ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Επεξεργασία Φωνής και Φυσικής Γλώσσας

Προπαρασκευή 2ου Εργαστηρίου: Αναγνώριση φωνής με το Kaldi toolkit

1 Περιγραφή

Σκοπός της άσκησης αυτής είναι η υλοποίηση ενός συστήματος επεξεργασίας και αναγνώρισης φωνής με το εργαλείο Kaldi, το οποίο χρησιμοποιείται ευρέως στον ερευνητικό τομέα, αλλά και όχι μόνο, για την εκπαίδευση state-of-the-art συστημάτων αναγνώρισης φωνής.

Πιο συγκεκριμένα, το σύστημα που θα αναπτύξετε αφορά σε αναγνώριση φωνημάτων (Phone Recognition) από ηχογραφήσεις της USC-TIMIT. Θα σας δοθούν δεδόμενα audio από 4 διαφορετικούς ομιλητές, με τα αντίστοιχα transcriptions, ώστε να εκπαιδεύσετε και να εκτιμήσετε το σύστημά σας.

Η διαδικασία σχεδιασμού του συστήματος μπορεί να χωριστεί σε 4 μέρη. Το πρώτο μέρος αποσκοπεί στην εξαγωγή κατάλληλων ακουστικών χαρακτηριστικών από τα φωνητικά δεδομένα (Mel-Frequency Cepstral Coefficients). Τα εν λόγω χαρακτηριστικά είναι στην ουσία ένας αριθμός συντελεστών cepstrum που εξάγονται μετά από ανάλυση των σημάτων σημάτων φωνής με μια ειδικά σχεδιασμένη συστοιχία φίλτρων (Mel filterbank). Η συστοιχία αυτή είναι εμπνευσμένη από το μη γραμμικό τρόπο που το ανθρώπινο αυτί αντιλαβάνεται τον ήχο και ειδικά σχεδιασμένη από ψυχοακουστικές μελέτες. Το δεύτερο μέρος αφορά τη δημιουργία γλωσσικών μοντέλων από τα transcriptions του σετ δεδομένων, τα οποία θα δίνουν την a priori πιθανότητα στο τελικό σύστημα. Το τρίτο μέρος αφορά την εκπαίδευση των ακουστικών μοντέλων χρησιμοποιώντας τα ακουστικά χαρακτηριστικά τα οποία εξήχθησαν. Τέλος, συνδυάζοντας τις παραπάνω μονάδες μπορεί να κατασκευαστεί το τελικό σύστημα αναγνώρισης φωνής, το οποίο δεδομένου ενός σήματος φωνής, εξάγει τα ακουστικά χαρακτηριστικά και τα χρησιμοποιεί ώστε να αποκωδικοποιήσει το σήμα σε μία ακολουθία φωνημάτων ή λέξεων.

2 Θεωρητικό υπόβαθρο

Κατά την προπαρασκευή θα πρέπει να εξοικειωθείτε με συγκεκριμένες έννοιες που θα χρησιμοποιηθούν κατά τη διεξαγωγή του εργαστηρίου. Συγκεκριμένα, θα θέλαμε να γνωρίζετε για τις παρακάτω έννοιες:

1. Mel-frequency Cepstral Coefficients (MFCCs)
2. Γλωσσικά Μοντέλα (Language Models)
3. Φωνητικά Μοντέλα (Acoustic Models)

Στην τελική σας αναφορά, θα θέλαμε εν συντομία να αναπτύξετε τόσο τις παραπάνω έννοιες όσο και να σχολιάσετε την απόδοση τους. Μην μείνετε στα βήματα των βασικών αλγορίθμων αλλά προσπαθείστε να προτείνετε/εκτιμήσετε πως θα μπορούσε να βελτιωθεί το σύστημα αναγνώρισης φωνής που έχετε αναπτύξει.

Ως προετοιμασία για το εργαστήριο διαβάστε τα παρακάτω:

- Κεφάλαιο 14 από το βιβλίο του μαθήματος **[R&S] Theory and Applications of Digital Speech Processing** των Lawrence R. Rabiner and Ronald W. Schafer (Pearson, 2011), σχετικά με Automatic Speech Recognition (ASR) συστήματα.
- Mel-frequency Cepstral Coefficients (MFCCs)
- GMM-HMM for acoustic modeling
- Kaldi tutorial 1
- Kaldi tutorial 2
- Kaldi tutorial 3

3 Βήματα προπαρασκευής

1. Εγκαταστήστε το Kaldi σύμφωνα με τις οδηγίες που θα δωθούν στις διευκρινίσεις του helios.

2. Εξοικειωθείτε με το εργαλείο Kaldi . Η γλώσσα με την οποία έχει αναπτυχθεί είναι C++, αλλά οι κύριες λειτουργίες που μας ενδιαφέρουν καλούνται από bash scripts. Υπάρχουν ήδη υλοποιημένες διαδικασίες για την ανάπτυξη μοντέλων αναγνώρισης φωνής για πολλά συνηθισμένα σετ δεδομένων μέσα στο φάκελο *egs* του Kaldi. Παρ' όλα αυτά, για το σετ δεδομένων που θα σας δοθεί θα πρέπει να το υλοποιήσετε μόνοι σας τη διαδικασία από την αρχή.

3. Κατεβάστε τα δεδομένα από το παρακάτω link:

   `https://drive.google.com/file/d/1_mIoioHMeC2HZtIbGs1LcL4kkIF696nB/view?usp=sharing`

   Τα δεδομένα περιλαμβάνουν ηχογραφήσεις από 4 ομιλητές με ονόματα: m1, m3 (άντρες) και f1, f5 (γυναίκες). Σε κάθε ομιλητή αντιστοιχούν 460 προτάσεις (Προσοχή: στον ομιλητή m1 λείπουν οι προτάσεις 231 έως 235 λόγω σφάλματος στην ηχογράφηση).

   Τα αρχεία ήχου βρίσκονται στο φάκελο *wav*, χωρισμένα σε φακέλους ανάλογα με το όνομα του ομιλητή και το όνομα κάθε αρχείου περιγράφει σε ποιον ομιλητή και σε ποια πρόταση αντιστοιχεί. Στο αρχείο *transcription.txt* θα βρείτε το κείμενο που εκφωνούν οι ομιλητές σε κάθε πρόταση (1η γραμμή → 1η πρόταση, 2η γραμμή → 2η πρόταση κ.ο.κ.) και στο φάκελο *filesets* θα βρείτε ποιές προτάσεις αντιστοιχούν στο σετ εκπαίδευσης, στο σετ επαλήθευσης και στο σετ αποτίμησης (training, validation, testing).

4. Κατασκευή αρχικού σκελετού:

   - Μέσα στο φάκελο *egs* δημιουργείστε ένα φάκελο *usc*, μέσα στον οποίο θα εργάζεστε από εδώ και πέρα.
   - Δημιουργείστε το φάκελο *data* και τους υποφακέλους *data/train*, *data/dev*, *data/test*, μέσα στους οποίους θα δημιουργήσετε αρχεία-δείκτες τα οποία θα περιγράφουν τα δεδομένα εκπαίδευσης, επαλήθευσης και αποτίμησης αντίστοιχα.
   - Μέσα σε κάθε έναν από αυτούς τους 3 φακέλους θα πρέπει να δημιουργήσετε τα εξής αρχεία:
     - uttids: περιέχει στην κάθε του γραμμή ένα μοναδικό συμβολικό όνομα για κάθε πρόταση του συγκεκριμένου συνόλου δεδομένων (δηλαδή το περιεχόμενο των αρχείων στο φάκελο *filesets*) τα οποία από εδώ και πέρα θα αναφέρουμε ως utterance_ids
     - utt2spk: περιέχει σε κάθε γραμμή τον ομιλητή που αντιστοιχεί σε κάθε πρόταση και είναι της μορφής:

       utterance_id_1 <κενό> speaker_id
       utterance_id_2 <κενό> speaker_id
       κ.ο.κ.

       όπου ως speaker_id επιλέγουμε αντίστοιχα τα m1, m3, f1, f5
     - wav.scp: περιέχει τη θέση του αρχείου ήχου που αντιστοιχεί σε κάθε πρόταση και είναι της μορφής:

       utterance_id_1 <κενό> /path/to/wav1
       utterance_id_2 <κενό> /path/to/wav2
       κ.ο.κ.

     - text: περιέχει το κείμενο που αντιστοιχεί στην κάθε πρόταση και είναι της μορφής:

       utterance_id_1 <κενό> <utterance 1 text>
       utterance_id_2 <κενό> <utterance 2 text>
       κ.ο.κ.

   - Τέλος, για κάθε αρχείο text που δημιουργήσατε πρέπει να αντικαταστήσετε τις λέξεις που περιέχουν οι προτάσεις με τις αντίστοιχες αλληλουχίες φωνημάτων. Για το λόγο αυτό σας δίνεται μαζί με τα υπόλοιπα δεδομένα το λεξικό (lexicon.txt), το οποίο αντιστοιχίζει κάθε λέξη της αγγλικής γλώσσας στην αλληλουχία φωνημάτων που της αντιστοιχεί. Προσέξτε σε αυτό το βήμα να μετατρέψετε αρχικά όλους το χαρακτήρες σε lower case, καθώς και να αφαιρέσετε τους ειδικούς χαρακτήρες (π.χ. τελείες, παύλες κτλ.) εκτός από τα single quotes (’). Επίσης, στην αρχή και στο τέλος πρέπει να προσθέσετε το φώνημα της σιωπής (sil). Δίνεται το παράδειγμα για την 1η πρόταση του ομιλητή f1:

     This was easy for us.

     Θα πρέπει να μετασχηματιστεί σε:

     sil dh ih s w ao z iy z iy f r er ah s sil

4 Βήματα κυρίως μέρους

4.1 Προετοιμασία διαδικασίας αναγνώρισης φωνής για τη USC-TIMIT

1. Από τη διαδικασία για τη Wall Street Journal (*wsj*) που βρίσκεται στο φάκελο *egs* πάρτε τα αρχεία `path.sh` και `cmd.sh`. Στο αρχείο path.sh πρέπει να θέσετε τη μεταβλητή `KALDI_ROOT` στο directory που βρίσκεται ο κύριος φάκελος της εγκατάστασης του Kaldi. Το αρχείο αυτό το κάνετε source στην αρχή κάθε bash script σας, ώστε να έχετε διαθέσιμες όλες τις εντολές του Kaldi.

   Επίσης, στο `cmd.sh` πρέπει να αλλάξετε τις τιμές των μεταβλητών `train_cmd`, `decode_cmd` και `cuda_cmd` σε `run.pl`.

2. Δημιουργείστε soft links μέσα στο φάκελο της δικής σας διαδικασίας με ονόματα 'steps' και 'utils' τα οποία θα δείχνουν στους αντίστοιχους φακέλους της *wsj*.

3. Δημιουργείστε το φάκελο *local* και μέσα σε αυτόν ένα soft link που να δείχνει στο αρχείο score_kaldi.sh που βρίσκεται μέσα στο 'steps'.

4. Δημιουργείστε το φάκελο *conf* και μέσα σε αυτόν αντιγράψτε το αρχείο mfcc.conf που σας δώθηκε στις διευκρινίσεις.

5. Τέλος, δημιουργείστε τους εξής φακέλους: *data/lang*, *data/local/dict*, *data/local/lm_tmp*, *data/local/nist_lm*.

4.2 Προετοιμασία γλωσσικού μοντέλου

1. Μέσα στο φάκελο *data/local/dict* θα αποθηκεύσετε τα βασικά αρχεία που θα χρησιμεύσουν για τη δημιουργία του γλωσσικού μοντέλου.

   - Τα αρχεία silence_phones.txt και optional_silence.txt που θα περιέχουν μόνο το φώνημα της σιωπής (sil).
   - Το αρχείο nonsilence_phones.txt το οποίο θα περιέχει όλα τα υπόλοιπα φωνήματα (1 σε κάθε γραμμή και sorted).
   - Το αρχείο lexicon.txt το οποίο αποτελεί το λεξικό του γλωσσικού μοντέλου. Επειδή θα ασχοληθούμε με αναγνώριση φωνημάτων και όχι λέξεων, το λεξικό θα πρέπει να είναι μία 1-1 αντιστοιχία των φωνημάτων με τον εαυτό τους. Οπότε, σε κάθε του γραμμή θα περιέχει ένα φώνημα, έπειτα <κενό> και μετά πάλι το ίδιο φώνημα. Μην ξεχάσετε να συμπεριλάβετε το φώνημα της σιωπής.
   - Δημιουργείστε το αρχείο lm_train.text προσθέτοντας στο αρχείο text που δημιουργήσατε στην προπαρασκευή τις ειδικές μονάδες <s> και </s> στην αρχή και στο τέλος της κάθε πρότασης αντίστοιχα. 3 τέτοια αρχεία πρέπει να δημιουργηθούν, ένα για κάθε σετ.
   - Τέλος, δημιουργείστε το αρχείο extra_questions.txt το οποίο θα είναι κενό.

2. Μέσα στο φάκελο *data/local/lm_tmp* θα δημιουργήσετε την ενδιάμεση μορφή του γλωσσικού μοντέλου. Χρησιμοποιήστε την εντολή **build-lm.sh** του πακέτου IRSTLM που έχει εγκατασταθεί μαζί με το Kaldi.

   build-lm.sh -i <αρχείο lm_train.text> -n <τάξη γλωσσικού μοντέλου> -o <αρχείο_εξόδου.ilm.gz>

   Σημείωση: Δημιουργείστε unigram και bigram μοντέλα.

3. Μέσα στο φάκελο *data/local/nist_lm* θα αποθηκευτεί το compiled γλωσσικό μοντέλο σε μορφή ARPA. Χρησιμοποιείστε την εντολή **compile-lm**.

   compile-lm <αρχείο .ilm.gz> -t=yes /dev/stdout | grep -v unk | gzip -c > <αρχείο_εξόδου.arpa.gz>

   Το unigram μοντέλο να ονομαστεί *lm_phone_ug.arpa.gz* ενώ το bigram *lm_phone_bg.arpa.gz*

4. Μέσα στο φάκελο *data/lang* θα δημιουργήσετε το FST του λεξικού της γλώσσας (L.fst) χρησιμοποιώντας την εντολή του Kaldi **prepare_lang.sh**.

5. Χρησιμοποιήστε την εντολή `sort` για να ταξινομήσετε τα αρχεία wav.scp, text και utt2spk στους φακέλους data/train, data/dev και data/test.

6. Εκτελέστε το script `utils/utt2spk_to_spk2utt.pl` ώστε να δημιουργήσετε το αρχείο spk2utt

7. Τέλος, θα πρέπει να δημιουργήσετε το FST της γραμματικής (G.fst). Ακολουθήστε την ίδια διαδικασία με τη διαδικασία *timit* του Kaldi, η οποία βρίσκεται στο αρχείο local/timit_format_data.sh.

Ερώτημα 1: Για τα γλωσσικά μοντέλα που δημιουργήσατε υπολογίστε το perplexity στο validation και στο test set. Τι δείχνουν αυτές οι τιμές?

4.3 Εξαγωγή ακουστικών χαρακτηριστικών

Εξάγετε τα MFCCs και για τα 3 σετ, χρησιμοποιώντας τις εντολές του Kaldi (make_mfcc.sh, compute_cmvn_stats.sh).

Ερώτημα 2: Με τη δεύτερη εντολή πραγματοποιείται το λεγόμενο Cepstral Mean and Variance Normalization. Τι σκοπό εξυπηρετεί? (Bonus: Δώστε μια μαθηματικά τεκμηριωμένη απάντηση)

Ερώτημα 3: Πόσα ακουστικά frames εξήχθησαν για κάθε μία από τις 5 πρώτες προτάσεις του training set? Τι διάσταση έχουν τα χαρακτηριστικά?

4.4 Εκπαίδευση ακουστικών μοντέλων και αποκωδικοποίηση προτάσεων

1. Εκπαιδεύστε ένα monophone GMM-HMM ακουστικό μοντέλο πάνω στα train δεδομένα.
   (Hint: `steps/train_mono.sh`)

2. Δημιουργήστε το γράφο HCLG του Kaldi σύμφωνα με τη γραμματική (G) του προηγούμενου βήματος. Υπενθύμιση: πρέπει να δοκιμάσετε και unigrams και bigrams. (Hint: `utils/mkgraph.sh`)

3. Αποκωδικοποιήστε τις προτάσεις των validation και των test δεδομένων με τον αλγόριθμο Viterbi.
   (Hint: `steps/decode.sh`)

4. Παρουσιάστε τα αποτελέσματα της αποκωδικοποίησης με τη μετρική Phone Error Rate (PER):

$$PER = 100\frac{insertions + substitutions + deletions}{\#phonemes}$$

   όπου $\#phonemes$ είναι ο συνολικός αριθμός φωνημάτων μέσα στο transcription.

   Αναφέρετε ποιές είναι οι 2 υπερπαράμετροι της διαδικασίας scoring σε αυτό το βήμα και τι αντιπροσωπεύουν. Τι τιμές πήραν στο καλύτερό σας μοντέλο? (Hint: `local/score.sh` και αποτελέσματα στο φάκελο *exp_mono_bg/decode_test/scoring_kaldi/best_wer*)

5. Κάντε alignment των φωνημάτων χρησιμοποιώντας το monophone μοντέλο. Έπειτα, χρησιμοποιώντας αυτά τα alignments εκπαιδεύστε ένα triphone μοντέλο. Δημιουργήστε το γράφο HCLG. Πραγματοποιήστε εκ νέου αποκωδικοποίηση και παρουσιάστε τα αποτελέσματά σας. (Hint: `steps/align_si.sh`, `steps/train_deltas.sh`)

Ερώτημα 4: Εξηγήστε τη δομή ενός ακουστικού μοντέλου GMM-HMM. Τι σκοπό εξυπηρετούν τα μαρκοβιανά μοντέλα στη συγκεκριμένη περίπτωση και τι τα μίγματα γκαουσιανών? Με ποιό τρόπο γίνεται η εκπαίδευση ενός τέτοιου μοντέλου? Περιγράψτε τη διαδικασία εκπαίδευσης ενός μονοφωνικού μοντέλου.

Ερώτημα 5: Γράψτε πώς υπολογίζεται η a posteriori πιθανότητα σύμφωνα με τον τύπο του Bayes για το πρόβλημα της αναγνώρισης φωνής. Συγκεκριμένα, πώς βρίσκεται η πιο πιθανή λέξη (ή φώνημα στην περίπτωσή μας) δεδομένης μίας ακολουθίας ακουστικών χαρακτηριστικών?

Ερώτημα 6: Εξηγήστε τη δομή του γράφου HCLG του Kaldi περιγραφικά.

4.5 Bonus: Μοντέλο DNN-HMM με PyTorch

Για το βήμα αυτό θα σας δωθεί βοηθητικό υλικό στις διευκρινίσεις του mycourses. Εγκαταστήστε τη βιβλιοθήκη kaldi-io-for-python (https://github.com/vesis84/kaldi-io-for-python). Μπορείτε να χρησιμοποιήσετε αυτή την εντολή:

```
pip install git+https://github.com/vesis84/kaldi-io-for-python
```

Έπειτα κατεβάστε το βοηθητικό κώδικα που σας δίνεται μέσα στο φάκελο .../egs/usc

1. Εξάγετε τα triphone alignments για τα train, validation και test sets.

2. Εξάγετε τα cmvn statistics όπως φαίνεται στο run_dnn.sh

3. Χρησιμοποιώντας την κλάση TorchSpeechDataset που σας δίνεται μπορείτε να διαβάσετε τα δεδομένα train, validation, test. Η κλάση αυτή περιέχει τις ιδιότητες: *feats*, που είναι ένας πίνακας στον οποίο θα αποθηκευτούν τα MFCCs, *labels* όπου αποθηκεύονται τα φωνήματα με αριθμούς που αρχίζουν από το 0, *uttids* όπου αποθηκεύονται τα utterance_ids και *end_indices* όπου αποθηκεύεται η θέση του πίνακα όπου τελειώνουν τα χαρακτηριστικά της μίας πρότασης και αρχίζουν της επόμενης στον πίνακα *feats*.

4. Δημιουργείστε ένα Βαθύ Νευρωνικό Δίκτυο (DNN) το οποίο θα ταξινομεί τα φωνήματα (labels). Το δίκτυό σας πρέπει να το εκπαιδεύσετε πάνω στο train set, κάνοντας ρύθμιση των υπερπαραμέτρων στο validation set. Το δίκτυο πρέπει να αποτελείται από feedforward layers, ακολουθούμενα από dropout και relu activations (εκτός από το τελευταίο (output layer). Στην είσοδο κάθε Layer μπορεί να προστεθεί και 1D Batch Normalization.

5. Χρησιμοποιώντας το καλύτερό σας μοντέλο, κάντε πρόβλεψη των φωνημάτων του test set και αποθηκεύστε τις a posteriori πιθανότητες για κάθε ακουστικό frame σε μορφή αναγνωρίσιμη από τα scripts του Kaldi (θα δοθεί παράδειγμα).

6. Τέλος, κάντε αποκωδικοποίηση στο test set χρησιμοποιώντας τις παραπάνω πιθανότητες και το βοηθητικό script (decode_dnn.sh), όπως φαίνεται στο run_dnn.sh. Παρουσιάστε τα αποτελέσματά σας.

Ερώτημα 7: Εξηγήστε σε τι διαφέρει το DNN-HMM από το GMM-HMM και πώς συνδυάζεται το DNN με ένα HMM. Που πιθανώς να μας οφελεί η προσθήκη DNN έναντι των GMM ? Θα μπορούσαμε να έχουνε εκπαιδεύσει εξαρχής ένα DNN-HMM?

Ερώτημα 8: Εξηγήστε τη λειτουργία του Batch Normalization.

Παραδοτέα:

1. Αναπτύξτε και σχολιάστε τις έννοιες που σας δόθηκαν (MFCCs, γλωσσικά και ακουστικά μοντέλα). Μην μείνετε στα βήματα του βασικού αλγορίθμου αλλά προσπαθείστε να προτείνετε/εκτιμήσετε πως θα μπορούσε να βελτιωθεί το σύστημα αναγνώρισης φωνής που έχετε αναπτύξει.

2. Σύντομη αναφορά που θα περιγράφει την διαδικασία που ακολουθήθηκε σε κάθε βήμα, τις απαντήσεις στα ερωτήματα, καθώς και τα σχετικά αποτελέσματα.

3. Κώδικας συνοδευόμενος από σύντομα σχόλια.