Machine Learning per la fisica applicata e la fisica delle alte energie

Lezione 24: Generative Adversarial Neural Networks (GANS)

Le Generative Adversarial Neural Networks (GANs) è una classe di metodi di machine learning in cui due reti neurali vengono addestrate in modo competitivo. Una rete neurale, chiamata GENERATOR partendo da un certo data set, viene addestrata a produrre dati nuovi che da questo data set sono indistinguibili. Per esempio, supponendo di avere una collezione di immagini di galassie, la rete neurale GENERATOR è in grado di produrre altre immagini (realistiche, MA non reali) di galassie. Una seconda rete neurale, chiamata DISCRIMINATORE viene allenata per discernere quali immagini sono state generate dalla rete neurale GENERATOR e quali no. Il GENERATOR viene quindi allenato per "ingannare" il DISCRIMINATOR. Il framework, che non è quindi basato su labelled data sets, è di tipo unsupervised learning. Le GANs sono state proposte da Ian Goodfellow e collaboratori nel 2014.

24.1 CENNI DI TEORIA DELL'INFORMAZIONE

Data una variabile stocastica X ed una sua realizzazione x la quantità d'informazione associata alla variabile dipende dalla sua distribuzione di probabilità $p(x)$. Vogliamo definire una quantità che chiami la quantità d'informazione realizzata dalla realizzazione di una variabile stocastica che sia additiva rispetto all'osservazione di due (o più) realizzazioni della variabile stocastica

$h(x,y) = h(x) + h(y)$ per eventi indipendenti $p(x,y) = p(x)p(y)$

Definisco $h(x) = -\ln p(x)$ {il segno "−" assicura che $0 \le p(x) \le 1$}

L'informazione totale contenuta in un set di istanze della variabile stocastica X è detta ENTROPIA

$$H(x) = \sum_x p(x) h(x) = -\sum_x p(x) \ln p(x) \simeq -\int dx\, p(x) \ln(p(x))$$

eventi poco probabili hanno una grande informazione, ma poco piccola distribuzioni lisce hanno grande entropia (massima per distribuzione uniforme); più disperso è la distribuzione, maggiore è il disordine

Nel ML, può essere utile determinare il grado di similarità di due distribuzioni di probabilità. Possiamo considerare, date due distribuzioni di probabilità $p(x)$ e $q(x)$ la quantità

$$KL(p(x) \| q(x)) = \int dx\, p(x) \ln p(x) - \int dx\, p(x) \ln q(x) = \int dx\, p(x) \ln \frac{p(x)}{q(x)}$$

Questa quantità è chiamata Kullback-Leiber divergence.

Esempio: si considerino due distribuzioni gaussiane

$$p(x) = \frac{1}{\sqrt{2\pi\sigma_p^2}} e^{-(x-x_p)^2/2\sigma_p^2} \qquad q(x) = \frac{1}{\sqrt{2\pi\sigma_q^2}} e^{-(x-x_q)^2/2\sigma_q^2}$$

$$D_{KL}(p(x) \| q(x)) = \ln \frac{\sigma_q}{\sigma_p} + \frac{1}{2\sigma_q^2}\left\{(x_q - x_p)^2 + (\sigma_p^2 - \sigma_q^2)\right\}$$

Le due distribuzioni sono tanto più simili quanto minore è la differenza tra i loro valori centrali e le loro varianze. Si ha

$$D_{KL}(p(x) \| q(x)) \ge 0 \quad \text{e} \quad D_{KL} = 0 \iff p(x) \equiv q(x)$$

In generale, l'integrale su dx è sostituito da un sampling discreto.

La metrica KL non è simmetrica. La versione simmetrica della KL divergence è la Jensen-Shannon (JS) divergence:

$$D_{JS}(p(x)\|q(x)) = \frac{1}{2}\left\{ D_{KL}\left(p\,\middle\|\,\frac{p+q}{2}\right) + D_{KL}\left(q\,\middle\|\,\frac{p+q}{2}\right)\right\}$$

Un'altra metrica particolarmente utilizzata è la statistica di Kolmogorov-Smirnov. Supponiamo di avere n istanze della variabile casuale X. Definisco

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{[-\infty,x]}(x_i) \qquad \begin{cases} I_{[-\infty,x]}(x_i) = 1 \text{ per } x_i \leq x \\ I_{[-\infty,x]}(x_i) = 0 \text{ per } x_i > x \end{cases}$$

$$KS = \sup_x \left[F_{1,n}(x) - F_{2,m}(x)\right] \qquad \begin{cases} F_{1,n} \text{ sample \#1 di } n \text{ elementi} \\ F_{2,m} \text{ sample \#2 di } m \text{ elementi} \end{cases}$$

24.2 MODELLI GENERATIVI

MODELLI DISCRIMINATIVI	MODELLI GENERATIVI
$p(C_k/\vec{x})$	$p(\vec{x}/C_k)$
probabilità che un'istanza $\vec{x}$ appartenga alla classe C_k	distribuzione di probabilità di x associate alla classe C_k
INFERENZA (dai dati reali)	PRODUZIONE (di dati di misure)

BENEFIT: accedere alla densità marginale $p(x)$ nello spazio dei dati

I modelli generativi possono essere meglio compresi nel contesto della classificazione. Supponiamo di avere un set di n istanze $\vec{x}$, con p features che vogliamo classificare in categorie C_k.

1 risolviamo il problema dell'inferenza statistica e determiniamo

le probabilità condizionate

$p(\bar{x}'/\mathcal{C}_k)$ probabilità che, data la classe $\mathcal{C}_k$, l'istanza $\bar{x}'$ vi appartenga

2 inferiamo le prior class probabilities $p(\mathcal{C}_k)$

3 determiniamo le posterior class probabilities con il teorema di Bayes

$$p(\mathcal{C}_k/\bar{x}') = \frac{p(\bar{x}'/\mathcal{C}_k)\,p(\mathcal{C}_k)}{p(\bar{x}')} \quad \text{probabilità che } \bar{x}' \in \mathcal{C}_k$$

4 determiniamo $p(\bar{x}') = \sum_k^1 p(\bar{x}'/\mathcal{C}_k)\,p(\mathcal{C}_k)$

cioè le probabilità di generare l'istanza $\bar{x}'$.

MODELLI DISCRIMINATIVI — MODELLI GENERATIVI

$$p(\mathcal{C}_k/\bar{x}') = \frac{p(\bar{x}'/\mathcal{C}_k)\,p(\mathcal{C}_k)}{p(\bar{x}')} = p(\bar{x}')\,p(\bar{x}') = \sum_k^1 p(\bar{x}'/\mathcal{C}_k)\,p(\mathcal{C}_k)$$

Conoscendo $p(\bar{x}')$ si possono generare dati di misteri come input.
Costruiamo esplicitamente un modello generativo nel contesto di un problema di classificazione a due categorie ($K=2$)

$$p(\mathcal{C}_1/\bar{x}') = \frac{p(\bar{x}'/\mathcal{C}_1)\,p(\mathcal{C}_1)}{p(\bar{x}'/\mathcal{C}_1)\,p(\mathcal{C}_1) + p(\bar{x}'/\mathcal{C}_2)\,p(\mathcal{C}_2)} = \frac{1}{1+\exp(-a)} = \sigma(a)$$

dove $\sigma(a)$ è la funzione sigmoide e $a = \ln \dfrac{p(\bar{x}'/\mathcal{C}_1)\,p(\mathcal{C}_1)}{p(\bar{x}'/\mathcal{C}_2)\,p(\mathcal{C}_2)}$

Per $K > 2$ si può generalizzare a

$$p(\mathcal{C}_k/\bar{x}') = \frac{\exp(a_k)}{\sum_j^1 \exp(a_j)} \qquad a_k = \ln\{p(\bar{x}'/\mathcal{C}_k)\,p(\mathcal{C}_k)\}$$

24.3 GENERATIVE ADVERSARIAL NEURAL NETWORKS

Supponiamo che una variabile stocastica abbia una distribuzione di probabilità $p_{\vec{d}}(\vec{x})$. Vogliamo generare dati di misteri distribuiti secondo una distribuzione di probabilità $p_{\vec{\vartheta}}(\vec{x})$. Cioè vogliamo costruire un modello tale per cui $p_{\vec{\vartheta}}(\vec{x})$ sia molto simile a $p_{\vec{d}}(\vec{x})$. Noi vogliamo minimizzare la quantità:

$$D_{KL}\left(p_{\vec{d}}(\vec{x}) \,\|\, p_{\vec{\vartheta}}(\vec{x})\right) = \int d\vec{x}\, p_{\vec{d}}(\vec{x}) \ln \frac{p_{\vec{d}}(\vec{x})}{p_{\vec{\vartheta}}(\vec{x})}$$

$$= \int dx\, p_{\vec{d}}(\vec{x}) \ln p_{\vec{d}}(\vec{x}) - \int dx\, p_{\vec{d}}(\vec{x}) \ln p_{\vec{\vartheta}}(\vec{x}) = S_p\left[p_{\vec{d}}\right] - \langle \ln p_{\vec{\vartheta}} \rangle_{\vec{d}}$$

dove
- $S_p[p_{\vec{d}}]$ entropia
- $\langle \ln p_{\vec{\vartheta}} \rangle_{\vec{d}}$ valore di aspettazione di $p_{\vec{\vartheta}}$, dati i dati

Ricavo:

$$\langle \ln p_{\vec{\vartheta}}(\vec{x}) \rangle_{\vec{d}} = -D_{KL}\left(p_{\vec{d}}(\vec{x}) \,\|\, p_{\vec{\vartheta}}(\vec{x})\right) + S_p\left[p_{\vec{d}}(\vec{x})\right]$$

log-likelihood dei dati, per un modello — KL divergence — entropia dei dati (indipendente dal modello)

Minimizzare D_{KL} è equivalente a massimizzare $\langle \ln p_{\vec{\vartheta}}(\vec{x}) \rangle_{\vec{d}}$.

Osservazione: che cosa minimizzare?

1) $D_{KL}\left(p_{\vec{d}}(\vec{x}) \,\|\, p_{\vec{\vartheta}}(\vec{x})\right)$ si può calcolare con un sampling

2) $D_{KL}\left(p_{\vec{\vartheta}}(\vec{x}) \,\|\, p_{\vec{d}}(\vec{x})\right)$ non si può calcolare $p_{\vec{d}}$ non noto

Nell'ADVERSARIAL LEARNING si minimizza 2) allenando il DISCRIMINATORE a distinguere tra dati reali e dati di misteri. Il modello viene sfavorito se genera dati di misteri che possono essere facilmente distinti dai dati reali, cioè il peso delle

regioni lontane dai dati è piccola.

Il generatore viene allenato in modo che la probabilità che il discriminatore commetta un errore sia alta, cioè il generatore è allenato in modo da generare dati di misure indistinguibili dai dati reali. Questo è un esempio di unsupervised learning

Il training di una GANs si può fare come quello di una NN, ma i parametri di G (generatore) e D (discriminatore) vanno aggiornati in modo sequenziale

1) Prendiamo una sample di N dati dal data set di training

$$\{\vec{x}_n\}_{n=1}^{N} \qquad \vec{x}_n = (x_{n,1}, x_{n,2}, \dots x_{n,p}) \quad p: \text{numero di features per sample}$$

2) Produciamo una sample di N dati di misure da un generatore $G_{\vec{\vartheta}}$ (allo step iniziale questa sample sarà rumore statistico).

$$\{\vec{z}_n\}_{n=1}^{N} \qquad \vec{z}_n = (z_{n,1}, z_{n,2}, \dots, z_{n,p}) \quad ; \quad G_{\vec{\vartheta}} : \mathcal{Z} \to \mathcal{X}$$

(latent space)

3) Risolvere in modo sequenziale

$$\min_{\vec{\vartheta}} \max_{\Phi} V(G_{\vartheta}, D_{\Phi}) = \mathbb{E}_{x \sim p_{\vec{d}}} (\ln D_{\vec{\Phi}}(\vec{x})) + \mathbb{E}_{z \sim p_{\vec{\vartheta}}} (\ln (1 - D_{\Phi}(G_{\vartheta}$$

Per un fissato generatore, il discriminatore risolve un problema di classificazione binaria col che efficienza ottimale per

$$D^*(\vec{x}) = \frac{p_{\vec{d}}(\vec{x})}{p_{\vec{\vartheta}}(\vec{x}) + p_{\vec{d}}(\vec{x})}$$

(assegna probabilità nulle ai dati di misure e probabilità unitarie ai dati reali)

In tal caso, si ha

$$V(G_{\vartheta}, D^*_{\vec{\Phi}}) = 2\, JS_D(p_{\vec{d}}, p_{\vec{\vartheta}}) - 2 \ln 2$$