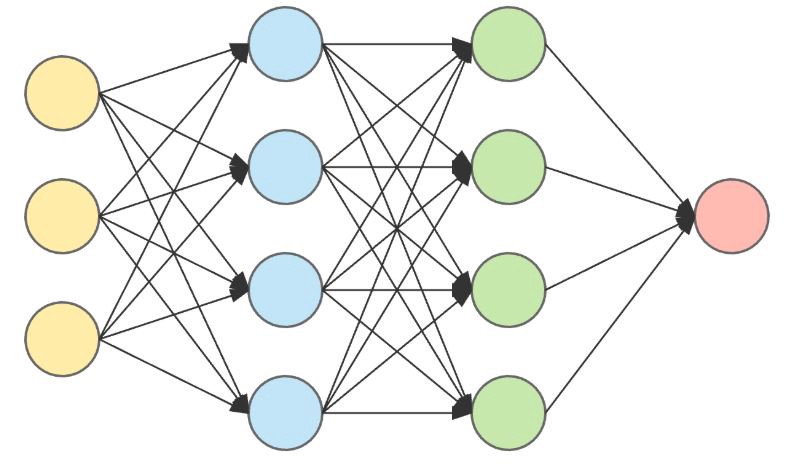
Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα

Εργασία 3



Παπαχρήστου Δημήτρης-1115201500124

Κανταρτζής Θεόφιλος-1115201400061

Χειμερινό 2020-2021

Περιεχόμενα

Contents

[Ανάπτυξη Λογισμικού για Αλγοριθμικά Προβλήματα 1](#_Toc61309839)

[Περιεχόμενα 2](#_Toc61309844)

[Σύντομη Περιγραφή 2](#_Toc61309845)

[Κατάλογος Αρχείων 4](#_Toc61309846)

[Οδηγίες Χρήσης 5](#_Toc61309847)

[Σχολιασμός Κώδικα 6](#_Toc61309848)

[Υλοποίηση “bottleneck” 6](#_Toc61309849)

[Αποτελέσματα Autoencoder 7](#_Toc61309850)

[**1. Μέγεθος Φίλτρων** 8](#_Toc61309851)

[**2. Αριθμός Layers** 9](#_Toc61309852)

[**3. Batch Size** 10](#_Toc61309853)

[**4. Αριθμός Φίλτρων** 11](#_Toc61309854)

[**5. Αριθμός Εποχών** 12](#_Toc61309855)

[**6. Διάσταση Συμπίεσης** 13](#_Toc61309856)

Σύντομη Περιγραφή

Εκπαίδευση και αποτίμηση νευρωνικού δικτύου αυτοκωδικοποίησης εικόνων αριθμητικών ψηφίων.

Χρήση του κωδικοποιητή για τη δημιουργία νευρωνικού δικτύου κατηγοριοποίησης των εικόνων. Εκπαίδευση και αποτίμηση του νευρωνικού δικτύου ταξινόμησης.

Χρησιμοποιήθηκε η γλώσσα Python (3.8) και η προγραμματιστική διεπαφή Keras επί της πλατφόρμας νευρωνικών δικτύων TensorFlow.

Η εργασία αναπτύχθηκε σε περιβάλλον linux.

Κατάλογος Αρχείων

Το πρώτο μέρος της εργασίας υλοποιήθηκε στο αρχείο

* reduce.py

Οδηγίες Χρήσης

Ά ερώτημα:

$python reduce.py –d <dataset> -q <query set> -d <output\_dataset\_file> -oq <output\_query\_file>

Παράδειγμα για τα δεδομένα μας:

$python reduce.py -d ./data/train-images.idx3-ubyte -q ./data/t10k-images.idx3-ubyte -od train\_lat -oq test\_lat

Σχολιασμός Κώδικα

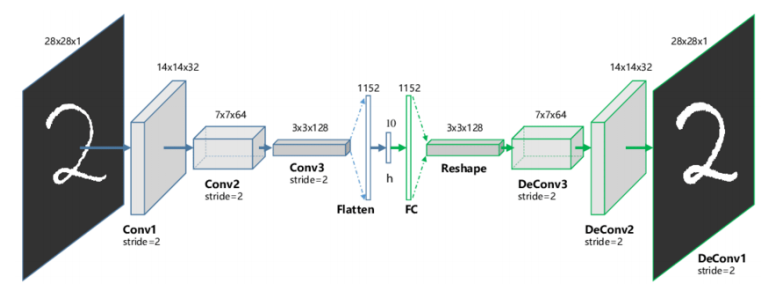
Α΄ ΕΡΩΤΗΜΑ

Υλοποίηση “bottleneck”

Η υλοποίηση του autoencoder με στρώματα συμπίεσης και αποσυμπίεσης (“bottleneck”) βρίσκεται στο reduce.py. Η δημιουργία της δομής του δικτύου ακολουθεί την ίδια λογική με αυτή της δεύτερης εργασίας. Ο χρήστης επιλέγει αρχικά αριθμό στρωμάτων, μέγεθος batch size και αριθμό εποχών. Στην συνέχεια επιλέγει για κάθε layer τον αριθμό των φίλτρων και το μέγεθος αυτών. Επίσης, διαλέγει πότε θα προσθέσει pooling layers στον encoder. Για τον decoder δεν δίνουμε επιπλέον παραμέτρους καθώς εφαρμόζεται mirroring στα layers του, από τον encoder. Η προσθήκη αυτή της εργασίας είναι ότι μεταξύ encoder και decoder ο χρήστης επιλέγει την διάσταση συμπίεσης για την αναπαράσταση των εικόνων σε πολύ μικρότερη διάσταση απο την αρχική.

Α΄ Μέρος

Αποτελέσματα Autoencoder

Σκοπός μας είναι να βρούμε τον βέλτιστο συνδυασμό υπερπαραμέτρων για την ελαχιστοποίηση του σφάλματος και την αποφυγή του overfitting.

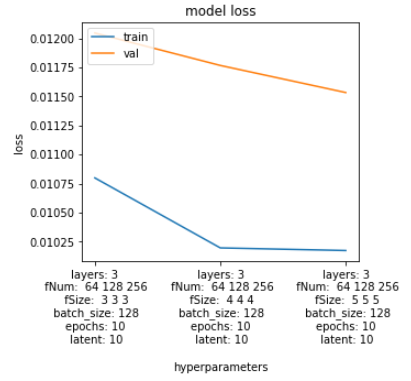
Οι υπερπαράμετροι που θα μελετηθούν είναι:

1. Μέγεθος Φίλτρων
2. Αριθμός Layers
3. Batch Size
4. Αριθμός Φίλτρων
5. Αριθμός Εποχών
6. Διάστασης Συμπίεσης

# **1. Μέγεθος Φίλτρων**

Σε αυτό το βήμα θα μελετήσουμε την μεταβολή των loss για διάφορα μεγέθη φίλτρων. Όπως έχει αναφερθεί και παραπάνω μπορούμε να επιλέξουμε το μέγεθος των φίλτρων σε κάθε layer ξεχωριστά. Για την περιεκτικότητα των αποτελεσμάτων σε κάθε δοκιμή επιλέγουμε ένα μέγεθος το οποίο χρησιμοποιείται σε κάθε layer.

Κρατάμε fixed τις υπόλοιπες υπερπαραμέτρους (batch\_size 128, εναρκτήριο αριθμό φίλτρων 64, εποχές 10, layers 3, διάσταση συμπίεσης 10) και μεταβάλουμε το μέγεθος φίλτρων από 3, 4, 5.

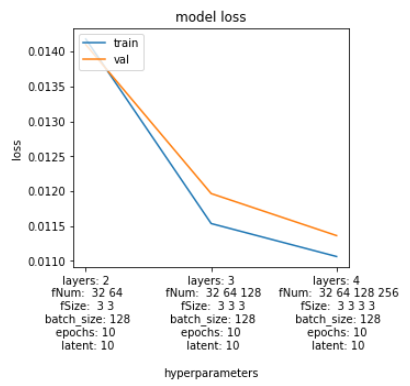


Σχολιασμός Αποτελεσμάτων:

1. Παρατηρούμε και στα τρία διαγράμματα ότι παίρνουμε τις πολύ καλες τιμές loss.
2. Ακόμα παρατηρούμε και στα τρία διαγράμματα ότι για φίλτρα μεγέθους 3x3 έχουμε και την μικρότερη διαφορά ανάμεσα στο train\_loss και val\_loss.

Συμπέρασμα: Επιλέγουμε φίλτρα μεγέθους 3x3.

# **2. Αριθμός Layers**

Μελετήσαμε την μεταβολή των train\_loss και val\_loss με fixed τις τιμές των υπόλοιπων υπερπαραμέτρων (batch\_size 128, εποχές 10, διάσταση συμπίεσης 10, εναρκτήριο αριθμό φίλτρων 32 και μέγεθος 3x3) με 2, 3, 4 layers.

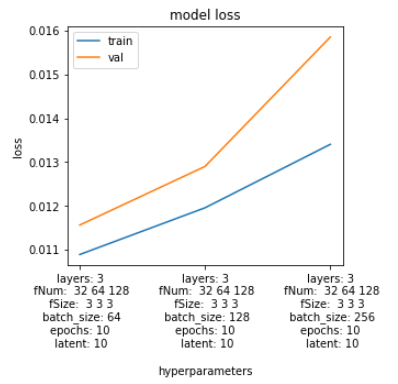
Σχολιασμός Αποτελεσμάτων:

1. Παρατηρούμε ότι όσο αυξάνονται τα layers τόσο μικρότερο loss έχουμε.
2. Ακόμα βλέπουμε η μείωση δεν είναι πολύ σημαντική από τα 3 στα 4 layers, ενώ με τα 4 έχουμε μεγαλύτερο χρόνο εκτέλεσης.

Συμπέρασμα: Τα καλύτερα αποτελέσματα στα παραπάνω πειράματα θεωρούμε ότι παρατηρούνται για 3 layers.

# **3. Batch Size**

Μέχρι τώρα έχουμε συμπεράνει ότι ο πιο κατάλληλος αριθμός layer είναι 3 και μέγεθος φίλτρων 3x3. Θα εκτελέσουμε ένα πείραμα στο οποίο θα μεταβάλλουμε το batch size από 128, 256, 512, κρατώντας fixed τις υπόλοιπες παραμέτρους (layers 3, μέγεθος φίλτρων 3x3, εναρκτήριος αριθμός φίλτρων 32, εποχές 10, διάσταση συμπίεσης 10).



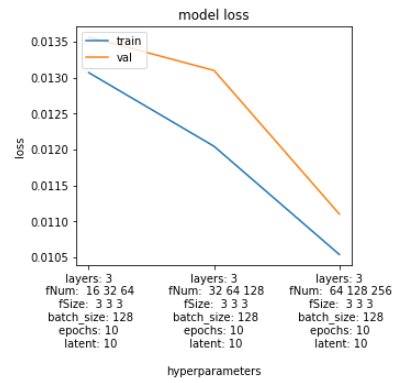
Σχολιασμός Αποτελεσμάτων:

1. Παρατηρούμε ότι όσο αυξάνεται το batch\_size τόσο χειρότερες τιμές έχουμε για train\_loss και val\_loss.
2. Επίσης βλέπουμε ότι η διαφορά μεταξύ train\_loss και val\_loss για 128 είναι αρκετά μικρή παραμένοντας σε χαμηλές τιμές.

Συμπέρασμα: Τα καλύτερα αποτελέσματα στo παραπάνω πείραμα θεωρούμε ότι παρατηρούνται για batch\_size 128.

# **4. Αριθμός Φίλτρων**

Μέχρι τώρα έχουμε συμπεράνει ότι ο πιο κατάλληλος αριθμός layer είναι 3, μέγεθος φίλτρων 3x3, και batch size 128. Όπως και με το μέγεθος φίλτρων μπορούμε να επιλέξουμε τον αριθμό φίλτρων σε κάθε layer. Στα παρακάτω διαγράμματα θα μελετήσουμε τις μεταβολές των loss για διάφορες εναρκτήριες τιμές αριθμού φίλτρων στο πρώτο layer, όπου σε κάθε επόμενο layer διπλάσιαζεται αυτός ο αριθμός.

Π.χ. 1ο layer: 8 > 2o layer: 16 > 3o layer: 32 > 4o layer: 64

Σχολιασμός Αποτελεσμάτων:

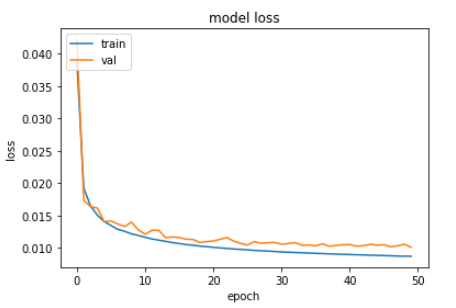
1. Παρατηρούμε ότι όσα περισσότερα είναι τα φίλτρα τόσο μικρότερα είναι τα train\_loss και val\_loss.
2. Επιπλέον, στο τρίτο πείραμα (64) η διαφορά μεταξύ train\_loss και val\_loss είναι αρκετά μικρότερη από τα άλλα δύο πειράματα.

Συμπέρασμα: Τα καλύτερα αποτελέσματα στα παραπάνω πειράματα θεωρούμε ότι παρατηρούνται για αρχικό αριθμό φίλτρων 64.

# **5. Αριθμός Εποχών**

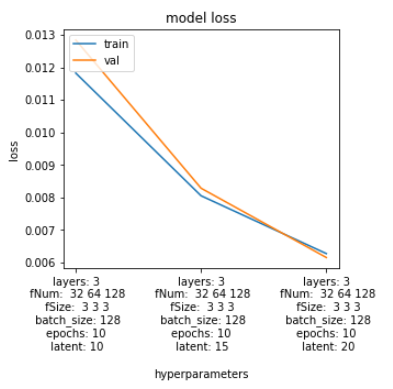
Μέχρι τώρα οι υπερπαράμετροι που έχουμε συμπεράνει ότι δίνουν το βέλτιστο αποτέλεσμα για το ζητούμενο μας είναι οι εξής:

* Μέγεθος Φίλτρων : 3x3
* Αριθμός layers : 3
* Batch size : 128
* Αριθμός φίλτρων : 64 ( στο πρώτο layer )

Οπότε εκπαιδεύσαμε το μοντέλο για 50 εποχές με τις παραπάνω τιμές για να δούμε που συγκλίνει.

Συμπέρασμα: Σύμφωνα με το παραπάνω διάγραμμα θεωρούμε ότι ο αριθμός εποχών για τον οποίο τα loss συγκλίνουν ικανοποιητικά είναι 10.

# **6. Διάσταση Συμπίεσης**

  
Με βάση όλα τα προηγούμενα πειράματα , τώρα θα πειραματιστούμε ως προς την διάσταση συμπίεσης (bottleneck). Κρατώντας σταθερές τις υπόλοιπες υπερπαραμέτρους, δοκιμάζουμε για διάσταση 10, 15, 20.

Σχολιασμός Αποτελεσμάτων:

1. Παρατηρούμε ότι όσο μεγαλύτερη είναι η διάσταση συμπίεσης τόσο μικρότερες τιμές έχουμε στα train\_loss και val\_loss.
2. Tο ζήτημά εδώ όμως είναι η μείωση των διαστάσεων ώστε να επιτευχθεί καλύτερη απόδοση στην αναζήτηση στην συνέχεια.

Συμπέρασμα: Με βάση τις δύο παραπάνω παρατηρήσεις κρατάμε για διάσταση συμπίεσης την τιμή 10.