

ΝΕΥΡΟ-ΑΣΑΦΗΣ ΥΠΟΛΟΓΙΣΤΙΚΗ
Προγραμματιστική εργασία
ΖΙΩΓΑΣ ΔΗΜΗΤΡΙΟΣ 2214
ΠΕΤΣΗΣ ΚΩΝΣΤΑΝΤΙΝΟΣ 2168

1)Γενικές πληροφορίες

Το αρχείο `extract_diff_length_of_timelines.py` παίρνει 2 ορίσματα πριν την έναρξή του, που δείχνουν τα όρια για τα οποία θα ταξινομηθούν οι χρονοσειρές των άρθρων ανά μήκος χρονοσειράς. Δηλαδή, αν το τρέξουμε ως εξής: `python extract_diff_length_of_timelines.py 10 16` τότε το πρόγραμμα θα διαβάσει το `outputacm.txt` και θα δημιουργήσει 5 αρχεία. Τα `timeline10_years_refs.txt`, `timeline11_years_refs.txt` ... `timeline15_years_refs.txt`. Κάθε τέτοιο αρχείο έχει μία σειρά από αριθμούς αναφορών σε μία γραμμή και μια σειρά από έτη στα οποία κάποιο άρθρο πήρε αυτές τις αναφορές. Το `timeline10_years_refs.txt` περιέχει τέτοιες λίστες των 10 ετών, το `timeline11_years_refs.txt` περιέχει των 11 κ.ο.κ

Το αρχείο `final1.py` διαβάζει τα `timeseries0x.txt` με $x=[1,2,3...20]$, φορτώνει το μοντέλο για την αντίστοιχη μήκους χρονοσειρά και κάνει πρόβλεψη για το επόμενο έτος και για μετά από πέντε έτη.

Το αρχείο `make_keras_models.py` και το `make_keras_models_5.py` προπονούν μοντέλα των 10,11,12...44 μήκους χρονοσειρών για πρόβλεψη του επόμενου έτους και των πέντε επομένων αντίστοιχα.

2)Αρχιτεκτονική δικτύου

Χρησιμοποιούμε από το `keras` ένα δίκτυο LSTM, που είναι ιδανικό για προβλέψεις τιμών ανά τακτά χρονικά διαστήματα. Το LSTM είναι ένα Recurrent neural network. Σαν βελτιστοποιητή (optimizer), χρησιμοποιούμε τον `adam` (ο αλγόριθμος στο τέλος).

Έστω ότι εκπαιδεύουμε ένα δίκτυο για χρονολογία μήκους 10
έστω $TRAIN_SIZE = 10 - 1 = 9$.

Το δίκτυο 10 θα έχει για είσοδο έναν νευρώνα, στον οποίο διοχετεύεται μήκος εισόδου `TRAIN_SIZE`.

Στο hidden layer έχουμε `TRAIN_SIZE` κρυφές καταστάσεις και βγάζουν έξοδο προς τον εξωτερικό layer μήκους `TRAIN_SIZE`.

Σαν έξοδος, βγαίνει μία τιμή, η πρόβλεψη για 1 έτος ή για 5 έτη.

3)Αποτίμηση της επίδοσης

Για χρονοσειρά των 10

loss: 3.3609317515163966

accuracy:0.48408710217755446

Time for the system to answer is 0.6987732230008987 seconds.

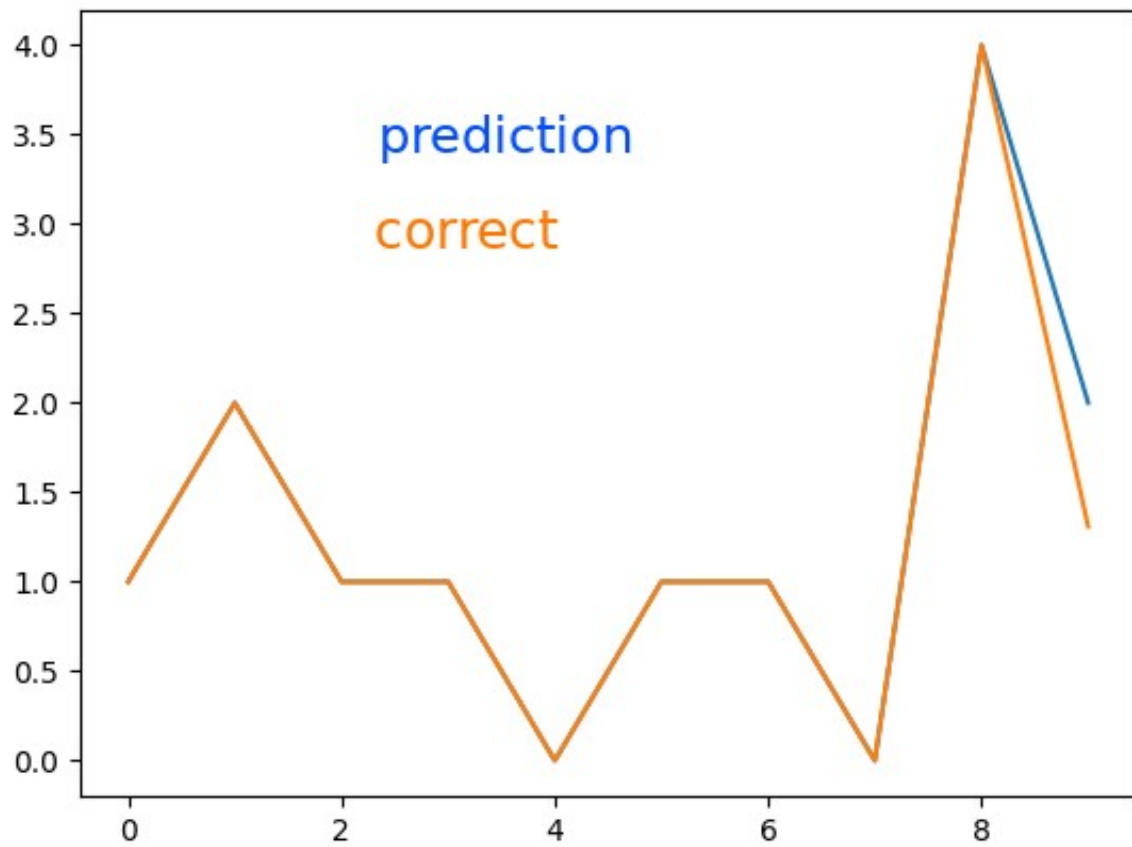
Prediction για 1 χρόνο

[1 2 1 1 0 1 1 0 4 2]

[1, 2, 1, 1, 0, 1, 1, 0, 4, 1.308309]

Prediction = 1.308309

Correct = 2



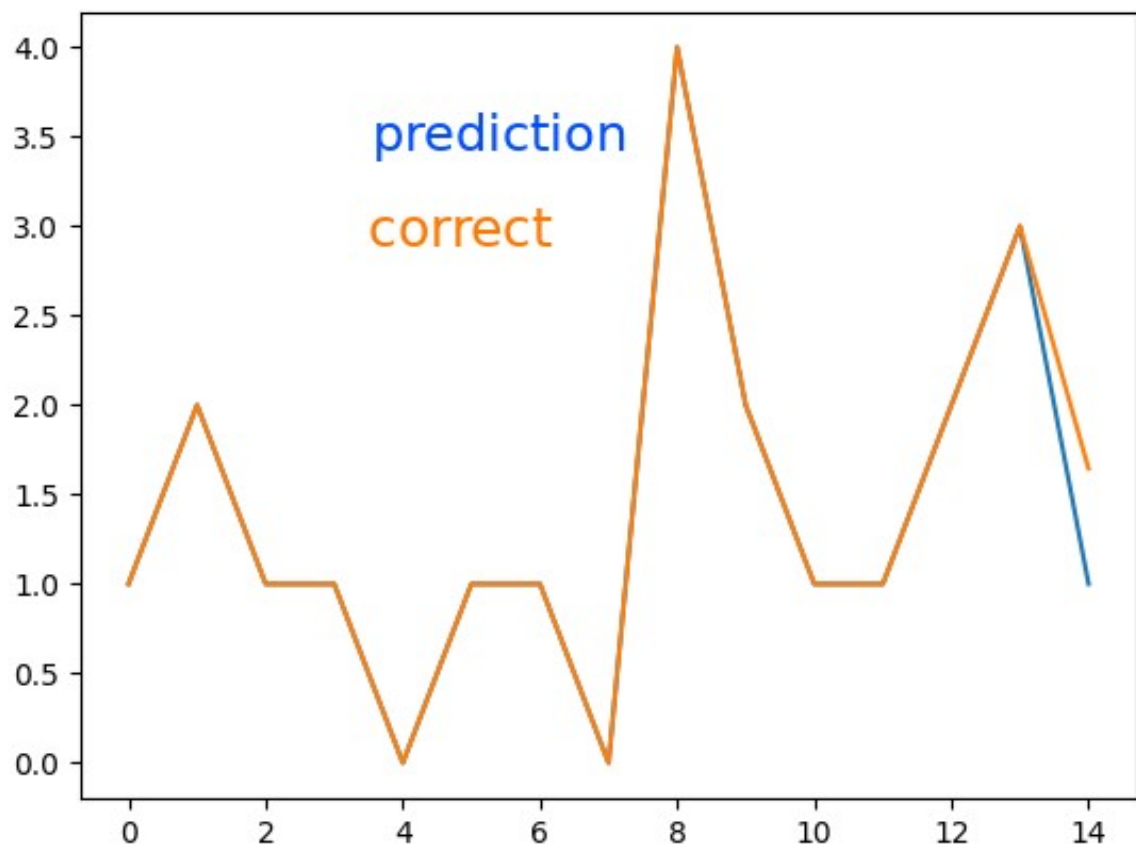
prediction για 5 χρόνια

[1,2,1,1,0,1,1,0,4,2,1,1,2,3,1]

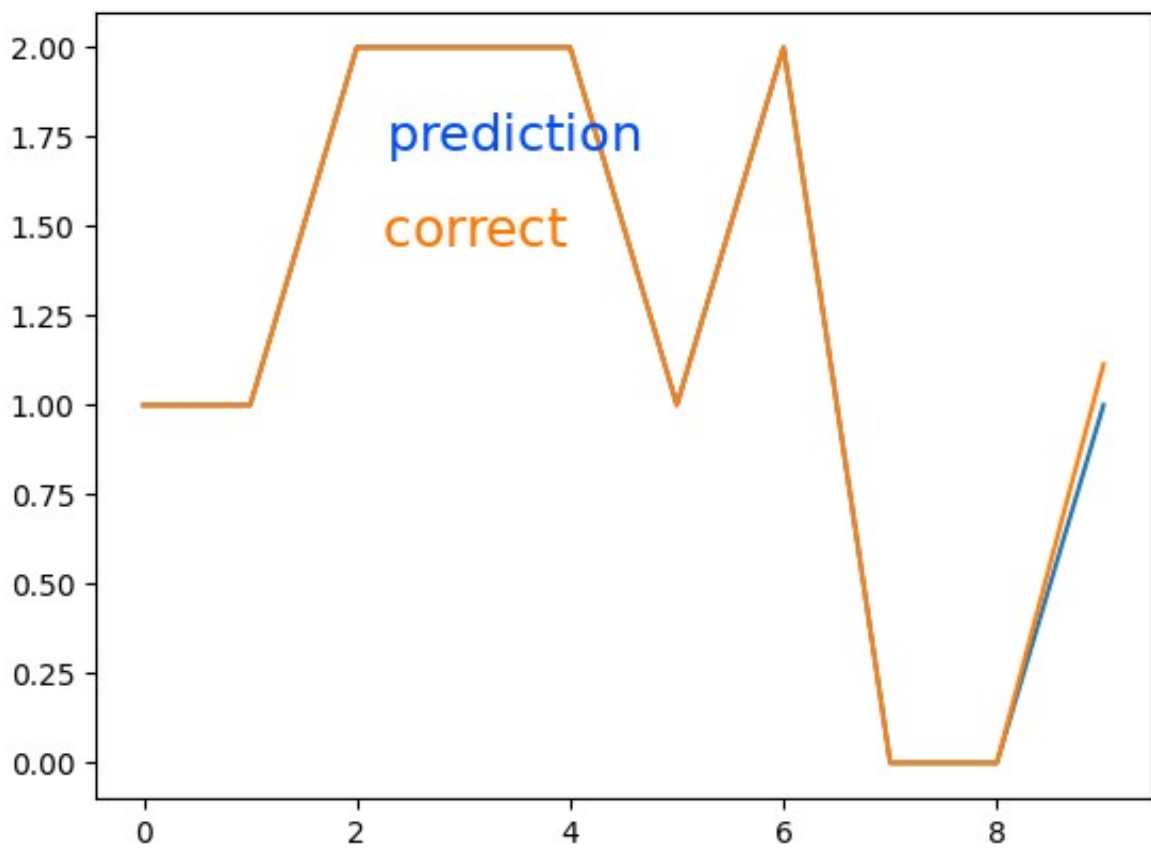
[1,2,1,1,0,1,1,0,4,2,1,1,2,3,1.6450815]

Prediction is : 1.6450815

Correct is :1



prediction για 1 χρόνο
[1 1 2 2 2 1 2 0 0 1]
[1, 1, 2, 2, 2, 1, 2, 0, 0, 1.1137747]
Prediction = 1.1137747
Correct = 1



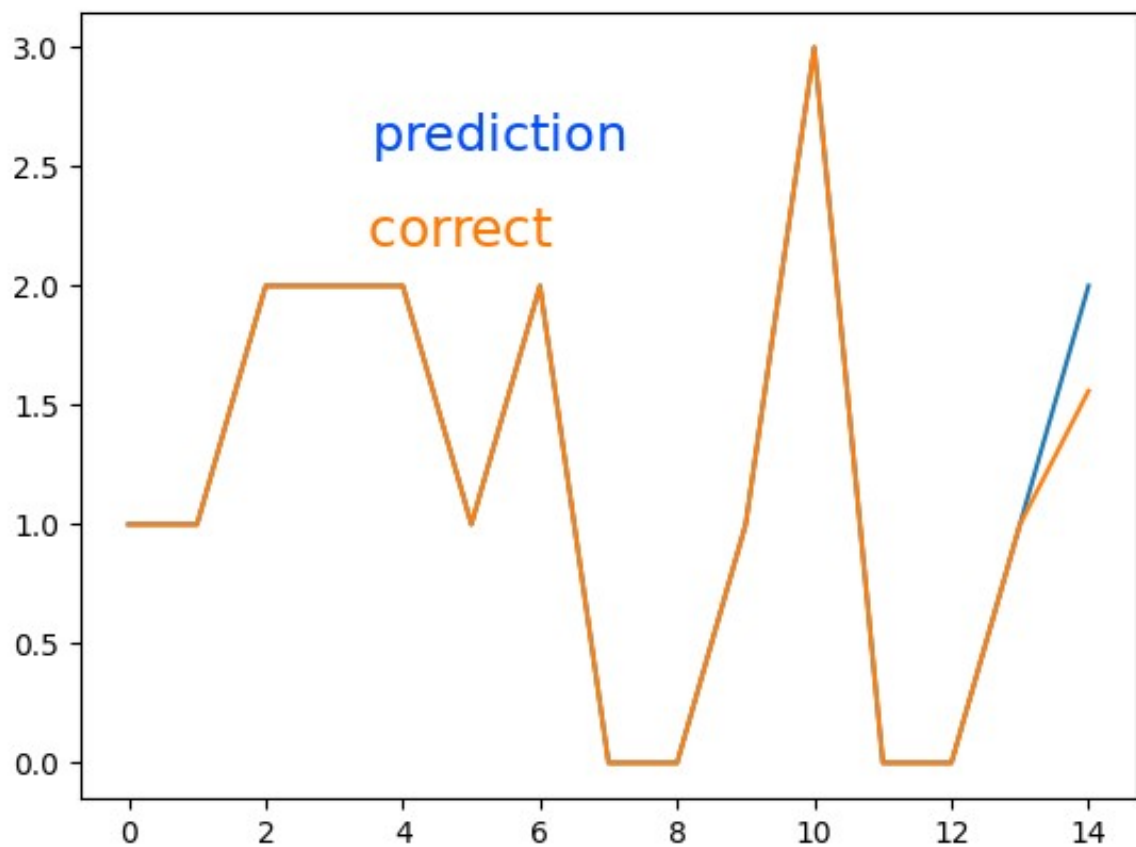
prediction για 5 χρόνια

[1,1,2,2,2,1,2,0,0,1,3,0,0,1,2]

[1,1,2,2,2,1,2,0,0,1,3,0,0,1,1.5597068]

Prediction is : 1.5597068

Correct is :2



Για χρονοσειρά των 20

prediction για 1 χρόνο

loss:0.7404102678881602

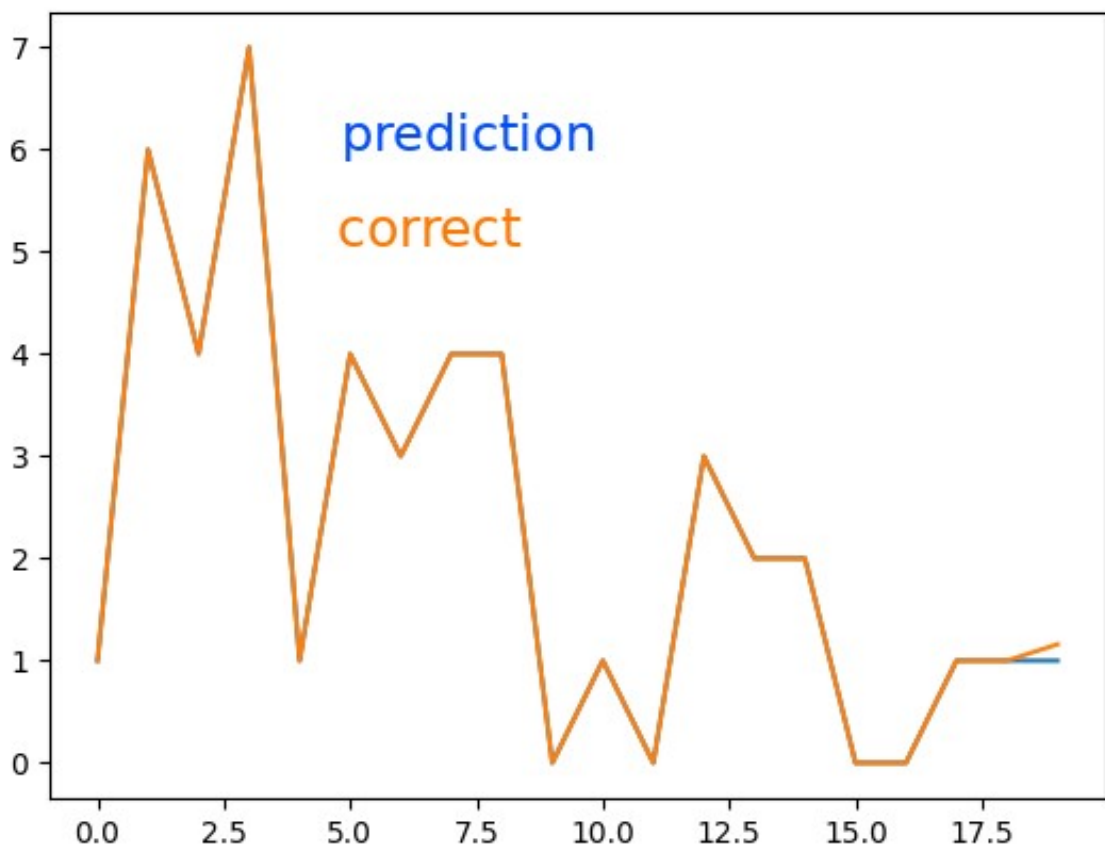
accuracy: 0.9588014981273408

[1 6 4 7 1 4 3 4 4 0 1 0 3 2 2 0 0 1 1 1]

[1, 6, 4, 7, 1, 4, 3, 4, 4, 0, 1, 0, 3, 2, 2, 0, 0, 1, 1, 1.1598287]

correct=1

prediction=1.1598287



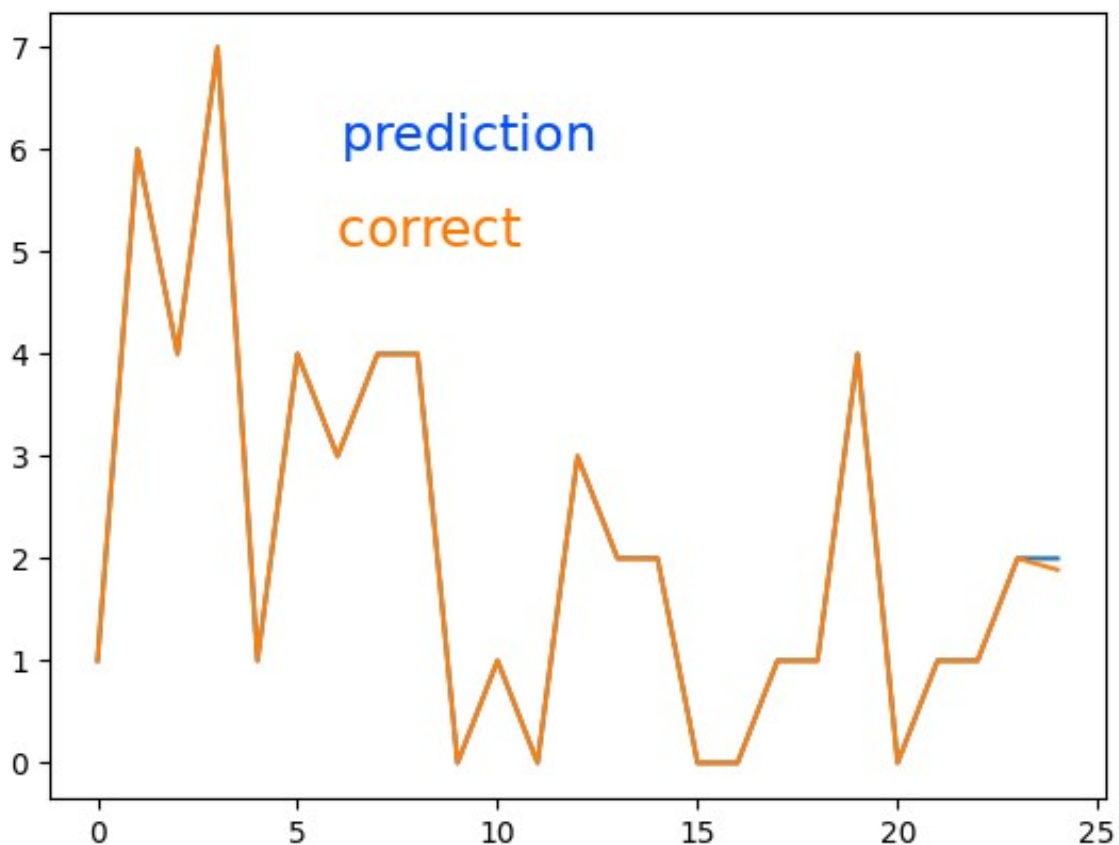
prediction για 5 χρόνια

[1 6 4 7 1 4 3 4 4 0 1 0 3 2 2 0 0 1 1 1 4 0 1 1 2 2]

[1 6 4 7 1 4 3 4 4 0 1 0 3 2 2 0 0 1 1 1 4 0 1 1 2 1.8887237]

correct=2

prediction=1.8887237



4)Χρόνοι εκπαίδευσης ως συνάρτηση του μεγέθους των δεδομένων εισόδου

Για μήκος χρονολογίας 10, 1194 άρθρα
330.084248505 seconds.

Για μήκος χρονολογίας 11, 987 άρθρα
147.79023395500008 seconds

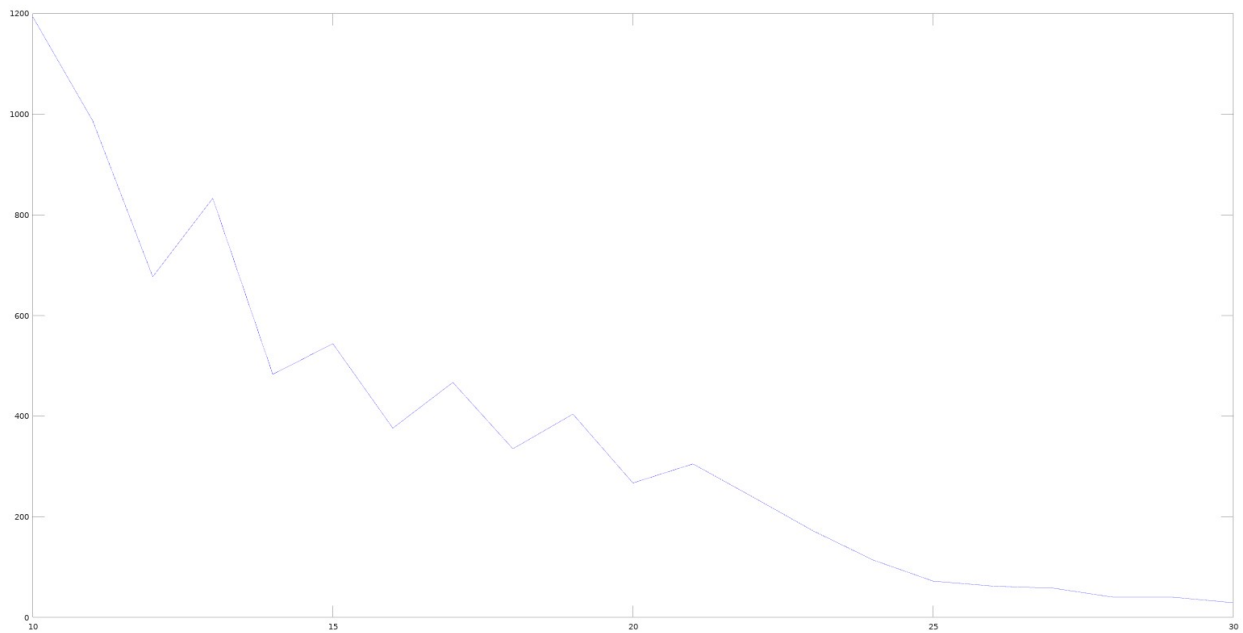
Για μήκος χρονολογίας 12, 677 άρθρα
173.29364429700036 seconds

.

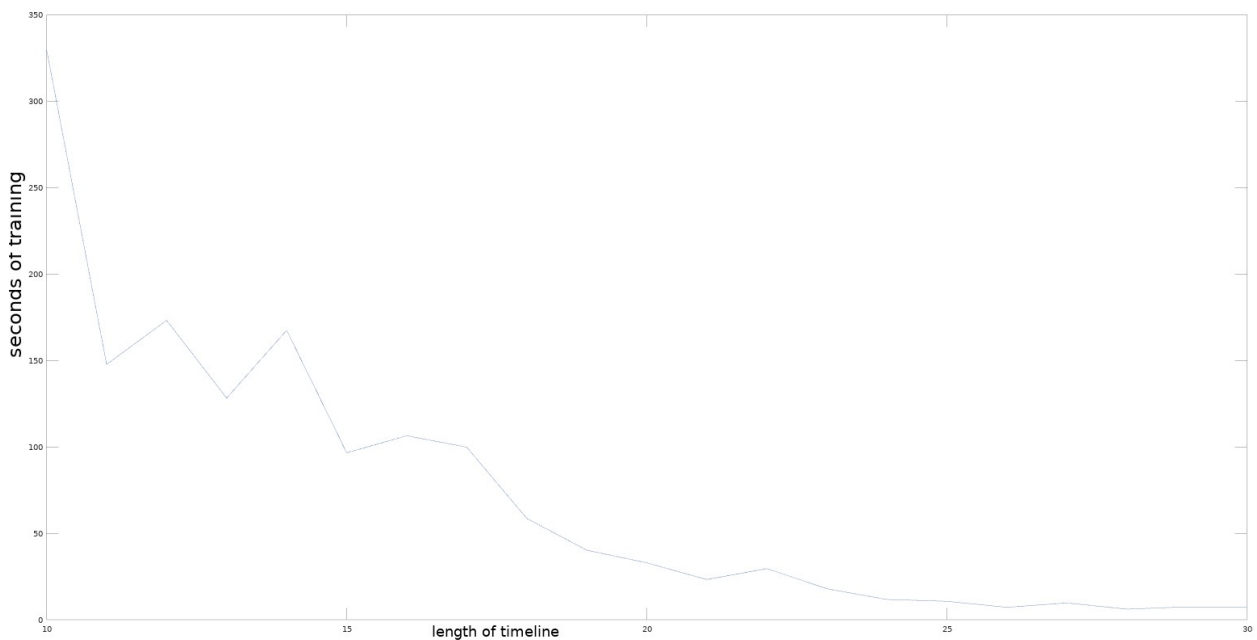
Για μήκος χρονολογίας 29, 40 άρθρα
7.437733655999182 seconds

Για μήκος χρονολογίας 30, 29 άρθρα
7.076767807999204 seconds

Στο παρακάτω διάγραμμα είναι το μήκος των χρονοσειρών(10,11,12...30) προς τον αριθμό των άρθρων που έχει κάθε χρονοσειρά.



Στο παρακάτω διάγραμμα είναι το μήκος των χρονοσειρών(10,11,12...30) προς τον χρόνο εκπαίδευσής του εκάστοτε μοντέλο.



5)Χρόνος απόκρισης του δικτύου

Ο χρόνος απόκρισης του δικτύου, δηλαδή το διάβασμα ενός αρχείου και η απάντηση του δικτύου σαν πρόβλεψη για χρονοσειρά των:

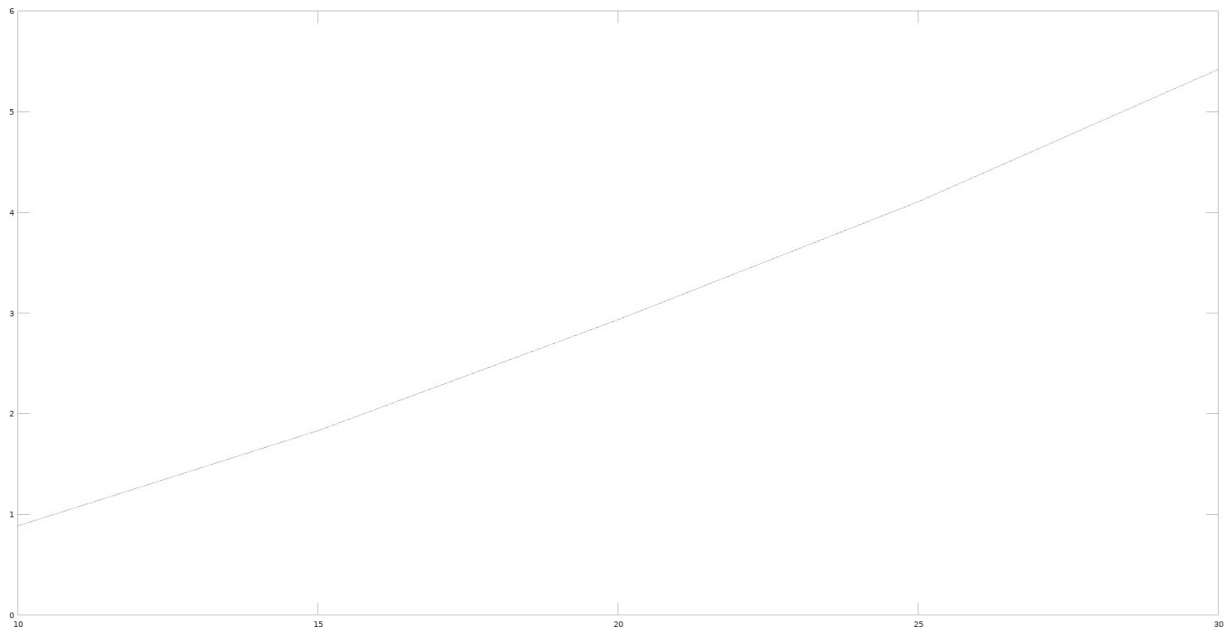
10 → 0.8857718209983432 seconds

15 → 1.8314685660006944 seconds

20 → 2.935860141999001 seconds

25 → 4.106523941001797 seconds

30 → 5.422715535998577 seconds



6) Τελικές εκπαιδευμένες τιμές των παραμέτρων

learning rate = 0.001

beta_1 = 0.9

beta_2 = 0.999

epsilon=0.00000001

decay=0.0

7) Αλγόριθμος εκπαίδευσης

Adam optimization algorithm

$V_{dw}=0, S_{dw}=0, V_{db}=0, S_{db}=0$

On iteration t :

Compute dW, db using causal mini-batch

$$V_{dw} = \beta_1 V_{dw} + (1 - \beta_1) dw, V_{db} = \beta_1 V_{db} + (1 - \beta_1) db$$

$$S_{dw} = \beta_2 S_{dw} + (1 - \beta_2) dw^2, S_{db} = \beta_2 S_{db} + (1 - \beta_2) db^2$$

$$V_{dw}^{corrected} = \frac{V_{dw}}{1 - \beta_1^t}, V_{db}^{corrected} = \frac{V_{db}}{1 - \beta_1^t}$$

$$S_{dw}^{corrected} = \frac{S_{dw}}{1 - \beta_2^t}, S_{db}^{corrected} = \frac{S_{db}}{1 - \beta_2^t}$$

$$W = W - a \frac{V_{dw}^{corrected}}{\sqrt{(S_{dw}^{corrected}) + \epsilon}}, b = b - a \frac{V_{db}^{corrected}}{\sqrt{(S_{db}^{corrected}) + \epsilon}}$$

ΠΗΓΕΣ:

Adam optimization algorithm.

-[https://www.youtube.com/watch?](https://www.youtube.com/watch?v=JXQT_vxqwIs&fbclid=IwAR1lHXiNhduMfPWXXKXTCJYoCIGH8XWcHgMO_D5EWYwUU1bFLqeqXQed064s)

[v=JXQT_vxqwIs&fbclid=IwAR1lHXiNhduMfPWXXKXTCJYoCIGH8XWcHgMO_D5EWYwUU1bFLqeqXQed064s](https://www.youtube.com/watch?v=JXQT_vxqwIs&fbclid=IwAR1lHXiNhduMfPWXXKXTCJYoCIGH8XWcHgMO_D5EWYwUU1bFLqeqXQed064s)

Παραδείγματα για υλοποίηση ενός lstm recurrent network σε keras.

-<https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/?fbclid=IwAR2qQE6zteIqyXHNtbNOoioz74qRAWadTah5VV0NV3A4N67Pnm9823jNYpQ>