ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ & ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών
ΙΔΡΥΘΕΝ ΤΟ 1837

Νευρωνικά Δίκτυα σε Python

Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

3η Προγραμματιστική Εργασία

ΔΗΜΗΤΡΗΣ ΧΑΜΑΡΙΑΣ - 1115201600190

ΑΝΤΩΝΙΑ ΡΟΥΣΣΟΥ - 1115201600147

Χειμερινό εξάμηνο 2021-2022

Εισαγωγή

Η εργασία αποτελείται από τέσσερις ενότητες με σκοπό i) την πρόγνωση τιμών μετοχών με χρήση αναδρομικού νευρωνικού δικτύου LSTM, ii) την ανίχνευση ανωμαλιών με χρήση νευρωνικού δικτύου αυτοκωδικοποιήσης LSTM ανά μετοχή, iii) την αναζήτηση και συσταδοποίηση μετοχών βάσει της αναπαράστασης των χρονοσειρών που προκύπτει από συνελικτική αυτοκωδικοποίηση και iv) τη σύγκριση αποτελεσμάτων αναζήτησης και συσταδοποίησης με την αρχική αναπαράσταση.

Δομή αρχείων

Αρχεία

- forecast.py (LSTM πολλαπλών στρωμάτων για πρόγνωση τιμών)
- forecast.ipynb (Αντίστοιχο notebook του forecast με plots)
- detect.py (LSTM αυτοκωδικοποίησης για ανίχνευση ανωμαλιών)
- detect.ipynb (Αντίστοιχο notebook του detect με plots)
- reduce.py (Autoencoder)
- reduce.ipynb (Αντίστοιχο notebook του autoencoder)
- functions.py (Γενικές λειτουργίες + Κατασκευή μοντέλων)

Φάκελοι

- models (Περιέχει τα αποθηκευμένα εκπαιδευμένα μοντέλα)

Νευρωνικό δίκτυο LSTM πολλαπλών στρωμάτων για πρόγνωση τιμών μετοχών

Περιγραφή προγράμματος

Το πρόγραμμα (***forecast.py***) κατασκευάζει ένα αναδρομικό νευρωνικό δίκτυο LSTM πολλαπλών στρωμάτων, το οποίο χρησιμοποιείται για την πρόγνωση τιμών μετοχών. Με βάση τις κατάλληλες υπερπαραμέτρους [*αριθμού στρωμάτων, μεγέθους στρωμάτων, αριθμού εποχών εκπαίδευσης (epochs), μεγέθους δέσμης (batch size)*] επιλέχθηκε το βέλτιστο μοντέλο που ελαχιστοποιεί το σφάλμα. Η εκπαίδευση του

μοντέλου έγινε ανά χρονοσειρά και ανά σύνολο χρονοσειρών. Τέλος, τα αποτελέσματα οπτικοποιούνται τόσο ως προς την εξέλιξη των πραγματικών τιμών όσο και ως προς την εξέλιξη των τιμών βάσει της πρόγνωσης με τις αντίστοιχες γραφικές παραστάσεις.

Οδηγίες χρήσης προγράμματος

Run:

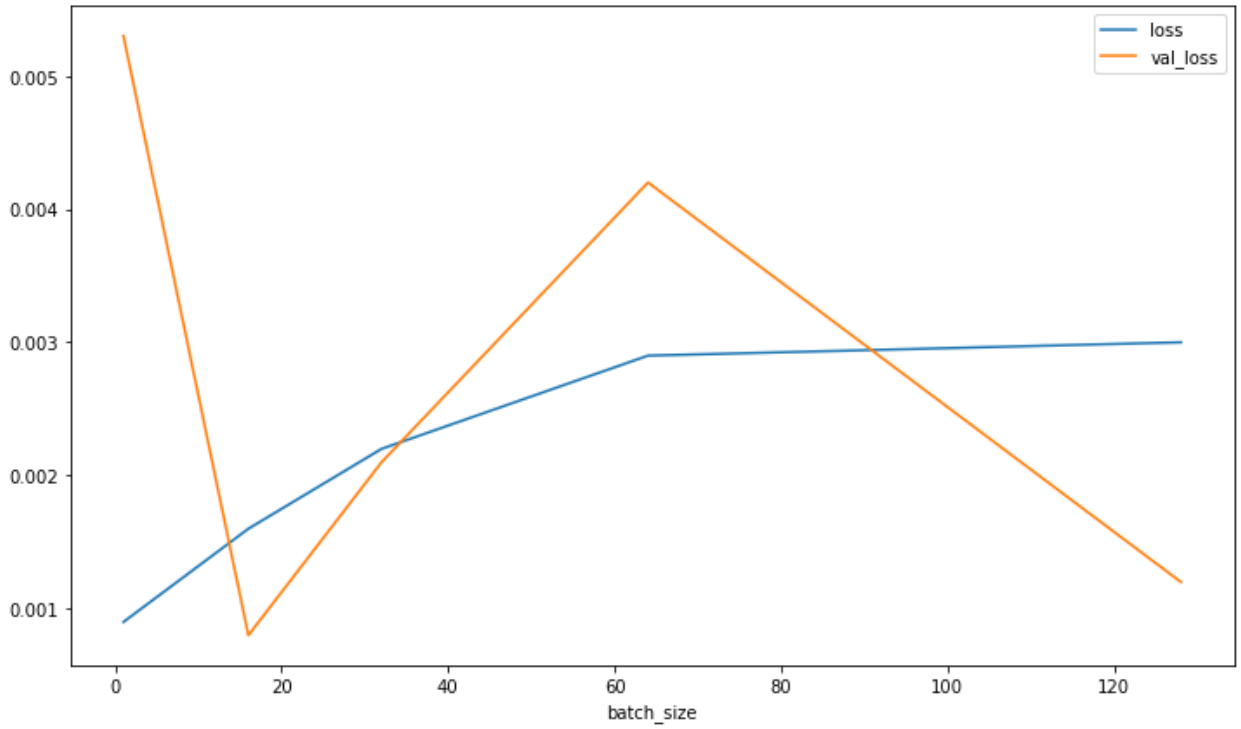
```
python forecast.py -d <dataset> -n <number of time series selected>
```

Αποτελέσματα πειραμάτων

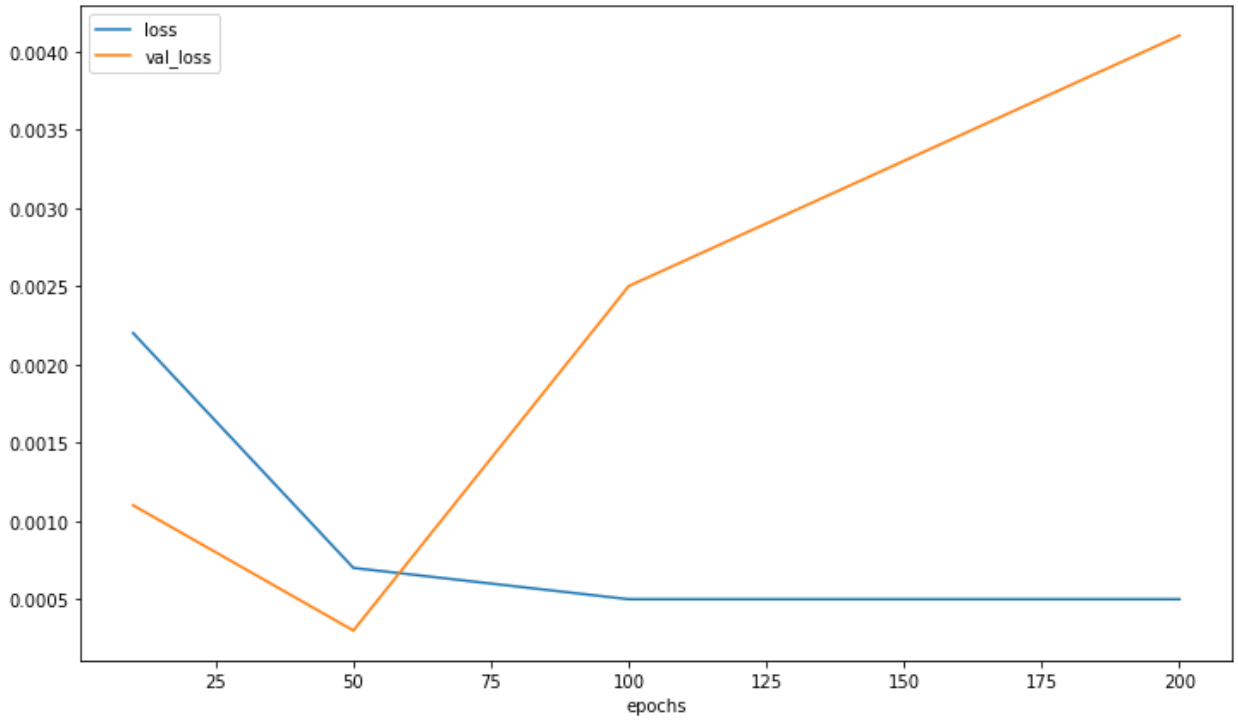
Layers	Units	Epochs	Batch Size	Lookback	Train Loss	Val Loss
4	50	10	1	60	0.0009	0.0053
4	50	10	16	60	0.0016	0.0008
4	50	10	32	60	0.0022	0.0021
4	50	10	64	60	0.0029	0.0042
4	50	10	128	60	0.0030	0.0012
4	50	10	32	60	0.0022	0.0011
4	50	50	32	60	0.0007	0.0003
4	50	100	32	60	0.0005	0.0025
4	50	200	32	60	0.0005	0.0041
3	50	10	32	60	0.0020	0.0018
4	50	10	32	60	0.0022	0.0085
5	50	10	32	60	0.0023	0.0057
6	50	10	32	60	0.0026	0.0012
4	16	10	32	60	0.0038	0.0018
4	50	10	32	60	0.0022	0.0010
4	64	10	32	60	0.0018	0.0020

4	128	10	32	60	0.0014	0.0010
4	256	10	32	60	0.0014	0.0051
4	512	10	32	60	0.0009	0.0009
4	50	10	16	1	0.0014	0.0014
4	50	10	16	10	0.0018	0.0010
4	50	10	16	30	0.0017	0.0020
4	50	10	16	60	0.0017	0.0013
4	50	10	16	120	0.0016	0.0026

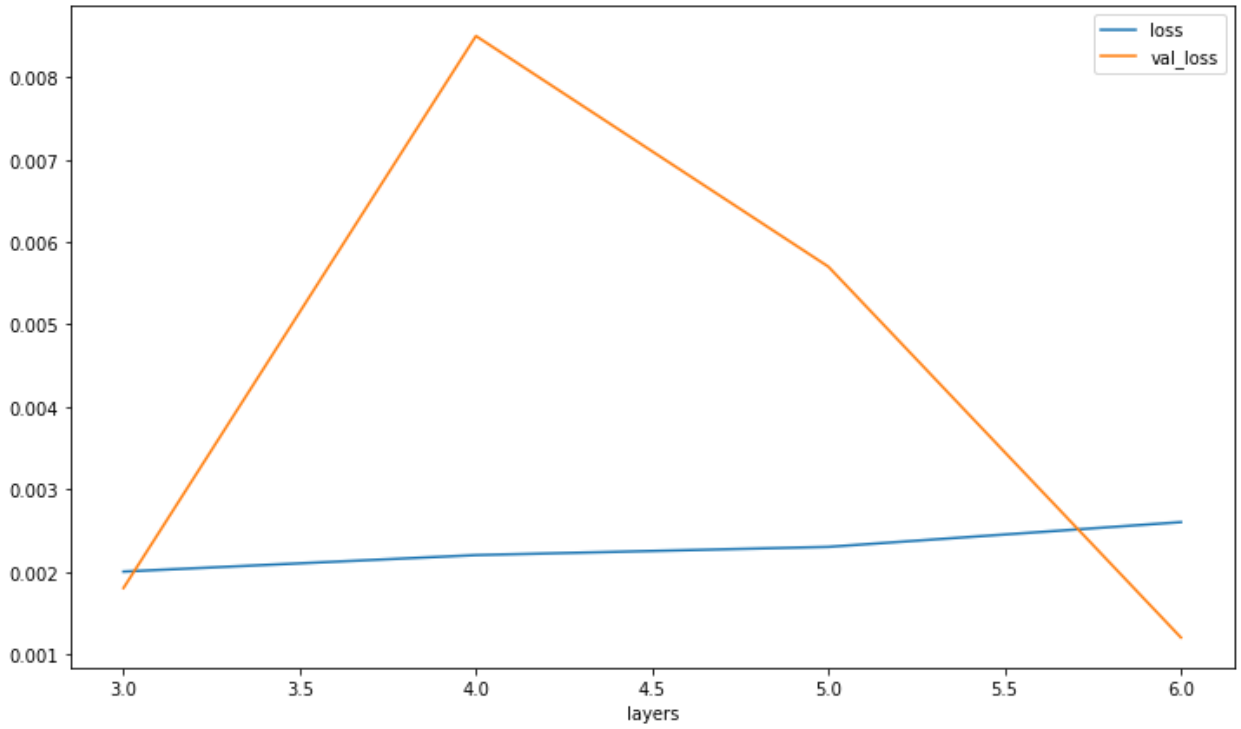
Σχολιασμός αποτελεσμάτων



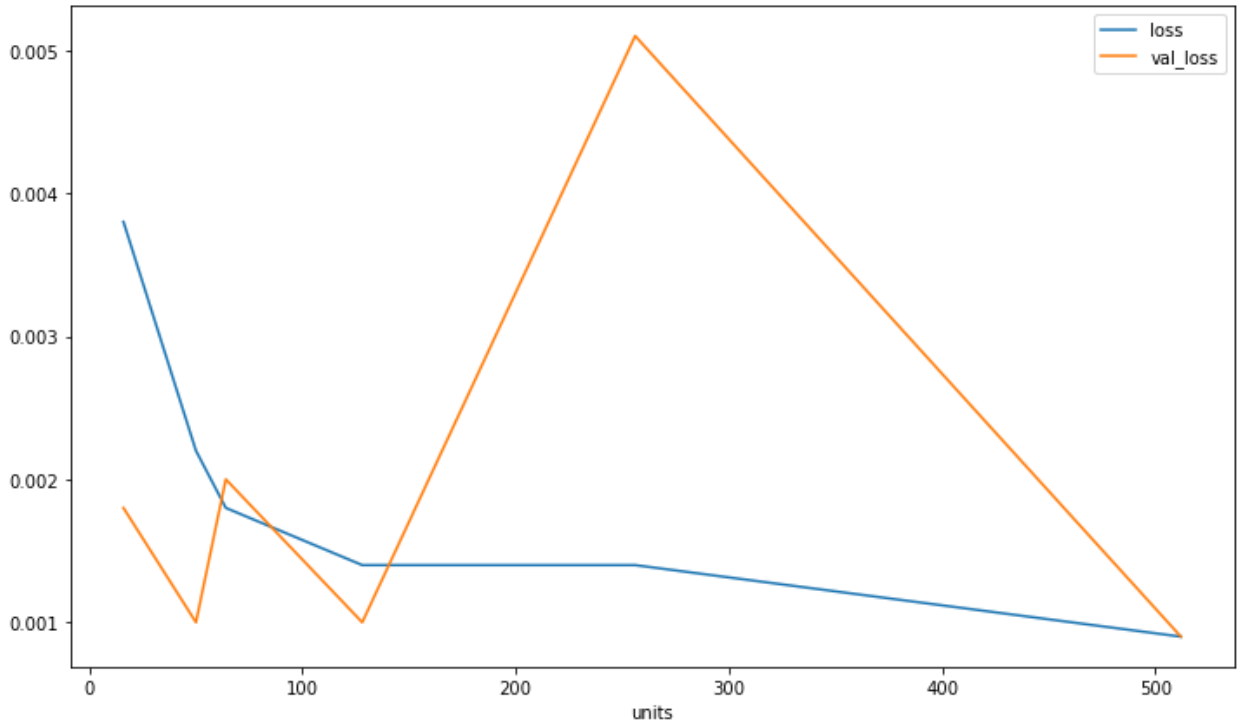
Από το γράφημα παρατηρούμε οτι η καλύτερη τιμή για batch size είναι 128 καθώς αποκρίνεται καλά σε νέα δεδομένα (χαμηλό val_loss) και επειδή αποφεύγει το overfitting στο training set. Επιπλέον ο χρόνος εκπαίδευσης είναι και χαμηλότερος από το batch size 16 που έχει αντίστοιχα αποτελέσματα.



Από το γράφημα παρατηρούμε ότι η καλύτερη τιμή για epochs είναι 50 καθώς αποκρίνεται καλά σε νέα δεδομένα (χαμηλό val_loss) και επειδή αποφεύγει το overfitting στο training set. Όμως μπορούμε να πετύχουμε παρόμοια αποτελέσματα σε λιγότερο χρόνο αν επιλέξουμε την τιμή 10.

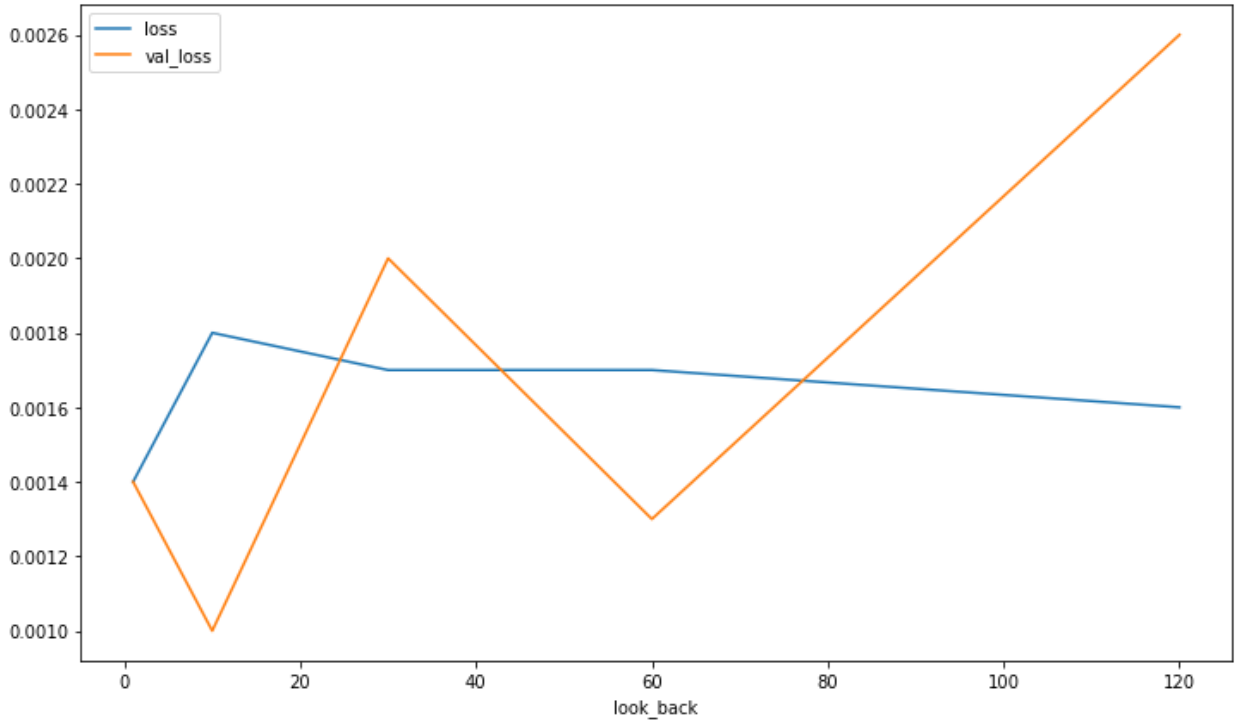


Από το γράφημα παρατηρούμε ότι η καλύτερη τιμή για layers είναι 6 καθώς αποκρίνεται καλά σε νέα δεδομένα (χαμηλό val_loss) και επειδή αποφεύγει το overfitting στο training set.



Από το γράφημα παρατηρούμε ότι η καλύτερη τιμή για units είναι 512 καθώς αποκρίνεται καλά

σε νέα δεδομένα (χαμηλό val_loss) και επειδή αποφεύγει το overfitting στο training set.



Από το γράφημα παρατηρούμε ότι η καλύτερη τιμή για look back είναι 60 καθώς αποκρίνεται καλά σε νέα δεδομένα (χαμηλό val_loss) και επειδή αποφεύγει το overfitting στο training set.

Νευρωνικό δίκτυο LSTM αυτοκωδικοποίησης χρονοσειρών για ανίχνευση ανωμαλιών

Περιγραφή προγράμματος

Το πρόγραμμα (***detect.py***) κατασκευάζει ένα συνελικτικό νευρωνικό δίκτυο αυτοκωδικοποίησης χρονοσειρών το οποίο περιλαμβάνει στρώματα συμπίεσης και αποσυμπίεσης με σκοπό την ανίχνευση ανωμαλιών. Μετά από κατάλληλη επιλογή των υπερπαραμέτρων [*αριθμού στρωμάτων, μεγέθους στρωμάτων, αριθμού εποχών εκπαίδευσης (epochs), μεγέθους δέσμης (batch size), στρώματα dropout*] καθώς και του κατάλληλου κατωφλιού, επιλέχθηκε η βέλτιστη δομή για το νευρωνικό δίκτυο.

Οδηγίες χρήσης προγράμματος

Run:

```
python detect.py -d <dataset> -n <number of time series selected> -mae
```

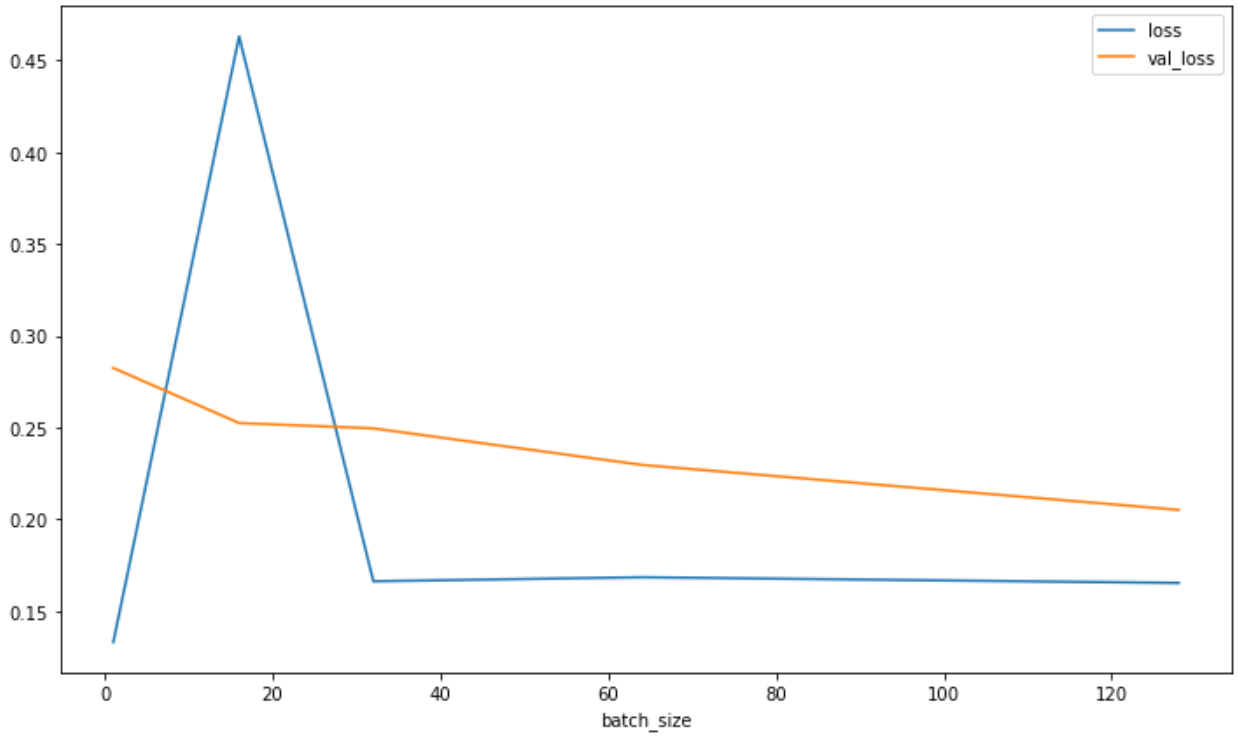
```
<error value as double>
```

Αποτελέσματα πειραμάτων

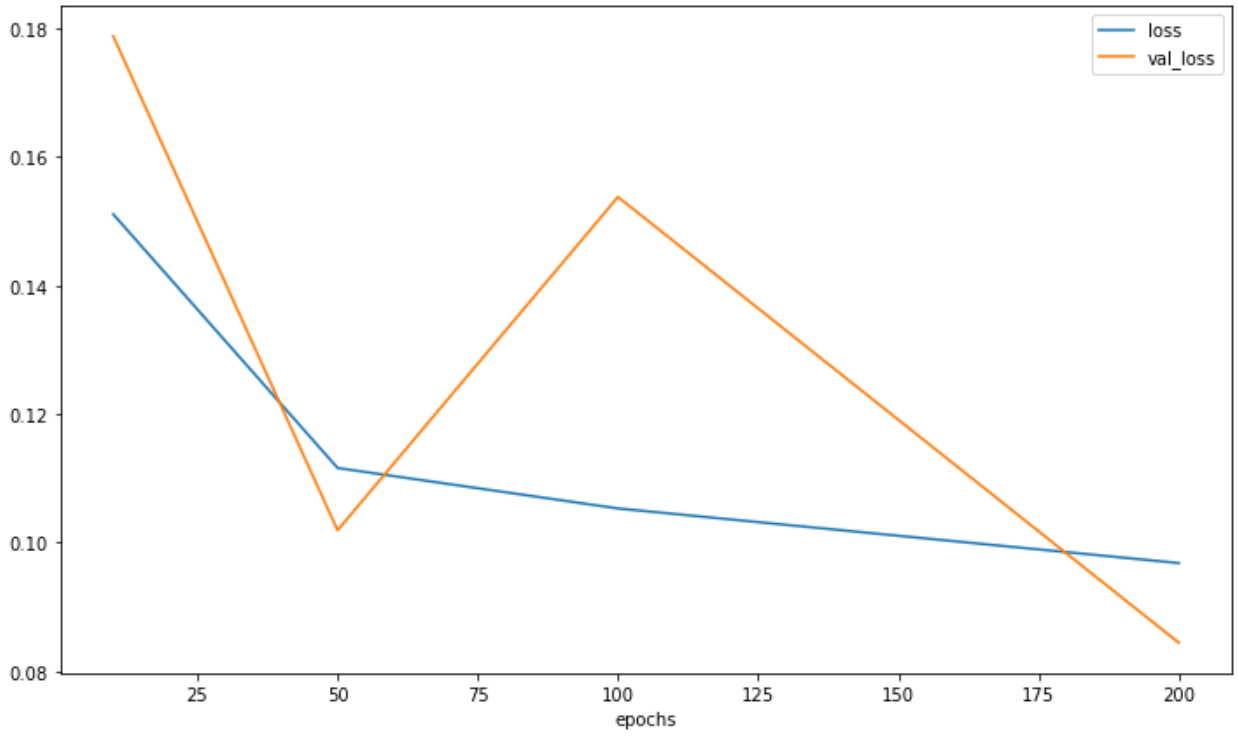
Layers	Units	Epochs	Batch Size	Lookback	Train Loss	Val Loss
2	64	10	1	30	0.1331	0.2825
2	64	10	16	30	0.4631	0.2525
2	64	10	32	30	0.1663	0.2496
2	64	10	64	30	0.1685	0.2297
2	64	10	128	30	0.1654	0.2052
2	64	10	128	30	0.1511	0.1788
2	64	10	128	30	0.1116	0.1019
2	64	10	128	30	0.1053	0.1538
2	64	10	128	30	0.0968	0.0844
2	16	10	128	30	0.3049	0.5212
2	64	10	128	30	0.1562	0.1997
2	128	10	128	30	0.1316	0.1409
2	256	10	128	30	0.1458	0.1347
2	512	10	128	30	0.1357	0.1143
2	64	10	128	1	0.2289	0.2183
2	64	10	128	10	0.2196	0.2597
2	64	10	128	30	0.1538	0.1800
2	64	10	128	60	0.1414	0.1481
2	64	10	128	120	0.1583	0.1391
2	64	10	128	30	0.1722	0.2202
3	64	10	128	30	0.1743	0.2622

4	64	10	128	30	0.3710	0.3796
5	64	10	128	30	0.3791	0.4743

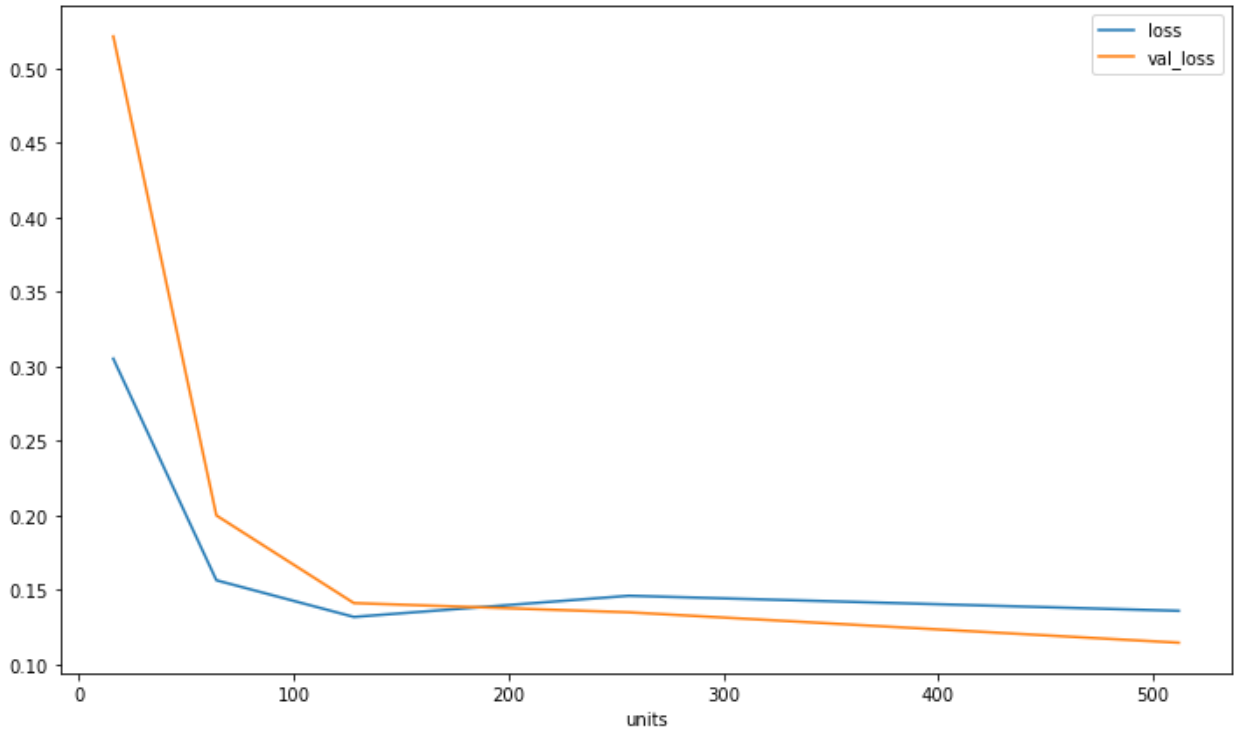
Σχολιασμός αποτελεσμάτων



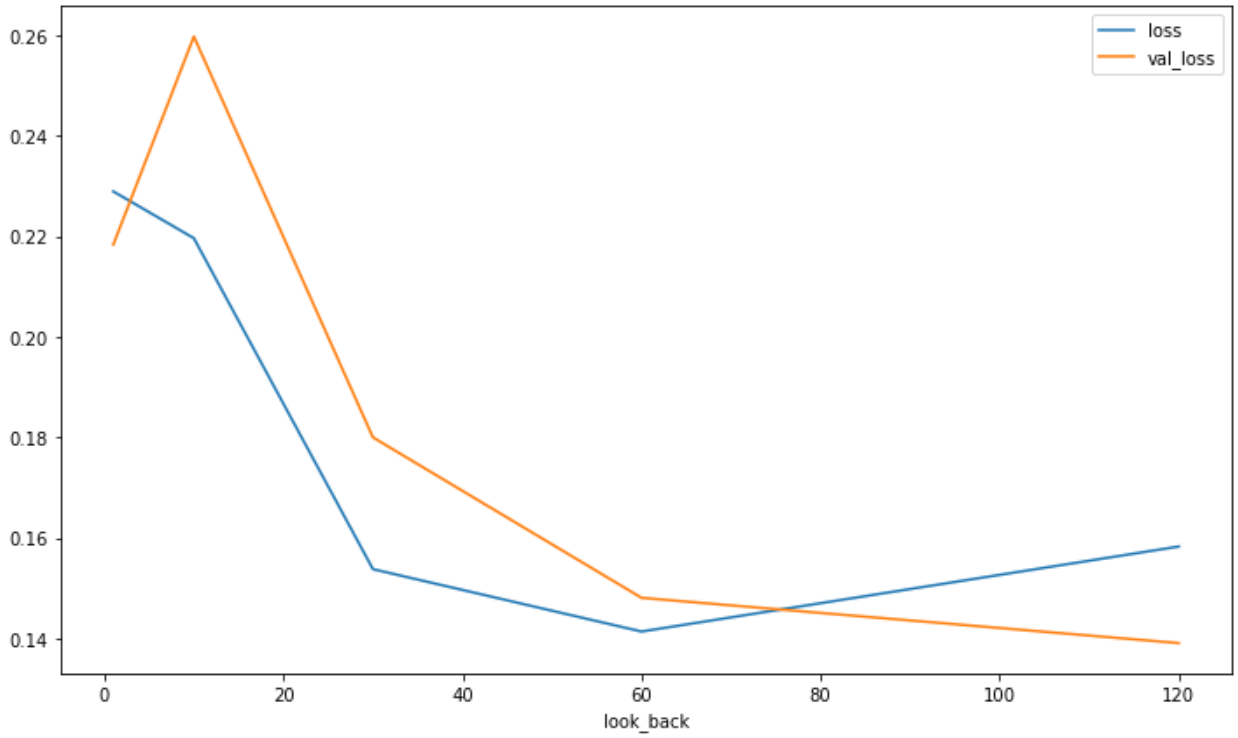
Από το γράφημα παρατηρούμε οτι η καλύτερη τιμή για batch size είναι 128 καθώς αποκρίνεται καλά σε νέα δεδομένα (χαμηλό val_loss) και επειδή αποφεύγει το overfitting στο training set.



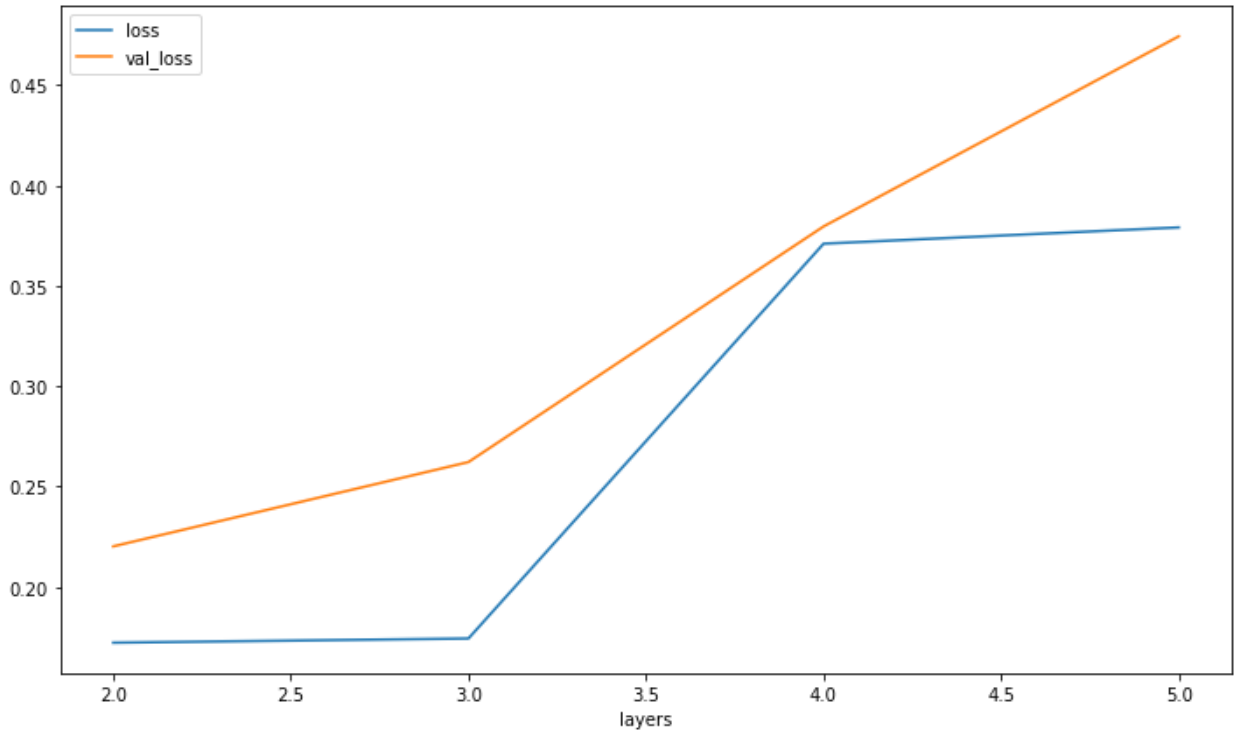
Από το γράφημα παρατηρούμε ότι η καλύτερη τιμή για epochs είναι 200 καθώς αποκρίνεται καλά σε νέα δεδομένα (χαμηλό val_loss) και επειδή αποφεύγει το overfitting στο training set.



Από το γράφημα παρατηρούμε ότι η καλύτερη τιμή για units είναι 512 καθώς αποκρίνεται καλά σε νέα δεδομένα (χαμηλό val_loss) και επειδή αποφεύγει το overfitting στο training set.



Από το γράφημα παρατηρούμε ότι η καλύτερη τιμή για look back είναι 120 καθώς αποκρίνεται καλά σε νέα δεδομένα (χαμηλό val_loss) και επειδή αποφεύγει το overfitting στο training set.



Από το γράφημα παρατηρούμε ότι η καλύτερη τιμή για layers είναι 2 καθώς αποκρίνεται καλά σε νέα δεδομένα (χαμηλό val_loss).

Συνελικτικό νευρωνικό δίκτυο αυτοκωδικοποίησης χρονοσειρών

Περιγραφή προγράμματος

Το πρόγραμμα (***reduce.py***) κατασκευάζει ένα συνελικτικό νευρωνικό δίκτυο αυτοκωδικοποίησης χρονοσειρών το οποίο περιλαμβάνει στρώματα συμπίεσης και αποσυμπίεσης. Μετά από κατάλληλη επιλογή των υπερπαραμέτρων [*αριθμού συνελικτικών στρωμάτων, μεγέθους συνελικτικών φίλτρων, αριθμού εποχών εκπαίδευσης (epochs), μεγέθους δέσμης (batch size), διάστασης συμπίεσης*] καθώς και του κατάλληλου κατωφλιού, επιλέχθηκε η βέλτιστη δομή για το νευρωνικό δίκτυο. Το εκπαιδευμένο μοντέλο χρησιμοποιείται για την παραγωγή νέων αναπαραστάσεων του συνόλου μετοχών, οι οποίες αποθηκεύονται σε tab-separated αρχείο.

Οδηγίες χρήσης προγράμματος

Run:

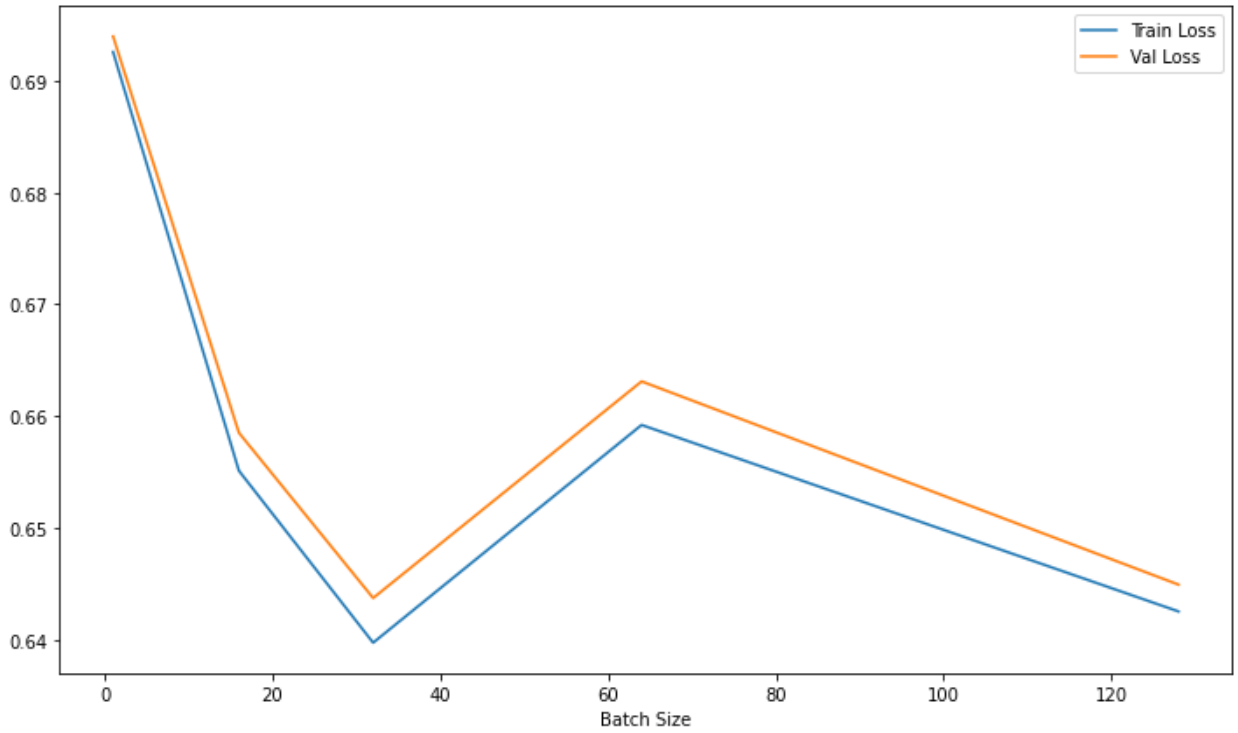
```
python reduce.py -d <dataset> -q <queryset> -od <output_dataset_file> -oq
<output_query_file>
```

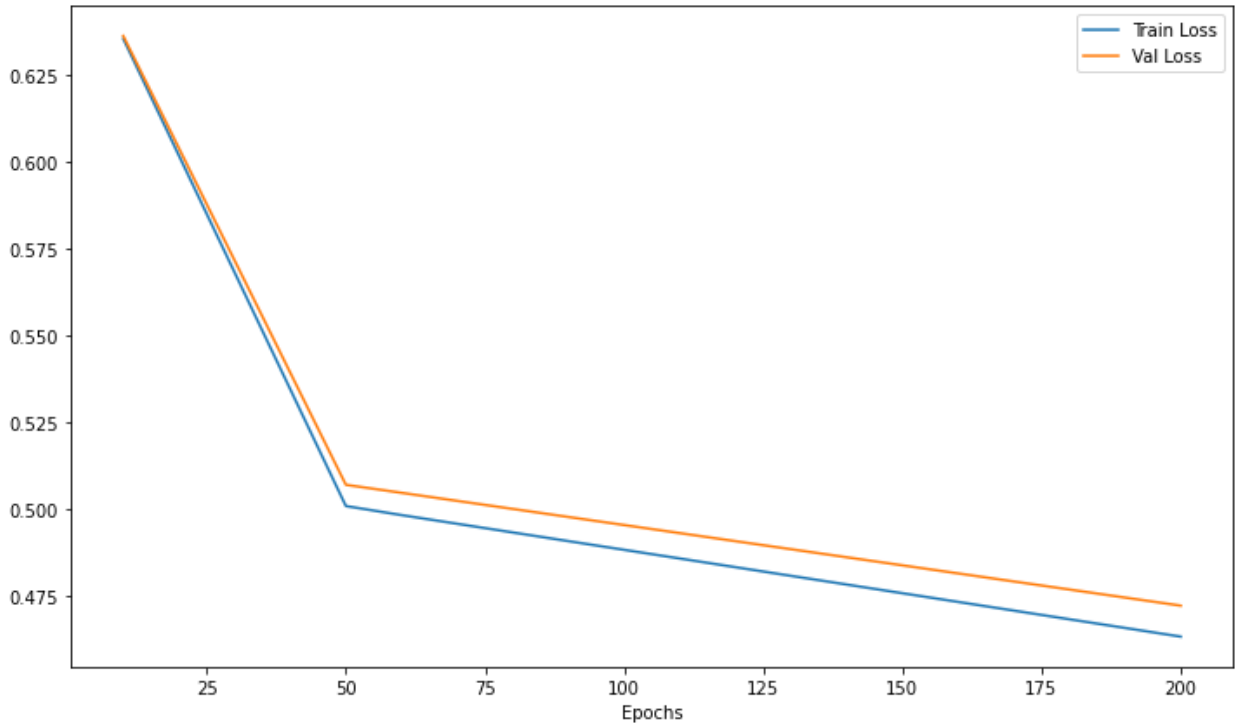
Αποτελέσματα πειραμάτων

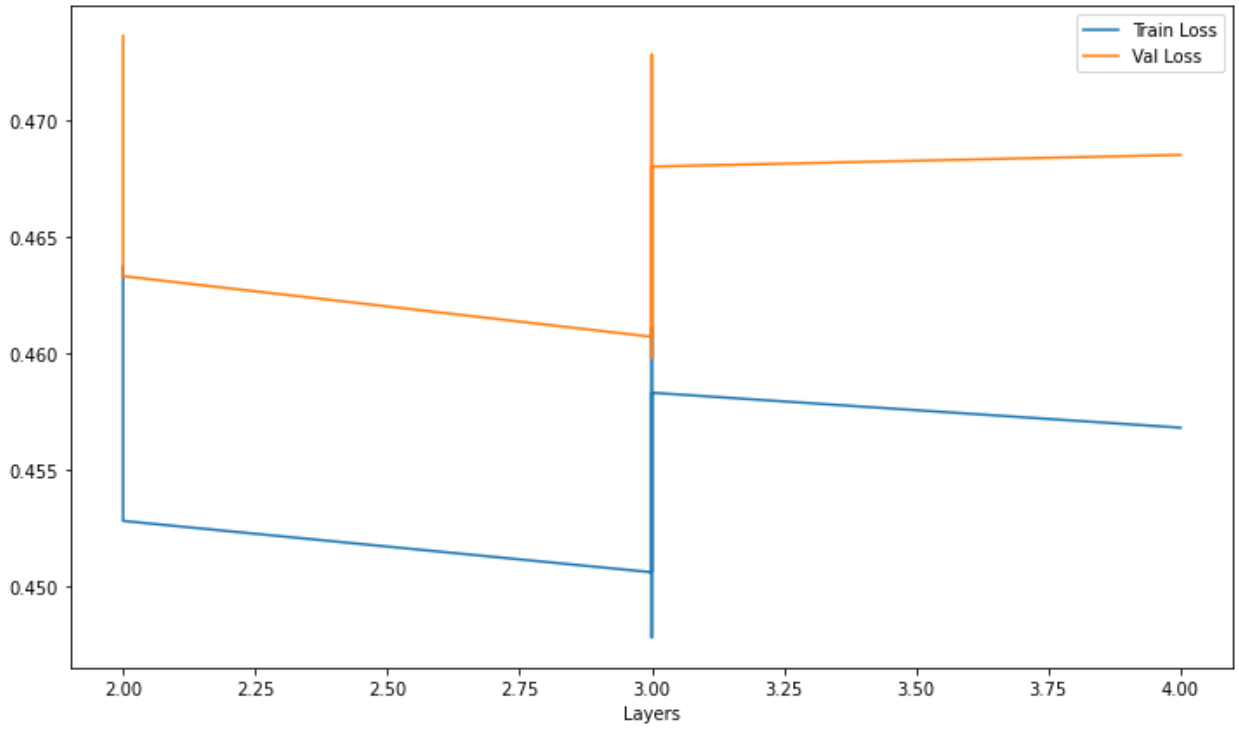
Layers	Encoding dimension	Epochs	Filters	Kernel Size	Batch Size	Train Loss	Val Loss
2	3	100	[16, 1]	3	1	0.6926	0.6940
2	3	100	[16, 1]	3	16	0.6551	0.6585
2	3	100	[16, 1]	3	32	0.6397	0.6437
2	3	100	[16, 1]	3	64	0.6592	0.6631
2	3	100	[16, 1]	3	128	0.6425	0.6449
2	3	100	[16, 16]	3	256	0.5549	0.5613
2	3	10	[32, 32]	3	128	0.6352	0.6360
2	3	50	[32, 32]	3	128	0.5010	0.5071
2	3	60	[32, 32]	3	128	0.4984	0.5048
2	3	200	[32, 32]	3	128	0.4635	0.4724
2	3	100	[32, 64]	3	32	0.4590	0.4679
2	3	100	[16, 32]	3	32	0.4637	0.4736
2	3	200	[32, 64]	3	32	0.4528	0.4633
3	3	100	[32, 64, 128]	3	32	0.4506	0.4607
3	3	100	[32, 64, 128]	3	128	0.4611	0.4728
3	3	200	[32, 64, 128]	3	32	0.4478	0.4598
3	3	100	[32, 64, 128]	3	128	0.4583	0.4680
4	3	100	[32, 64, 128,	4	128	0.4568	0.4685

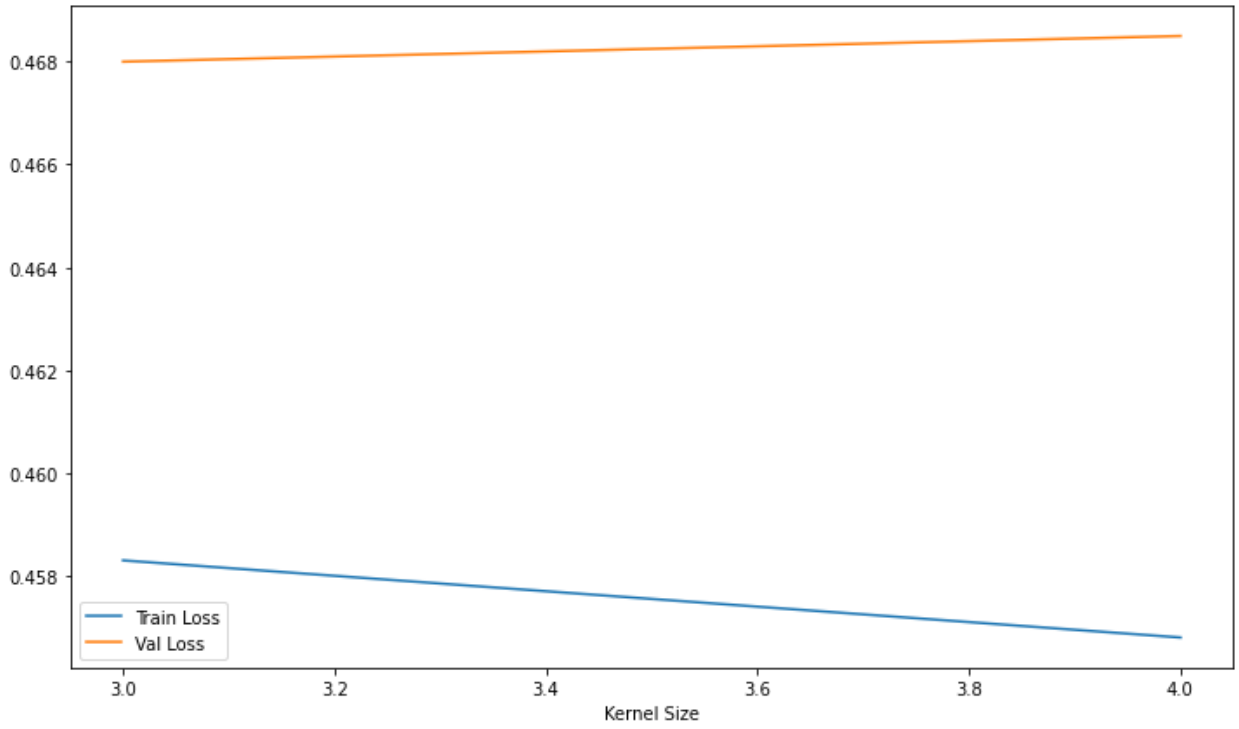
			256]				
4	3	200	[32, 64, 128, 256]	4	128	0.4491	0.4604
4	3	200	[32, 64, 128, 256]	4	32	0.4461	0.4582
4	3	100	[32, 64, 128, 256]	4	32	0.4481	0.4601
4	3	200	[32, 64, 128, 256]	3	32	0.4459	0.4583
4	3	100	[32, 64, 128, 256]	3	128	0.4539	0.4631
4	2	100	[32, 64, 128, 256]	4	32	0.4194	0.4135
4	2	100	[32, 64, 128, 256]	4	128	0.4335	0.4248
4	2	200	[32, 64, 128, 256]	4	128	0.4215	0.4154

Σχολιασμός αποτελεσμάτων






Train Loss
Val Loss
0.470
0.465
0.460
0.455
0.450
2.00
2.25
2.50
2.75
3.00
3.25
3.50
3.75
4.00
Layers


0.468
0.466
0.464
0.462
0.460
0.458
Train Loss
Val Loss
3.0
3.2
3.4
3.6
3.8
4.0
Kernel Size

Github repository:

https://github.com/DImiTrisXam/algo_project1/tree/main/project3