ΔΠΜΣ Επιστήμη Δεδομένων και Μηχανική Μάθηση

Επεξεργασία Φωνής και Φυσικής Γλώσσας

2η Εργαστηριακή Άσκηση

Ονοματεπώνυμο: Ευάγγελος Τσόγκας

Αριθμός Μητρώου: 03400120

Θεωρητικό υπόβαθρο

MFCCs: Στην παρούσα εργασία ως χαρακτηριστικά επιλέχθηκαν τα Mel-Frequency Cepstral Coefficients (MFCCs). Πριν την εξαγωγή αυτών των χαρακτηριστικών υπολογίζεται πρώτα η συστοιχία φίλτρων Mel Filterbank και στη συνέχεια με τον μετασχηματισμό DCT προκύπτουν τα MFCCs. Ο λόγος που τα MFCCs είναι τόσο διαδεδομένα και πράγματι δίνουν καλά αποτελέσματα είναι επειδή είναι αρκετά διακριτά μεταξύ τους, κάτι το οποίο αποτελεί προϋπόθεση για πολλούς αλγορίθμους μηχανικής μάθησης. Αυτός είναι και ο κύριος λόγος που εφαρμόζουμε τον μετασχηματισμό DCT, καθώς τα Filterbanks σχετίζονται πολύ μεταξύ τους.

Μια επιλογή για βελτίωση της επίδοσης του συστήματος αναγνώρισης φωνής είναι ο συνδυασμός κι άλλων χαρακτηριστικών μαζί με τα MFCCs, όπως για παράδειγμα, το zero-crossing rate το οποίο θεωρείται πολύ σημαντικό χαρακτηριστικό για την κατηγοριοποίηση κρουστικών ήχων. Άλλα πιθανά χαρακτηριστικά είναι οι συχνότητες των formants, η περίοδος του pitch και η ενέργεια.

Γλωσσικό μοντέλο: Το γλωσσικό μοντέλο έχει ως σκοπό την αναπαράσταση των prior πιθανοτήτων των λέξεων/φωνημάτων. Στην παρούσα εργασία ως γλωσσικά μοντέλα χρησιμοποιούμε τις unigram και bigram πιθανότητες των φωνημάτων και τα συγκρίνουμε παρατηρώντας πως το bigram δίνει καλύτερα αποτελέσματα.

Μια πιθανή βελτίωση θα μπορούσε ίσως να ήταν και η χρήση trigram γλωσσικού μοντέλου, δηλαδή οι πιθανότητες ακολουθιών τριών λέξεων/φωνημάτων.

Φωνητικό μοντέλο: Το φωνητικό μοντέλο, στην περίπτωση μας το GMM-HMM, χρησιμοποιείται για τον υπολογισμό των posterior πιθανοτήτων σε συνδυασμό με το γλωσσικό μοντέλο. Δηλαδή, το φωνητικό μοντέλο είναι που τελικά δεδομένου των χαρακτηριστικών ενός σήματος ήχου θα υπολογίσει την πιθανότητα αυτά τα χαρακτηριστικά να ανήκουν σε κάποιο συγκεκριμένο φώνημα ή λέξη.

Μια πιθανή βελτίωση την οποία και δοκιμάζουμε είναι η χρήση DNN-HMM, αντί GMM-HMM.

Προετοιμασία διαδικασίας αναγνώρισης φωνής για τη USC-TIMIT

Το πρώτο πράγμα που κάνουμε πριν κατασκευάσουμε το σύστημα αναγνώρισης φωνής είναι να δημιουργήσουμε ένα φάκελο στο kaldi για το project μας με τα απαραίτητα αρχεία. Αρχικά με τη χρήση του script **create_files.py** δημιουργούνται οι φάκελοι data/train, data/dev και data/test εκ των οποίων κάθε φάκελος περιέχει τα αντίστοιχα δεδομένα σε μορφή αρχείων uttids, utt2spk, wav.scp και text.

Στη συνέχεια, παίρνουμε τα αρχεία path.sh και cmd.sh από το example wsj κάνοντας τις κατάλληλες τροποποιήσεις. Το path.sh συγκεκριμένα το κάνουμε source σε όλα τα bash scripts ώστε να έχουμε διαθέσιμες τις εντολές του kaldi και να μην χρειάζεται να γράφουμε τα paths. Επίσης, με το script **create_softlinks.py** δημιουργούνται τα αντίστοιχα soft links για τους φακέλους steps, utils και το script score.sh, καθώς και οι φάκελοι conf, local, data/lang, data/local/lm_tmp, data/local/nist_lm.

Προετοιμασία γλωσσικού μοντέλου

Πριν τη κατασκευή του γλωσσικού μοντέλου χρησιμοποιούμε το script **create_dict_files.py** για να δημιουργήσουμε όλα τα αρχεία που θα μας χρειαστούν. Συγκεκριμένα δημιουργούμε τα αρχεία φωνημάτων (silence_phones.txt, optional_silence.txt, nonsilence_phones.txt), το λεξικό φωνημάτων (lexicon.txt) τα αρχεία κειμένου (lm_train.text, lm_dev.text, lm_test.text) και το κενό αρχείο extra_questions.txt.

Έχοντας δημιουργήσει τα απαραίτητα αρχεία τρέχουμε το script **create_language_models.sh** για να κατασκευάσουμε ένα unigram και ένα bigram γλωσσικό μοντέλο καθώς και τα FST λεξικού και γραμματικής. Στο συγκεκριμένο script ακολουθούμε τα εξής βήματα:

1. Δημιουργία ενδιάμεσης μορφής μοντέλων.
2. Compilation μοντέλων σε μορφή ARPA.
3. Δημιουργία FST λεξικού (L.fst).
4. Δημιουργία αρχείων spk2utt.
5. Δημιουργία των FST γραμματικής (G.fst) για τα δύο μοντέλα.
6. Υπολογισμός perplexity μοντέλων.

Ερώτημα 1: Στον παρακάτω πίνακα φαίνονται οι τιμές του perplexity των γλωσσικών μοντέλων στο validation και στο test set χρησιμοποιώντας το όρισμα -eval στην εντολή compile-lm. Αυτές οι τιμές δείχνουν πόσο καλά μπορεί ένα μοντέλο να προβλέψει ένα δείγμα. Όσο μικρότερο είναι το perplexity τόσο καλύτερο είναι το μοντέλο, κάτι το οποίο οφείλεται στο γεγονός ότι σχετίζεται με την εντροπία. Βλέπουμε ότι το bigram μοντέλο έχει χαμηλότερο perplexity, άρα είναι και καλύτερο.

Language Model	Perplexity	
	Validation	Test
Unigram	32.48	31.92
Bigram	16.44	15.86

Εξαγωγή ακουστικών χαρακτηριστικών

Χρησιμοποιώντας το script **extract_mfccs.sh** εξάγουμε τα χαρακτηριστικά MFCCs, πραγματοποιούμε Cepstral Mean and Variance Normalization και παίρνουμε μερικές πληροφορίες όσων αφορά τον αριθμό των frames και τη διάσταση των MFCCs.

Ερώτημα 2: Ο λόγος που πραγματοποιούμε normalization είναι ώστε το ακουστικό μοντέλο να γίνει πιο ανθεκτικό σε θόρυβο, όπως για παράδειγμα είναι ο γκαουσιανός λευκός θόρυβος που μετατοπίζει τις μέσες τιμές και ελαττώνει τις διασπορές. Με το Cepstral Mean and Variance Normalization όμως, μετασχηματίζουμε γραμμικά τα χαρακτηριστικά ώστε να έχουν μηδενική μέση τιμή και μοναδιαία διασπορά αντιμετωπίζοντας έτσι τυχόν έκτοπες τιμές. Ο τρόπος που επιτυγχάνεται αυτό είναι αφαιρώντας από κάθε στοιχείο ενός διανύσματος χαρακτηριστικών τη μέση τιμή και διαιρώντας με την τυπική απόκλιση.

Ερώτημα 3: Με την εντολή feat-to-len του kaldi υπολογίζουμε τον αριθμό των frames που έχουν εξαχθεί για κάθε πρόταση. Ο αριθμός των frames για τις 5 πρώτες προτάσεις του train set φαίνεται στον παρακάτω πίνακα.

Utterance id	Frames
usctimit_ema_f1_001	237
usctimit_ema_f1_002	377
usctimit_ema_f1_003	317
usctimit_ema_f1_005	399
usctimit_ema_f1_006	338

Επίσης, με την εντολή feat-to-dim εξάγουμε τη διάσταση των **MFCC** η οποία είναι ίση με **13**, δεδομένου ότι εμπειρικά αυτός είναι ο αριθμός των MFCCs που χρειάζονται για να δώσουν τα καλύτερα αποτελέσματα.

Εκπαίδευση ακουστικών μοντέλων και αποκωδικοποίηση προτάσεων

Όλη η διαδικασία εκπαίδευσης και αποκωδικοποίησης γίνεται με το script **train_and_decode.sh.** Τα βήματα που ακολουθούμε στο script είναι τα εξής:

1. Εκπαίδευση monophone GMM-HMM.
2. Δημιουργία γράφων HCLG για τα unigram και bigram μοντέλα.
3. Αποκωδικοποίηση των προτάσεων των validation και test δεδομένων και με τα δύο μοντέλα.
4. Alignment φωνημάτων και επανάληψη των παραπάνω για triphone GMM-HMM.

Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα της αποκωδικοποίησης με τη μετρική Phone Error Rate.

Language Model	PER (monophone)	
	Validation	Test
Unigram	51.67	48.99
Bigram	46.50	43.60

Όπως παρατηρούμε το καλύτερο μοντέλο είναι το bigram, αφού έχει μικρότερο Phone Error Rate και στο test set, αλλά και στο validation set. Κατά την αποκωδικοποίηση οι δύο υπερπαράμετροι του scoring είναι οι εξής:

- lmwt: Το βάρος του γλωσσικού μοντέλου για το scaling των βαρών του γράφου. Εξετάζονται τιμές από 7 έως 17.
- wip: Το κόστος εισαγωγής λέξης που προστίθεται στον γράφο και πιο συγκεκριμένα στα κόστη του γλωσσικού μοντέλου. Εξετάζονται οι τιμές [0, 0.5, 1].

Στο καλύτερο μοντέλο που είναι το bigram οι τιμές των υπερπαραμέτρων είναι l**mwt=7** και **wip=0.** Αυτές οι τιμές αποθηκεύονται στο φάκελο scoring_kaldi/wer_details.

Στην περίπτωση χρήσης triphone μοντέλου HMM-GMM τα Phone Error Rates φαίνονται στον παρακάτω πίνακα:

Language Model	**PER (triphone)**	
	Validation	**Test**
Unigram	37.80	35.41
Bigram	34.60	31.59

Και πάλι παρατηρούμε πως το bigram είναι το καλύτερο γλωσσικό μοντέλο και επίσης βλέπουμε μεγάλη βελτίωση των αποτελεσμάτων χρησιμοποιώντας triphone GMM-HMM. Οι τιμές των υπερπαραμέτρων του scoring με το bigram μοντέλο σε αυτή την περίπτωση είναι **lmwt=8** και **wip=0**.

Ερώτημα 4: Ένα ακουστικό μοντέλο GMM-HMM αποτελείται από διαφορετικά μαρκοβιανά μοντέλα για κάθε φώνημα/λέξη τα οποία αποτελούνται από κρυφές καταστάσεις που παράγουν χαρακτηριστικά βάση ενός μίγματος γκαουσιανών σε κάθε κατάσταση. Επομένως, στη συγκεκριμένη περίπτωση τα HMM χρησιμοποιούνται για να μοντελοποιήσουν τις μεταβάσεις μεταξύ των κρυφών καταστάσεων ενός φωνήματος δεδομένου των αντίστοιχων παρατηρήσεων (χαρακτηριστικών), ενώ τα GMM χρησιμοποιούνται για τη μοντελοποίηση της κατανομής των χαρακτηριστικών για ένα φώνημα.

Η εκπαίδευση ενός GMM-HMM γίνεται επαναληπτικά με τον forward-backward(Baum Welch) αλγόριθμο ο οποίος στηρίζεται στον αλγόριθμο EM, μπορεί όμως να χρησιμοποιηθεί και ο απλούστερος Viterbi αλγόριθμος (hard EM). Η εκπαίδευση γίνεται ως εξής:

1. Αρχικοποίηση πίνακα μεταβάσεων A και παραμέτρων GMM (π, μ, Σ).
2. Στο Expectation step υπολογίζουμε με τον forward-backward αλγόριθμο τις πιθανότητες γ(j) (πιθανότητα να βρισκόμαστε στην κατάσταση j) και ξ(i, j) (πιθανότητα μετάβασης από την κατάσταση i στην κατάσταση j).
3. Στο Maximization step με χρήση των γ και ξ ενημερώνουμε τις τιμές του πίνακα μετάβασης (a_{ii}) την πιθανότητα της αρχικής κατάστασης (π_i) και τις παραμέτρους των GMM (π, μ, Σ) κάθε κατάστασης.

Η διαδικασία εκπαίδευσης ενός μονοφωνικού μοντέλου είναι όπως περιγράφηκε παραπάνω. Στο μονοφωνικό μοντέλο σπάμε μια έκφραση στα επιμέρους φωνήματά της και κάθε φώνημα αντιστοιχεί σε ένα HMM κάθε κατάσταση του οποίου αναπαρίσταται από ένα μείγμα γκαουσιανών των οποίων οι παράμετροι εκπαιδεύονται χρησιμοποιώντας διανύσματα παρατηρήσεων των δεδομένων εκπαίδευσης.

Ερώτημα 5: Η posterior πιθανότητα σύμφωνα με τον τύπο του Bayes είναι η:

$$P(W \mid X) = \frac{P(X \mid W)P(W)}{P(X)}$$

όπου για το πρόβλημα αναγνώρισης φωνής το W είναι μια λέξη ή φώνημα και το X είναι μια ακολουθία χαρακτηριστικών.

Η πιο πιθανή λέξη δεδομένης μια ακολουθίας χαρακτηριστικών επομένως είναι η:

$$W^* = argmax\, P(X \mid W)P(W)$$

και η εύρεσή της ονομάζεται decoding και γίνεται με τον υπολογισμό του καλύτερου μονοπατιού. Πιο συγκεκριμένα η πιθανότητα $P(X \mid W)$ ονομάζεται likelihood και υπολογίζεται από το ακουστικό μοντέλο, ενώ η πιθανότητα $P(W)$ προκύπτει από το γλωσσικό μοντέλο.

Ερώτημα 6: Ο γράφος HCLG του kaldi όπως περιγράφεται και στην ιστοσελίδα του αποτελεί τη σύνθεση των H, C, L, και G τα όποια είναι τα εξής:

H: Περιέχει τις πληροφορίες των HMM του οποίου το output αναπαριστά τα context-dependent φωνήματα (πχ. triphone μοντέλο)

C: Αναπαριστά το context dependency και το output είναι φωνήματα.

L: Είναι το FST λεξικού. Στη δική μας περίπτωση αντιστοιχούμε φωνήματα σε φωνήματα, αλλά θα μπορούσαμε να αντιστοιχούμε φωνήματα σε λέξεις.

G: Είναι ο αποδοχέας γραμματικής, δηλαδή το FST του γλωσσικού μοντέλου που στη δική μας περίπτωση είναι γλωσσικό μοντέλο φωνημάτων.

Μοντέλο DNN-HMM με PyTorch

Αφού εκπαιδεύσαμε μοντέλα GMM-HMM, τώρα θα εκπαιδεύσουμε ένα μοντέλο DNN-HMM, δηλαδή θα εξάγουμε τις πιθανότητες των φωνημάτων από ένα νευρωνικό δίκτυο. Χρησιμοποιούμε bigram γλωσσικό μοντέλο και triphone alignments.

Στο παρακάτω screenshot φαίνονται πληροφορίες για την αρχιτεκτονική και τη διαδικασία εκπαίδευσης του δικτύου.

```
The network architecture is:
 TorchDNN(
  (f): Sequential(
    (0): FeedForwardBlock(
      (b): BatchNorm1d(195, eps=1e-05, momentum=0.1, affine=True, track
      (f): Linear(in_features=195, out_features=256, bias=True)
      (d): Dropout(p=0.2, inplace=False)
      (a): ReLU()
    )
    (1): FeedForwardBlock(
      (b): BatchNorm1d(256, eps=1e-05, momentum=0.1, affine=True, track
      (f): Linear(in_features=256, out_features=256, bias=True)
      (d): Dropout(p=0.2, inplace=False)
      (a): ReLU()
    )
  )
  (clf): Linear(in_features=256, out_features=1040, bias=True)
)
Epoch: 1  -  loss: 2.3006622838016004  -  val_loss: 1.8488804649353028
Epoch: 2  -  loss: 1.564631116350312  -  val_loss: 1.721786088180542
Epoch: 3  -  loss: 1.370309769461562  -  val_loss: 1.6927316911697388
Epoch: 4  -  loss: 1.25938327681814  -  val_loss: 1.70913095703125
Epoch: 5  -  loss: 1.1834682016619693  -  val_loss: 1.7348075288772582
Epoch: 6  -  loss: 1.1263130050206105  -  val_loss: 1.7675369354248047
Early stopping
```

Το Phone Error Rate στο test set αυτή τη φορά ήταν **PER=32.12** που είναι παρόμοιο με αυτό του αντίστοιχου μοντέλου GMM-HMM.

Ερώτημα 7: Η διαφορά ενός DNN-HMM από ένα GMM-HMM είναι στο πώς υπολογίζονται οι πιθανότητες των παρατηρήσεων δεδομένων των χαρακτηριστικών. Στην περίπτωση των GMM όπως έχουμε ήδη περιγράψει οι πιθανότητες υπολογίζονται βάση μιγμάτων γκαουσιανών κατανομών των οποίων οι παράμετροι εκπαιδεύονται μαζί με το transition matrix του HMM. Αν όμως εκπαιδεύσουμε ένα DNN μπορούμε να εξάγουμε τις πιθανότητες παρατηρήσεων από το output layer ως πιθανότητες της συνάρτησης Softmax και στη συνέχεια τα HMM χρησιμοποιούν αυτές τις πιθανότητες και κατά την εκπαίδευση ενημερώνονται μόνο οι πίνακες μετάβασης.

Η προσθήκη ενός DNN μπορεί να μας ωφελεί όταν έχουμε πάρα πολλά δεδομένα και πολύ μεγάλο λεξιλόγιο μιας και σε τέτοιες περιπτώσεις είναι που πραγματικά λάμπουν τα νευρωνικά δίκτυα.

Θα μπορούσαμε να είχαμε εκπαιδεύσει εξαρχής ένα DNN-HMM, αλλά σε μικρά προβλήματα όπως το δικό μας το κόστος δημιουργίας και εκπαίδευσης ενός DNN ξεπερνά τη βελτίωση, αφού παρατηρούμε πως το αποτέλεσμα δεν βελτιώθηκε.

Ερώτημα 8: Ο σκοπός του Batch Normalization είναι ο ίδιος με το γενικό normalization των δεδομένων όταν εκπαιδεύουμε ένα νευρωνικό δίκτυο. Το να φέρνουμε τα δεδομένα σε μια μορφή με μηδενική μέση τιμή και μοναδιαίο variance αποδεδειγμένα κάνει την εκπαίδευση πιο γρήγορη και πιο σταθερή. Όταν όμως έχουμε βαθιά νευρωνικά δίκτυα οι τιμές των χαρακτηριστικών μεταβάλλονται κατά το output των συναρτήσεων ενεργοποίησης και γι' αυτό χρησιμοποιούμε το Batch Normalization ώστε να κάνουμε re-scale των δεδομένων στα πλαίσια κάθε batch.