Απόστολος Φλωράκης, επιβλέπων καθηγητής Εμίρης Ιωάννης
(Apostolos Florakis, supervised by Ioannis Z.EMIRIS)
Οκτώβριος 2018-Φεβρουάριος 2019
(October 2018 – February 2019)

Κατηγοριοποίηση Κρυπτονομισμάτων και πρόβλεψη τιμής, με τη χρήση Νευρωνικών Δικτύων (Cryptocurrencies Classification and price prediction by Neural Nets)

Στόχος της εργασίας
(Goal of this Project)

Σκοπός είναι να σχεδιαστεί και εκπαιδευτεί ένα νευρωνικό δίκτυο (neural network), το οποίο θα ταξινομεί νομίσματα (cryptocurrencies) ως ‘good’ η ‘bad’ (classification problem). Επίσης υλοποιείται ένα ακόμα μοντέλο, το οποίο επιχειρεί να προβλέψει για κάποιο νόμισμα, τη μέση τιμή (mean price) για την επόμενη μέρα (regression problem). Και τα δυο δίκτυα εκπαιδεύονται και πραγματοποιούν προβλέψεις δεδομένων κάποιων χαρακτηριστικών, που αφορούν τα νομίσματα τα οποία έχουμε στη διάθεση μας, για το διάστημα πρίν από τη μέρα που θέλουμε να κατηγοριοποιήσουμε-προβλέψουμε.

Μορφή δεδομένων – Χαρακτηριστικά (features)

Τα χαρακτηριστικά χωρίζονται σε δυο γενικές κατηγορίες: technical analysis (TA) και foundamental and technological analysis (FTA). Το TA περιέχει τιμές από οικονομικούς δείκτες για κάθε νόμισμα, ενώ το FTA τιμές σχετικά με το development των νομισμάτων (source code repos) καθώς και κάποια blockchain features.

Τα features είναι τα εξής:

- blockcount
- medianfee
- txcount
- activeaddresses
- forks
- stars
- subscribers
- total_issues
- closed_issues
- pull_requests_merged
- pull_request_contributors
- commit_count_4_weeks

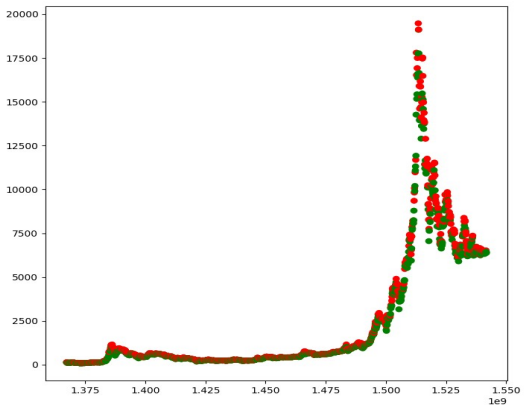
- open
- high
- low
- close
- volume
- market_cup

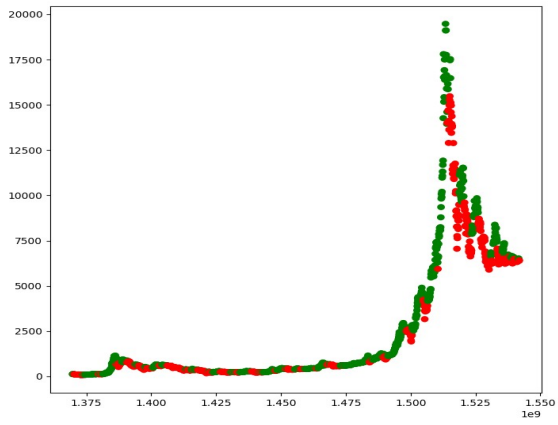
Το σύνολο αυτών των 18 χαρακτηριστικών είναι daily, κάτι που σημαίνει πως για κάθε μέρα έχουμε για κάθε νόμισμα ένα πίνακα με τα χαρακτηριστικά που αντιστοιχούν σε αυτό.

Labelling

Δεδομένου πως το πρώτο πρόβλημα που προσπαθούμε να επιλύσουμε είναι ένα πρόβλημα κατηγοριοποίησης (classification) θα πρέπει να πραγματοποιήσουμε κάποιο labelling στα δεδομένα μας. Ο διαχωρισμός που θέλουμε να επιτύχουμε σε ‘good’-’bad’ είναι αρκετά αυθαίρετος και υποκειμενικός και για τον αυστηρό ορισμό του πιθανώς να απαιτούνται γνώσεις οικονομικών. Έτσι ακολούθησα δυο διαισθητικές εναλλακτικές χρησιμοποιώντας την τιμή (price) του εκάστοτε νομίσματος που έχω στη διάθεση μου για κάθε μέρα. Βάσει της αρχικής ιδέας που είχα το label για κάθε τρέχουσα μέρα υπολογίζονταν ως η διαφορά ((τιμή επόμενης μέρας - τιμή τρέχουσας μέρας) - (τιμή τρέχουσας μέρας - τιμή προηγούμενης μέρας)) και αναλόγως άμα αυτό ήταν θετικό η αρνητικό καθορίζονταν το labelling. Η ιδέα αυτή δεν απέδιδε καθώς το labelling δεν οδηγούσε σε επαρκή και ξεκάθαρο διαχωρισμό των δεδομένων μου, οπότε προχώρησα στην προσέγγιση που ακολουθώ τώρα. Υπολογίζω τη μέση τιμή των προηγούμενων N μερών και άμα αυτή είναι μικρότερη η μεγαλύτερη από αυτή της τρέχουσας μέρας, καθορίζω το label ως θετικό η αρνητικό αντίστοιχα.

Ακολουθούν οι γραφικές παραστάσεις για το (bitcoin) τιμής (price) σε συνάρτηση με την ημερομηνία (date) όπου απεικονίζεται η χρονική εξέλιξη της τιμής του νομίσματος. Με πράσινο αναπαριστώνται οι μέρες που χαρακτηρίζονται ως ‘good’ ενώ με κόκκινο αυτές που θεωρούνται ‘bad’. Παρόμοια δουλειά έχει γίνει για όλα τα νομίσματα.





Απόστολος Φλωράκης, επιβλέπων καθηγητής Εμίρης Ιωάννης
(Apostolos Florakis, supervised by Ioannis Z.EMIRIS)
Οκτώβριος 2018-Φεβρουάριος 2019
(October 2018 – February 2019)

Οι παραπάνω γραφικές παραστάσεις απεικονίζουν το labelling που πραγματοποιείται βάσει της πρώτης και της δεύτερης προσέγγισης αντίστοιχα. Με πράσινο απεικονίζονται οι ‘καλές’
ενώ με κόκκινο οι ‘κακές’ μέρες. Παρατηρούμε πως στη δεύτερη περίπτωση είναι πολύ πιο ευδιάκριτο και σαφές το πότε η τιμή ενός νομίσματος μπορεί να θεωρηθεί ‘good’ η ‘bad’ σε αντίθεση με την πρώτη όπου οι δυο κλάσεις είναι σαφώς πιο μπλεγμένες και δυσδιάκριτες.
Ως προς το πρόβλημα του regression ώς στόχο έχουμε να προβλέψουμε τη μέση τιμή της επόμενης ημέρας. Η μέση τιμή ορίζεται ως ο μέσος όρος των High-Low χαρακτηριστικών.

Προεπεξεργασία δεδομένων - δημιουργία διανυσμάτων και χρονοσειρών
(Data Preprocessing – Vectorization and Timeseries Creation)

Αρχικό μέλημα είναι να επεξεργαστούμε τα δεδομένα που έχουμε στη διάθεση μας και να τα φέρουμε σε μορφή βάσει της οποίας θα μπορέσουμε να εκπαιδεύσουμε το δίκτυο μας. Τα features αναφέρονται παραπάνω συλλέγονται από διαφορετικές πηγές οπότε θα πρέπει να τα συγκεντρώσουμε και να τα επεξεργαστούμε ώστε να δημιουργήσουμε τα τελικά δεδομένα τα οποία θα χρησιμοποιήσουμε. Το obdain_coinmetrics_data.py είναι υπεύθυνο για τη συλλογή των features blockcount,meadianfee,txcount,activeaddresses από το https://coinmetrics.io/. Χρησιμοποιείται ο python wrapper https://github.com/man-c/coinmetrics. Για όλα τα διαθέσιμα νομίσματα για κάθε μια από τις ημερομηνίες εντός ενός εύρους (επιλέγουμε ώς αρχική ημερομηνία μια αρκετά παλιά και ως τελική την τρέχουσα ώστε να συλλέξουμε όσα περισσότερα δεδομένα μπορούμε) δημιουργείται ένα διάνυσμα εφόσον όλα τα χαρακτηριστικά είναι διαθέσιμα. Θέτουμε σαν προτεραιότητα να έχουμε τιμές για όλα τα χαρακτηριστικά και για τον λόγο αυτό σε όλη τη διάρκεια της προεπεξεργασίας απορρίπτουμε διανύσματα τα οποία δεν έχουν τιμές για κάποια από τα χαρακτηριστικά που εξετάζουμε (missing values). Στη συνέχεια υπολογίζεται το label για κάθε ένα διάνυσμα και τα αποτελέσματα γράφονται σε ένα csv αρχείο. Το αρχείο αυτό αποτελεί και τη βασική πηγή δεδομένων μας καθώς καλύπτει αρκετά μεγάλο χρονικό εύρος. Όσα δεδομένα προστεθούν στη συνέχεια θα έιναι χρονικά μεταγενέστερα και χρονικά υποσύνολο των δεδομένων αυτών.

Τα υπόλοιπα χαρακτηριστικά που έχουμε στη διάθεση μας βρίσκονται αποθηκευμένα σε .csv αρχεία όποτε στο επόμενο βήμα θα πρέπει να συγχωνεύσουμε τα αρχεία αυτά και το αρχείο που έχουμε δημιουργήσει πρωτύτερα ώστε να παράξουμε τα πλήρη διανύσματα. Δημιουργούμε και για τα τρία αρχεία λεξικά (dictionaries) με σύνθετο κλειδί την ημερομηνία και το νόμισμα και με τιμή τα χαρακτηριστικά που αφορά το κλειδί. Στη συνέχεια τα λεξικά συγχωνεύονται και προκύπτει εν τέλει ένα συνολικό λεξικό το οποίο περιέχει για όλα τα νομίσματα μόνο τις ημερομηνίες εκείνες για τις οποίες όλα τα χαρακτηριστικά είναι διαθέσιμα. Το λεξικό αυτό γράφεται σε ένα νέο .csv αρχείο.

Το σκεπτικό αυτό έχει προκριθεί ώστε το μοντέλο παραγωγής των διανυσμάτων να είναι επεκτάσιμο. Αυτό σημαίνει πως άμα είχαμε στη διάθεση μας ένα αρχείο με κάποια επιπλέον χαρακτηριστικά θα μπορούσαμε να εμπλουτίσουμε τα ήδη υπάρχοντα δεδομένα μας για τα νομίσματα και ημερομηνίες που έχουμε ήδη διαθέσιμες. Η παραπάνω εργασία συνολικά έχει ως αποτέλεσμα τη δημιουργία ενός .csv αρχείου με στήλες (κατά σειρά): asset,date,open,high,low,close,volume,market_cup,forks,stars,subscribers,total_issues,closed_issues,pull_requests_merged,pull_request_contributors,coomit_count_4_weeks,blockcount,medianfee,txcount,activeaddresses,label. Οι πρώτες δύο αφορούν το όνομα του νομίσματος και την ημερομηνία οι επόμενες 18 αφορούν τα χαρακτηριστικά μας και η τελευταία αφορά το label.

Στο σημείο αυτό πραγματοποιούμε standarization στα δεδομένα μας για κάθε χαρακτηριστικό κάθε νομίσματος. Κατά τη διαδικασία αυτή οι τιμές των χαρακτηριστικών αναπροσαρμόζονται ώστε να έχουν μέση τιμή 0 και τυπική απόκλιση ίση με 1. Τα χαρακτηριστικά μας έχουν διαφορετικά εύρη τιμών και με τη διαδικασία αυτή επιθυμούμε να τα αναπροσαρμόσουμε μειώνοντας την απόκλιση από τη μέση τιμή του χαρακτηριστικού του νομίσματος, διευκολύνοντας έτσι την εκπαίδευση του δικτύου μας. Χρησιμοποιείται η εξίσωση y=(x-mean)/standard_deviation όπου x η παλιά τιμή του χαρακτηριστικού και y η καινούρια, για δεδομένη μέρα, mean η μέση τιμή του χαρακτηριστικού για ένα νόμισμα για όλες τις διαθέσιμες μέρες και standard_deviation η τυπική απόκλιση για το ίδιο διάστημα.

Για το πρόβλημα του regression έχουν χρησιμοποιηθεί χαρακτηριστικά που αφορούν κυρίως οικονομικά μεγέθη και κατ’επεκταση είναι άμεσα συσχετιζόμενα και την τιμή που επιδιώκουμε να προβλέψουμε. Έτσι το διάνυσμα διαμορφώνεται από τα χαρακτηριστικά Open,High,Close,Open,Marketcup,Low. Για το πρόβλημα αυτό πραγματοποιείται normalization μέσω της εξίσωσης y = (x – min)/(max-min), όπου x η παλιά τιμή του χαρακτηριστικού y η καινούρια και min,max η μέγιστη και ελάχιστη τιμή αντίστοιχα.

Τα χαρακτηριστικά μας είναι daily αυτό σημαίνει πως για κάθε νόμισμα έχουμε τις τιμές των χαρακτηριστικών αυτών για διαδοχικές ημερομηνίες. Για να εκμεταλλευτούμε τη χρονική αυτή πληροφορία και συσχέτιση και δεδομένου πως ο χαρακτηρισμός ενός νομίσματος ως ‘good’ η ‘bad’ ή η πρόβλεψη που επιχειρούμε, επηρεάζεται από τη παλαιότερη συμπεριφορά του νομίσματος, μοντελοποιούμε τα δεδομένα μας ως χρονοσειρές (timeseries). Δεδομένου λοιπόν ενός μεταβλητού παραθύρου που εμείς καθορίζουμε (για παράδειγμα 7 ημέρες) το label της κάθε έβδομης ημέρας αφορά τη συνένωση των χαρακτηριστικών των ημερών 1 έως 7 διατηρώντας τη χρονολογική σειρά που αυτά εμφανίζονται. Στην συνέχεια το παράθυρο μετατοπίζεται χρονικά κατά ένα και η διαδικασία επαναλαμβάνεται για τις μέρες 2 έως 8. Σε κάθε βήμα δημιουργείται μια χρονοσειρά ως το άθροισμα των επιμέρους πινάκων χαρακτηριστικών για τις ημέρες εντός του παραθύρου και ως label ή target-mean price τίθενται αυτά της τελευταίας η επόμενης μέρας αντίστοιχα (classificaation,regression). Τα τελικά λοιπόν δεδομένα με τα οποία και θα εκπαιδεύσουμε το δίκτυο μας (για παράθυρο 7 ημερών) έχουν διάσταση 7*18, αποτελούν δηλαδή τη συνένωση 7 ‘πακέτων’ 18 χαρακτηριστικών με τη σειρά που αυτά εμφανίζονται ανά ημέρα στην περίπτωση του classification και 7*6 (7 πακέτα των 6 χαρακτηριστικών) στην περίπτωση του regression.

Αρχιτεκτονική Δικτύων (Networks Architecture)

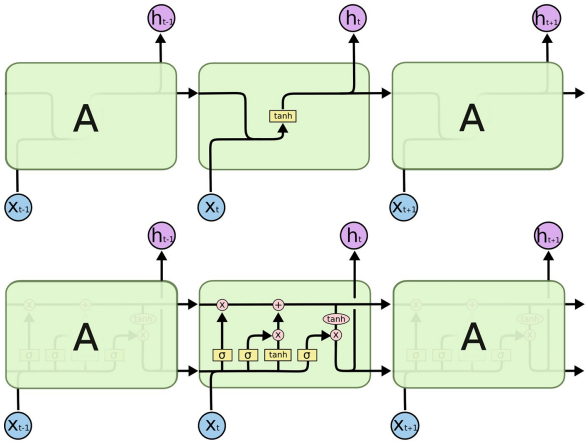
Για το πρόβλημα της κατηγοριοποίησης κρυπτονομισμάτων έχουν επιλεγεί και υλοποιηθεί δυο μοντέλα ένα Feedforward Neural Net και ένα RNN(Recurrent Neural Network). Το πρώτο δίκτυο είναι ένα κλασσικό νευρωνικό δίκτυο εμπρόσθιας διάδοσης (Multilayer Perceptron). Αναφορικά με

το δεύτερο δίκτυο, έχει υλοποιηθεί ένα RNN το οποίο ανταποκρίνεται αποτελεσματικότερα σε ακολουθιακά δεδομένα (sequences).

Feedforward Network: Το δίκτυο αυτό αποτελείται από ένα επίπεδο εισόδου, δυο κρυφά επίπεδα και ένα επίπεδο εξόδου. Το επίπεδο εισόδου έχει διάσταση [days_window*num_of_featres] πρόκειται δηλαδή για μια επιπεδοποιημένη (flattened) μορφή των χαρακτηριστικών για διαδοχικές μέρες. Το δίκτυο αυτό δεν έχει τη δυνατότητα να εκμεταλλευτεί τη χρονική πληροφορία και έτσι θεωρεί ως είσοδο μια επιπεδοποιημένη χρονοσειρά την οποία και επεξεργάζεται με ενιαίο τρόπο, θεωρώντας την απλά μια “πακεταρισμένη” συλλογή χαρακτηριστικών. Ως προς τα κρυφά επίπεδα έχουν δοκιμαστεί διαφορετικοί αριθμοί νευρώνων και ως διάσταση του επιπέδου εξόδου έχει οριστεί το 2 (δυο) καθώς δυο είναι οι κλάσεις του προβλήματος μας (‘good’-’bad’). Ως συνάρτηση των κρυφών επιπέδων έχει οριστεί η RELU(Rectifier Layer Unit). Ως συνάρτηση κόστους έχει επιλεγεί η softmax_cross_entropy_with_logits και για το optimization χρησιμοποιείται ο AdamOptimizer.

Recurrent Network: Στην παράγραφο αυτή θα προσπαθήσω εκτός της τυπικής περιγραφής του δικτύου να αποτυπώσω τη γενικότερη φιλοσοφία και χρησιμότητα των αναδρομικών δικτύων καθώς και χρήσιμες-κρίσιμες ιδιαιτερότητες που σχετίζονται με αυτά. Το δίκτυο αυτό δέχεται ως είσοδο τις χρονοσειρές σε διανύσματα διάστασης: [days_window,num_of_features]. Έχει λοιπόν τη δυνατότητα να εκμεταλλευτεί και να διατηρήσει τη χρονική αλληλουχία, επεξεργαζόμενο τα χαρακτηριστικά κάθε μέρας ξεχωριστά και διατηρώντας μνήμη και συνυπολογίζοντας κατά την επεξεργασία των χαρακτηριστικών για κάθε timestep (μέρα) εκείνες τις μέρες που έχουν προηγηθεί. Διατηρεί λοιπόν μια “μνήμη” (memory) το λεγόμενο κρυφό επίπεδο (hidden state). Ονομάζεται αναδρομικό γιατί κατά την επεξεργασία μιας χρονοσειράς για κάθε μέρα παράγεται μια έξοδος η οποία ανατροφοδοτείται στο δίκτυο μεταβάλλοντας την μνήμη του. Η μνήμη που διαμορφώνεται με τη σειρά της επηρεάζει την επεξεργασία για κάθε επόμενη μέρα κοκ. Περιγραφικά έστω πως έχουμε μια πρόταση με τρείς λέξεις και θέλουμε να την επεξεργαστούμε με ένα RNN. Θα εισαγάγουμε την πρώτη λέξη και κατα τη φάση αυτή η μνήμη είναι κενή (η ακριβέστερα τυχαία αρχικοποιημένη) οπότε δεν συμβάλει ουσιωδώς στην διαμόρφωση της εξόδου. Στο δεύτερο βήμα και την επεξεργασία της δεύτερης λέξης θα ληφθεί υπόψην η μνήμη η οποία έχει διαμορφωθεί από την επεξεργασία της πρώτης λέξης. Στην τρίτη λέξη θα ληφθεί υπόψιν η μνήμη που περιέχει πληροφορία και για τις δυο προηγούμενες λέξεις. Η έξοδος του τελευταίου timestep θα είναι και η έξοδος του δικτύου μας και κατανοούμε πως στην διαμόρφωση της τιμής της καθοριστικό ρόλο έχει η χρονική αλληλουχία. Παρότι τα RNN είναι εύκολα εκπαιδεύσιμα σύχνα αντιμετωπίζουν το πρόβλημα του vanishing gradient. Κατά τον αλγόριθμο backpropagation και κτά την εκπαίδευση (training) που στοχεύει στην επιδιόρθωση των εσωτερικών παραμέτρων του δικτύου (weights,biases), αχικά εισάγουμε στο δίκτυο ορισμένα δεδομένα. Μετά την επεξεργασία αυτών από το δίκτυο υπολογίζεται στο επίπεδο εξόδου και λαμβάνοντας υπόψιν το label της εισόδου ένα σφάλμα, σύμφωνα με τη συνάρτηση κόστους που έχουμε ορίσει. Δεδομένου λοιπόν του σφάλματος προσπαθούμε να διορθώσουμε τα βάρη των προηγούμενων επιπέδων για να βελτιώσουμε την ακρίβεια μας. Το gradient όμως (αλγόριθμος gradient descent) στα δίκτυα της αρχιτεκτονικής αυτής, συχνά μικραίνει όσο προχωράμε στα πρώτα επίπεδα, κινούμενοι από το τέλος προς την αρχή. Το εκάστοτε ανανεωμένο βάρος δίνεται από την εξίσωση new_weight = old_weigh – learning_rate*gradient οπότε όσο μικρότερο είναι το gradient τόσο πιο μικρή θα είναι και η ανανέωση του βάρους. Οπότε και αυτόματα, αρχικά επίπεδα ουσιαστικά δεν ανανεώνονται-’μαθαίνουν’. Κατά συνέπεια σημαντική πληροφορία που τυχόν βρίσκεται στην αρχή της χρονοσειράς χάνεται και δεν αποτυπώνονται-αποθηκεύονται μακροχρόνιες συσχετίσεις (long-term dependencies), παρά μόνο βραχείας μορφής χρονικές συσχετίσεις που αφορούν μόνο τα

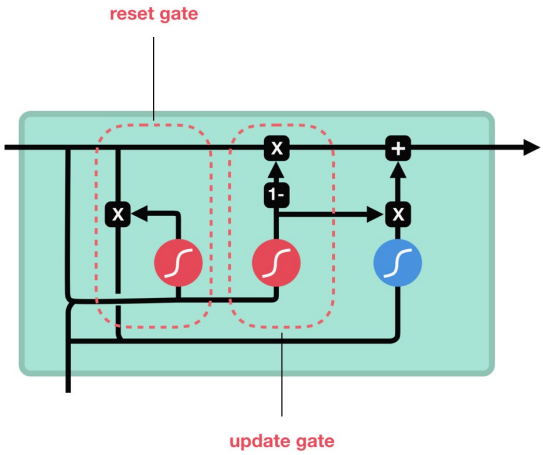
τελευταία χρονικά timesteps (short-term dependencies). Τη λύση στο πρόβλημα αυτό δίνουν οι πιο εξελιγμένες μορφές αναδρομικών νευρώνων που ονομάζονται LSTM (Long Sort term Memory) cell και GRU (Gated Recurrent Unit) cell. Συνοπτικά οι νευρώνες που βασίζονται στις παραπάνω πιο εξελιγμένες αρχιτεκτονικές μεταχειρίζονται τη μνήμη με πιο αποτελεσματικό και έξυπνο τρόπο ανανεώνοντας την συνεχώς βάσει της σημασίας και κρισιμότητας της χρονικής πληροφορίας. Κατά το σκεπτικό αυτό χρονικά προγενέστερα timesteps που πιθανώς είναι κρίσιμα λαμβάνονται εξίσου υπόψιν με μεταγενέστερα και έτσι αποτυπώνεται ομοιόμορφα τόσο η μακροχρόνια τόσο και η βραχυχρόνια χρονική πληροφορία.



Στο παραπάνω σχήμα βλέπουμε τη διαφορά δυο αναδρομικών δικτύων, ενός με βασικό RNNcell και του δεύτερου με LSTMcell. Η διαδικασία επεξεργασίας μια χρονοσειράς εφόσον “ξετυλιχθεί” μπορεί να προσομοιωθεί ακολουθιακά όπως στο σχήμα και να είναι σχηματικά πιο ευκρινής και κατανοητή, άμα όμως θεωρήσουμε πως υπάρχει μόνο ένα unit το οποίο ανατροφοδοτεί τον εαυτό του τόσο ως προς την είσοδο-έξοδο όσο και ως προς τη μνήμη έχουμε ουσιαστικά μια αναδρομική εικόνα. Στο πρώτο δίκτυο παρατηρούμε πως σε κάθε timestep η μνήμη (h(t)) είναι ένας συνδυασμός του τρέχοντος timestep (x(t)) και αυτούσιας της μνήμης του προηγούμενου επιπέδου, φιλτραρισμένα από μια συνάρτηση ενεργοποίησης tahn η οποία αναδιατάσσει τις τιμές στο διάστημα (-1,1). Στο lstm cell έξοδος (cell state) και μνήμη (hidden state) καθορίζονται με ένα πιο σύνθετο τρόπο. Υπάρχουν επιμέρους δίκτυα input,output,forget τα υλοποιούνται μέσω γραμμικών πράξεων (προσθέσεις και πολλαπλασιασμοί), και συναρτήσεων ενεργοποίησης than και sigmoid. Αρχικά στη forget gate αποφασίζονται ποιες από τις τιμές του της εισόδου του δικτύου h(t-1) (εξόδου hidden state προηγούμενου timestep) θεωρούνται σημαντικές και κατά πόσο θα “διατηρηθούν” ή θα “ξεχαστούν”. Εφαρμόζοντας λοιπόν σιγμοειδή συνάρτηση με πεδίο τιμών [0,1] και στη συνέχεια πολλαπλασιάζοντας τη μάσκα που θα προκύψει με το cell state του προηγούμενου επιπέδου, καταφέρνουμε να αναπροσδιορίσουμε την βαρύτητα των στοιχείων της μνήμης του δικτύου μας. Τιμές κοντινές στο 1 υποδηλώνουν πληροφορία που πρέπει να “διατηρηθεί” ενώ κοντά στο 0 πληροφορία που πρέπει να ξεχαστεί. Στη συνέχεια στο input gate αποφασίζεται ποιά στοιχεία του τρέχοντος timestep x(t) θα προστεθούν στο cell state, πάλι με τη χρήση σιγμοειδούς συνάρτησης, με την επιπρόσθετη χρήση tahn συνάρτησης ενεργοποίησης για το scaling στο (-1,1) των τιμών αλλά και με την πράξη της πρόσθεσης, δεδομένου πως επιθυμούμε να προσθέσουμε στοιχεία στο state αντί να το φιλτράρουμε. Η σιγμοειδής συνάρτηση επιλέγει τα στοιχεία που θα

προστεθούν και η συνάρτηση tahn δημιουργεί το υποψήφιο διάνυσμα. Με παρεμφερείς πράξεις προκύπτει η έξοδος και η συνολική νέα μνήμη του εκάστοτε timestep του νευρώνα.

Αναφορικά με τα GRU cell λειτουργούν με παρόμοιο τρόπο έχοντας τις επιμέρους reset και update gates ανά cell. Στοχεύουν στον να διατηρήσουν “μακροχρόνια” πληροφορία χωρίς να την επεξεργάζονται συνεχώς και στον να απορρίπτουν αποτελεσματικά αχρείαστη χρονική πληροφορία. Ακολουθεί παράδειγμα τέτοιου cell.



Στην υλοποίηση μου έχω χρησιμοποιήσει τόσο lstm όσο και gru cell. Το gru cell έχει προτιμηθεί στο πρόβλημα του classification, ενώ το lstm σε αυτό του regression. Έχουν δοκιμαστεί διάφορες αρχιτεκτόνικες ώς προς το πλήθος νευρώνων και επιπέδων. Καλύτερα αποτελέσματα απέφερε το ένα επίπεδο με 64-128 νευρώνες. Μετά το recurrent δίκτυο στην περίπτωση του rgression έχει προστεθεί και ένα dense (fully connected) επίπεδο όπου πραγματοποιούνται μόνο γραμμικές πράξεις ώστε να γίνει εν τέλει η πρόβλεψη.

Εκπαίδευση (Training)

Δεδομένου πως διαχειρίζομαι χρονοσειρές επιλέχθηκε να εκπαιδεύσω το δίκτυο μου (training) σε μια διαφορετική χρονική περίοδο και να το τεστάρω (testing) σε μια άλλη. Ο λόγος της τακτικής αυτής είναι πως δεν θέλω να έχουν καμία χρονική επικάλυψη τα δεδομένα μου, να μην έχει δηλαδή

το δίκτυο μου καμία πληροφορία κατά την εκπαίδευση του για την περίοδο που θα κληθεί να κάνει προβλέψεις. Χρησιμοποιείται επίσης Mini-Batch Mode στο gradient descent. Αυτό σημαίνει πως το training set διασπάται σε μικρότερα κομμάτια τα οποία τροφοδοτούνται ένα ένα στο δίκτυο. Στο τέλος κάθε τροφοδοσίας γίνεται update των παραμέτρων του δικτύου. Η διαδικασία πραγματοποιείται επαναληπτικά για μεταβλητό αριθμό epochs. Σε κάθε epoch όλο το training set δίνεται στο δίκτυο σε mini batch mode. Μια σημαντική λεπτομέρεια είναι η εξής: τα recurrent δίκτυα που έχω σχεδιάσει είναι stateless, γεγονός που σημαίνει πως οι εσωτερικές παράμετροι του hidden state του rnn επανα-αρχικοποιούνται στο τέλος της επεξεργασίας κάθε batch. Στην περίπτωση που αποθηκεύονταν (statefull) θα έπαιρνα καλύτερα αποτελέσματα άμα έδινα τις χρονοσειρές σε κάθε epoch με τη χρονική σειρά που αυτές εμφανίζονται χρονικά. Όμως επέλεξα να υλοποιήσω stateless δίκτυο οπότε λαμβάνω καλύτερα αποτελέσματα, ανακατεύοντας όλο το training set (suffling) ανά epoch. Επίσης λόγω του μικρού αριθμού δεδομένων αρά και κατά συνέπεια των μικρών σε μέγεθος δικτύων που υλοποιώ, έχω προσθέσει dropout στα περισσότερα layers ώστε να περιορίσω το φαινόμενο του overfitting . Σύμφωνα με αυτό υπάρχουν φορές που το δίκτυο μπορεί να συμπεριφέρεται καλά κατά το training αλλά επειδή υπερεξειδικεύεται στο train set και ‘μαθαίνει’ παραπάνω λεπτομέρειες για αυτό από ότι χρειάζεται, να παρουσιάζει χαμηλή απόδοση στο test set. Για το λόγο αυτό κάποιοι νευρώνες απενεργοποιούνται με πιθανότητα p ώστε να ‘καταστραφούν’ και συσχετίσεις νευρώνων του δικτύου που κατ’ επέκταση μπορεί να σχετίζονται ‘επικίνδυνα’ με ιδιαιτερότητες του train set και να οδηγούν σε λάθη στο test set. Λόγω λίγων δεδομένων τα epochs τουλάχιστον στο classification είναι λίγα, τάξης μεγέθους 200-500. Ως προς τα πειράματα που έχουν γίνει έχει γίνει τόσο στο classification όσο και στο regression εκπαίδευση και πρόβλεψη για το bitcoin και έχει πραγατοποιηθεί και μια μεικτή εκπαίδευση για bitcoin,ethereum,lightcoin και πρόβλεψη για το dash στο classification πρόβλημα.

Μετρικές Αξιολόγησης (Evaluation Metrics)

Για το classification υπολογίζονται τα tp(true positive),tn(true negative),fp(false positive),fn(false negative) και κατ’επέκταση acuracy,precision,recall,f-measure. Για το regression υπολογίζεται το total mae (mean absolute error) των prediction.

Ο ορισμός των tp,fp,tn,fp είναι ο εξής:

		Actual	
		Positive	Negative
Predicted	Positive	**True Positive**	**False Positive**
	Negative	**False Negative**	**True Negative**

Βάσει των παραπάνω ορίζονται οι παρακάτω μετρικές αξιολόγησης της κατηγοριοποίησης:

- Accuracy = (TP+TN)/(TP+TN+FP+FN)
- Precision = TP/(TP+FP)
- Recall = TP/(TP+FN)
- F-measure = 2*((precision * recall)/(precision + recall))

Το mean absolute error το οποίο χρησιμοποιείται για την αξιολόγηση του regression ορίζεται ως:

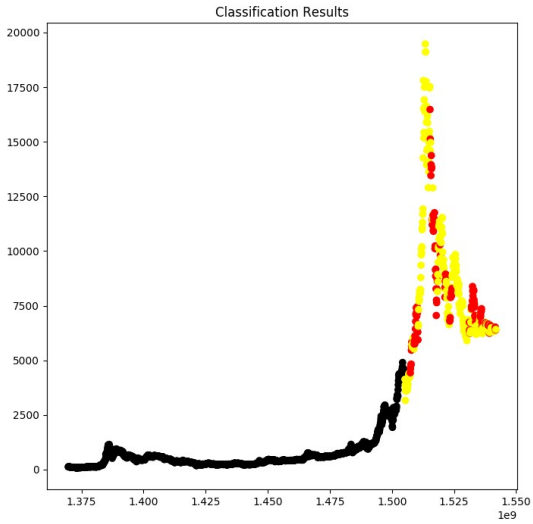
$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n} |\, x_i - x \,|$$

Όπου n είναι το πλήθος των προβλέψεων, x η τιμή πρόβλεψης και xi η πραγματική τιμή.

Απόστολος Φλωράκης, επιβλέπων καθηγητής Εμίρης Ιωάννης
(Apostolos Florakis, supervised by Ioannis Z.EMIRIS)
Οκτώβριος 2018-Φεβρουάριος 2019
(October 2018 – February 2019)

Πειράματα - Αποτελέσματα (Experiments – Results)

Classification - RNN

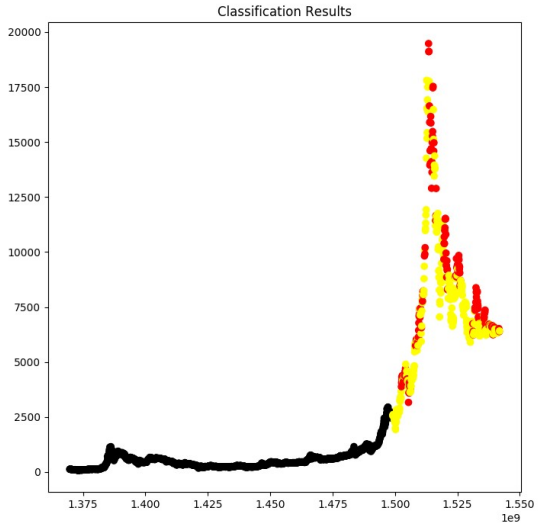


Στο παραπάνω πείραμα που πραγματοποιήθηκε για το bitcoin με μαύρο αναπαριστώνται οι μέρες που χρησιμοποιήθηκαν για την εκπαίδευση του δικτύου, και αναφορικά με το testing με κίτρινο παρουσιάζονται οι σωστές κατηγοριοποιήσεις (well classified days) ενώ με κόκκινο οι λανθασμένες (misclassified days). Για το bitcoin έχω στη διάθεση μου 1415 διανύσματα. Τα διανύσματα είναι timeseries 4 ημερών με 18 χαρακτηριστικά ανά ημέρα. Από αυτά το πρώτο 70% χρησιμοποιήθηκε για την εκπαίδευση και το υπόλοιπο 30% για το testing. Από τα διανύσματα εκπαίδευσης, 572 είναι labeled ως good ενώ 418 ως bad. Χρησιμοποιήθηκε RNN με GRUcell. Το accuracy του μοντέλου είναι 60 – 70%.

Accuracy	Precision	Recall	F-measure
0.71	0.67	0.71	0.69

Απόστολος Φλωράκης, επιβλέπων καθηγητής Εμίρης Ιωάννης
(Apostolos Florakis, supervised by Ioannis Z.EMIRIS)
Οκτώβριος 2018-Φεβρουάριος 2019
(October 2018 – February 2019)

Classification - MLP



Παραπάνω παρατίθενται τα αντίστοιχα αποτελέσματα κατηγοριοποίησης για το feedforward δίκτυο, όπου για το training έχει χρησιμοποιηθεί το 65%. Το μοντέλο αυτό συμπεριφέρεται κατά 5-10% χειρότερα στη γενική περίπτωση.

Accuracy	Precision	Recall	F-Measure
0.66	0.62	0.88	0.73

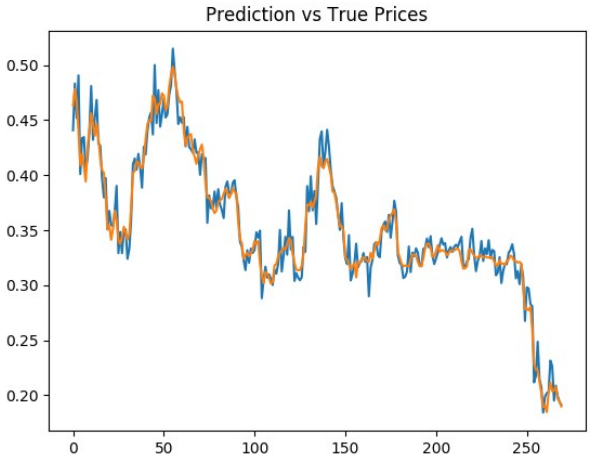
Υπήρξε και μια ‘μεικτή’ εκπαίδευση του δικτύου με δεδομένα από τα νομίσματα bitcoin,ethereum,lightcoin και κατηγοριοποίηση στη συνέχεια για το dash νόμισμα. Χρησιμοποιήθηκαν χρονοσειρές παραθύρου 10, LSTMcell και τα αποτελέσματα ήταν τα εξής:

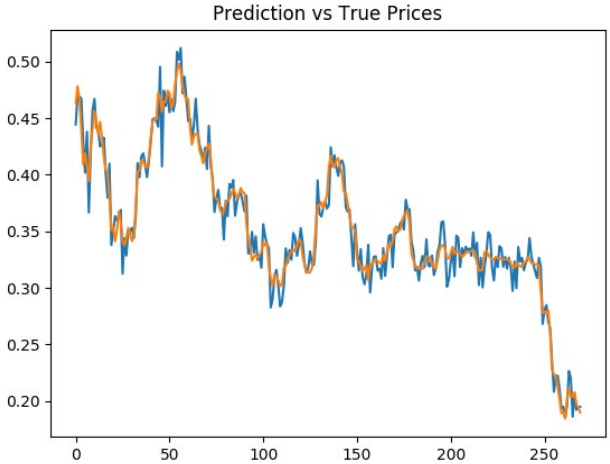
Accuracy	Precision	Recall	F-measure
0.64	0.49	0.82	0.61

Απόστολος Φλωράκης, επιβλέπων καθηγητής Εμίρης Ιωάννης
(Apostolos Florakis, supervised by Ioannis Z.EMIRIS)
Οκτώβριος 2018-Φεβρουάριος 2019
(October 2018 – February 2019)

Regression - RNN

Τα παρακάτω αποτελέσματα αφορούν το regression στο οποίο χρησιμοποιήθηκε μόνο το RNN με LSTM cell. Το παράθυρο το οποίο επιλέχθηκε ήταν 7 ημερών, και οι τιμές απεικονίζονται παραπάνω σε normalized μορφή. Για το training χρησιμοποιήθηκε το 85% των δεδομένων. Το συνολικο σφάλμα είναι 0.008 - 0.02 (total mae).





Απόστολος Φλωράκης, επιβλέπων καθηγητής Εμίρης Ιωάννης
(Apostolos Florakis, supervised by Ioannis Z.EMIRIS)
Οκτώβριος 2018-Φεβρουάριος 2019
(October 2018 – February 2019)

Συμπεράσματα και Μελλοντική Δουλεία
(Conclusions and Future work)

Προσέγγισα το θέμα από μια διερευνητική σκοπιά. Εξ ορισμού το πρόβλημα κατηγοριοποίησης που τέθηκε ήταν διαισθητικό καθώς ο χαρακτηρισμός ‘καλό’ ή ‘κακό’ για ένα νόμισμα ίσως επιδέχεται ενός πιο ακριβούς και σύνθετου ορισμού και ερμηνείας από αυτόν που έχω δώσει, παρά το γεγονός πως δοκίμασα αρκετές εναλλακτικές. Προσπάθησα εκτός του προβλήματος της κατηγοριοποίησης να επιλύσω και ένα εντελώς διαφορετικής φιλοσοφίας πρόβλημα όπως αυτό της πρόβλεψης της επόμενης τιμής του νομίσματος. Στόχος μου ήταν να προσεγγίσω τα προβλήματα μέσα από διαφορετικές φιλοσοφίες και αρχιτεκτονικές νευρωνικών δικτύων και να μελετήσω τα αποτελέσματα και τις επιδόσεις τους στα δεδομένα για κρυπτονομίσματα που είχα στη διάθεση μου. Πειραματίστηκα με παραμετροποίηση τόσο της δομής των δικτύων, όσο και εσωτερικής, καθώς και με διάφορες εναλλακτικές ως προς την προεπεξεργασία των δεδομένων. Συμπεραίνω πως τα δίκτυα συμπεριφέρονται καλά (ειδικά τα RNN) και είναι αποτελεσματικά και ελπιδοφόρα για την επεξεργασία χρονικών δεδομένων που αφορούν κρυπτονομίσματα. RNN και Feedforward δεν μπορούν να συγκριθούν άμεσα καθώς δεν έχουν ίδια αρχιτεκτονική πλήθος νευρώνων και αριθμό επιπέδων έχουν υλοποιηθεί όμως και παρουσιαστεί και τα δυο, καθώς ήθελα να δω το πρόβλημα από μια ευρύτερη οπτική γωνία και προσέγγιση. Αρχικά στόχευα να υλοποιήσω ένα βασικό δίκτυο και στη συνέχεια να διερευνήσω τη δυναμική ενός θεωρητικά ισχυρότερου και πιο ενδεδειγμένου δικτύου. Η δουλειά μου ουσιαστικά αποτελεί μια ευρεία και εκτενή προεργασία και διερεύνηση πάνω το πρόβλημα. Περισσότερα δεδομένα, διαφορετικοί ορισμοί και αυστηρή συγκεκριμενοποίηση των ζητουμένων, προσθήκη καινούριων χαρακτηριστικών στα δεδομένα (πχ sentiment analysis), επιπλέον παραμετροποίηση, ενιαία επεξεργασία όλων των νομισμάτων, αλλά

και υλοποίηση πιο εξελιγμένων δίκτυων σίγουρα θα βελτιώσουν τα αποτελέσματα αλλά και θα γεννήσουν νέα ζητούμενα.

Πηγές (References)

- **http://blog.echen.me/2017/05/30/exploring-lstms/**
- **http://colah.github.io/posts/2015-08-Understanding-LSTMs/**
- **http://www.bioinf.jku.at/publications/older/2604.pdf**
- **https://www.tensorflow.org/**
- **https://machinelearningmastery.com/normalize-standardize-time-series-data-python/**
- **https://www.analyticsvidhya.com/blog/2018/09/multivariate-time-series-guide-forecasting-modeling-python-codes/**
- **https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d**
- **https://towardsdatascience.com/rnn-training-tips-and-tricks-2bf687e67527**
- **https://gombru.github.io/2018/05/23/cross_entropy_loss/**
- **http://laid.delanover.com/dropout-explained-and-implementation-in-tensorflow/**