DEEP LEARNING PER LA MATEMATICA SIMBOLICA

Elia Mercatanti

Relatore: *Donatella Merlini*

Università degli Studi di Firenze
Scuola di Scienze Matematiche, Fisiche e Naturali
Corso di Laurea in Informatica

Anno Accademico 2020-2021

Motivazioni

Successo delle reti neurali:

- Sono lo stato dell'arte in un ampia varietà di problemi.
- Sono estremamente efficaci nel *pattern recognition*.
- Limitato successo nel calcolo simbolico.
- Grande successo su compiti dell'elaborazione dei linguaggi naturali e traduzioni: manipolazione simbolica.

Perché applicare il *deep learning* al calcolo simbolico?

- Persone e macchine hanno difficoltà nell'eseguire complessi calcoli simbolici.
- Il *pattern recognition* può essere utile per l'integrazione.
- Gli approcci precedenti hanno quasi sempre considerato dataset molto piccoli.

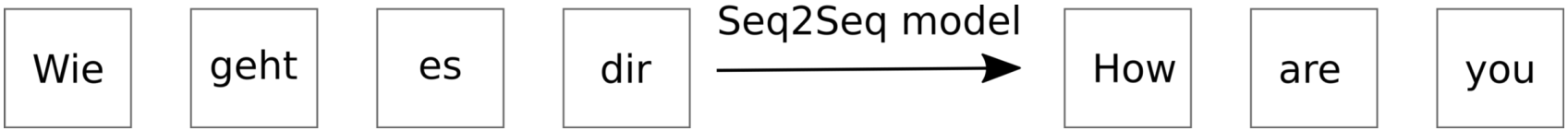
Intuizione di Base

Architetture per la Traduzione Automatica:

- Lavorano su frasi considerate come sequenze di *tokens*.
- Non hanno bisogno di specifiche informazioni sul problema.
- Usano enormi dataset.

Matematica Simbolica:

- Può essere considerata come un linguaggio.
- Possono essere generati grandi dataset.
- Risolvere un problema equivale a "tradurre" quest'ultimo nella sua soluzione.



Cosa è stato fatto:

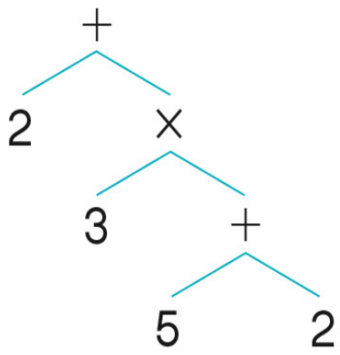
- Sono state usate tecniche per la traduzione automatica.
- Su due problemi: integrazioni, equazioni differenziali.
- Usando modelli *Sequence to Sequence* (Seq2Seq), in particolare il *Transformer*.
- Valutando la loro accuratezza e le soluzioni che restituiscono.

Cosa è necessario:

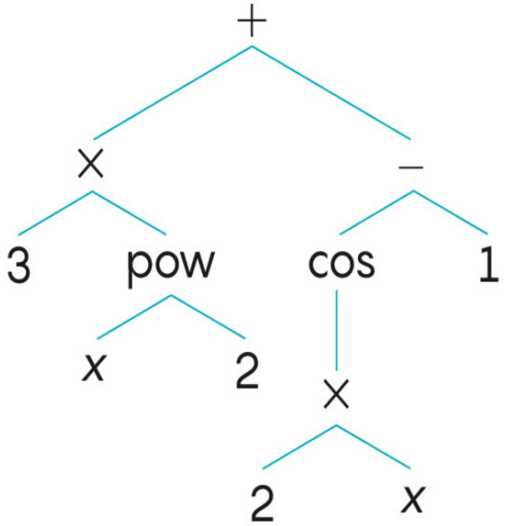
- Rappresentare problemi e soluzioni tramite sequenze.
- Metodi per generare grandi dataset di problemi e soluzioni.

Espressioni Matematiche in Forma di Alberi

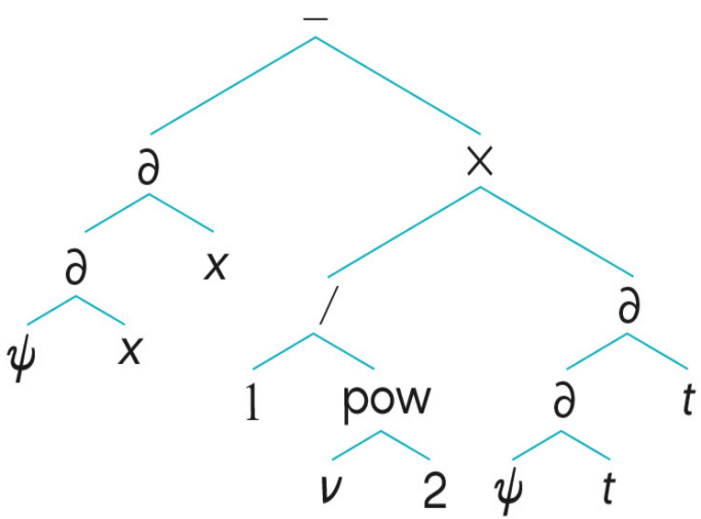
$2 + 3 \times (5 + 2)$



$3x^2 + \cos(2x) - 1$



$\frac{\partial^2 \psi}{\partial x^2} - \frac{1}{\nu^2} \frac{\partial^2 \psi}{\partial t^2}$

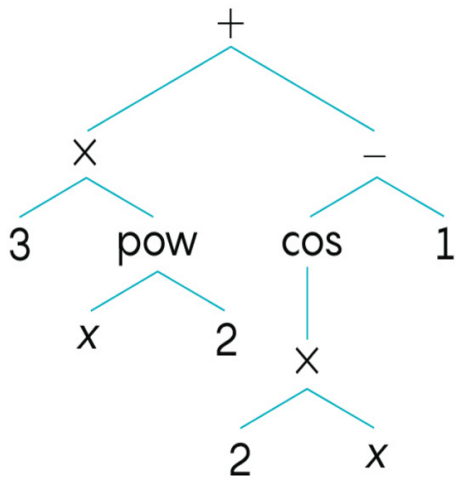


Vantaggi:

- Non ambiguità nell'ordine delle operazioni.
- Rimozione di simboli non significativi (parentesi, spazi, ecc.).
- Ad ogni espressione diversa corrisponde un albero diverso.
- Corrispondenza biunivoca tra espressioni ed alberi.

Dagli Alberi alle Sequenze

$3x^2 + \cos(2x) - 1$



Sequenza in Notazione Prefissa:

$[+ \times 3 \text{ pow } x \ 2 - \cos \times 2 \ x \ 1]$

Vantaggi sull'uso della notazione prefissa:

- Trasforma facilmente gli alberi in sequenze.
- Garantisce rappresentazione biunivoca tra sequenze ed alberi.
- Non necessita di parentesi finché ogni operatore ha un numero fisso di operandi.

Generare Espressioni Matematiche Casuali

Sono stati generati grandi dataset di espressioni matematiche casuali.

- Ogni tipo di problema con una strategia diversa.

Generazione di un singolo problema o soluzione casuale:

- Viene generato un albero unario-binario casuale.
- Ogni forma di albero ha la stessa probabilità di essere generata.
- Ogni nodo interno viene sostituito con un operatore/funzione casuale.
- Ogni nodo foglia viene sostituito con una costante, un intero, o una variabile casuale.

Integrazioni - Forward Generation (FWD)

Strategia del Generatore:

- Viene generata un funzione casuale f.
- Viene calcolata la sua primitiva F con un framework di matematica simbolica (*SymPy*).
- La coppia (f, F) viene aggiunta al dataset.

Caratteristiche:

- Richiede una framework di calcolo simbolico esterno.
- Limitato alle sole funzioni che il framework può integrare.
- Lento dal punto di vista computazionale.
- Tende a generare problemi corti con soluzioni lunghe.

Integrazioni - Backward Generation (BWD)

Strategia del Generatore:

- Viene generata un funzione casuale f.
- Viene calcolata la sua derivata f' con *SymPy*.
- La coppia (f', f) viene aggiunta al dataset.

Caratteristiche:

- La differenziazione è sempre possibile ed estremamente veloce.
- Non dipende da un sistema di integrazione simbolica esterno.
- Tende a generare problemi lunghi con soluzioni corte.
- Improbabile che venga generato l'integrale di funzioni semplici.

Integrazioni - Backward con Integrazione per Parti (IBP)

Strategia del Generatore:

- Vengono generate due funzione casuali F e G.
- Vengono calcolate la loro derivata f e g con *SymPy*.
- Se $f * G$ è presente nel dataset, viene calcolato l'integrale di $F * g$ con:

$$\int Fg = FG - \int fG$$

- Il nuovo integrale scoperto viene aggiunto al dataset.

Caratteristiche:

- Può generare gli integrali di funzioni semplici.
- Lento dal punto di vista computazionale.
- Espressioni generate simili al metodo Forward (FWD).

Strategia del Generatore:

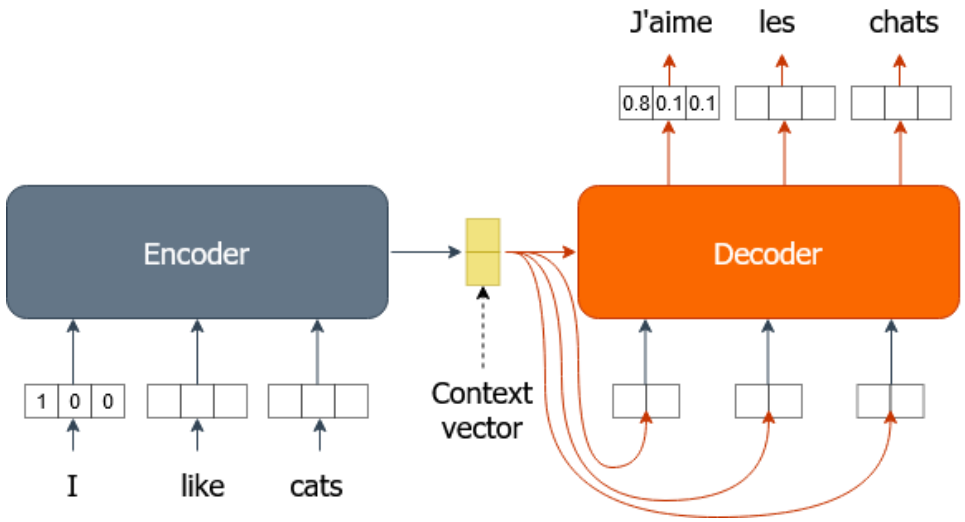
- Le soluzioni hanno un coefficiente costante c.
- Si parte da una soluzione $y = f(x, c)$ generata casualmente.
- La funzione viene risolta rispetto a c, ovvero ricaviamo $F(x, y) = c$.
- F viene differenziata rispetto a x.
- Il risultato viene semplificato.
- Otteniamo così un'equazione differenziale soddisfatta da y.

Caratteristiche:

- Lento dal punto di vista computazionale.
- Tende a generare problemi lunghi con soluzioni corte.

Modelli Sequence to Sequence

- Input ed output: sequenza di *tokens*.
- Due moduli: ***Encoder*** e ***Decoder***. Due reti ricorrenti.
  - ***Encoder***: accetta l'input del modello e lo codifica in un *context vector* di dimensione fissa.
  - ***Decoder***: utilizza il *context vector* come "seme" da cui generare una sequenza di output.
- La fase di inferenza può essere migliorata dalla *beam search*.



Attention Mechanism

Problemi:

- Le reti Seq2Seq sono costrette a codificare un'intera sequenza di testo in un vettore di lunghezza finita.
- Alcune parti dell'input possono rivelarsi più importanti di altre.
- Difficoltà nel memorizzare parole o il significato di una frase molto lunga.

Soluzione:

- Uso dell'*attention*, tecnica che imita l'attenzione cognitiva.
- Assegna un significato di importanza a diverse parti dell'input per ogni fase dell'output.
- Aiuta a tradurre in modo efficiente frasi lunghe.

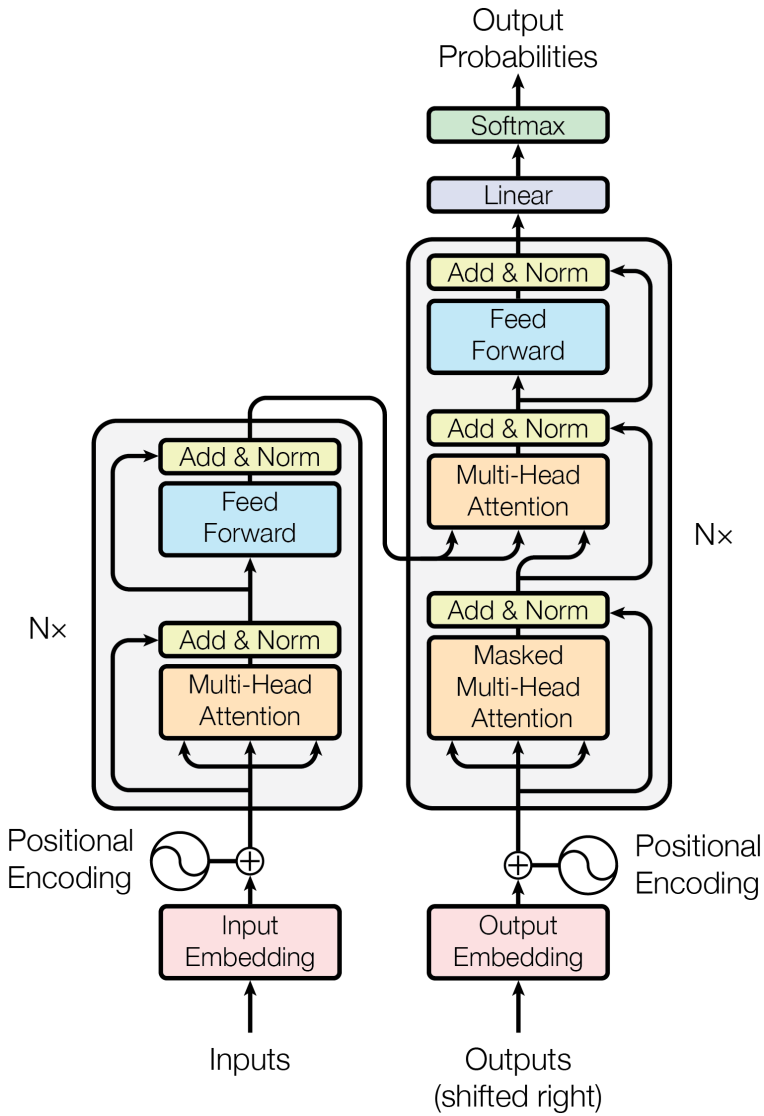
Il Modello Transformer e la Self-Attention

Caratteristiche:

- Nuovo stato dell'arte
- Maggiore parallelizzazione.
- Più efficiente e veloce.
- Sfrutta la *self-attention*.

Self-Attention:

- Estrapola la correlazione che esiste tra i *tokens* della sequenza di input.
- Trova una correlazione tra i *tokens* generati e il prossimo *token* di output.



Verifiche Sperimentali - Dataset e Modelli Utilizzati

- Espressioni con un massimo di $n = 15$ nodi interni e con una lunghezza massima pari a 512 *tokens*.
- Valori per i nodi foglia compresi in $\{x\} \cup \{-5, \ldots, 5\} \setminus 0$.
- Quattro operatori binari: $+$, $-$, $\times$, $/$.
- Quindici operatori unari: exp, log, sqrt, sin, cos, tan, $\sin^{-1}$, $\cos^{-1}$, $\tan^{-1}$, sinh, cosh, tanh, $\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$.

Dataset	Dimensione Train Set	Dimensione Validation/Test Set
Integrazioni - FWD	45 Milioni	10000
Integrazioni - BWD	88 Milioni	9000
Integrazioni - IBP	23 Milioni	8000
Equazioni Differenziali - ODE 1	65 Milioni	8000
Equazioni Differenziali - ODE 2	32 Milioni	9000

Tabella: Dimensioni dei dataset per integrazioni ed equazioni differenziali.

Verifiche Sperimentali - Metodo di Valutazione

- I modelli sono stati testati sui *test set* dei vari dataset.
- Utilizzando una *beam search* di dimensione 1 e 10.
- Le ipotesi delle soluzioni vengono controllate con *SymPy*.
- Se almeno una di esse è corretta allora viene approvato che il modello ha risolto il problema.

Accuratezza dei Modelli

Integrazioni	Forward (FWD)	Backward (BWD)	Integrazione per Parti (IBP)
Beam Search 1	95.6	98.4	97.7
Beam Search 10	97.2	99.5	99.4

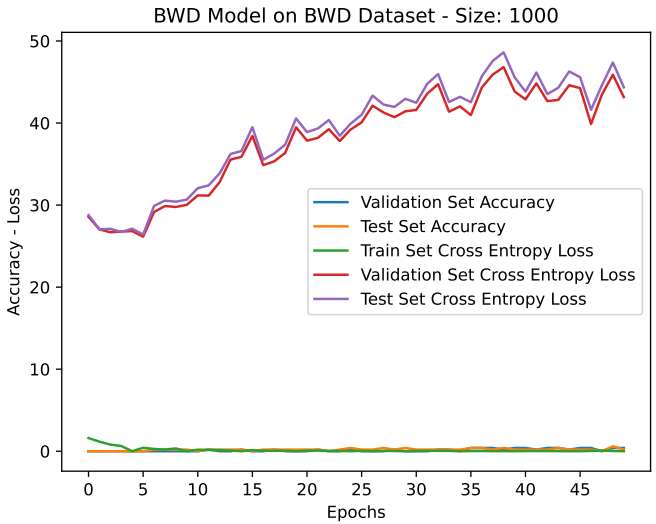
Tabella: Percentuali di accuratezza dei modelli per l'integrazione.

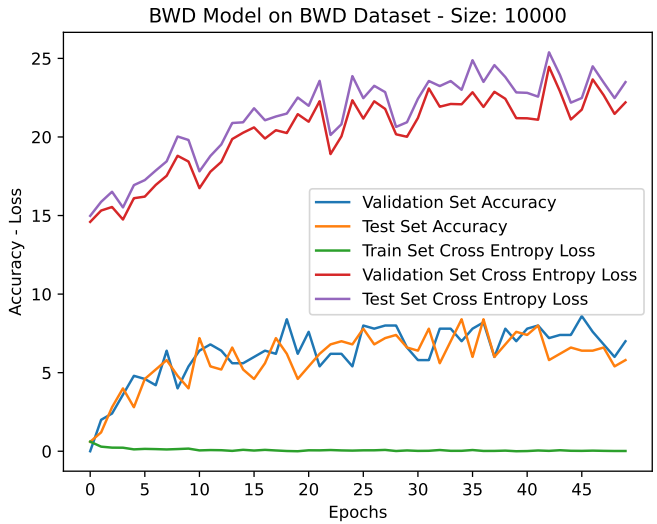
Equazioni Differenziali	ODE 1	ODE 2
Beam Search 1	89.5	75.0
Beam Search 10	96.8	85.5

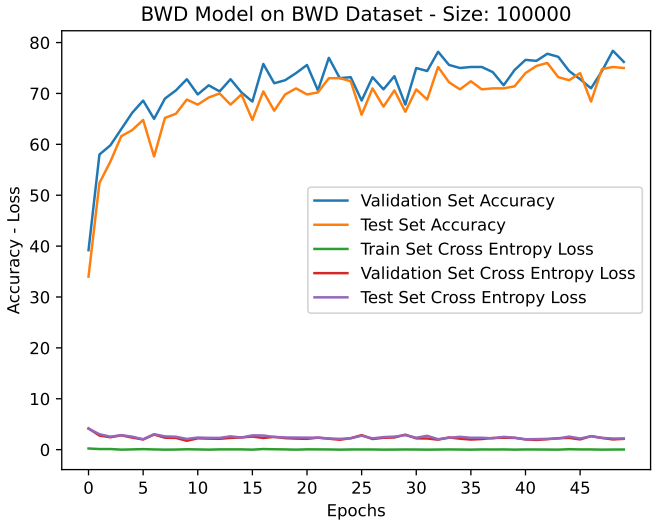
Tabella: Percentuali di accuratezza dei modelli per equazioni differenziali.

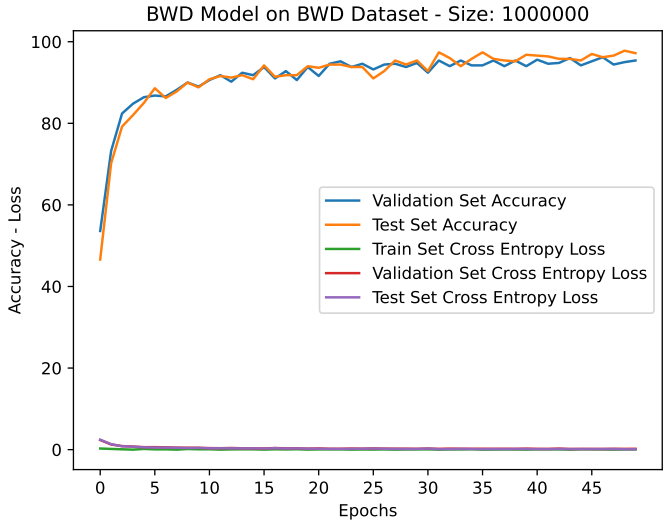
- **Integrazioni**: sopra al 95%, anche con *greedy search*.
- **Equazioni Differenziali**: sotto all'85%, la *beam search* aiuta.

Comportamento del Modello BWD all'Aumento dei Dati









Soluzioni Equivalenti

Ipotesi di soluzioni per l'equazione differenziale:

$$y' - y - xe^x = 0 \qquad \Longrightarrow \qquad y = e^x \left(\frac{x^2}{2} + c \right)$$

N.	Score	Hypothesis	N.	Score	Hypothesis
1	-0.083924	$x \left(\frac{c}{x} + \frac{x}{2} \right) e^x$	6	-0.194016	$\frac{x^2 e^x}{2} + e^{c+x}$
2	-0.089934	$\left(c + \frac{x^2}{2} \right) e^x$	7	-0.209069	$\frac{\left(c+x^2\right)e^x}{2} + e^x$
3	-0.129691	$ce^x + \frac{x^2 e^x}{2}$	8	-0.262188	$x \left(\frac{c}{x} + \sinh\left(\log\left(x\right)\right) \right) e^x$
4	-0.143644	$\frac{\left(c+x^2\right)e^x}{2}$	9	-0.275931	$x \left(\frac{c}{x} + \cosh\left(\log\left(x\right)\right) \right) e^x$
5	-0.164454	$\frac{x\left(\frac{c}{x}+x\right)e^x}{2}$	10	-0.309132	$\left(c + x \cosh\left(\log\left(x\right)\right) \right) e^x$

Comparazione con Frameworks di Matematica

	Integrazione (BWD)	ODE 1	ODE 2
Mathematica (30 s)	84.0	77.2	61.6
Matlab	65.2	-	-
Maple	67.4	-	-
Beam Search 1	98.4	81.2	40.8
Beam Search 10	99.6	94.0	73.2
Beam Search 50	99.6	97.0	81.0

Tabella: Confronto dei modelli con *Mathematica*, *Maple* e *Matlab*.

Conclusioni

- Il *deep learning* applicato alla matematica simbolica può dare ottimi risultati.
- Soluzioni valide rappresentate con espressioni differenti ma equivalenti.
- Le soluzioni proposte possono essere a volte errate, spesso è necessario concedere più ipotesi al modello.
- La correttezza delle soluzione non è fornita dal modello stesso.
- Sono richiesti dataset molto grandi.
- I generatori dei dati devono rappresentare correttamente l'intero spazio dei problemi.
- Grande vantaggio se integrati nei risolutori dei framework standard di matematica.

Grazie per l'Attenzione

Bibliografia:
- Guillaume Lample, François Charton, *Deep Learning for Symbolic Mathematics*, arXiv preprint arXiv:1912.01412, 2019.