ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΜΟΥΣΙΚΗΣ ΚΑΙ ΟΠΤΟΑΚΟΥΣΤΙΚΩΝ ΤΕΧΝΟΛΟΓΙΩΝ

Τμήμα Μουσικής Τεχνολογίας και Ακουστικής

Πτυχιακή εργασία

Αναγνώριση Ακουστικών Γεγονότων με Βαθιά Μάθηση: Μια Εφαρμογή στο Σύνολο Δεδομένων FSD Kaggle 2018

ΑΝΔΡΕΑΣ ΚΑΤΣΙΝΟΥΛΑΣ

Επιβλέπων: ***Καλιακάτσος-Παπακώστας Μάξιμος***

Αναπληρωτής Καθηγητής

Ρέθυμνο, Ιούνιος 2024

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΜΟΥΣΙΚΗΣ ΚΑΙ ΟΠΤΟΑΚΟΥΣΤΙΚΩΝ ΤΕΧΝΟΛΟΓΙΩΝ

Τμήμα Μουσικής Τεχνολογίας και Ακουστικής

Αναγνώριση Ακουστικών Γεγονότων με Βαθιά Μάθηση: Μια Εφαρμογή στο Σύνολο Δεδομένων FSD Kaggle 2018

Επιβλέπων: ***Καλιακάτσος-Παπακώστας Μάξιμος***

Αναπληρωτής Καθηγητής

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την [Ημερομηνία …η …. ….].

.....	.....	.....
XXX XXX	XXX XXX	XXX XXX
Βαθμίδα	Βαθμίδα	Βαθμίδα

Ρέθυμνο, Ιούνιος 2024

Πνευματικά δικαιώματα

Η έγκριση της πτυχιακής εργασίας από το Τμήμα Μουσικής Τεχνολογίας & Ακουστικής του Ελληνικού Μεσογειακού Πανεπιστημίου δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαίτερα τον καθηγητή κ. Μάξιμο Καλιακάτσο-Παπακώστα για την πολύτιμη καθοδήγηση και υποστήριξή του καθ' όλη τη διάρκεια της πτυχιακής μου εργασίας. Επίσης, θα ήθελα να εκφράσω την ευγνωμοσύνη μου προς τους γονείς μου για την αδιάκοπη υποστήριξη και την ενθάρρυνσή τους καθ' όλη τη διάρκεια των ακαδημαϊκών μου χρόνων. Χωρίς την αγάπη, την πίστη και την οικονομική στήριξή τους, η επίτευξη των στόχων μου δεν θα ήταν δυνατή.

ΠΕΡΙΛΗΨΗ

Η παρούσα πτυχιακή εργασία επικεντρώνεται στην αναγνώριση ήχων μέσα στο πλαίσιο του συνόλου δεδομένων FSD Kaggle 2018, το οποίο περιλαμβάνει μια ποικιλία ακουστικών γεγονότων, όπως μουσικά όργανα, ανθρώπινοι ήχοι, ήχοι στο σπίτι και ήχοι ζώων. Ο κύριος στόχος είναι η αποτελεσματική χρήση της μηχανικής μάθησης, κυρίως της βαθιάς μάθησης, για την ακριβή αναγνώριση και ταξινόμηση αυτών των ποικίλων ηχητικών σημάτων.

Η έρευνα αξιοποιεί διάφορα εργαλεία και βιβλιοθήκες, όπως η Python, το PyTorch και η Librosa, για την προεπεξεργασία και ανάλυση των ηχητικών δεδομένων. Επικεντρώνεται σε βασικές έννοιες, όπως η ψηφιακή Επεξεργασία Σήματος (Digital Signal Processing - DSP), με έμφαση στη σημασία του Μετασχηματισμού Fourier για τη μετατροπή των ηχητικών σημάτων σε κατάλληλη μορφή για την είσοδο στα μοντέλα.

Η μελέτη αναφέρει τις προκλήσεις που αντιμετωπίζονται κατά την προσπάθεια να κατατεθούν ακατέργαστα ηχητικά δεδομένα απευθείας σε νευρωνικά δίκτυα, υπογραμμίζοντας την αναγκαιότητα τεχνικών όπως ο Μετασχηματισμός Fourier Σύντομου Χρόνου (Short-Time Fourier Transform - STFT) για την αποτελεσματική εξαγωγή χαρακτηριστικών. Ο STFT επιτρέπει τη χρονική-συχνοτική αναπαράσταση των ηχητικών σημάτων, αντιμετωπίζοντας τον περιορισμό της σταθερής ανάλυσης σε χρόνο και συχνότητα.

Επιπλέον, η πτυχιακή εργασία παρουσιάζει τις ευρύτερες έννοιες της μηχανικής μάθησης και της βαθιάς μάθησης, αναδεικνύοντας τις εφαρμογές τους σε διάφορους τομείς, συμπεριλαμβανομένης της όρασης υπολογιστή, της αναγνώρισης ομιλίας, της επεξεργασίας φυσικής γλώσσας και άλλων. Η βαθιά μάθηση, συγκεκριμένα, εξερευνάται ως ένα ισχυρό παράδειγμα μέσα στον τομέα, με τις αρχιτεκτονικές των νευρωνικών δικτύων να υπόσχονται υψηλή απόδοση στην αναγνώριση και ταξινόμηση ηχητικών σημάτων.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

Κατάλογος Πινάκων και Εικόνων

1 Εισαγωγή

1.1 Βασικά στοιχεία θεωρίας

Για την υλοποίηση της πτυχιακής εργασίας χρησιμοποιήθηκαν διάφορα εργαλεία όπως η python και πολλές διαφορετικές βιβλιοθήκες και Frameworks όπως το Pytorch , η Librosa και το Tensorboard. Σε θεωρητικό επίπεδο θα πρέπει να γινει κατανόηση της ψηφιακής επεξεργασίας σήματος (DSP) ώστε να μπορέσουμε να επεξεργαστούμε τα σήματα και να τα φέρουμε στη μορφή που θέλουμε με τη χρήση του μετασχηματισμού Fourier.

1.1.1 Python

Η Python είναι μια διερμηνευόμενη (interpreted), αντικειμενοστραφής γλώσσα προγραμματισμού. Είναι υψηλού επιπέδου και για τον λόγο αυτό χρησιμοποιείται αρκετά για την ταχεία ανάπτυξη εφαρμογών. Ο διερμηνέας (interpreter) της Python και η εκτεταμένη πρότυπη βιβλιοθήκη είναι διαθέσιμα σε πηγαία ή δυαδική μορφή για όλες τις κύριες πλατφόρμες και μπορούν να διανεμηθούν ελεύθερα.

1.1.2 Librosa

Η Librosa είναι μια βιβλιοθήκη της Python για την ανάλυση ήχου. Κάθε εργασία που έχει να κάνει με ήχο είναι απο τις κυριότερες βιβλιοθήκες που χρησιμοποιούνται γιατι παρέχει πάρα πολλά εργαλεία για την ανάλυση και επεξεργασία ήχου με τη χρήση της Python.

Με τη Librosa μπορούμε να φορτώσουμε τους ήχους και να τους επεξεργαστούμε και να τους αναπαραστήσουμε σαν εικόνα. Η αναπαράση του ήχου σε μορφή εικόνας ονομάζεται time domain representation (αναπαράσταση χρονικού πεδίου) και μας δείχνει την ένταση του ήχου αν μεταβάλλεται σε σχέση με το χρόνο.

1.1.3 Γρήγορος μετασχηματισμός Fourier (FFT)

Ο γρήγορος μετασχηματισμός Fourier είναι ένας αλγόριθμος με τον οποίο μπορούμε να υπολογίσουμε το διακριτό μετασχηματισμό Fourier (DFT) μιας ακολουθίας ή τον αντίστροφο της. Η ανάλυση Fourier μας βοηθάει στην μετατροπή ενός σήματος απο το αρχικό του πεδίο ένταση στο πεδίο του χρόνου σε μια αναπαράσταση στο πεδίο της συχνότητας. Ο DFT λαμβάνεται με την αποσύνθεση μιας ακολουθίας τιμών σε συνιστώσες διαφορετικών συχνοτήτων. Η πράξη αυτή είναι χρήσιμη σε πολλούς τομείς, αλλά ο υπολογισμός της απευθείας από τον ορισμό είναι συχνά πολύ αργός για να είναι πρακτικός. Ένας FFT υπολογίζει γρήγορα τέτοιους μετασχηματισμούς με παραγοντοποίηση του πίνακα DFT σε ένα γινόμενο αραιών (κυρίως μηδενικών) παραγόντων. Με την παρουσία σφάλματος στρογγυλοποίησης, πολλοί αλγόριθμοι FFT είναι πολύ πιο ακριβείς από την αξιολόγηση του ορισμού DFT άμεσα ή έμμεσα.

1.1.4 Μετασχηματισμός Fourier σύντομου χρόνου (Short-time Fourier transform -STFT)

Ο μετασχηματισμός Fourier μικρού χρόνου (STFT) χρησιμοποιείται για τη υλοποίηση μετασχηματισμών ήχου. Ένας περιορισμός του STFT είναι οτι κανένας ήχος δεν μπορεί να αναλυθεί με βέλτιστη χρονική και συχνοτική ανάλυση ταυτόχρονα. Επίσης η χρονική και συχνοτική ανάλυση είναι σταθερή τόσο κατα τη διάρκεια της συχνότητας όσο και κατα τη διάρκεια του χρόνου. Ο STFT χρησιμοποιείται για την αναπαράσταση σημάτων απο το πεδίο του χρόνου σε αναπαράσταση χρόνου-συχνότητας.

1.1.5 Μηχανική Μάθηση (Machine Learning)

Η μηχανική μάθηση είναι μια σειρά απο αλγόριθμους που βοηθά στην πρόβλεψη αποτελεσμάτων χωρίς να είναι προγραμματισμένοι για να το κάνουν. Οι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούν ιστορικά δεδομένα ως είσοδο για να προβλέψουν νέες τιμές εξόδου.

Η κλασική μηχανική μάθηση συχνά κατηγοριοποιείται με βάση τον τρόπο με τον οποίο ένας αλγόριθμος μαθαίνει να γίνεται πιο ακριβής στις προβλέψεις του. Υπάρχουν τέσσερις βασικές προσεγγίσεις: μάθηση με επίβλεψη, μάθηση χωρίς επίβλεψη, μάθηση με ημιεπίβλεψη και ενισχυτική μάθηση. Ο τύπος αλγορίθμου που επιλέγουν να χρησιμοποιήσουν οι επιστήμονες δεδομένων εξαρτάται από τον τύπο των δεδομένων που θέλουν να προβλέψουν.

1.1.6 Βαθιά Μάθηση (Deep learning)

Το deep learning αποτελεί μέρος της μηχανικής μάθησης που βασίζεται σε τεχνητά νευρωνικά δίκτυα με μάθηση αναπαράστασης. Η μάθηση μπορεί να είναι εποπτευόμενη, ημι-εποπτευόμενη ή μη εποπτευόμενη. Αρχιτεκτονικές βαθιάς μάθησης όπως βαθιά νευρωνικά δίκτυα, δίκτυα βαθιάς πεποίθησης, μάθηση βαθιάς ενίσχυσης, επαναλαμβανόμενα νευρωνικά δίκτυα και συνελικτικά νευρωνικά δίκτυα έχουν εφαρμοστεί σε πεδία όπως η όραση υπολογιστή (Computer Vision), η αναγνώριση ομιλίας, η επεξεργασία φυσικής γλώσσας, η μηχανική μετάφραση, η βιοπληροφορική, ο σχεδιασμός φαρμάκων κ.α όπου παρήγαγαν αποτελέσματα συγκρίσιμα και σε ορισμένες περιπτώσεις ξεπερνούν τις επιδόσεις των ανθρώπινων ειδικών.

1.1.7 Σύνολο Δεδομένων (Data set)

Τα δεδομένα αποτελούν βασικό συστατικό κάθε μοντέλου τεχνητής νοημοσύνης και, ουσιαστικά, είναι ο μοναδικός λόγος για την έξαρση της δημοτικότητας της μηχανικής μάθησης που παρατηρούμε σήμερα. Το σύνολο δεδομένων περιέχει πολλά ξεχωριστά κομμάτια δεδομένων, αλλά μπορεί να χρησιμοποιηθεί για την εκπαίδευση ενός αλγορίθμου με στόχο την εύρεση προβλέψιμων μοτίβων μέσα στο σύνολο των δεδομένων.

2 Προσέγγιση του συγκεκριμένου θέματος

2.1 Επεξεργασία του Dataset και STFT

Το σύνολο δεδομένων είναι το FSDKaggle2018, το οποίο αποτελείται από δείγματα ήχου από το Freesound, σχολιασμένα με τη χρήση ενός λεξιλογίου 41 ετικετών από το AudioSet Ontology της Google:

- Tearing
- Bus
- Shatter
- Gunshot, gunfire
- Fireworks
- Writing
- Computer keyboard
- Scissors
- Microwave oven
- Keys jangling
- Drawer open or close
- Squeak
- Knock
- Telephone
- Saxophone
- Oboe
- Flute
- Clarinet
- Acoustic guitar
- Tambourine
- Glockenspiel
- Gong
- Snare drum
- Bass drum
- Hi-hat
- Electric piano
- Harmonica
- Trumpet
- Violin, fiddle
- Double bass
- Cello
- Chime
- Cough
- Laughter
- Applause
- Finger snapping
- Fart
- Burping, eructation
- Cowbell
- Bark
- Meow

Για την εκπαίδευση του νευρωνικού δικτύου θα χρησιμοποιηθεί το παραπάνω σύνολο δεδομένων. Όλα τα δείγματα ήχου σε αυτό το σύνολο δεδομένων παρέχονται ως ασυμπίεστα αρχεία ήχου PCM 16 bit, 44,1 kHz, μονοφωνικά.

Η αναπαράσταση του ήχου μπορεί να γίνει με διάφορους τρόπους, ο καλύτερος τρόπος εξαρτάται πάντα από την χρήση που επιθυμούμε. Τα φασματογραφήματα είναι δισδιάστατες εικόνες που αναπαριστούν ακολουθίες φασμάτων με το χρόνο κατά μήκος ενός άξονα, τη συχνότητα και τη φωτεινότητα ή το χρώμα που αντιπροσωπεύει την ένταση μιας συχνότητας σε κάθε χρονικό πλαίσιο.

Εξετάζουμε τη χρήση ενός δικτύου με αρχιτεκτονική που λαμβάνει ως είσοδο κυματομορφές χρονοσειρών, παρουσιάζοντας τις ως ένα μακρύ διάνυσμα 1D. Ενώ αυτή η προσέγγιση είναι εφικτή, σκεφτόμαστε επίσης τον πιθανό ρόλο του φασματογραφήματος στη βελτίωση της απόδοσης του ταξινομητή. Έτσι, εξετάζουμε τη διαδικασία μετατροπής όλων των δειγμάτων σε μετασχηματισμό Fourier μικρού χρόνου, προκειμένου να αξιοποιήσουμε την πλούσια πληροφορία που παρέχει στο πεδίο της συχνότητας.

Ο μετασχηματισμός Fourier μικρού χρόνου (STFT) προκύπτει από τον υπολογισμό του μετασχηματισμού Fourier για διαδοχικά πλαίσια ενός σήματος.

$$X(m,\omega) = \sum_{n} x(n)w(n-m)e^{-j\omega n}$$

Καθώς αυξάνεται το m, ολισθαίνουμε τη συνάρτηση παραθύρου w προς τα δεξιά. Για το προκύπτον πλαίσιο, x(n)w(n-m), υπολογίζουμε τον μετασχηματισμό Fourier. Επομένως, ο STFT X είναι συνάρτηση τόσο του χρόνου, m, όσο και της συχνότητας, ω.

Με τη βιβλιοθηκη της python Librosa χρησιμοποιούμε τη συνάρτηση librosa.stft που υπολογίζει ένα STFT. Του δίνουμε ένα μέγεθος πλαισίου, δηλαδή το μέγεθος του FFT, και ένα μήκος hop, δηλαδή τον αριθμό δειγμάτων μεταξύ διαδοχικών πλαισίων στον υπολογισμό του STFT:

```
hop_length = 512
n_fft = 2048
X = librosa.stft(x, n_fft=n_fft, hop_length=hop_length)
```

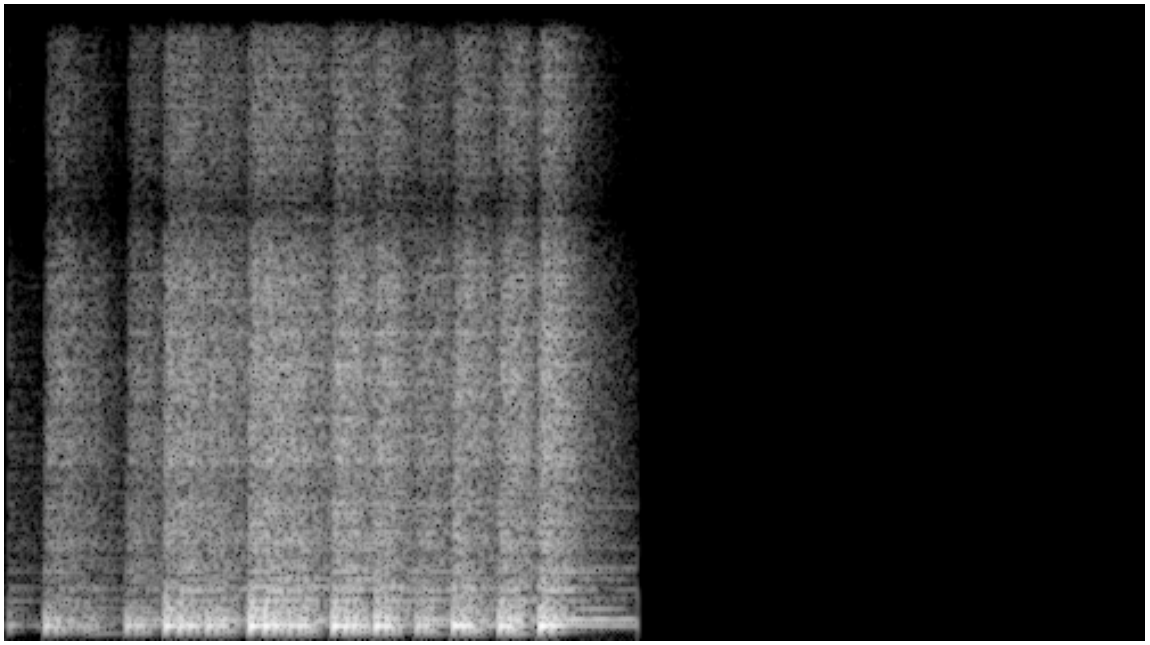
Για να υπολογίσω το hop_length και το n_fft σε μονάδες δευτερολέπτων αρκεί μόνο να τα διαιρέσω με το sampling rate:

```
float(hop_length)/sr
float(n_fft)/sr
```

Το φασματογράφημα δείχνει την ένταση των συχνοτήτων με την πάροδο του χρόνου. Ένα φασματογράφημα είναι απλώς το τετραγωνικό μέγεθος του STFT:

$$S(m,\omega) = |X(m,\omega)|^2$$

Ένα παράδειγμα φάσματος είναι η παρακάτω εικόνα η οποία είναι ένα τυχαίο δείγμα απο το dataset.



Εικόνα. 1. gray scale δείγμα (Snare-drum) με zero padding.

2.2.1 Βαθιά Μάθηση (Deep Learning)

Στον κλάδο της μηχανικής μάθησης η βαθιά μάθηση έχει μεγάλη επιτυχία και μεγάλο ερευνητικό ενδιαφέρων. Το συνελικτικό νευρωνικό δίκτυο (CNN) είναι ένα από τα από τα πιο δημοφιλή και χρησιμοποιούμενα δίκτυα DL, λόγο του CNN η βαθιά μάθηση έχει γίνει δημοφιλής στις μέρες μας. Το κύριο πλεονέκτημα των CNN σε σύγκριση με τους προκατόχους του είναι ότι ανιχνεύει αυτόματα τα σημαντικά χαρακτηριστικά χωρίς ανθρώπινη επίβλεψη. Για την πειραματική διαδικασία απαιτείται η κατανόηση των πτυχών της βαθιάς μάθησης συμπεριλαμβανομένων των εννοιών και των προκλήσεων που προκύπτουν για να καταλήξουμε στην αρχιτεκτονική που επιθυμούμε για την πειραματική διαδικασία.

Οι τεχνικές DL ταξινομούνται σε τρεις μεγάλες κατηγορίες: μη επιβλεπόμενες, μερικώς επιτηρούμενες (ημι-επιτηρούμενες) και επιβλεπόμενες. Επιπλέον, η βαθιά ενισχυτική μάθηση (DRL), επίσης γνωστή ως Reinforcement Learning (RL), είναι ένας άλλος τύπος τεχνικής μάθησης, ο οποίος θεωρείται ως επί το πλείστον ότι εμπίπτει στην κατηγορία της μερικώς επιβλεπόμενης (και περιστασιακά μη επιβλεπόμενης) τεχνικής μάθησης.

2.2.2 Deep supervised learning (Βαθιά Επιτηρούμενη Μάθηση)

Αυτή η τεχνική ασχολείται με δεδομένα με ετικέτες. Κατά την εξέταση μιας τέτοιας τεχνικής, οι περιβάλλουσες έχουν μια συλλογή εισόδων και αποτελεσμάτων εξόδων. Οι παράμετροι του δικτύου ενημερώνονται επανειλημμένα από τον πράκτορα (agent) για να αποκτήσει μια βελτιωμένη εκτίμηση για τις προτεινόμενες εξόδους. Μετά από ένα θετικό αποτέλεσμα εκπαίδευσης, ο πράκτορας αποκτά την ικανότητα να λαμβάνει τις σωστές λύσεις στα ερωτήματα από το περιβάλλον. Για το DL, υπάρχουν διάφορες τεχνικές μάθησης με επίβλεψη, όπως τα αναδρομικά νευρωνικά δίκτυα (RNN), τα νευρωνικά δίκτυα συνέλιξης (CNN) και τα βαθιά νευρωνικά δίκτυα (DNN). Επιπλέον, η κατηγορία RNN περιλαμβάνει τις προσεγγίσεις gated recurrent units (GRUs) και long short-term memory (LSTM). Το κύριο πλεονέκτημα αυτής της τεχνικής είναι η δυνατότητα συλλογής δεδομένων ή παραγωγής μιας εξόδου δεδομένων από την προηγούμενη γνώση. Ωστόσο, το μειονέκτημα αυτής της τεχνικής είναι ότι το όριο απόφασης μπορεί να είναι υπερβολικά περιορισμένο όταν το σύνολο εκπαίδευσης δεν διαθέτει δείγματα που θα έπρεπε να ανήκουν σε μια κατηγορία. Συνολικά, αυτή η τεχνική είναι απλούστερη από άλλες τεχνικές στον τρόπο εκμάθησης με υψηλή απόδοση.

2.2.3 Deep semi-supervised learning (Μερικώς Επιτηρούμενη Μάθηση)

Η τεχνική της μερικώς επιβλεπόμενης μάθησης βασίζεται σε ημι-επισημειωμένα σύνολα δεδομένων, εκμεταλλευόμενη τόσο επισημειωμένα όσο και μη επισημειωμένα δεδομένα για την εκπαίδευση. Κατά καιρούς, τεχνικές όπως τα generative adversarial

networks (GAN) και τα deep reinforcement learning (DRL) εφαρμόζονται με παρόμοιο τρόπο. Επιπλέον, τα Recurrent Neural Networks (RNNs) με τις εκδοχές τους όπως οι Gated Recurrent Units (GRUs) και τα Long Short-Term Memory networks (LSTMs) χρησιμοποιούνται ευρέως για μερικώς επιβλεπόμενη μάθηση.

Ένα από τα πλεονεκτήματα αυτής της προσέγγισης είναι η ελαχιστοποίηση του αριθμού των απαιτούμενων επισημειωμένων δεδομένων, ενώ παράλληλα επισημαίνεται ένα μειονέκτημα: η πιθανή επίδραση ασχετοσύνης των μη επισημειωμένων δεδομένων εκπαίδευσης στις αποφάσεις του μοντέλου.

Στον τομέα της ταξινόμησης εγγράφων κειμένου, η μερικώς επιβλεπόμενη μάθηση είναι εξαιρετικά χρήσιμη, καθώς επιτρέπει την αντιμετώπιση της πρόκλησης της δυσκολίας απόκτησης μεγάλου όγκου επισημειωμένων εγγράφων κειμένου.

Ενδεχομένως, η μετάβαση σε εφαρμογές που σχετίζονται με τον ήχο, όπως το few-shot learning (λίγων δειγμάτων), μπορεί να προσφέρει μια ευρύτερη κατανόηση της μερικώς επιβλεπόμενης μάθησης στο πλαίσιο της επεξεργασίας ήχου.

2.2.4 Deep reinforcement learning (Βαθιά ενισχυτική μάθηση)

Η ενισχυτική μάθηση λειτουργεί μέσω αλληλεπίδρασης με το περιβάλλον, σε αντίθεση με τις υπόλοιπες τεχνικές μηχανικής μάθησης. Αυτή η τεχνική αξιοποιήθηκε το 2013 από την Google DeepMind για την ανάπτυξη του AlphaGo, το οποίο μπορεί να νικήσει τον παγκόσμιο πρωταθλητή στο επιτραπέζιο παιχνίδι Go, γεγονός που φαινόταν αδύνατο να συμβεί, καθώς το Go έχει εκθετικά περισσότερες διακλαδώσεις σε σχέση με το σκάκι και θεωρείται άθλημα κυρίως "διαίσθησης" παρά "αριθμητικής λογικής". Σε σύγκριση με τις παραδοσιακές τεχνικές με επίβλεψη, η εκτέλεση αυτής της μάθησης είναι πολύ πιο δύσκολη, καθώς δεν υπάρχουν απλές συναρτήσεις απώλειας διαθέσιμες στην τεχνική ενισχυτικής μάθησης.

Επιπλέον, υπάρχουν δύο ουσιαστικές διαφορές μεταξύ της μάθησης με επίβλεψη και της μάθησης με ενίσχυση (reinforcement learning). Πρώτον, δεν υπάρχει αναλυτική συνάρτηση σφάλματος, η οποία απαιτεί βελτιστοποίηση, πράγμα που σημαίνει ότι θα πρέπει να

αξιολογηθεί μέσω αλληλεπίδρασης. Δεύτερον, η κατάσταση με την οποία γίνεται αλληλεπίδραση βασίζεται σε ένα περιβάλλον, όπου η είσοδος εξαρτάται από προηγούμενες ενέργειες.

Η Ενισχυτική Μάθηση έχει έναν πράκτορα (agent) μάθησης ο οποίος βελτιώνεται με βάση μια λειτουργία ανταμοιβής. Για την επίλυση μιας εργασίας, η επιλογή του τύπου της ενισχυτικής μάθησης που πρέπει να εκτελεστεί βασίζεται στο χώρο ή στο πεδίο εφαρμογής του προβλήματος. Για παράδειγμα, η βαθιά ενισχυτική μάθηση είναι ο καλύτερος τρόπος για προβλήματα που περιλαμβάνουν πολλές παραμέτρους προς βελτιστοποίηση. Αντίθετα, η ενισχυτική μάθηση χωρίς παράγωγα είναι μια τεχνική που αποδίδει καλά για προβλήματα με περιορισμένες παραμέτρους.

Ορισμένες από τις εφαρμογές της ενισχυτικής μάθησης είναι η επιχειρηματική επενδυτική στρατηγική και η ρομποτική για βιομηχανικό αυτοματισμό. Το κύριο μειονέκτημα της ενισχυτικής μάθησης είναι ότι οι παράμετροι μπορεί να επηρεάσουν την ταχύτητα της μάθησης. Τα κίνητρα για τη χρήση της βαθιάς ενισχυτικής μάθησης είναι ότι βοηθάει να προσδιοριστεί ποια ενέργεια παράγει την υψηλότερη ανταμοιβή για δεδομένες συνθήκες του περιβάλλοντος και επιτρέπει να κατανοηθεί η καλύτερη προσέγγιση για την επίτευξη μεγάλων ανταμοιβών.

2.3 Τύποι δικτύων Βαθιάς Μάθησης

2.3.1 Αναδρομικά νευρωνικά δίκτυα (Recursive neural networks)

Το Hierarchical Recurrent Neural Network (RvNN) είναι ένα είδος αρχιτεκτονικής νευρωνικού δικτύου που επεκτείνει τα απλά αναδρομικά νευρωνικά δίκτυα (RNNs) για την επεξεργασία ιεραρχικών δομών. Ενώ τα απλά RNNs έχουν τη δυνατότητα να αντιληφθούν χρονικές σειρές δεδομένων, τα Hierarchical RNNs επιτρέπουν την αναγνώριση και εξαγωγή πληροφοριών από πολυεπίπεδες ιεραρχίες. Η αναδρομική αυτο-συνδετική μνήμη (RAAM) είναι η πρωταρχική έμπνευση για την ανάπτυξη του RvNN. Η αρχιτεκτονική RvNN δημιουργείται για την επεξεργασία αντικειμένων, τα οποία έχουν τυχαία διαμορφωμένες δομές όπως γράφοι ή δέντρα. Αυτή η προσέγγιση παράγει μια κατανεμημένη αναπαράσταση

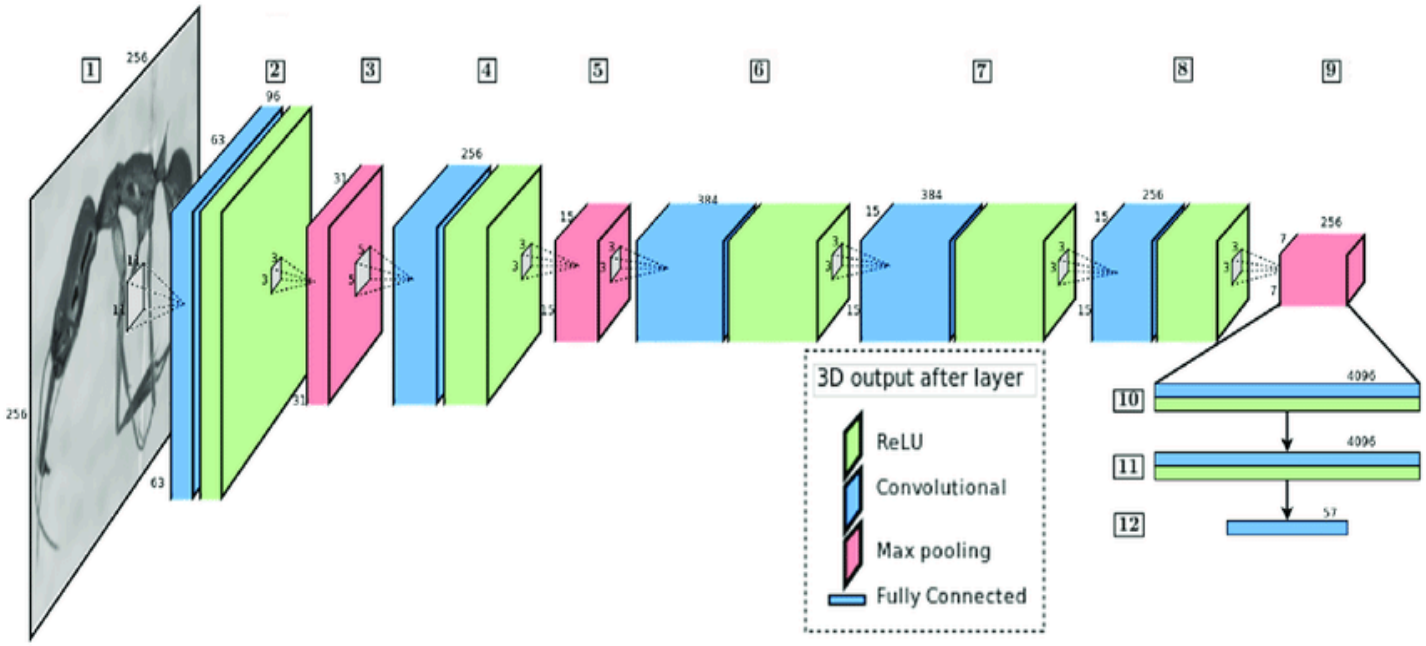
σταθερού πλάτους από μια μεταβλητού μεγέθους αναδρομική δομή δεδομένων. Το δίκτυο εκπαιδεύεται χρησιμοποιώντας ένα εισαγόμενο σύστημα εκμάθησης με τη χρήση του αλγορίθμου back-propagation. Το σύστημα BTS παρακολουθεί την ίδια τεχνική με τον αλγόριθμο back-propagation και έχει τη δυνατότητα να υποστηρίζει μια δενδροειδή δομή. Το RvNN υπολογίζει ένα πιθανό ζεύγος βαθμολογιών για συγχώνευση και κατασκευάζει ένα συντακτικό δέντρο. Επιπλέον, το RvNN υπολογίζει μια βαθμολογία που σχετίζεται με την πιθανότητα συγχώνευσης για κάθε ζεύγος μονάδων. Στη συνέχεια, το ζεύγος με τη μεγαλύτερη βαθμολογία συγχωνεύεται εντός ενός διανύσματος σύνθεσης. Μετά από κάθε συγχώνευση, το RvNN παράγει (α) μια μεγαλύτερη περιοχή με πολυάριθμες μονάδες, (β) ένα διάνυσμα σύνθεσης της περιοχής, και (γ) μια ετικέτα για την κλάση (για παράδειγμα, μια ουσιαστική φράση θα γίνει η ετικέτα κλάσης για τη νέα περιοχή εάν δύο μονάδες είναι ουσιαστικές λέξεις).

2.3.2 Επαναλαμβανόμενα νευρωνικά δίκτυα (Recurrent neural networks)

Τα RNNs είναι ένας ευρέως χρησιμοποιούμενος και γνωστός αλγόριθμος στην επιστήμη της βαθιάς μάθησης (Deep Learning). Σε αντίθεση με τα συμβατικά δίκτυα, τα RNN χρησιμοποιούν διαδοχικά δεδομένα στο δίκτυο. Δεδομένου ότι το η ενσωματωμένη δομή στην ακολουθία των δεδομένων παρέχει πολύτιμες πληροφορίες, αυτό το χαρακτηριστικό είναι θεμελιώδες για μια σειρά διαφορετικών εφαρμογών. Ένα επαναλαμβανόμενο νευρωνικό δίκτυο (RNN) είναι μια κατηγορία τεχνητών νευρωνικών δικτύων όπου οι συνδέσεις μεταξύ των κόμβων σχηματίζουν ένα κατευθυνόμενο ή μη κατευθυνόμενο γράφημα κατά μήκος μιας χρονικής ακολουθίας. Αυτό του επιτρέπει να παρουσιάζει διαχρονική δυναμική συμπεριφορά. Προερχόμενα από τα νευρωνικά δίκτυα τροφοδότησης, τα RNN μπορούν να χρησιμοποιούν την εσωτερική τους κατάσταση (μνήμη) για να επεξεργάζονται ακολουθίες εισόδων μεταβλητού μήκους. Αυτό τα καθιστά εφαρμόσιμα σε εργασίες όπως η μη τμηματοποιημένη, συνδεδεμένη αναγνώριση γραφής ή η αναγνώριση ομιλίας. Τόσο τα επαναλαμβανόμενα δίκτυα πεπερασμένης ώθησης όσο και τα

δίκτυα άπειρης ώθησης μπορούν να έχουν πρόσθετες αποθηκευμένες καταστάσεις και η αποθήκευση μπορεί να βρίσκεται υπό τον άμεσο έλεγχο του νευρωνικού δικτύου. Η αποθήκευση μπορεί επίσης να αντικατασταθεί από άλλο δίκτυο ή γράφημα, εάν αυτό ενσωματώνει χρονικές καθυστερήσεις ή έχει βρόχους ανάδρασης. Τέτοιες ελεγχόμενες καταστάσεις αναφέρονται ως ελεγχόμενη κατάσταση ή ελεγχόμενη μνήμη και αποτελούν μέρος των δικτύων μακράς βραχυπρόθεσμης μνήμης (LSTM) και των ελεγχόμενων αναδρομικών μονάδων.

2.3.3 Συνελικτικά Νευρωνικά Δίκτυα (Convolutional neural networks)



Εικόνα. 2. Παράδειγμα ενός με τα στρώματα (layers) CNN .

Στον τομέα της βαθιάς μάθησης, το CNN είναι ο πιο διάσημος και συνήθως χρησιμοποιούμενος αλγόριθμος. Το κύριο πλεονέκτημα του CNN σε σύγκριση με τους προκατόχους του είναι ότι αυτόματα προσδιορίζει τα σχετικά χαρακτηριστικά χωρίς ανθρώπινη επίβλεψη. Τα CNNs εφαρμόζονται εκτενώς σε μια σειρά διαφορετικών πεδίων, όπως η όραση υπολογιστών , η επεξεργασία ομιλίας , αναγνώριση προσώπου , κ.λπ. Η δομή των CNNs είναι εμπνευσμένη από τη δομή των νευρώνων στους εγκεφάλους ανθρώπων και ζώων. Πιο συγκεκριμένα, στον εγκέφαλο μιας γάτας, μια πολύπλοκη ακολουθία κυττάρων σχηματίζει τον οπτικό φλοιό, αυτή η ακολουθία προσομοιώνεται από το CNN. Οι Goodfellow και συν-συγγραφείς προσδιόρισαν τρία βασικά πλεονεκτήματα των Συνελικτικών Νευρωνικών Δικτύων (CNN) σε σχέση με τα πλήρως συνδεδεμένα δίκτυα (Fully Connected - FC): ισοδύναμες αναπαραστάσεις, αραιές αλληλεπιδράσεις και διαμοιρασμός παραμέτρων. Οι ισοδύναμες αναπαραστάσεις αναφέρονται στη δυνατότητα των CNN να αντιληφθούν χαρακτηριστικά σε διάφορα μέρη της εικόνας χωρίς την ανάγκη πλήρους σύνδεσης με κάθε πιθανό pixel. Οι αραιές αλληλεπιδράσεις αναφέρονται στο γεγονός ότι κάθε νευρώνας ενεργοποιείται μόνο από ένα μικρό υποσύνολο των προηγούμενων νευρώνων, μειώνοντας τη συνολική πολυπλοκότητα του δικτύου.

Ο διαμοιρασμός παραμέτρων αφορά το γεγονός ότι οι ίδιες παράμετροι χρησιμοποιούνται για την εξαγωγή χαρακτηριστικών σε διάφορα μέρη της εικόνας, επιτρέποντας την εκμάθηση πιο γενικευμένων χαρακτηριστικών. Είναι σημαντικό να σημειωθεί ότι, ενώ τα CNN μπορούν να χρησιμοποιηθούν σε αυθαίρετα πολλές διαστάσεις, σε αυτήν την εργασία εξετάζεται μόνο η περίπτωση των 2D δομών, όπως είναι συνήθης στην επεξεργασία εικόνας. Αυτή η λειτουργία χρησιμοποιεί έναν εξαιρετικά μικρό αριθμό παραμέτρων, γεγονός που απλοποιεί τη διαδικασία εκπαίδευσης και επιταχύνει το δίκτυο.

Αξίζει να σημειωθεί ότι μόνο μικρές περιοχές μιας σκηνής γίνονται αντιληπτές από αυτά τα κύτταρα και όχι η ολόκληρη τη σκηνή (δηλαδή, αυτά τα κύτταρα εξάγουν χωρικά την τοπική συσχέτιση που είναι διαθέσιμη στην είσοδο, όπως τα τοπικά φίλτρα πάνω στην είσοδο). Ένας ευρέως χρησιμοποιούμενος τύπος CNN αποτελείται από πολυάριθμα επίπεδα συνέλιξης που προηγούνται των επιπέδων υποδειγματοληψίας (pooling), ενώ τα τελικά στρώματα είναι στρώματα FC. Η είσοδος x κάθε στρώματος σε ένα μοντέλο CNN οργανώνεται σε τρεις διαστάσεις: ύψος, πλάτος και βάθος, ή m × m × r, όπου το ύψος (m) ισούται με το πλάτος. Το βάθος αναφέρεται επίσης ως αριθμός καναλιού. Για παράδειγμα, σε μια εικόνα RGB, το βάθος (r) είναι ισούται με τρία. Οι διάφοροι πυρήνες (φίλτρα) που είναι διαθέσιμοι σε κάθε στρώμα συνελικτικής ανάλυσης συμβολίζονται με k και έχουν επίσης τρεις διαστάσεις (n × n × q), παρόμοιες με την εικόνα εισόδου, ωστόσο, το n πρέπει να είναι μικρότερο από το m, ενώ το q είναι είτε ίσο είτε μικρότερο από το r. Επιπλέον, οι πυρήνες αποτελούν τη βάση των τοπικών συνδέσεων, οι οποίες μοιράζονται παρόμοιες παραμέτρους (bias b k και weight W k) για τη δημιουργία k χαρτών χαρακτηριστικών h k με μέγεθος (m - n - 1) ο καθένας. και συνελίσσονται με την είσοδο, όπως αναφέρθηκε παραπάνω. Το στρώμα συνέλιξης υπολογίζει ένα τετραγωνικό γινόμενο μεταξύ της εισόδου του και των βαρών, παρόμοια με το NLP, αλλά οι είσοδοι είναι υποδιαστασιολογημένες περιοχές του αρχικού μεγέθους της εικόνας. Στη συνέχεια, εφαρμόζοντας τη μη γραμμικότητα ή μια συνάρτησης ενεργοποίησης στην έξοδο του στρώματος συνέλιξης, λαμβάνουμε τα εξής:

$$h^k = f(W^k * x + b^k$$

Το επόμενο βήμα είναι η υποδειγματοληψία κάθε χάρτη χαρακτηριστικών στα στρώματα υποδειγματοληψίας. Αυτό οδηγεί σε μείωση των παραμέτρων του δικτύου, η οποία επιταχύνει τη διαδικασία εκπαίδευσης και με τη σειρά της επιτρέπει τον χειρισμό του προβλήματος της υπερπροσαρμογής (overfitting). Για όλους τους χάρτες χαρακτηριστικών, η συνάρτηση συγκέντρωσης (π.χ. max ή μέσος όρος) εφαρμόζεται σε μια γειτονική περιοχή μεγέθους p × p, όπου p είναι το μέγεθος του πυρήνα (kernel size). Τέλος, τα στρώματα FC λαμβάνουν τα χαρακτηριστικά μεσαίου και χαμηλού επιπέδου και δημιουργούν την αφαίρεση υψηλού επιπέδου, η οποία αντιπροσωπεύει τα στρώματα του τελευταίου σταδίου, όπως σε ένα τυπικό νευρωνικό δίκτυο. Οι βαθμολογίες ταξινόμησης δημιουργούνται με τη χρήση του τελικού στρώματος [π.χ. μηχανές διανυσμάτων υποστήριξης (SVM) ή softmax].

Για μια δεδομένη περίπτωση, κάθε βαθμολογία αντιπροσωπεύει την πιθανότητα μιας συγκεκριμένης κλάσης. Τα πλεονεκτήματα της χρήσης των CNN έναντι άλλων παραδοσιακών νευρωνικών δικτύων απαριθμούνται ως εξής:

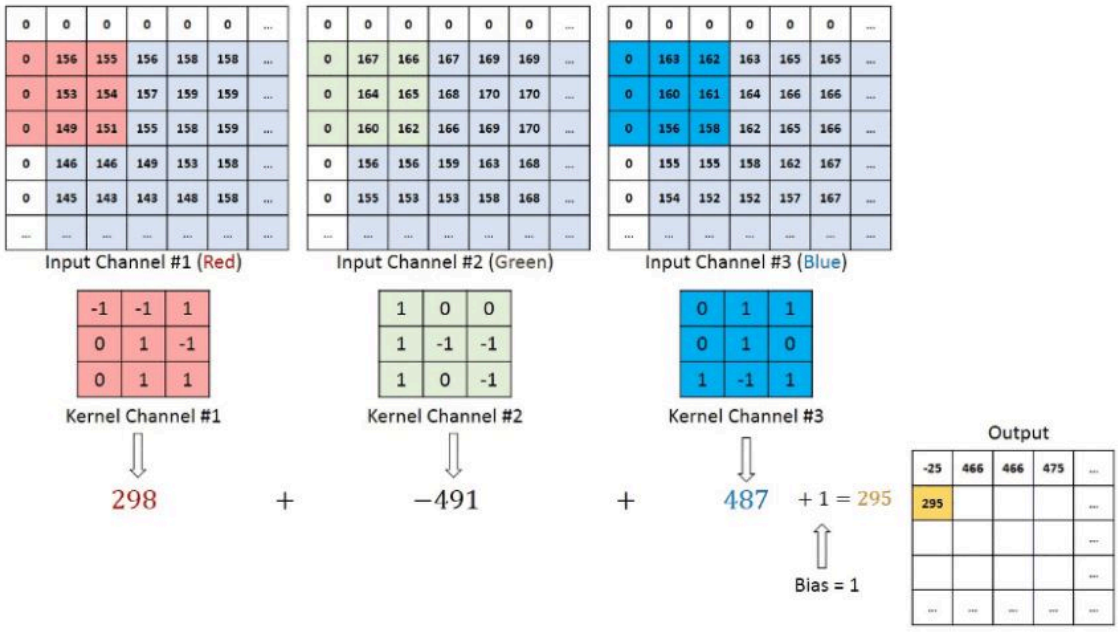
1. Ο κύριος λόγος για να εξετάσουμε το CNN είναι η δυνατότητα καταμερισμού βάρους, η οποία μειώνει τον αριθμό των εκπαιδεύσιμων παραμέτρων του δικτύου και με τη σειρά του βοηθά το δίκτυο να βελτιώσει γενίκευση και να αποφύγει την υπερβολική προσαρμογή.
2. Η ταυτόχρονη εκμάθηση των επιπέδων εξαγωγής χαρακτηριστικών και του επιπέδου ταξινόμησης προκαλεί την έξοδο του μοντέλου να είναι τόσο πολύ οργανωμένη όσο και πολύ εξαρτώμενη από τα εξαγόμενα χαρακτηριστικά
3. Η υλοποίηση δικτύων μεγάλης κλίμακας είναι πολύ πιο εύκολη με το CNN από ό,τι με άλλα νευρωνικά δίκτυα.

2.3.4 Στρώματα Συνελικτικού Νευρωνικού Δικτύου (CNN Layers)

Η αρχιτεκτονική CNN αποτελείται απο έναν αριθμό στρωμάτων - επιπέδων (ή αλλιώς πολυδομικά στοιχεία).

1. **Συνελικτικό επίπεδο (Convolutional Layer)**: Στην αρχιτεκτονική του CNN, το σημαντικότερο συστατικό είναι το στρώμα συνελίξεων. Αποτελείται απο φίλτρα συνελίξεων (Kernels). Η εικόνα εισόδου εκφράζεται ως N-dimensional metrics (N μετρικές διαστάσεις) και συνελίσεται με αυτά τα φίλτρα για τη δημιουργία του χάρτη των χαρακτηριστικών εξόδου (output feature map).
    - Ορισμός πυρήνα (Kernel): Το Kernel περιγράφεται απο ένα πλέγμα διακριτών αριθμών ή τιμών. Κάθε τιμή ονομάζεται βάρος του πυρήνα (Kernel weight). Οι τυχαίοι αριθμοί ανατίθενται για να λειτουργήσουν ως τα βάρη του πυρήνα στην αρχή της διαδικασίας εκπαίδευσης του CNN. Στη συνέχεια αυτά τα βάρη προσαρμόζονται σε κάθε epoch έτσι το Kernel μαθαίνει να εξάγει τα σημαντικά χαρακτηριστικά.
    - Συνελικτική Λειτουργία (Convolutional Operation): Αρχικά, περιγράφεται η μορφή εισόδου του CNN. Η διανυσματική μορφή είναι η είσοδος του παραδοσιακού νευρωνικού δικτύου, ενώ η

πολυκαναλική εικόνα είναι η είσοδος του CNN. Για παράδειγμα, μονοκάναλη είναι η μορφή της εικόνας σε κλίμακα του γκρι, ενώ η μορφή της εικόνας RGB είναι τρικάναλη. Για να γίνει κατανοητή η λειτουργία της συνελικτικής λειτουργίας, ας πάρουμε ένα παράδειγμα μιας εικόνας 4 × 4 κλίμακας του γκρι με έναν πυρήνα 2 × 2 με τυχαίο αρχικό βάρος. Αρχικά, ο πυρήνας ολισθαίνει σε ολόκληρη την εικόνα οριζόντια και κάθετα. Επιπλέον, προσδιορίζεται το τετραγωνικό γινόμενο μεταξύ της εικόνας εισόδου και του πυρήνα, όπου οι αντίστοιχες τιμές τους πολλαπλασιάζονται και στη συνέχεια αθροίζονται για να δημιουργηθεί μια ενιαία κλιμακωτή τιμή, η οποία υπολογίζεται ταυτόχρονα. Η όλη διαδικασία επαναλαμβάνεται στη συνέχεια μέχρι να μην είναι δυνατή περαιτέρω ολίσθηση. Στην παρακάτω εικόνα φαίνεται πως λειτουργεί ο πυρήνας σε μια είκονα με 3 κανάλια και πως υπολογίζεται η τιμή εισόδου στον χάρτη χαρακτηριστικών εξόδου.



Εικόνα. 3. Παράδειγμα Kernel σε εικόνα με 3 κανάλια RGB.

- Αραιή Συνδεσιμότητα (Sparse Connectivity): Κάθε νευρώνας ενός στρώματος (layer) στα FC νευρωνικά δίκτυα συνδέεται με όλους τους νευρώνες του επόμενου στρώματος σε αντίθεση με τα CNN που μόνο μερικά βάρη είναι μεταξύ δύο γειτονικών στρωμάτων. Έτσι ο αριθμός των απετούμενων βαρών ή συνδέσεων είναι μικρός, ενώ η μνήμη που χρειάζεται για την αποθήκευση των βαρών είναι επίσης μικρή.

- Διαμοιρασμός βάρους (Weight Sharing): Δεν υπάρχουν κατανεμημένα βάρη μεταξύ δύο νευρώνων γειτονικών στρωμάτων στα CNN, καθώς όλα τα βάρη λειτουργούν με ένα. Η εκμάθηση μιας ενιαίας ομάδας βαρών για ολόκληρη την είσοδο θα μειώσει σημαντικά τον απαιτούμενο χρόνο εκπαίδευσης, καθώς δεν είναι απαραίτητη η εκμάθηση πρόσθετων βαρών για κάθε νευρώνα.

2. **Στρώμα υποδειγματοληψίας (Pooling Layer)**: Η κύρια λειτουργία του στρώματος συγκέντρωσης είναι η υποδειγματοληψία στους χάρτες χαρακτηριστικών. Αυτοί οι χάρτες παράγονται ακολουθώντας τις πράξεις συνελίξεων. Η προσέγγιση αυτή συρρικνώνει χάρτες χαρακτηριστικών μεγάλου μεγέθους για τη δημιουργία μικρότερων χαρτών χαρακτηριστικών. Ταυτόχρονα, διατηρεί την πλειονότητα των κυρίαρχων πληροφοριών (ή χαρακτηριστικών) σε σε κάθε βήμα του σταδίου συγκέντρωσης. Με παρόμοιο τρόπο με τη συνελικτική λειτουργία, τόσο το stride όσο και ο πυρήνας αρχικοποιούνται πρίν απο την εκτελεση της λειτουργίας συγκέντρωσης. Διάφοροι τύποι μεθόδων συγκέντρωσης είναι διαθέσιμοι για χρήση σε διάφορα στρώματα συγκέντρωσης. Αυτές οι μέθοδοι περιλαμβάνουν τη δενδρική συγκέντρωση (tree pooling) , τη συγκέντρωση με πύλες (Gated pooling), τη μέση συγκέντρωση (Average Pooling), min pooling, max pooling, global average pooling (GAP) και global max pooling. Οι πιο γνωστές και συχνά χρησιμοποιούμενες μέθοδοι συγκέντρωσης είναι οι μέθοδοι max, min και GAP pooling.

3. **Συνάρτηση ενεργοποίησης (Activation Function)**: Η αντιστοίχιση της εισόδου στην έξοδο είναι η βασική λειτουργία όλων των τύπων συνάρτησης ενεργοποίησης σε όλους τους τύπους νευρωνικών δικτύων. Η τιμή εισόδου προσδιορίζεται με τον υπολογισμό του σταθμισμένου αθροίσματος της εισόδου του νευρώνα μαζί με τη μεροληψία του (εάν υπάρχει). Αυτό σημαίνει ότι η συνάρτηση ενεργοποίησης λαμβάνει την απόφαση σχετικά με το αν θα πυροδοτηθεί ή όχι ένας νευρώνας σε σχέση με μια συγκεκριμένη είσοδο δημιουργώντας την αντίστοιχη έξοδο. Τα στρώματα μη γραμμικής ενεργοποίησης χρησιμοποιούνται μετά από όλα τα στρώματα με βάρη (τα λεγόμενα μαθησιακά στρώματα, όπως τα στρώματα FC και τα συνελικτικά στρώματα) στην αρχιτεκτονική του CNN. Αυτή η μη γραμμική απόδοση των στρωμάτων ενεργοποίησης σημαίνει ότι η απεικόνιση της εισόδου στην έξοδο θα είναι μη γραμμική, τα στρώματα αυτά δίνουν στο CNN τη δυνατότητα να μαθαίνει εξαιρετικά περίπλοκα αντικείμενα. Η συνάρτηση ενεργοποίησης πρέπει επίσης να έχει τη δυνατότητα διαφοροποίησης, η οποία είναι ένα εξαιρετικά σημαντικό χαρακτηριστικό, καθώς επιτρέπει την οπισθοδιάδοση σφαλμάτων (back-propagation) να χρησιμοποιηθεί για την εκπαίδευση του δικτύου. Οι ακόλουθοι τύποι συναρτήσεων ενεργοποίησης είναι οι πιο συνήθως χρησιμοποιούνται στα CNN και σε άλλα βαθιά νευρωνικά δίκτυα.

    - **Sigmoid**: Η είσοδος αυτής της συνάρτησης ενεργοποίησης είναι πραγματικοί αριθμοί, ενώ η έξοδος περιορίζεται μεταξύ μηδέν και ένα. Η καμπύλη της σιγμοειδούς συνάρτησης έχει σχήμα S και μπορεί να αναπαρασταθεί μαθηματικά από την παρακάτω εξίσωση:

      $$f(x)sigm = 1/1 + e^{-x}$$

    - **Tanh**: Είναι παρόμοια με τη σιγμοειδή συνάρτηση, καθώς η είσοδός της είναι πραγματικοί αριθμοί, αλλά η έξοδος περιορίζεται μεταξύ - 1 και 1. Μπορεί να αναπαρασταθεί μαθηματικά από την παρακάτω εξίσωση:

$$f\,(x)tanh\;=\;e^{x}-e^{-x}\,/\,e^{x}\;+\;e^{-x}$$

- **ReLU**: Η πιο συχνά χρησιμοποιούμενη συνάρτηση στο πλαίσιο του CNN. Μετατρέπει ολόκληρες τιμές της εισόδου σε θετικούς αριθμούς. Το χαμηλότερο υπολογιστικό φορτίο είναι η το κύριο πλεονέκτημα της ReLU έναντι των άλλων. Μπορεί να αναπαρασταθεί μαθηματικά από την παρακάτω εξίσωση:

$$f\,(x)ReLU\;=\;max(0,\,x)$$

Περιστασιακά, ενδέχεται να προκύψουν ορισμένα σημαντικά προβλήματα κατά τη χρήση του ReLU. Για το για παράδειγμα, θεωρήστε έναν αλγόριθμο οπισθοδιάδοσης σφάλματος (Back-propagation) με μεγαλύτερη κλίση που τον διατρέχει. Το πέρασμα αυτής της κλίσης μέσα στη συνάρτηση ReLU θα ενημερώσει το βάρη με τρόπο που κάνει τον νευρώνα να μην ενεργοποιείται σίγουρα για άλλη μια φορά. Αυτό το ζήτημα αναφέρεται ως "Dying ReLU". Υπάρχουν ορισμένες εναλλακτικές λύσεις ReLU για την επίλυση τέτοιων ζητήματα. Στη συνέχεια αναλύονται ορισμένες από αυτές.

- **Leaky ReLU**: Αντί η ReLU να υποβαθμίζει τις αρνητικές εισόδους, αυτή η ενεργοποίηση διασφαλίζει ότι αυτές οι είσοδοι δεν αγνοούνται ποτέ. Χρησιμοποιείται για την επίλυση του Dying ReLU. Η Leaky ReLU μπορεί να αναπαρασταθεί μαθηματικά από την παρακάτω εξίσωση:

$$LeakyReLU(x)\;=\;max(0,x)\;+\;negative\_slope{*}min(0,x)$$

Σημειώστε ότι ο συντελεστής διαρροής συμβολίζεται με m. Συνήθως ορίζεται σε πολύ μικρή τιμή, όπως 0,001.

- **Noisy ReLU**: Αυτή η συνάρτηση χρησιμοποιεί μια κατανομή Gauss για να κάνει την ReLU θορυβώδη. Μπορεί να αναπαρασταθεί μαθηματικά από την παρακάτω εξίσωση:

  $$f(x)\text{NoisyReLU} = \max(x + Y), \text{ with } Y \sim N(0, \sigma(x))$$

- **Parametric Linear Units**: Αυτό είναι ως επί το πλείστον το ίδιο με το Leaky ReLU. Η κύρια διαφορά είναι ότι ο παράγοντας διαρροής σε αυτή τη συνάρτηση ενημερώνεται μέσω της διαδικασίας εκπαίδευσης του μοντέλου.

4. **Πλήρως συνδεδεμένο επίπεδο (Fully Connected Layer)**: Συνήθως, αυτό το στρώμα βρίσκεται στο τέλος κάθε αρχιτεκτονικής CNN. Στο εσωτερικό αυτού του στρώματος, κάθε νευρώνας συνδέεται με όλους τους νευρώνες του προηγούμενου στρώματος, η λεγόμενη προσέγγιση Fully Connected (FC). Χρησιμοποιείται ως ταξινομητής CNN. Ακολουθεί τη βασική μέθοδο του συμβατικού νευρωνικού δικτύου perceptron πολλαπλών στρωμάτων (conventional multiple-layer perceptron neural network), καθώς είναι ένας τύπος ANN με τροφοδότηση προς τα εμπρός (feed forward). Η είσοδος του στρώματος FC προέρχεται από το τελευταίο στρώμα συγκέντρωσης ή συνελικτικού στρώματος. Η είσοδος αυτή έχει τη μορφή διανύσματος, το οποίο δημιουργείται από τους χάρτες χαρακτηριστικών μετά την ισοπέδωση.

5. **Συναρτήσεις Απώλειας (Loss Functions):** Στην προηγούμενη ενότητα παρουσιάστηκαν διάφοροι τύποι επιπέδων της αρχιτεκτονικής του CNN. Επιπλέον, η τελική ταξινόμηση επιτυγχάνεται από το στρώμα εξόδου, το οποίο αποτελεί το

τελευταίο στρώμα της αρχιτεκτονικής CNN. Ορισμένες συναρτήσεις απωλειών χρησιμοποιούνται στο στρώμα εξόδου για τον υπολογισμό του προβλεπόμενου σφάλματος που δημιουργείται σε όλα τα δείγματα εκπαίδευσης στο μοντέλο CNN. Αυτό το σφάλμα αποκαλύπτει τη διαφορά μεταξύ της πραγματικής εξόδου και της προβλεπόμενης. Στη συνέχεια, θα βελτιστοποιηθεί μέσω της διαδικασίας εκμάθησης CNN. Ωστόσο, δύο παράμετροι χρησιμοποιούνται από τη συνάρτηση απώλειας για τον υπολογισμό του σφάλματος. Η εκτιμώμενη έξοδος του CNN (που αναφέρεται ως πρόβλεψη) είναι η πρώτη παράμετρος. Η πραγματική έξοδος (που αναφέρεται ως ετικέτα) είναι η δεύτερη παράμετρος. Διάφοροι τύποι απωλειών χρησιμοποιούνται σε διάφορους τύπους προβλημάτων. Τα ακόλουθα εξηγούν συνοπτικά ορισμένους από τους τύπους συναρτήσεων απώλειας

- **Cross-Entropy or Softmax Loss Function**: Η λειτουργία αυτή χρησιμοποιείται συνήθως για τη μέτρηση της απόδοσης του μοντέλου CNN. Αναφέρεται επίσης ως log συνάρτηση απώλειας. Η έξοδός της είναι η πιθανότητα p $\in$ {0 , 1}. Επιπλέον, χρησιμοποιείται συνήθως ως υποκατάστατο της συνάρτησης απώλειας τετραγωνικού σφάλματος σε προβλήματα ταξινόμησης πολλαπλών κλάσεων. Στο στρώμα εξόδου, χρησιμοποιεί τις ενεργοποιήσεις softmax για να παράγει την έξοδο εντός μιας κατανομής πιθανότητας.

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^{N} e_k^a}$$

Εδώ, το e^{a_i} αντιπροσωπεύει τη μη κανονικοποιημένη έξοδο από το προηγούμενο στρώμα, ενώ το N αντιπροσωπεύει τον αριθμό των νευρώνων στο στρώμα εξόδου.

2.3.5 Κανονικοποίηση στο CNN

Για τα μοντέλα CNN, η υπερπροσαρμογή (over-fitting) αποτελεί το κεντρικό ζήτημα που σχετίζεται με την απόκτηση καλής γενίκευσης. Το μοντέλο δικαιούται υπερπροσαρμογή στις περιπτώσεις που το μοντέλο εκτελείται ιδιαίτερα καλά σε δεδομένα εκπαίδευσης και δεν πετυχαίνει σε δεδομένα δοκιμής (αθέατα δεδομένα). Ένα υποπροσαρμοσμένο (under-fitted) μοντέλο είναι το αντίθετο και αυτό συμβαίνει όταν το μοντέλο δεν μαθαίνει επαρκή ποσότητα από τα δεδομένα εκπαίδευσης.Το μοντέλο αναφέρεται ως "μόλις προσαρμοσμένο" (just-fitted) εάν εκτελείται καλά τόσο σε εκπαιδευτικά όσο και σε δοκιμαστικά δεδομένα.

1. **Dropout**: Πρόκειται για μια ευρέως χρησιμοποιούμενη τεχνική γενίκευσης. Κατά τη διάρκεια κάθε εποχής εκπαίδευσης, οι νευρώνες απορρίπτονται τυχαία. Με αυτόν τον τρόπο, η δύναμη επιλογής χαρακτηριστικών κατανέμεται εξίσου σε ολόκληρη την ομάδα των νευρώνων, καθώς και αναγκάζεται το μοντέλο να μάθει διαφορετικά ανεξάρτητα χαρακτηριστικά. Κατά τη διάρκεια της διαδικασίας εκπαίδευσης, ο νευρώνας που απορρίπτεται δεν θα αποτελεί μέρος της οπισθοδιάδοσης (back-propagation) ή της εμπρόσθιας διάδοσης (forward-propagation). Αντίθετα, το δίκτυο πλήρους κλίμακας χρησιμοποιείται για την εκτέλεση πρόβλεψης κατά τη διαδικασία δοκιμής
2. **Drop-Weights:** Αυτή η μέθοδος είναι πολύ παρόμοια με την dropout. Σε κάθε εποχή εκπαίδευσης, οι συνδέσεις μεταξύ των νευρώνων (βάρη) εγκαταλείπονται αντί να εγκαταλείπονται οι νευρώνες, αυτή είναι η μόνη διαφορά μεταξύ της μεθόδου drop-weights και της μεθόδου dropout.
3. **Ενίσχυση δεδομένων (Data Augmentation)**: Ο ευκολότερος τρόπος για την αποφυγεί της υπερπροσαρμογής (over-fitting) είναι η εκπαίδευση του μοντέλου σε έναν σημαντικό όγκο δεδομένων. Για να επιτευχθεί αυτό, χρησιμοποιείται η επαύξηση δεδομένων. Χρησιμοποιούνται διάφορες τεχνικές για την τεχνητή επέκταση του μεγέθους του συνόλου δεδομένων εκπαίδευσης.

4. **Κανονικοποίηση παρτίδας (Batch Normalization)**: Αυτή η μέθοδος εξασφαλίζει την απόδοση των ενεργοποιήσεων εξόδου. Αυτή η απόδοση ακολουθεί μια μοναδιαία κατανομή Gauss. Η αφαίρεση του μέσου όρου και η διαίρεση με την τυπική απόκλιση θα κανονικοποιήσει την έξοδο σε κάθε επίπεδο. Ενώ είναι δυνατόν να θεωρηθεί ως εργασία προεπεξεργασίας σε κάθε στρώμα του δικτύου, είναι επίσης δυνατόν να διαφοροποιηθεί και να ενσωματωθεί σε άλλα δίκτυα. Επιπλέον, χρησιμοποιείται για τη μείωση της "εσωτερικής μετατόπισης συνδιακύμανσης" (internal covariance shift) των στρωμάτων ενεργοποίησης. Σε κάθε στρώμα, η μεταβολή της κατανομής ενεργοποίησης ορίζει την εσωτερική μετατόπιση συνδιακύμανσης. Αυτή η μετατόπιση γίνεται πολύ υψηλή λόγω της συνεχούς ενημέρωσης των βαρών μέσω της εκπαίδευσης, η οποία μπορεί να συμβεί εάν τα δείγματα των δεδομένων εκπαίδευσης συλλέγονται από πολλές ανόμοιες πηγές (για παράδειγμα, εικόνες ημέρας και νύχτας). Έτσι, το μοντέλο θα καταναλώσει επιπλέον χρόνο για τη σύγκλιση και, με τη σειρά του, θα αυξηθεί και ο χρόνος που απαιτείται για την εκπαίδευση. Για την επίλυση αυτού του ζητήματος, στην αρχιτεκτονική του CNN εφαρμόζεται ένα στρώμα που αντιπροσωπεύει τη λειτουργία της ομαλοποίησης παρτίδας.

2.3.6 Επιλογή βελτιστοποιητή (Optimizers)

Δύο σημαντικά ζητήματα περιλαμβάνονται στη διαδικασία μάθησης: το πρώτο ζήτημα είναι η επιλογή του αλγορίθμου μάθησης (βελτιστοποιητής), ενώ το δεύτερο ζήτημα είναι η χρήση πολλών βελτιώσεων (όπως AdaDelta, Adagrad και momentum) μαζί με τον αλγόριθμο μάθησης για τη βελτίωση της εξόδου. Οι συναρτήσεις απώλειας, οι οποίες βασίζονται σε πολυάριθμες μαθησιακές παραμέτρους (π.χ. biases, βάρη κ.λπ.) ή στην ελαχιστοποίηση του σφάλματος (απόκλιση μεταξύ πραγματικής και προβλεπόμενης εξόδου), αποτελούν τον βασικό σκοπό όλων των αλγορίθμων μάθησης με επίβλεψη. Οι τεχνικές μάθησης με βάση την κλίση για ένα δίκτυο CNN εμφανίζονται ως η συνήθης επιλογή. Οι παράμετροι του δικτύου θα πρέπει πάντα να ενημερώνονται σε όλες τις εποχές εκπαίδευσης, ενώ το δίκτυο θα πρέπει επίσης να αναζητήσει την τοπικά βελτιστοποιημένη απάντηση σε όλες τις εποχές εκπαίδευσης, προκειμένου να ελαχιστοποιήσει το σφάλμα. Ο ρυθμός μάθησης ορίζεται ως το μέγεθος βήματος της ενημέρωσης των παραμέτρων. Η εποχή

εκπαίδευσης αντιπροσωπεύει μια πλήρη επανάληψη της ενημέρωσης των παραμέτρων που περιλαμβάνει ολόκληρο το σύνολο δεδομένων εκπαίδευσης σε μία φορά. Σημειώνεται ότι πρέπει να επιλέγεται ο ρυθμός μάθησης με σύνεση, ώστε να μην επηρεάζει τη διαδικασία μάθησης.

Για να ελαχιστοποιήσουμε το σφάλμα εκπαίδευσης υπάρχει ένας αλγόριθμος που ενημερώνει επαναληπτικά τις παραμέτρους του δικτύου σε κάθε εποχή (Epoch) εκπαίδευσης. Ο αλγόριθμος αυτός ονομάζετε Gradient Descent ή Gradient-based learning algorithm. Πιο συγκεκριμένα, για να ενημερώσει σωστά τις παραμέτρους, πρέπει να υπολογίσει την κλίση (κλίση) της αντικειμενικής συνάρτησης εφαρμόζοντας μια παράγωγο πρώτης τάξης σε σχέση με τις παραμέτρους του δικτύου. Στη συνέχεια, η παράμετρος ενημερώνεται προς την αντίστροφη κατεύθυνση της κλίσης για τη μείωση του σφάλματος. Η διαδικασία ενημέρωσης των παραμέτρων πραγματοποιείται μέσω της οπισθοδιάδοσης του δικτύου, κατά την οποία η κλίση σε κάθε νευρώνα διαδίδεται προς τα πίσω σε όλους τους νευρώνες του προηγούμενου στρώματος. Υπάρχουν διάφορες εναλλακτικές λύσεις του αλγορίθμου οι οποίες περιλαμβάνουν τις ακόλουθες:

1. Batch Gradient Descent: Κατά την εκτέλεση αυτής της τεχνικής, οι παράμετροι του δικτύου ενημερώνονται μόνο μία φορά πίσω από την εξέταση όλων των συνόλων δεδομένων εκπαίδευσης μέσω του δικτύου. Σε μεγαλύτερο βάθος, υπολογίζει την κλίση ολόκληρου του συνόλου εκπαίδευσης και στη συνέχεια χρησιμοποιεί αυτή την κλίση για την ενημέρωση των παραμέτρων. Για ένα σύνολο δεδομένων μικρού μεγέθους, το μοντέλο συγκλίνει ταχύτερα και δημιουργεί μια εξαιρετικά σταθερή κλίση με τη χρήση BGD. Δεδομένου ότι οι παράμετροι αλλάζουν μόνο μία φορά για κάθε εποχή εκπαίδευσης, απαιτεί σημαντική ποσότητα πόρων.

2. Stochastic Gradient Descent: Οι παράμετροι ενημερώνονται σε κάθε δείγμα εκπαίδευσης σε αυτή την τεχνική. Προτιμάται η αυθαίρετη δειγματοληψία των δειγμάτων εκπαίδευσης σε κάθε εποχή πριν από την εκπαίδευση. Για ένα σύνολο δεδομένων εκπαίδευσης μεγάλου μεγέθους, αυτή η τεχνική είναι τόσο πιο αποδοτική στη μνήμη όσο και πολύ ταχύτερη από την BGD. Ωστόσο, επειδή ενημερώνεται

συχνά, κάνει εξαιρετικά θορυβώδη βήματα προς την κατεύθυνση της απάντησης, γεγονός που με τη σειρά του προκαλεί μεγάλη αστάθεια στη συμπεριφορά σύγκλισης.

3. Mini-batch Gradient Descent: Σε αυτή την προσέγγιση, τα δείγματα εκπαίδευσης χωρίζονται σε διάφορες μίνι-παρτίδες, στις οποίες κάθε μίνι-παρτίδα μπορεί να θεωρηθεί μια υπομεγέθης συλλογή δειγμάτων χωρίς επικάλυψη μεταξύ τους. Στη συνέχεια, η ενημέρωση των παραμέτρων πραγματοποιείται μετά τον υπολογισμό της κλίσης σε κάθε μίνι-παρτίδα. Το πλεονέκτημα αυτής της μεθόδου προκύπτει από τον συνδυασμό των πλεονεκτημάτων των τεχνικών BGD και SGD. Έτσι, έχει σταθερή σύγκλιση, μεγαλύτερη υπολογιστική απόδοση και επιπλέον αποτελεσματικότητα στη μνήμη.

4. Momentum: Για τα νευρωνικά δίκτυα, η τεχνική αυτή χρησιμοποιείται στην αντικειμενική συνάρτηση. Βελτιώνει τόσο την ακρίβεια όσο και την ταχύτητα εκπαίδευσης αθροίζοντας την υπολογισμένη κλίση στο προηγούμενο βήμα εκπαίδευσης, η οποία σταθμίζεται μέσω ενός συντελεστή (γνωστού ως momentum factor). Ωστόσο, ως εκ τούτου, απλώς κολλάει σε ένα τοπικό ελάχιστο αντί για ένα παγκόσμιο ελάχιστο. Αυτό αποτελεί το κύριο μειονέκτημα των αλγορίθμων μάθησης με βάση την κλίση. Η τιμή του συντελεστή ορμής διατηρείται εντός του εύρους 0 έως 1. Με τη σειρά του, το μέγεθος του βήματος της ενημέρωσης του βάρους αυξάνεται προς την κατεύθυνση του ελάχιστου δυνατού για την ελαχιστοποίηση του σφάλματος. Καθώς η τιμή του συντελεστή ορμής γίνεται πολύ χαμηλή, το μοντέλο χάνει την ικανότητά του να αποφεύγει το τοπικό γυμνό ελάχιστο. Αντίθετα, καθώς η τιμή του παράγοντα ορμής γίνεται υψηλή, το μοντέλο αναπτύσσει την ικανότητα να συγκλίνει πολύ πιο γρήγορα. Εάν μια υψηλή τιμή του συντελεστή ορμής χρησιμοποιείται μαζί με το LR, τότε το μοντέλο θα μπορούσε να χάσει το τοπικό ελάχιστο περνώντας πάνω από αυτό. Ωστόσο, όταν η κλίση μεταβάλλει συνεχώς την κατεύθυνσή της καθ' όλη τη διάρκεια της διαδικασίας εκπαίδευσης, τότε η κατάλληλη τιμή του παράγοντα

ορμής (που είναι μια υπερ-παράμετρος) προκαλεί εξομάλυνση των μεταβολών της ενημέρωσης του βάρους.

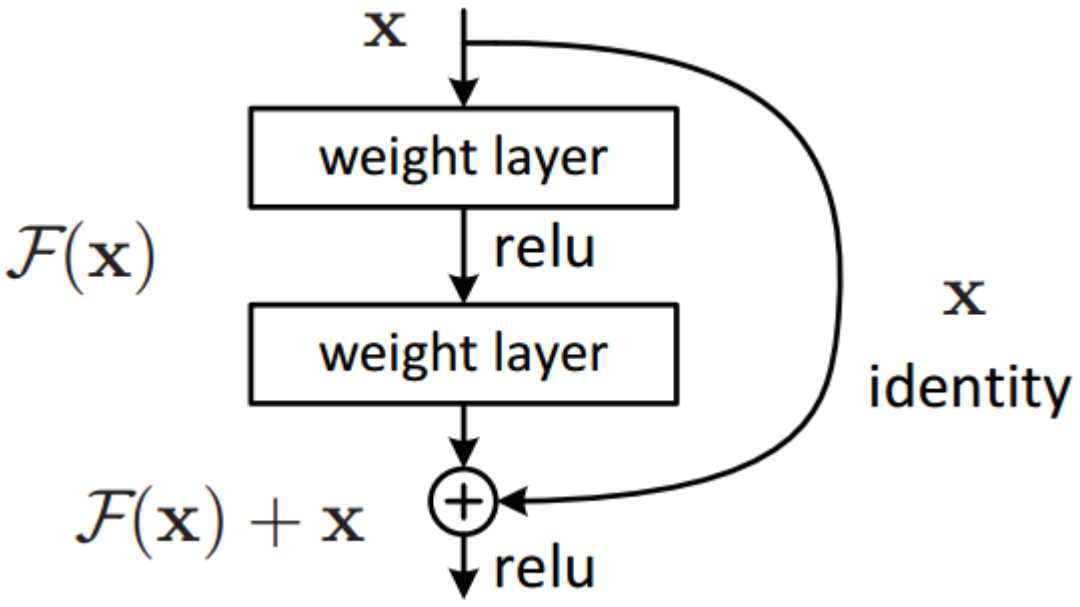
5. Adaptive Moment Estimation (Adam): Ο αλγόριθμος Adam αντιπροσωπεύει τις τελευταίες τάσεις στη βελτιστοποίηση της βαθιάς μάθησης. Αυτό παριστάνεται από τον πίνακα Hessian, ο οποίος χρησιμοποιεί μια παράγωγο δεύτερης τάξης. Ο Adam είναι μια στρατηγική μάθησης που έχει σχεδιαστεί ειδικά για την εκπαίδευση βαθιών νευρωνικών δικτύων. Η αποδοτικότερη μνήμη και η μικρότερη υπολογιστική ισχύς είναι δύο πλεονεκτήματα του Adam. Ο μηχανισμός του Adam είναι ο υπολογισμός για κάθε παράμετρο του μοντέλου. Ενσωματώνει τα πλεονεκτήματα τόσο του Momentum όσο και του RMSprop. Χρησιμοποιεί τις τετραγωνικές κλίσεις για την κλιμάκωση του ρυθμού μάθησης όπως το RMSprop και είναι παρόμοιο με το momentum χρησιμοποιώντας τον κινητό μέσο όρο της κλίσης.

2.4 DenseNet

Η διερεύνηση των αρχιτεκτονικών των δικτύων αποτελεί μέρος της έρευνας των νευρωνικών δικτύων από την αρχική τους ανακάλυψη. Ο αυξανόμενος αριθμός επιπέδων στα σύγχρονα δίκτυα ενισχύει τις διαφορές μεταξύ των αρχιτεκτονικών και παρακινεί τη διερεύνηση διαφορετικών μοτίβων συνδεσιμότητας και την επανεξέταση παλαιών ερευνητικών ιδεών. Τα Highway Networks ήταν από τις πρώτες αρχιτεκτονικές που παρέχουν ένα μέσο για την αποτελεσματική εκπαίδευση δικτύων από άκρο σε άκρο με περισσότερα από 100 επίπεδα. Χρησιμοποιώντας μονοπάτια παράκαμψης (bypassing paths) τα Highway Networks με εκατοντάδες στρώματα μπορούν να βελτιστοποιηθούν χωρίς δυσκολία. Τα μονοπάτια παράκαμψης εκτιμάται ότι είναι ο βασικός παράγοντας που διευκολύνει την εκπαίδευση αυτών των πολύ βαθιών δικτύων. Αυτό το σημείο υποστηρίζεται περαιτέρω από τα ResNets, στα οποία χρησιμοποιούν καθαρές αντιστοιχίσεις ταυτότητας ως μονοπάτια παράκαμψης. Τα ResNets έχουν επιτύχει εντυπωσιακές επιδόσεις που σπάνε

ρεκόρ σε πολλές δύσκολες εργασίες αναγνώρισης, εντοπισμού και ανίχνευσης εικόνων, όπως το ImageNet και η ανίχνευση αντικειμένων COCO.

Μια προσέγγιση για να γίνουν τα δίκτυα βαθύτερα (π.χ. με τη βοήθεια των συνδέσεων παράλειψης) είναι η αύξηση του πλάτους του δικτύου. Το GoogLeNet χρησιμοποιεί ένα "Inception module" το οποίο συνενώνει χάρτες χαρακτηριστικών (feature-maps) που παράγονται από φίλτρα διαφορετικού μεγέθους. Στο [1], προτάθηκε μια παραλλαγή των ResNets με πλατιά γενικευμένα υπολειμματικά μπλοκ. Τα υπολειμματικά μπλοκ είναι δομές σε νευρωνικά δίκτυα που επιτρέπουν την παράκαμψη (skip connection) της αρχικής είσοδου του μπλοκ στην έξοδο. Αυτό διευκολύνει την εκμάθηση βαθύτερων δικτύων, επιτρέποντας στο δίκτυο να μαθαίνει την διαφορά (υπόλειμμα) μεταξύ της επιθυμητής εξόδου και της αρχικής εισόδου, βελτιώνοντας την απόδοση και σταθερότητα της εκπαίδευσης. Τα υπολειμματικά μπλοκ αναφέρονται σε μια αρχιτεκτονική δικτύου που εισήγαγε η οικογένεια των ResNets (Δίκτυα Υπολειμμάτων). Στα υπολειμματικά μπλοκ, η είσοδος στο μπλοκ συνδυάζεται με την έξοδο του μπλοκ, καθιστώντας δυνατή την εκμάθηση της διαφοράς (υπολείμματος) μεταξύ των δύο. Συγκεκριμένα, τα υπολειμματικά μπλοκ προσθέτουν την είσοδο του μπλοκ στην έξοδο, ενώ παράλληλα εφαρμόζουν μια συνάρτηση ενεργοποίησης, όπως το ReLU (Rectified Linear Unit), στο σύνολο.



Εικόνα. 4. Παράδειγμα residual Block απο: Residual blocks — Building blocks of ResNet | by Sabyasachi Sahoo | Towards Data Science

Η προσθήκη του υπολείμματος επιτρέπει την εκπαίδευση του δικτύου να επικεντρώνεται στην μάθηση της διαφοράς μεταξύ της προσδοκώμενης εξόδου και της πραγματικής, κάνοντας την εκπαίδευση πιο αποτελεσματική και βοηθώντας στην αντιμετώπιση προβλημάτων όπως η εξαφάνιση των κλιμάκων. Αυτή η δομή βοηθά στη δημιουργία βαθύτερων δικτύων χωρίς τα προβλήματα που συνήθως σχετίζονται με την εκπαίδευση βαθιών μοντέλων, όπως η δυσκολία της εκμάθησης επιπλέον χαρακτηριστικών. Στην πραγματικότητα, η απλή αύξηση του αριθμού των φίλτρων σε κάθε στρώμα των ResNets μπορεί να βελτιώσει την απόδοσή του, υπό την προϋπόθεση ότι το βάθος είναι επαρκές.

Τα DenseNets εκμεταλλεύονται τις δυνατότητες του δικτύου μέσω της επαναχρησιμοποίησης χαρακτηριστικών, δημιουργώντας συμπυκνωμένα μοντέλα που είναι εύκολο να εκπαιδευτούν και εξαιρετικά αποδοτικά ως προς τις παραμέτρους. Στο πλαίσιο των νευρωνικών δικτύων, ο όρος "χαρτογράφηση χαρακτηριστικών" (feature map) αναφέρεται στις εξόδους των φίλτρων ή νευρώνων από ένα συγκεκριμένο στρώμα. Κάθε χάρτης χαρακτηριστικών αντιπροσωπεύει διάφορα χαρακτηριστικά της εισόδου, τα οποία ενεργοποιούνται από τα αντίστοιχα φίλτρα. Η συνένωση χαρτογράφησης χαρακτηριστικών (feature map) που μαθαίνονται από διαφορετικά στρώματα αυξάνει τη διακύμανση στην είσοδο των επόμενων στρωμάτων και βελτιώνει την αποδοτικότητα. Αυτό αποτελεί μια σημαντική διαφορά μεταξύ των DenseNets και των ResNets. Σε σύγκριση με τα δίκτυα Inception, τα οποία επίσης συνενώνουν χαρακτηριστικά από διαφορετικά στρώματα, τα DenseNets είναι απλούστερα και αποδοτικότερα.

Θεωρούμε μια απλή εικόνα x_0 που περνάει μέσα από ένα συνελικτικό δίκτυο. Το δίκτυο αποτελείται από L επίπεδα, καθένα από τα οποία από τα οποία υλοποιεί έναν μη γραμμικό μετασχηματισμό $H_l(\cdot)$, όπου l δείχνει το επίπεδο. Το $H_l(\cdot)$, μπορεί να είναι μια σύνθετη συνάρτηση λειτουργιών όπως η ομαδική κανονικοποίηση (Batch Normalization), οι διορθωμένες γραμμικές μονάδες (ReLU) , η ομαδοποίηση (Pooling) ή η συνέλιξη (Conv). Συμβολίζουμε την έξοδο του l^{th} στρώματος ως $x_{l.}$. Το παραδοσιακό συνελικτικό δίκτυο

τροφοδότησης-προώθησης (feed forward) συνδέει την έξοδο του l^{th} στρώματος ως είσοδο στο $(l + 1)^{th}$ στρώμα , γεγονός που οδηγεί στην ακόλουθη μετάβαση στρώματος: $x_l = H_l(x_{l-1})$. Τα ResNets προσθέτουν μια σύνδεση παράκαμψης που παρακάμπτει τους μη γραμμικούς μετασχηματισμούς με μια συνάρτηση ταυτότητας:

$$x_l = H_l(x_{l-1}) + x_{l-1}.$$

Ένα πλεονέκτημα των ResNets είναι ότι η κλίση μπορεί να ρέει απευθείας μέσω της ταυτοτικής συνάρτησης από τα μεταγενέστερα στρώματα στα πρώτα στρώματα. Ωστόσο, η συνάρτηση ταυτότητας και η έξοδος του H_l συνδυάζονται με άθροιση, γεγονός που μπορεί να εμποδίσει την ροή πληροφοριών στο δίκτυο.

Dense connectivity: Για να βελτιώσουμε περαιτέρω τη ροή πληροφοριών μεταξύ των στρωμάτων προτείνουμε ένα διαφορετικό μοτίβο συνδεσιμότητας: εισάγουμε απευθείας συνδέσεις από κάθε στρώμα προς όλα τα επόμενα στρώματα. Κατά συνέπεια, το l^{th} επίπεδο λαμβάνει τους χαρακτηριστικούς χάρτες όλων των προηγούμενων επιπέδων, $x_0, \dots, x_{l-1}$, ως είσοδο $x_l = H_l([x_0, x_1, \dots, x_{l-1}])$, όπου $[x_0, x_1, \dots, x_{l-1}]$ αναφέρεται στη συνένωση των χαρτών χαρακτηριστικών που παράγονται στα επίπεδα $0, \dots, l-1$. Λόγω της πυκνής συνδεσιμότητας αναφερόμαστε σε αυτή την αρχιτεκτονική δικτύου ως πυκνό συνεπτυγμένο δίκτυο (DenseNet). Για ευκολία στην υλοποίηση, συνδέουμε τις πολλαπλές εισόδους του $H_l(\cdot)$ στην παραπάνω εξίσωση σε έναν ενιαίο τένσορα (tensor).

Composite Function: Ορίζουμε το $H_l(\cdot)$ ως σύνθετη συνάρτηση τριών διαδοχικών πράξεων. Ομαδοποιημένη κανονικοποίηση (batch normalization), ακολουθούμενη από μια διορθωμένη γραμμική μονάδα (ReLU) και μια συνέλιξη 3 × 3 (Conv).

Στρώματα υποδειγματοληψίας (Pooling Layers): Η εξίσωση

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}])$$

δεν είναι βιώσιμη όταν το μέγεθος των χαρτών χαρακτηριστικών αλλάζει. Ωστόσο, ένα ουσιαστικό μέρος των συνελικτικών δικτύων είναι τα στρώματα δειγματοληψίας που

αλλάζουν το μέγεθος των χαρτών χαρακτηριστικών. Για να διευκολύνουμε τη δειγματοληψία προς τα κάτω στην αρχιτεκτονική μας διαιρούμε το δίκτυο σε πολλαπλά πυκνά συνδεδεμένα μπλοκ, Αναφερόμαστε στα στρώματα μεταξύ των μπλοκ ως μεταβατικά στρώματα, τα οποία κάνουν συνέλιξη και συγκέντρωση (densely connected dense blocks).

Ρυθμός Ανάπτυξης (Growth Rate): Εάν κάθε συνάρτηση H_l παράγει k χαρακτηριστικές απεικονίσεις, προκύπτει ότι το l^{th} επίπεδο έχει $k_0 + k * (l - 1)$ χαρακτηριστικές χαρτογραφήσεις εισόδου, όπου k_0 είναι ο αριθμός των καναλιών στο επίπεδο εισόδου. Μια σημαντική διαφορά μεταξύ του DenseNet και των υφιστάμενων αρχιτεκτονικών δικτύων είναι ότι το DenseNet μπορεί να έχει πολύ στενά στρώματα, π.χ. k = 12. Αναφερόμαστε στην υπερπαράμετρο k ως ρυθμό ανάπτυξης του δικτύου. Μια εξήγηση γι' αυτό είναι ότι κάθε στρώμα έχει πρόσβαση σε όλους τους προηγούμενους χάρτες χαρακτηριστικών στο μπλοκ του και, επομένως, στη "συλλογική γνώση" του δικτύου. Μπορεί κανείς να θεωρήσει τα feature-maps ως τη συνολική κατάσταση του δικτύου. Κάθε επίπεδο προσθέτει k δικά του feature-maps σε αυτή την κατάσταση. Ο ρυθμός ανάπτυξης ρυθμίζει πόση νέα πληροφορία συνεισφέρει κάθε στρώμα στην συνολική κατάσταση. Η συνολική κατάσταση, αφού γραφτεί, μπορεί να προσπελαστεί από παντού μέσα στο δίκτυο και, σε αντίθεση με τις παραδοσιακές αρχιτεκτονικές δικτύων, δεν υπάρχει ανάγκη να αναπαραχθεί από στρώμα σε στρώμα.

Bottleneck layers: Το πρόσθετο στρώμα συμφόρησης (bottleneck layer) είναι ένα κοινό στοιχείο που συχνά χρησιμοποιείται σε συνελικτικά νευρωνικά δίκτυα (Convolutional Neural Networks - CNNs), και συγκεκριμένα στην αρχιτεκτονική DenseNet, για να μειώσει τον αριθμό των χαρακτηριστικών χαρτών εισόδου προτού περάσουν από ένα στρώμα συνέλιξης 3x3. Αυτή η προσέγγιση βοηθά στην ελαχιστοποίηση της υπολογιστικής πολυπλοκότητας και της κατανάλωσης μνήμης, ενώ διατηρεί την απόδοση του δικτύου. Αν και κάθε στρώμα παράγει μόνο k χάρτες χαρακτηριστικών εξόδου, συνήθως έχει πολύ περισσότερες εισόδους. Είναι έχει σημειωθεί στο ότι μια συνέλιξη 1×1 μπορεί να εισαχθεί ως

στρώμα συμφόρησης(bottleneck) πριν από κάθε συνέλιξη 3×3 για να μειωθεί ο αριθμός των χαρακτηριστικών χαρτών εισόδου και, συνεπώς, να βελτιωθεί η υπολογιστική απόδοση.

Συμπίεση (Compression): Για περαιτέρω βελτίωση της συμπίεσης του μοντέλου, μπορούμε να μειώσουμε τον αριθμό των χαρακτηριστικών χαρτών στα transition layers. Εάν ένα πυκνό μπλοκ περιέχει m feature-maps, αφήνουμε το επόμενο στρώμα μετάβασης να παράγει θ_m χαρακτηριστικών χαρτών εξόδου, όπου $0 < \theta \leq 1$ αναφέρεται ως συντελεστής συμπίεσης. Όταν $\theta = 1$, ο αριθμός των feature-maps στα στρώματα μετάβασης παραμένει αμετάβλητος. Αναφερόμαστε στο DenseNet με $\theta < 1$ ως DenseNet-C. Όταν τόσο το στρώμα συμφόρησης όσο και το στρώμα μετάβασης με $\theta < 1$ χρησιμοποιούνται, αναφερόμαστε στο μοντέλο μας ως DenseNet-BC.

2.5 Εφαρμογή του DenseNet121

Στο σύνολο δεδομένων στην εργασία αυτή χρησημοποιήσα το DenseNet121 όπου έχει 4 μπλοκ που το καθένα έχει ίσο αριθμό επιπέδων (layers) με μια συνέλιξη (Convolution) 7x7, πενήντα οκτώ (58) συνελίξεις 3x3, εξήντα ένα (61) 1x1 συνελίξεις, τέσσερα (4) Avarage pooling (Μέσος όρος συγκέντρωσης) και ένα πλήρως συνδεδεμένο επίπεδο (Fully connected layer). Εν ολίγοις, το DenseNet-121 έχει 121 συνελίξεις (Convolutions) και 4 AvgPool. Όλα τα στρώματα, δηλαδή αυτά που βρίσκονται μέσα στο ίδιο πυκνό μπλοκ και τα μεταβατικά στρώματα, κατανέμουν τα βάρη τους σε πολλαπλές εισόδους, γεγονός που επιτρέπει στα βαθύτερα στρώματα να χρησιμοποιούν χαρακτηριστικά που έχουν εξαχθεί νωρίς. Δεδομένου ότι τα στρώματα μετάβασης εξάγουν πολλά περιττά χαρακτηριστικά, τα στρώματα στο δεύτερο και τρίτο πυκνό μπλοκ αναθέτουν τα λιγότερα βάρη στην έξοδο των στρωμάτων μετάβασης. Επίσης, παρόλο που τα βάρη ολόκληρου του πυκνού μπλοκ χρησιμοποιούνται από τα τελικά στρώματα, ενδέχεται να υπάρχουν περισσότερα χαρακτηριστικά υψηλού επιπέδου που παράγονται βαθύτερα στο μοντέλο, καθώς φάνηκε να υπάρχει μεγαλύτερη συγκέντρωση προς τους τελικούς χάρτες χαρακτηριστικών στα πειράματα.

3 Πειράματα και Υλοποίηση

3.1 Στόχος Πειραμάτων

Για το **FSD Kaggle 2018** dataset που στόχος είναι η κατασκευή ενός μοντέλου νευρωνικού δικτύου που να μπορεί να αναγνωρίσει έναν αριθμό ηχητικών γεγονότων διαφορετικής φύσης, συμπεριλαμβανομένων μουσικών οργάνων, ανθρώπινων ήχων, οικιακών ήχων, ζώων κ.λπ. Στόχος μου είναι να εστιάσω κυρίως στο πώς να εκπαιδεύσω αποτελεσματικά ένα μοντέλο βαθιάς μάθησης (deep learning) σε ανόμοιους ήχους σε διάρκεια αλλά και με θορυβο. Πριν καταλήξω στο πώς θα γίνει η εκμάθηση του μοντέλου δοκίμασα διάφορες τεχνικές και προσεγγίσεις του θέματος όπως να βρώ ένα νευρωνικό δίκτυο που να δέχεται κατευθείαν τον ήχο και να μην χρειάζεται να τον κάνω STFT αλλά παρατήρησα οτι η διαδικασία δεν είναι τόσο απλή. Αυτό που έπρεπε να γίνει ήταν να φορτώσω τα δεδομένα ήχου από το αρχείο και να τα επεξεργαστώ ώστε να είναι σε μορφή που αναμένει το μοντέλο.

3.1.1 DenseNet 121 (Πρώτο Train)

Αφου έκανα τους ήχους από το dataset σε μορφή STFT (Short Time Fourier Transform) το μοντέλο νευρωνικού δικτύου που επέλεξα για να προχωρήσω στη διαδικασία του train ήταν το DenseNet 121. Οι επιλογές που είχα ήταν να χρησιμοποιήσω το μοντέλο έτοιμο απο την βιβλιοθήκη του PyTorch ή να το φτιάξω εγώ (from scratch) σε pytorch. Επέλεξα να το γράψω από την αρχή ώστε να μπορέσω να έχω τον έλεγχο στις παραμέτρους του μοντέλου. Πριν ξεκινήσει το train αυτο που έκανα για να μπορώ να υπολογίσω το ποσοστό ακρίβειας ήταν να πάρω το 20% απο τους ήχους και να μην τους δώσω στο μοντέλο ώστε μετά από κάθε EPOCH να κάνει inference σε αυτό το 20% των ήχων με αποτέλεσμα να μπορώ να υπολογίσω το ποσοστό ακρίβειας που πετυχαίνει το DenseNet 121.

Κάνοντας train το DenseNet 121 για 30 epochs (δηλαδή 30 φορές το μοντέλο πέρασε ολόκληρο το dataset) το ποσοστό επιτυχίας ηταν 72% δηλαδή στα 1895 stft (short time fourier transform) που του δόθηκαν βρήκε σωστά τα 1364.

Model	**Epochs**	**Accuracy (%)**
DenseNet 121	30 it/Dataset	72%

3.1.2 DenseNet 121 (Δεύτερο Train)

Στο πρώτο train του μοντέλου φτάσαμε το ποσοστό ακρίβειας στα 72%. Δεν είναι πάρα πολύ μεγάλο σε σχέση με αυτά που μπορεί να μας δώσει η αρχιτεκτονική του DenseNet. Χωρίς να πειράξω τις παραμέτρους του μοντέλου ώστε να πετύχω μεγαλύτερη ακρίβεια ήθελα να πειράξω πρώτα τους ήχους που είχα από το dataset. Αναλύοντας το dataset παρατηρούμε ότι οι ήχοι είναι διαφορετική σε διάρκεια μεταξύ τους και αυτό δημιουργεί ένα πρόβλημα όταν τους μετατρέπω σε STFT (short time Fourier transform) γιατί το νευρωνικό δίκτυο παίρνει σαν είσοδο μία εικόνα ανάλυσης 256 x 256 το οποίο αντιστοιχεί σε περίπου 10 δευτερόλεπτα.Η τελική ανάλυση είναι τόσο μικρή γιατι για να του δώσω εικόνα μεγαλύτερης ανάλυσης χρειαζόμουνα μεγαλύτερη κάρτα γραφικών με μεγαλύτερη Video Ram ενώ

ταυτόχρονα το training θα διαρκούσε μεγαλύτερο χρόνο από ότι έκανε τώρα στα 30 Epoch και επίσης δεν σημαινει οτι η μεγαλύτερη ανάλυση θα προσφέρει καλύτερη ακρίβεια στο τέλος του train. Έτσι αυτό που έκανα ήταν να βρω τον μέσο όρο διάρκειας των ήχων στο dataset το οποίο ήταν περίπου 10 δευτερόλεπτα και να φέρω τις ηχητικές μονάδες όσο πιο κοντά γίνεται σε αυτό το χρόνο με αποτέλεσμα το νευρωνικό δίκτυο να παίρνει όμοια πράγματα σαν είσοδο και να συγκρίνει. Αρχικά αυτό που θα έπρεπε να σκεφτούμε είναι τι χαρακτηρίζει έναν ήχο. Δηλαδή αν ακούσω έναν ήχο λεωφορείου στη μισή διάρκεια ενδεχομένως να μην καταλάβω αν είναι λεωφορείο ή αν ακούσω έναν ήχο από ένα ταμπούρο για ένα δευτερόλεπτο και το προσθέσω στο τέλος του ήχου για να φτάσει τα 10 δευτερόλεπτα θα έχω ένα πολύ διαφορετικό αποτέλεσμα από το αρχικό και ενδεχομένως να μην ακούγεται και στο ανθρώπινο αυτί σαν ταμπούρο.

Επειδή το dataset είναι μεγάλο θα πρέπει η διαδικασία αυτή να γίνει αυτόματα με τη χρήση της python. Χρησιμοποιώντας τη βιβλιοθήκη numpy μετέτρεψα τα φάσματα σε πίνακα 2 διαστάσεων όπου η μία διάσταση είναι η ένταση του ήχου (amplitude) και η άλλη διάσταση είναι οι συχνότητες (frequency). Γνωρίζοντας το sampling rate το οποίο μας δίνετε στις προδιαγραφές του dataset μπορούμε να υπολογίσουμε τον χρόνο κάθε ήχου. Στο παρακάτω παράδειγμα κώδικα το sampling rate το παίρνουμε κατευθείαν από το αρχείο του ήχου όπως και τα samples και τέλος για να βρούμε τον χρόνο διαιρούμε τα samples με το sample rate.

```
import soundfile as sf
f = sf.SoundFile(audio_file_1.wav')
print('samples = {}'.format(len(f)))
print('sample rate = {}'.format(f.samplerate))
print('seconds = {}'.format(len(f) / f.samplerate))
```

Αφού το κάνουμε αυτό για όλους τους ήχους υπολογίζουμε τον μέσο όρο διαρκείας και επιλέγουμε τα 10 δευτερόλεπτα στα οποία θα πρέπει να φέρουμε όλους τους ήχους. Όπως είπαμε παραπάνω τους ήχους αυτή τη στιγμή τους έχουμε στη μορφή ενός πίνακα δύο διαστάσεων, τους μικρότερους ήχους λοιπόν αυτό που κάνουμε είναι να πάρουμε όλα τα στοιχεία του πίνακα και να τα προσθέσουμε ξανά στο τέλος του ώστε να έχουμε διάρκεια μεγαλύτερη από τον αρχικό ήχο και επαναλαμβάνουμε μεχρι να φτάσει κοντά στα 10s. Για τους μεγάλους ήχους η διαδικασία είναι περίπου ίδια δηλαδή σπάμε τον πίνακα στη μέση και

συνεχίζουμε να τον διαιρούμε μέχρι να γίνει περίπου 10s και κρατάμε τα υπόλοιπα μέρη του που έχουμε σπάσει σαν extra ηχητικές μονάδες τις οποίες θα τις δώσουμε και αυτές για train. Τέλος με τις παραπάνω ενέργειες καταλήγουμε σε ένα data set καινούργιο με περισσότερα αρχεία ήχου και λίγο πιο ξεκάθαρο για το νευρωνικό δίκτυο. Στη συνέχεια περνάω στη διαδικασία του train όπως έκανα ακριβώς και στο προηγούμενο πείραμα με την ίδια ανάλυση φάσματος 256 x 256 και 30 Epochs.

Παρατηρούμε ότι δεν έχει αλλάξει τίποτα στην ακρίβεια του μοντέλου ενώ κανονικά θα έπρεπε να έχει ανέβει το ποσοστό. Τα προβλήματα που έχουν δημιουργηθεί είναι δύο. Πρώτον όπως ανέφερα και παραπάνω οι μικρότερες ηχιτικές μονάδες έχουν αλλάξει δηλαδή είναι διαφορετικός ήχος σε σχέση με τον αρχικό επειδή προσθέσαμε τον ίδιο ήχο μέχρι να γίνει 10 δευτερόλεπτα και αντίστοιχα οι μεγαλύτερες ηχιτικές μονάδες δεν αρκούν τα δευτερόλεπτα για να μπορέσει το νευρωνικό δίκτυο να πάρει τα στοιχεία που θέλει για να συγκρίνει να κρατήσει και να μάθει. Επίσης ανατρέχοντας πίσω στο αρχικό dataset παρατηρούμε ότι οι 41 κλάσεις που έχουμε για τους ήχους δεν έχουν όμοια κατανομή ηχητικών αρχείων δηλαδή οι ηχητικές μονάδες που έχουμε για το γαύγισμα σκύλων έχουν λιγότερους ήχους από τις ηχητικές μονάδες που έχουμε για τα πυροτεχνήματα και αυτό συμβαίνει σχεδόν για όλες τις κλάσεις που υπάρχουν. Το πρόβλημα που δημιουργείται από αυτό λέγεται imbalance και έχει ως αποτέλεσμα το νευρωνικό δίκτυο να μην μπορεί να δώσει το ίδιο βάρος σε όλες τις κλάσεις δηλαδή σε αυτές που έχουν λιγότερα δείγματα δίνει λιγότερο βάρος και σε αυτές που έχουν περισσότερα δίνει περισσότερο με αποτέλεσμα αφού τελειώσει το train και κάνουμε classify, παρατηρούμε οτι τα δείγματα με το λιγότερο βάρος δηλαδή οι κλάσεις οι οποίες είχαν τα λιγότερα samples δεν θα μπορέσει ο ταξινομητής να τις βρει με ευκολία με αποτέλεσμα το ποσοστό ακρίβειας να μείνει στο 72% και να μη δούμε κάποια άνοδο.

Model	**Epochs**	**Accuracy (%)**
DenseNet 121	30 it/Dataset	72%

3.1.3 DenseNet 121 (Τρίτο Train)

Στο 3ο train η προσέγγιση είναι όμοια με του δεύτερου πειράματος με διαφορά ότι αντί για 10 δευτερόλεπτα, προσέγγισα το dataset με την ίδια διαδικασία στα πέντε δευτερόλεπτα. Αυτή τη φορά όμως δεν επεξεργάστηκα τους ήχους αλλά το STFT. Δηλαδή φόρτωσα την εικόνα κάθε δείγματος στην python και την ανέλυσα σαν έναν πίνακα 3 διαστάσεων με τις διαστάσεις να είναι ως εξείς:

1) Η πρώτη διάσταση είναι τα pixel στον άξονα X
2) Η δεύτερη διάσταση είναι τα pixel στον άξονα Y
3) Η τρίτη διάσταση είναι ο αριθμός των καναλιών της εικόνας (RGB) στον άξονα Z

Με την βοήθεια της python στον άξονα X μπορούμε να υπολογίσουμε το χρόνο από το STFT. Αν ο ήχος ήταν μεγαλύτερος από 5 δευτερόλεπτα τότε αυτό που έκανα ήταν να πάρω στην τύχη κάποιες χρονικές στιγμές όπου υπάρχει ήχος οι οποίες να είναι πέντε δευτερόλεπτα και αν ο ήχος ήταν μικρότερος τότε έκανα zero padding στην αρχή και στο τέλος του ήχου δηλαδή πρόσθετα στην εικόνα μηδενικά που μεταφράζονται σε μαύρο χρώμα το οποίο σημαίνει οτι δεν υπάρχει ήχος και αντίστοιχα στο τέλος ώστε να γίνει 5 δευτερόλεπτα.

Το επόμενο πρόβλημα που έπρεπε να λύσω ήταν οτι το dataset δεν είναι όμοια κατανεμημένο σε όλες τις κλάσεις.Η μέθοδος που εφάρμοσα για να επιλύσω το πρόβλημα αυτό ήταν η υπερδειγματοληψία (oversampling) η οποία προσπαθεί να εξισορροπήσει το dataset αυξάνοντας το σύνολο των λιγότερων δειγμάτων.Αυτό θα μας δώσει ακρίβεια αλλά μπορεί να δημιουργήσει και overfit στο μοντέλο και όταν του δώσουμε στο τέλος δείγματα τα οποία δεν τα έχει δει ποτέ ξανά να μην μπορέσει να βρει τι είναι.

Κάνοντας train για 30 Epohs το μοντέλο καταφέρνει και πιάνει 88.9% ακρίβεια.Η βελτίωση είναι αρκετά μεγάλη και το μόνο που έκανα ήταν να φτιάξω λίγο καλύτερα τα δείγματα ως προς τον τρόπο που θα τα δει το νευρωνικό δίκτυο και να ισορροπήσω λίγο τις κλάσεις ώστε να δώσει το νευρωνικό δίκτυο μεγαλύτερο βάρος σε αυτές με τα λιγότερα δείγματα.

Model	Epochs	Accuracy (%)
DenseNet 121	30 it/Dataset	88.9%

4 ΣΥΜΠΕΡΑΣΜΑΤΑ

Για το Task2 του DCASE2018 το DenseNet 121 κατάφερε να ξεπεράσει τη βάση στις βαθμολογίες με ποσοστό ακρίβειας 88.8% (late submission).Στο σύνολο οι πειραματικές διαδικασίες ήταν περισσότερες από αυτές που έχουν αναφερθεί στην εργασία απλά οι αλλαγές και οι διαφοροποιήσεις από πείραμα σε πείραμα δεν ήταν μεγάλες. Το dataset πέρασε από πολλές δοκιμές και αλλαγές μέχρι να καταλήξω σε STFT. Κάποιες από αυτές ήταν να κρατήσω το αρχικό SR (sampling rate) που ήταν 44.1KHz και να δώσω τους ήχους σε ένα 1D CNN αλλά τα αποτελέσματα ήταν πάρα πολύ χαμηλά. Μια άλλη προσέγγιση ήταν τα mel-spectrograms αντι του STFT αλλά δεν υπήρξε κάποια άνοδο στο ποσοστο ακρίβειας.

Ο τρόπος που έκανα inference για να υπολογίσω το ποσοστό ακρίβειας δεν ήταν απόλυτα σωστός. Δηλαδή αφού τελειώνει η διαδικασία του train και δίνω στο μοντέλο το 20% απο το train set για να δω το σκορ του θα πρέπει να επαναλαβω τη διαδικασία της προεπεξεργασίας. Στους μικρούς ήχους θα πρέπει να κάνω zero pad και στους μεγάλους θα πρέπει να βρω την ετικέτα που προέβλεψε το μοντέλο για κάθε τμήμα των 5s και να στο τέλος να κρατήσω αυτή με το μεγαλύτερο ποσοστό ακρίβειας.

Για αυτό το Task το DenseNet θα μπορούσε να πετύχει και μεγαλύτερο σκορ απο το 88.8% με κάποιες αλλαγές όπως τα του δώσω λίγο μεγαλύτερη ανάλυση στην εικόνα ως προς τον άξονα X που είναι ο χρόνος. Σε ένα δοκιμαστικό train έδωσα τα δείγματα με ανάλυση 256x627 που αντιστοιχεί περίπου 5.013s. Το ποσοστό ακρίβειας παρεμεινε στο 88.8% αλλά αυτή τη φόρα χρειάστηκαν 17 EPOCHS. Εδώ παρατήρησα το λάθος μου, στις πειραματικές διαδικασίες παραπάνω τα STFT τα έκανα 5s και οι εικόνες έχουν ανάλυση 256x627 αλλά εγώ τις έδινα στο μοντέλο με ανάλυση 256x256 που σημαίνει οτι έχανα στον άξονα x περίπου 2.7s με αποτέλεσμα το μοντέλο να μην παίρνει ποτέ ολόκληρο το ηχητικό δείγμα.

Σύμφωνα με τα αποτελέσματα στο τέλος το ποσοστό είναι το ίδιο με τη διαφορα οτι στο ένα το πετύχαμε νωρίς και στο άλλο χρειάστηκαν και τα 30 Epochs, η διαδικασία του train οταν δώσαμε τη μεγαλύτερη ανάλυση διήρκησε περισσότερο χρόνο και αναγκάστηκα να του δώσω μικρότερο batch size γιατί δεν αρκούσε η μνήμη στην κάρτα γραφικών που είχα στη διάθεση μου.

4.1 Μελλοντικές προεκτάσεις

4.2 Pre trained Model και Διαφορετική Αρχιτεκτονική

Με τη βοήθεια του Google Colab που μου δίνει για 12 ώρες ένα εικονικό υπολογιστή με ένα GPU (graphics processing unit) την Nvidia P100 με 16GB Video Ram μπόρεσα και δοκίμασα να κάνω train ένα μοντέλο με pre trained weights το EfficientNet-b1 και να δώσω την μέγιστη ανάλυση στις εικόνες που είναι 256x627 που αντιστοιχεί σε ενα STFT με διάρκεια 5s, κατάφερε να φτάσει το ποσοστό ακρίβειας στο ένατο (9) Epoch στο 88% και να τελειώσει με 92% ακρίβεια στο validation set με μέση απώλεια 0.45.

4.3 Dataset

Σύμφωνα με το Dcase2018 ένας αριθμός απο τα δείγματα στο dataset η αντιστοίχιση τους έγινε αυτόματα με αποτέλεσμα κάποιες ετικέτες να είναι λάθος και να μην αντιπροσωπεύουν τι είναι στην πραγματικότητα. Αν έκανα ταξινόμηση αυτές τις ετικέτες χειροκίνητα θα είχα κάποια μικρή άνοδο στο train αλλά εστίασα στο να φέρω πρώτα τα δείγματα στην καλύτερη δυνατή μορφή για να προχωρήσω στη διαδικασία του train.

4.4 Mixup augmentation

Με βάση το οτι στο dataset υπάρχουν ετικέτες και δεν έχει ίση κατανομή στις κλάσεις του, το Mixup είναι μια τεχνική η οποία είναι ιδιαίτερα χρήσιμη όταν δεν είμαστε

δίγουρει ποια τεχνική Augmentation θέλουμε να χρησιμοποιήσουμε ή όπως στην περίπτωση μας το dataset δεν μπορούμε να το πειράξουμε ιδιαίτερα π.χ. όταν έχουμε να κάνουμε με ήχο μια πολύ συνηθισμένη τεχνική Augmentation είναι το Pitch Shift αλλά στην περίπτωση μας δεν θα μας βοηθήσει.

Στο Mixup Augmentation αναμειγνύει τα χαρακτηριστικά και τις αντίστοιχες ετικέτες τους. Τα νευρωνικά δίκτυα είναι επιρρεπή στο να απομνημονεύουν αλλοιωμένες ετικέτες, το mixup συνδυάζει διαφορετικά χαρακτηριστικά μεταξύ τους έτσι ώστε ένα δίκτυο να μην αποκτά υπερβολική εμπιστοσύνη στη σχέση μεταξύ των χαρακτηριστικών και των ετικετών τους.

4.5 Stacking

Το Stacking συνδυάζει τις προβλέψεις από διάφορα train σε διάφορα νευρωνικά δίκτυα. Οι προβλέψεις του βασικού μοντέλου χρησιμοποιούνται ως χαρακτηριστικά για την εκπαίδευση ενός ταξινομητή δεύτερου επιπέδου. Η έξοδος του βασικού μοντέλου είναι ένα διάνυσμα πιθανοτήτων N x K , οπου N είναι ο αριθμός των δειγμάτων και K ο αριθμός των κλάσεων.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] S. Targ, D. Almeida, and K. Lyman. Resnet in resnet: Generalizing residual architectures. arXiv preprint arXiv:1603.08029, 2016

[2] Francesco Camastra andAlessandro Vinciarelli: Deep Learning for Audio, Image and Video Analysis

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep Learning

[4] Christopher M. Bishop: Pattern Recognition and Machine Learning

[5] Maurice A. de Gosson: Introduction to the Theory of Distributions and Applications

Ερευνητικά Άρθρα:

1. Hershey, S., Chaudhuri, S., Ellis, D.P.W., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M., Platt, D., Saurous, R.A., Seybold, B. (2017). "CNN Architectures for Large-Scale Audio Classification." *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.
2. Salamon, J., Bello, J.P. (2017). "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification." *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.
3. Choi, K., Fazekas, G., Sandler, M., Cho, K. (2017). "Convolutional Recurrent Neural Networks for Music Classification." *arXiv preprint arXiv:1703.08082*.
4. Gao Huang, Zhuang Liu, Laurens van der Maaten, και Kilian Q. Weinberger: Densely Connected Convolutional Networks
5. Allen V. Oppenheim and Ronald W. Schafer: Short-time Fourier transform