3η Εργασία: Βαθιά Μάθηση

3η Εργασία: Βαθιά Μάθηση
- Image Captioning
- Σύνολο δεδομένων
  - Εποπτεία των δεδομένων
  - Κατέβασμα του συνόλου δεδομένων
  - Google Drive mount (προαιρετικό)
- Το μοντέλο
- Αξιολόγηση της ποιότητας του captioning (BLEU)
- Βελτιώσεις (και παραδοτέα)
  - Encoder
  - Προεπεξεργασία κειμένου
  - Embeddings
  - Sentence Generator (Beam Search)
  - Υπερπαράμετροι του decoder
- Παραδοτέο

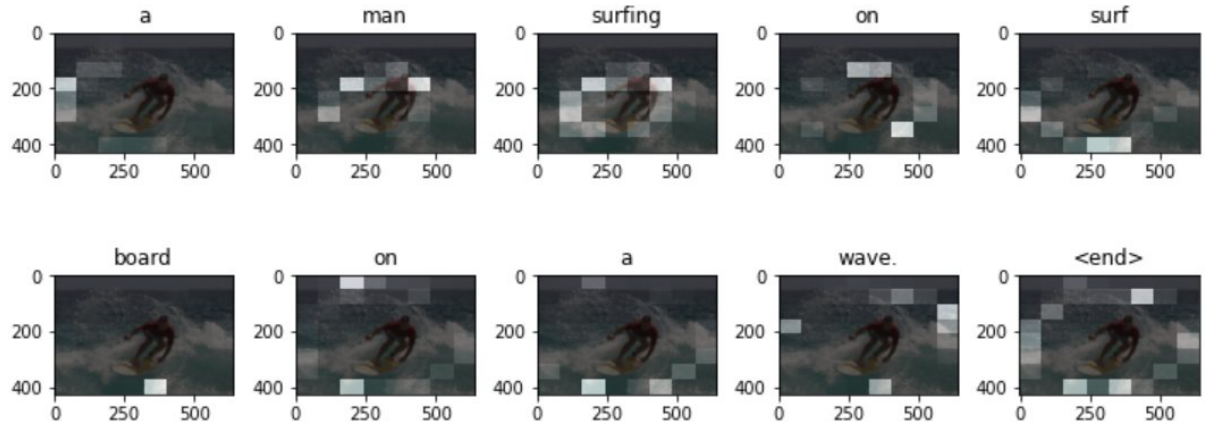
Διαγωνισμός (προαιρετικός)
- Εικόνες διαγωνισμού
- Δημιουργία αρχείου υποβολής
- Το site του διαγωνισμού στο Codalab
- Υποβολή απάντησης και leaderboard

Image Captioning

Θέμα της 3ης εργασίας του μαθήματος είναι η Βαθιά Μάθηση. Θα μελετήσουμε ένα πρόβλημα που συνδυάζει Όραση Υπολογιστών και Επεξεργασία Φυσικής Γλώσσας. Συγκεκριμένα, θα φτιάξουμε ένα νευρωνικό δίκτυο παραγωγής λεκτικών περιγραφών από εικόνες (Image Captioning).

Σημείο εκκίνησης θα είναι το επίσημο tutorial (και notebook) του TensorFlow "Image captioning with visual attention". Θα δουλέψουμε ωστόσο σε άλλο dataset και θα προσπαθήσουμε να βελτιώσουμε το tutorial σε διάφορα σημεία.

Τέλος, θα υπάρχει προαιρετικά η δυνατότητα υποβολής προβλέψεων πάνω σε δεδομένα ελέγχου χωρίς λεκτικές περιγραφές για τη συμμετοχή σε ένα μικρό in-class competition χωρίς καμία βαθμολογική σημασία.



Ανοίξτε το notebook στο Colab ώστε να βλέπετε όλα τα κελιά. Στο web version χρειάζεται να κάνετε "Toggle code" και "Toggle section".

Σύνολο δεδομένων

Τα ευρύτερα χρησιμοποιούμενα datasets στο Image Captioning είναι τα Flickr8k, Flickr30k, και το COCO. Το παράδειγμα του TensorFlow χρησιμοποιεί το Flickr8k και το Conceptual Captions. Εμείς θα χρησιμοποιήσουμε το "flickr30k-images-ecemod", ένα split του Flick30k ειδικά για το μάθημά μας.

Τα δεδομένα του flickr30k-images-ecemod είναι τα εξής:

- ένας φάκελος "image_dir" με 31.783 εικόνες από το Flickr
- ένα αρχείο "captions_new.csv" με 148.915 captions για τις εικόνες του "image_dir"
- ένα αρχείο "train_files.csv" λίστα των 21.000 εικόνων που αποτελούν το training set
- ένα αρχείο "test_files.csv" λίστα των 4.524 εικόνων που αποτελούν το test set

Εποπτεία των δεδομένων

Το flickr30k-images-ecemod έχει παρόμοια οργάνωση με το COCO. Κάθε εικόνα έχει 5 captions που έχουν γίνει από διαφορετικούς ανθρώπους μέσω της υπηρεσίας Mechanical Turk της Amazon. Ένα παράδειγμα:

Παράδειγμα εικόνας: _100007487.jpg

```
_100007487.jpg#0    A young child walks down a gravel path lined with a row of red outdoor chairs .
```

```
_100007487.jpg#1    A racetrack with red chairs stacked beside fence with a child walking .
```

```
_100007487.jpg#2    A child in a striped shirt walks by some red chairs .
```

```
_100007487.jpg#3    A child walking and leaving a trail behind them .
```

```
_100007487.jpg#4    A little kid is walking next to red banners .
```

Κάθε caption έχει τρία πεδία, το όνομα του αρχείου της εικόνας, τον αύξοντα αριθμό του caption και τέλος το ίδιο το caption.

Κατέβασμα του συνόλου δεδομένων

Με τον κώδικα που ακολουθεί, θα κατεβάσουμε το δικό μας dataset, αντικαθιστώντας τα sections "Choose a dataset" και "Download the dataset". Τα προηγούμενα κελιά με τα installations και imports πρέπει να τα τρέξετε.

Στο επόμενο κελί κατεβάζουμε τις εικόνες του dataset:

```
# Download image files
image_zip = tf.keras.utils.get_file('flickr30k-images-ecemod.zip',
                                    cache_subdir=os.path.abspath('.'),
                                    origin='https://spartacus.1337.cx/flickr-
mod/flickr30k-images-ecemod.zip',
                                    extract=True)
os.remove(image_zip)
```

και με το επόμενο τα "captions_new.csv", "train_files.csv" και "test_files.csv":

```
# Download captions file
captions_file = tf.keras.utils.get_file('captions_new.csv',
                                        cache_subdir=os.path.abspath('.'),
                                        origin='https://spartacus.1337.cx/flickr-
mod/captions_new.csv',
                                        extract=False)

# Download train files list
train_files_list = tf.keras.utils.get_file('train_files.csv',
                                        cache_subdir=os.path.abspath('.'),
                                        origin='https://spartacus.1337.cx/flickr-
mod/train_files.csv',
                                        extract=False)

# Download test files list
test_files_list = tf.keras.utils.get_file('test_files.csv',
                                        cache_subdir=os.path.abspath('.'),
                                        origin='https://spartacus.1337.cx/flickr-
mod/test_files.csv',
                                        extract=False)
```

Με τον ακόλουθο κώδικα οργανώνουμε τα filenames και τα captions σε λίστες και ετοιμάζουμε τα train και test sets για το TensorFlow.

```
path="."
IMAGE_DIR="image_dir"
path = pathlib.Path(path)

captions = (path/captions_file).read_text().splitlines()
captions = (line.split('\t') for line in captions)
captions = ((fname.split('#')[0], caption) for (fname, caption) in captions)

cap_dict = collections.defaultdict(list)
for fname, cap in captions:
  cap_dict[fname].append(cap)

train_files = (path/train_files_list).read_text().splitlines()
train_captions = [(str(path/IMAGE_DIR/fname), cap_dict[fname]) for fname in
train_files]
```

```
test_files = (path/test_files_list).read_text().splitlines()
test_captions = [(str(path/IMAGE_DIR/fname), cap_dict[fname]) for fname in
test_files]

train_raw = tf.data.experimental.from_list(train_captions)
test_raw = tf.data.experimental.from_list(test_captions)
```

Μετά των κώδικα αυτό, μπορείτε να συνεχίσετε από το κελί

```
train_raw.element_spec
```

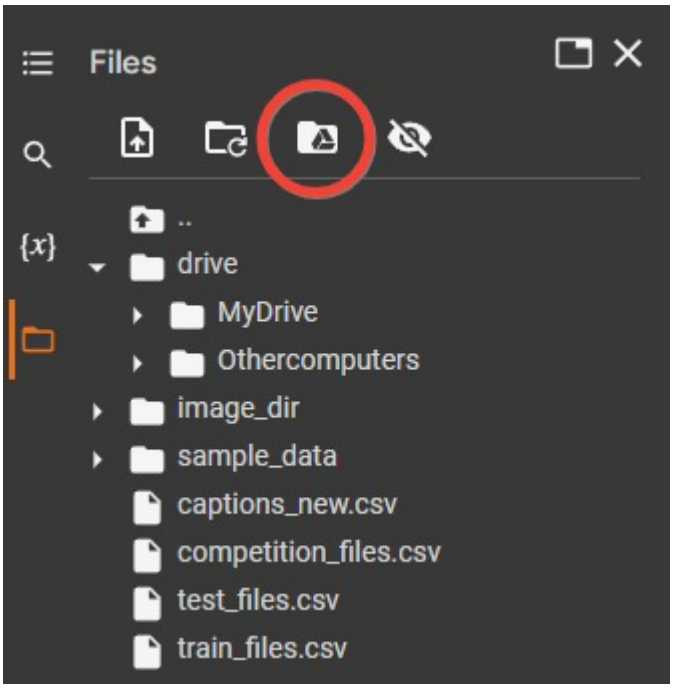
του παραδείγματος, δουλεύοντας πλέον με το δικό μας dataset.

Google Drive mount (προαιρετικό)

Επειδή το κατέβασμα των εικόνων παίρνει κάποια λεπτά μια καλή ιδέα θα ήταν να τις έχετε μόνιμα σε ένα directory στο Google Drive που θα κάνετε mount μέσα στο notebook.

1. Φτιάξτε ένα καινούριο λογαριασμό Gmail για να έχετε χώρο.
2. Κατεβάστε τις εικόνες, κάντε τες unzip και ανεβάστε το folder "image_dir" στο Google Drive.
3. Εντός του notebook, χρησιμοποιείστε το button στο αριστερό sidebar για να εισάγετε τον κώδικα που χρειάζεται να τρέξετε για να κάνει mount το Google Drive. Το Drive θα εμφανιστεί στο sidebar αν κάνετε ένα refresh. Με hover πάνω στα αρχεία και τους φακέλους, εμφανίζεται ένα hamburger button που σας επιτρέπει να αντιγράψετε το path τους.

Google Drive mount



4. Αν το "image_dir" είναι στο root του Drive τότε μπορείτε να τροποποιήσετε τις προηγούμενες μεταβλητές `train_captions` και `test_captions` ως εξής:

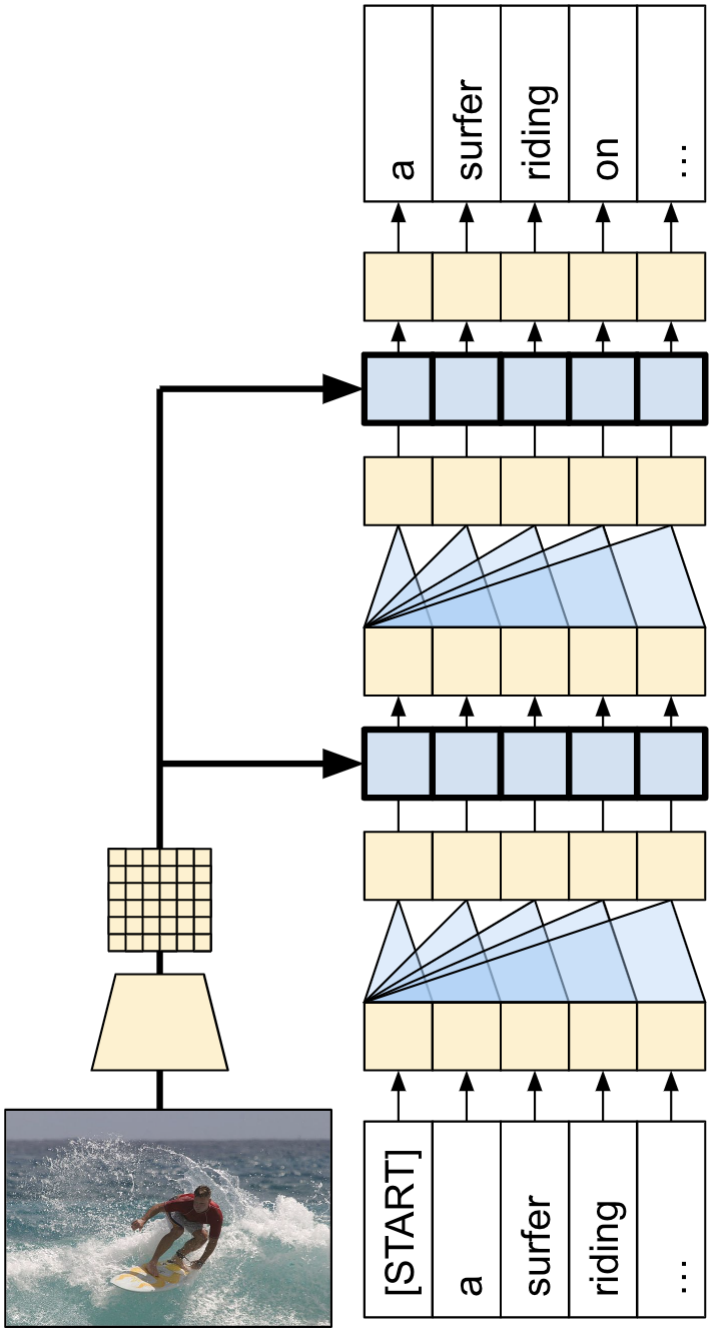
```
new_path = '/content/drive/MyDrive'
new_path = pathlib.Path(new_path)

train_captions = [(str(new_path/IMAGE_DIR/fname), cap_dict[fname]) for fname in
train_files]
test_captions = [(str(new_path/IMAGE_DIR/fname), cap_dict[fname]) for fname in
test_files]
```

Πλέον θα αρκεί να είναι mounted το Drive για να μπορούμε να διαβάσουμε τις εικόνες. Παρόμοια μπορείτε αν θέλετε να κάνετε και για τα υπόλοιπα αρχεία.

Το μοντέλο

Το μοντέλο βασίζεται στην γενική αρχιτεκτονική των μετασχηματιστών. Ένα συνελικτικό δίκτυο χρησιμοποιείται ως encoder της οπτικής πληροφορίας και μια σειρά από επίπεδα transformer-decoder παράγουν την λεκτική περιγραφή. Τα επίπεδα του transformer-decoder περιλαμβάνουν και επίπεδα προσοχής (attention).

Transformer-decoder architecture



Για την εξοικείωσή σας με τις αρχιτεκτονικές αυτές μπορείτε να διαβάσετε τα tutorials του TensorFlow Text generation, sequence-to-sequence, και Transformers.

Στη συνέχεια μπορείτε να τρέξετε όλο το notebook για να δείτε τί κάνει συνολικά.

Αξιολόγηση της ποιότητας του captioning (BLEU)

Το tutorial δεν περιλαμβάνει κάποια αναφορά στην ποιότητα του παραγόμενου captioning. Αν θεωρήσουμε ότι κάθε εικόνα έχει κάποια αληθινά captions (references) και το νευρωνικό παράγει ένα δικό του caption (hypothesis) θα χρησιμοποιήσουμε το BLEU (Bilingual Evaluation Understudy) score, μεταξύ hypothesis και references. Συνοπτικά, το BLEU είναι ένας σταθμισμένος μέσος όρος του πλήθους των κοινών unigrams, bigrams, trigrams, και fourgrams μεταξύ hypothesis και references. Το χειρότερο captioning λαμβάνει 0 και το καλύτερο 1. Δείτε ένα αναλυτικό παράδειγμα υπολογισμού του BLEU εδώ.

Για να παραχθεί ένα caption για μια εικόνα, πρώτα την φορτώνουμε με τη μέθοδο `image = load_image(image_path)` και στη συνέχεια καλούμε τη μέθοδο `model.simple_gen(image)`.

Η NLTK στο nltk.translate.bleu_score παρέχει τις απαραίτητες συναρτήσεις για τον υπολογισμό των BLEU scores:

- Για να μπορείτε να αξιολογείτε το captioning ενός μεμονωμένου παραδείγματος φτιάξτε μια συνάρτηση που να υπολογίζει την `sentence_bleu` μεταξύ hypothesis και αληθινών captions (references) για μια εικόνα.
- Για να αξιολογείτε το captioning περισσότερων εικόνων πχ ενός μέρους ή όλου του test set φτιάξτε συνάρτηση που θα υπολογίζει την `corpus_bleu` μεταξύ όλων των hypotheses και references των εικόνων που του δίνονται. Σημειώστε ότι το `corpus_bleu` δεν είναι μέσος όρος των `sentence_bleu`.

Σε όλες τις περιπτώσεις χρησιμοποιήστε τις εξής παραμέτρους: `weights=(0.4, 0.3, 0.2, 0.1)` και `smoothing_function=SmoothingFunction().method1`.

Πριν τον υπολογισμό, αφαιρείτε πάντοτε από τα references την τελική τελεία (το τελευταίο στοιχείο της λίστας).

Βελτιώσεις (και παραδοτέα)

Το πρώτο παραδοτέο είναι οι συναρτήσεις για τα BLEU scores της προηγούμενης παραγράφου.

Στη συνέχεια, εφόσον αποκτήσετε μια εικόνα της επίδοσης του default δικτύου (loss, accuracy, plots, χρόνοι εκπαίδευσης, BLEU scores) θα δοκιμάσουμε κάποιες ιδέες για βελτιώσεις στο δίκτυο (λοιπά παραδοτέα).

Encoder

Το παράδειγμα χρησιμοποιεί για encoder με μεταφορά μάθησης το MobileNetV3Small. Στα έτοιμα μοντέλα CNN του Keras υπάρχουν μοντέλα που εμφανίζονται να έχουν καλύτερες benchmark επιδόσεις από το MobileNet (V2).

Κρατήστε σταθερή τη διαδικασία train του decoder και δείτε αν μπορείτε να πάρετε καλύτερες επιδόσεις χρησιμοποιώντας άλλο συνελικτικό για encoder.

Όταν καταλήξετε στον encoder μπορείτε να χρησιμοποιήσετε τη δυνατότητα caching των features που περιέχεται στο notebook ώστε να τα διαβάζετε / σώζετε από και προς το Drive αντί να τα παράγετε κάθε φορά.

Προεπεξεργασία κειμένου

1. Τα captions έχουν διαφορετικά μήκη. Ενδεχομένως τα πολύ σύντομα και τα πολύ εκτενή να μην είναι χρήσιμα στην εκπαίδευση. Στο tutorial χρησιμοποιείται ad-hoc ένα μέγιστο μήκος 50 λέξεων. Μπορείτε να φιλτράρετε το dataset έτσι ώστε να έχετε ένα range από διαφορετικά μήκη, χωρίς ούτε τα πολύ μικρά και χωρίς ούτε τα πολύ μεγάλα (δοκιμάστε ένα ιστόγραμμα). Θα πρέπει να προσαρμόσετε αναλόγως και την `max_length`.
2. Μελετήστε τα captions. Θα μπορούσε η συνάρτηση `standardize` να περιλαμβάνει και άλλα φίλτρα κανονικοποίησης; Προσοχή, εδώ δεν κάνουμε stemming.

3. Το tutorial αποφασίζει ad hoc για ένα vocabulary 5000 λέξεων. Δοκιμάστε και κάποια διαφορετικά (μικρότερα ή μεγαλύτερα) μεγέθη vocabulary.

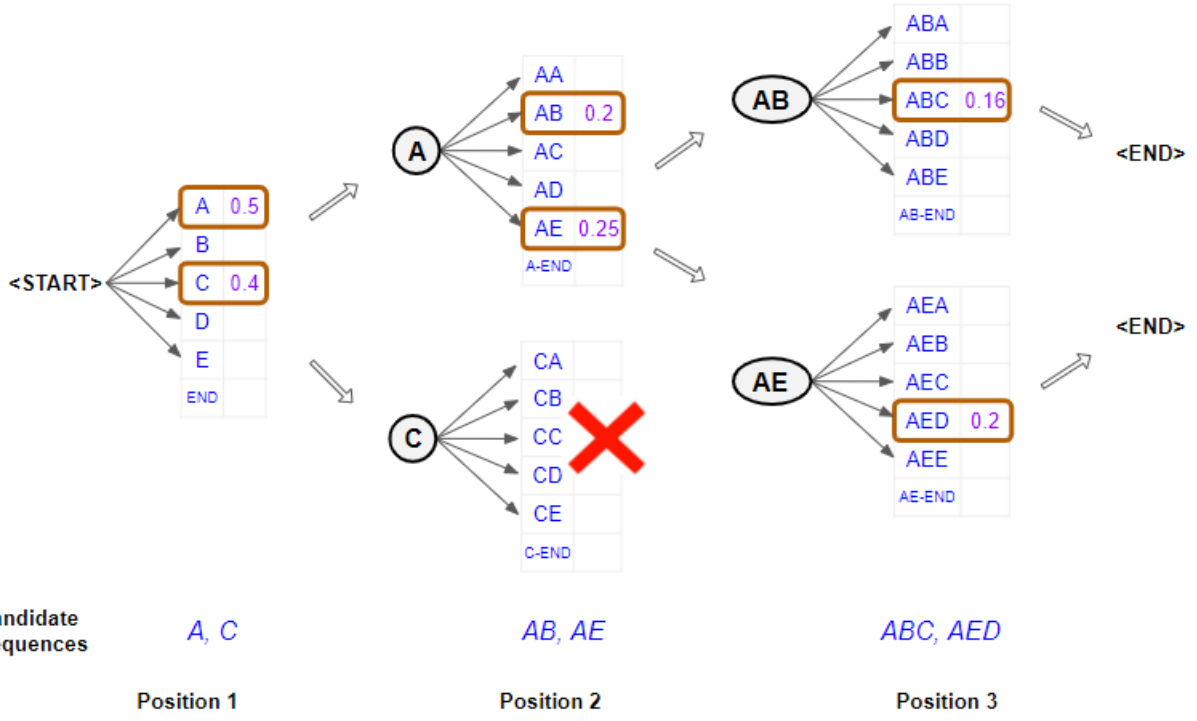
Embeddings

Στο παράδειγμα του tutorial τα embeddings μαθαίνονται κατά την εκπαίδευση του μοντέλου. Αντί αυτού μπορούμε να χρησιμοποιήσουμε έτοιμα embeddings με μεταφορά μάθησης. Δείτε πώς αποδίδει το δίκτυο για παράδειγμα με τα embeddings glove-wiki του Gensim διαφόρων διαστάσεων (50, 100, 200, 300).

Sentence Generator (Beam Search)

Στο παράδειγμα χρησιμοποείται η μεταβλητή της "θερμοκρασίας" (`temperature`) για να παράγονται οι προτάσεις. Αν η θερμοκρασία είναι 0.0, έχουμε greedy decoding και επιλέγεται το πιο πιθανό token σε κάθε βήμα. Αν η θερμοκρασία είναι 1.0 κάνει τυχαίο sampling ως προς τις πιθανότητες (logits) του κάθε token. Αν η θερμοκρασία είναι πολύ μεγαλύτερη του 1.0 καταλήγουμε σε ομοιόμορφο τυχαίο sampling.

Beam Search sentence generation



Στη βιβλιογραφία ωστόσο αναφέρεται ότι η μέθοδος Beam Search παράγει σημαντικά καλύτερα αποτελέσματα. Η Beam Search έχει μια υπερπαράμετρο που είναι το πλάτος της ακτίνας b. Για να διαλέξει την επόμενη λέξη ξεκινάει από την αρχή της πρότασης `[START]` και κρατάει τις b το πλήθος καλύτερες (πιθανότερες) λέξεις για το επόμενο βήμα. Με τον τρόπο αυτό δημιουργούντα b κλαδιά. Για το επόμενο βήμα υπολογίζει τις πιθανότητες των επόμενων λέξεων για όλα τα κλαδιά και κρατάει τις b πιθανότερες κοκ. Στο τέλος `[END]` καταλήγουμε να έχουμε b πιθανές προτάσεις και διαλέγουμε αυτή με τις καλύτερες πιθανότητες συνολικά. Για το τελευταίο, θα μπορούσαμε για παράδειγμα να χρησιμοποιήσουμε το Σ του log των πιθανοτήτων δια του μήκους της κάθε πρότασης.

Θα βρείτε στο διαδίκτυο έτοιμες υλοποιήσεις για το Beam Search που μπορείτε να προσαρμόσετε, όπως επίσης και συνήθεις τιμές για το b. Μπορείτε να τροποποιήσετε τα ορίσματα της `simple_gen` ώστε να καλύπτει και το Beam Search.

Υπερπαράμετροι του decoder

Μελετήστε την επίδραση των υπερπαραμέτρων `units`, `dropout_rate`, `num_layers`, και `num_heads` του μοντέλου.

Οι χρόνοι εκπαίδευσης μπορεί να είναι σημαντικοί. Χρησιμοποιήστε κάποια μέθοδο αποθήκευσης του μοντέλου ώστε να μπορείτε να συνεχίζετε την εκπαίδευση.

Προφανώς δεν μπορούμε να κάνουμε cross-validation των παραμέτρων και αρχιτεκτονικών. Χρησιμοποιείτε σταδιακό training (σχετικά λίγα epochs) και εκτίμηση της απόδοσης του δικτύου με βάση το loss και τον απαιτούμενο χρόνο, ενώ παράλληλα εξετάζετε την ποιότητα του captioning με την `sentence_bleu` για επιλεγμένες εικόνες και κυρίως την `corpus_bleu` για ένα κομμάτι του test set.

Παραδοτέο

Το notebook με το καλύτερο μοντέλο σας ως προς το `corpus_bleu` σε τουλάχιστον 1000 εικόνες από το test set. Μπορείτε να δοκιμάσετε και σε περισσότερες εικόνες, ιδανικά σε όλο το test set, απλά είναι θέμα χρόνου η παραγωγή των captions.

Χρησιμοποιήστε markdown για να εξηγήσετε τις επιλογές σας. Μπορείτε να σημειώσετε τιμές του BLEU και για ενδιάμεσες επιλογές και να τις παρουσιάσετε σε πίνακα. Δώστε παραδείγματα εικόνων με επιτυχημένα και λιγότερο επιτυχημένα captioning.

Διαγωνισμός (προαιρετικός)

Η συμμετοχή στο διαγωνισμό είναι προαιρετική και δεν έχει καμία βαθμολογική σημασία. Ωστόσο, επειδή η κατάταξη γίνεται με βάση το `corpus_bleu`, η συμμετοχή και σύγκριση με άλλες υποβολές μπορεί να σας δείχνει πόσο έχετε προχωρήσει στη βελτίωση των captions σας.

Εικόνες διαγωνισμού

Για τον διαγωνισμό έχουμε επιλέξει 500 εικόνες για τις οποίες δεν έχετε καθόλου captions. Τα ονόματά τους περιλαμβάνονται στο "competition_files.csv". Μπορείτε να τις διαβάσετε με τον ακόλουθο κώδικα:

```
# Download competition files list
competition_files_list = tf.keras.utils.get_file('competition_files.csv',
                                                 cache_subdir=os.path.abspath('.'),
                                                 origin='https://spartacus.1337.cx/flickr-
mod/competition_files.csv',
                                                 extract=False)

path="."
IMAGE_DIR="image_dir"
path = pathlib.Path(path)

competition_files = (path/competition_files_list).read_text().splitlines()
competition_files = [str(path/IMAGE_DIR/fname) for fname in competition_files]
```

Μην ξεχάσετε να αλλάξετε τα paths για τα αρχεία των εικόνων αν χρησιμοποιείτε το Google Drive.

Δημιουργία αρχείου υποβολής

Για όλες τις εικόνες του "competition_files.csv" και με την σειρά που έχουν οι εικόνες (κάνοντας append δηλαδή) δημιουργήστε captions με τη μορφή λιστών. Αν ας πούμε είχαμε δύο εικόνες θα αποθηκεύαμε σε μια μεταβλητή `test_hypotheses` τα captions ως εξής:

```
[['a', 'woman', 'floats', 'with', 'her', 'face', 'out', 'of', 'water', 'in',
'a', 'pool', 'with', 'another', 'woman', 'nearby', 'posing', 'for', 'the',
'camera'], ['a', 'black', 'and', 'white', 'dog', 'walks', 'along', 'a', 'sandy',
'beach']]
```

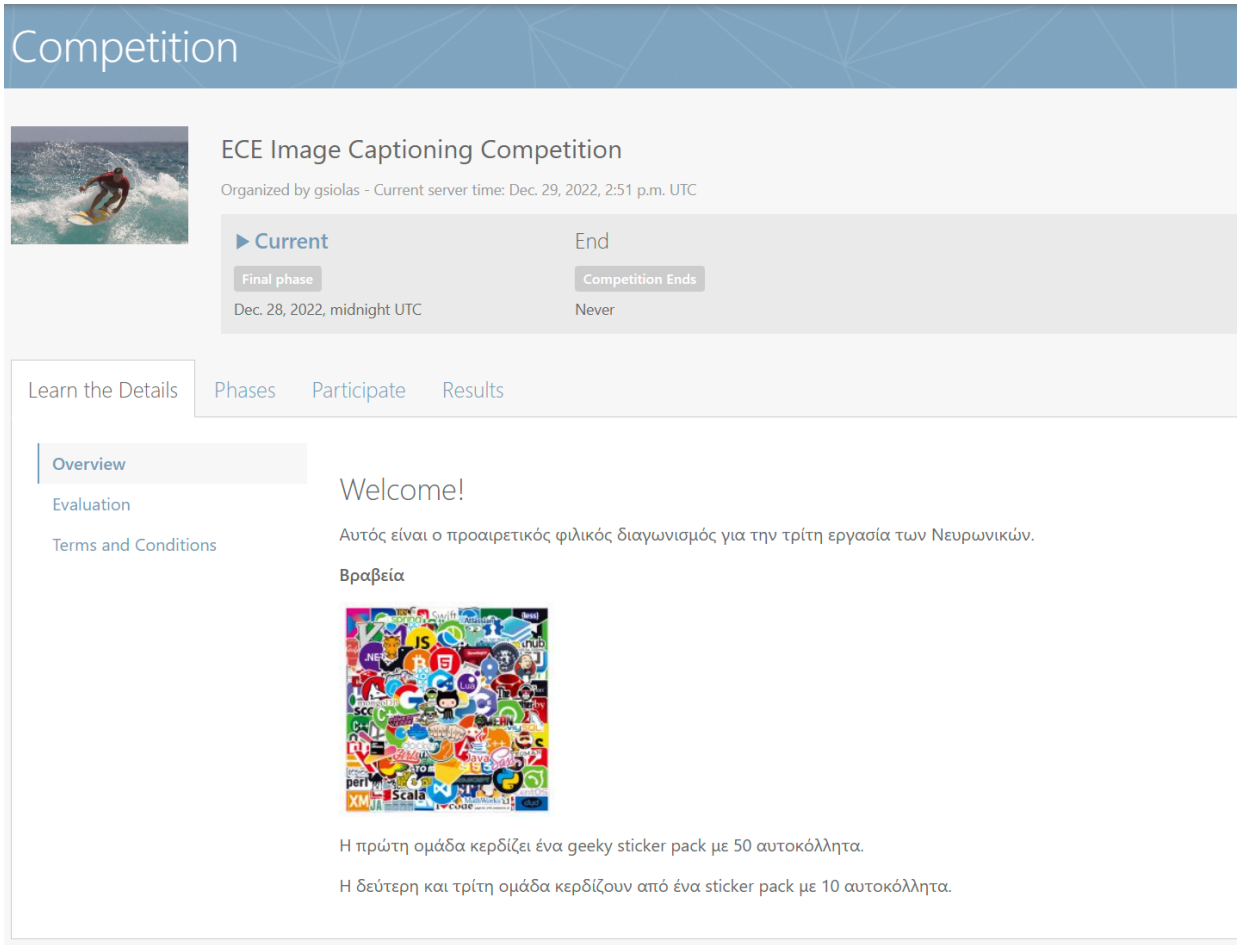
Υπολογίστε τα 500 captions στη μεταβλητή test_hypotheses και αποθηκεύετε την ως JSON ως εξής:

```
import json

jsonString = json.dumps(test_hypotheses)
jsonFile = open("test_hypotheses.json", "w")
jsonFile.write(jsonString)
jsonFile.close()
```

Κατεβάζετε τοπικά το αρχείο test_ypotheses.json, το μετονομάζετε υποχρεωτικά σε answer.txt και το ζιράρετε σε zip file το όνομα του οποίου δεν έχει σημασία.

Συνημμένο στην εκφώνηση θα βρείτε ένα παράδειγμα λειτουργικού answer.txt.

Το site του διαγωνισμού στο Codalab



Ο διαγωνισμός βρίσκεται στο σύνδεσμο αυτό προς το Codalab. Χρησιμοποιήθηκε το Codalab αντί του Kaggle γιατί το Kaggle στο community tier δεν επιτρέπει custom μετρικές όπως η BLEU. Το Codalab είναι επίσης το επίσημο αποθετήριο του COCO.

Θα πρέπει να κάνετε register. Σημειώστε ότι αν πάτε στα settings στο profile σας μπορείτε να ορίσετε όνομα ομάδας και μέλη της με τα username όλων σας στο Codalab. Προφανώς το όνομα της ομάδας μπορεί να είναι ό,τι θέλετε, άσχετο με τον αριθμό στο εργαστήριο.

Υποβολή απάντησης και leaderboard

Προκειμένου να ανεβάσετε μια απάντηση, θα πρέπει να πάτε στο tab "Participate", στο "Submit/View Results" και να πατήσετε "Submit" για να ανεβάσετε το αρχείο υποβολής. Μπορείτε να κάνετε refresh της κατάστασης της υποβολής μέσω του διαθέσιμου button "Refresh Status". Όταν η υποβολή ολοκληρωθεί, και εφόσον είναι έγκυρη, κάντε refresh την σελίδα μέσω του browser για να δείτε το score σας. Εάν το score σας σας ικανοποιεί, μπορείτε να κάνετε "Submit to leaderboard".

Στο tab "Results" θα βρείτε το leaderboard των submissions όλων των ομάδων με βάση το `corpus_bleu`.