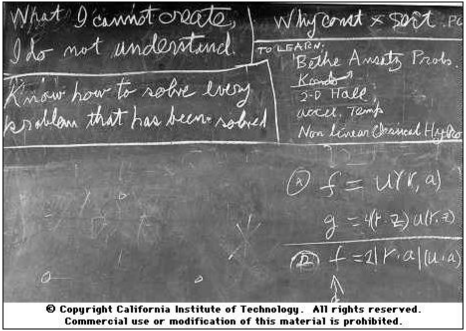
Generative vs Discriminative

MIPT

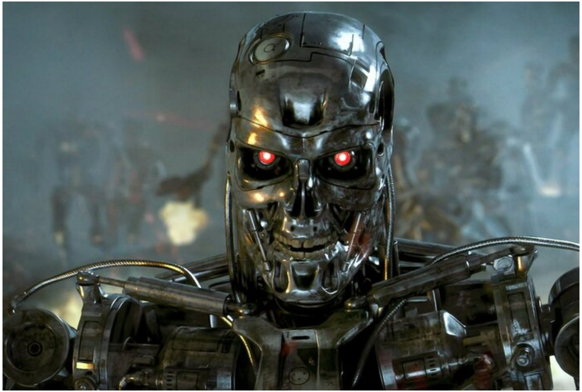
2023

Idea of generative models



Idea of discriminative models

Plato: *"A human is featherless biped"*



Sometimes it's easier to solve a target problem (i.e. classification, regression) than describe the analyzed object nature.

Generative and discriminative models

Discriminative models

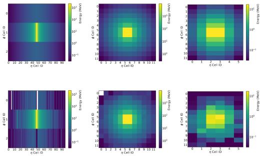
Model: $p(y|\mathbf{x})$.

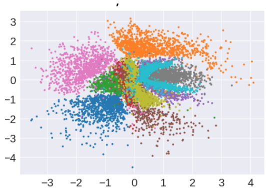
Generative models

Model: $p(y, \mathbf{x})$.

Why generative models:

- When dataset generation is a target problem
- Synthetic dataset generation
- Latent properties obtaining





Model selection: coherent Bayesian inference

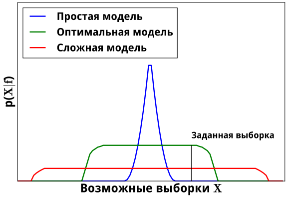
First level: find optimal parameters:

$$\mathbf{w} = \arg\max \frac{p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})}{p(\mathfrak{D}|\mathbf{h})},$$

Second level: find optimal model:

Evidence:

$$p(\mathfrak{D}|\mathbf{h}) = \int_{\mathbf{w}} p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})d\mathbf{w}.$$



What is $\mathfrak{D}$ for generative and discriminative models? Why?

Plate notation

Plate notation is an alternative visuzliation for graphical models.

Elements:

- White circles (random variables);
- Grey circels(observed variables);
- Small circles (deterministic values);
- Plates (batching).

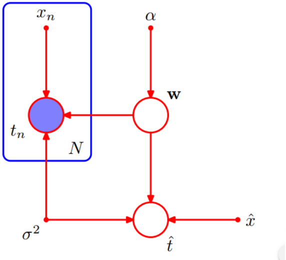
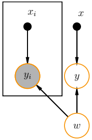


Plate notation for linear regression (Bishop)

Plate notation: discriminative and generative models

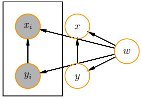
Discriminative models:

- Generate (or deterministically obtain!) x
- Generate w
- Generate $Y \sim p(y|X, w)$



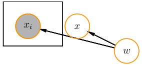
Generative model:

- Generate y
- Generate w
- Generate $x \sim p(X|y, w)$



Generative unsupervised model:

- Generate w
- Generate $x \sim p(X|w)$



Generative models and unsupervised learning

Are the generative models always unsupervised?

Generative models and unsupervised learning

Are the generative models always unsupervised?
No! Linear classification is an example

Logistic regression:

$$\mathrm{E}\left(\mathbf{y}\,|\mathbf{X}\,\right) \equiv g^{-1}\left(\mathbf{X}\mathbf{w}\right),$$

$$g^{-1}(x)\frac{e^x}{1+e^x} \in [0,1]$$

The decision function is a sigmoid.

Generative model:

$$p(y=1|x,w) = \frac{p(x|w,y=1)p(y=1)}{\sum_{k=0}^{1} p(x|w,y=k)p(y=k)},$$

$$p(x|w,y=k) \sim \mathcal{N}(w_m^k, w_s^k).$$

The decision function is a sigmoid.

Discriminative + generative

Naive approach: introduce a prior on class labels

$$p(\mathbf{x}, y|\mathbf{w}) = p(y|\mathbf{w}_y)p(x|y, \mathbf{w}_x).$$

Two optimization functions:

$$L_G = p(\mathbf{w}) \prod_{\mathbf{x},y} p(\mathbf{x}, y|\mathbf{w}),$$

$$L_D = p(\mathbf{w}) \prod_{\mathbf{x},y} p(y|\mathbf{x}, \mathbf{w}).$$

Combine them:

$$\lambda L_G + (1 - \lambda) L_D \to \max .$$

This optimization is heuristic, it does not give us ML results, nor MAP.

Discriminative + generative

(Bishop et al., 2007): introduce two probabilistic models: "discriminative" and "generative":

$$p(\mathbf{x}, y|\mathbf{w}_G, \mathbf{w}_D) = p(y|\mathbf{x}, \mathbf{w}_D)p(\mathbf{x}|\mathbf{w}_G)p(\mathbf{w}_G, \mathbf{w}_D).$$

Optimization:

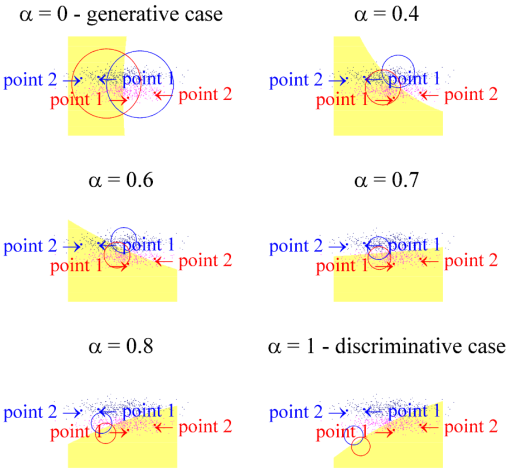
$$p(\mathbf{w}_G, \mathbf{w}_D)\prod_{\mathbf{x},y} p(y|\mathbf{x}, \mathbf{w}_D)p(\mathbf{x}|\mathbf{w}_G).$$

How to select $p(\mathbf{w}_G, \mathbf{w}_D)$?

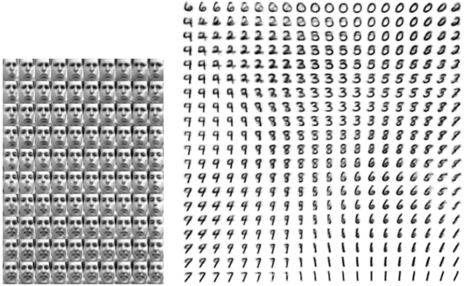
- $p(\mathbf{w}_G, \mathbf{w}_D) = p(\mathbf{w}_G)p(\mathbf{w}_D)$: obtain L_D;
- $p(\mathbf{w}_G, \mathbf{w}_D) = p(\mathbf{w}_G)\delta(\mathbf{w}_G - \mathbf{w}_D)$: obtain L_G;
- Trade-off: $p(\mathbf{w}_G, \mathbf{w}_D) \propto p(\mathbf{w}_G)p(\mathbf{w}_D)\exp(-\frac{1}{2\sigma^2}||\mathbf{w}_G - \mathbf{w}_D||^2)$.

Discriminative + generative

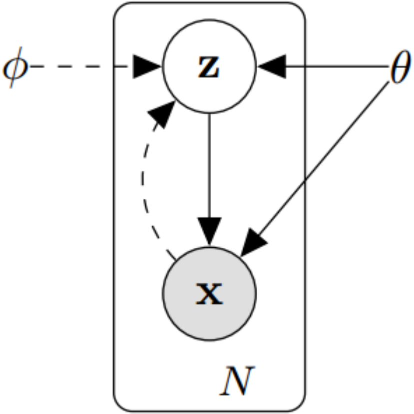
(Bishop et al., 2007): example of different combinations of these optimizations for the synthetic dataset. The dataset contains only 2 labeled objects for each class.



VAE: generation process



(a) Learned Frey Face manifold (b) Learned MNIST manifold



Semi-supervised VAE (Kingma et al., 2014)

$$\text{M1: } q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(\mathbf{x}), \text{diag}(\boldsymbol{\sigma}^2_\phi(\mathbf{x}))), \tag{3}$$

$$\text{M2: } q_\phi(\mathbf{z}|y, \mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}_\phi(y, \mathbf{x}), \text{diag}(\boldsymbol{\sigma}^2_\phi(\mathbf{x}))); \;\; q_\phi(y|\mathbf{x}) = \text{Cat}(y|\boldsymbol{\pi}_\phi(\mathbf{x})), \tag{4}$$

For this model, we have two cases to consider. In the first case, the label corresponding to a data point is observed and the variational bound is a simple extension of equation (5):

$$\log p_\theta(\mathbf{x}, y) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},y)}\left[\log p_\theta(\mathbf{x}|y, \mathbf{z}) + \log p_\theta(y) + \log p(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x}, y)\right] = -\mathcal{L}(\mathbf{x}, y), \tag{6}$$

For the case where the label is missing, it is treated as a latent variable over which we perform posterior inference and the resulting bound for handling data points with an unobserved label y is:

$$\begin{aligned} \log p_\theta(\mathbf{x}) &\geq \mathbb{E}_{q_\phi(y,\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}|y, \mathbf{z}) + \log p_\theta(y) + \log p(\mathbf{z}) - \log q_\phi(y, \mathbf{z}|\mathbf{x})\right] \\ &= \sum_y q_\phi(y|\mathbf{x})(-\mathcal{L}(\mathbf{x}, y)) + \mathcal{H}(q_\phi(y|\mathbf{x})) = -\mathcal{U}(\mathbf{x}). \end{aligned} \tag{7}$$

The bound on the marginal likelihood for the entire dataset is now:

$$\mathcal{J} = \sum_{(\mathbf{x},y)\sim\widetilde{p}_l} \mathcal{L}(\mathbf{x}, y) + \sum_{\mathbf{x}\sim\widetilde{p}_u} \mathcal{U}(\mathbf{x}) \tag{8}$$

Semi-supervised VAE (Kingma et al., 2014)

$$\mathcal{J}^{\alpha} = \mathcal{J} + \alpha \cdot \mathbb{E}_{\tilde{p}_l(\mathbf{x},y)}\left[-\log q_{\phi}(y|\mathbf{x})\right], \tag{9}$$

Algorithm 1 Learning in model M1

while generativeTraining() **do**
  $\mathcal{D} \leftarrow$ getRandomMiniBatch()
  $\mathbf{z}_i \sim q_{\phi}(\mathbf{z}_i|\mathbf{x}_i) \quad \forall \mathbf{x}_i \in \mathcal{D}$
  $\mathcal{J} \leftarrow \sum_n \mathcal{J}(\mathbf{x}_i)$
  $(\mathbf{g}_{\theta}, \mathbf{g}_{\phi}) \leftarrow (\frac{\partial \mathcal{J}}{\partial \theta}, \frac{\partial \mathcal{J}}{\partial \phi})$
  $(\boldsymbol{\theta}, \boldsymbol{\phi}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{\phi}) + \boldsymbol{\Gamma}(\mathbf{g}_{\theta}, \mathbf{g}_{\phi})$
end while
while discriminativeTraining() **do**
  $\mathcal{D} \leftarrow$ getLabeledRandomMiniBatch()
  $\mathbf{z}_i \sim q_{\phi}(\mathbf{z}_i|\mathbf{x}_i) \quad \forall \{\mathbf{x}_i, y_i\} \in \mathcal{D}$
  trainClassifier($\{\mathbf{z}_i, y_i\}$)
end while

Algorithm 2 Learning in model M2

while training() **do**
  $\mathcal{D} \leftarrow$ getRandomMiniBatch()
  $y_i \sim q_{\phi}(y_i|\mathbf{x}_i) \quad \forall \{\mathbf{x}_i, y_i\} \notin \mathcal{O}$
  $\mathbf{z}_i \sim q_{\phi}(\mathbf{z}_i|y_i, \mathbf{x}_i)$
  $\mathcal{J}^{\alpha} \leftarrow$ eq. (9)
  $(\mathbf{g}_{\theta}, \mathbf{g}_{\phi}) \leftarrow (\frac{\partial \mathcal{L}^{\alpha}}{\partial \theta}, \frac{\partial \mathcal{L}^{\alpha}}{\partial \phi})$
  $(\boldsymbol{\theta}, \boldsymbol{\phi}) \leftarrow (\boldsymbol{\theta}, \boldsymbol{\phi}) + \boldsymbol{\Gamma}(\mathbf{g}_{\theta}, \mathbf{g}_{\phi})$
end while

Semi-supervised VAE (Kingma et al., 2014)

Table 1: Benchmark results of semi-supervised classification on MNIST with few labels.

N	NN	CNN	TSVM	CAE	MTC	AtlasRBF	M1+TSVM	M2	M1+M2
100	25.81	22.98	16.81	13.47	12.03	8.10 ($\pm$ 0.95)	11.82 ($\pm$ 0.25)	11.97 ($\pm$ 1.71)	**3.33** ($\pm$ 0.14)
600	11.44	7.68	6.16	6.3	5.13	–	5.72 ($\pm$ 0.049)	4.94 ($\pm$ 0.13)	**2.59** ($\pm$ 0.05)
1000	10.7	6.45	5.38	4.77	3.64	3.68 ($\pm$ 0.12)	4.24 ($\pm$ 0.07)	3.60 ($\pm$ 0.56)	**2.40** ($\pm$ 0.02)
3000	6.04	3.35	3.45	3.22	2.57	–	3.49 ($\pm$ 0.04)	3.92 ($\pm$ 0.63)	**2.18** ($\pm$ 0.04)

(a) Handwriting styles for MNIST obtained by fixing the class label and varying the 2D latent vari

Generative and discriminative models hierarchy

Can we reduce generative model to disciriminative?
Can we reduce disciriminative model to generative?

Basic Concept of EBM[1]

Energy-Based Models (EBMs) capture dependencies by associating a scalar energy to each configuration of the variables

Learning: finding an energy function that associates low energies to correct values, and higher energies to incorrect values

Inference: finding parameter that minimize the energy

Pros: no requirement for proper normalization, in comparison with probabilistic models

Probability density

$p(\mathbf{x})$ for $\mathbf{x} \in \mathbb{R}^D$: $p_\theta(\mathbf{x}) = \frac{\exp(-E_\theta(\mathbf{x}))}{Z(\theta)}$, where $E_\theta(\mathbf{x})$ is energy function and $Z(\theta) = \int_{\mathbf{x}} \exp(-E_\theta(\mathbf{x})) d\mathbf{x}$ is normalizing constant or partition function

[1] See talk of Maria Kovaleva, 2022

Learning of EBM

Estimate $Z(\theta)$ can be challenging for most energy functions. Derivative of the log-likelihood for a single example x with respect to θ: $\frac{\partial \log p_\theta(\mathbf{x})}{\partial \theta} = \mathbb{E}_{p_\theta(\mathbf{x}')}\left[\frac{\partial E_\theta(x')}{\partial \theta}\right] - \frac{\partial E_\theta(x)}{\partial \theta}$

Learning

Approximate the expectation in derivative of the log-likelihood using a sampler based on Stochastic Gradient Langevin Dynamics (SGLD):

$\mathbf{x}_0 \sim p_0(\mathbf{x})$, $\mathbf{x}_{i+1} = \mathbf{x}_i - \frac{\alpha}{2}\frac{\partial E_\theta(\mathbf{x}_i)}{\partial \mathbf{x}_i} + \epsilon$, $\epsilon \sim \mathcal{N}(0, \alpha)$,

where $p_0(\mathbf{x})$ is typically a Uniform distribution, α - step-size (should be decayed following a polynomial schedule).

Practically α and ϵ is often chosen separately leading to a biased sampler which allows for faster training

In classifiers

Classification: D parameters, K classes, parametric function, $f_\theta : \mathbb{R}^D \to \mathbb{R}^K$ return logits
Categorical distribution parameterized by f_θ via Softmax transfer function:
$p_\theta(y|\mathbf{x}) = \frac{\exp(f_\theta(\mathbf{x})[y])}{\sum_{y'} \exp(f_\theta(\mathbf{x})[y'])}$

Reinterpretation of the logits

Let's define an energy based model with $E_\theta(\mathbf{x}, y) = -f_\theta(\mathbf{x})[y]$ and unknown normalizing constant Z_θ :

$$p_\theta(\mathbf{x}, y) = \frac{\exp\left(f_\theta(\mathbf{x})[y]\right)}{Z(\theta)}$$

Then

$$p_\theta(\mathbf{x}) = \sum_y p_\theta(\mathbf{x}, y) = \frac{\sum_y \exp\left(f_\theta(\mathbf{x})[y]\right)}{Z(\theta)}$$

Conclusion

The LogSumExp$(\cdot)$ of the logits of any classifier can be re-used to define the energy function as $E_\theta(\mathbf{x}) = -LogSumExp(f_\theta(\mathbf{x})[y]) = -\log \sum_y \exp f_\theta(\mathbf{x})[y]$
A generative model hidden within every standard discriminative model!

Optimization

$\log p_\theta(\mathbf{x}, y) = \log p_\theta(\mathbf{x}) + \log p_\theta(y|\mathbf{x})$
Where:
1. $p(y|\mathbf{x})$ optimized using standard cross-entropy
2. $\log p(\mathbf{x})$ optimized with SGLD where gradients are taken with respect to $LogSumExp(f_\theta(\mathbf{x})[y])$ as it described for EBMs

Remarks

1. We don't know Z_θ, because of it we use SLGD sampler
2. The estimator of $\log p(\mathbf{x})$ will be biased when using a MCMC sampler with a finite number of steps

Classifier as generative model: results

Figure 7: Class-conditional Samples. Left to right: CIFAR10, SVHN.

Model selection problem: recap

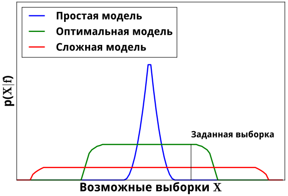
First level: find optimal parameters:

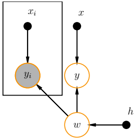
$$\mathbf{w} = \arg\max \frac{p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})}{p(\mathfrak{D}|\mathbf{h})},$$

Second level: find optimal model:

Evidence:

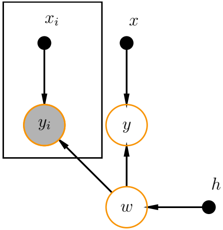
$$p(\mathfrak{D}|\mathbf{h}) = \int_{\mathbf{w}} p(\mathfrak{D}|\mathbf{w})p(\mathbf{w}|\mathbf{h})d\mathbf{w}.$$

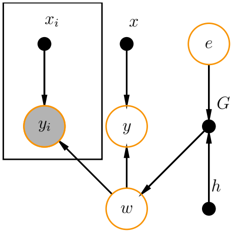




Model selection problem: recap

Can we generate target models parameters using a generative model?





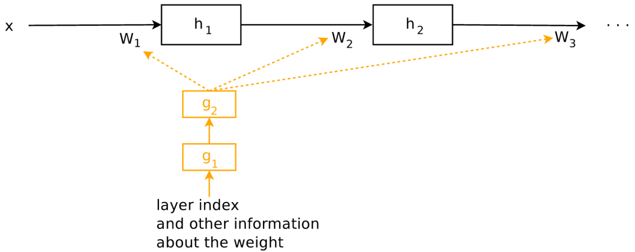
Model selection: hybrid approach

Definition

Given a set Λ.
Hypernetwork is a parametric mapping from Λ to set $\mathbb{R}^n$ of the model $\mathbf{f}$ parameters:

$$G : \Lambda \times \mathbb{R}^u \to \mathbb{R}^n,$$

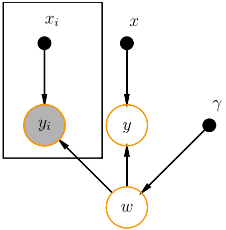
where $\mathbb{R}^u$ is a set of hypernetwork parameters.



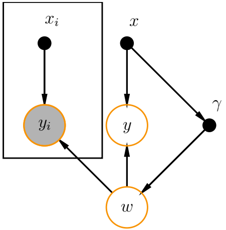
Ha et al., 2016

Model selection: discriminative approach

$$\mathbf{w}_{\mathsf{MOE}} = \langle \gamma(\mathbf{x}), [\mathbf{w}_1, \ldots, \mathbf{w}_n] \rangle$$



Model generation scheme



MOE optimization as a discriminative model

References

- Bishop C. M. Pattern recognition //Machine learning. – 2006. – Т. 128. – №. 9.
- Генератор котиков: https://github.com/aleju/cat-generator
- Paganini M., de Oliveira L., Nachman B. Accelerating science with generative adversarial networks: an application to 3D particle showers in multilayer calorimeters //Physical review letters. – 2018. – Т. 120. – №. 4. – С. 042003.
- Antoran J., Miguel A. Disentangling and learning robust representations with natural clustering //2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA). – IEEE, 2019. – С. 694-699.
- Лекции по LDA: https://personal.utdallas.edu/~nrr150130/cs6347/2017sp/lects/Lecture_18_LDA.pdf
- Bernardo J. M. et al. Generative or discriminative? getting the best of both worlds //Bayesian statistics. – 2007. – Т. 8. – №. 3. – С. 3-24.
- Гребенькова О. С., Бахтеев О. Ю., Стрижов В. В. Вариационная оптимизация модели глубокого обучения с контролем сложности //Информатика и её применения. – 2021. – Т. 15. – №. 1. – С. 42-49.
- Ha D., Dai A., Le Q. V. Hypernetworks //arXiv preprint arXiv:1609.09106. – 2016.
- Адуенко А. А. Выбор мультимоделей в задачах классификации : дис. – Федер. исслед. центр"Информатика и управление"РАН, 2017.
- Grathwohl W. et al. Your classifier is secretly an energy based model and you should treat it like one //arXiv preprint arXiv:1912.03263. – 2019.