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

Lezione 4: Ottimizzazione.

Abbiamo visto nelle lezioni precedenti che lo scopo del machine learning è la stima dei parametri $\vec{\vartheta}$ (e/o degli iperparametri Φ) di un modello, cioè la determinazione dei loro valori di best-fit e dei corrispondenti intervalli di credibilità. Ciò si realizza minimizzando una funzione scalare (la loss function) $\mathcal{L}: \Theta \to \mathbb{R}$ (con $\vec{\vartheta} \in \Theta$)

$$\boxed{\vartheta^* = \arg\min_{\vec{\vartheta}} \mathcal{L}(\vec{\vartheta})} \quad \text{o meglio} \quad \boxed{\hat{\vartheta} = \arg\min_{\vec{\vartheta}} \mathcal{L}(\vec{\vartheta})} \quad \text{per } \vartheta^* \sim \hat{\vartheta}.$$

Ammettiamo per semplicità che $\Theta \subseteq \mathbb{R}^D$, con D il numero delle variabili ottimizzate. L'ottimizzazione è dunque CONTINUA.

Osservazioni

1) Lo spazio dei parametri è tale che può esistere un solo minimo globale o più minimi globali, oltre ad una serie di minimi locali. Il punto di minimo è, computazionalmente, sempre definito in termini locali, cioè $\exists\, \delta > 0,\ \forall \vec{\vartheta} \in \Theta \,/\, \|\vec{\vartheta} - \hat{\vec{\vartheta}}\| < \delta,\ \mathcal{L}(\hat{\vec{\vartheta}}) \leq \mathcal{L}(\vec{\vartheta})$.
Un minimo locale può essere circondato da minimi simili (flat direction). Un algoritmo che converge ad un punto stazionario è detto GLOBALLY CONVERGENT.

2) Per loss functions che siano due volte differenziabili, possiamo considerare $g(\vec{\theta}) = \nabla \mathcal{L}(\vec{\vartheta})$ il vettore gradiente e $H(\vec{\vartheta}) = \nabla^2 \mathcal{L}(\vec{\vartheta})$ la matrice Hessiana. Per un punto $\hat{\vec{\vartheta}} \in \mathbb{R}^D$ e $\hat{g} = g(\vec{\vartheta})|_{\hat{\vec{\vartheta}}}$ e $\hat{H} = H(\vec{\vartheta})|_{\hat{\vec{\vartheta}}}$

- CONDIZIONE NECESSARIA: se $\hat{\vec{\vartheta}}$ è un minimo locale, allora $\hat{g} = \vec{0}$ e $\hat{H}$ è positiva semi-definita
- CONDIZIONE SUFFICIENTE: se $\hat{g} = \vec{0}$ e $\hat{H}$ è positiva definita, allora $\hat{\vartheta}$ è un minimo locale.

3) Talora potremmo avere dei vincoli nella parametrizzazione (CONSTRAINED PARAMETRISATION). I vincoli sono generalmente categorizzati in equazioni e disequazioni (e.g. $h_k(\vec{\vartheta})=0$ per $k \in \mathcal{E}$; $g_j(\vec{\vartheta}) \leq 0$ per $j \in \mathcal{I}$). Il set di vincoli è dunque

$$C = \{ \vec{\vartheta} : g_j(\vec{\vartheta}) \leq 0 : j \in \mathcal{I}, h_k(\vec{\vartheta}) = 0 ; k \in \mathcal{E} \} \subseteq \mathbb{R}^D$$

$$\hat{\vartheta} = \underset{\vec{\vartheta} \in C}{\operatorname{argmin}} \, \mathcal{L}(\vec{\vartheta})$$ (se $\vec{\vartheta} \in \mathbb{R}^D$ si parla di UNCONSTRAINED PARAM.)

4) Può essere utile determinare se una funzione di loss è convessa. Se $f : \mathbb{R}^n \to \mathbb{R}$ è doppio differenziabile, allora f è convessa se e solo se $H = \nabla^2 f(x)$ è positiva semi-definita $\forall x$; f si dice strettamente convessa se H è positiva definita.

5) Qualora la loss function non sia continua (in un punto) allora si usa definire i sottogradienti: $\vec{g} \in \mathbb{R}^n$ è un sottogradiente di f in $\vec{x}$, con $\vec{x} \in \mathrm{Dom}(f)$, se $\forall \vec{z} \in \mathrm{Dom}(f)$, $f(\vec{z}) \geq f(\vec{x}) + \vec{g}^T(\vec{z} - \vec{x})$.

4.1 METODI DEL PRIMO ORDINE

Questi metodi sono basati sulle derivate prime della loss function, cioè guardano quali sono le direzioni negative nello spazio dei parametri. Un punto di partenza DEVE essere specificato, ed è detto $\vec{\vartheta}_0$. Per ogni iterazione t, i parametri vengono aggiornati come

$$\vec{\vartheta}_{t+1} = \vec{\vartheta}_t + \eta_t \vec{d}_t$$

η_t : STEP SIZE (o LEARNING RATE) $\quad \vec{d}_t$: DESCENT DIRECTION

$$\vec{g}_t = \nabla_{\vec{\vartheta}} \mathcal{L}(\vec{\vartheta})|_{\vec{\vartheta}_t} \sim d_t$$ STEEPEST DESCENT

L'algoritmo viene iterato finché un punto stazionario viene raggiunto. Formalmente, richiediamo che esiste $\eta_{max} > 0 \,/\, \mathcal{L}(\vec{\vartheta} + \eta \vec{d}) < \mathcal{L}(\vec{\vartheta})$ $\forall \; 0 < \eta < \eta_{max}$. Il gradiente $\vec{g}_t$ punta nella direzione di massima variazione di $\mathcal{L}(\vec{\vartheta}_t)$. La scelta ovvia sembra quindi essere $\vec{g}_t = -\vec{d}_t$.

La step size identifica, in sequenza $\{\eta_t\}$ il LEARNING RATE SCHEDULE.

La maniera più semplice è di scegliere η_t costante. Tuttavia, se η_t è troppo grande, il metodo potrebbe non convergere, se è troppo piccolo la convergenza potrebbe essere troppo lenta.
In linea generale è meglio scegliere η_t in modo adattivo, in modo che comporti la massima riduzione della loss function lungo la direzione scelta

$$\eta_t = \underset{\eta > 0}{\arg\min}\, \Phi_t(\eta) = \underset{\eta > 0}{\arg\min}\, \mathcal{L}(\vec{\vartheta}_t + \eta \vec{d}_t)$$

Questo approccio è definito LINE SEARCH perché cerchiamo lungo la direzione identificata da $\vec{d}_t$.
Inoltre, vogliamo definire algoritmi che convergano rapidamente.
Nel caso in cui la loss function sia convessa, si può mostrare che la gradient descent converge con tasso lineare.

$$|\mathcal{L}(\vec{\vartheta}_{t+1}) - \mathcal{L}(\hat{\vartheta}^*)| \leq \mu |\mathcal{L}(\vec{\vartheta}_t) - \mathcal{L}(\hat{\vartheta}^*)| \quad \text{con } \mu \text{ il tasso di convergenza} \quad 0 < \mu < 1$$

Il metodo di gradient descent può essere molto inefficiente nel caso di direzioni piatte. Una tecnica, nota come metodo dei momenti o della palla pesante, consiste nello spostarsi velocemente lungo direzioni "buone" (in base alle iterazioni precedenti) e lentamente in direzioni ove il gradiente cambia rapidamente.

$$\vec{m}_t = \beta \vec{m}_{t-1} + \vec{g}_{t-1} \qquad e \qquad \vec{\vartheta}_t = \vec{\vartheta}_{t-1} - \eta_t \vec{m}_t$$

dove $\vec{m}_t$ è la "quantità di moto" e $0 < \beta < 1$ (tipicamente $\beta \sim 0{,}9$)

$$\vec{m}_t = \beta \vec{m}_{t-1} + \vec{g}_{t-1} = \beta^2 \vec{m}_{t-2} + \beta \vec{g}_{t-2} + \vec{g}_{t-1} = \dots = \sum_{\tau=0}^{t-1} \beta^\tau \vec{g}_{t-\tau-1}$$

se tutti i gradienti nel passato sono costanti, allora

$$\vec{m}_t = \vec{g} \sum_{\tau=0}^{t-1} \beta^\tau$$ e lo scaling factor è una serie geometrica

$$\sum_{\tau=0}^{t-1} \beta^\tau \xrightarrow{t \to \infty} \frac{1}{1-\beta}$$ se $\beta \sim 0{,}9$, moltiplichiamo il gradiente per 10

4.2 METODI DEL SECONDO ORDINE

Il classico metodo alle derivate seconde è il metodo di Newton.
Esso consiste nell'aggiornare i parametri come:

$$\vec{\vartheta}_{t+1} = \vec{\vartheta}_t - \eta_t \vec{H}_t^{-1} \vec{g}_t \quad \text{con} \quad H_t \overset{\Delta}{=} \nabla^2 \mathcal{L}(\vec{\vartheta})\big|_{\vec{\vartheta}_t} = H(\vec{\vartheta}_t) \quad \text{MATRICE HESSIANA}$$

Poiché l'inversa H_t^{-1} regolarizza la "skewness" locale della curvatura il metodo di Newton è vantaggioso.

1 Inizializzare $\vec{\vartheta}_0$
2 finché $t = 1, 2, \dots,$ si arriva a convergenza
  3 valutare $\vec{g}_t = \nabla \mathcal{L}(\vec{\vartheta}_t)$
  4 valutare $H_t = \nabla^2 \mathcal{L}(\vec{\vartheta}_t)$
  5 risolvere $H_t \vec{d}_t = -\vec{g}_t$ per $\vec{d}_t$
  6 usare il line search per determinare η_t lungo $\vec{d}_t$
  7 $\vec{\vartheta}_{t+1} = \vec{\vartheta}_t + \eta_t \vec{d}_t$

Il metodo segue dalla taylor espansion di $\mathcal{L}(\vec{\vartheta})$ attorno a $\vec{\vartheta}_t$

$$\mathcal{L}_{quad}(\vec{\vartheta}) = \mathcal{L}(\vec{\vartheta}_t) + \vec{g}_t^T(\vec{\vartheta} - \vec{\vartheta}_t) + \frac{1}{2}(\vec{\vartheta} - \vec{\vartheta}_t)^T H_t (\vec{\vartheta} - \vec{\vartheta}_t)$$

Il minimo di $\mathcal{L}_{quad}$ è in $\vec{\vartheta} = \vec{\vartheta}_t - H_t^{-1} \vec{g}_t$.

Esistono poi metodi "quasi-Newton" che generano un'approssimazione iterativa della matrice Hessiana. Uno di questi è il metodo BFGS (Broyden, Fletcher, Goldfarb, Shanno) che approssima la matrice Hessiana con $B_t \approx H_t$

$$B_{t+1} = B_t + \frac{\vec{y}_t \vec{y}_t^T}{\vec{y}_t^T \vec{s}_t} - \frac{(B_t \vec{s}_t)(B_t \vec{s}_t)^T}{\vec{s}_t^T B_t \vec{s}_t} \qquad \begin{cases} \vec{s}_t = \vec{\vartheta}_t - \vec{\vartheta}_{t-1} \\ \vec{y}_t = \vec{g}_t - \vec{g}_{t+1} \end{cases}$$

e l'inversa

$$H_t^{-1} \approx C_t \quad \text{con} \quad C_{t+1} = \left(\mathbb{1} - \frac{\vec{s}_t \vec{y}_t^T}{\vec{y}_t^T \vec{s}_t}\right) C_t \left(\mathbb{1} - \frac{\vec{y}_t \vec{s}_t^T}{\vec{y}_t^T \vec{s}_t}\right) + \frac{\vec{s}_t \vec{s}_t^T}{\vec{y}_t^T \vec{s}_t}$$

4.3. STOCHASTIC GRADIENT DESCENT (SDG)

Lo scopo del metodo è di minimizzare la media

$$\mathcal{L}(\vec{\theta}) = \mathbb{E}_{q(\vec{z})}\left[\mathcal{L}(\vec{\theta}, z)\right]$$

dove z è una variabile casuale usata in input. Ad ogni iterazione assumiamo di osservare $\mathcal{L}_t(\vec{\theta}) = \mathcal{L}(\vec{\theta}, \vec{z}_t)$ con $\vec{z}_t \sim q$ (una funzione) se la distribuzione $q(\vec{z})$ è indipendente dai parametri che stiamo ottimizzando, possiamo usare $\vec{g}_t = \nabla_{\vec{\theta}} \mathcal{L}_t(\vec{\theta}_t)$. L'algoritmo è:

$$\vec{\theta}_{t+1} = \vec{\theta}_t - \eta_t \nabla \mathcal{L}(\vec{\theta}_t, z_t) = \vec{\theta}_t - \eta_t \vec{g}_t$$

Ricordiamo che molte procedure di fit sono basate sull'empirical risk minimisation che implicano la minimizzazione di

$$\mathcal{L}(\vec{\theta}_t) = \frac{1}{N}\sum_{n=1}^{N} \ell(\vec{y}_n, f(\vec{x}_n; \vec{\theta}_t)) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}_n(\vec{\theta}_t)$$

Segue che

$$g_t = \frac{1}{N}\sum_{n=1}^{N} \nabla_\theta \mathcal{L}_n(\vec{\theta}_t) = \frac{1}{N}\sum_{n=1}^{N} \nabla_\theta \ell(\vec{y}_n, f(\vec{x}_n; \vec{\theta}_t)) \overset{\text{minibatch approximation}}{\approx} \frac{1}{|B_t|}\sum_{n \in B_t} \nabla_{\vec{\theta}} \mathcal{L}_n(\vec{\theta}_t)$$

dove $|B_t|$ sono un set di esempi casuali utilizzati all'iterazione t. Una maniera per scegliere η_t è di iniziare con un valore piccolo e di incrementarlo con t, valutando la performance su minibatches.

<u>Esempi</u>:

$\eta_t = \eta_i$ se $t_i \leq t \leq t_{i+1}$ — piece-wise constant

$\eta_t = \eta_0 e^{-\lambda t}$ — exponential decay

$\eta_t = \eta_0 (\beta t + 1)^{-\alpha}$ — polynomial decay

Può essere utile utilizzare una matrice di pre-condizionamento

$$\vec{\theta}_{t+1} = \vec{\theta}_t - \eta_t M_t^{-1} \vec{g}_t$$ dove M_t è positiva definita.

<u>Esempi di pre-conditioning</u>

- ADAGRAD (adaptive gradient): utile se molti elementi del gradiente sono nulli (p. esempio se ci sono features nell'input che sono rare)

$$\vartheta_{t+1,d} = \vartheta_{t,d} - \eta_t \frac{1}{\sqrt{s_{t,d}+\varepsilon}} g_{t,d} \; ; \quad d = 1, \dots, D \qquad s_{t,d} = \sum_{i=1}^{t} g_{i,d}^2$$

ε piccolo; evita la divisione per zero

$$\Delta \vec{\theta}_t = -\eta_t \frac{1}{\sqrt{s_t+\varepsilon}} \vec{g}_t \quad \leadsto \quad M_t = \mathrm{diag}\left(\vec{s}_t + \vec{\varepsilon}\right)^{1/2}$$

- ADADELTA: come ADAGRAD, ma evita che il denominatore diventi troppo grande con il tempo, cosa che può compromettere l'efficienza.

$$\Delta \vec{\vartheta}_t = -\eta_t \frac{\sqrt{\delta_{t-1}+\varepsilon}}{\sqrt{s_t+\varepsilon}} \vec{g}_t \quad \text{con} \quad \vec{\delta}_t = \beta \vec{\delta}_{t-1} + (1-\beta)(\Delta \vec{\vartheta}_t)^2$$

- ADAM: combina ADADELTA con i momenti.

$$\begin{cases} \vec{m}_t = \beta_1 \vec{m}_{t-1} + (1-\beta_1)\vec{g}_t \\ \vec{s}_t = \beta_2 \vec{s}_{t-1} + (1-\beta_2)\vec{g}_t^{\,2} \end{cases} \qquad \Delta \vec{\vartheta}_t = -\eta_t \frac{1}{\sqrt{s_t}+\varepsilon} \vec{m}_t$$

ADAM = adaptive moment estimation. $\beta_1 = 0{,}9$; $\beta_2 = 0{,}999$; $\varepsilon = 10^{-6}$.

4.4. CONSTRAINED MINIMISATION

Riconsideriamo il caso in cui

$$\hat{\vec{\vartheta}} = \underset{\vec{\vartheta} \in C}{\arg\min} \, \mathcal{L}(\vec{\vartheta}) \qquad \text{con} \quad C = \{ \vec{\vartheta} \in \mathbb{R}^D : h_i(\vec{\vartheta}) = 0, \, i \in \mathcal{E}, \, g_j(\vec{\vartheta}) \leq 0, \, j \in \mathcal{I} \}$$

$\mathcal{E}$ set di eguaglianze; $\mathcal{I}$ set di diseguaglianze

Assumiamo di avere solamente $h(\vec{\vartheta}) = 0$. Notiamo che, per ogni punto sulla superficie $h(\vec{\vartheta}) = 0$, $\nabla h(\vec{\vartheta})$ sarà ortogonale a $h(\vec{\vartheta}) = 0$. Questo è ovvio se si considera un punto $\vec{\vartheta} + \vec{\varepsilon}$ su $h(\vec{\vartheta})$. Se facciamo un'espansione di Taylor al prim'ordine attorno a $\vec{\vartheta}$, otteniamo

$$h(\vec{\vartheta} + \vec{\varepsilon}) \approx h(\vec{\vartheta}) + \varepsilon^T \nabla h(\vec{\vartheta})$$

Poiché sia $\vec{\vartheta}$ sia $\vec{\vartheta} + \vec{\varepsilon}$ sono su $h(\vec{\vartheta}) = 0$, segue che $h(\vec{\vartheta}) = h(\vec{\vartheta} + \vec{\varepsilon})$ e quindi $\vec{\varepsilon}^T \nabla h(\vec{\vartheta}) \approx 0$. Poiché $\vec{\varepsilon} \,/\!/\, h(\vec{\vartheta})$, $\Rightarrow \nabla h(\vec{\vartheta}) \perp h(\vec{\vartheta}) = 0$.

Cerchiamo un punto ϑ^* su $h(\vec{\vartheta})=0$ tale che $\mathcal{L}(\vec{\vartheta})$ è minimizzata.
Questo punto dev'essere tale che $\nabla h(\vec{\vartheta}) \perp h(\vec{\vartheta})=0$ e deve avere la proprietà per cui $\nabla \mathcal{L}(\vec{\vartheta})$ è perpendicolare a $h(\vec{\vartheta})=0$. Poiché sia $\nabla h(\vec{\vartheta})$ e $\nabla \mathcal{L}(\vec{\vartheta})$ sono $\perp$ a $h(\vec{\vartheta})=0$ in ϑ^*, devono essere paralleli. Cioè esiste λ^* costante tale che $(\lambda^* \in \mathbb{R})$:

$$\nabla \mathcal{L}(\vec{\vartheta}^*) = \lambda^* \nabla h(\vec{\vartheta}^*)$$

λ^* è chiamato moltiplicatore di Lagrange. Possiamo definire

$$L(\vec{\vartheta}, \lambda) \triangleq \mathcal{L}(\vec{\vartheta}) + \lambda h(\vec{\vartheta})$$

Un punto stazionario è tale che

$$\nabla_{\vec{\vartheta}, \lambda} L(\vec{\vartheta}, \lambda) = 0 \iff \lambda \nabla_{\vec{\vartheta}} h(\vec{\vartheta}) = \nabla \mathcal{L}(\vec{\vartheta}), \ h(\vec{\vartheta}) = 0.$$