Transformers e la tutela della proprietà dei dati: Un'analisi matematica e applicativa delle Deep Neural Networks

Tesi di Laurea Magistrale in Matematica

Candidato: Bernardo Valente
Relatore : Prof. Vittorio Colao
Correlatori : Ph.D. Angelica Liguori, Ph.D. Ettore Ritacco

23/04/2024

Università della Calabria
Dipartimento di Matematica e Informatica

Organizzazione della tesi

Introduzione

- **Generative AI**
  - La Generative AI è un campo dell'Intelligenza Artificiale che si occupa della creazione di Modelli capaci di generare dati (testo, immagini, audio, video, ...) che siano:
    - **Nuovi**: i dati generati non devono essere una copia di dati esistenti;
    - **Coerenti**: i dati generati devono esibire le caratteristiche del dominio di riferimento
- **Problema:**
  - Gli attuali modelli generativi sono davvero capaci di generare contenuti del tutto **nuovi**?

Introduzione

Perchè è importante risolvere il problema?

- Oggi giorno il problema della violazione dei diritti di copyright è al centro di numerose controversie tra diverse società, l'esempio più recente è la disputa sorta tra il NYT e OpenAI. Pertanto, riconoscere una violazione di dati sensibili diventa una sfida importante da affrontare.

Output from GPT-4:

exempted it from regulations, subsidized its operations and promoted its practices, records and interviews showed.

Their actions turned one of the best-known symbols of New York — its yellow cabs — into a financial trap for thousands of immigrant drivers. More than 950 have filed for bankruptcy, according to a Times analysis of court records, and many more struggle to stay afloat.

"Nobody wanted to upset the industry," said David Klahr, who from 2007 to 2016 held several management posts at the Taxi and Limousine Commission, the city agency that oversees medallions. "Nobody wanted to kill the golden goose."

New York City in particular failed the taxi industry, The Times found. Two former mayors, Rudolph W. Giuliani and Michael R. Bloomberg, placed political allies inside the Taxi and Limousine Commission and directed it to sell medallions to help them balance budgets and fund key initiatives.

During that period, much like in the mortgage lending crisis, a group of industry leaders enriched themselves by artificially inflating medallion prices. They encouraged medallion buyers to borrow as much as possible and ensnared them in interest-only loans and other one-sided deals that often required borrowers to pay hefty fees, forfeit their legal rights and give up most of their monthly incomes.

When the market collapsed, the government largely abandoned the drivers who bore the brunt of the crisis. Officials did not bail out borrowers or persuade banks to soften loan

Actual text from NYTimes:

exempted it from regulations, subsidized its operations and promoted its practices, records and interviews showed.

Their actions turned one of the best-known symbols of New York — its signature yellow cabs — into a financial trap for thousands of immigrant drivers. More than 950 have filed for bankruptcy, according to a Times analysis of court records, and many more struggle to stay afloat.

"Nobody wanted to upset the industry," said David Klahr, who from 2007 to 2016 held several management posts at the Taxi and Limousine Commission, the city agency that oversees cabs. "Nobody wanted to kill the golden goose."

New York City in particular failed the taxi industry, The Times found. Two former mayors, Rudolph W. Giuliani and Michael R. Bloomberg, placed political allies inside the Taxi and Limousine Commission and directed it to sell medallions to help them balance budgets and fund priorities. Mayor Bill de Blasio continued the policies.

Under Mr. Bloomberg and Mr. de Blasio, the city made more than $855 million by selling taxi medallions and collecting taxes on private sales, according to the city.

But during that period, much like in the mortgage lending crisis, a group of industry leaders enriched themselves by artificially inflating medallion prices. They encouraged medallion buyers to borrow as much as possible and ensnared them in interest-only loans and other one-sided deals that often required them to pay hefty fees, forfeit their legal rights and give up most of their monthly incomes.

(a) Articolo generato da GPT vs Articolo

(b) Immagine generata da AI

Modelli Generativi

Modelli Generativi: I modelli generativi cercano di apprendere la distribuzione di probabilità reale dei dati del training set al fine di generare nuovi dati. Le applicazioni dei modelli generativi sono svariate:

- Creazione immagini
- Generazione di testi
- Sintesi Vocale
- Games & Simulation

Modelli Generativi

In un task di generazione di dati consideriamo tre distribuzioni di probabilità:

- p_T la distribuzione reale dei dati del Dataset(che nella pratica è sconosciuta)
- p_D la distribuzione empirica dei dati ottenuta dall'analisi empirica effettuata sul fenomeno che stiamo studiando
- p_M la distribuzione di probabilità che proponiamo per approssimare p_T

Obiettivo di un modello generativo:

- Riuscire a generare dati campionando secondo la distribuzione p_M che sono simili ai dati generati da p_T.

$$\text{divergence}(p_M, p_T) \to 0 \tag{1}$$

Modelli Generativi

- **Assunzioni**: Dal momento che p_T è sconosciuta nella pratica, sotto alcune condizioni possiamo assumere la distribuzione di probabilità empirica p_D come distribuzione di probabilità che descrive la generazione dei dati nel Dataset.
- **Obiettivo**:

$$\mathrm{divergence}(\mathrm{p_M}, \mathrm{p_D}) \to 0 \tag{2}$$

Sotto queste condizioni scegliamo come modello p_M una famiglia di distribuzioni parametriche $p(x|\theta) := p_\theta(x)$. Il problema 2 si traduce nel stimare quei parametri che rendono p_θ una 'buona' approssimazione per p_D:

$$\underset{\theta \in \Theta}{\mathrm{argmin}}\, \mathrm{divergence}(p_\theta, p_D) \tag{3}$$

Modelli Generativi

Problema: Come avviene la generazione dei dati?

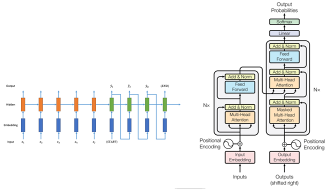
Data una sequenza di elementi $x_1, x_2 \dots x_n \in \mathcal{X}$ un modello generativo p_θ massimizza la *likelihood*:

$$\mathbb{L}(\theta|x) = \sum_i p_\theta(x_i|x_{i-k}, \dots x_{i-1}) \tag{4}$$

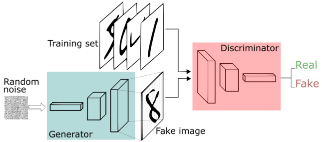
dove k è la dimensione della finestra di contesto. L'obiettivo è dunque risolvere il seguente problema di massimizzazione:

$$\hat{\theta} = \underset{\theta \in \Theta}{\operatorname{argmax}} \, \mathbb{L}(\theta|x) \tag{5}$$

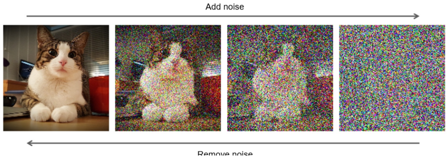
Modelli Generativi



(a) Modello Transformer



(b) GAN



(c) Diffusion Models

Metodologia

L'obiettivo principale dell'architettura che proponiamo, è quello di individuare eventuali violazioni della proprietà dei dati.

Architettura Modello-Generatore:

- Generatore (G): Il cui scopo è indurre un modello M (i.e un transformer) a replicare dati di un certo set di riferimento
- Modello (M): Riceve in input un prompt prodotto dal generatore e genera un output.

Training set:

- Training set M è il set di addestramento del modello
- Training set di oggetti coperti da copyright

Metodologia

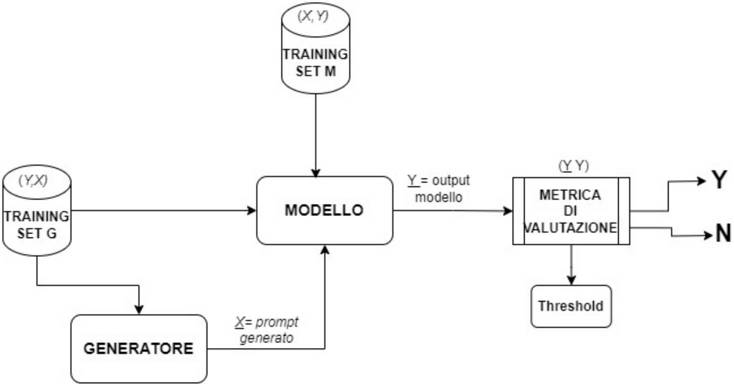
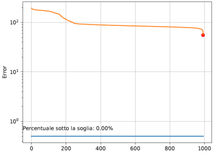
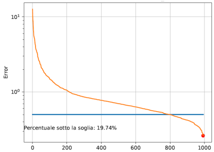


Figura 3: Architettura

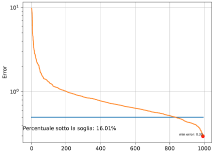
Risultati ottenuti



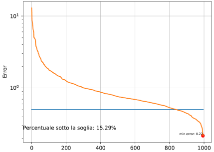
Errore di ricostruzione modello basic



Errore di ricostruzione Modello complesso



Errore di ricostruzione del best model con *overlapping alto*



Errore di ricostruzione del best model con *overlapping* basso

Transformers

Descrizione matematica dei transformers

Architettura Transformer

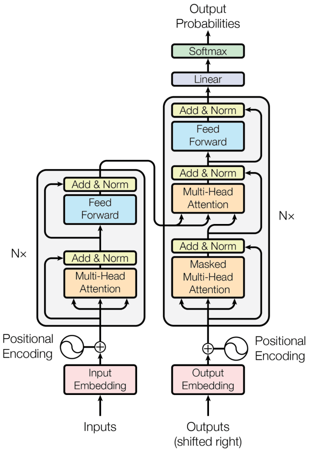


Figura 4: Architettura transformer

Transformers come flow maps

Idea: Neural ordinal differential equations

$$\begin{cases} \dot{x}(t) = w(t)\sigma(a(t)x(t) + b(t)), \quad t \in (0, T), \\ x(0) = x. \end{cases} \tag{6}$$

Che si sono dimostrati un metodo efficace per lo studio di reti neurali residuali con L hidden layers. Sia $x \in \mathbb{R}^d$, una residual network può essere vista come:

$$\begin{cases} x(k) = x(k-1) + f_\theta(x(k-1)) \quad \text{con } k = 1, \dots, L-1 \\ x(0) = x \end{cases} \tag{7}$$

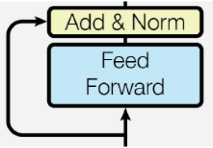
$f_\theta(x(k)) = w(k)\sigma(a(k)x(s)+b(k))$ Lipschitziana e $\theta(\cdot) = (w(\cdot), a(\cdot), b(\cdot)) \in \mathbb{R}^{d\times d} \times \mathbb{R}^{d\times d} \times \mathbb{R}^d$ sono i parametri del modello.

Transformers come flow maps

I Transformers processano *sequenze di input*

- $x_i(0)$ è chiamato *token* o *particella*
- $(x_i(0))_{i=1...n}$ è chiamato *prompt*

Le implementazioni pratiche del transformer utilizzano un layer di normalizzazione,



che costringono la dinamica delle particelle ad evolversi su un elissoide allineato con gli assi che variano nel tempo. Assumeremo che $x_i(t) \in \mathbb{S}^{d-1}$ e dunque il *prompt* $(x_i(t))_{i=1...n} \in (\mathbb{S}^{d-1})^n$

Transformers come flow maps

Un transformer può essere visto, dunque, come una mappa di flusso (flow map): dato un prompt come input $(x_i(0))_{i=1,\ldots,n} \in (\mathbb{S}^{d-1})^n$, questo evolve nel tempo secondo la dinamica

$$\dot{x}_i(t) = \mathbb{P}_{x_i}(t) \left(\frac{1}{Z_{\beta,i}(t)} \sum_{j=1}^{n} e^{\beta \langle Q(t)x_i(t), K(t)x_j(t) \rangle} V(t)x_j(t) \right) \tag{8}$$

per ogni $i \in \{1, \ldots, n\}$ e $t \geq 0$. Con $\mathbb{P}_{x_i}(t)$ ci riferiamo alla mappa di proiezione sullo spazio tangente alla sfera unitaria $d - dimensionale$ in x, $\mathbb{P}_x : S^{d-1} \to T_x S^{d-1}$ cosi definita:

$$\mathbb{P}_x y = y - \langle x, y \rangle x \tag{9}$$

$Q(\cdot), K(\cdot), V(\cdot)$ sono le matrici parametriche del modello, mentre $\beta > 0$ rappresenta un parametro intrinseco al modello.

Transformers come flow maps

Self-Attention:

- Il meccanismo rivoluzionario introdotto nei transformers è la *self attention*. La self-attention permette di catturare le relazioni tra i tokens all'interno di una stessa sequenza.

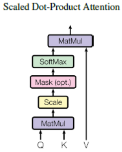


Figura 5: Self-Attention

$$A_{ij}(t) := \frac{e^{\beta \langle Q(t)x_i(t), K(t)x_j(t)\rangle}}{Z_{\beta,i}(t)}, \quad (i,j) \in \{1,\dots,n\} \times \{1,\dots,n\}$$

Transformers come flow map

Scenario semplificato: $Q = K = V = I_d$

$$\dot{x}_i(t) = P_{x_i}(t)\left(\frac{1}{Z_{\beta,i}(t)}\sum_{j=1}^{n} e^{\beta\langle x_i(t),x_j(t)\rangle}x_j(t)\right), \tag{10}$$

con $Z_{\beta,i}(t) = \sum_{k=1}^{n} e^{\beta\langle x_i,x_k\rangle}$. La dinamica della self attention (10) presenta, inoltre, delle caratteristiche simili al modello di Kuramoto che descrive la sincronizzazione degli oscillatori [5] e ancora di più al modello di Krause [4]:

$$\dot{x}(t) = \sum_{j=1}^{n} a_{ij}(x_j(t) - x_i(t)), \qquad a_{ij} = \frac{\phi(||x_i - x_j||^2)}{\sum_{k=1}^{n}\phi(||x_i - x_k||^2)}. \tag{11}$$

Quando $\phi(\cdot)$ è a supporto compatto, è stato mostrato [3] che nel modello di Krause le particelle tendono a formare diversi clusters per $t \to \infty$.

Transformers come flow maps

Multi-head-attention:

- Invece di proiettare *query,key,values* in uno spazio d_{model} dimensionale, verranno proiettate h volte tramite matrici di parametri, negli spazi di dimensione d_k, d_k, d_v rispettivamente. Per ognuna di queste versioni proiettate si applica la self-attention. Questo permette di ricavare informazioni da diversi sottospazi di rappresentazione.

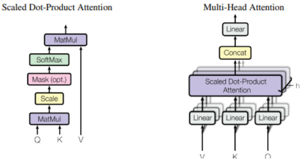


Figura 6: Self attention VS Multi-head attention

Transformers come flow maps

Dinamica transformers con Multi-Head-Attention:

$$\dot{x}_i(t) = P_{x_i(t)} \left(\sum_{h=1}^{H} \sum_{j=1}^{n} \frac{e^{\beta \langle Q_h x_i(t), K_h x_j(t) \rangle}}{Z_{\beta,i,h}(t)} V_h x_j(t) \right) \tag{12}$$

con $Z_{\beta,i,h}(t)$ definita con le matrici Q_h, K_h. Qui $H \geq 1$ è un intero. L'analisi del meccanismo della multi-head-attention ci porta a formulare un problema di approssimazione ancora aperto.

Definition 1

Una funzione $f : (\mathbb{S}^{d-1})^n \to (\mathbb{S}^{d-1})^n$ è detta permutation-equivariant se $f(\pi X) = \pi(f_1(X), f_2(X), \ldots, f_n(X))$ per ogni $X \in (\mathbb{R}^d)^n$ e per ogni $\pi \in S_n$.

Transformers come flow maps

Open Question 1

Sia $d, n > 2$ e consideriamo un transformer con un solo hidden layer con la multi-head self attention $f_\theta^H : (\mathbb{S}^{d-1})^n \to (\mathbb{S}^{d-1})^n$ definita come:

$$f_\theta^H(x_1, \ldots x_n)_i = P_{x_i}\left(\sum_{h=1}^{H}\sum_{j=1}^{n}\frac{e^{\beta<Q_h x_i, K_h x_j>}}{Z_{\beta,i,h}} V_h x_j\right)$$

dove $H \geq 1$ e $\theta = (Q_h, K_h, V_h)$ come in (12). Si può approssimare, in qualche topologia appropriata, una qualsiasi funzione permutation-equivariant $f : (\mathbb{S}^{d-1})^n \to (\mathbb{S}^{d-1})^n$ attraverso qualche funzione f_θ^H per $H \to +\infty$?. Lo stesso problema, ma senza il layer di normalizzazione rimane aperto: $f_\theta^H : (\mathbb{R}^d)^n \to (\mathbb{R}^d)^n$

$$f_\theta^H(x_1, \ldots x_n)_i = \left(\sum_{h=1}^{H}\sum_{j=1}^{n}\frac{e^{\beta<Q_h x_i, K_h x_j>}}{Z_{\beta,i,h}} V_h x_j\right)$$

Transformers come flow maps tra misure

Problema:

- Non essendoci ricorrenze, abbiamo bisogno di un meccanismo che codifica la posizione relativa o assoluta di un token in una sequenza.

Soluzione proposta:

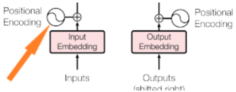


Figura 7

- Positional Encoding. Sia $(w_i)_{i=1\ldots n} \in (\mathbb{R}^d)^n$, il positional encoding del vettore w_i è definito come il vettore:

$$(p_i)_{2k} = \operatorname{sen}\left(\frac{i}{M^{2k/d}}\right), \quad (p_i)_{2k+1} = \cos\left(\frac{i}{M^{2k/d}}\right),$$

con $k \in \{1, \ldots, \frac{d}{2} - 1\}$ e $M > 0$. Infine l'$i-esimo$ token è rappresentato da $x_i(0) = p_i + w_i$.

Transformers come flow maps tra misure

Osservazione: L'output di un transformer è una misura di probabilità. Quindi possiamo rappresentare ogni sequenza di input $\{x_1(0), \ldots, x_n(0)\}$ tramite la misura empirica dei tokens costituenti la sequenza di input: $\frac{1}{n}\sum_{i=1}^{n}\delta_{x_i(0)}$.

Cambio prospettiva: Consideriamo ora i transformers come delle mappe di flusso tra spazi di probabilità sulla sfera $\mathcal{P}(\mathbb{S}^{d-1})$. Dunque possiamo riscrivere la dinamica (10) come segue:

$$\dot{x}_i(t) = \mathcal{X}[\mu(t)](x_i(t)) \tag{13}$$

per ogni $i \in \{1, \ldots, n\}$, e per $t \geq 0$. dove:

$$\mu(t, \cdot) = \frac{1}{n}\sum_{i=1}^{n}\delta_{x_i(t)}(\cdot) \tag{14}$$

è la misura empirica che codifica una sequenza.

Transformers come flow maps

Il campo vettoriale è $\mathcal{X}[\mu] : \mathbb{S}^{d-1} \to T\mathbb{S}^{d-1}$ è definito da:

$$\mathcal{X}[\mu](x) = P_x \left(\frac{1}{Z_{\beta,\mu}(x)} \int e^{\beta\langle x,y\rangle} y \, d\mu(y) \right) \tag{15}$$

con la funzione di partizione:$Z_{\beta,\mu}(x) = \int e^{\beta\langle x,y\rangle} \, d\mu(y)$. L'evoluzione della misura empirica $\mu(t)$ è guidata dall'*equazione di continuità*:

$$\begin{cases} \partial_t \mu + \mathrm{div}(\mathcal{X}[\mu]\mu) = 0 & \text{su } \mathbb{R}_{\leq 0} \times \mathbb{S}^{d-1} \\ \mu_{|t=0} = \mu(0) & su \ \mathbb{S}^{d-1} \end{cases} \tag{16}$$

soddisfatta nel senso di distribuzioni.

Funzionale di interazione di energia

Problema: Esistono delle quantità che sono monotone quando valutate lungo il flusso di (16)?.

Soluzione:

- Funzionale Interazione Energia

$$E_\beta[\mu] = \frac{1}{2\beta} \int \int e^{\beta \langle x, x' \rangle} \, d\mu(x) \, d\mu(x') \tag{17}$$

Infatti,

$$\frac{d}{dt} E_\beta[\mu(t)] = \int \|\mathcal{X}[\mu(t)](x)\|^2 Z_{\beta,\mu(t)}(x) \, d\mu(t,x) \tag{18}$$

per ogni $t \geq 0$ utilizzando l'integrazione per parti. Richiamando la definizione di $Z_{\beta,\mu}(x)$, vediamo che $e^{-\beta} \leq Z_{\beta,\mu}(x) \leq e^{\beta}$ per ogni $x \in \mathbb{S}^{d-1}$. L'identità (18), quindi, indica che E_β cresce lungo traiettorie di (16), analogamente, se $V = -I_d$ l'energia E_β diminuirebbe lungo le traiettorie.

Funzionale di interazione di energia

Questo solleva la questione di caratterizzare i minimi e i massimi globali di E_β, che è l'obiettivo del risultato seguente.

Proposizione 4.1

Sia $\beta > 0$ e $d \geq 2$. L'unico minimizzatore globale di E_β su $\mathcal{P}(\mathbb{S}^{d-1})$ è la misura uniforme σ_d. Ogni massimizzatore globale E_β su $\mathcal{P}(\mathbb{S}^{d-1})$ è una massa di Dirac δ_{x^*} centrata in qualche punto $x^* \in \mathbb{S}^{d-1}$.

Approssimazione del flusso del gradiente di Wasserstein

Problema:

- Si può vedere l'equazione di continuità (16) come il flusso del gradiente rispetto alla metrica di Wasserstein, del funzionale E_β o di qualsiasi altro funzionale.? Affinchè (16) sia il flusso del gradiente di E_β rispetto alla metrica di *Wasserstein*, il campo vettoriale $\mathcal{X}[\mu]$ definito come in (15), deve essere il gradiente della variazione prima δE_β di E_β.

Approssimazione del flusso del gradiente di Wasserstein

Soluzione Proposta:

- Modello surrogato di self-attention che chiameremo USA:

$$\dot{x}_i(t) = P_{x_i(t)}\left(\frac{1}{n}\sum_{j=1}^{n} e^{\beta\langle x_i(t),x_j(t)\rangle} x_j(t)\right) \tag{19}$$

dove sostituiamo la funzione di partizione $Z_{\beta,i}(t)$ con n. L'equazione di continuità derivante da questa dinamica è la seguente

$$\begin{cases} \partial_t \mu(t,x) + \mathrm{div}\left(P_x\left(\int e^{\beta\langle x,x'\rangle} x' \, d\mu(t,x')\right)\mu(t,x)\right) = 0 \\ \mu_{|t=0} = \mu_0 \end{cases} \tag{20}$$

con $(t,x) \in \mathbb{R}_{\geq 0} \times \mathbb{S}^{d-1}$. Quest'ultima può essere vista ora come flusso del gradiente di Wasserstein del funzionale interazione di energia. Otteniamo allora la seguente proposizione:

Approssimazione del flusso del Gradiente di Wasserstein

Proposizione 4.2

Consideriamo il funzionale di energia $E_\beta : \mathcal{P}(S^{d-1}) \to \mathbb{R}_{\geq 0}$ *definito come in* (17) *il campo vettoriale:*

$$\mathcal{X}[\mu](x) = P_x \left(\int e^{\beta \langle x, x' \rangle} x' \, dx' \right)$$

soddisfa

$$\mathcal{X}[u](x) = \nabla \delta E_\beta[\mu](x) \tag{21}$$

per ogni misura $\mu \in \mathcal{P}(S^{d-1})$*, e* $x \in \mathbb{S}^{d-1}$ *dove* δE_β *è la variazione prima del funzionale* E_β*.*

Transformers

Open Questions

Clustering e Caso Repulsivo

- **Clustering**:

Open Question 2

Può la dinamica delle particelle entrare in uno stato metastabile a lungo periodo, nel senso che per $\beta \gg 1$ tutte le particelle si raggruppano in $m < n$ clusters per un lungo periodo, prima di collassare in un singolo cluster $\{x^*\}$?

- **Caso repulsivo**:

Open Question 3

Minimizzazione E_β sullo spazio delle misure empiriche $\mathcal{P}_n(\mathbb{S}^{d-1})$ nel caso $V = -I_d$

Clustering e Caso Repulsivo

- Matrici Generali:

Open Question 4

Possiamo determinare la geometria limite del fenomeno di clustering nel caso di matrici generali?

Contributi

Contributi della tesi

- Implementazione transformer.
- Sperimentazione su come la complessità dell'architettura generatore-modello influenzasse sulla capacità di replicare dati.
- Sperimentazione sulla sovrapposizione dei dati tra il training set del generatore e del modello.
- Descrizione matematica dei transformers e presentazione di alcuni problemi aperti.

GRAZIE PER L'ATTENZIONE

Appendice

Modelli Ricorrenti

Modelli Ricorrenti

Una sequenza di dati è una successioni di osservazioni legate da una dipendenza temporale o da un principio causa effetto: $X = (x_1, x_2, \ldots x_t)$. Le reti neurali classiche trattano i dati di input come indipendenti tra loro, per questo motivo non riescono ad apprendere compiti in cui c'è bisogno di tener traccia del flusso di informazioni passate. Alcuni esempi di questi compiti sono:

- Stock prediction
- Textual Generation
- Machine Translation

Modelli ricorrenti

Le reti neurali ricorrenti sono delle reti neurali utilizzate per l'elaborazione di sequenze di dati. Sia $X = (x_1, x_2 \dots x_t),\ x_t \in \mathbb{R}^d$ una sequenza di dati un modello ricorrente fornisce l'output al tempo t in funzione del dato di input al tempo t e dell'hidden output al tempo $t-1$:

$$O_t = f_\Theta(x_t, h_{t-1}) \tag{22}$$

Modelli ricorrenti

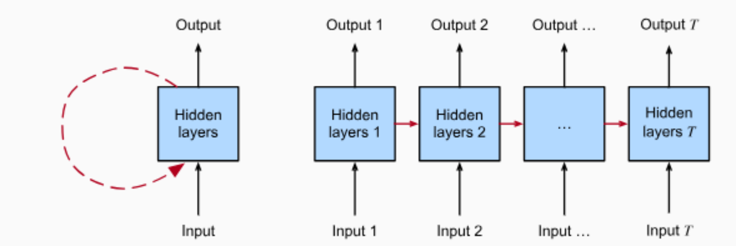


Figura 8: Reti neurali ricorrenti

Modelli ricorrenti

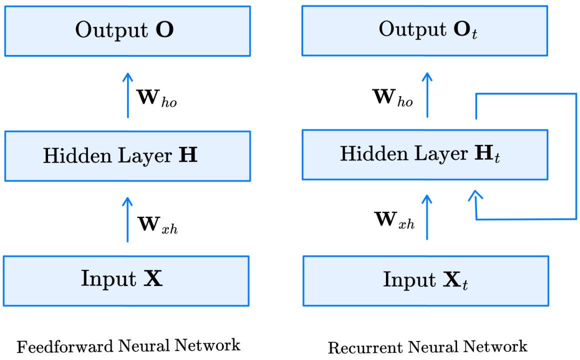


Figura 9: Differenza tra RRN e FFN

Modelli Ricorrenti

Tuttavia questi modelli presentano una serie di problemi. Nel momento in cui la sequenza di dati da processare è molto lunga, non riescono a catturare bene le dipendenze tra i dati della sequenza. Questo problema è noto anche come *Long-Term dependency problem*. Il problema di inserire le ricorrenze è l'annullamento del gradiente (Vanishing Gradient)calcolato durante la backpropagation(*BBT*) che non permette la propagazione degli errori e quindi l'apprendimento.

Modelli ricorrenti

Una possibile soluzione al vanishing gradient è cambiare le funzioni di attivazione dei neuroni:

- (Funzione lineare a pezzi) **ReLU** : $\mathbb{R} \to \mathbb{R}$, **ReLu**$(x) = \max\{0, x\}$.
- ReLU parametrica($\alpha << 1$)

$$f(x) = \begin{cases} x & se\ x > 0 \\ \alpha x & se\ x \leq 0 \end{cases} \tag{23}$$

- *Softplus.* La funzione di *softplus* è cosi definita:

$$f(x) = log(1 + e^{-x}) \tag{24}$$

- Funzione lineare $f(x) = x$

Modelli ricorrenti

In ogni caso non c'è controllo del flusso di informazioni nel tempo. Dunque per risolvere il problema della Long-Term dependency vengono proposte delle architetture ricorrenti *LST*(Long-Short Term Memory) o le *GRU*(Gated Recurrent Unit)

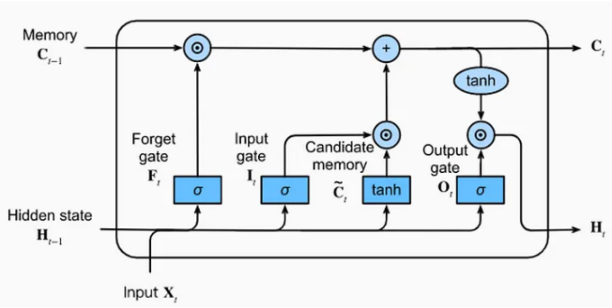


Figura 10: Architettura LSTM

Modelli Ricorrenti

- (Forget Gate) Decidere quanta informazione selezionare dalla cella di memoria.

$$f_t = \sigma(\theta_{fx}x_t + \theta_{hf}h_{t-1} + b_f) \tag{25}$$

- (Input Gate) Inserimento della nuova informazione processata nella cella di memoria.

$$i_t = \sigma(\theta_{ix}x_t + \theta_{hi}h_{t-1} + b_i) \tag{26}$$

Modelli Ricorrenti

- (Creazione memoria corrente) Viene creata una memoria corrente per aggiornare la memoria a lungo termine

$$\tilde{C}_t = tanh(\theta_{cx}x_t + \theta_{xh}h_{t-1}) \tag{27}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \tag{28}$$

- (Output Gate) Il compito è la creazione dell'hidden state

$$o_t = \sigma(\theta_{ox}x_t + \theta_{oh}h_{t-1} + b_0) \tag{29}$$

$$h_t = o_t \circ tanh(C_t) \tag{30}$$

Modelli Ricorrenti

Le Gated Recurrent Units (GRUs) sono delle varianti delle LSTM. In questo tipo di reti neurali ricorrenti i tre gate di un'unit'a LSTM sono sostituiti da due gate: *reset gate* e *update gate*. Il primo ha lo scopo di stabilire quanta informazione dello stato precedente vorremmo ricordare, mentre il secondo gate ha lo scopo di controllare quanta informazione dello stato attuale è una copia dello stato precedente

Modelli Ricorrenti

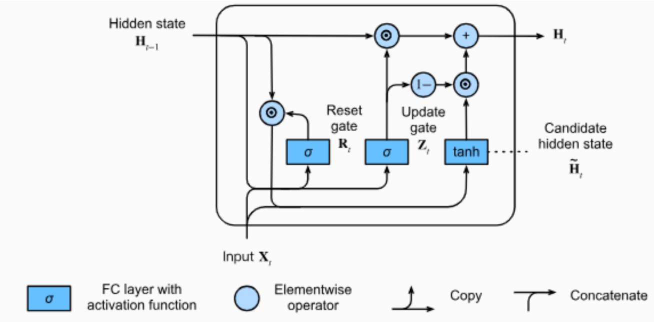


Figura 11: Calcolo dell'hidden state in una GRU

Appendice

Architettura Encoder-Decoder

Architettura Encoder-Decoder

I modelli Seq2Seq processano delle sequenze di dati per produrre nuove sequenze. L'archittetura encoder-decoder è un tipo di architettura neurale che viene utilizzata per implementare modelli Sequence2Sequence. L'archittetura è formata principalmente da due parti : un encoder che processa una sequenza di input producendo un insieme di vettori detti *context vectors* che vengono poi utilizzati dal decoder per produrre degli output.Il context vector riassume in un certo modo tutte le proprietà delle parole in quella frase e verrà poi passata in input al decoder.

Architettura Encoder-Decoder

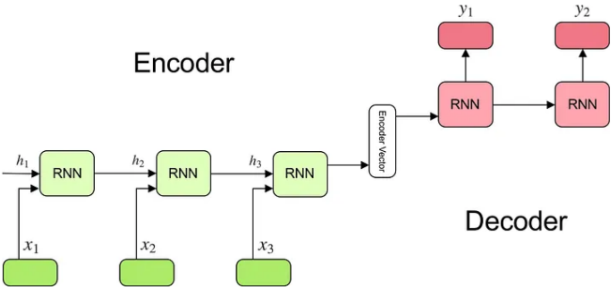


Figura 12: Archittetura encoder-decoder

Appendice

Transformer

Attention is all you need

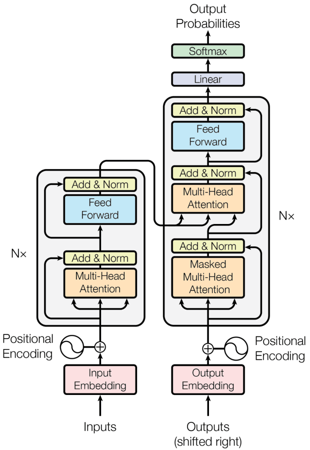


Figura 13: Architettura transformer

Architettura Transformer

Un modello transformer è basato sull' architettura encoder-decoder.

- L'encoder è composto da $N = 6$ layers identici. Ciascun layer ha due sublayer. Il primo di questi sub-layer è il layer di multi-head attention mentre il secondo è costituito da una rete feed-forward a multi strato che elabora le informazioni della sequenza elemento per elemento (*positionwise fully connected*) proprio come le usuali reti neurali multi strato che conosciamo.Inoltre vengono implementate le connessioni residuali ed un layer di normalizzazione ottenendo come output: $output = Norm(x + Sublayer(x))$. Sia nell'econder che nel decoder insieme ad il layer di embedding viene implementato un meccanismo di positional encoding.
- Il decoder è composto in maniera simile all'encoder con l'aggiunta di un terzo sublayer che implementa il meccanismo di multi-head-attention.

Multi-head attention

Multi-head-attention:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, \ldots head_h)W^O \tag{31}$$

$$head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{32}$$

dove $W^O \in \mathbb{R}^{hd_v \times d_{model}}$. e $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$ $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$ $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$ e $h = 8, \quad d_k = d_v = \frac{d_{model}}{h}$.

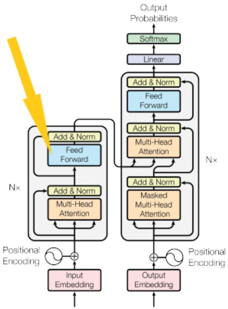
Appendice

Dinamica dei Transformers

Transformers come flow maps

Formuliamo ora per completezza la dinamica risultante di un transformer, che tiene in considerazione i layers di *feed-forward*:

$$\dot{x}_i(t) = P_{x_i(t)}\left(w(t)\sigma\left(\sum_{h=1}^{H}\sum_{j=1}^{n}\frac{e^{\beta<Q_h x_i(t),K_h x_j(t)>}}{Z_{\beta,i,h(t)}}V_h x_j(t) + b(t)\right)\right).$$



Approssimazione del flusso del gradiente di Wasserstein

Introduciamo gli spazi di *Polish*.

Definition 2

Sia (X, τ) uno spazio topologico, diremo che è *completamente metrizzabile* se ammette una metrica $d : X \times X \to R$ tale che induce la topologia τ su X e (X, d) è uno spazio metrico completo.

Definition 3

Uno spazio topologico (X, τ) è detto di Polish se è completamente metrizzabile ed è anche separabile.

Approssimazione del flusso del gradiente di Wasserstein

Definition 4

Sia $(\mathcal{X}, d)$ uno spazio metrico di Polish, e sia $p \in [0, +\infty)$. Siano μ, η due probabilità di misura qualsiasi su $\mathcal{X}$, la distanza di Wasserstein di ordine p tra μ e η è definita come:

$$W_p(\mu, \eta) = \left(\inf_{\gamma \in \Gamma(\mu, \eta)} \int_{\mathcal{X} \times \mathcal{X}} d(x, y)^p \, d\gamma(x, y) \right)^{\frac{1}{p}} \tag{33}$$

dove $\Gamma(\mu, \eta)$ è lo spazio di misure di probabilità congiunta sullo spazio prodotto $\mathcal{X} \times \mathcal{X}$ con marginali rispettivamente μ e η.

Approssimazione del flusso del gradiente di Wasserstein

Si potrebbe pensare di vedere l'equazione di continuità (16) come il flusso del gradiente rispetto alla metrica di Wasserstein, del funzionale E_β o di qualsiasi altro funzionale. Ricordando la definizione di gradiente rispetto alla metrica di Wasserstein di un funzionale,per far si che (16) sia il flusso del gradiente di E_β rispetto alla metrica di *Wasserstein*, il campo vettoriale $\mathcal{X}[\mu]$ definito come in (15), deve essere il gradiente della variazione prima δE_β di E_β. Tuttavia si osservi che abbiamo:

$$\mathcal{X}[\mu](x) = \nabla \log \int \beta^{-1} e^{\beta\langle x,y\rangle}\, d\mu(y). \tag{34}$$

Approssimazione del flusso del gradiente di Wasserstein

Nei lavori di [6] si fa uso delle ipotesi di simmetria del sistema, (vedi ipotesi 1 nel paper citato) in modo tale che si riesce ad ottenere (16) come flusso del gradiente di Wasserstein di un certo funzionale (vedi [6, Proposition 2]). Tuttavia non facendo queste assunzioni è stato mostrato che (34) non è il flusso del gradiente di Wasserstein di un funzionale.

Approssimazione del flusso del Gradiente di Wasserstein

Sketch dimostrazione per un caso particolare di flusso del gradiente di Wasserstein. Quando $\mu(t) = \frac{1}{n}\sum_{i=1}^{n}\delta_{x_i(t)}$ l'interazione di energia diventa:

$$E_\beta(X) = \frac{1}{2\beta n^2}\sum_{j=1}^{n}\sum_{i=1}^{n} e^{\beta\langle x_i, x_j\rangle} \tag{35}$$

dove $X = (x_1 \dots x_n) \in (\mathbb{S}^{d-1})^n$.

Denotando con ∇_X il gradiente associato alla metrica Riemanniana su $\mathbb{S}^{d-1}$ otteniamo la dinamica

$$\dot{X}(t) = n\nabla_X E_\beta(X(t)). \tag{36}$$

Approssimazione del flusso del Gradiente di Wasserstein

Infatti, consideriamo il gradiente gradiente $\nabla = (\partial_1 \ldots \partial_n)$ dove δ_i agisce sulla i-esima copia di $(S^{d-1})^n$. Abbiamo quindi:

$$\partial_i E_\beta(X(t)) = \frac{1}{\beta n^2} \sum_{j=1}^{n} P_{x_i(t)} \left(e^{\beta \langle x_i(t), x_j(t) \rangle} \beta x_j(t) \right) = \frac{1}{n} \dot{x}_i(t) \qquad (37)$$

e (36) risulta soddisfatta.

Approssimazione del flusso del gradiente di Wasserstein

Per una particolare scelta delle matrici parametriche (Q, K, V), la dinamica della self attention (10) può essere vista come flusso del gradiente di E_β con un'opportuna metrica modificata dello spazio tangente su $(\mathbb{S}^{d-1})^n$. Supponiamo allora che $Q^T K$ sia una matrice simmetrica e $V = Q^T K$. Allora poniamo $X = (x_1, \ldots, x_n) \in (\mathbb{S}^{d-1})^n$. Consideriamo il prodotto interno su $T_X(\mathbb{S}^{d-1})^n$ dato da:

$$\langle (a_1, \ldots, a_n), (b_1, \ldots, b_n) \rangle = \sum_{i=1}^{n} Z_{\beta,i}(X) a_i \cdot b_i \tag{38}$$

dove $a_i,\ b_i \in T_{x_i}\mathbb{S}^{d-1}$ e

$$Z_{\beta,i}(X) = \sum_{j=1}^{n} e^{\beta \langle V x_i, x_j \rangle}.$$

Approssimazione del flusso del gradiente di Wasserstein

Poniamo

$$E_\beta(X) = \frac{1}{2\beta} \sum_{i=1}^{n} \sum_{j=1}^{n} e^{\beta \langle V x_i, x_j \rangle}. \tag{39}$$

Si dimostra che (10) può essere equivalentemente scritta come

$$\dot{X} = \nabla E_\beta(X(t)) \tag{40}$$

dove il gradiente è calcolato rispetto alla metrica definita in (38) su $(\mathbb{S}^{d-1})^n$.

Appendice

Fenomeno di clustering

Clustering

Il fenomeno di clustering è particolarmente rilevante in compiti come la predizione del token successivo. In tale contesto, la misura di output codifica la distribuzione di probabilità del token successivo, e il formarsi di clusters indica un numero limitato di possibili risultati. Si ottengono diversi risultati che mostrano che la distribuzione limite è una massa puntiforme e questi risultati valgono per una scelta specifica di matrici parametriche. Tutto ciò sembra essere in contrasto con le osservazioni pratiche, infatti esperimenti numerici indicano un quadro più sottile per diverse matrici di parametri: ad esempio, compare una lunga fase metastabile durante la quale le particelle si coalizzano in un numero limitato di cluster, il che sembra coerente con il comportamento nei modelli preaddestrati.

Clustering

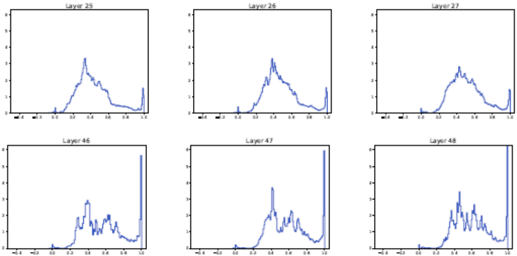


Figura 15: Istogramma di $\{\langle x_i(t), x_j(t)\rangle, i \neq j\}$ al variare dei layers t del modello preaddestrato *ALBERT XLARGE v2* che ha delle matrici parametriche costanti. Viene selezionato in modo randomico un prompt, che in questo caso rappresenta un paragrafo di Wikipedia.

Cluster singolo in regime di alta dimensionalità

Teorema 5

Sia $n \geq 1$ e $\beta \geq 0$. Supponiamo che $d \geq n$. Consideriamo la soluzione unica $(x_i(\cdot))_{i=1,\ldots,n} \in (\mathbb{S}^{d-1})^n$ rispettivamente per i problemi di Cauchy (10) e (19), corrispondente alla sequenza di punti iniziale $(x_i(0))_{i=1,\ldots,n} \in (\mathbb{S}^{d-1})^n$ distribuiti in modo uniforme. Allora quasi certamente esiste $x^* \in (\mathbb{S}^{d-1})^n$ e costanti $C, \lambda > 0$ tali che:

$$\|x_i(t) - x^*\| \leq Ce^{-\lambda t}$$

per ogni $i \in \{1, \ldots, n\}$ e $t \geq 0$.

Il teorema di sopra vale anche per Q, K matrici quadrate arbitrarie e $V = I_d$.

Clustering singolo in regime di alta dimensionalità

Se $d < n$ grazie al teorema di Wendel possiamo ottenere un risultato di clustering singolo con probabilità almeno $p_{n,d} \in (0,1)$

Teorema 6

(Wendel) Siano $d, n \geq 1$ tale che $d \leq n$. Siano $x_1, x_2, \ldots, x_n$ una sequenza di punti in $\mathbb{S}^{d-1}$ indipendenti identicamente distribuiti in modo uniforme allora la probabilità che questi punti vivono nella stessa semisfera è:

$$\mathbb{P}\left(\exists\, w \in \mathbb{S}^{d-1} \mid \langle x_i, w\rangle > 0\ \forall i \in \{1, \ldots n\}\right) = 2^{-(n-1)} \sum_{k=1}^{d-1} \binom{n-1}{k}. \quad (41)$$

Cluster singolo in regime di alta dimensionalità

Osservazione: Il prodotto interno non è altro che una misura della similarità di due particelle. Dunque la formazione dei clusters avviene quando $\langle x_i(t), x_j(t)\rangle \to 1\ \forall i, j \in \{1, \dots n\}$ e per $t \to +\infty$.

Teorema 7

Siano $\beta \geq 0$, $d, n \geq 2$ arbitrarie. Consideriamo una sequenza iniziale $(x_i(0))_{i\in\{1,\dots,n\}} \in (\mathbb{S}^{d-1})^n$ di n punti ortogonali a due a due: $\langle x_i(0), x_j(0)\rangle = 0$ per $i \neq j$, e con $(x_i(\cdot))_{i\in\{1,\dots,n\}} \in C^0\left(\mathbb{R}_{\geq 0}; (\mathbb{S}^{d-1})^n\right)$ denotiamo l'unica soluzione del corrispondente problema di Cauchy per (10) (rispettivamente per (19)). Allora l'angolo $\angle(x_i(t), x_j(t))$ è lo stesso per tutti $i, j \in \{1, \dots, n\}$ distinti:

$$\angle(x_i(t), x_j(t)) = \theta_\beta(t)$$

per $t \geq 0$ e qualche $\theta_\beta \in C^0\left(\mathbb{R}_{\geq 0}; \mathbb{T}\right)$.

Cluster singolo in regime di alta dimensionalità

Teorema (6)

Inoltre, per (10), $\gamma_\beta(t) := \cos(\theta_\beta(t))$ soddisfa

$$\begin{cases} \dot{\gamma}_\beta(t) = \frac{2e^{\beta\gamma_\beta(t)}(1-\gamma_\beta(t))((n-1)\gamma_\beta(t)+1)}{e^\beta+(n-1)e^{\beta\gamma_\beta(t)}} & \text{per } t \geq 0, \\ \gamma_\beta(0) = 0, \end{cases} \tag{42}$$

e per (19), abbiamo

$$\begin{cases} \dot{\gamma}_\beta(t) = \frac{2}{n}e^{\beta\gamma_\beta(t)}(1-\gamma_\beta(t))((n-1)\gamma_\beta(t)+1) & \text{per } t \geq 0, \\ \gamma_\beta(0) = 0. \end{cases} \tag{43}$$

Il seguente risultato mostra che quando $d \gg n$, $t \to \gamma_\beta(t)$ è un'approssimazione valida per $t \to \langle x_i(t), x_j(t) \rangle$ ogni $i \neq j \in \{1, \ldots, n\}$ distinti.

Cluster singolo in regime di alta dimensionalità

Teorema 8

Fissiamo $\beta \geq 0$ *e* $n \geq 2$. *Allora esiste qualche* $d^*(n,\beta) \geq n$ *tale che per ogni* $d \geq d^*(n,\beta)$, *vale quanto segue. Considera una sequenza* $(x_i(0))_{i\in\{1,\dots,n\}}$ *di* n *punti indipendenti e uniformemente distribuiti su* $\mathbb{S}^{d-1}$, *e con* $(x_i(\cdot))_{i\in\{1,\dots,n\}} \in C^0\left(\mathbb{R}_{\geq 0}; (\mathbb{S}^{d-1})^n\right)$ *denotiamo l'unica soluzione del corrispondente problema di Cauchy per* (10). *Allora esiste* $C = C(n,\beta)$ *e* $\lambda = \lambda(n,\beta) > 0$, *tale che con probabilità almeno* $1 - 2n^2d^{-1/64}$, *vale*

$$|\langle x_i(t), x_j(t)\rangle - \gamma_\beta(t)| \leq \min\left\{2 \cdot c(\beta)^{nt}\sqrt{\frac{\log d}{d}}, Ce^{-\lambda t}\right\} \qquad (44)$$

per ogni $i \neq j$ *e* $t \geq 0$, *dove* $c(\beta) = e^{10\max\{1,\beta\}}$, *e* γ_β *è l'unica soluzione di* (42).

Curva di transizione di fase e fase metastabile

Data una sequenza iniziale di punti $(x_i(0))_{i\in 1,\dots,n} \in \mathbb{S}^{d-1}$ di punti randomici distribuiti in modo indipendente secondo la distribuzione uniforme su $\mathbb{S}^{d-1}$ e per ogni $0 < \sigma \ll 1$ fissato, definiamo la curva di transizione di fase come il bordo:

$$\Gamma_{d,\delta} = \partial \left\{ t, \beta \geq 0 \mid \underset{s\geq 0}{\operatorname{arginf}} \left(\mathbb{P}(\langle x_1(s), x_2(s)\rangle \ \geq 1-\delta) = 1 - 2n^2 d^{-\frac{1}{64}} \right) \right\} \tag{45}$$

dove $(x_i)(\cdot)_{i\in\{1,\dots,n\}}$ denota la soluzione corrispondente al problema di Cauchy (10). Il Teorema 8 ci fornisce allora l'intuizione di approssimare la curva di transizione di fase $\Gamma_{d,\delta}$ sui sottoinsiemi compatti di $(\mathbb{R}_{\geq 0})^2$ con

$$\Gamma_{\infty,\delta} = \{t, \beta \geq 0 \mid \gamma_\beta(t) = 1-\delta\}. \tag{46}$$

Curva di transizione di fase e fase metastabile

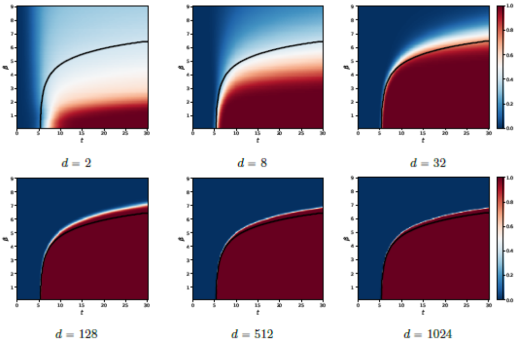


Figura 16: plot della probabilità di particelle randomiche che seguono la dinamica (10) e tendono a formare un singolo cluster. Si plotta la mappa: $(t, \beta) \mapsto \mathbb{P}_{(x_1(0), x_2(0), \dots, x_n(0)) \sim \sigma_d}(\{\langle x_1(t), x_2(t)\rangle \geq 1 - \delta\})$

Curva di transizione di fase e fase metastabile

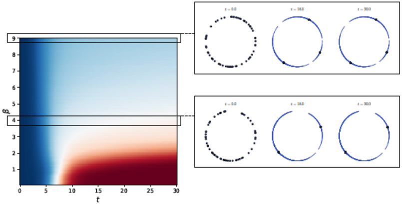


Figura 17: Diagramma di fase per la dinamica della self attention sul cerchio $d = 2$ per$\beta = 4, 9$.

Problema 9

Può la dinamica delle particelle entrare in uno stato metastabile a lungo periodo, nel senso che per $\beta \gg 1$ tutte le particelle si raggruppano in $m < n$ clusters per un lungo periodo, prima di collassare in un singolo cluster $\{x^*\}$?

Curva di transizione di fase e fase metastabile

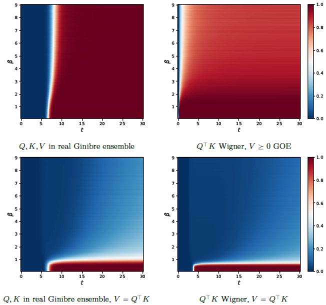


Figura 18: I diagrammi di fase per alcune matrici particolari.

Problema 10

Il Teorema 7 e il Teorema 8 può essere generalizzato per matrici (Q, K, V) più generiche?

Cluster singolo per valori piccoli di β

Caso $\beta = 0$. Nel caso $\beta = 0$ la dinamica della self attention (10) e di (19) diventa

$$\dot{x} = P_{x_i(t)}\left(\frac{1}{n}\sum_{j=1}^{n} x_j(t)\right) \tag{47}$$

Teorema 11

Sia $d, n \geq 2$. Per quasi ogni sequenza iniziale $(x_i)_{i\in\{1,\ldots,n\}} \in (\mathbb{S}^{d-1})^n$ rispetto alla misura di Lesbegue, esiste un qualche punto $x^* \in \mathbb{S}^{d-1}$ tale che la soluzione unica $(x_i(t))_{i\in\{1,\ldots,n\}} \in C^0(\mathbb{R}_{\geq 0}, (\mathbb{S}^{d-1})^n)$ associata al problema di Cauchy (47) soddisfa:

$$\lim_{t\to+\infty} x_i(t) = x^* \tag{48}$$

per ogni $i \in \{1, \ldots, n\}$.

Cluster singolo per valori piccoli di β

Caso $\beta \ll 1$. Il Teorema 11 ha anche implicazioni per β piccolo ma positivo. Otteniamo qualcosa di simile visto nei diagrammi precedenti. Infatti questo è dovuto al fatto che la dinamica della self attention può essere riscritta come:

$$\dot{x} = P_{x_i(t)}\left(\frac{1}{n}\sum_{j=1}^{n} x_j(t)\right) + \mathcal{O}(\beta) \tag{49}$$

Quindi per un tempo $t \ll \beta^{-1}$ le particelle non sentono l'influenza del resto $O(\beta)$ e il comportamento è lo stesso del caso $\beta = 0$.

Appendice

Matrici Generali

Caso Repulsivo

Caso $V = -I_d$. Minimizzare il funzionale di energia:

$$E_\beta[\mu] = \frac{e^{2\beta}}{2\beta} \int \int e^{-\beta \|x-x'\|} \, d\mu(x) d\mu(x') \tag{50}$$

Sullo spazio delle misure empiriche $\mathcal{P}_n(\mathbb{S}^{d-1})$ coincide col problema di trovare una configurazione ottimale della sfera. Riscriviamo $E_\beta[\mu]$ in termini dell'insieme di supporto $\mathscr{C} \subset \mathbb{S}^{d-1}$, $\#\mathscr{C} = n$

$$E_\beta[\mu] = H_\beta[\mathscr{C}] = \frac{e^{2\beta}}{2n^2\beta} \sum_{x,x' \in \mathscr{C}} e^{-\beta\|x-x'\|} \tag{51}$$

Caso Repulsivo

Definition 12

Sia $n \geq 2$. Un insieme di punti $\mathscr{C} = \{x_1, \ldots x_n\} \subset \mathbb{S}^{d-1}$ è chiamato spherical t-design se

$$\int p(x)\, d\sigma_d(x) = \frac{1}{n} \sum_{i=1}^{n} p(x_i) \tag{52}$$

per tutti i polinomi p in d variabili e grado al più t. $\mathscr{C}$ è una sharp configuration se ci sono m prodotti interni distinti tra le coppie di punti e se c'è uno spherical $(2m-1)$-design.

Teorema 13

Ogni minimo globale di H_β su $\mathscr{C} \subset \mathbb{S}^{d-1}$, con $\#\mathscr{C} = n$, è o una sharp configuration o i vertici di una 600-cell (un politopo convesso in 4 dimensioni con $n = 120$ vertici).

Un quadro completo della dinamica dei transformers nel caso repulsivo è ancora aperto.

Pure Self Attention

Un'alternativa per studiare la dinamica della (10) nel caso di matrici generali che ha portato notevoli risultati, è studiare una dinamica semplificata chiamata *pure self-attention*

$$\dot{x}(t) = \frac{1}{Z_{\beta,i}(t)} \sum_{i=1}^{n} e^{\beta \langle Qx_i(t), Kx_j(t) \rangle} Vx_j(t) \tag{53}$$

con $i \in \{1, \dots n\}$ e $t \in \mathbb{R}_{t \geq 0}$. Considerando le particelle "riscalate" $z_i(t) = e^{-tV} x_i(t)$:

$$\dot{z}_i(t) = \frac{1}{Z_{\beta,i}(t)} \sum_{j=1}^{n} e^{\beta \langle Qe^{tV} z_i(t), Ke^{tV} z_j(t) \rangle} V(z_j(t) - z_i(t)) \tag{54}$$

Pure Self Attention

V	K and Q	Limit geometry	Result in [GLPR23]
$V = I_d$	$Q^\top K \succ 0$	vertices of convex polytope	Theorem 3.1
$\lambda_1(V) > 0$, simple	$\langle Q\varphi_1, K\varphi_1 \rangle > 0$	union of 3 parallel hyperplanes	Theorem 4.1
V paranormal	$Q^\top K \succ 0$	polytope $\times$ subspaces	Theorem 5.1
$V = -I_d$	$Q^\top K = I_d$	single cluster at origin*	Theorem C.5

Figura 19: Geometria limite dei clusters

Pure Self Attention

Per esempio quando $V = I_d$ tranne in casi eccezionali [2] l'inviluppo convesso (*convex hull*) $\mathcal{K}$ delle particelle $z_i(t)$, si restringe e converge a qualche politopo convesso.

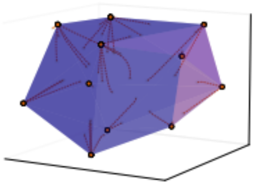


Figura 20: Quando $V = I_3$

Problema 14

Possiamo estendere i risultati di clustering ottenuti nella tabella 19 ad altre matrici Q, K, V? Inoltre quali sono le forme limite risultante?

Bibliografia i

[1] B. Geshkovski et al. **"A mathematical perspective on Transformers".** In: *arXiv preprint arXiv:2312.10794* (2023).

[2] B. Geshkovski et al. **"The emergence of clusters in self-attention dynamics".** In: *Advances in Neural Information Processing Systems*. Vol. 36. 2024.

[3] P. E. Jabin e S. Motsch. **"Clustering and asymptotic behavior in opinion formation".** In: *Journal of Differential Equations* 257.11 (2014), pp. 4165–4187.

[4] U. Krause. **"A discrete nonlinear and non-autonomous model of consensus".** In: *Communications in Difference Equations: Proceedings of the Fourth International Conference on Difference Equations*. CRC Press, 2000.

Bibliografia ii

[5] Y. Kuramoto. **“Self-entrainment of a population of coupled non-linear oscillators”.** In: *International Symposium on Mathematical Problems in Theoretical Physics*. Kyoto University, Kyoto/Japan, 1975.

[6] M. E. Sander et al. **“Sinkformers: Transformers with doubly stochastic attention”.** In: *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022.