باسمه تعالی

فاز دوم پروژه‌ی درس مقدمه ای بر یادگیری ماشین

مدل های گرافی و ماشین بولتزمن

استاد درس

دکتر محمّدحسین یاسائی میبدی

دانشکده‌ی مهندسی برق
دانشگاه صنعتی شریف

پاییز ۱۴۰۳

آخرین مهلت تحویل:
۱۵ بهمن ۱۴۰۳

فهرست مطالب

حتما قبل از شروع پروژه بخش ۳ را مطالعه نمایید.

۱ بولتزمن امن

در فاز قبلی با ماشین بولتزمن آشنا شدید و نحوه استفاده از آن و به روز رسانی پارامتر ها را فراگرفتید. در این قسمت از فاز دوم پروژه، قصد داریم ماشین بولتزمن را به صورت امن پیاده سازی کنیم.
فرض کنید که دو گروه A و B به ترتیب تعداد m_A و m_B نورون نمایان و پنهان دارند. این دو گروه می خواهند به طور مشترک ماشین بولتزمن را آموزش بدهند بطوریکه هیچکدام از گروه ها نباید از مقادیر نورون های یکدیگر مطلع بشوند. به همین دلیل از یک الگوریتم رمز گذاری استفاده می کنیم که در قسمت های بعدی با آن آشنا می شوید.

پرسش تئوری ۱. الگوریتم رمزگذاری ElGamal را کامل توضیح داده و روابط آن را بنویسید.

پرسش تئوری ۲. رمز گشایی جزئی را توضیح دهید.

۱.۱ نمونه برداری گیبس

همانطور که می دانید با نمونه برداری گیبس می توان مقادیر نورون های پنهان و نمایان را به روز رسانی کرد.
الگوریتم به روز رسانی نورون ها به شرح زیر می باشد:
همانطور که می دانید که متغیر های پنهان و نمایان با احتمال های زیر مقدار دهی می شوند:

$$h^{(n)} \sim sigmoid(W'.v^{(n)} + c)$$
$$v^{(n+1)} \sim sigmoid(W.h^{(n)} + b).$$

اما در این الگوریتم رمزگذاری، مقدار هر نورون را دقیقا برابر مقادیر گفته شده به عنوان پارامتر توزیع برنولی در عبارت قبل مقدار دهی می کنیم. حال برای به روز رسانی مقادیر نورون ها بدین صورت عمل می کنیم:

۱. تقسیم داده ها بین دو گروه A و B

- داده‌ها به صورت عمودی تقسیم می‌شوند، به‌طوری که هر طرف فقط به بخشی از داده‌ها دسترسی دارد.
- به عنوان ورودی:
  - گروه A مجموعه داده $v_A^\circ = (v_1^\circ, v_2^\circ, \dots, v_{m_A}^\circ)$ را دارد.
  - گروه B مجموعه داده $v_B^\circ = (v_{m_1+1}^\circ, \dots, v_{m_A+m_B}^\circ)$ را دارد.

۲. محاسبه مجموع وزن‌دار ورودی‌ها

- هر طرف ابتدا داده‌های قابل مشاهده خود را در ضرایب وزن w_{ij} ضرب کرده و مجموع مقادیر را محاسبه می‌کند. این کار شامل واحد های نمایان و پنهان می باشد.

۳. رمزنگاری توسط گروه A

- گروه A تمام مقادیر مجموع وزن‌دار ورودی‌های خود را با تمامی حالات ممکن برای مجموع وزن دار ورودی های گروه B جمع می کند(هر کدام از این مجموع ها بایستی به عددی صحیح گرد بشود) و سپس از این مجموع تابع سیگموید می گیرد و از حاصل تابع عدد تصادفی R را کم می کند.
- حال به ترتیب حالات ممکن برای مجموع وزن دار ورودی های گروه B این مقادیر را رمزگذاری کرده و به گروه B می فرستد.

پیامی که گروه A در نهایت می فرستند به صورت رمز شده عبارات به صورت زیر می باشد:

$$\text{sigmoid}\left(\sum_{k \le m_A} (w_{jk} v_k^\circ + c_k) + i\right) - R.$$

۴. ارسال به گروه B و بازگشت اطلاعات رمزگشایی‌شده

- گروه B پیام متناظر با مجموع وزن دار ورودی های خودش را رمزگشایی کرده و این پیام را نویزی کرده و آن را به گروه A می فرستد.

۵. رمزگشایی جزئی

- حال گروه A پیام نویزی شده ارسالی از گروه B را به صورت جزئی رمزگشایی کرده و دوباره برای گروه B آن را ارسال می کند.

۶. رمزگشایی جزئی دیگر!

- حال گروه B پیامی که گروه A به صورت جزئی رمزگشایی کرده بود را دوباره رمزگشایی می کند. اینگونه گروه B پیام

$$\text{sigmoid}\left(\sum_{k\le m_A}(w_{jk}v_k^{\circ}+c_k)+\sum_{m_A\le k\le m_A+m_B}(w_{jk}v_k^{\circ}+c_k)\right)-R$$

را خواهد داشت که این مقدار از گروه B و مقدار R از گروه A باعث می شود که مقدار اصلی برای به روز رسانی مقادیر ورودی ها بر اساس نمونه برداری گیبس به دست بیاید. یعنی

$$h_j^{\backprime}=\text{sigmoid}\left(\sum_{k\le m_A}(w_{jk}v_k^{\circ}+c_k)+\sum_{m_A\le k\le m_A+m_B}(w_{jk}v_k^{\circ}+c_k)\right)$$

به همین ترتیب بقیه ورودی های نمایان و پنهان مقدار دهی می شوند.

برای درک بهتر الگوریتم، به کد قرار داده شده برای جمع امن دو عدد کمتر مساوی ۱۰ مراجعه کنید.[۱]

پرسش تئوری ۳. الگوریتمی مشابه آنچه برای محاسبه سیگموید در بالا دید برای محاسبه ضرب دو عدد ارائه دهید، فرض کنید پارتی A عدد M را میداند و پارتی B عدد N را میداند. و میخواهند MN را محاسبه کنند.

۲.۱ به روز رسانی امن پارامتر ها

میدانیم در ماشین بولتزمن گرادیان از رابطه زیر بدیت می آید(با فرض یک مرحله برای گیبس سمپلینگ):

$$\left(\langle v_i^{\circ}h_j^{\circ}\rangle_{\text{data}}-\langle v_i^{\backprime}h_j^{\backprime}\rangle_{\text{model}}\right)$$

که برای مسئله دو پارتی ما میتوان آنرا به شکل زیر نوشت:

$$h^{\circ}=h_1^{\circ}+h_2^{\circ},\quad V^{\backprime}=V_1^{\backprime}+V_2^{\backprime},\quad \text{و}\quad h^{\backprime}=h_1^{\backprime}+h_2^{\backprime},$$

$$V^{\circ}h^{\circ}-V^{\backprime}h^{\backprime}=V^{\circ}(h_1^{\circ}+h_2^{\circ})-(V_1^{\backprime}+V_2^{\backprime})(h_1^{\backprime}+h_2^{\backprime}).$$

$$V^{\circ}h^{\circ}-V^{\backprime}h^{\backprime}=V^{\circ}h_1^{\circ}+V^{\circ}h_2^{\circ}-V_1^{\backprime}h_1^{\backprime}-V_1^{\backprime}h_2^{\backprime}-V_2^{\backprime}h_1^{\backprime}-V_2^{\backprime}h_2^{\backprime}.$$

که در آن عدد بالا نشان‌دهنده مرحله گیبس و عدد پایین نشان‌دهنده طرفی است که داده متعلق به آن است. حال الگوریتمی ارائه میدهیم که با استفاده از آن میتوان به صورت امن عبارت بالا را محاسبه کرد.

مرحله ۱: مقداردهی اولیه

- وزن‌ها (W)، بایاس لایه ورودی (b) و بایاس لایه مخفی (c) را با مقادیر کوچک تصادفی مقداردهی اولیه کنید.
- مقادیر اولیه W، b، و c را با هر دو طرف A و B به اشتراک بگذارید.

[۱] Secure_Addition.ipynb - Colab.pdf

مرحله ۲: نمونه‌گیری گیبس
هدف این مرحله نمونه‌گیری ایمن از لایه‌های مخفی و مرئی با استفاده از یک مرحله نمونه‌گیری گیبس است.

مرحله ۱.۲: محاسبه احتمالات لایه مخفی ($h_j^{(0)}$)

- **طرف A:**

$$\sum_{k \le m_A} (w_{jk} V_k^{(0)} + c_k).$$

- **طرف B:**

$$\sum_{m_A < k \le m_A + m_B} (w_{jk} V_k^{(0)} + c_k).$$

- **هر دو طرف:** با استفاده از **الگوریتم ۱**، تابع سیگموئید برای کل مقدار محاسبه می‌شود:

$$h_j^{(0)} = h_{j1}^{(0)} + h_{j2}^{(0)},$$

که $h_{j1}^{(0)}$ و $h_{j2}^{(0)}$ سهم‌های تصادفی مربوط به طرف A و B هستند.

مرحله ۲.۲: محاسبه احتمالات لایه مرئی ($v_i^{(1)}$)

- **طرف A:**

$$\sum_{k \le m_A} (w_{ik} h_k^{(0)} + b_k).$$

- **طرف B:**

$$\sum_{m_A < k \le m_A + m_B} (w_{ik} h_k^{(0)} + b_k).$$

- **هر دو طرف:** با استفاده از **الگوریتم ۱**، تابع سیگموئید برای کل مقدار محاسبه می‌شود:

$$v_i^{(1)} = v_{i1}^{(1)} + v_{i2}^{(1)},$$

که $v_{i1}^{(1)}$ و $v_{i2}^{(1)}$ سهم‌های تصادفی مربوط به طرف A و B هستند.

مرحله ۳.۲: محاسبه احتمالات لایه مخفی ($h_j^{(1)}$)

- **طرف A:**

$$\sum_{k \le m_A} (w_{jk} v_k^{(1)} + c_k).$$

- **طرف B:**

$$\sum_{m_A < k \le m_A + m_B} (w_{jk} v_k^{(1)} + c_k).$$

- **هر دو طرف:** با استفاده از **الگوریتم ۱**، تابع سیگموئید برای کل مقدار محاسبه می‌شود:

$$h_j^{(1)} = h_{j1}^{(1)} + h_{j2}^{(1)}.$$

مرحله ۳: به‌روزرسانی وزن‌ها

مرحله ۱.۳: محاسبه مقادیر لازم برای به‌روزرسانی وزن‌ها

- طرف A محاسبه می‌کند:

$$V_1^{(0)} h_1^{(0)}, \quad V_1^{(1)} h_1^{(1)}$$

- طرف B محاسبه می‌کند:

$$V_2^{(0)} h_2^{(0)}, \quad V_2^{(1)} h_2^{(1)}$$

این مقادیر سهم‌های محلی هر طرف از ارتباطات مثبت و منفی برای به‌روزرسانی وزن‌ها هستند.

مرحله ۲.۳: محاسبه ایمن ضرب‌های متقابل

برای محاسبه ترم‌های متقابل (مانند $V_1^{(0)} h_2^{(0)}$ و $V_2^{(0)} h_1^{(0)}$)، هر دو طرف از **الگوریتم ضرب** استفاده می‌کنند:

- **ارتباطات مثبت ($V_1^{(0)} h_2^{(0)}$):**
  - طرفین از **الگوریتم ضرب** برای محاسبه ایمن ضرب $V_1^{(0)} h_2^{(0)}$ استفاده می‌کنند.
  - این محاسبه منجر به دو سهم تصادفی می‌شود:

$$r_{11}^{(0)}, \quad r_{12}^{(0)},$$

به‌طوری‌که:

$$r_{11}^{(0)} + r_{12}^{(0)} = V_1^{(0)} h_2^{(0)}.$$

- **ارتباطات مثبت ($V_2^{(0)} h_1^{(0)}$):**
  - به‌طور مشابه، طرفین $V_2^{(0)} h_1^{(0)}$ را با استفاده از **الگوریتم ضرب** محاسبه می‌کنند.
  - این محاسبه منجر به دو سهم تصادفی می‌شود:

$$r_{21}^{(0)}, \quad r_{22}^{(0)},$$

به‌طوری‌که:

$$r_{21}^{(0)} + r_{22}^{(0)} = V_2^{(0)} h_1^{(0)}.$$

- **ارتباطات منفی ($V_1^{(1)} h_2^{(1)}$):**
  - طرفین از **الگوریتم ضرب** برای محاسبه $V_1^{(1)} h_2^{(1)}$ استفاده می‌کنند، و به مقادیر زیر می‌رسند:

$$r_{11}^{(1)}, \quad r_{12}^{(1)},$$

به‌طوری‌که:

$$r_{11}^{(1)} + r_{12}^{(1)} = V_1^{(1)} h_2^{(1)}.$$

- **ارتباطات منفی ($V_2^{(1)} h_1^{(1)}$):**
  - به‌طور مشابه، طرفین $V_2^{(1)} h_1^{(1)}$ را محاسبه می‌کنند، و به مقادیر زیر می‌رسند:

$$r_{21}^{(1)}, \quad r_{22}^{(1)},$$

به‌طوری‌که:

$$r_{21}^{(1)} + r_{22}^{(1)} = V_2^{(1)} h_1^{(1)}.$$

مرحله ۳.۳: محاسبه گرادیان برای به‌روزرسانی وزن‌ها

- **اگر طرف A واحد مرئی $V_1^{(0)}$ را داشته باشد:**
  - طرف A محاسبه می‌کند:

$$G_A = V_1^{(0)} h_1^{(0)} + r_{11}^{(0)} - V_1^{(1)} h_1^{(1)} - r_{11}^{(1)} - r_{21}^{(1)}.$$

  - طرف B محاسبه می‌کند:

$$G_B = r_{12}^{(0)} - V_2^{(1)} h_2^{(1)} - r_{12}^{(1)} - r_{22}^{(1)}.$$

  - طرف B، مقدار G_B را به طرف A می‌فرستد.
  - طرف A، مقادیر $G_A + G_B$ را برای محاسبه گرادیان کامل جمع می‌کند:

$$G = G_A + G_B.$$

- **اگر طرف B واحد مرئی $V_0^{(2)}$ را داشته باشد:**

– فرآیند مشابه انجام می‌شود و طرف B گرادیان را ترکیب می‌کند.

مرحله ۴.۳: به‌روزرسانی وزن‌ها

- پس از محاسبه گرادیان G، طرف A (یا B) وزن‌ها را با استفاده از نزول گرادیان به‌روزرسانی می‌کند:

$$w_{\text{new}} = w_{\text{old}} - \eta G,$$

که در آن η نرخ یادگیری است.

مرحله ۵.۳: به‌روزرسانی بایاس‌ها

- با استفاده از روش مشابه، بایاس‌ها (b و c) به‌روزرسانی می‌شوند:

$$b_{\text{new}} = b_{\text{old}} - \eta \Delta b,$$

$$c_{\text{new}} = c_{\text{old}} - \eta \Delta c.$$

مرحله ۶.۳: بازگشت مقادیر به‌روزرسانی‌شده

- مقادیر به‌روزرسانی‌شده وزن (W) و بایاس (b, c) به هر دو طرف بازگردانده می‌شوند.

مرحله ۴: تکرار تا همگرایی

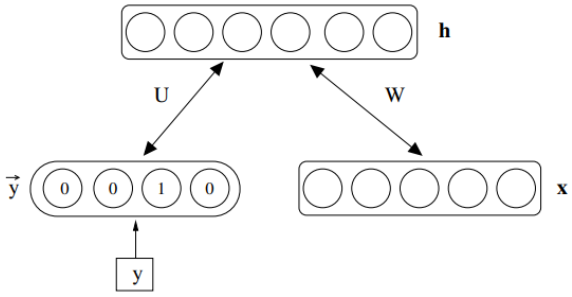
- نمونه‌گیری گیبس و به‌روزرسانی وزن‌ها را برای تمام نمونه‌های آموزشی تکرار کنید.
- تکرار کنید تا شرط توقف (مثلاً تعداد دوره‌های ثابت یا همگرایی وزن‌ها) برقرار شود.

پرسش شبیه‌سازی ۱. با مجموعه داده MNIST و به کارگیری ماشین بولتزمن امن به عنوان استخراج کننده داده، تسک طبقه بندی را انجام بدهید و عملکرد آن را با حالت ماشین بولتزمن ساده مقایسه کنید.

---

۲ ماشین بولتزمن تمایزگر

در قسمت قبل شما با مدل بولتزمن به عنوان یک مدل مولد ساده ولی قدرتمند آشنا شدید. حال، می‌خواهیم از قابلیت‌های این مدل، و به طول کل مدل های بر پایه انرژی برای طبقه بندی داده‌ها استفاده کنیم. در این بخش با مدل‌های بولتزمن تمایزگر (DRBM) [۲] آشنا خواهید شد. این مدل‌ها برخلاف RBM های استاندارد که برای یادگیری توزیع مشترک داده‌ها $p(x,y)$ طراحی شده‌اند، به طور خاص برای یادگیری توزیع شرطی بین داده‌ها و برچسب‌های آن‌ها $p(y|x)$ استفاده می‌شوند.



شکل ۱: ماشین بولتزمن تمایزگر

مدل بولتزمن تمایزگر با تغییر در تابع انرژی RBM استاندارد ساخته می‌شود. در DRBM برچسب داده y به عنوان بخشی از لایه ورودی مدل (لایه قابل مشاهده) اضافه می‌شود. تابع انرژی در DRBM به شکل زیر تعریف می‌شود:

$$E(x,y,h) = -\sum_i \sum_j W_{ij} x_i h_j - \sum_i b_i x_i - \sum_j c_j h_j - \sum_k d_k y_k$$

و در نهایت خواهیم داشت $p(x,y,h) \propto \exp(-E(x,y,h))$. برای باینری کردن برچسب ها و استفاده بهینه (و منطقی) از آنها، اینجا نیز y به صورت One-hot Encoding از برچسب ها استفاده میشود.
هدف طبقه بندی مناسب است. در نتیجه به دنبال بیشینه درست نمایی $p(y|x) \propto \exp(-\mathcal{F}(x,y))$ هستیم.
این توزیع بیانگر احتمال تخصیص یک برچسب خاص y به داده ورودی x و $\mathcal{F}$ بیانگر انرژی آزاد است.

پرسش تئوری ۴. با توجه به تابع انرژی در نظر گرفته شده $E(x,y,h)$، به پرسش های زیر پاسخ دهید.

۱. توزیع مشترک $p(y,x)$ را محاسبه کنید.

۲. توزیع شرطی $p(y|x)$ را محاسبه کنید.

۳. مسئله بهینه سازی لگاریتم درست نمایی $p(y,x)$ و گرادیان آن را به صورت ریاضی بنویسید. آیا این مسئله با روش های مرسوم بهینه سازی قابل حل است؟ اگر بله توضیح دهید، اگر نه برای آن راه حلی ارائه دهید.

۴. مسئله بهینه سازی لگاریتم درست نمایی $p(y|x)$ و گرادیان آن را به صورت ریاضی بنویسید. آیا این مسئله با روش های مرسوم بهینه سازی قابل حل است؟ اگر بله توضیح دهید، اگر نه برای آن راه حلی ارائه دهید.

۵. یک نمایش تابع انرژی آزاد $\mathcal{F}(x,y)$ را بدست بیاورید.

در مرحله اول، میتوان برای تولید $p(y|x)$ از $p(y,x)$ استفاده کنیم. همانطور که انتظار میرود، یادگیری $p(y,x)$ همانند یادگیری مدل بولتزمن عادی است، و به رابطه زیر منجر میشود.

$$\frac{\partial \log p(y_\circ, x_\circ)}{\partial \theta} = -\mathbb{E}_{h|y_\circ,x_\circ}\left[\frac{\partial}{\partial \theta} E(y_\circ, x_\circ, h)\right] + \mathbb{E}_{h,y,x}\left[\frac{\partial}{\partial \theta} E(y,x,h)\right].$$

برای این کار از الگوریتم یادگیری ماشین بولتزمن با استفاده از Constrastive-Divergence استفاده میکنیم. و سپس با استفاده از مدل نهایی، $p(y|x)$ را برای تمامی کلاس ها محاسبه کرده و طبقه بندی را میتوان انجام داد.

[۲] Discriminative Restricted Boltzmann Machine

Algorithm 1 Training update for RBM over $(y, \mathbf{x})$ using Contrastive Divergence

Input: training pair $(y_i, \mathbf{x}_i)$ and learning rate λ
% Notation: $a \leftarrow b$ means a is set to value b
% $a \sim p$ means a is sampled from p

% Positive phase
$y^0 \leftarrow y_i$, $\mathbf{x}^0 \leftarrow \mathbf{x}_i$, $\widehat{\mathbf{h}}^0 \leftarrow \text{sigm}(\mathbf{c} + W\mathbf{x}^0 + \mathbf{U}\vec{y^0})$

% Negative phase
$\mathbf{h}^0 \sim p(\mathbf{h}|y^0, \mathbf{x}^0)$, $y^1 \sim p(y|\mathbf{h}^0)$, $\mathbf{x}^1 \sim p(\mathbf{x}|\mathbf{h}^0)$
$\widehat{\mathbf{h}}^1 \leftarrow \text{sigm}(\mathbf{c} + W\mathbf{x}^1 + \mathbf{U}\vec{y^1})$

% Update
for $\theta \in \Theta$ **do**
$\theta \leftarrow \theta - \lambda \left(\frac{\partial}{\partial \theta} E(y^0, \mathbf{x}^0, \widehat{\mathbf{h}}^0) - \frac{\partial}{\partial \theta} E(y^1, \mathbf{x}^1, \widehat{\mathbf{h}}^1) \right)$
end for

شکل ۲: الگوریتم یادگیری ماشین بولتزمن با برچسب

پرسش شبیه‌سازی ۲. داده‌های MNIST را بارگذاری کرده و آن‌ها را به حالت باینری تبدیل کنید (مثلاً با اعمال یک آستانه روی شدت پیکسل‌ها)، سپس با آموزش یک RBM توزیع $p(y, x)$ را با استفاده از الگوریتم شکل ۲ تخمین بزنید. بعد از آموزش مدل، طبقه بندی مدل نهایی را با استفاده از $p(y|x)$ انجام داده و دقت آن را گزارش کنید.

اگر دقت ما فقط روی طبقه بندی باشد، میتوانیم بجای بهینه سازی $p(y, x)$ و تولید یک مدل مولد با یادگیری توزیع داده، فقط $p(y|x)$ را مورد توجه قرار بدهیم و آن را بهینه کنیم.

پرسش تئوری ۵. رابطه زیر را اثبات کنید.

$$\frac{\partial \log p(y|x)}{\partial \theta} = \sum_j \sigma(o_{yj}(x)) \cdot \frac{\partial o_{yj}(x)}{\partial \theta} - \sum_{y',j} \sigma(o_{y'j}(x)) \cdot \frac{\partial o_{y'j}(x)}{\partial \theta}$$

که در آن $o_{yj}(x) = c_j + \sum_k W_{jk} x_k + U_{jy}$ میباشد، در نتیجه میتوانیم محاسبه دقیق گرادیان انجام دهیم.

با توجه به اینکه $|y| = K$ که در آن K تعداد کلاس است، بهینه سازی یک DRBM بسیار ساده تر از بهینه سازی RBM های معمول است. همانطور که میتوان دید، DRBM ها شباهت بسیاری با شبکه های Feedforward دارند و مزیت اضافی وجود یک مدل بر پایه انرژی، در ذات RBM ای آنها به ما قابلیت های تولید داده جدید، به صورت کنترل شده می دهد.
حال میتوانیم توانایی های این مدل را برای طبقه بندی ارزیابی کنیم.

پرسش شبیه‌سازی ۳. داده‌های MNIST را بارگذاری کرده و آن‌ها را به حالت باینری تبدیل کنید (مثلاً با اعمال یک آستانه روی شدت پیکسل‌ها)، سپس با آموزش یک DRBM توزیع $p(y|x)$ را تخمین بزنید. بعد از آموزش مدل، دقت طبقه بندی مدل نهایی را گزارش کنید.

به صورت کلی، برای یادگیری یک مدل تمایز دهنده از $\mathcal{L}_{\text{disc}}$ و برای یادگیری یک مدل مولد از $\mathcal{L}_{\text{gen}}$ میتوان استفاده کرد.

$$\mathcal{L}_{\text{disc}}(\mathcal{D}) = -\sum_{i=1}^{|\mathcal{D}|} \log p(y_i|x_i), \quad \mathcal{L}_{\text{gen}}(\mathcal{D}) = -\sum_{i=1}^{|\mathcal{D}|} \log p(y_i, x_i)$$

یادگیری مدل با استفاده از $\mathcal{L}_{\text{disc}}$ ساده تر است، ولی از توانایی های یک مدل مولد دیگر استفاده نمیکنیم و فقط به طور مستقیم قدرت طبقه بندی را بهینه میکنیم. $\mathcal{L}_{\text{gen}}$ برای بهینه سازی یک مدل مولد مناسب است، ولی سخت بودن بهینه سازی آن، و لحاظ نشدن مستقیم قدرت طبقه بندی ممکن است به بهترین قدرت طبقه بندی منجر نشود. به صورت کلی پیدا شده است که برای مجموعه داده های کوچک تر، روش یادگیری یک توزیع مولد مناسب تر است، و برای مجموعه داده های بزرگ، روش یادگیری یک مدل تمایزگر مناسب تر است.
به صورت کلی میتوانیم یک روش هیبرید استفاده کنیم، که در عمل میتواند مفید باشد:

$$\mathcal{L}_{\text{hybrid}}(\mathcal{D}) = \mathcal{L}_{\text{disc}}(\mathcal{D}) + \alpha\mathcal{L}_{\text{gen}}(\mathcal{D}).$$

پرسش شبیه‌سازی ۴. باری دیگر، یادگیری را با روش هیبرید و $\alpha \in \{۰/۰۱, ۰/۱, ۱\}$ انجام دهید، و با استفاده از $p(y|x)$ طبقه بندی را انجام داده، و دقت نهایی را گزارش کنید.

پرسش تئوری ۶. نتیجه نهایی یادگیری و دقت طبقه بندی را برای تمامی مدل های قبلی گزارش کرده و مقایسه کنید، آیا نتایج آن مطابق انتظار شماست؟

در تمامی مراحل، با استفاده از یک مجموعه داده $\mathcal{D}_{\text{sup}} = \{x_i, y_i\}$، یک مدل تمایزگر طراحی کردیم که قابلیت مدل‌سازی مولد را بر اساس یک تابع انرژی دارد. با این حال، جمع‌آوری داده‌های برچسب‌دار در بسیاری از موارد هزینه‌بر است و معمولاً حجم این نوع داده‌ها محدود است. در مقابل، داده‌های بدون برچسب $\mathcal{D}_{\text{unsup}} = \{x_i\}$ به‌طور گسترده‌تر در دسترس هستند و می‌توان از آن‌ها برای بهبود یادگیری مدل بهره برد.

پرسش تئوری ۷. آیا میتوان در کنار داده های برچسب دار $\mathcal{D}_{\text{sup}}$ از داده های $\mathcal{D}_{\text{unsup}}$ نیز استفاده کرد؟
به صورت دقیق تا ما دنبال یک تابع هدف نهایی به صورت زیر هستیم.

$$\mathcal{L}_{\text{semi-sup}}(\mathcal{D}_{\text{sup}}, \mathcal{D}_{\text{unsup}}) = \mathcal{L}_{\text{sup}}(\mathcal{D}_{\text{sup}}) + \beta\mathcal{L}_{\text{unsup}}(\mathcal{D}_{\text{unsup}})$$

یک پیشنهاد مناسب برای $\mathcal{L}_{\text{unsup}}$ ارائه دهید، و جزئیات بهینه سازی آن را توضیح دهید.

پرسش شبیه‌سازی ۵. (امتیازی) باری دیگر، فرآیند یادگیری را به روش شبه‌نظارتی انجام دهید، به‌طوری که:

- مقدار n یکی از مقادیر $n \in \{۰/۰۱N, ۰/۱N, N\}$ باشد.
- مجموعه نظارتی $\mathcal{D}_{\text{sup}}$ شامل n داده اول از مجموعه MNIST در نظر گرفته شود.
- مجموعه غیرنظارتی $\mathcal{D}_{\text{unsup}}$ شامل $N - n$ داده باقی‌مانده باشد.

پس از یادگیری مدل، طبقه‌بندی را با استفاده از $p(y|x)$ انجام داده و دقت نهایی را گزارش کنید.

---

۳ نکات مهم

لطفاً به نکات زیر دقّت کنید:

۱. پروژه شامل دو فاز خواهد بود.

۲. پروژه را میتوانید به صورت انفرادی یا به شکل گروه های دو نفره انجام دهید. دقت کنید چه به شکل انفرادی و چه به صورت گروهی باید تمام بخش های پروژه را انجام دهید و انجام انفرادی آن امتیاز اضافه ای برای شما نخواهد داشت.

۳. دو فاز این پروژه در مجموع ۲ تا ۳ نمره از نمره درس را تشکیل می دهند.

۴. پس از پایان پروژه یک روز برای تحویل حضوری پروژه در نظر گرفته می شود و باید کد ها و خروجی های خود را در حضور دستیاران آموزشی ارائه دهید و به پرسش های دستیاران پاسخ دهید. دقت کنید که تمام اعضای گروه باید به تمام بخش های پروژه مسلط باشند. در نهایت برای تمام اعضای گروه یک نمره در نظر گرفته خواهد شد.

۵. برای فاز نخست پروژه میتوانید حداکثر ۳ روز تاخیر مجاز استفاده نمایید اما به دلیل وجود ددلاین ثبت نمرات ددلاین فاز دوم پروژه سخت خواهد بود و امکان استفاده از تاخیر برای آن وجود ندارد.

۶. فاز دوم شامل یک سوال امتیازی می باشد که حداکثر ۰.۲ نمره اضافی خواهد داشت.

۷. تمامی شبیه‌سازی‌ها باید با کمک زبان Python انجام شود. همچنین مجاز هستید از تمام کتابخانه هایی که در طول تمرین ها از آنها استفاده کرده اید مانند `numpy`، `scipy` و `pytorch` استفاده نمایید اما دقت کنید پیاده سازی الگوریتم ها باید توسط شما انجام شده باشد و نمیتوانید از کتابخانه هایی که الگوریتم را به صورت آماده پیاده سازی کرده اند استفاده نمایید.

۸. تحویل پروژه به صورت گزارش و کدهای نوشته‌شده است. گزارش باید شامل پاسخ پرسش‌ها، تصاویر و نمودارها و نتیجه‌گیری‌های لازم باشد. در نهایت یک فایل شامل کد ها و یک گزارش به فرمت pdf را در سامانه CW آپلود نمایید. آپلود کردن پروژه توسط یکی از اعضای گروه کافی میباشد.

۹. اگر برای پاسخ به پرسش‌ها، از منبعی (کتاب، مقاله، سایت و...) کمک گرفته‌اید، حتماً به آن ارجاع دهید.

۱۰. درصورت مشاهده‌ی تقلّب، نمره‌ی هردو فرد صفر منظور خواهد شد.

۱۱. مسئول پروژه آقای سلیمان بیگی میباشد و در صورت داشتن مشکلاتی در گروهبندی، زمان تحویل حضوری و ... به ایشان (@amirr62a) پیام دهید.

۱۲. در صورت داشتن پرسش در بخش ۱ به @amirrezazameni یا @Mahdi_h721 و در بحش ۲ به @BornaKhodabandeh پیام دهید.

موفّق باشید!