

Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα - Εργασία 3

Κωνσταντίνα Έλληνα - 1115201600046

Απόστολος Λύρας - 1115201600097

Παρατηρήσεις για Ερώτημα Α

Τα πειράματα που κάναμε για το συγκεκριμένο ερώτημα αφορούν τις υπερπαραμέτρους που ζητούνται από την εκφώνηση και παρακάτω παραθέτουμε τις παρατηρήσεις μας. Τα πειράματα έχουν γίνει για $n = 1$ και έχουμε βάλει early stopping, έτσι ώστε να σταματάει το μοντέλο όταν δεν έχει βελτιωθεί το validation loss για 3 συνεχόμενες φορές. Το αρχείο των χρονοσειρών που χρησιμοποιήσαμε είναι το 'nasdaq2007_17.csv'.

Delay

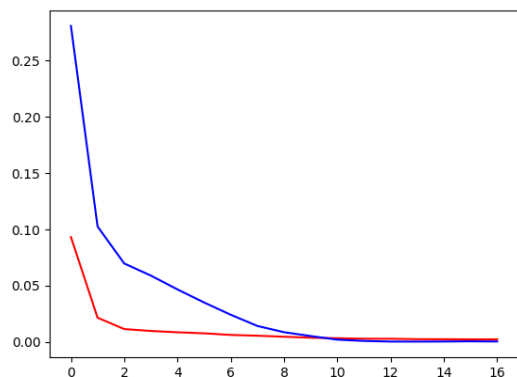
Αρχικά θέλαμε να δούμε τις διαφορές που θα έχει το εκτελέσιμο με διαφορετικό delay. Γι' αυτό το λόγο αφήσαμε σταθερές τις υπόλοιπες παραμέτρους και είδαμε τα εξής:

Με lstm_units = 8, lstm_layers = 1, dropout = 0.1, batch_size = 32, max_epochs = 20

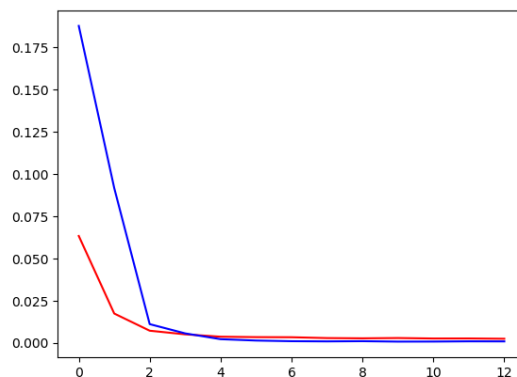
Delay	1	2	10	50	100	200
Test loss	0.00068	0.00093	0.02204	0.01161	0.01859	0.01067

Βλέπουμε επίσης και την καλυτέρευση του **validation loss**, όσο αυξάνεται το delay, όπου Blue line = validation loss, Red line = Train loss.

Delay = 1

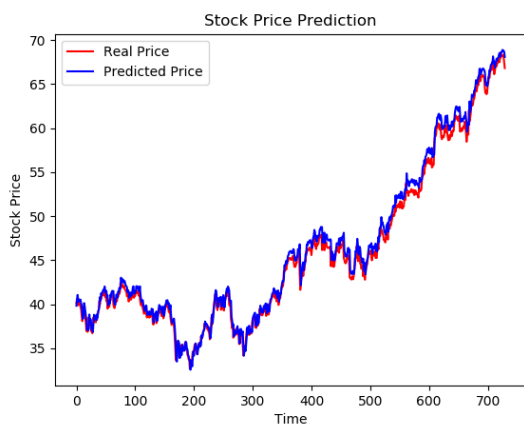


Delay = 200

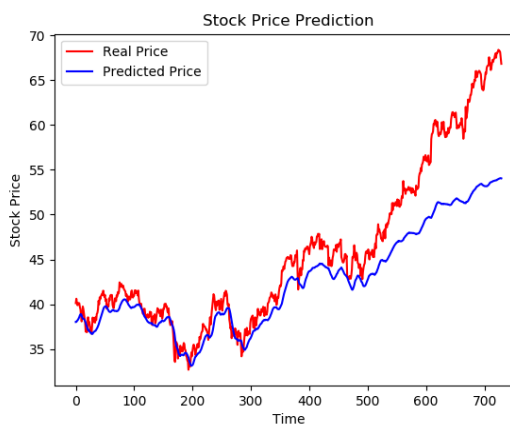


Παρακάτω φαίνεται πώς αλλάζει το **prediction** ανάλογα με την τιμή του delay. Ξεκινάει η τιμή του από 1 που βλέπουμε ότι ακολουθεί ακριβώς το μοντέλο, ύστερα με delay = 10 έως και delay = 100 η ακρίβεια δεν είναι τόσο καλή αλλά εξακολουθεί να κρατάει τη μορφή του μοντέλου και τέλος με delay = 200 ξαναβρίσκει την ακολουθία και το prediction είναι πάρα πολύ καλό.

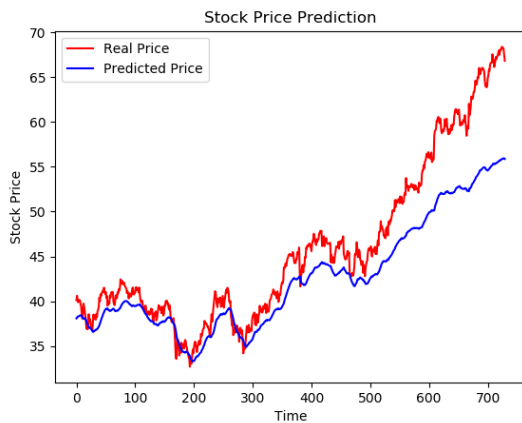
Delay = 1



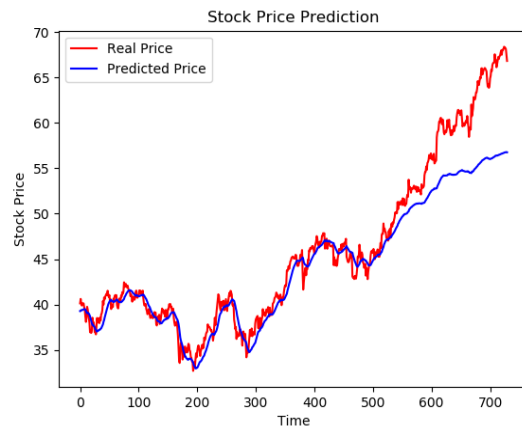
Delay = 10



Delay = 100



Delay = 200



Παρατηρούμε ότι η περίπτωση με $\text{delay}=1$ ακολουθεί σχεδόν ακριβώς το μοντέλο, ενώ η περίπτωση με $\text{delay}=200$ ακολουθεί το μοντέλο αλλά δεν υπάρχει η τόση μεγάλη ακρίβεια όσο στην πρώτη περίπτωση. Αυτό συμβαίνει διότι στο παράδειγμά μας έχουμε τιμές μετοχών, οι οποίες αλλάζουν ανά καιρούς. Οι διαφορές όμως μεταξύ 2 συνεχόμενων τιμών στην ίδια χρονοσειρά είναι πολύ μικρές, καθώς οι μετοχές δεν κάνουν τόσο μεγάλα άλματα αλλαγών τιμών. Οπότε το παράθυρο που βλέπει το συγκεκριμένο μοντέλο είναι 1 και βλέπει την προηγούμενη τιμή της χρονοσειράς, η οποία όπως είπαμε δεν έχει μεγάλη διαφορά από την τιμή που είμαστε τώρα. Άρα το μοντέλο ακολουθείται ακριβώς, σε αντίθεση με το παράθυρο 200 που βλέπει πολλές τιμές πίσω. Αυτό όμως δεν είναι καλό για κάποιο άλλο παράδειγμα που οι τιμές δεν είναι τόσο παρόμοιες σε μικρά διαστήματα και χρειάζεται να παρατηρήσουμε το “ιστορικό” της χρονοσειράς για να καταλάβουμε τη μορφή της. Επίσης, αν βάλουμε το μοντέλο αυτό να τρέχει για μελλοντικές τιμές τις οποίες δεν έχουμε μέχρι στιγμής και θέλουμε να προβλέψουμε τιμές μετοχών σε επόμενο χρονικό διάστημα δεν θα έχουμε σίγουρα σωστά αποτελέσματα, αφού θα λειτουργεί μόνο με predicted values, οι οποίες θα βλέπουν μόνο την προηγούμενη τιμή τους.

Άρα διαλέγουμε ως βέλτιστη τιμή του delay το 200.

LSTM units

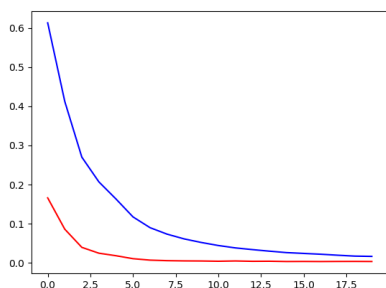
Με lstm_layers = 1, dropout = 0.1, max_epochs = 20, batch_size = 32, delay = 200

Units	2	8	16	32	64
Test loss	0.14590	0.02685	0.03303	0.00267	0.00245

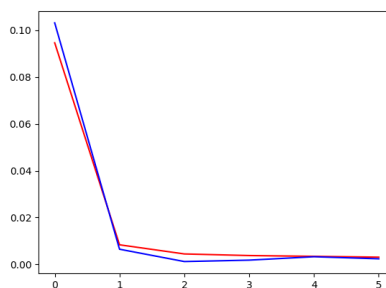
Όσο αυξάνονται τα units τόσο πιο λίγες εποχές χρειάζονται για να μειωθεί το loss και με το early stopping σταματάει η εκτέλεση.

Βλέπουμε τη διαφορά που κάνουν τα units και στο validation αλλά και στο prediction.
Το **validation** αλλάζει ως εξής:

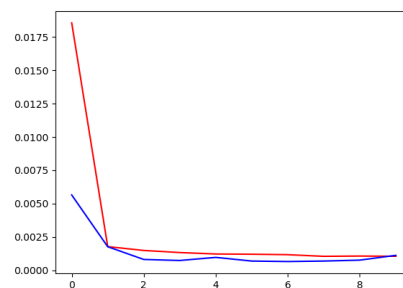
Units = 2



Units = 16

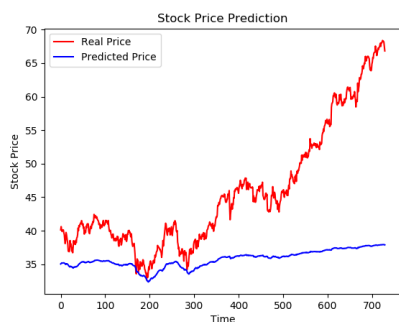


Units = 64

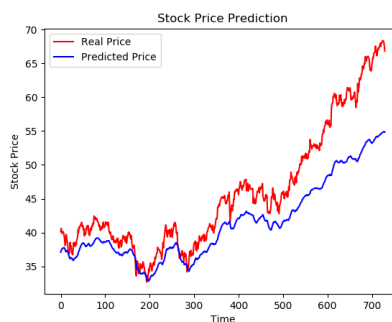


Ενώ βλέπουμε πόσο πολύ βελτιώνεται το **prediction**.

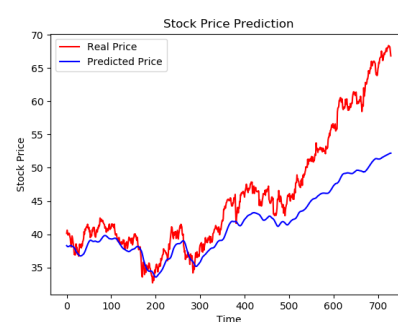
Units = 2



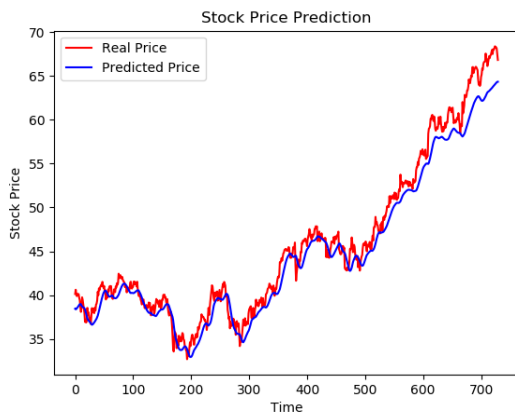
Units = 8



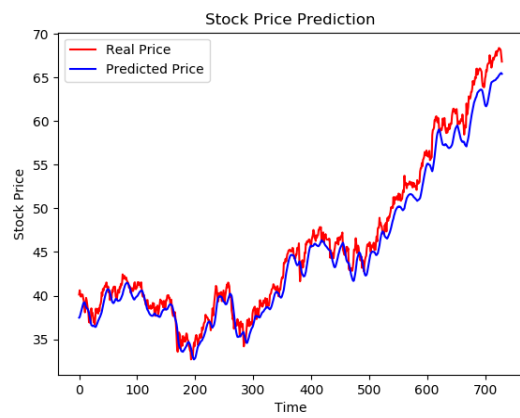
Units = 16



Units = 16



Units = 32



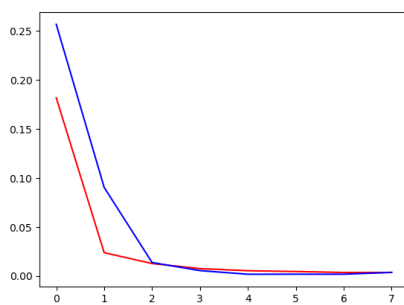
LSTM layers

Me lstm_units = 8, dropout = 0.1, max_epochs = 20, batch_size = 32, delay = 200

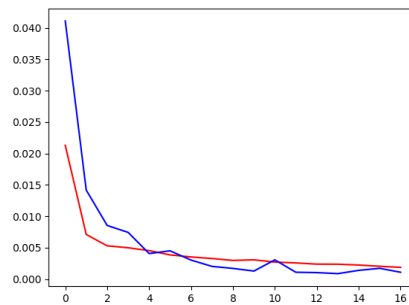
Layers	1	2	3
Test loss	0.03627	0.00861	0.04627

To validation loss:

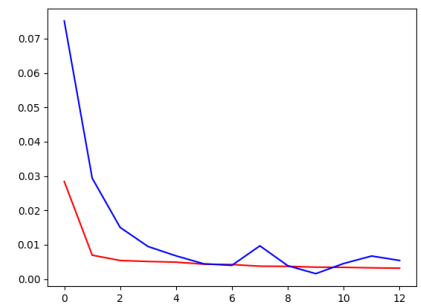
Layers = 1



Layers = 2

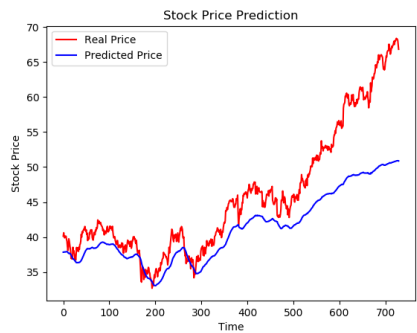


Layers = 3

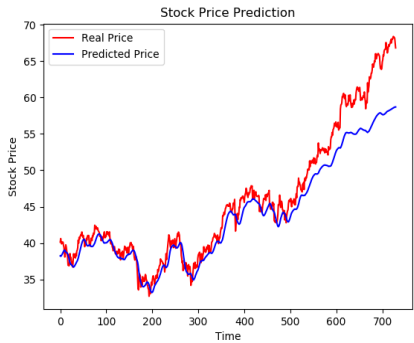


Παρατηρούμε ότι για 1 layer και για 3 layers το prediction ακολουθεί τη μορφή, αλλά δεν είναι πολύ ακριβές, ενώ για layers = 2 το prediction έχει βγει πολύ καλό.

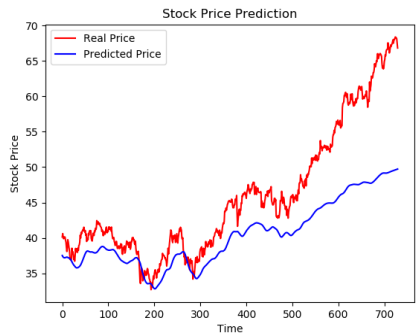
Layers = 1



Layers = 2



Layers = 3



Epochs

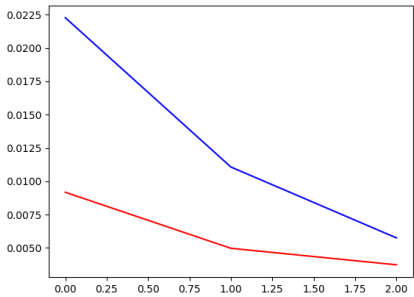
Με lstm_units = 8, lstm_layers = 1, dropout = 0.1, batch_size = 32, delay = 200

Όσο περισσότερες εποχές τόσο πιο πολύ μειώνεται το loss, καθώς κάνει περισσότερα περάσματα στο train set.

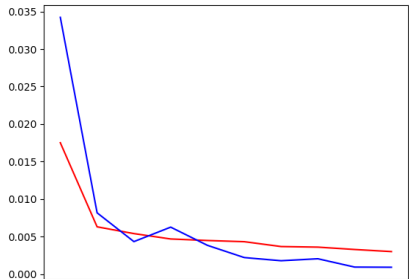
Epochs	1	3	10	40
Test loss	0.39935	0.06403	0.02355	0.00310

Το **validation** με epochs=3, epochs=10, epochs=40.

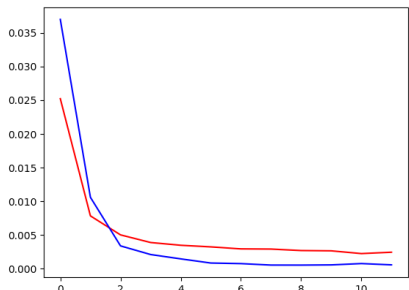
Epochs = 3



Epochs = 10

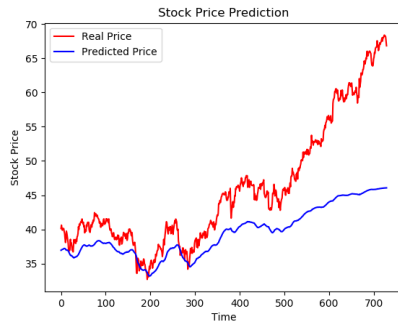


Epochs = 40

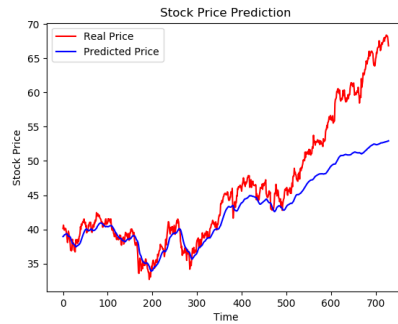


Όσο περισσότερες εποχές τόσο καλύτερο **prediction** κάνει το μοντέλο, καθώς σε κάθε πέρασμα μπορεί να βρει κι άλλες τιμές που ίσως να έχασε προηγουμένως και αυτό κάνει το αποτέλεσμα πιο ακριβές.

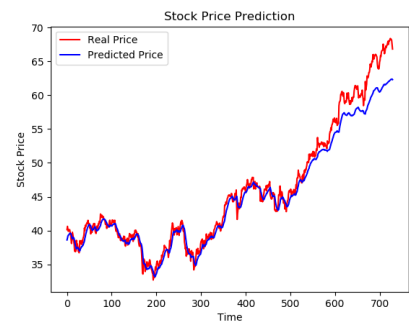
Epochs = 3



Epochs = 10



Epochs = 40



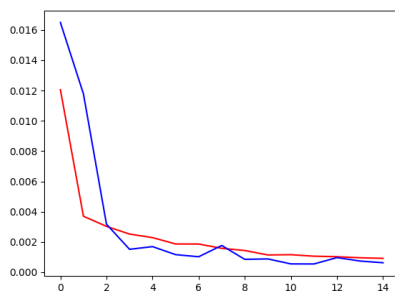
Batch Size

Με lstm_units = 8, lstm_layers = 1, dropout = 0.1, epochs = 20, delay = 200

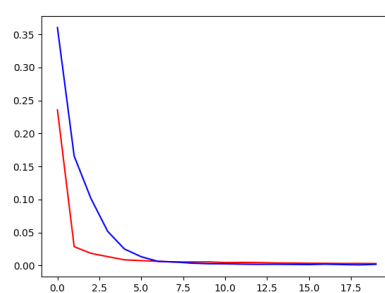
Batch size	8	32	64	128
Test loss	0.00100	0.02377	0.00399	0.00766

Το **validation** αλλάζει ως εξής:

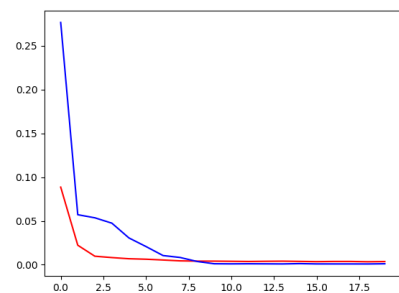
Batch_size = 8



Batch_size = 32

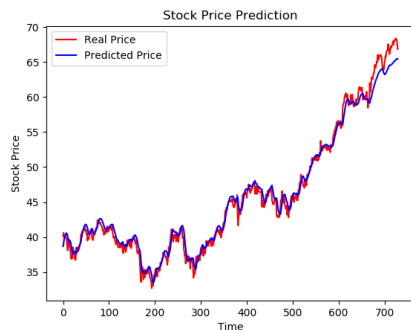


Batch_size = 128

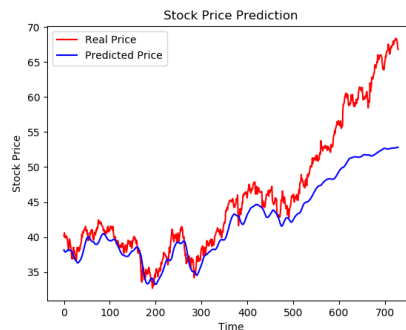


Εδώ βλέπουμε ότι το **prediction** χαλάει για μεγάλα νούμερα του batch_size. Παρατηρούμε δηλαδή ότι όταν ο αριθμός των δειγμάτων εκπαίδευσης που πρέπει να επεξεργαστούν είναι μικρός τότε το prediction είναι πιο ακριβές.

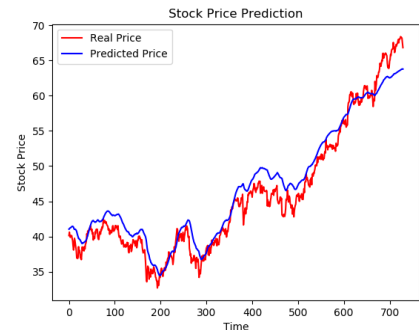
Batch_size = 8



Batch_size = 32



Batch_size = 128



Optimal Model

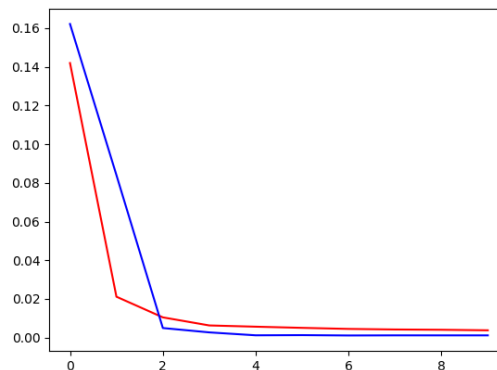
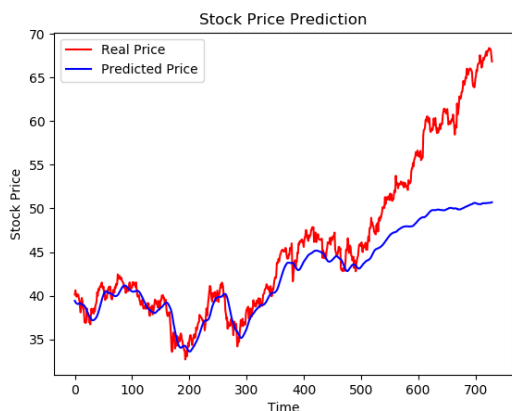
Ως βέλτιστο μοντέλο έχουμε βάλει τις παρακάτω υπερπαραμέτρους:

- Lstm_units = 8
- Lstm_layers = 1
- Dropout = 0.1
- Epochs = 20
- Batch_size = 32
- Delay = 200

Το αποτέλεσμα που πήραμε για το **loss** είναι:

Test sample 0 loss: 0.029269871888809824

Το αποτέλεσμα του plot για το **prediction** είναι στα αριστερά και το αποτέλεσμα του **validation loss** είναι στα δεξιά:



Optimal Model for $n > 1$

Τρέξαμε το ερώτημα και με $n = 5$ και πήραμε τα παρακάτω αποτελέσματα:

Test sample 0 loss: 7.413928293342357e-05

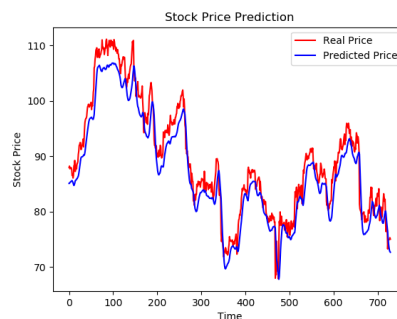
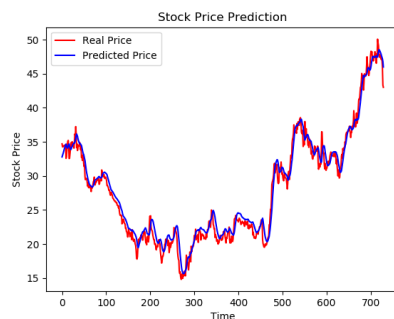
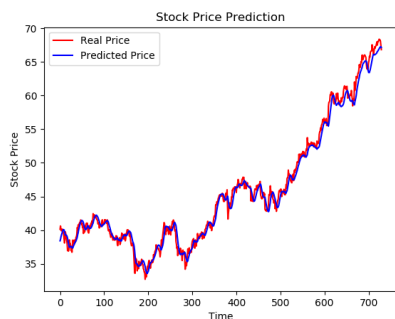
Test sample 1 loss: 0.00012191090021563107

Test sample 2 loss: 0.00010201230829371114

Test sample 3 loss: 0.002527747966655313

Test sample 4 loss: 0.0007679698316616401

Παραδείγματα του **prediction**:



Στο φάκελο του ερωτήματος Α υπάρχουν στους αντίστοιχους φακέλους όλα τα αποτελέσματα που βγάλαμε, τα οποία δείχνουμε παραπάνω.