Protein analysis

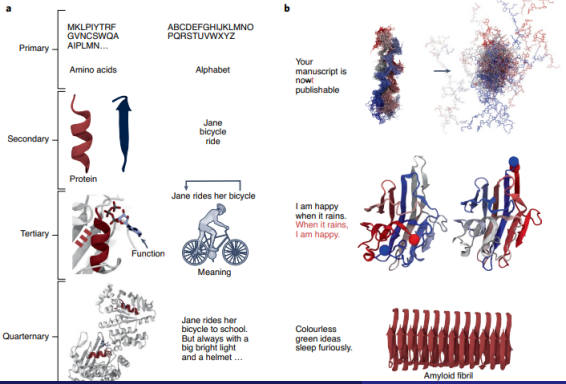
Paolo Marino

October 21, 2024

Indice

Introduzione

Il lavoro svolto dal paper consiste nell'appliccare il deep learning per "comprendere" le caratteristiche fondamentali delle proteine da cui poi, si possono eseguire ulteriori analisi. Per far ciò si possono usare per esempio reti ricorrenti usate per l'eleborazione del linguaggio naturale: questo è possibile poiché l'insieme di amminoacidi che compongono una proteina possono essere visti come un insieme di caratteri che compongono una parola

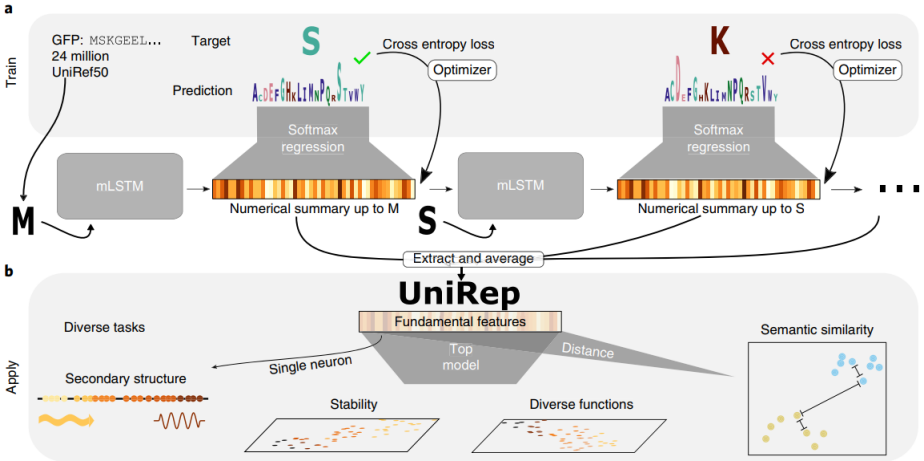


Modello

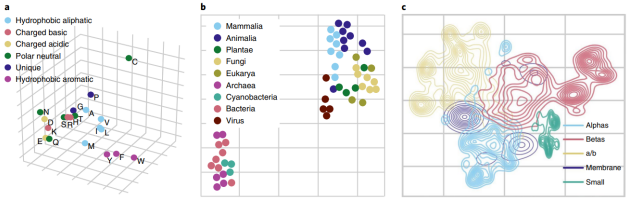
Per far questo il modello utilizza una variante della LSTM chiamato multiplicative LSTM o mLSTM:

- mLSTM = mRNN + LSTM
  - mRNN: progettata per garantire transizioni flessibili e dipendenti dall'input
    $m_t = (W_{mx} \cdot x_t) \circ (W_{mh} \cdot h_{t-1})$
    $\hat{h}_t = W_{hm} \cdot m_t + W_{hx} \cdot x_t$
  - LSTM: progettata per controllare informazioni lungo la rete, anche lontane tra di loro
  - mLSTM: combina i due vantaggi delle reti

Modello


a
Train
GFP: MSKGEEL...
24 million
UniRef50
Target
S
Prediction
Cross entropy loss
Optimizer
Softmax regression
mLSTM
M
Numerical summary up to M
S
mLSTM
Softmax regression
Numerical summary up to S
K
Cross entropy loss
Optimizer
...
Extract and average
b
Apply
UniRep
Fundamental features
Diverse tasks
Secondary structure
Single neuron
Top model
Distance
Semantic similarity
Stability
Diverse functions

Alcuni risultati

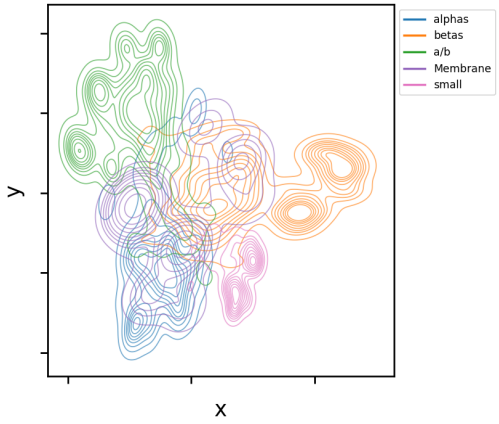


Alcuni esempi di risultati di unirep, ottenuti estraendo gli stati interni della rete per ogni sequenza:

- PCA degli embeddings imparati da UniRep
- Visualizzazione dei cluster di organismi
- T-sne per visualizzare cluster di strutture secondarie

Obiettivi

- Creare un modello che sia in grado di comprendere le relazioni tra i vari amminoacidi
- Utilizzare gli stati interni delle varie sequenze per eseguire ulteriori analisi. In particolare cercare di riordurre i cluster di queste strutture proteiche:



Dataset

Per svolgere le analisi sono state utilizzati due dataset:

- UniRef(UniProt Reference Cluster): che contiene cluster di sequenze proteiche che sono simili. Il livello di similitudine è deciso dalla versione del dataset scelto (nel mio caso al 50
- SCOP: sta per "Structural Classification of Proteins", ed è una classificazione gerarchica delle proteine in base alla loro struttura. L'analisi in questo caso è stata fatta in base alla loro struttura secondaria (alpha, beta o small), e in base al tipo di proteina (proteina di membrana)

Spiegazione procedimento

Il procedimento si divide nei seguenti step:

- Analisi su UniRep:
  1. Pre-processing
  2. Training and evaluation
  3. Analisi dei risultati
- Analisi su SCOP:
  1. Pre-processing
  2. Training and evaluation
  3. Analisi dei risultati e confronto con i risultati del paper

Uniref pre-processing

Il dataset UniRef contiene sequenze di proteine formate da amminoacidi che devono essere codificate in numeri applicando questo tipo di encoding

```
# Lookup tables
aa_to_int = {
    'M':1, 'R':2, 'H':3, 'K':4, 'D':5, 'E':6,'S':7, 'T':8, 'N':9, 'Q':10, 'C':11, 'U':12, 'G':13, 'P':14, 'A':15, 'V':16, 'I':17, 'F':18, 'Y':19, 'W':20, 'L':21,
    'O':22, #Pyrrolysine
    'X':23, # Unknown
    'Z':23, # Glutamic acid or GLutamine
    'B':23, # Asparagine or aspartic acid
    'J':23, # Leucine or isoleucine
    'start':24,
    'stop':25,
}
```

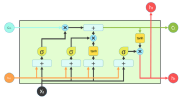
I vari step quindi sono:

- Conversione delle sequenze di amminoacidi in numeri
- Creazione di file di training e test e applicazione di un padding di zeri
- Creazione dei batch

Modelli usati

Modelli usati:

LSTM



biLSTM

Ho scelto di usare questi due modelli in modo da eseguire una comparazione tra le 3 versioni di lstm (Lstm, biLstm, mLstm) e vedere i vantaggi e gli svantaggi di usare un approccio piuttosto che un altro

Modelli usati

Entrambi i modelli sono formati da:

- Layer di embedding in modo da rappresentare dati discreti come l'encoding degli amminoacidi in dati continui
- Due lstm layer (unidirezionali nel primo caso e bi-direzonali nel secondo)
- Un layer lineare per riportare la dimensione dell'embedding al numero di vocaboli
- Una log-softmax in modo da normalizzare l'output e fare poi classificazione multiclasse
- Una lista in cui salvare i vettori di embedding

Funzioni e iperparametri

Per entrambe le reti ho deciso di utilzzare le stesse funzioni e iperparametri in modo da avere un confronto più preciso:

- Come loss ho scelto una negative log-likelihood usata per confrontare l'output del modello escluso l'ultimo amminoacido e come target la sequenza di partenza escluso il primo amminoacido
- Come ottimizzatore ho scelto Adam con un learning rate di 0,001
- Come batch size ho scelto 64
- Come iperparametri alle reti ho passato la dimensione del vocabolario di 26, l'embedding size di 10 e l'hiddend size (ovvero la dimensione del layer lstm) di 64

Next character prediction

Per vedere quanto i modelli fossero accurati nel "comprendere" le relazioni tra amminoacidi ho calcolato la loss sul test set e anche l'accuratezza con cui vengono previste le sequenze di amminoacidi:

Esempio	Accuratezza Lstm	Accuratezza biLstm
1	73.47%	100.00%
2	73.71%	100.00%
3	73.37%	100.00%
4	74.00%	100.00%
5	73.37%	100.00%
6	73.46%	100.00%
7	74.12%	100.00%
8	73.31%	100.00%
9	73.59%	100.00%
10	73.59%	100.00%

Table: Risultati di accuratezza

Loss Lstm	Loss biLstm
0.792052	1.718052e-08

Table: Confronto delle loss

SCOP pre-processing

Il dataset consiste in 28000 sequenze prese da SCOP che contengono le strutture di amminoacidi, le classi di appartenenza e gli embedding corrispondenti a ciascuna sequenza. Il pre-pricessing consiste nel:

- Eliminare le colonne che non servono per l'analisi
- Aggiungere il padding e creare i vari batch esattamente come si è fatto in precedenza

Risultati

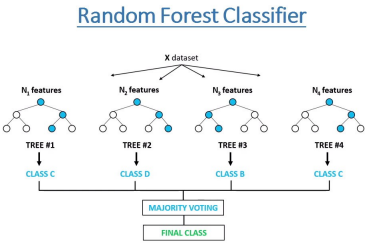
Ho deciso di fare un confronto con i risultati del paper su due task:

- Embedding classification: Overo vedere quanto un modello di classifiazione di machine learning sia in grado di assegnare la corretta sequenza all'etichetta SCOP giusta
- Riproduzione di cluster tramite tecniche di riduzione della dimensionalità

L'analisi è stata fatta allenando i modelli su sequenze estratte da UniRef. Circa 5 milioni e poi nuovamente su 15 milioni per vedere quanto impatta un training più corposo sul fine-tuning successivo fatto su SCOP. Nel primo caso ho allenato per 4 epoche e nel secondo per una

Embedding classification

Per eseguire la classificazione degli embedding ho applicato un classificatore random forest sugli embedding delle sequenze:



Sono state usate metriche come accuracy, precision, recall e f1 score

Embedding classification: risultati

Ho applicato il random forest sia sugli hidden states della mLstm che quelli implementati dalla biLstm e dalla Lstm:

Class	Precision	Recall	F1-score	Support
a	0.96	0.97	0.96	1021
b	0.98	0.98	0.98	1596
c	0.98	0.96	0.97	1309
f	1.00	0.89	0.94	66
g	0.97	0.97	0.97	277
Accuracy			0.97	4269
Macro avg	0.98	0.96	0.97	4269
Weighted avg	0.97	0.97	0.97	4269

Table: Classification Report mLSTM

Class	Precision	Recall	F1-score	Support
a	0.88	0.81	0.84	1021
b	0.91	0.91	0.91	1596
c	0.85	0.93	0.89	1309
f	1.00	0.65	0.79	66
g	0.97	0.94	0.96	277
Accuracy			0.89	4269
Macro avg	0.92	0.85	0.88	4269
Weighted avg	0.89	0.89	0.89	4269

Table: biLSTM trained on 5 million sequences UniRef

Class	Precision	Recall	F1-score	Support
a	0.91	0.85	0.88	1021
b	0.91	0.93	0.92	1596
c	0.89	0.93	0.91	1309
f	1.00	0.71	0.83	66
g	0.98	0.94	0.96	277
Accuracy			0.91	4269
Macro avg	0.94	0.87	0.90	4269
Weighted avg	0.91	0.91	0.91	4269

Table: biLSTM trained on 15 million sequences UniRef

Class	Precision	Recall	F1-score	Support
a	0.93	0.92	0.93	1021
b	0.94	0.96	0.95	1596
c	0.94	0.94	0.94	1309
f	1.00	0.80	0.89	66
g	0.97	0.92	0.95	277
Accuracy			0.94	4269
Macro avg	0.96	0.91	0.93	4269
Weighted avg	0.94	0.94	0.94	4269

Table: LSTM trained on 5 million sequences UniRef

Class	Precision	Recall	F1-score	Support
a	0.94	0.92	0.93	1021
b	0.94	0.97	0.95	1596
c	0.95	0.95	0.95	1309
f	0.96	0.82	0.89	66
g	0.98	0.92	0.95	277
Accuracy			0.94	4269
Macro avg	0.95	0.91	0.93	4269
Weighted avg	0.94	0.94	0.94	4269

Table: LSTM trained on 15 million sequences UniRef

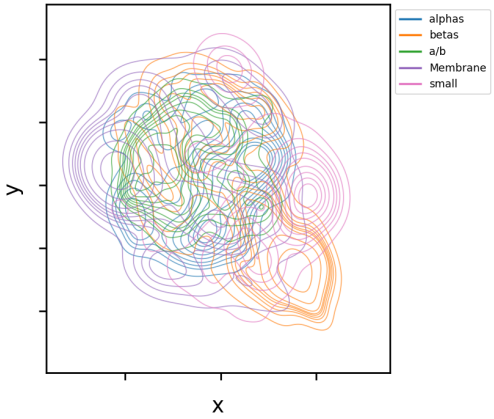
Riproduzione cluster

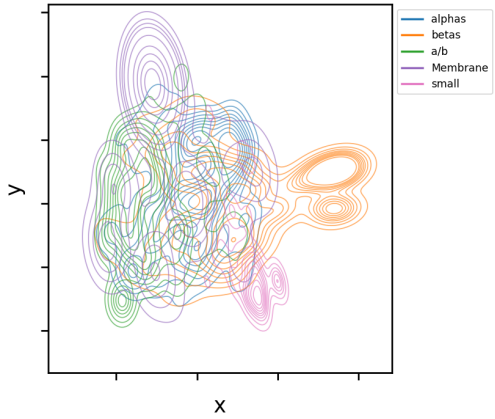
I cluster sono stati riprodotti usando due tecniche di riduzione dimensionale: tsne(usata anche dagli autori del paper) e umap

- **Tsne:** converte le somiglianze tra i punti dati nello spazio ad alta dimensione in probabilità, e quindi mappa queste probabilità in uno spazio a dimensione inferiore in un modo che preservi il più possibile le relazioni tra i punti dati
- **Umap:** utilizza un approccio basato su grafi per costruire una rappresentazione topologica dei dati, che viene poi incorporata in uno spazio a bassa dimensionalità utilizzando la discesa del gradiente stocastico

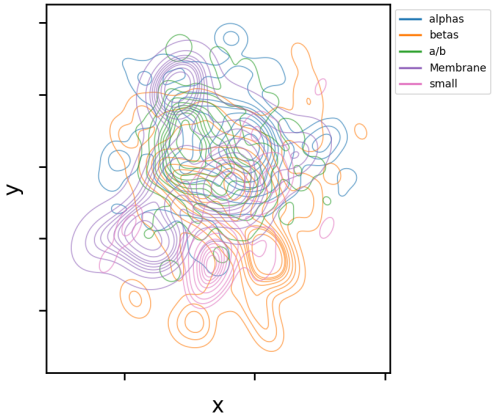
Riproduzione cluster: risultati

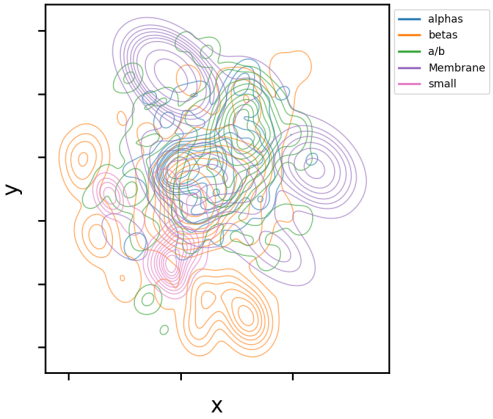
Come si può vedere dalle immagini sottostanti la normale LSTM riesce a riprodurre cluster più "riconoscibili" rispetto alla biLSTM:





Ho provato anche ad applicare umap sia alla LSTM che alla biLSTM





Considerazioni finali e conclusione

Da queste analisi si può evincere che:

- biLSTM sembra essere più veloce nel capire come predire il prossimo amminoacido dati quelli precedenti e questo si può evincere anche da come la loss cala molto più velocemente rispetto alla normale LSTM. Tuttavia, la LSTM sembra essere più accurata quando si tratta di eseguire analisi succesive su SCOP
- T-sne sembra essere migliore rispetto ad umap per costruire una rappresentazione locale dei dati
- La mLSTM presentata dal paper batte le altre due reti sia nella classificazione di etichette, sia nella riproduzione di cluster

Bibliografia

- https://www.ebi.ac.uk/training/online/courses/uniprot-quick-tour/the-uniprot-databases/uniref/
- https://scop2.mrc-lmb.cam.ac.uk/about
- https://medium.com/@mrmaster907/introduction-random-forest-classification-by-example-6983d95c7b91
- Unified rational protein engineering with sequence-based deep representation learning, Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi George M. Church
- Multiplicative lstm for sequence modelling, Ben Krause, Iain Murray, Steve Renals, Liang Lu