NLP, tra democrazia e sostenibilità

Luca M.

Hackmeeting - Torino

2022

Mi presento

Nome: Luca M.

Professione: Studente, secondo anno laurea magistrale in informatica, curriculum artificial intelligence.

Mail: terraformer.144@gmail.com

Introduzione

Verranno trattati i seguenti punti,

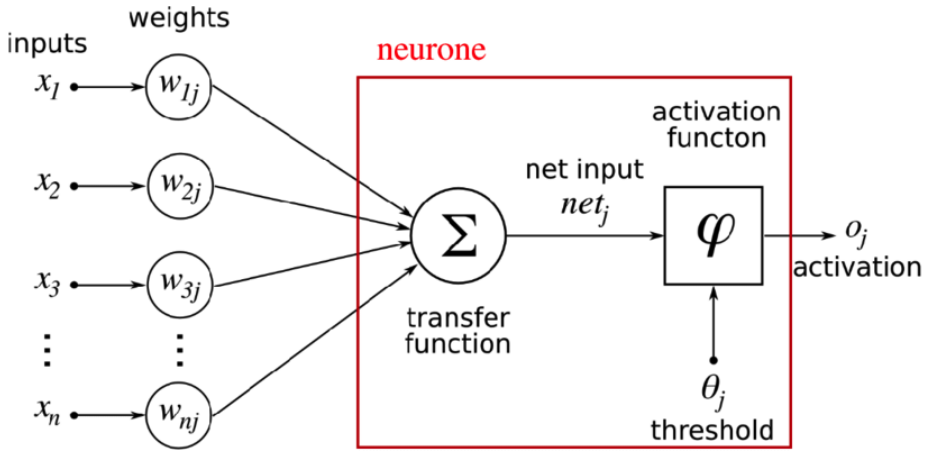
- Cosa è una rete neurale.
- Deep Learning, Transformers e BERT.
- É solo un fatto di dimensione ?.
- Un problema di democrazia e rappresentatività.
- Impatto ambientale.
- Cambiare direzione.
- Extreme Machine Learning e Reservoir Transformers.

Percettrone e Reti Neurali Artificiali

Le componenti atomiche di una rete neurale artificiale sono i **Percettroni**. Un percettrone è un modello computazionale che tenta di emulare il funzionamento di un neurone biologico.

$$o = \psi(\sum w_i * x_i + \theta)$$

Dove, le x_i rappresentano l'input, w_i vengono chiamati pesi, θ il bias, ψ è una funzione non lineare di attivazione e o è l'output del percettrone.



Percettrone e Reti Neurali

Una rete neurale artificiale è solitamente composta da più strati di percettroni.

Theorem (Cybenko)

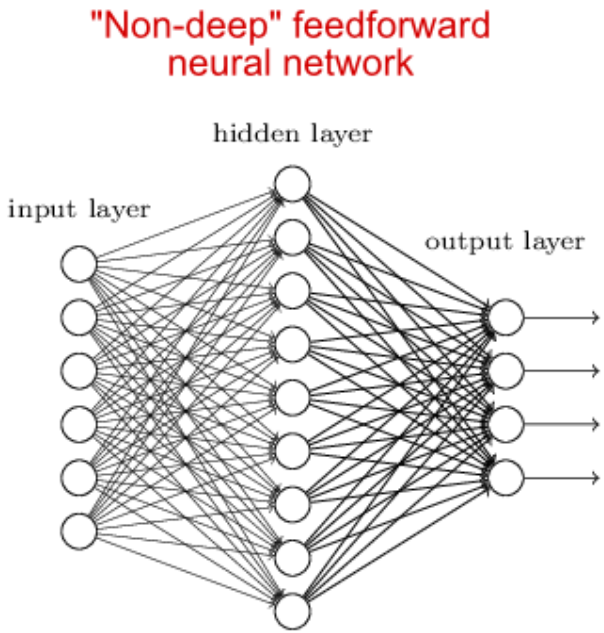
Ogni funzione continua definita in un sottoinsieme compatto di $[0-1]^n$ *può essere approssimata da una rete neurale feed-forward con un singolo layer con un numero finito di neuroni.*

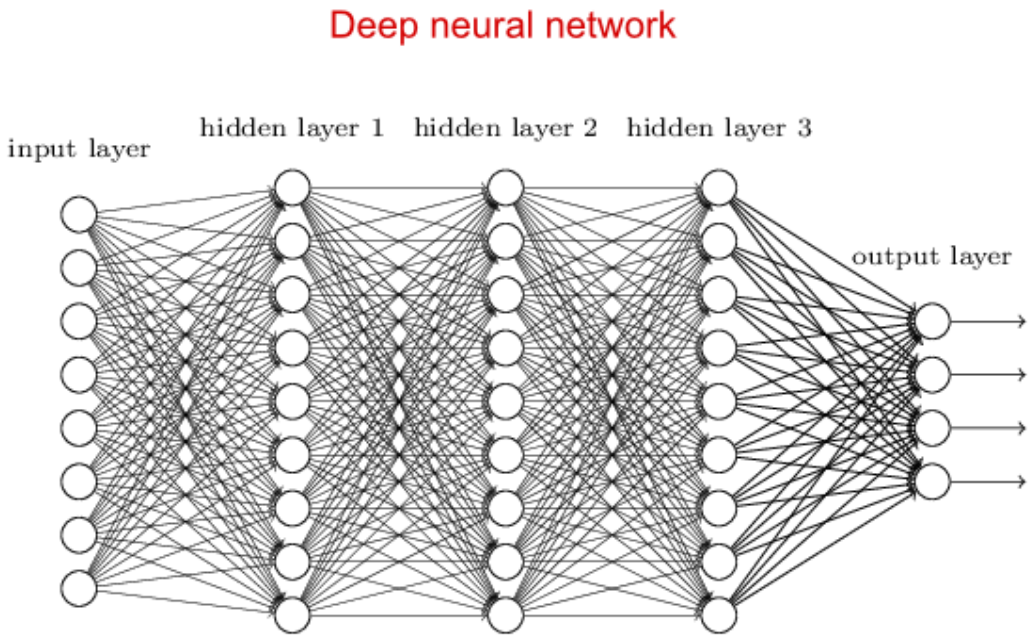
Le reti neurali hanno una forte capacità espressiva.

É possibile dunque vedere una rete neurale come una funzione $y = f(x_1, ..., x_n, w_1, ..., w_m)$ definita in funzione dell'input e dei pesi.

Deep Learning

Un modello di Deep Learning ha come particolarità l'utilizzo di una ingente quantità di percettroni impilati, formando dunque delle reti neurali cosiddette profonde.





Discesa del gradiente

Possiamo definire la funzione d'errore che la rete neurale compie nei dati in funzione dei pesi che la componngono. L'obiettivo è trovare la giusta configurazione dei pesi (w_i) nei percettroni che minimizzano tale funzione d'errore nei dati.

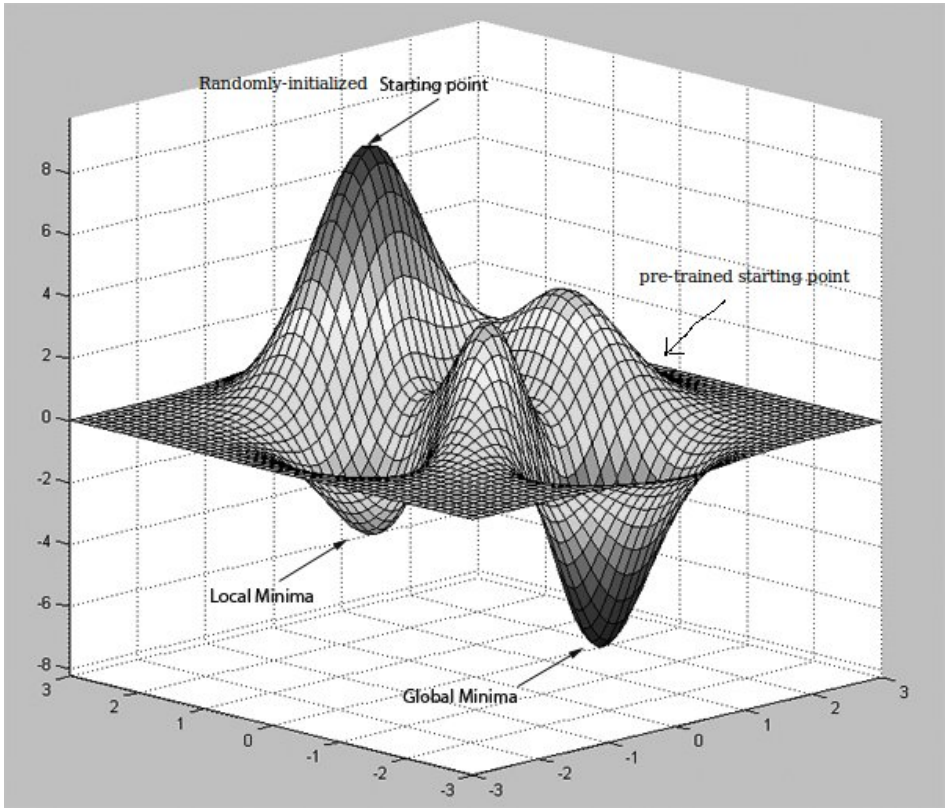
$$e.g. \quad M.S.E. = \frac{1}{N}\sum_{1}^{N}(y_i - \hat{y}_i)^2$$

Solitamente l'allenamento di una rete neurale artificiale utilizza algoritmi di discesa del gradiente, dunque *metodi iterativi le cui iterazioni hanno un costo computazionale proporzionale al numero di parametri del modello*.

Per il calcolo del gradiente si applica la regola della catena.

Discesa del gradiente

$$\Delta w = -\frac{\partial E_{tot}}{\partial w}$$



Neural Architectural Search

Un modello di deep learning può avere diverse configurazioni, ovvero è possibile definire differenti architetture, normalmente l'architettura ed tipo di iperparametri ottimali dipendono dall'ambito applicativo in cui il modello dovrà agire.

Con **Iperparametri** intendiamo tutti quei parametri che una volta impostati non verranno modificati dai metodi di apprendimento (e.g. parametri degli algoritmi di discesa del gradiente).

I metodi automatici per definire l'architettura ottimale a seconda del problema che dobbiamo affrontare prendono il nome di **Neural Architecture Search**.

DL: un approccio recente

I modelli di deep learning sono di rilevante importanza poiché hanno stravolto la concorrenza ed attualmente rappresentano lo stato dell'arte in vari settori, sono dunque la tipologia di modelli predominante nella maggior parte delle applicazioni di intelligenza artificiale.
Il deep learning è un approccio abbastanza recente.

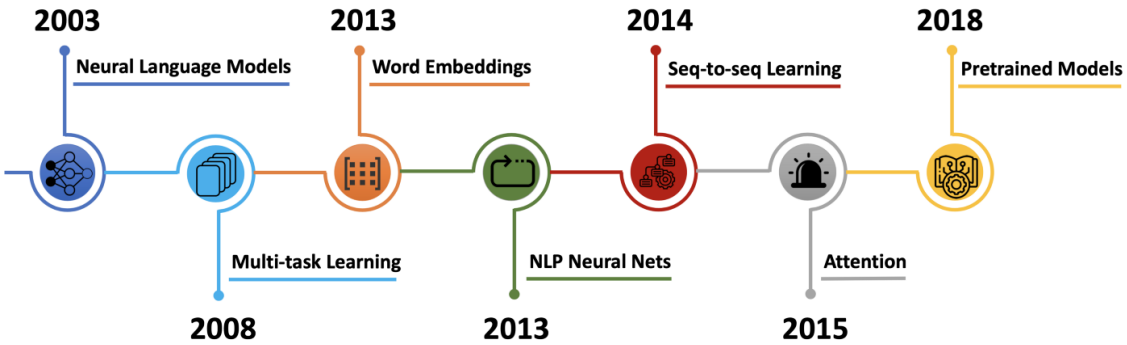


Figure: timeline Natuarl Language Processing

DL: un approccio recente

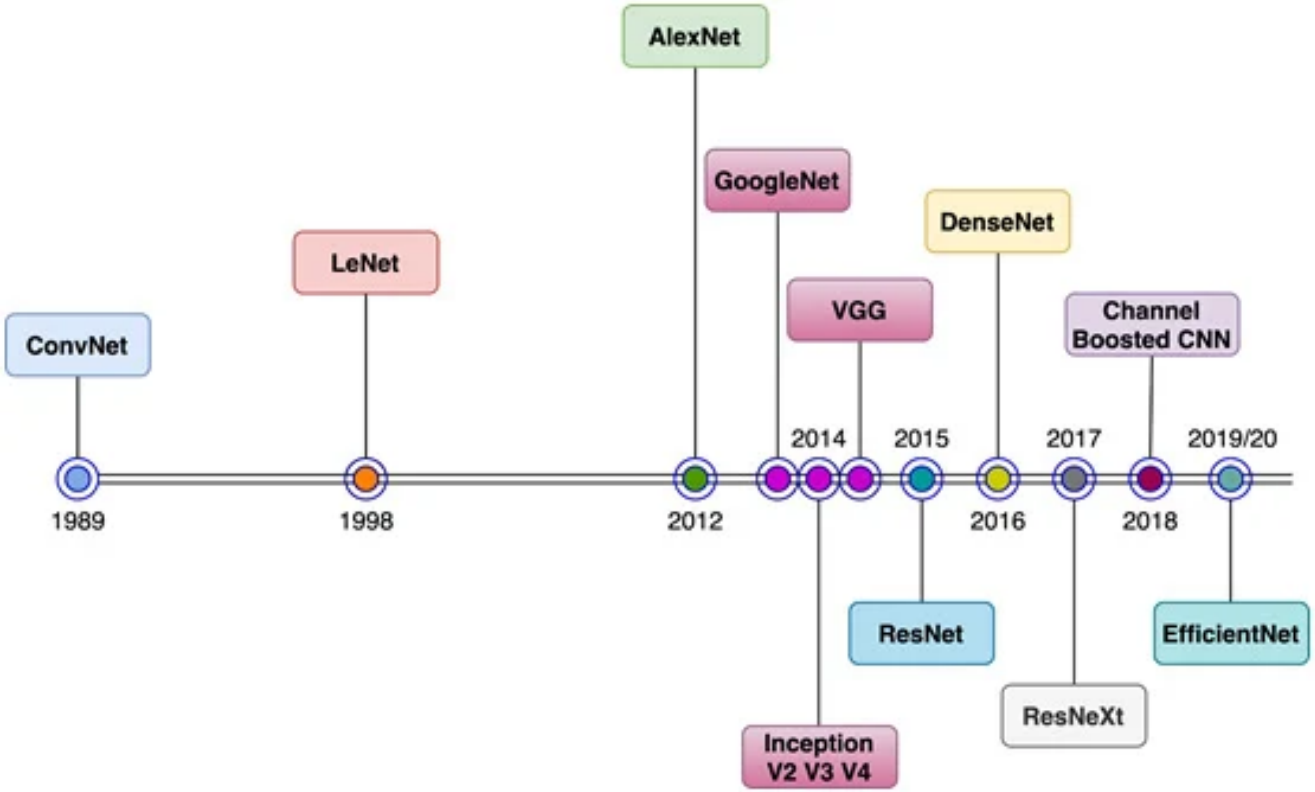
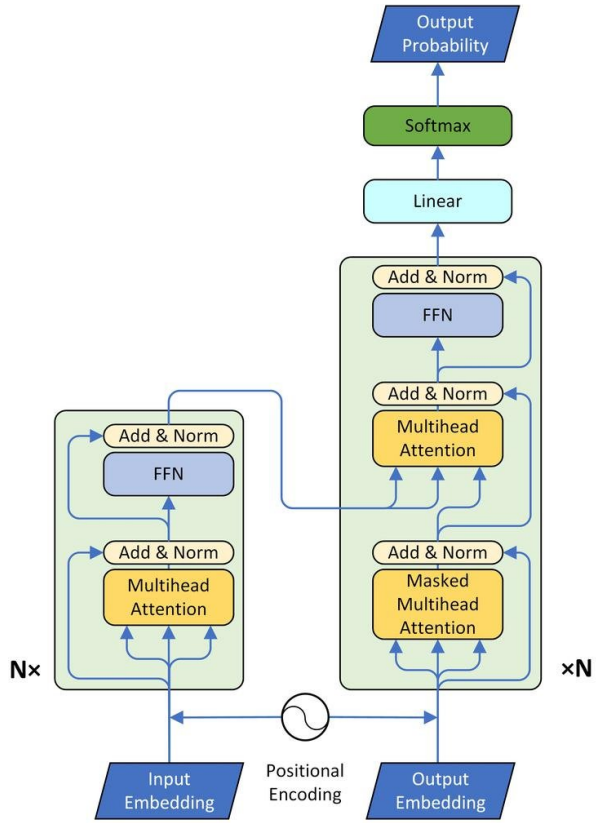


Figure: timeline Computer Vision

Attenzione! ai modelli che prestano attenzione

Il concetto di attenzione, in un modello di intelligenza artificiale, indica la capacità dello stesso di poter dare più o meno conto a differenti parti dell'input.

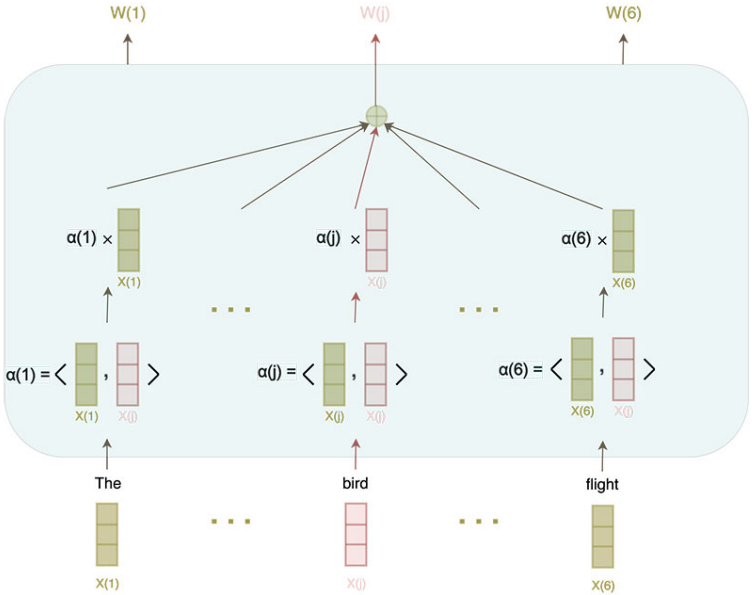
Recentemente è stata sviluppata una tipologia architetturale, chiamata Transformer, basata interamente sul concetto di attenzione (Vaswani et al. [5]).



Attenzione! ai modelli che prestano attenzione

La componente principale di un transformer è la **componente di attenzione**, ovvero un modulo che prende in input una sequenza di vettori di lunghezza l e dimensione d e restituisce una sequenza di vettori della stessa lunghezza e dimensione, ogni elemento della sequenza risultante è il risultato di una combinazione convessa di trasformazioni lineari degli elementi della sequenza di input.

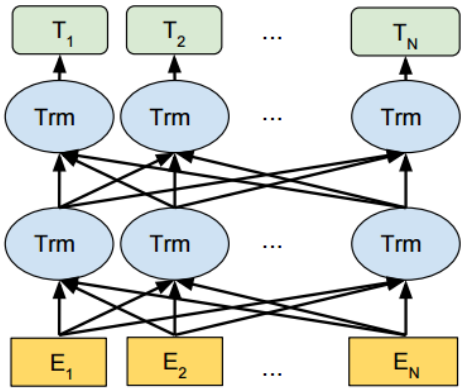
$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d}})V$$



BERT: Bidirectional Encoder Representations from Transformers

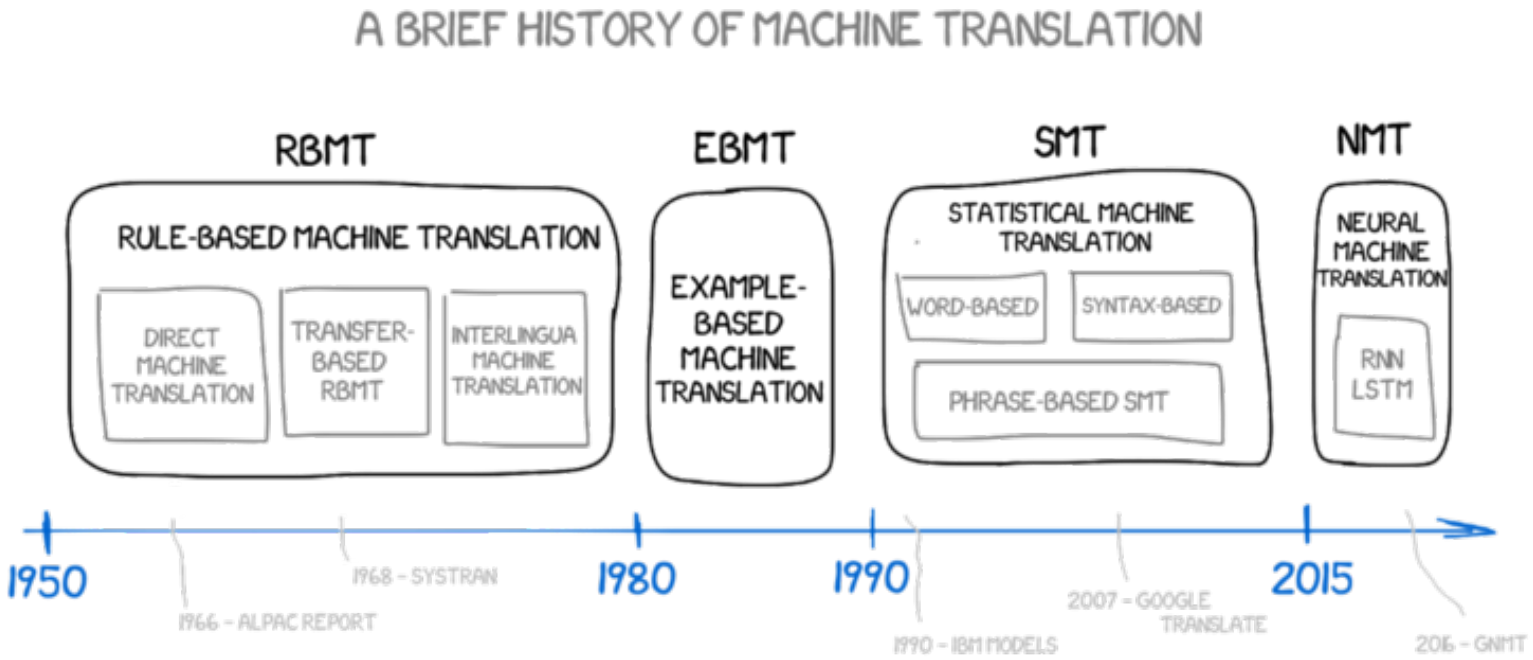
Ad oggi modelli che fanno uso di transformers sono largamente utilizzati nei più svariati ambiti applicativi: computer vision, NLP, design molecolare, ecc. I transformers sono stati dapprima applicati nell'ambito del NLP, tramite lo sviluppo di Neural Language Models come **BERT**.

BERT è un modello linguistico, sviluppato impilando vari strati di transformer encoders, utilizzato solitamente per generare rappresentazioni dipendenti dal contesto.



BERT:

L'avvento di bert è stato rivoluzionario, è stato da subito utilizzato come principale tipologia architetturale nei modelli per la traduzione automatica di google.



É solo un fatto di dimensione ?

Studi recenti (Schwartz et al. [2], Bender et al. [1]) hanno fatto vedere come l'andamento dei modelli è quello di richiedere sempre più potenza di calcolo per fare previsioni e per l'allenamento. E' possibile notare come la necessità di calcolo per l'allenamento di un modello sia aumentata del 300.000x in 6 anni.

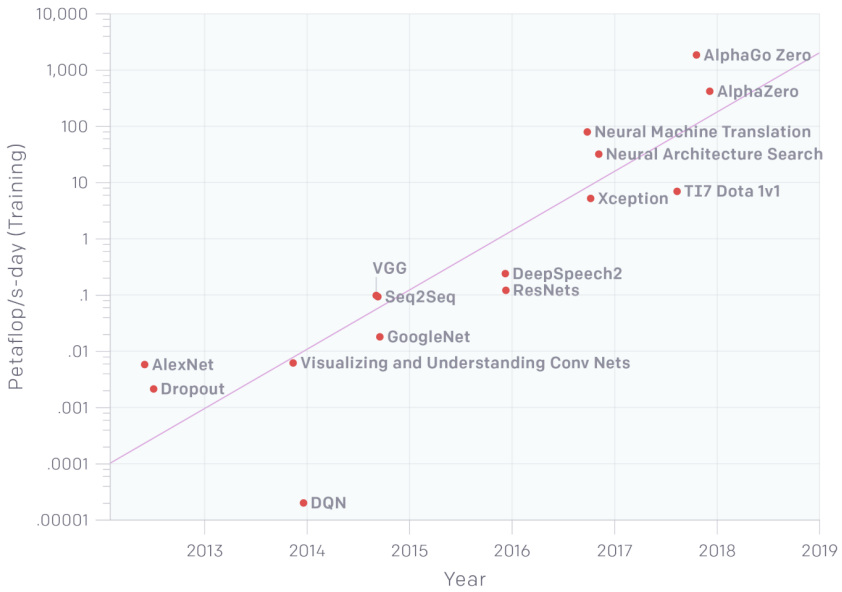


Figure: Figura 1 da [2]

Year	Model	# of Parameters	Dataset Size
2019	BERT [39]	3.4E+08	16GB
2019	DistilBERT [113]	6.60E+07	16GB
2019	ALBERT [70]	2.23E+08	16GB
2019	XLNet (Large) [150]	3.40E+08	126GB
2020	ERNIE-GEN (Large) [145]	3.40E+08	16GB
2019	RoBERTa (Large) [74]	3.55E+08	161GB
2019	MegatronLM [122]	8.30E+09	174GB
2020	T5-11B [107]	1.10E+10	745GB
2020	T-NLG [112]	1.70E+10	174GB
2020	GPT-3 [25]	1.75E+11	570GB
2020	GShard [73]	6.00E+11	–
2021	Switch-C [43]	1.57E+12	745GB

Figure: Tabella 1 da [1]

É solo un fatto di dimensione ?

Il feroce aumento nella necessità di calcolo è dovuto ad un aumento esponenziale nella dimensione dei modelli attualmente nel mercato, e dei dati su cui questi ultimi si allenano. Sono parecchie le recenti ricerche nell'ambito dell'intelligenza artificiale che fanno parte della RED AI (definito in Schwartz et al. [2]), ovvero approcci che tentano di raggiungere lo stato dell'arte tramite il massiccio utilizzo di capacità computazionali, essenzialmente comprando migliori risultati.

Nelle maggiori conferenze del 2018/2019 il focus principale è stato nell'accuracy dei modelli e non nell'efficienza.

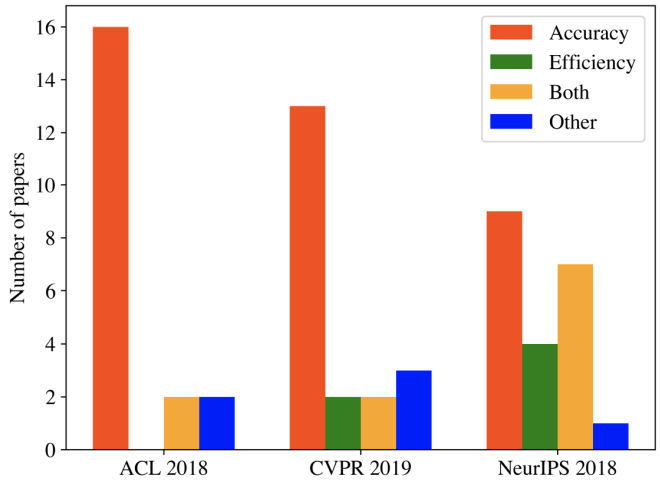
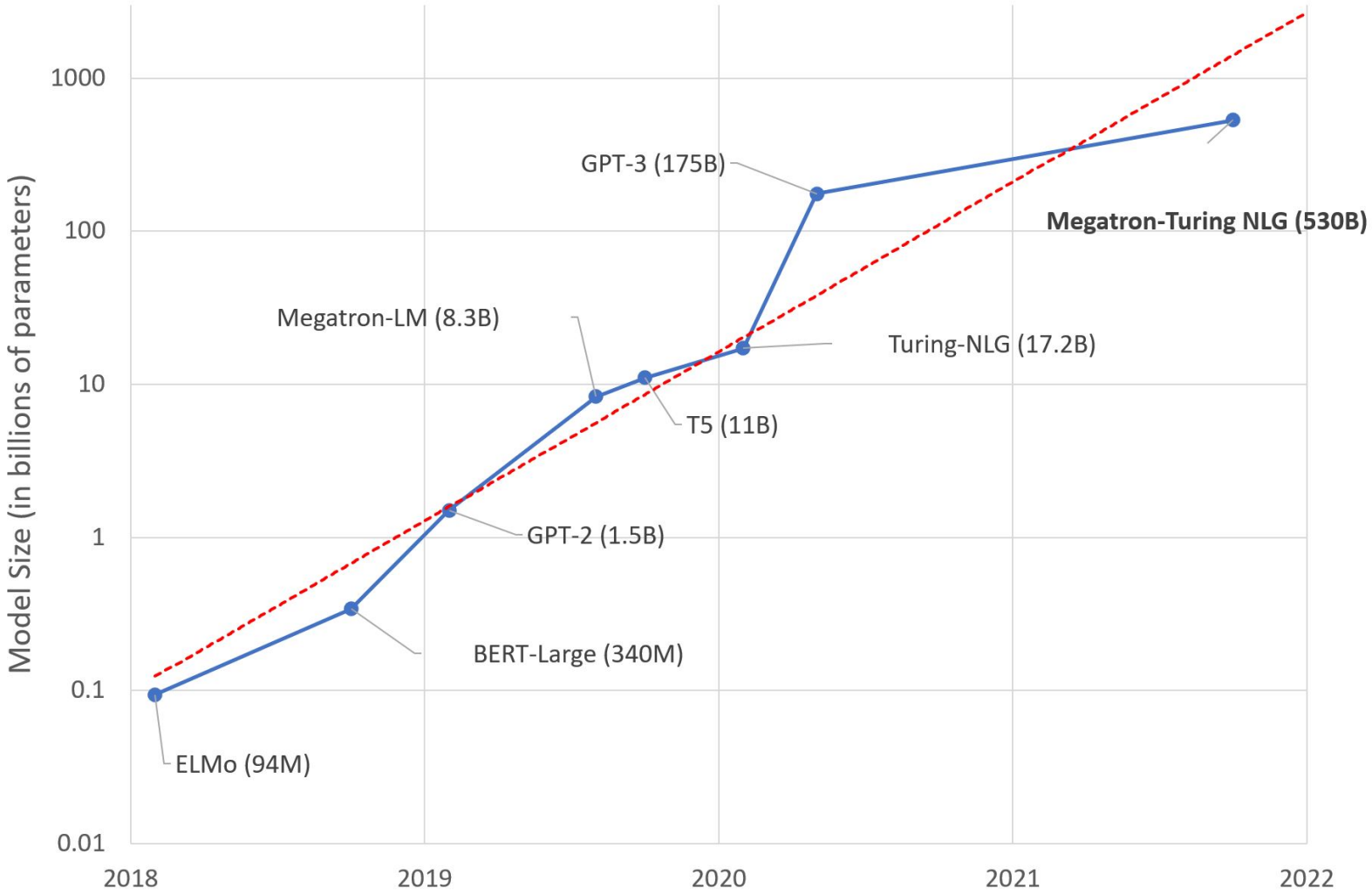


Figure: Figura 2 da [2]

É solo un fatto di dimensione ?



É solo un fatto di dimensione ?

E' comunque presente un trend, nella ricerca più recente, dal quale si evince una volontà nel ridurre la dimensionalità dei modelli.
Tramite l'utilizzo di knowledge distillation (DISTILBERT), quantizzazione, fattorizzazione delle matrici di embedding (ALBERT), ecc.

Alcuni di questi modelli non solo tentano, e riescono, a ridurre la dimensioni dei precedenti, ma alle volte ne aumentano persino le prestazioni.

- **DISTILBERT:** Sanh et al. 2020, link
- **ALBERT:** Lan et al. 2020, link

É solo un fatto di dimensione ?

E' di cattivo auspicio l'aumento nel numero di dati per l'allenamento.

Ad esempio BERT-large nel 2018 è stata allenata su **3 miliardi** di token, e modelli che successivamente hanno superato nella performance BERT-large, come XLNet e openGPT-2-XL hanno fatto uso di dataset molto più grandi, rispettivamente **32 e 40 miliardi** di token.

Un simile andamento è possibile da riscontrare nei modelli per la computer vision.

É possibile riscontrare un aumento non lineare delle performance rispetto all'aumento del numero di dati.

É solo un fatto di dimensione ?

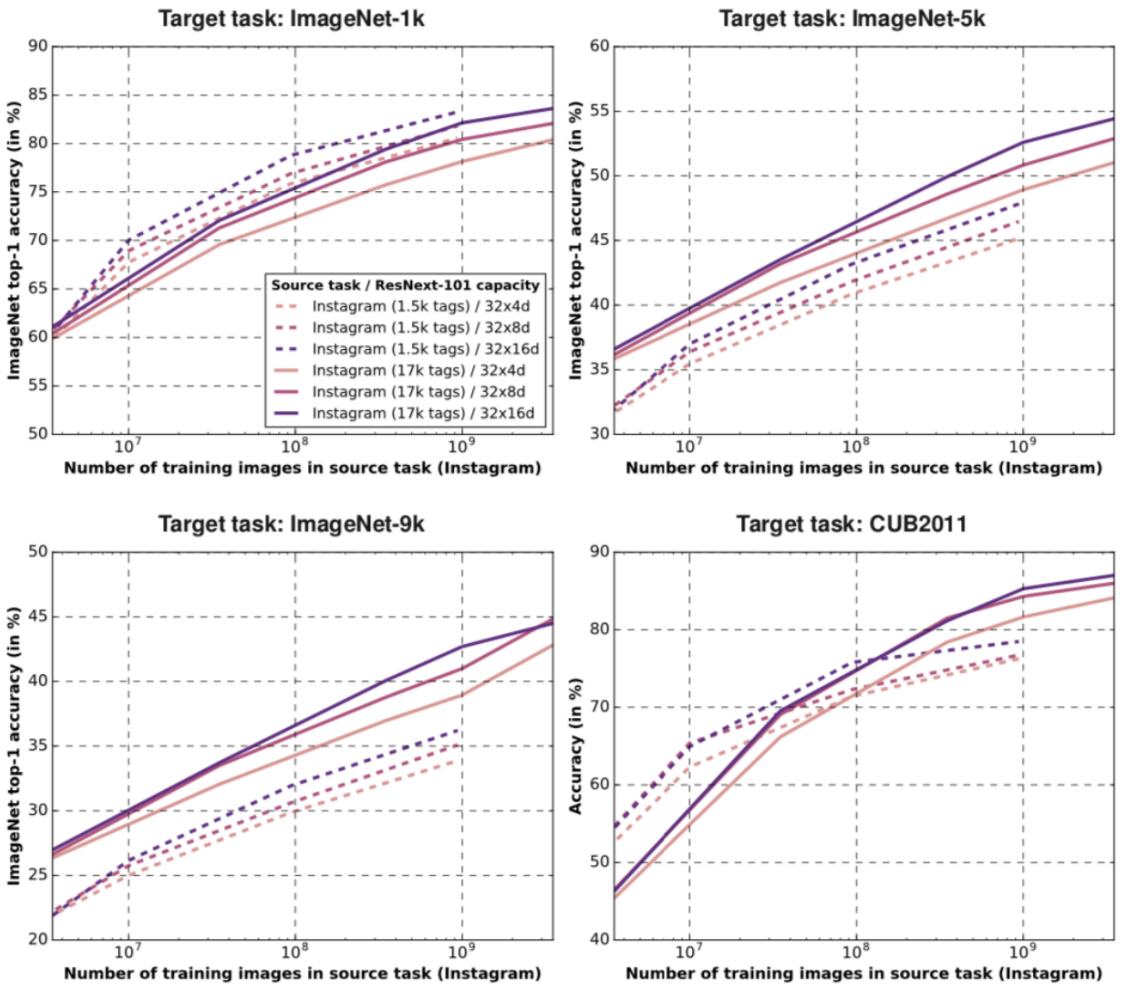


Figure: Figura 3 da [2]

Un problema di democrazia e rappresentatività

Un utilizzo così massiccio di dati nei modelli di NLP è deleterio (Bender et al. [1]), soprattutto se non viene effettuata un'adeguata fase di documentazione degli stessi.

I dati presi online senza alcuna cura hanno fatto vedere caratteristiche problematiche, i modelli risultanti presentano dei Bias dispregiativi di genere, razza, etnia e stati di disabilità.

I dataset basati interamente sullo scraping incondizionato del web sono predisposti a dare una visione egemonica/dominante, sotto rappresentando minoranze etniche e di pensiero.

> *Feeding AI systems on the world's beauty, ugliness, and cruelty, but expecting it to reflect only the beauty is a fantasy.*

Un problema di democrazia e rappresentatività

Tanti dati non garantiscono diversità.

Ad esempio il dataset Common Crawl (link), petabytes di dati collezzionati in anni di web crawling, ad un'attenta analisi non garantisce una rapppresentatività equa.

L'accesso al web non è equamente distribuito, sovrarappresentando utenti giovani ed appartenenti a paesi sviluppati.

Ad esempio i dati utilizzati per allenare GPT-2 vengono dallo scraping di link presenti su Reddit.
Un survey di Pew Internet Research ha fatto vedere come il **67%** degli utenti di Reddit in america sono **uomini** ed il **64%** appartengono ad una fascia d'età **tra i 18 ed i 29 anni**.

Un problema di democrazia e rappresentatività

I dati sono statici, le visioni sociali cambiano.
Movimenti sociali creano nuovi linguaggi, norme e modi di comunicare.
Questo rappresenta una sfida nello sviluppo di LM.

É importante notare che i movimenti sociali che sono poco documentati e che non ricevono un significativo impatto mediatico non verranno rappresentati in alcun modo o comunque potrebbero avere una rappresentazione non corretta.

I media tendono ad ignorare attività di proteste pacifiche concentrando l'attenzione su eventi drammatici e violenti a favore di servizi più accattivanti ed appetibili.

Un problema di democrazia e rappresentatività

Chi paga le conseguenze di un utilizzo improprio delle materie prime del nostro pianeta sono soprattutto paesi meno agiati di noi, sono gli stessi paesi le cui lingue non sono sviluppate per la maggior parte dei modelli di linguaggio attualmente nel mercato.

Come riportato in [1]:

> *È giusto dunque che i residenti delle Maldive (che probabilmente si troveranno sott'acqua entro il 2100) o che le 800.000 persone in Sudan colpite da drastiche inondazioni paghino il prezzo ambientale dello sviluppo di modelli per la lingua inglese sempre più grandi, quando modelli simili su larga scala non vengono prodotti per il Dhivehi o per l'Arabo Sudanese?*

Impatto Ambientale

Un andamento verso modelli sempre più grandi, ed una ingente quantità di dati su cui allenarli, comporta un uso maggiore di energia elettrica e dunque un impatto ambientale sostanziale.

É stato mostrato (Strubell et al. [4]) a quanto ammonta l'effettivo consumo nel deploying di modelli nell'ambito del NLP.

Consumption	**CO_2e (lbs)**
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

Figure: Tabella 1 da [4]

Impatto Ambientale

L'impatto ambientale è dovuto dal fatto che alcuni modelli richiedono tempi di allenamento sostanziali, settimane o mesi alle volte.

Alcune fonti energetiche potrebbero effettivamente derivare da fonti rinnovabili, ma purtroppo non è sempre così, in alcuni paesi l'energia deriva in buona parte da fonti non rinnovabili.

Consumer	Renew.	Gas	Coal	Nuc.
China	22%	3%	65%	4%
Germany	40%	7%	38%	13%
United States	17%	35%	27%	19%
Amazon-AWS	17%	24%	30%	26%
Google	56%	14%	15%	10%
Microsoft	32%	23%	31%	10%

Figure: Tabella 2 da [4]

Impatto ambientale

Riassumendo, siè fatto vedere come come allenare un singolo BERT su GPU è equivalente ad un volo trans-americano.

E' inoltre stato fatto vedere come l'utilizzo di NAS ha permesso di raggiungere un nuovo stato dell'arte, con un incremento di solo lo 0.1 BLEU (BiLingual Evaluation Understanding) con un aumento del costo computazione di 105K dollari ed una sostanziale emissione di Co2.

Compute ML CO2 Impact

Impatto Ambientale

Model	Hardware	Power (W)	Hours	kWh·PUE	CO_2e	Cloud compute cost
$\text{Transformer}_{base}$	P100x8	1415.78	12	27	26	\$41–\$140
Transformer_{big}	P100x8	1515.43	84	201	192	\$289–\$981
ELMo	P100x3	517.66	336	275	262	\$433–\$1472
BERT_{base}	V100x64	12,041.51	79	1507	1438	\$3751–\$12,571
BERT_{base}	TPUv2x16	—	96	—	—	\$2074–\$6912
NAS	P100x8	1515.43	274,120	656,347	626,155	\$942,973–\$3,201,722
NAS	TPUv2x1	—	32,623	—	—	\$44,055–\$146,848
GPT-2	TPUv3x32	—	168	—	—	\$12,902–\$43,008

Figure: Tabella 3 da [4]

”E noi come stronzi rimanemmo a guardare”

L’aumento esponenziale nella grandezza dei dataset, e nella dimensionalità dei più recenti modelli, comporta l’impossibilità da parte della maggior parte dei piccoli gruppi di ricerca di poter partecipare attivamente.

Parecchi esperimenti presenti nei più recenti paper fanno uso di dataset molto grandi, fino a 750 GB per modelli linguistici (2021).

Le tempistiche necessarie per l’allenamento di tali modelli, l’impatto ambientale ed il consumo di denaro, sia energetico (considerati i vari rincari) sia per affittare calcolo in cloud, possono rappresentare un ostacolo alle volte invalicabile.

Cambiare direzione

Vari ricercatori hanno proposto ai governi l'investimento in calcoloratori in cloud per permettere un accesso equo, e svincolarsi dai costi verso i grandi player privati.

Passi verso questa direzione sono già in atto:

- **HPC Cineca**, consorzio no-profit con la partecipazione di 69 università e 33 istituzioni.
- **Tecnopolo di Bologna**, centro nazionale di supercalcolo, finanziato dal governo italiano e dall'unione europea, per circa 300.000.000 euro.

Cambiare direzione

Nei lavori già citati (Schwartz et al. [2], Bender et al. [1]) vengono proposte alternative e posture di lavoro differenti, per mitigare uno sviluppo ed un utilizzo improprio delle tecnologie di AI.

E' possibile contrapporre allo sviluppo della RED AI, uno sviluppo più sostenibile, ovvero quello della GREEN AI.

Le tecniche di GREEN AI sono quelle che tentano di produrre risultati importanti senza aumentare il costo computazionale dei modelli, anzi tentando di ridurlo.

Cambiare direzione

Durante lo sviluppo di un nuovo modello è dunque importante riportare e considerare il costo finanziario e ambientale.

Una buona parte del tempo dovrebbe essere spesa per sviluppare datasets che ben si prestano allo specifico task invece di prendere ingenti quantità di dati direttamente dal web senza alcun filtro.

Una fase di documentazione dei dataset dovrebbe rientrare nelle spese di sviluppo di un progetto.

Cambiare direzione

Cercare capire come i modelli raggiungono gli obiettivi preposti e come possano entrare a far parte del sistema tecnico-sociale per cui sono stati pensati.

Ci si dovrebbe predisporre dunque per un tavolo di collaborazione con gli stakeholders (clienti diretti ed indiretti) del progetto, per comprendere a pieno gli obiettivi dei modelli sviluppati, per aiutare gli sviluppatori a creare sistemi che possano far interagire in modo sostenibile tecnologia e società.

Mettendo dunque tra gli obiettivi il benessere dell'individuo.

Misurare l'efficienza e non solo l'efficacia

É consigliabile di riportare l'intero lavoro richiesto per la produzione dei risultati di un progetto di AI [2], ovvero computazione necessaria all'allenamento e per la selezione degli iperparametri.

É importante evidenziare la dimensione del dataset utilizzato.

E' necessario utilizzare un'unità di misura valida per effettuare un confronto tra differenti lavori.

Un'unità di misura valida è il numero di **floating point operations** FPO (Schwartz et al. [2]), necessaria per stimare la quantità di lavoro effettuata da un processo computazionale.

Misurare l'efficienza e non solo l'efficacia

FPO può essere calcolata analiticamente definendo il costo di due operazioni di base, ADD e MUL, FPO può essere dunque calcolata in modo ricorsivo rispetto le operazioni di base.

FPO è un'unità di misura agnostica rispetto alle macchine utilizzate, ed è direttamente correlata al tempo di esecuzione del modello.

E' quindi auspicabile valutare la qualità di un lavoro non in funzione dei risultati rispetto allo stato dell'arte ma anche rispetto all'efficienza dei modelli presentati.

Extreme Machine Learning

Un secondo approccio alla GREEN AI potrebbe essere quello di utilizzare metodi di **Extreme Machine Learning (EML)**.

L'extreme machine learning comprende quelle tecniche di machine learning nelle quali gli strati interni di una rete neurale non vengono allenati ma inizializzati casualmente (dunque proiezioni randomiche non lineari), nell'accezione classica di EML solo l'ultimo strato è allenato, quando possibile in un unico step, tramite l'utilizzo del metodo dei minimi quadrati.

E' stato appurato che questi metodi hanno buone capacità di generalizzazione e richiedono molto meno tempo per essere allenati.

Extreme Machine Learning

Alla base dei modelli di EML è presente un importante teorema.

Theorem (Cover)

Un complesso problema di classificazione, proiettato, in modo non lineare, in uno spazio di dimensione maggiore, ha più probabilità di essere linearmente separabile rispetto al problema nello spazio di partenza, a condizione che lo spazio non sia densamente popolato.

EML in NLP

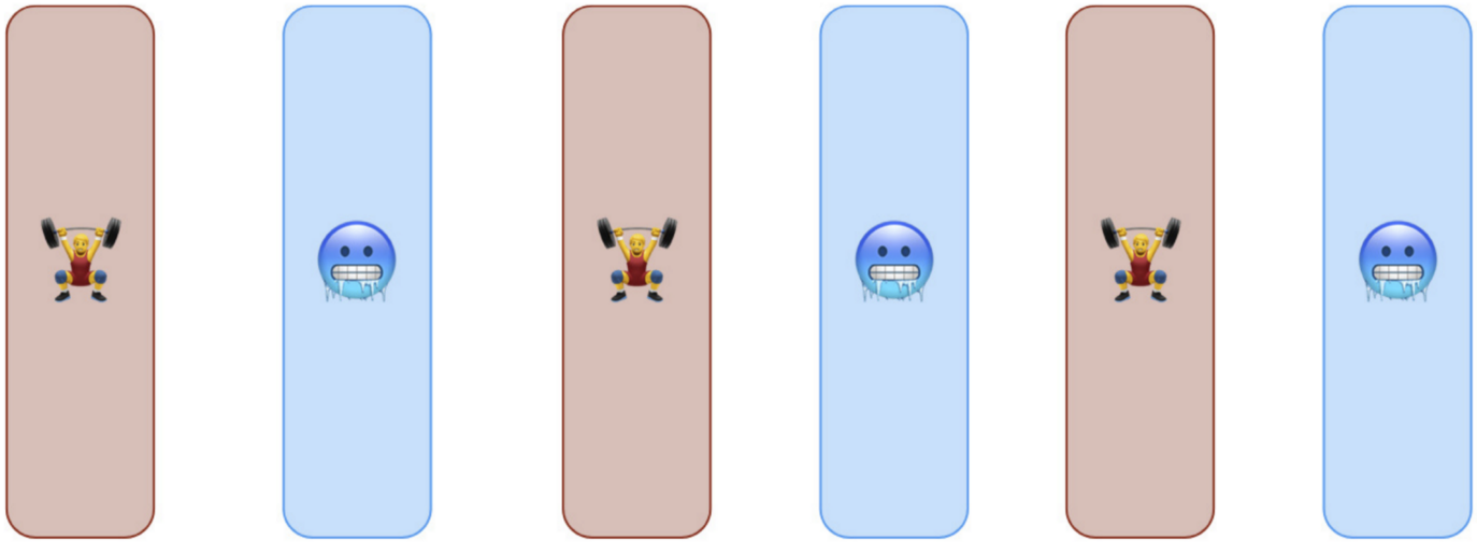
Un buon connubio tra Extreme machine learning e Natural language processing è presente nel lavoro **Reservoir Transformers** (Sheng, Kiela et al. [3]), nel quale vengono presentati modelli per il linguaggio basati sull'utilizzo dei transformers, nei quali alcune parti non sono allenate ma inizializzate randomicamente.

É stato utilizzato ROBERTA come Language Model, dal quale vengono derivati altri modelli che fanno uso di tecniche di EML.

- **Transformer Reservoir**, standard transformer layer ma con tutti i parametri fissati in modo casuale.
- **FFN Reservoir**, $FFN(LayerNorm(Previous\_Layer)) + Previous\_Layer$

Reservoir Transformers

Transformer Reservoir e FFN Reservoir vengono inseriti in modello BERT-like sostituendo alcuni layer del modello originale.



BackSkipping

Un altro fattore su cui possiamo risparmiare tempo è il calcolo del gradiente durante la fase di backpropagation nel layer randomici.

Logicamente, se un layer è inizializzato casualmente e mai allenato il suo apporto al gradiente rispetto al layer successivo non ha senso che venga calcolato in modo analitico tramite la chain rule

Il metodo di **Backskipping** si basa su tecniche di REINFORCEMENT LEARNING.

> *Backskipping is shown to be a promising approach to further reduce computational costs, and would be even more efficient from a hardware perspective since the circuitry for such layers (which do not need to propagate gradients) can be hardwired.*

Area Under the Convergence Curve

AUCC, misura le performance di un modello indipendentemente dalla capacità di computazione a disposizione.

$$\int_{t=0}^{\hat{T}} \sum_{x,y \in D} g_t(f(x), y),$$

con f il modello e g la metrica di valutazione, $\hat{T}$ rappresenta il tempo massimo di convergenza e non le iterazioni.

AUCC rappresenta un'approccio pratico allo sviluppo di metriche che tentano di valorizzare maggiormente l'efficienza dei modelli e non solo l'efficacia.

Reservoir Transformers: Risultati

Model	# Layers	Frozen	Max BLEU	Train time until max (in hours)	Ratio	# Params Trainable (Total)	Train Time each epoch (in hours)
Transformer	12	0	24.46 ± 0.04	15.15 ± 0.15	1	75.6M	0.505 ± 0.005
	16	0	24.52 ± 0.03	16.05 ± 0.18	1	88.2M	0.643 ± 0.006
	24	0	24.69 ± 0.05	17.61 ± 0.85	1	113.4M	0.877 ± 0.029
	32	0	24.83 ± 0.04	18.42 ± 0.28	1	138.6M	1.036 ± 0.010
T Reservoir	12	4	24.26 ± 0.08	14.11 ± 0.21	0.93	72.4M (75.6M)	0.472 ± 0.007
	16	4	24.50 ± 0.05	15.25 ± 0.28	0.95	75.6M (88.2M)	0.596 ± 0.009
	24	4	25.11 ± 0.07	15.89 ± 0.74	0.90	100.8M (113.4M)	0.776 ± 0.024
	32	4	24.66 ± 0.04	16.38 ± 0.24	0.88	126.0M (138.6M)	0.998 ± 0.009
FFN Reservoir	12	4	24.42 ± 0.05	14.01 ± 0.09	0.92	72.4M (71.4M)	0.441 ± 0.003
	16	4	24.65 ± 0.07	14.53 ± 0.17	0.91	75.6M (83.9M)	0.524 ± 0.006
	24	4	24.93 ± 0.04	12.62 ± 1.53	0.71	100.8M (109.2M)	0.743 ± 0.018
	32	4	24.98 ± 0.03	13.96 ± 0.19	0.73	126.0M (134.4M)	0.964 ± 0.007
LayerDrop	12	4	24.27 ± 0.03	14.61 ± 0.14	0.96	72.4M (75.6M)	0.489 ± 0.006
	16	4	24.15 ± 0.06	15.55 ± 0.54	0.97	75.6M (88.2M)	0.597 ± 0.017
	24	4	24.37 ± 0.05	16.25 ± 0.36	0.92	100.8M (113.4M)	0.823 ± 0.013
	32	4	23.84 ± 0.03	15.27 ± 0.38	0.83	126.0M (138.6M)	1.028 ± 0.012

Figure: Tabella 2 da [3]: Risultati di traduzione automatica sul dataset WMT.

Reservoir Transformers: Risultati (III)

Model	**Max BLEU**	**AUCC**	**Train time**
Transformer	34.59 ± 0.11	114.57 ± 0.08	142.28 ± 1.87
T Reservoir	34.80 ± 0.07	115.26 ± 0.26	134.49 ± 1.70
Backskip Reservoir	34.75 ± 0.05	115.99 ± 0.23	119.54 ± 1.78

Figure: Tabella 3 da [3]: Accuratezza, BLEU e Training Time sul dataset IWSLT.

Reservoir Transformers: Risultati (II)

Model	Layer	SentLen (Surface)	TreeDepth (Syntactic)	TopConst (Syntactic)	BShift (Syntactic)	Tense (Semantic)	SubjNum (Semantic)	ObjNum (Semantic)	SOMO (Semantic)	CoordInv (Semantic)
Transformer	1	84.56 ± 0.54	32.30 ± 0.41	54.40 ± 0.33	49.99 ± 0.01	80.98 ± 0.32	76.26 ± 0.09	50.01 ± 0.19	76.38 ± 0.61	54.33 ± 0.47
	2	87.22 ± 0.07	33.63 ± 0.57	58.38 ± 0.20	50.12 ± 0.17	82.84 ± 0.68	78.65 ± 0.19	51.47 ± 0.53	78.00 ± 1.12	54.66 ± 0.55
	3	84.25 ± 0.16	32.60 ± 0.17	54.41 ± 0.10	50.02 ± 0.01	81.72 ± 0.59	77.00 ± 0.13	51.32 ± 0.64	76.57 ± 1.13	54.13 ± 0.51
	4	87.37 ± 0.20	32.59 ± 0.29	50.06 ± 0.21	69.76 ± 0.26	81.63 ± 1.17	76.47 ± 0.09	52.41 ± 1.49	76.15 ± 0.84	52.62 ± 1.34
	5	84.61 ± 0.24	31.14 ± 0.48	44.76 ± 0.38	74.82 ± 0.11	80.16 ± 0.19	73.66 ± 0.16	52.95 ± 1.77	72.90 ± 0.21	51.26 ± 1.14
	6	82.56 ± 0.25	30.31 ± 0.40	39.30 ± 0.40	78.80 ± 0.38	81.88 ± 0.47	75.30 ± 0.07	56.21 ± 1.26	74.37 ± 0.16	51.44 ± 1.04
	7	70.85 ± 0.13	26.65 ± 0.72	40.70 ± 0.13	78.98 ± 0.32	85.11 ± 0.31	72.03 ± 0.46	58.15 ± 0.46	68.71 ± 0.91	55.39 ± 0.27
	8	66.23 ± 1.33	23.46 ± 0.44	25.19 ± 1.02	77.42 ± 0.27	80.35 ± 0.45	67.55 ± 0.99	54.94 ± 2.04	63.69 ± 2.32	50.58 ± 0.83
	9	71.17 ± 0.29	31.21 ± 0.31	58.42 ± 0.29	85.55 ± 0.44	86.77 ± 0.19	80.30 ± 0.08	64.36 ± 1.20	81.68 ± 0.45	66.90 ± 0.49
	10	73.19 ± 0.50	27.74 ± 0.53	41.01 ± 0.22	83.56 ± 0.96	86.13 ± 0.35	83.04 ± 0.04	62.01 ± 0.59	79.73 ± 0.21	62.60 ± 1.04
	11	71.37 ± 0.42	30.22 ± 0.28	48.58 ± 0.35	84.40 ± 0.44	87.28 ± 0.59	82.34 ± 0.15	61.10 ± 0.14	80.00 ± 0.40	64.44 ± 0.38
	12	71.66 ± 0.12	33.43 ± 0.18	64.38 ± 0.20	87.38 ± 0.02	88.41 ± 0.09	84.46 ± 0.25	63.01 ± 0.05	81.80 ± 0.27	65.72 ± 0.16
T Reservoir	1	87.75 ± 0.10	31.60 ± 0.21	50.38 ± 0.23	50.00 ± 0.00	80.40 ± 0.18	76.47 ± 0.20	50.53 ± 0.14	73.48 ± 0.15	53.55 ± 0.70
	2	81.28 ± 0.23	34.20 ± 0.41	61.41 ± 0.42	60.64 ± 0.65	81.50 ± 0.77	76.33 ± 0.08	50.73 ± 0.34	74.28 ± 0.67	56.82 ± 0.10
	3	**89.28 ± 0.09**	**36.42 ± 0.11**	**67.36 ± 0.45**	**75.64 ± 0.52**	**85.42 ± 0.18**	**80.53 ± 0.02**	**52.50 ± 1.80**	**78.47 ± 1.81**	**57.16 ± 0.27**
	4	74.31 ± 0.32	32.42 ± 0.83	55.19 ± 0.33	73.41 ± 0.00	79.56 ± 0.00	75.15 ± 0.08	53.68 ± 0.66	75.02 ± 0.19	56.89 ± 0.08
	5	**88.03 ± 0.22**	**38.34 ± 0.64**	**68.65 ± 0.29**	**82.25 ± 0.12**	**86.80 ± 0.02**	**82.27 ± 0.33**	**57.95 ± 0.24**	**80.82 ± 0.91**	**58.05 ± 0.10**
	6	74.55 ± 0.37	33.13 ± 0.29	52.70 ± 0.81	79.21 ± 0.13	85.70 ± 0.36	77.43 ± 0.03	57.26 ± 0.19	75.38 ± 0.66	51.95 ± 1.30
	7	**85.82 ± 0.37**	**37.63 ± 0.13**	**70.43 ± 0.05**	**84.12 ± 0.35**	**86.88 ± 0.07**	**82.86 ± 0.30**	**61.17 ± 0.21**	**80.79 ± 0.17**	**61.83 ± 0.95**
	8	71.69 ± 0.71	30.32 ± 0.01	48.44 ± 0.30	79.12 ± 0.12	84.75 ± 0.09	79.23 ± 0.11	59.53 ± 0.16	76.80 ± 0.41	57.34 ± 0.14
	9	**85.86 ± 0.12**	**37.89 ± 0.03**	**69.53 ± 0.37**	**85.55 ± 0.12**	**87.98 ± 0.22**	**84.13 ± 0.01**	**63.06 ± 0.01**	**82.55 ± 0.31**	**66.07 ± 0.05**
	10	69.22 ± 0.23	25.58 ± 0.35	29.20 ± 0.58	78.57 ± 0.09	85.02 ± 0.03	75.68 ± 0.16	57.55 ± 1.57	74.70 ± 0.02	55.02 ± 0.64
	11	65.70 ± 0.05	30.57 ± 0.03	47.56 ± 0.02	81.20 ± 0.00	86.78 ± 0.02	83.73 ± 0.05	60.38 ± 0.17	80.59 ± 0.15	62.50 ± 0.11
	12	70.61 ± 0.18	34.45 ± 0.20	64.19 ± 0.10	84.53 ± 0.03	87.48 ± 0.16	84.86 ± 0.14	62.75 ± 0.14	82.08 ± 0.03	64.73 ± 0.06

Figure: Tabella 4 da [3]: Risultati di Probing.

Reservoir Transformers: Considerazioni

Il lavoro "Reservoir Transformers" (Shen, Kiela et al. [3]) ha indubbiamente dei contro, utilizzare dei metodi randomici per il linguaggio è controintuitivo rispetto all'andamento degli ultimi anni volto allo sviluppo di metodi sempre più explainable e con capacità di attenzione volte a farci comprendere le logiche di comprensione dei modelli stessi.

É molto interessante notare come sono proprio i layer randomici quelli a performare meglio nei task di probing.

Sembra dunque che in Meta (ex Facebook) non ci sia la volontà di continuare con tali esperimenti, nonostante non sia il primo lavoro di Kiela nell'ambito della randomness nei modelli per il linguaggio prodotto dai ricercatori di Meta.

Kiela attualmente lavora per huggingface.

Bibliography

[1] Emily M Bender et al. "On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?" In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 2021, pp. 610–623.

[2] Roy Schwartz et al. "Green ai". In: *Communications of the ACM* 63.12 (2020), pp. 54–63.

[3] Sheng Shen et al. "Reservoir transformers". In: *arXiv preprint arXiv:2012.15045* (2020).

[4] Emma Strubell, Ananya Ganesh, and Andrew McCallum. "Energy and policy considerations for deep learning in NLP". In: *arXiv preprint arXiv:1906.02243* (2019).

[5] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).