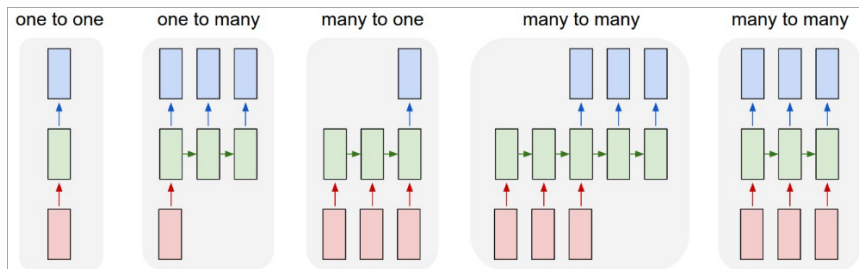


# Reti Neurali Ricorrenti

Costantino Carugno

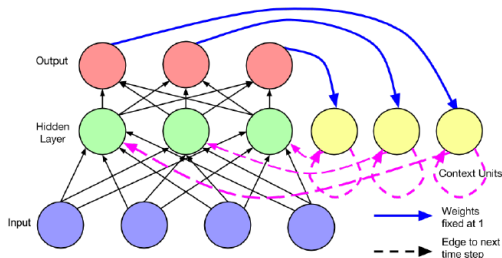
22 Maggio 2018

- Innumerevoli task di apprendimento hanno bisogno di gestire dati sequenziali o serie temporali.
- Le Reti Neurali Ricorrenti (RNN) sono modelli di connessione che catturano la dinamica delle sequenze tramite cicli nella rete dei nodi. A differenza delle reti neurali feed-forward, le reti ricorrenti utilizzano un circuito di feedback collegato alle loro decisioni passate, ingerendo le loro uscite momento dopo momento come input o in alternativa si può dire che utilizzano un vettore di stato che può rappresentare informazione da una finestra di contesto (arbitrariamente lunga).
- Le RNN hanno quindi memoria. L'aggiunta di memoria alle reti neurali ha uno scopo: ci sono informazioni nella sequenza stessa e le reti ricorrenti la usano per eseguire attività che le reti feedforward non possono svolgere.
- È stato dimostrato recentemente che sono un modello Turing-completo, possono eseguire algoritmi arbitrari (1995, Siegelmann).

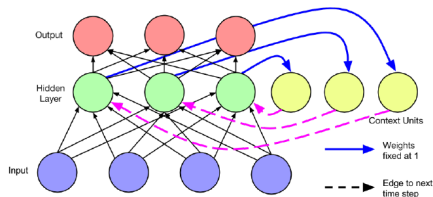


- Rete tradizionale: input vettore di dimensione fissata, output idem (classificazione immagini)
- Output sequenziale: Assegnare didascalie descrittive alle immagini (captioning)
- Input sequenziale: Sentiment analysis (frase in ingresso, valore in uscita)
- I/O sequenziale: Traduzione linguistica
- I/O sequenziale sincronizzato: Classificazione frame video

- La ricerca fondante sulle reti ricorrenti è avvenuta negli anni '80 ed è stata ispirata dal comportamento umano della memoria: proprio come la memoria umana circola invisibilmente all'interno di un corpo, influenzando il nostro comportamento senza rivelarne la piena forma, l'informazione circola negli stati nascosti delle reti ricorrenti.
- Nel 1982, Hopfield introduce una famiglia di reti neurali che avevano capacità di riconoscimento di pattern. Definiti dei valori di pesi simmetrici tra nodi completamente connessi un pattern viene memorizzato aggiornando lo stato dei nodi. Le reti di Hopfield sono utili per recuperare un pattern memorizzato da una versione corrotta e sono precursori delle macchine di Boltzmann e degli autoencoder.
- Una prima architettura di apprendimento supervisionato su sequenze è stata introdotta da Jordan nel 1986. Questo modello è una rete feedforward con un singolo strato nascosto che viene esteso con unità speciali, chiamate unità di stati o di contesto.

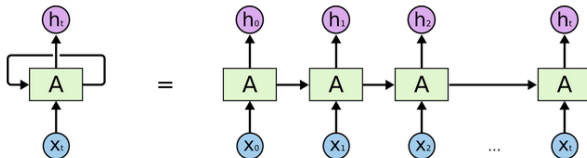


- Le unità speciali di una rete di Jordan sono autoconnesse. Intuitivamente, queste connessioni permettono di mandare informazione attraverso passi temporali multipli senza perturbare l'uscita ad ogni passo temporale intermedio.
- I valori dei nodi di uscita vengono dati in ingresso alle unità di stati, che poi restituiscono questi valori ai nodi nascosti al successivo passo temporale. Se i valori di uscita sono azioni, le unità di stato permettono alla rete di ricordare le azioni prese ai passi temporali precedenti.



- Elman nel 1990 introduce un modello più semplice. Associata con ogni unità dello strato nascosto c'è un'unità di stato. Ognuna di queste  $j'$  prende in ingresso lo stato del corrispettivo nodo  $j$  nascosto al tempo precedente, lungo una connessione di peso fissato  $w_{j'j} = 1$ . Questo valore viene fornito indietro allo stesso nodo nascosto  $j$  lungo una connessione tradizionale.
- Questa architettura è equivalente a una semplice rete ricorrente in cui ogni nodo nascosto ha una singola autoconnessione ricorrente. L'idea di connessioni ricorrenti con peso fissato che rendono i nodi nascosti autoconnessi è fondamentale nei lavori successivi sulle reti LSTM Hochreiter e Schmidhuber 1997.
- Elman allena la rete usando la backpropagation e dimostra che la rete può apprendere le dipendenze temporali (1990).

## Dipendenze a lungo termine



Una rete neurale ricorrente srotolata.

†

- Una rete neurale ricorrente può essere pensata come più copie della stessa rete, ciascuna che trasmette un messaggio a un successore.
- Le informazioni sequenziali sono conservate nello stato nascosto della rete ricorrente, che riesce a coprire molti passaggi temporali mentre procede a cascata in avanti per influenzare l'elaborazione di ogni nuovo esempio. Le correlazioni tra eventi separati da molti passi temporali sono chiamate “dipendenze a lungo termine”, perché un evento a valle nel tempo dipende da uno o più eventi precedenti.

$$h^{(t)} = \sigma(W^{hx}x^{(t)} + W^{hh}h^{(t-1)}) \quad (1)$$

- Lo stato nascosto al momento in cui passo  $t$  è  $h^{(t)}$ . È una funzione dell'input allo stesso passo temporale  $x^{(t)}$ , modificato da una matrice di peso  $W^{hx}$  con in aggiunta un termine che connette lo stato nascosto del passo temporale precedente con se stesso  $h^{(t-1)}$  moltiplicato per la matrice dello stato nascosto  $W^{hh}$ , simile alla matrice di transizione di una catena Markov. Le matrici di peso sono filtri che determinano l'importanza da assegnare sia per l'input presente che per lo stato nascosto passato.
- Poiché questo ciclo di feedback si verifica in ogni fase della serie temporale, ogni stato nascosto contiene tracce non solo dello stato nascosto precedente, ma anche di tutti quelli che lo hanno preceduto  $h^{(t-1)}$  per tutto il tempo in cui la memoria può persistere.
- La dinamica di una rete ricorrente attraverso i diversi passi temporali può essere visualizzata con il dispiegamento (unfolding). Con questa rappresentazione, la rete può essere interpretata non come ciclica, ma piuttosto come una rete neurale profonda con uno strato ad ogni passo temporale e pesi condivisi attraverso diversi passi temporali. È chiaro quindi che la rete dispiegata può essere allenata attraverso passi temporali utilizzando la backpropagation. Questo algoritmo, chiamato backpropagation attraverso il tempo (BPTT) è stato introdotto da Werbos nel 1990

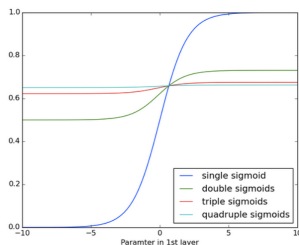


$$\delta_k = \frac{\partial \mathcal{L}(\hat{y}_k, y_k)}{\partial \hat{y}_k} \cdot l'_k(h_k) \quad (2)$$

$$\delta_j = l'(a_j) \sum_k \delta_k \cdot w_{kj} \quad (3)$$

- Le reti ricorrenti si basano su un'estensione della backpropagation chiamata backpropagation nel tempo (o BPTT). Il tempo, in questo caso, è semplicemente espresso da una sequenza ben definita di calcoli che collega un passaggio temporale a un altro. Le reti neurali, siano esse ricorrenti o meno, sono semplicemente funzioni composite annidate come  $f(g(h(x)))$ . L'aggiunta di un elemento temporale estende la sequenza di funzioni per le quali calcoliamo le derivate con la regola della catena.
- L'apprendimento nelle reti ricorrenti è da sempre considerato difficile. Già nelle reti feedforward standard, il task di ottimizzazione è un problema NP-completo, ma l'apprendimento con le reti ricorrenti può risultare particolarmente impegnativo a causa della difficoltà dell'apprendimento delle dipendenze a lungo termine.

## Gradienti che esplodono o svaniscono



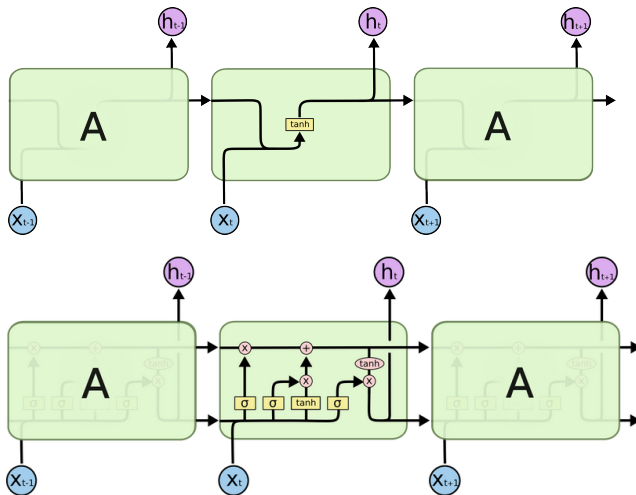
- Rete con: singolo nodo di ingresso, singola uscita, e nodo ricorrente nascosto. Consideriamo un input passato alla rete al tempo  $\tau$  e un errore calcolato al tempo  $t$ , assumendo un input zero nei passi temporali intercorrenti. Il legame dei pesi attraverso i passi temporali fanno sì che la connessione ricorrente al nodo nascosto  $j$  abbia sempre lo stesso peso. Perciò, il contributo dell'input al tempo  $\tau$  all'output al tempo  $t$  dovrà esplodere o approssimare zero, con velocità esponenziale, quanto rapidamente cresce  $t - \tau$ . Da qui segue che la derivata dell'errore rispetto all'input esplode o svanisce. Quale dei due fenomeni occorre dipende dal segno del peso della connessione ricorrente  $|w_{jj}| > 1$  o  $|w_{jj}| < 1$  e dalla funzione di attivazione nel nodo nascosto. Con una funzione di attivazione sigmoide, il problema della scomparsa del gradiente è più preminente, mentre con una funzione di attivazione ReLU, il problema di esplosione è più dominante.

## Gradienti che esplodono o svaniscono

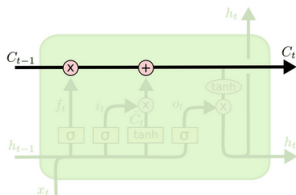
- Pascanu (et al) nel 2012 hanno fornito un trattamento matematico completo dei problemi di esplosione e scomparsa del gradiente, caratterizzando le condizioni esatte sotto le quali questi problemi accadono. Date queste condizioni, suggeriscono un approccio di apprendimento tramite un termine di regolarizzazione che forza i pesi a valori dove il gradiente non svanisce e non esplode.
- La backpropagation attraverso il tempo troncata è una soluzione al problema di esplosione del gradiente per reti continuamente (da controllare). Con TBPTT, viene stabilito un certo numero massimo di passi temporali attraverso cui l'errore può essere propagato. Mentre la TBPTT con un piccolo cutoff può essere usata per alleviare il problema di esplosione del gradiente, richiede il sacrificio dell'abilità di imparare dipendenze a lungo raggio.
- La questione dei minimi locali rimane un ostacolo all'allenamento effettivo che non può essere affrontato semplicemente modificando l'architettura della rete, ad ogni modo studi empirici e teorici suggeriscono che nella pratica, questa questione non siano così importante come si pensava un tempo.

- L'architettura LSTM usa nodi progettati attentamente con connessioni ricorrenti con peso di unità fissato come soluzione al problema di scomparsa del gradiente.
- Nel 1996, prima dell'introduzione delle LSTM, i tentativi di allenare le reti ricorrenti per collegare lunghi buchi temporali non performavano meglio di tirare a indovinare
- Hochreiter e Schmidhuber 1997 introdussero il modello LSTM primariamente per superare il problema dei gradienti che svaniscono. Questo modello ricorda una rete neurale ricorrente standard con uno strato nascosto, ma ogni nodo ordinario nello strato nascosto viene sostituito da una cella di memoria. Ogni cella di memoria contiene una autoconnessione ricorrente di peso fisso uguale a 1, garantendo che il gradiente possa passare attraverso tanti passi temporali senza svanire o esplodere.
- Il termine “Long Short-Term Memory” proviene dalla seguente intuizione. Le reti ricorrenti semplici hanno memoria a lungo termine nella forma di pesi. I pesi cambiano lentamente durante l'apprendimento, codificando le conoscenze generali riguardo i dati. I pesi hanno anche una memoria a breve termine, che passa da ogni nodo al successivo. Il modello LSTM introduce un tipo intermedio di storage utilizzando le celle di memoria, ovvero un'unità composita, costruita da nodi semplici usando uno specifico percorso connettivo, con l'innovativa inclusione di nodi moltiplicativi.

# Long-Short Term Memory

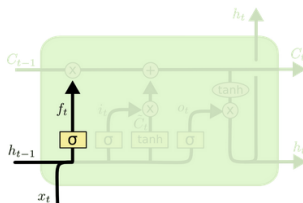


Il modulo ripetitivo in un LSTM contiene quattro livelli interagenti.



- L'idea alla base dell'LSTM è nodo con attivazione lineare, che viene definito nel paper originale come **stato interno** della cella. Lo stato interno ha un'auto-connessione ricorrente con una unità di peso unitario fissato. Poiché questa connessione abbraccia passi temporali adiacenti con peso costante, l'errore può scorrere attraverso tanti passi temporali senza svanire o esplodere.
- L'LSTM ha la capacità di rimuovere o aggiungere informazioni allo stato della cella, regolato con cura dalle strutture chiamate gate. Un gate è un'unità sigmoideale che, viene attivata dal punto dati corrente  $x^{(t)}$  così come dallo stato nascosto al passo temporale precedente  $h_{t-1}$ . Un gate è chiamato così perché il suo valore viene usato per moltiplicare il valore di un altro nodo. È un gate nel senso che se il suo valore è zero, allora il flusso dall'altro nodo viene tagliato fuori. Se il valore del gate è uno, tutto il flusso viene lasciato passare.

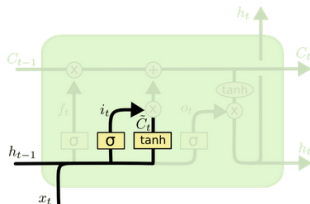
## Forget gate



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

- Un LSTM ha tre di questi gate, per proteggere e controllare lo stato della cella.
- Il primo passo è decidere quali informazioni verranno eliminate dallo stato della cella. Questa decisione è presa da uno strato neurale chiamato “forget gate layer”. Controlla  $h_{t-1}$  e  $x_t$ , e restituisce un numero tra 0 e 1 per ogni numero nello stato della cella  $C_{t-1}$ . Un 1 significa “mantieni completamente” mentre uno 0 “dimentica l’informazione”.
- Questi gate  $f_c$  sono stati introdotti da Gers et al. nel 2000 e sono utili per le reti continuously running.

# Input gate

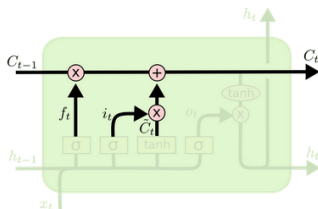


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

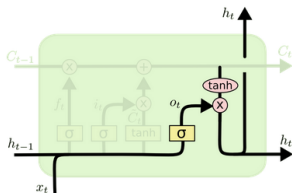
- Il gate di input ha il ruolo di decidere quali nuove informazioni verranno memorizzate nello stato della cella, a partire dal nodo di input e dal punto dati corrente.
- Si compone di due parti: un layer sigmoide che decide quali valori aggiorneremo combinato con un layer a  $\tanh$  che produce un vettore di nuovi valori candidati,  $\tilde{C}_t$ , che potrebbe essere aggiunto allo stato. Nel prossimo passaggio, combineremo questi due elementi per creare un aggiornamento allo stato.





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

- La memorizzazione avviene tramite l'aggiornamento del vecchio stato della cella,  $C_{t-1}$ , nel nuovo stato della cella  $C_t$ . I passi precedenti hanno già deciso cosa fare, abbiamo solo bisogno di farlo davvero.
- Moltiplichiamo il vecchio stato per  $f_t$ , dimenticando le cose che abbiamo deciso di dimenticare prima. Quindi aggiungiamo  $i_t * \tilde{C}_t$ . Questo è il nuovo valore candidato, scalato da quanto abbiamo deciso di aggiornare ogni valore di stato.

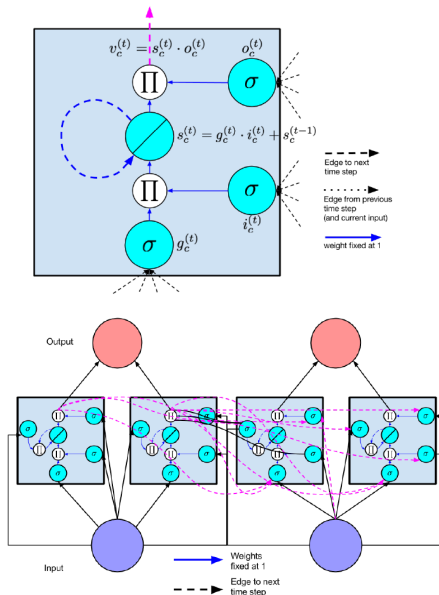


$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

- Infine, dobbiamo decidere cosa produrremo. Questo output sarà basato sul nostro stato della cella, ma sarà una versione filtrata. Innanzitutto, eseguiamo un layer sigmoide che decide quali parti dello stato della cella stiamo per produrre. Quindi, inseriamo lo stato della cella attraverso  $\tanh$  (per forzare i valori tra  $-1$  e  $1$ ) e lo moltiplichiamo per l'output del gate sigmoide, in modo che produrremo solo le parti che noi abbiamo deciso.

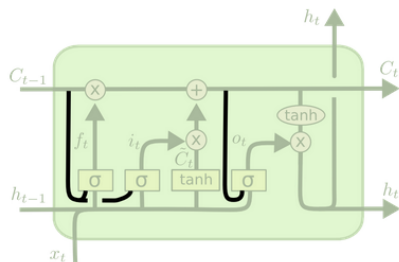
# LSTM completa unfolded



$$\begin{aligned}g^{(t)} &= \phi(W^{gx}x^{(t)} + W^{gh}h^{(t-1)} + b_g) \\i^{(t)} &= \sigma(W^{ix}x^{(t)} + W^{ih}h^{(t-1)} + b_i) \\f^{(t)} &= \sigma(W^{fx}x^{(t)} + W^{fh}h^{(t-1)} + b_f) \\o^{(t)} &= \sigma(W^{ox}x^{(t)} + W^{oh}h^{(t-1)} + b_o) \\s^{(t)} &= g^{(t)} \odot i^{(t)} + s^{(t-1)} \odot f^{(t)} \\h^{(t)} &= \phi(s^{(t)}) \odot o^{(t)}\end{aligned}\tag{4}$$

- È consuetudine che lo stato interno venga prima passato attraverso una funzione di attivazione *tanh*, così che questo dia all'output di ogni cella lo stesso range dinamico di un'unità nascosta ordinaria a tangente iperbolica. Ad ogni modo, come mostrato da altre ricerche su rete neurali di questo tipo, le unità ReLU, che hanno un range dinamico più ampio, sono più facili da allenare. Perciò sembra plausibile che la funzione non lineare sullo stato interno possa essere omessa.
- Da quando le prime unità LSTM furono introdotte, molte variazioni sono state proposte. I forget gate, descritti sopra, furono proposti nel 2000 e non erano parte dell'iniziale progettazione (design) delle LSTM originali. Ad ogni modo, si sono dimostrati efficaci e sono uno standard nelle implementazioni moderne.

## Peephole - Alternative a LSTM

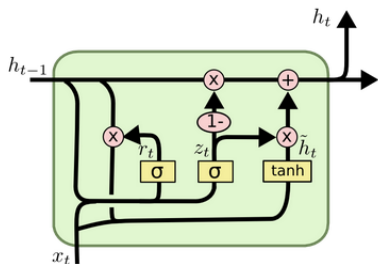


$$f_t = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i)$$

$$o_t = \sigma(W_o \cdot [C_t, h_{t-1}, x_t] + b_o)$$

- Lo stesso anno Gers e Schmidhuber hanno proposto le connessioni peephole (“spioncino”) che passano dallo stato interno direttamente ai gate di input e di output di quello stesso nodo senza prima essere stati modulati dal gate di output. Nel loro studio, riportano che queste connessioni hanno migliorato le performance su task di timing dove la rete deve imparare a misurare intervalli precisi tra eventi.



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

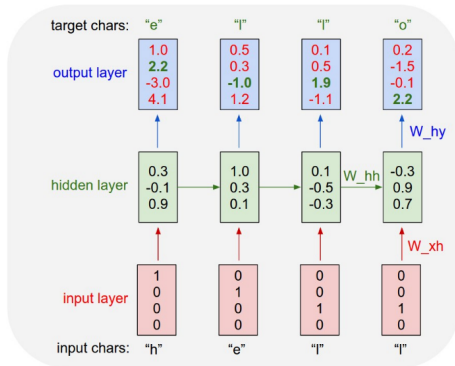
- Introdotta da Cho nel 2014, la Gated Recurrent Unit (GRU) è fondamentalmente una LSTM senza un gate di output, che quindi scrive completamente il contenuto dalla sua cella di memoria alla rete ad ogni passo temporale.
- Inoltre combina il forget e l'input gate in un “update gate” e unisce anche lo stato della cella con stato nascosto, apportando alcune modifiche.

- È importante notare che le celle di memoria di LSTM assegnano ruoli diversi all'addizione e alla moltiplicazione nella trasformazione dell'input. Il segno più centrale è essenzialmente il segreto di LSTM. Stupidamente semplice come potrebbe sembrare, questo li aiuta a mantenere un errore costante quando deve essere backpropagato in profondità. Invece di determinare lo stato della cella successiva moltiplicando il suo stato corrente con il nuovo input, aggiungono i due, e questo fa letteralmente la differenza.
- I gate imparano quando lasciare entrare errore e quando lasciarlo uscire, questo permette al gradiente di propagarsi indietro attraverso tanti passi temporali, senza che esploda o svanisca nella backpropagation.
- Jozefowicz et al. (2015) hanno testato più di diecimila architetture RNN, trovando alcune che funzionavano meglio delle LSTM su determinate attività. Hanno dimostrato euristicamente che LSTM classiche e GRU hanno le stesse performance, sebbene le GRU abbiano meno pesi da allenare e siano quindi più convenienti computazionalmente.

- Molte applicazioni delle RNNs trattano l'elaborazione del linguaggio scritto. Generalmente l'output consiste di un vettore softmax  $y^{(t)} \in \mathbb{R}^K$ , dove  $K$  è la dimensione del vocabolario. Gli output corrispondono alle probabilità che ogni parola sia l'output corretto ad ogni passo temporale.
- Per le applicazioni dove l'ingresso consiste di una sequenza di parole, tipicamente le parole sono fornite alla rete una alla volta in passi temporali consecutivi. In questi casi, il modo più semplice per rappresentare le parole è la codifica “one-hot”, usando vettori binari con una lunghezza uguale alla dimensione del vocabolario (proposta da Elman nel 1990).
- Ad ogni modo, questa codifica è inefficiente, poiché richiede tanti bit quanto la dimensione del vocabolario. Inoltre, non offre alcuna via diretta alla cattura dei diversi aspetti di analogie tra le parole nella codifica stessa.
- È uso comune modellizzare le parole con una rappresentazione distribuita, usando un “vettore significato” (meaning vector). In alcuni casi, i significati delle parole sono essi stessi appresi tramite una grande raccolta di dati supervisionati, ma è meglio inizializzare i vettori significato usando un'immersione (embedding) basata su statistiche di co-occorrenza delle parole.
- Codice disponibile pubblicamente per produrre vettori di parole da queste statistiche includono GloVe (sviluppato da Stanford), e word2vec.



## Generazione carattere per carattere



- In molti esperimenti con reti ricorrenti (Elman, 1990 Sutskever et al 2011, Zaremba e Sutskever 2014), l'input è fornito un carattere alla volta, e l'output è generato un carattere alla volta, invece di una parola alla volta.
- Questa idea è stata esplorata in dettaglio da Karpathy (2015) nell'articolo: "L'irragionevole efficacia delle reti neurali ricorrenti", in cui mostra applicazioni su testi di Shakespeare, Wikipedia, codice sorgente etc.

- Idea: Una rete neurale ricorrente allenata sulla Bibbia.
- Come funzione di errore si può usare la Cross Entropy:

$$E_{CE} = - \sum_{\mu, i} d_i^{\mu} \ln(y_i^{\mu} / d_i^{\mu})$$

- Come funzione di minimizzazione dell'errore utilizzo Adam, “adaptive moment estimation” un'estensione della discesa del gradiente stocastica, presentata nel 2015, che unisce i vantaggi di due altre estensioni della discesa del gradiente stocastico ovvero: Adaptive Gradient Algorithm (AdaGrad) che mantiene un tasso di apprendimento per parametro che migliora le prestazioni in caso di problemi con gradienti sparsi e RMSProp, che mantiene anche i tassi di apprendimento che vengono adattati in base alla media delle grandezze recenti dei gradienti per il peso (quanto velocemente sta cambiando). Ciò significa che l'algoritmo funziona bene su problemi online e non stazionari (ad es. Rumoroso). Adam, invece di adattare i tassi di apprendimento dei parametri in base al primo momento medio (la media) come in RMSProp, fa anche uso della media dei secondi momenti dei gradienti (la varianza non centrata).

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t & \theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t. \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \end{aligned}$$

- Come cella di memoria usiamo una GRU nella quale immagazzino un carattere alla volta con un encoding one-hot, ottengo un alfabeto di  $\approx 100$  simboli fra maiuscole, minuscole e punteggiatura.
- Il dataset a nostra disposizione è relativamente piccolo,  $\approx 5.5MB$  il che significa che il rischio di overfitting è grande. Per ovviare al problema di memorizzare troppo correttamente la Bibbia e non essere in grado di generare del testo originale utilizziamo un metodo di regolarizzazione, il dropout, che fa fuori il 20% delle cellule ogni ciclo di training.
- Il dropout può essere applicato sia agli input che agli output di un layer denso e questo non fa molta differenza. Osservando la matrice dei pesi di uno strato denso vedo che applicare il dropout agli input equivale a far cadere le linee nella matrice dei pesi mentre applicare il dropout agli output equivale a rimuovere le colonne. Negli RNN è consuetudine aggiungere il dropout agli input in tutti i layer della cella e l'output dell'ultimo layer, che in realtà serve come dropout di input del layer softmax, quindi non è necessario aggiungerlo esplicitamente.
- Il dropout dovrebbe essere applicato agli input e agli output RNN ma non agli stati. In questo approccio, una maschera di esclusione casuale viene ricalcolata ad ogni passo della sequenza srotolata. Questo approccio è chiamato "naive dropout" ed è quello implementato in questo esperimento.

```
train.py
4 import tensorflow as tf
3 from tensorflow.contrib import layers
2 from tensorflow.contrib import rnn # rnn stuff temporarily in contrib, moving back to code in TF 1.1
1 import os
5 import time
1 import math
2 import numpy as np
3 import my_txtutils as txt
4 tf.set_random_seed(0)
5
6 SEQLen = 200
7 BATCHSIZE = 80
8 ALPHASIZE = txt.ALPHASIZE
9 INTERNALSIZE = 512
10 NLAYERS = 3
11 learning_rate = 0.001 # fixed learning rate
12 dropout_pkeep = 0.8 # some dropout
13
14 bibbiadir = "bibbia/*.txt"
15 codetext, valitext, bookranges = txt.read_data_files(shkdir, validation=True)
16
17 # display some stats on the data
18 epoch_size = len(codetext) // (BATCHSIZE * SEQLen)
19 txt.print_data_stats(len(codetext), len(valitext), epoch_size)
20
21 lr = tf.placeholder(tf.float32, name='lr') # learning rate
22 pkeep = tf.placeholder(tf.float32, name='pkeep') # dropout parameter
23 batchsize = tf.placeholder(tf.int32, name='batchsize')
24
25 # inputs
26 X = tf.placeholder(tf.uint8, [None, None], name='X') # [ BATCHSIZE, SEQLen ]
27 Xo = tf.one_hot(X, ALPHASIZE, 1.0, 0.0) # [ BATCHSIZE, SEQLen, ALPHASIZE ]
28 # expected outputs = same sequence shifted by 1 since we are trying to predict the next character
29 Y_ = tf.placeholder(tf.uint8, [None, None], name='Y_') # [ BATCHSIZE, SEQLen ]
30 Yo = tf.one_hot(Y_, ALPHASIZE, 1.0, 0.0) # [ BATCHSIZE, SEQLen, ALPHASIZE ]
31 # input state
32 Hin = tf.placeholder(tf.float32, [None, INTERNALSIZE*NLAYERS], name='Hin') # [ BATCHSIZE, INTERNALSIZE * NLAYERS ]
33
34 # using a NLAYERS=3 layers of GRU cells, unrolled SEQLen=30 times
35 # dynamic_rnn infers SEQLen from the size of the inputs Xo
36
37 # How to properly apply dropout in RNNs: see README.md
38 cells = [rnn.GRUCell(INTERNALSIZE) for _ in range(NLAYERS)]
39 # "naive dropout" implementation
40 NORMAL train.py
"train.py" 179L, 8772C
```

python

```
train.py
38 cells = [rnn.GRUCell(INTERNALSIZE) for _ in range(NLAYERS)]
37 # "naive dropout" implementation
36 dropcells = [rnn.DropoutWrapper(cell, input_keep_prob=pkeep) for cell in cells]
35 multicell = rnn.MultiRNNCell(dropcells, state_is_tuple=False)
34 multicell = rnn.DropoutWrapper(multicell, output_keep_prob=pkeep) # dropout for the softmax layer
33
32 Yr, H = tf.nn.dynamic_rnn(multicell, Xo, dtype=tf.float32, initial_state=Hin)
31 # Yr: [ BATCHSIZE, SEQLen, INTERNALSIZE ]
30 # H: [ BATCHSIZE, INTERNALSIZE*NLAYERS ] # this is the last state in the sequence
29
28 H = tf.identity(H, name='H') # just to give it a name
27
26 # Softmax layer implementation:
25 # Flatten the first two dimension of the output [ BATCHSIZE, SEQLen, ALPHASIZE ] => [ BATCHSIZE x SEQLen, ALPHASIZE ]
24 # then apply softmax readout layer. This way, the weights and biases are shared across unrolled time steps.
23 # From the readout point of view, a value coming from a sequence time step or a minibatch item is the same thing.
22
21 Yflat = tf.reshape(Yr, [-1, INTERNALSIZE]) # [ BATCHSIZE x SEQLen, INTERNALSIZE ]
20 Ylogits = layers.linear(Yflat, ALPHASIZE) # [ BATCHSIZE x SEQLen, ALPHASIZE ]
19 Yflat_ = tf.reshape(Yo, [-1, ALPHASIZE]) # [ BATCHSIZE x SEQLen, ALPHASIZE ]
18 loss = tf.nn.softmax_cross_entropy_with_logits(logits=Ylogits, labels=Yflat_) # [ BATCHSIZE x SEQLen ]
17 loss = tf.reshape(loss, [batchsize, -1]) # [ BATCHSIZE, SEQLen ]
16 Yo = tf.nn.softmax(Ylogits, name='Yo') # [ BATCHSIZE x SEQLen, ALPHASIZE ]
15 Y = tf.argmax(Yo, 1) # [ BATCHSIZE x SEQLen ]
14 Y = tf.reshape(Y, [batchsize, -1], name='Y') # [ BATCHSIZE, SEQLen ]
13 train_step = tf.train.AdamOptimizer(lr).minimize(loss)
12
11 # stats for display
10 seqloss = tf.reduce_mean(loss, 1)
9 batchloss = tf.reduce_mean(seqloss)
8 accuracy = tf.reduce_mean(tf.cast(tf.equal(Y, tf.cast(Y, tf.uint8)), tf.float32))
7 loss_summary = tf.summary.scalar("batch_loss", batchloss)
6 acc_summary = tf.summary.scalar("batch_accuracy", accuracy)
5 summaries = tf.summary.merge([loss_summary, acc_summary])
4
3 # Init Tensorboard stuff. This will save Tensorboard information into a different
2 # folder at each run named 'log/<timestamp>/'. Two sets of data are saved so that
1 # you can compare training and validation curves visually in Tensorboard.
81 timestamp = str(math.trunc(time.time()))
1 summary_writer = tf.summary.FileWriter("log/" + timestamp + "-training")
2 validation_writer = tf.summary.FileWriter("log/" + timestamp + "-validation")
3
4 # Init for saving models. They will be saved into a directory named 'checkpoints'.
5 # Only the last checkpoint is kept.
```

NORMAL train.py

python

0%	Training on next 50 batches	100%
1600000 (epoch 0) bibbia3.txt	rni e quaranta notti. [13] In quello stesso giorno entr	io e eee t ee de ee [ ] io teo o eee ee ee ee e ee
1600200 (epoch 0) bibbia3.txt	àbano e Làbano corse fuori da quell'uomo al pozzo. [30]	d e e ae e ee ee eee e ee eee toee e e eereee [ ]
1600400 (epoch 0) bibbia3.txt	se con sé le mogli, i figli e le figlie e tutte le pers	t ee tee do ee eo deereeo e ee eeree e eeeee ee eere
1600600 (epoch 0) bibbia3.txt	o numerosi. [28] Giacobbe visse nella terra d'Egitto di	eeoe e e [1] i ee ee ee ee ee tee eeeee ee ee
1600800 (epoch 0) bibbia3.txt	o, lo intingerete nel sangue che sarà nel catino e spal	e de e ee te e ee ee tee eee ee te ee e e t e ee t
1601000 (epoch 0) bibbia3.txt	bronzio, introdurrà le fibbie nei cordoni e unirai insi	ee ee d'oo eeee ee eereee ee ee ee e eeo e e ee
1601200 (epoch 0) bibbia3.txt	loro uncini d'argento, il rivestimento del loro capitel	i ee e e eereee te e ee ee ee ee ee ee e ee e ee
1601400 (epoch 0) bibbia3.txt	e che striscia per terra, [47] per distinguere ciò che	eer eeee ee ee tee ee e ee ii tee ee ee e ee dee e
1601600 (epoch 0) bibbia3.txt	erete. [22] Quando mieterai la messe della vostra terra	e e [1] ao to ee e e ee ee ee ee ee eee ee te
1601800 (epoch 0) bibbia3.txt	nazireo consacrerà così il suo capo. [12] Consacrerà di	i e e ee ee t de d eee ee e [ ] i e ee t de
1602000 (epoch 0) bibbia3.txt	a presenza del Signore e li portò a tutti gli Israeliti	aae e te ee eereee e ee eere d eereeo eee eeee e
1602200 (epoch 0) bibbia3.txt	avuto conoscenza. [16] Ma se li annulla qualche tempo	doaaa ee t ee e [ ] lo ii ee e ee eee tee ee ee e
1602400 (epoch 0) bibbia3.txt	e le norme che dovrà insegnare loro, perché le mettano	ei ee ee ee ee ee e e ee ee e deeree do ee to t
1602600 (epoch 0) bibbia3.txt	ro il Signore, vostro Dio. [19] Quando cingerai d'assed	o e eereee d eeee ee [1] io to ee e e ee ee ee
1602800 (epoch 0) bibbia3.txt	6] Viva Ruben e non muoia, benché siano pochi i suoi uo	Ii i eeee ee ee eeee e eo de o ee eo e ee ee
1603000 (epoch 0) bibbia3.txt	dominano la valle, [20] Bet-Peor, le pendici del Pisga	di e o e eo eerto d ii i e e di ierte e ee eeree
1603200 (epoch 0) bibbia3.txt	li occhi del Signore. [13] Eglon radunò intorno a sé gl	o a eee ee eereee e [1] oio ee ee d eo e ee dee
1603400 (epoch 0) bibbia3.txt	En-Kore: essa esiste a Lechi ancora oggi. [20] Sansone	dia t e [ e i ae ee e ee ee d e ee t ee [1] i te
1603600 (epoch 0) bibbia3.txt	si accamparono presso Eben-Ezer mentre i Filistei s'er	d i ee ee te e ee ee eee t ee eee ee e eert ee eere
1603800 (epoch 0) bibbia3.txt	scezza». [34] Ma Davide disse a Saul: «Il tuo servo pas	ee l e [11] o ii e ee ee e eereo dee eee ee ee e
1604000 (epoch 0) bibbia3.txt	mandò parte del bottino agli anziani di Giuda, suoi am	ee te de ee ee ee ee e ee ee e ee eereee eee e e
1604200 (epoch 0) bibbia3.txt	Assalonnon». [22] Ioab si gettò con la faccia a terra, s	e eae t ee. [11] Io iee eereee de ee eereee e ee ee
1604400 (epoch 0) bibbia3.txt	Sadoc prese il corno dell'olio dalla tenda e unse Salo	dire eeee e ee eo ee e ee ee e ee e eeee eert
1604600 (epoch 0) bibbia3.txt	vide, mio servo, e per amore di Gerusalemme, che ho sce	e de ee te d eerte e ee eereee t ee dee ee eee
1604800 (epoch 0) bibbia3.txt	alti come sui tui». [5] Giosafat disse al re d'Israele	t ee e eee eee [ ] i e ee ee ee ee eeeeeee e
1605000 (epoch 0) bibbia3.txt	nire presso di sé nel tempio del Signore. Egli concluse	e eee ee ee ee de ee ee ee eereee e [ee ee eo e
1605200 (epoch 0) bibbia3.txt	le bruciò sull'altare, rendendolo impuro, secondo la pa	i eeeeee dee e ee e de e e e eeee ee ee ee ee
1605400 (epoch 0) bibbia3.txt	Meten, [44] Ozia di Astarot, Sama e Ieiel, figli di Cot	i e d' iia ee eeee e dere e ee e dereo ee eere
1605600 (epoch 0) bibbia3.txt	ci ciascun oggetto di culto; [15] inoltre l'oro dei cand	eer eee e ee ee ee ee d ] i oo eee e ee ee te
1605800 (epoch 0) bibbia3.txt	o padre. Ecco, ti mando argento e oro. Su, rompi la tua	ee ee [eee de di te e ee te e e e [ee di ee ee eer
1606000 (epoch 0) bibbia3.txt	lea mille giovenchi e settemila pecore; anche i capi av	e ee o eer e ee ee to e eere e d ee e eereee e
1606200 (epoch 0) bibbia3.txt	oi, ci hanno mandato un uomo assennato, dei figli di Ma	de il oo ee te ee ee e ee to e de eereee ee ee
1606400 (epoch 0) bibbia3.txt	tti una disposizione del re a loro riguardo e un ordine	to de ee ee e e ee ee ee ee ee ee ee ee ee ee
1606600 (epoch 0) bibbia3.txt	rusalemme saranno ricostruite con oro e i loro baluardi	eee t ee ee e ee ee eeee e ee e e e e ee te ee
1606800 (epoch 0) bibbia3.txt	ò: «Amen! Amen!». [14] [1] Giuditta rispose loro: «Asc	[aa t die t ] ] ] Ii ti io de ee e ee e deeee
1607000 (epoch 0) bibbia3.txt	avversario maligno per Israele in ogni momento. [37] V	i eerte e ee te ee eeeee e e e ee ee ee [1] o
1607200 (epoch 0) bibbia3.txt	si portarono a Berea con ventimila fanti e duemila cav	i eereee e e ee e ee tee te e eerte e eee e ee e
1607400 (epoch 0) bibbia3.txt	emetro, ad Attalo, ad Ariarate e Arsace [23] e a tutti	e ee e eieeo t d eeee e e e eeee e e [ e a airoo
1607600 (epoch 0) bibbia3.txt	[2] Infatti egli era giunto nella città chiamata Persép	1] i to d eo e e ee eee ee ee eo dee e ee te d
1607800 (epoch 0) bibbia3.txt	tare, lo qual è la mia fine, perché io debba pazientare?	o e eea te de ee eert de tee d ee ee eere ee e
1608000 (epoch 0) bibbia3.txt	i Ram. Si accese di sdegno contro Giobbe, perché si co	tira [ i eere ee ee ee ee ee ee de tee de eert
1608200 (epoch 0) bibbia3.txt	Israele; [25] perché egli non ha disprezzato né disdegn	aae e [ ] il teo d eo ee ee ee ee ee e e ce ee ee
1608400 (epoch 0) bibbia3.txt	Eccoli, la bava alla bocca; le loro labbra sono spade. V	iai de de e to ee ee e aa e ee eee ee t ee e [
1608600 (epoch 0) bibbia3.txt	per noi l'opera delle nostre mani, l'opera delle nostr	eerte eereere ee ee eee ee t ee e e ee ee eee

# Profeta Virtuale

1614200 (epoch 0) bibbia3.txt	segnato ai suoi discepoli». [2] Ed egli disse loro: «Qu	a eo e e eee ee ee e [ ] oii io ee ee ee e deeo	loss: 2
1614400 (epoch 0) bibbia3.txt	i è più grande, chi sta a tavola o chi serve? Non è for	i dir die te dee eee e ee e e eee ee ee e e dere	loss: 2
1614600 (epoch 0) bibbia3.txt	eli». Gesù disse loro: «Proprio ciò che io vi dico. [26	[e i di ie ee e dell eo ee dee e ee ee e [ ]	loss: 2
1614800 (epoch 0) bibbia3.txt	orni dopo i discepoli erano di nuovo in casa e c'era co	eo ee e e ee ee e e e e eee e ee e ee eee ee	loss: 2
1615000 (epoch 0) bibbia3.txt	la a nessuno fuorché ai Giudei. [20] Ma alcuni di loro,	e ae eee eee eee d eerte [ ] i l teo ee ee e	loss: 2
1615200 (epoch 0) bibbia3.txt	ngelo gli ha parlato». [10] La disputa si accese a tal	ti eee ee eereo e [1] o i l eee ee ee e ee te	loss: 2
1615400 (epoch 0) bibbia3.txt	ra avrà un figlio. [10] E non è tutto: anche Rebecca eb	e e ae de eereo [ ] iao te dioo d eo ee e ee e	loss: 2
1615600 (epoch 0) bibbia3.txt	o senza la donna. [12] Come infatti la donna deriva d	e de te ee ee te [ ] a te e eo ee ee ee ee e ee	loss: 2
1615800 (epoch 0) bibbia3.txt	riconoscerete che la prova non è contro di noi. [7] Noi	i e e ee t e eee eo eee e ee te de eo ee ee [1] o	loss: 2
INDEX BOOK NAME	TRAINING SEQUENCE	PREDICTED SEQUENCE	LOSS
TRAINING STATS: batch 100/274 in epoch 0, non batch loss: 2.59332, batch accuracy: 0.25775	base al primo mezzo medio (la media) come in RMSProp, fa anche uso della media dei secondi mezzi		
Validating on bibbia95.txt	293 (end(itemize))		
VALIDATION STATS:	294 (begin)	loss: 2.57673, accuracy: 0.26197	
	295 (includegraphics[height=1cm]{adam.png})		
	296 (includegraphics[height=1cm]{adam.png})	Generating random text from learned state	
01,1]			
968612760]QLDigola cisane oi pentalda oonotcrotll orsoroe ca eaotre torto pero cuon aa dce duitde iirtaa ora e onrtio. [1]6 Iaa aio ea di e dea oo i cc			
lloeta sen iro, are denoeono pa le oa i deraa elle oe irree do ssroere po cena etcorda, ei ine tn eerlloni sso e, coe erentii e certio orti earlti doie direl			
la tlnle cllnoe ea earsalsea, oae e ciottia dillando ael li ctle a i stona e eraldonsdi, puo cententia e aotsi deo duate a ce urlnoio coo li urra ooo, pla			
e sillete. [[1] She atssia ee anco al ca ee i o iitll, lo erori e oraoio i talssuote, ae orre si eo patieerde cllrlntroa eonaltto preon ereoeie cala eorel			
surli dello e a uato, oen ccoo de ao an orlata oirne til a ce ei elaoi ineltei eel scisciee pea uienalon o ea uene lonli ciocrono olana do eteirlordol			
i ie aro eni ii drasoo cern sea ilso i duaron ce me cellia di memoria usiamo una GRU nella quale immagazio un carattere alla volta con un end			
	End of generation		
0%	303		
	Il dataset è nella nostra disposizione e relativamente piccolo, il che significa che		
	il problema di memorizzare troppo correttamente la Bibbia e non essere in grado	100%	
2400000 (epoch 0) bibbia7.txt	loni sulla terra dopo il diluvio. [11] [1] Tutta la te	o e de a deree dilerd st oo [11]] [1] Ao ti di sor	loss: 2
2400200 (epoch 0) bibbia7.txt	che era di bell'aspetto. [8] Era là da molto tempo, q	e s di sir aarterte [1] Ala di si sin a de eir sue	loss: 2
2400400 (epoch 0) bibbia7.txt	uda disse alla nuora Tamar: «Ritorna a casa da tuo padr	se di se d a sor e dint a [Sist eo d lont di sort diree	loss: 2
2400600 (epoch 0) bibbia7.txt	Ma i fratelli di Giuseppe cominciarono ad aver paura, d	a siene a si siene eerdone oo e d illlarearto ce	loss: 2
2400800 (epoch 0) bibbia7.txt	ia contro il faraone, i suoi carri e i suoi cavalieri». del	don la d sirt e c cur donea d l sur done e e	loss: 2
2401000 (epoch 0) bibbia7.txt	giudizio gli urlm e i tumtum. Saranno così sopra il	i Sioni eo dii soa ii s siri e [int oon de ee d	loss: 2
2401200 (epoch 0) bibbia7.txt	gnore, come il Signore aveva ordinato a Mosè. [26] Coll	nore cone d signore d i a dei o e d lirt [1] Aon a	loss: 2
2401400 (epoch 0) bibbia7.txt	si laverà una seconda volta e sarà puro. [59] Questa è	ce si ore dn sara ee den a d le t der [1] Auento d	loss: 2
2401600 (epoch 0) bibbia7.txt	Ma se quella casa, posta in una città cinta di mura, n	Ao si sue lton certo d so donto do i di sine co	loss: 2
2401800 (epoch 0) bibbia7.txt	tezer, figlio di Ammisaddat. [72] L'undicesimo giorno f	e e [inli d sltioe ee [1] Aaanti o e a dio eo di	loss: 2
2402000 (epoch 0) bibbia7.txt	in questo deserto per far morire noi e il nostro besti	d sue to dilt ao der sirtal e do s so sortea dirte	loss: 2
2402200 (epoch 0) bibbia7.txt	occa ha promesso». [25] I figli di Gad e i figli di Rub	noonde sue ee [1] Allirni di sirelll sirni di sini	loss: 2
2402400 (epoch 0) bibbia7.txt	da dimenticare il Signore, tuo Dio, che ti ha fatto us	dilsie e ee a d signore ee sio deo si se sirt d t	loss: 2
2402600 (epoch 0) bibbia7.txt	l'accampamento e là andrai per i tuoi bisogni. [14] Nel	aanoneere ee d li sintie der s sir dint o [1] Ao	loss: 2
2402800 (epoch 0) bibbia7.txt	e, e là ella abitava, [16] e disse loro: «Andate verso i	c li sr a dnioe e e1] A ci te di o «Sl i e diree d	loss: 2
2403000 (epoch 0) bibbia7.txt	riore e fino a Ghezer, e faceva capo al mare. [4] I fig	e d ligt d diese e d lirt e donerd aone [1] A lirn	loss: 2
2403200 (epoch 0) bibbia7.txt	i vestì variopinte a ricamo; una veste variopinta a due	sorto dint e e d lo one [o sirt dint e e d lil	loss: 2
2403400 (epoch 0) bibbia7.txt	metallo fuso? Sappiate ora quello che dovete fare». [1	sirt ea dirt [inrnr d due la soe si e t dirt [1] A	loss: 2
2403600 (epoch 0) bibbia7.txt	le Astarti; indirizzate il vostro cuore al Signore e s	li srteree [ i a ii e d sortoa don e d signore d le	loss: 2
2403800 (epoch 0) bibbia7.txt	del capo un tessuto di pelo di capra e li coprì con una	i Sonerdo se te t di sor di sonee d la sonee don s	loss: 2

# Profeta Virtuale

nuovolibro.txt

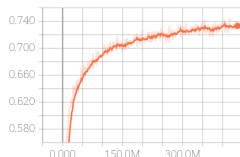
5  
4 {4}  
3 [1] Non avete ascoltato la parola del Signore, vostro Dio, come ha detto al Signore a Giosuè, figlio  
2 di Nun, e con tutto il suo esercito e con tutta la terra d'Egitto, perché sia scritto nella terra d  
1 'Egitto.  
393 [2] Poi ha fatto uscire dall'Ariba del monte Eran, con tutti gli abitanti della terra d'Egitto e dal  
1 la terra d'Egitto e di Gerusalemme, per sempre. [3] I sacerdoti che sono stati stabiliti in mezzo al  
2 primo posto di Giuda e di Baal, senza difetto, né sacerdoti né i saneri di questi servi che si trov  
3 ano su di loro per il Signore e non hanno permesso ai loro figli per i loro prini per il loro Dio, n  
4 ella loro città e il loro cuore non si poteva comprare il sangue dei loro padri.  
5 [4] Il Signore parlò a Mosè e disse: [5] «Parla agli Israeliti dicendo loro: "Prenditemi in mano al  
6 Signore e sarete salvati davanti al Signore". [6] Perché così dice il Signore Dio: Perché così dice  
7 il Signore Dio: Perché ci sarete come il Signore a vostro popolo e ci sarete stabiliti da questo mon  
8 te, perché il Signore vi ha fatto uscire dall'Egitto per darti in mano a Giacobbe, [7] ma se non vog  
9 lio osservare la vita del Signore, vostro Dio, che ha fatto al Signore e al suo servizio, per sempre  
10 . [8] Chiunque vi abbandonerà, si sarà scacciato davanti al Signore e si scaccerà davanti a lui, ch  
11 e avete conservato il suo popolo e si sarà consegnato in mano al Signore e sarà sterminato il suo po  
12 polo. [9] Se invece si accosterà con la faccia a terra e lo renderò con lui come se stesso con il su  
13 o prossimo e le mura della terra, con il suo cuore e il suo popolo, con le sue preghiere, le sue pre  
14 ghiera, le sue pecore e le sue lingue, perché io sono il Signore, tuo Dio, ed egli non ti sei perdon  
15 ato.  
16 [10] Solo tu sei un popolo di tutti i profeti che hai promesso a Mosè e divorerai la parola che il S  
17 ignore, tuo Dio, ti avrà dato al tuo popolo, tu sarai comprata in mezzo a te. [11] Se il Signore, tu  
18 o Dio, sta per darti in mano a te, per servire il Signore, tuo Dio, e tu sei il Signore, tuo Dio, e  
19 tu non ti avrai parlato con te. [12] Poiché il Signore ha fatto così contro il Signore, tuo Dio, e t  
20 i ha detto: "Perché tu hai dato un po' di te e ti ha fatto uscire il mare con te, non ho potuto fors  
21 e, non ti avevo dato alcun motivo, non hai fatto a cassa del tuo servo e alle tue parole e a causa d  
22 i una terra dove sono i nostri padri e nei tuoi figli mangiamo e come il Signore, tuo Dio, ti ha dat  
23 o per mezzo d'illo spirito di Dio. [13] Se tu sai che il Signore, tuo Dio, sta per servire, perché il  
24 Signore ha preparato le tue parole contro di te, perché tu non hai profetato il tuo servo e il Sign  
25 ore ti ha detto per te. [14] Poiché il Signore, tuo Dio, ti ha detto: "Per questo io sono il Signore  
26 , tuo Dio, che ha fatto al Signore, tuo Dio, e contro il Sionore e tutto il tuo popolo, che hai prom  
27 esso contro di te e tu hai dato a te con tutti i popoli che sono stati abbattuti a Gerusalemme. [15]  
28 Il Signore, tuo Dio, ti avrà dato a Gerusalemme con tutta la sua casa, perché sia scelto il tuo ser  
29 vo e i suoi profeti, perché il Signore ti aveva concesso una terra dove si tenga il tuo santuario e  
30 ti avrà consegnato nel tempio del Signore. [16] Perché tu sei un Dembro di tutti i popoli che sono s  
31 tati consegnati in mezzo a te presso, non te la potrai per il Signore, tuo Dio, perché ti abbiamo d  
32 etto contro di te per tutto il tempo di questo tempio e al Signore ti avrà risposto in questo libro.  
33 [17] Non ti serai perché sei un popolo di peccato, come abitamente dei tuoi padri e della casa d'Is  
34 raele, che hai fatto a te e tu sei uscito dall'Egitto, perché il Signore, tuo Dio, ti avrà comandato  
35 e con tutto il tuo popolo Israele non temere, non ti abbandonerà alla tua voce.  
36 [18] Perché io sono il Signore, tuo Dio, che hai fatto a Gerusalemme e tu hai fatto compitare in tut  
37 to Israele, perché il Signore ti ha detto così contro di te.

38  
NORMAL +0 ~0 -0 7 masterE nuovolibro.txt

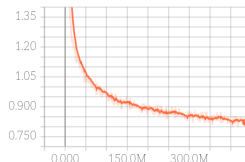
text [unix] 10604 words < 66%



batch\_accuracy



batch\_loss



Name	Smoothed	Value	Step	Time	Relative
●	0.7332	0.7287	437.6M	Sun May 20, 02:05:15	22h 56m 0s



Name	Smoothed	Value	Step	Time	Relative
●	0.8220	0.8003	435.2M	Sun May 20, 01:57:40	22h 48m 25s

- Tensorflow 1.8, Nvidia GeForce 940MX, 384 Cuda Cores.  $\approx 24h$  training
- Si notano già ottimi risultati dopo poche ore (4 o 5) con un hardware casalingo.
- Il dropout fa naturalmente aumentare il tempo di decadimento della funzione di errore ma è il prezzo da pagare per avere una migliore generalizzazione.
- Diverse finestre temporali sono state scelte. Solo con una finestra temporale abbastanza lunga,  $> 30$  la rete è stata in grado di apprendere l'ordine dei versetti e le regole sintattiche come aprire-chiudere virgolette.
- È possibile fare un sampling diverso dei caratteri, scegliendo anche quelli meno probabili con una certa frequenza, che equivale ad alzare o abbassare la “temperatura” del generatore.