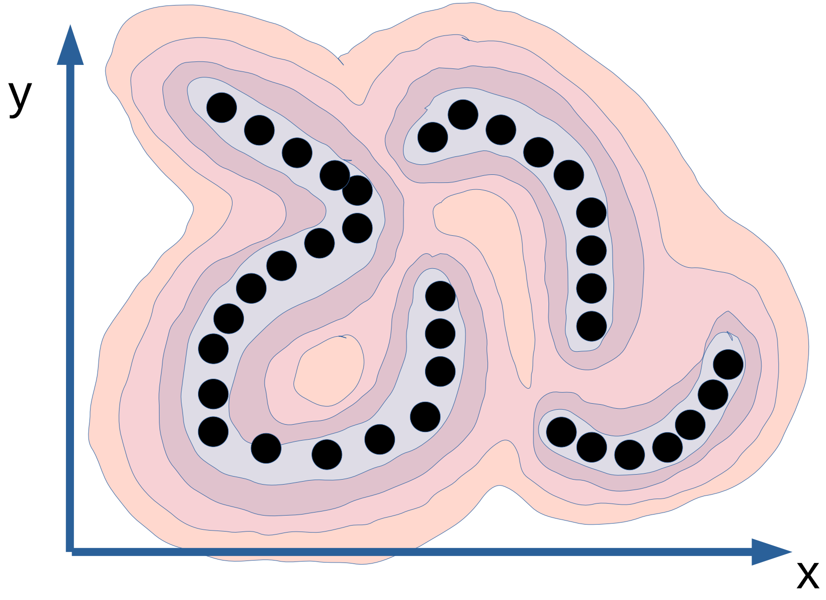
- **A feed-forward model is an explicit function that computes y from x.**
- **An EBM is an implicit function that captures the dependency between x and y**
- **Multiple y can be compatible with a single x**



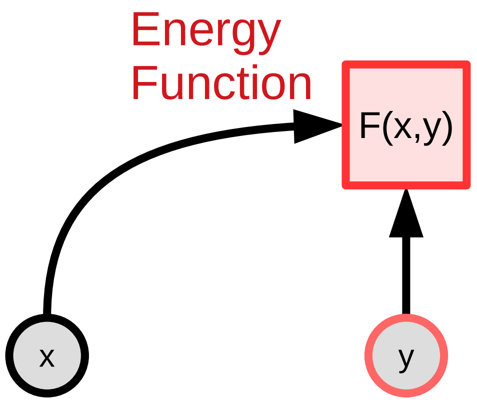


Energy-Based Model: implicit function

- **Energy function that captures the x,y dependencies:**
  - Low energy near the data points. Higher energy everywhere else.
  - If y is continuous, F should be smooth and differentiable, so we can use gradient-based inference algorithms.



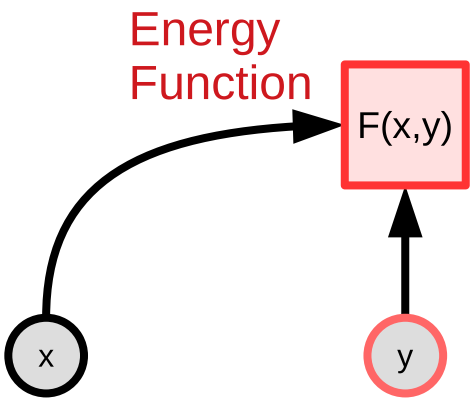
$$\check{y} = \mathrm{argmin}_y F(x, y)$$

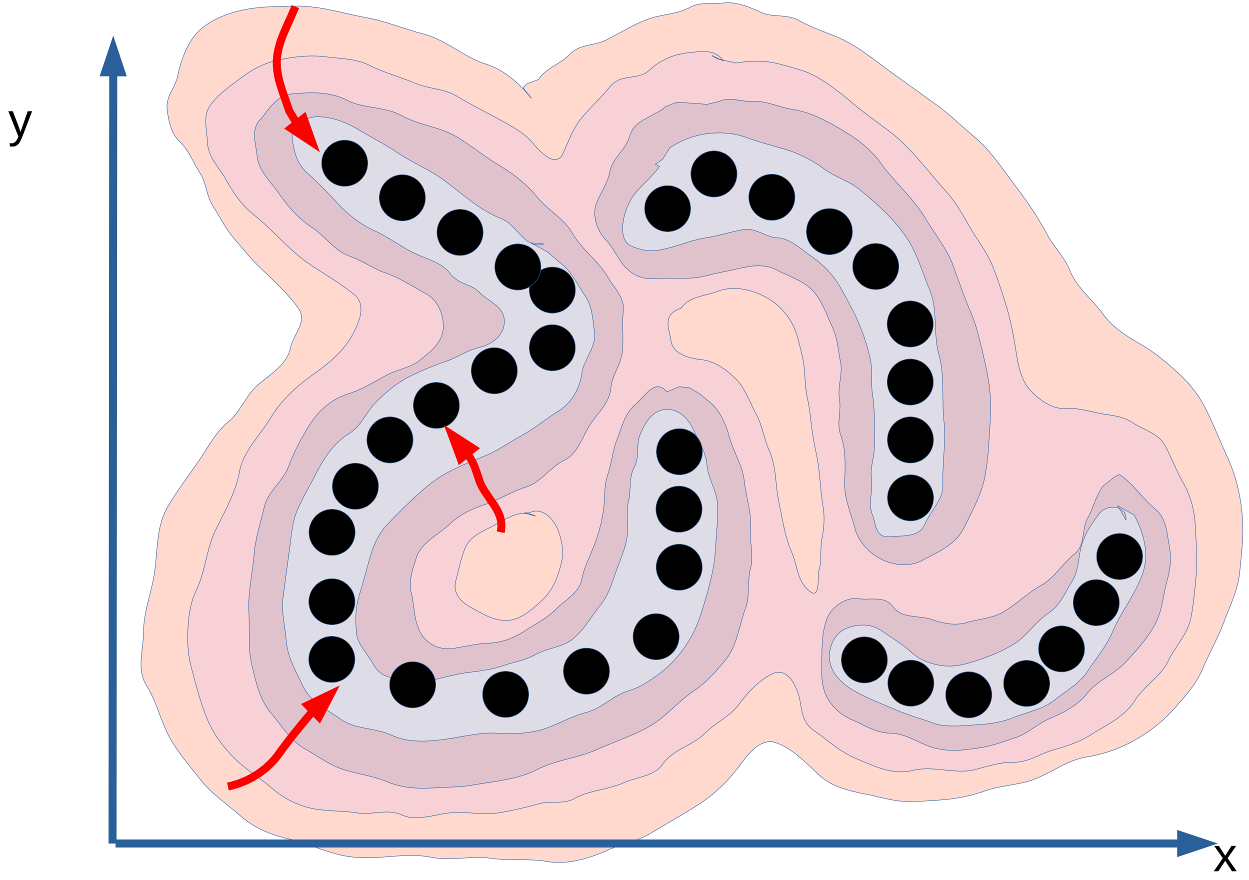


Energy-Based Model: gradient-based inference

- **If y is continuous**
  - We can use a gradient-based method for inference.

$$\check{y} = \text{argmin}_y F(x, y)$$

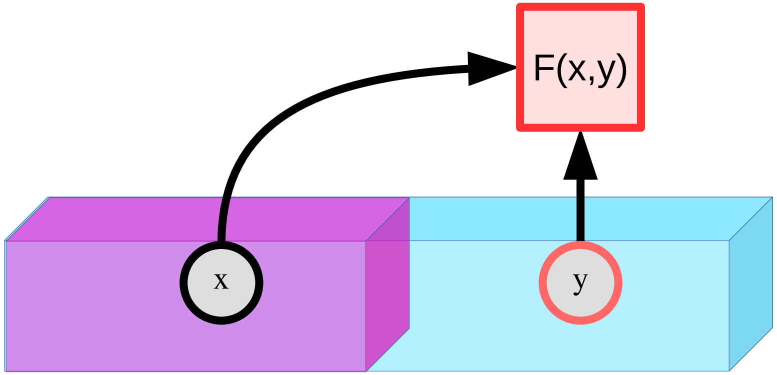




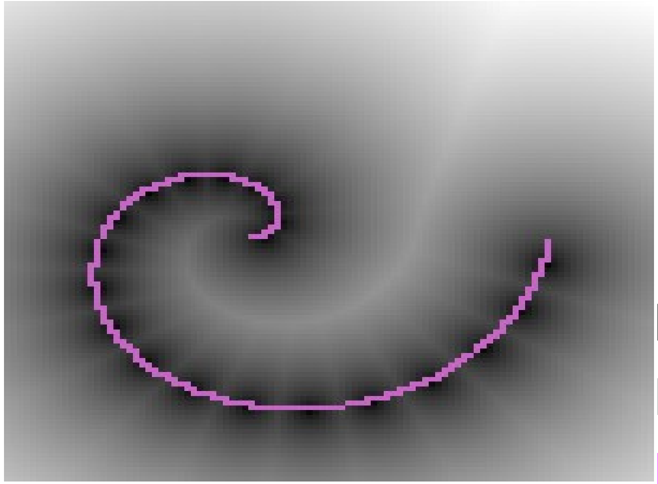
Energy-Based Model: unconditional version

- **Conditional EBM: F(x,y)**
- **Unconditional EBM: F(y)**
  - measures the compatibility between the components of y
  - If we don't know in advance which part of y is known and which part is unknown

Energy Function

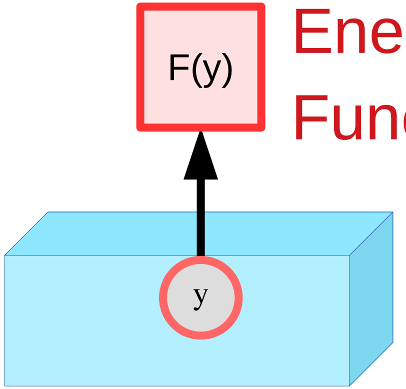


y2



y1

Dark = low energy (good)
Bright = high energy (bad)
Purple = data manifold

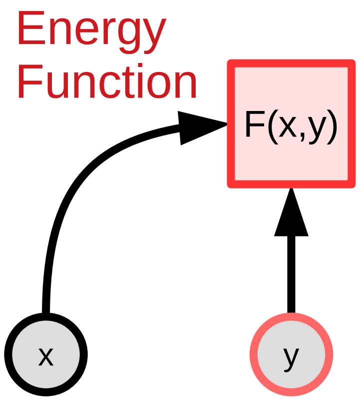


Energy Function

Energy-Based Models vs Probabilistic Models

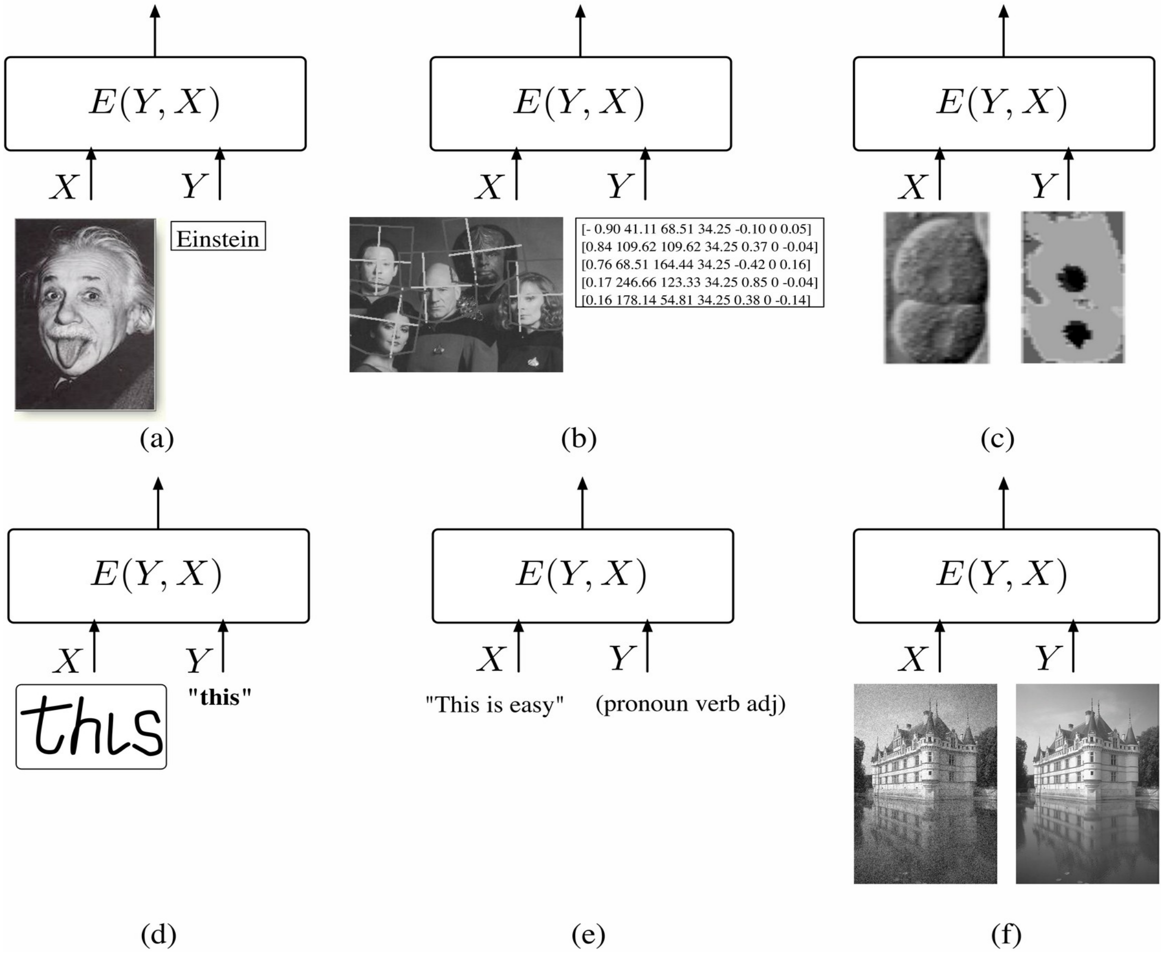
- **Probabilistic models are a special case of EBM**
  - Energies are like un-normalized negative log proabilities
- **Why use EBM instead of probabilistic models?**
  - EBM gives more flexibility in the choice of the scoring function.
  - More flexibility in the choice of objective function for learning
- **From energy to probability: Gibbs-Boltzmann distribution**
  - Beta is a positive constant

$$P(y|x) = \frac{e^{-\beta F(x,y)}}{\int_{y'} e^{-\beta F(x,y')}}$$



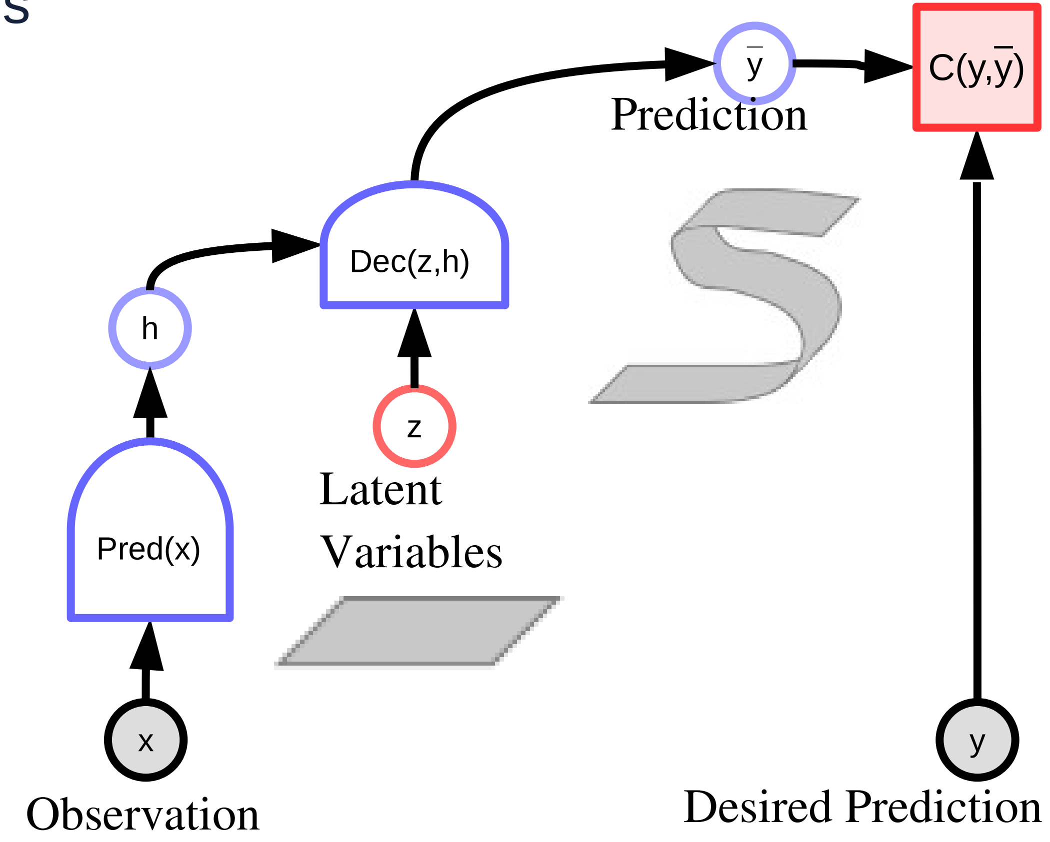
When inference is hard

- **Cases where inference is hard:**
  - Output is a high-dimensional object with structure:
    - Sequence, image, video,…
  - Output has compositional structure:
    - Text, action sequence,…
  - Output results from a long chain of reasoning
    - That can be reduced to an optimization problem



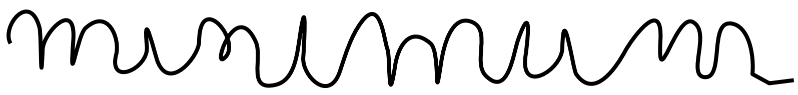
Architecture for Multimodal Output: latent variable EBM

- **Latent variables:**
  - parameterize the set of predictions
- **Ideally, the latent variable represents independent explanatory factors of variation of the prediction.**
- **The information capacity of the latent variable must be minimized.**
  - Otherwise all the information for the prediction will go into it.

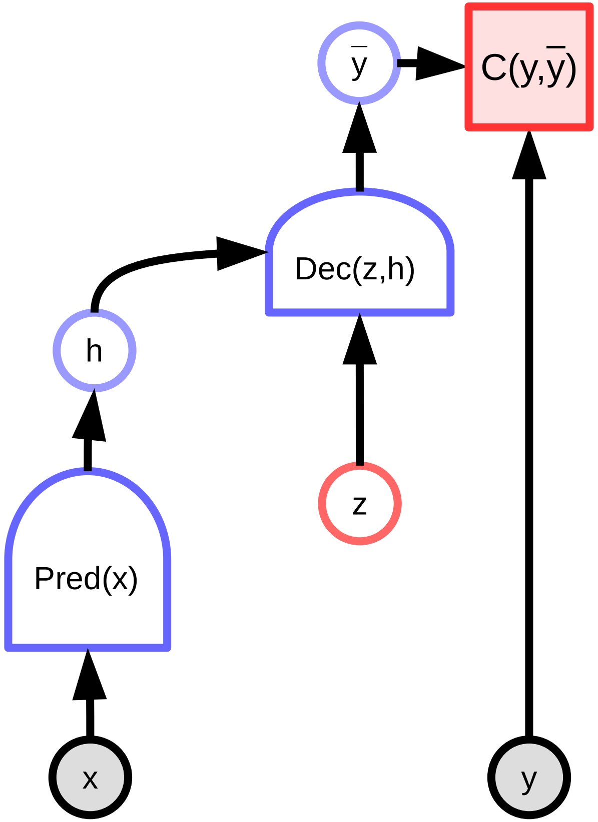


When inference involves latent variables

- **Latent variables are variables whose value is never given to us.**
  - Examples: to read a handwritten word, it helps to know where the characters are

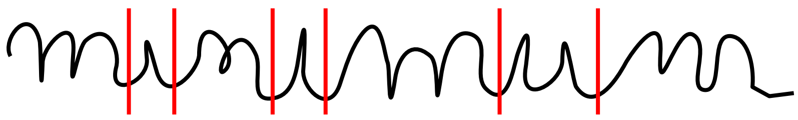


  - To recognize speech, it helps to know where the words and phonemes are
    - Youcantreadthisifyoudontunderstandenglish
    - Vousnepouvezpaslirececisivousneparlezpasfrançais

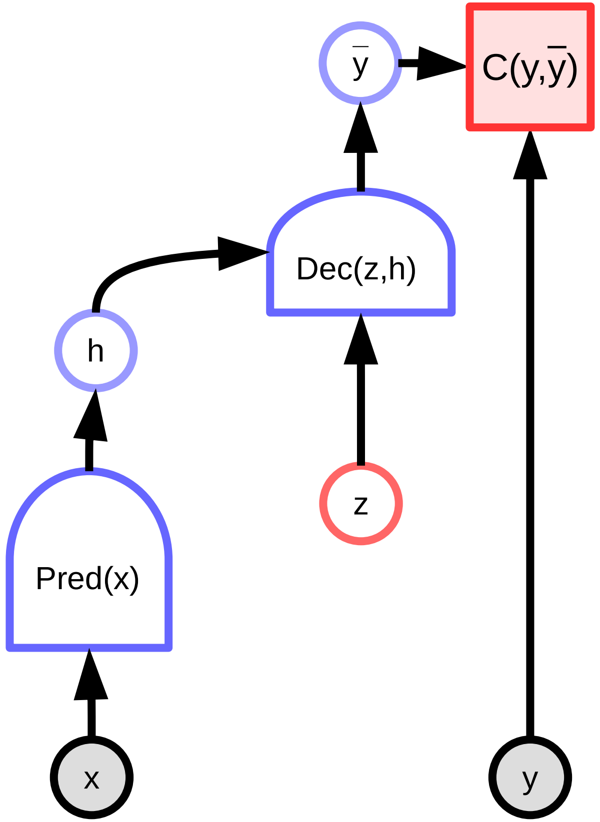


When inference involves latent variables

- **Latent variables are variables whose value is never given to us.**
  - Examples: to read a handwritten word, it helps to know where the characters are



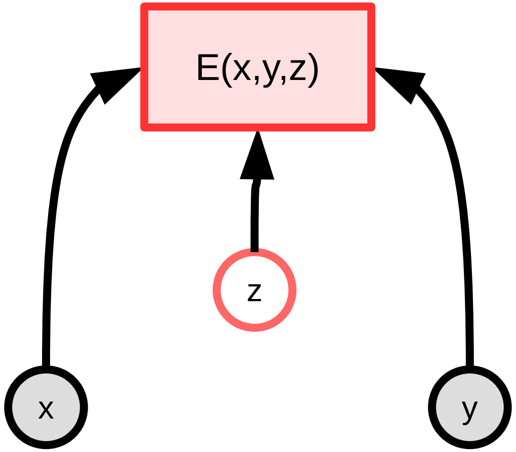
  - To recognize speech, it helps to know where the words and phonemes are
    - You can't read this if you don't understand english
    - Vous ne pouvez pas lire ceci si vous ne parlez pas français



Latent-Variable EBM: inference

- **Simultaneous minimization with respect to y and z**

$$\check{y}, \check{z} = \text{argmin}_{y,z} E(x, y, z)$$

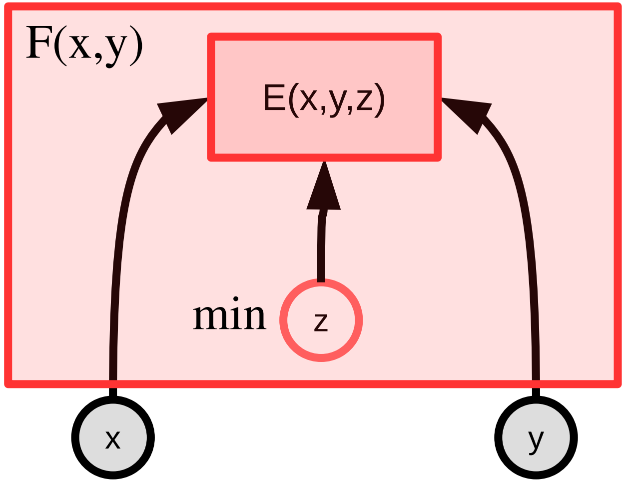


- **Redefinition of F(x,y)**

$$F_\infty(x, y) = \min_z E(x, y, z)$$

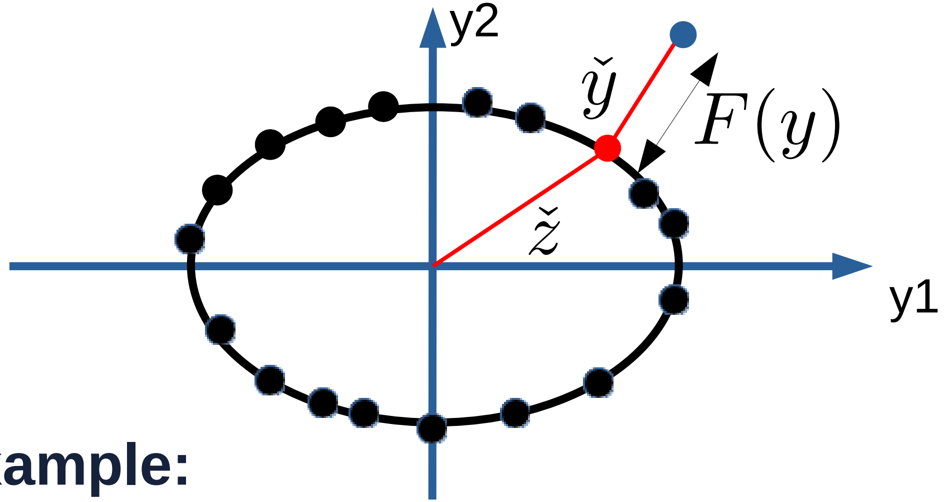
$$F_\beta(x, y) = -\frac{1}{\beta} \log \int_z e^{-\beta E(x,y,z)}$$

$$\check{y} = \text{argmin}_y F(x, y)$$



Inference with Latent Variable EBMs

- **The latent variable parameterizes the data manifold(s).**
- **The energy computes a distance to the learned manifold(s).**
- **The gradient of the energy points to the closest point on the data manifold(s).**



- **Example:**
  - learned manifold = ellipse
  - Latent variable = angle
  - Energy = distance of data point to ellipse

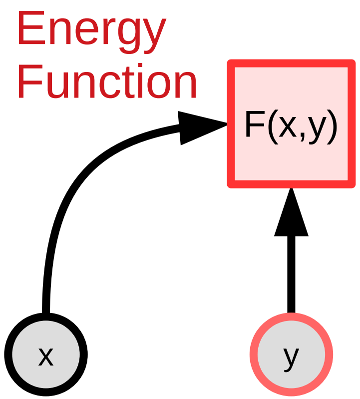
- **Model:**

$$E(y,z) = (y_1 - r_1 \sin(z))^2 + (y_2 - r_2 cos(z))^2$$

$$F(y) = \min_z (y_1 - r_1 \sin(z))^2 + (y_2 - r_2 cos(z))^2$$

Turning Energies to Probabilities

- **From energy to probability: Gibbs-Boltzmann distribution**
  - Beta is a positive constant
- **This is not always possible, not desirable.**



$$P(y|x) = \frac{e^{-\beta F(x,y)}}{\int_{y'} e^{-\beta F(x,y')}}$$

Marginalizing over a latent variable

$$P(y,z|x) = \frac{e^{-\beta E(x,y,z)}}{\int_y \int_z e^{-\beta E(x,y,z)}} \qquad P(y|x) = \int_z P(y,z|x)$$

$$P(y|x) = \frac{\int_z e^{-\beta E(x,y,z)}}{\int_y \int_z e^{-\beta E(x,y,z)}} = \frac{e^{-\beta\left[-\frac{1}{\beta}\log\int_z e^{-\beta E(x,y,z)}\right]}}{\int_y e^{-\beta\left[-\frac{1}{\beta}\log\int_z e^{-\beta E(x,y,z)}\right]}} = \frac{e^{-\beta F_\beta(x,y)}}{\int_y e^{\beta F_\beta(x,y)}}$$

- **Free energy F(x,y)** $$F_\beta(x,y) = -\frac{1}{\beta}\log\int_z e^{-\beta E(x,y,z)}$$

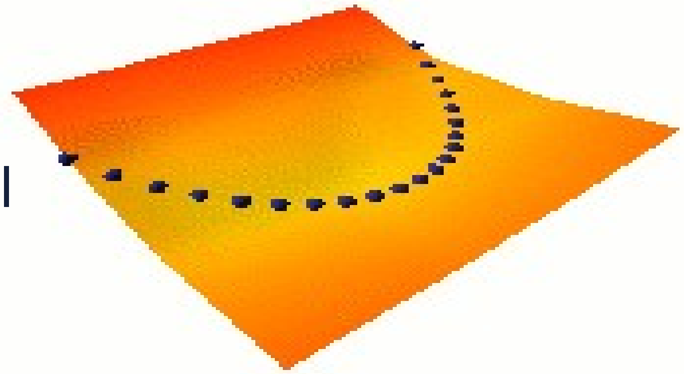
FACEBOOK AI

Training EBMs

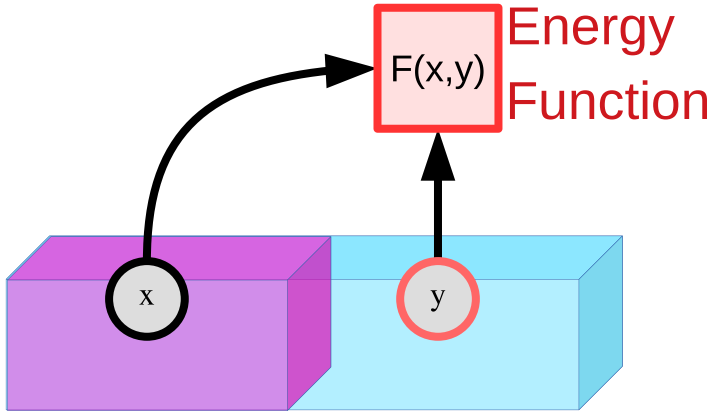
Push down on the energy of data points
Make sure the energy is higher elsewhere

Training an Energy-Based Model

- **Parameterize F(x,y)**
- **Training samples: x[i], y[i]**
- **Shape F(x,y) so that:**
  - F(x[i], y[i]) is strictly smaller than F(x[i], y) for all y different from y[i]
  - Keep F smooth
    - Max-likelihood probabilistic methods break that!



- **Two classes of learning methods:**
  - 1. **Contrastive methods**: push down on F(x[i], y[i]), push up on other points F(x[i], y')
  - 2. **Regularized/Architectural Methods**: build F(x,y) so that the volume of low energy regions is limited or minimized through regularization



Contrastive Methods vs Regularized/Architectural Methods

- **Contrastive: [they all are different ways to pick which points to push up]**
  - C1: push down of the energy of data points, push up everywhere else: Max likelihood (needs tractable partition function or variational approximation)
  - C2: push down of the energy of data points, push up on chosen locations: max likelihood with MC/MMC/HMC, Contrastive divergence, Metric learning/Siamese nets, Ratio Matching, Noise Contrastive Estimation, Min Probability Flow, adversarial generator/GANs
  - C3: train a function that maps points off the data manifold to points on the data manifold: denoising auto-encoder, masked auto-encoder (e.g. BERT)
- **Regularized/Architectural: [Different ways to limit the information capacity of the latent representation]**
  - A1: build the machine so that the volume of low energy space is bounded: PCA, K-means, Gaussian Mixture Model, Square ICA, normalizing flows…
  - A2: use a regularization term that measures the volume of space that has low energy: Sparse coding, sparse auto-encoder, LISTA, Variational Auto-Encoders, discretization/VQ/VQVAE.
  - A3: F(x,y) = C(y, G(x,y)), make G(x,y) as "constant" as possible with respect to y: Contracting auto-encoder, saturating auto-encoder
  - A4: minimize the gradient and maximize the curvature around data points: score matching

Contrastive Methods: Max likelihood / Probabilistic Methods

- **Push down on data points, push up of other points**
  - well chosen contrastive points
- **Max likelihood / probabilistic models**

$$P_w(y|x) = \frac{e^{-\beta F_w(x,y)}}{\int_{y'} e^{-\beta F_w(x,y')}}$$

  - Loss: $$\mathcal{L}(x,y,w) = F_w(x,y) + \frac{1}{\beta}\log\int_{y'} e^{-\beta F_w(x,y')}$$
  - Gradient: $$\frac{\partial\mathcal{L}(x,y,w)}{\partial w} = \frac{\partial F_w(x,y)}{\partial w} - \int_{y'} P_w(y'|x)\frac{\partial F_w(x,y')}{\partial w}$$
  - MC/MCMC/HMC/CD: $\hat{y}$ sampled from $P_w(y|x)$

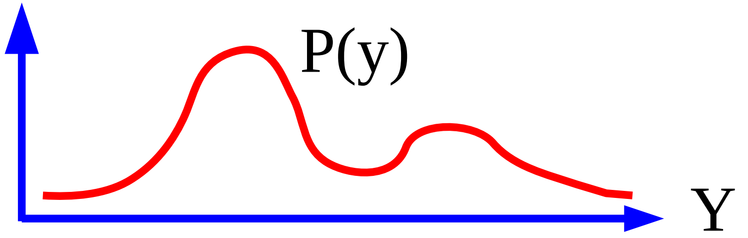
$$\frac{\partial\mathcal{L}(x,y,w)}{\partial w} = \frac{\partial F_w(x,y)}{\partial w} - \frac{\partial F_w(x,\hat{y})}{\partial w}$$

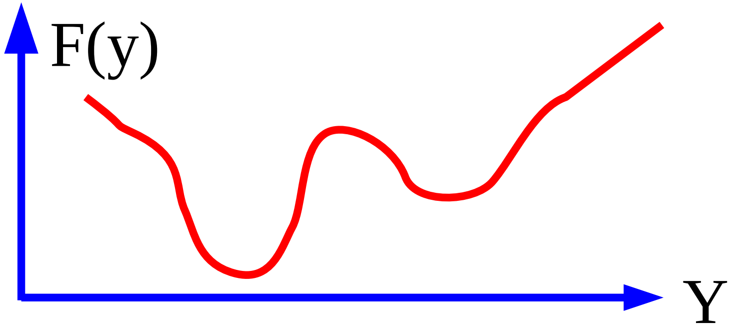
Familiar Example: Maximum Likelihood Learning

- **The energy can be interpreted as an unnormalized negative log density**
- **Gibbs distribution: Probability proportional to exp(-energy)**
  - Beta parameter is akin to an inverse temperature
- **Don't compute probabilities unless you absolutely have to**
  - Because the denominator is often intractable

$$P(y) = \frac{\exp[-\beta F(y)]}{\int_{y'} \exp[-\beta F(y')]}$$

$$P(y|x) = \frac{\exp[-\beta F(x,y)]}{\int_{y'} \exp[-\beta F(x,y')]}$$





push down of the energy of data points, push up everywhere else

- Max likelihood (requires a tractable partition function)

Maximizing P(Y|W) on training samples

make this big

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

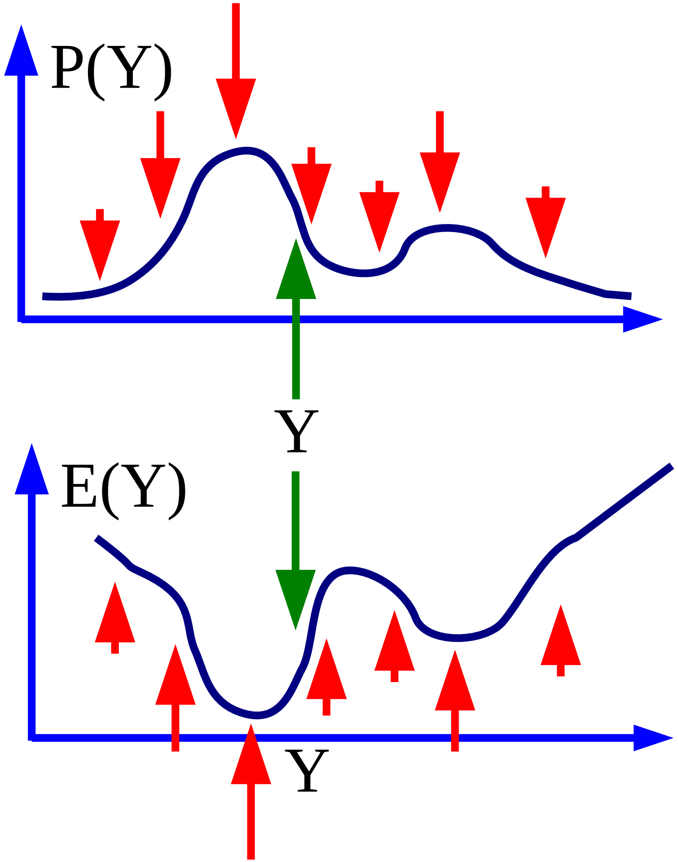
make this small

Minimizing -log P(Y,W) on training samples

$$L(Y,W) = E(Y,W) + \frac{1}{\beta} \log \int_y e^{-\beta E(y,W)}$$

make this small

make this big



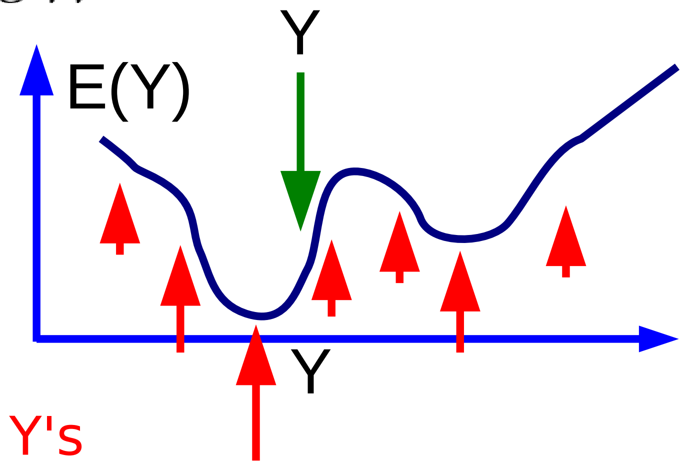
push down of the energy of data points, push up everywhere else

Gradient of the negative log-likelihood loss for one sample Y:

$$\frac{\partial L(Y,W)}{\partial W} = \frac{\partial E(Y,W)}{\partial W} - \int_y P(y|W)\frac{\partial E(y,W)}{\partial W}$$

Gradient descent:

$$W \leftarrow W - \eta\frac{\partial L(Y,W)}{\partial W}$$



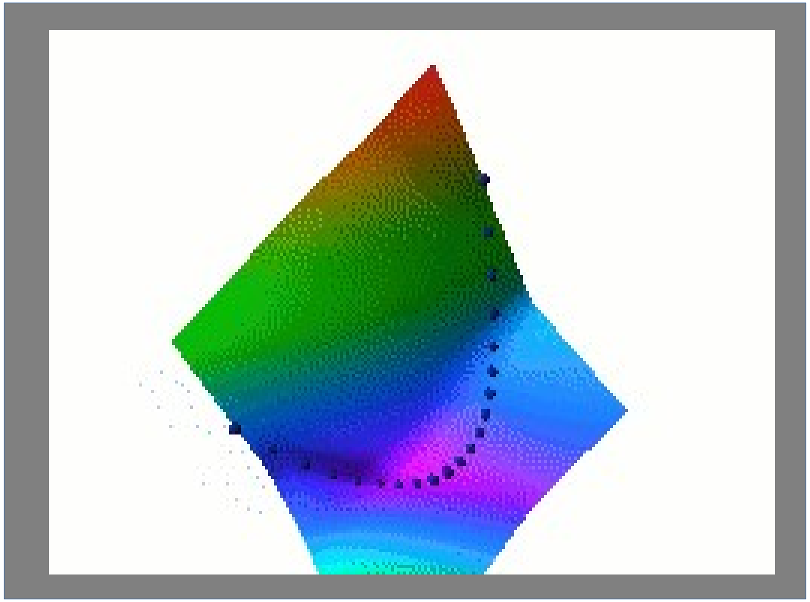
Pushes down on the energy of the samples

Pulls up on the energy of low-energy Y's

$$W \leftarrow W - \eta\frac{\partial E(Y,W)}{\partial W} + \eta\int_y P(y|W)\frac{\partial E(y,W)}{\partial W}$$

Problem with Max Likelihood / Probabilistic Methods

- **It wants to make the difference between the energy on the data manifold and the energy just outside of it infinitely large!**
- **It wants to make the data manifold an infinitely deep and infinitely narrow canyon.**
- **The loss must be regularized to keep the energy smooth**
  - e.g. à la Wassertstein GAN.
  - So that gradient-based inference works
  - Equivalent to a prior
  - But then, why use a probabilistic model?



Contrastive Methods: other losses

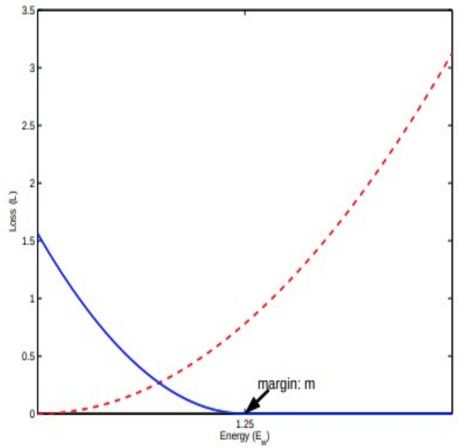
- **Push down on data points, push up of other points**
  - well chosen contrastive points
  - General margin loss: $\mathcal{L}(x, y, \hat{y}, w) = H(F_w(x, y), F_w(x, \hat{y}), m(y, \hat{y}))$
    - Where $H(F^+, F^-, m)$ is a strictly increasing function of F^+ and a strictly decreasing function of F^- , at least whenever $F^- - F^+ < m$.
  - Examples:
    - Simple [Bromley 1993]:

$$\mathcal{L}(x, y, \hat{y}, w) = [F_w(x, y)]^+ + [m(y, \hat{y}) - F_w(x, \hat{y})]^+$$

    - Hinge pair loss [Altun 2003], Ranking loss [Weston 2010]:

$$\mathcal{L}(x, y, \hat{y}, w) = [F_w(x, y) - F_w(x, \hat{y}) + m(y, \hat{y})]^+$$

    - Square-Square: [Chopra CVPR 2005] [Hadsell CVPR 2006]:



$$\mathcal{L}(x, y, \hat{y}, w) = \left([F_w(x, y)]^+\right)^2 + \left([m(y, \hat{y}) - F_w(x, \hat{y})]^+\right)^2$$

General margin loss

- **Considers all possible outputs**

$$\mathcal{L}(x, y, w) = \sum_{\check{y} \in \mathcal{Y}} H(F_w(x, y), F_w(x, \check{y}), m(y, \check{y}))$$

Contrastive Methods: group losses

- **Push down on a group of data points, push up on a group of contrastive points**
  - General group loss on p+ data points and p- contrastive points:

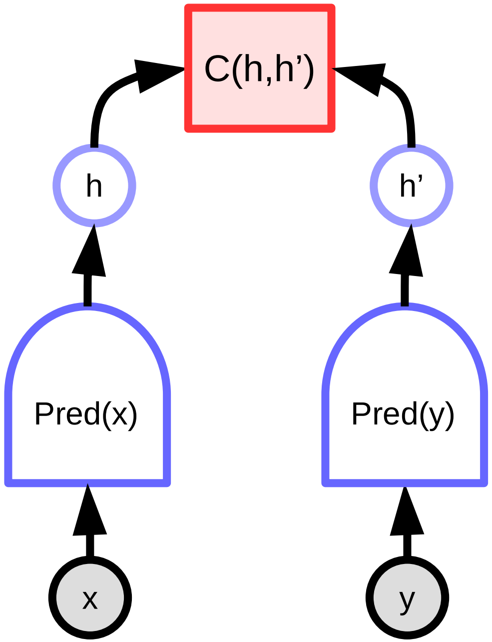
$$\mathcal{L}(x_1 \dots x_{p^+}, y_1 \dots y_{p^+}, \hat{y}_1 \dots \hat{y}_{p^-}, w) = H\left(E(x_1, y_1), \dots E(x_{p^+}, y_{p^+}), E(x_1, \hat{y_1}), \dots E(x_{p^+}, \hat{y_{p^+}}), M(Y_{1\dots p^+}, \hat{Y}_{1\dots p^-})\right)$$

    - Where H must be an increasing fn of the data energies and decreasing fn of the contrastive point energies within the margin.
    - M is a margin matrix for all pairs of y and ŷ in the group.
- Example: Neighborhood Component Analysis, Noise Contrastive Estimation (implicit infinite margin) [Goldberger 2005] [Gutmann 2010]...[Misra 2019] [Chen 2020]

$$\mathcal{L}(x, y, \hat{y_1}, \dots \hat{y}_{p^-}, w) = \frac{e^{-E_w(x,y)}}{e^{-E_w(x,y)} + \sum_{i=1}^{p^-} e^{-E_w(x,\hat{y}_i,w)}}$$

Contrastive Embedding

- Distance measured in feature space
- Multiple “predictions” through feature invariance
- Siamese nets, metric learning
  - [Bromley NIPS’93] [Chopra CVPR’05] [Hadsell CVPR’06]
- Advantage: no pixel-level reconstruction
- Difficulty: hard negative mining
- Successful examples for images:
  - DeepFace [Taigman et al. CVPR 2014]
  - PIRL [Misra et al. Arxiv:1912.01991]
  - MoCo [He et al. Arxiv:1911.05722]
  - SimCLR [Chen et al. Arxiv:2002.05709]
- Video / Audio
  - Temporal proximity [Taylor CVPR 2011]

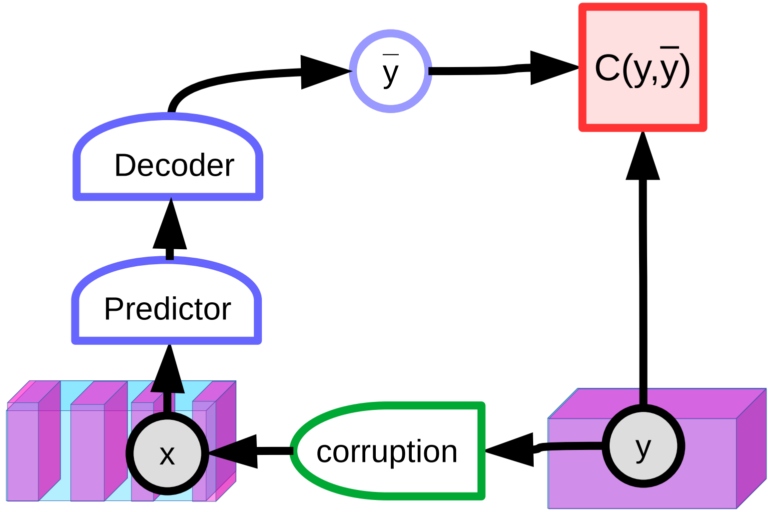


Positive pair:
Make F small

Negative pair:
Make F large

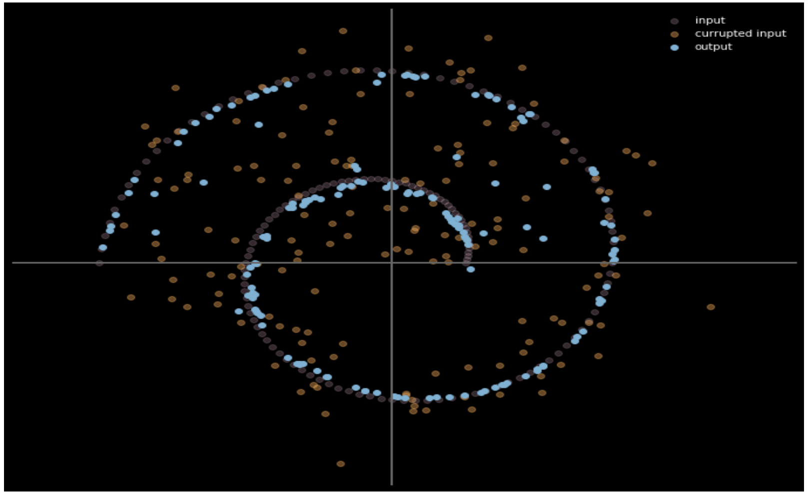
Contrastive Method: Denoising AE / Masked AE

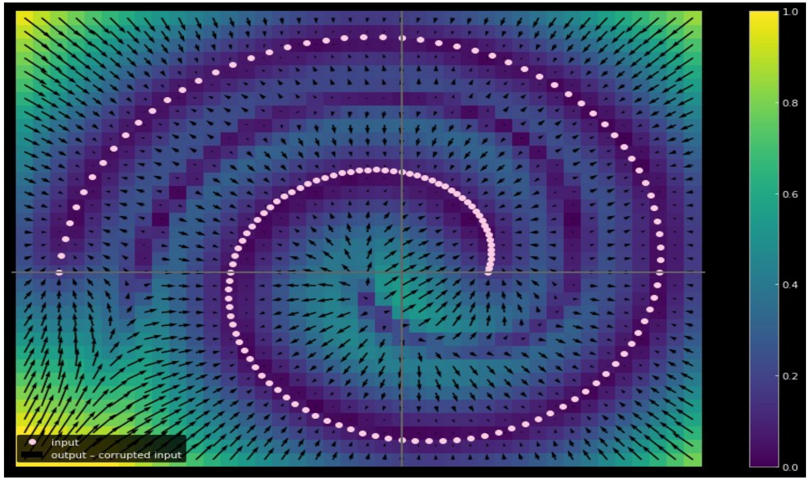
- **Denoising AE** [Vincent 2008] **/ Masked AE**
  - Learning text representations
  - [Collobert-Weston 2011], BERT, RoBERTa...



This is a [...] of text extracted [...] a large set of [...] articles

This is a piece of text extracted from a large set of news articles



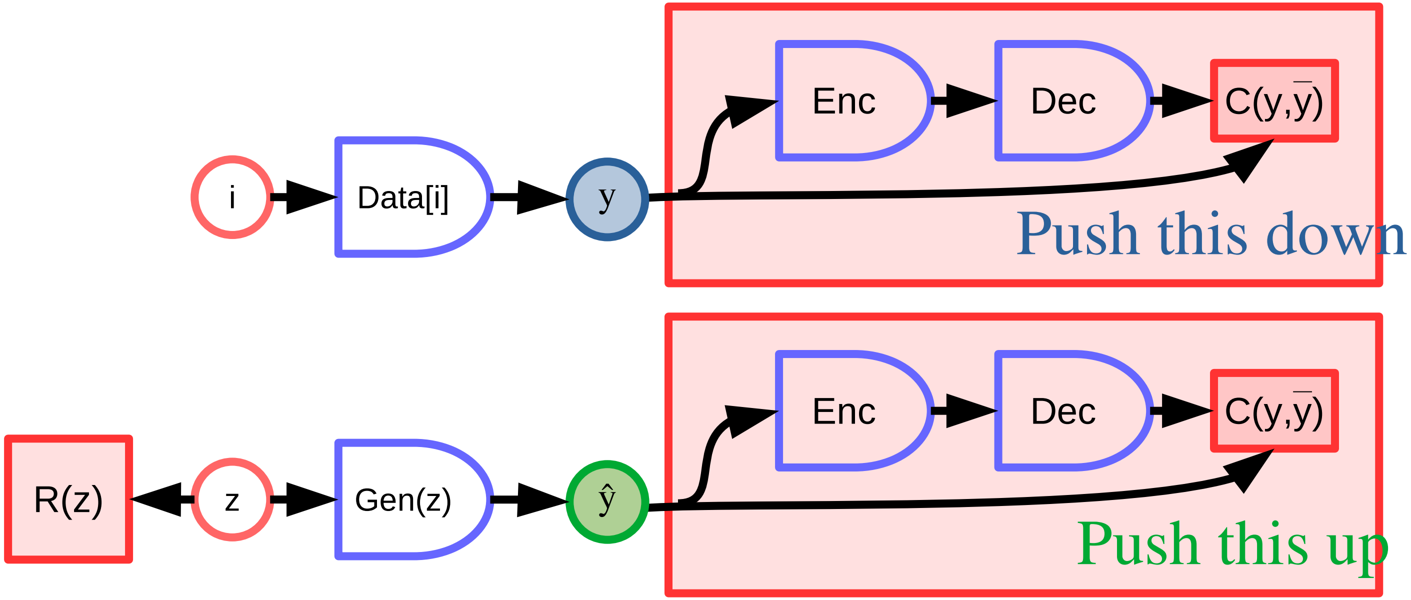


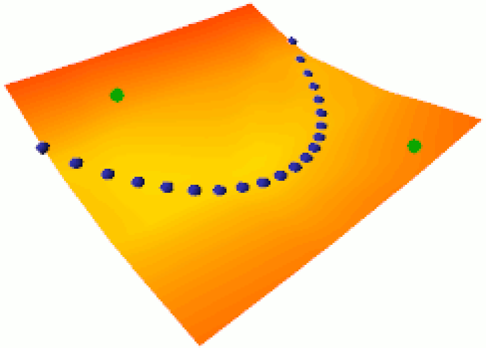
Figures: Alfredo Canziani

GANs: training a network to generate contrastive samples

- **Energy-Based GAN** [Zhao 2016], **Wasserstein GAN** [Arjovsky 2017],...
  - GANs generate nice images
  - But learning representations of image has not been successful.

$$\mathcal{L}(x, y, \hat{y}, w) = H(F_w(x, y), F_w(x, \hat{y}), m(y, \hat{y}))$$



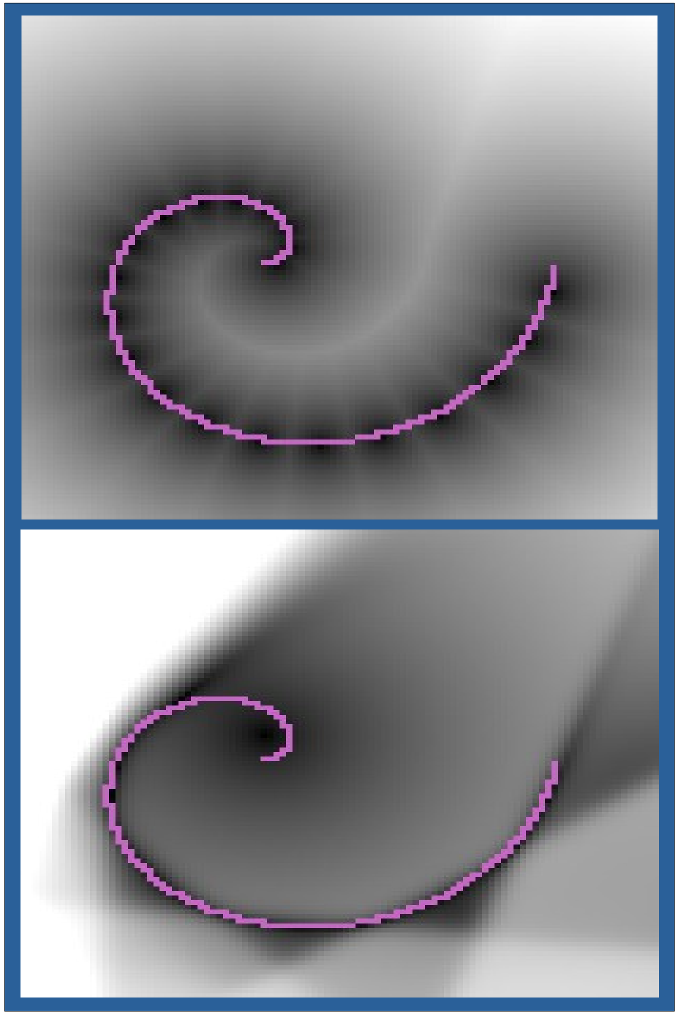


Architectural Methods & Regularized Methods

- **Different ways to limit the information capacity of the representation**

- **A1: build the machine so that the volume of the low energy regions is bounded:**
  - K-means, Gaussian Mixture Model, PCA, Bottleneck AE, Discretized AE (VQVAE),…

- **A2: regularize the volume of the low energy regions:**
  - Sparse coding, Sparse Auto-Encoder, LISTA, Variational Auto-Encoder.



Architectural Methods

- **A1: build the machine so that the volume of the low energy regions is bounded:**
  - K-means, Gaussian Mixture Model, PCA, Bottleneck AE, Discretized AE (VQVAE),…

- PCA

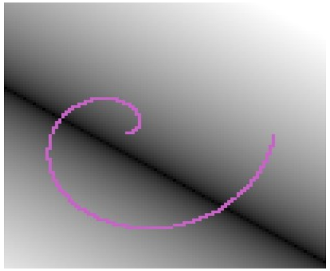
$$F(y) = ||y - w^t wy||^2$$

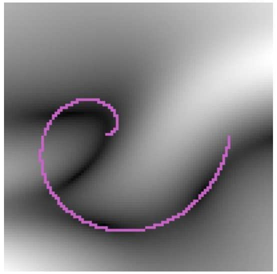
- Bottleneck AE

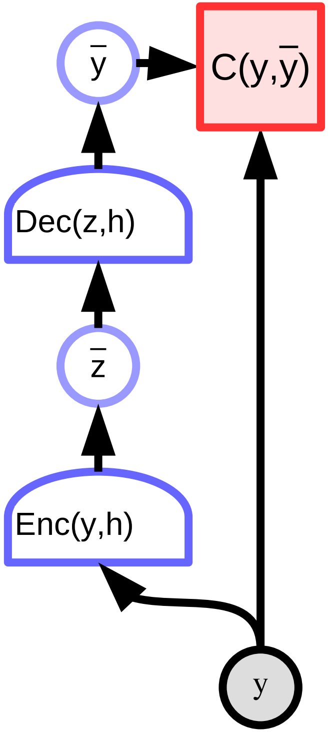
$$F(y) = C(y, \bar{y})$$

$$\bar{y} = \mathrm{Dec}(\bar{z})$$

$$\bar{z} = \mathrm{Enc}(y)$$



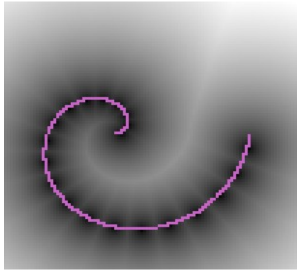


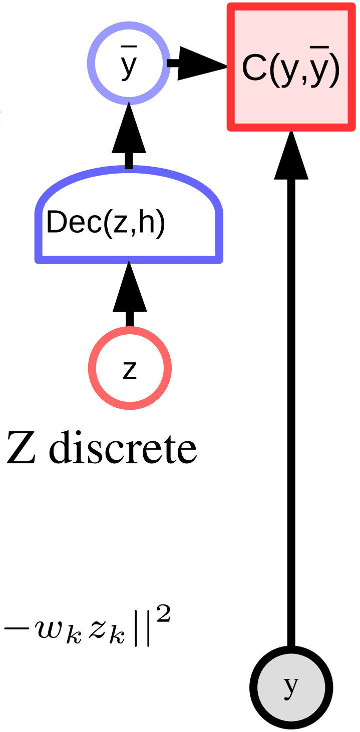


- K-means: z is a k-dim, 1-hot vector

$$E(y, z) = ||y - wz||^2$$

$$F(y) = \min_z ||y - wz||^2$$





- Gaussian Mixture

$$F(y) = -\log \sum_k \frac{e^{u_k}}{\sum_q e^{u_q}} e^{-||y - w_k z_k||^2}$$