Image Captioning

By Spencer Au, Ethan Tarnarider, Daniel Boudagian

Overview

- Generating Captions given an image
  - Using the Microsoft COCO (common objects in context) image dataset
    - Essentially pictures of all sorts of everything day things
      - People, cars, beaches, snow etc, with captions for every image
- We accomplish this by feeding the images into a pre-trained CNN, efficientnetB0 with no head, solely for feature extraction, then we feed the images, along with their captions into an LSTM for training on captioning any image
- The final result is 10 random captioned images from the dataset

Dataset

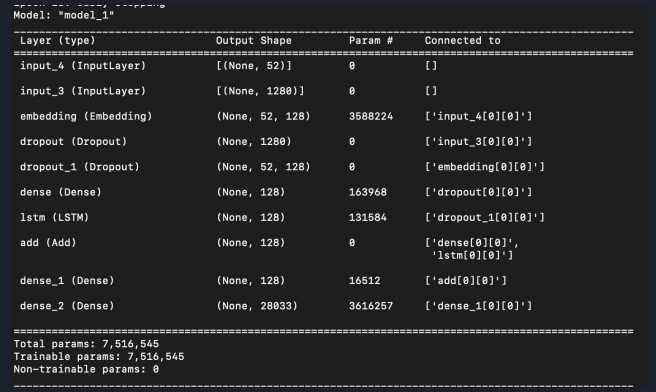
- Trained on MS-COCO (Common Objects in Context) 2017 25GB dataset
- 330K images (>200K labeled)
- 1.5 million object instances
- 80 object categories
- 91 stuff categories
- 5 captions per image
- Reshaped each image to 224 x 224 pixels
- Can’t really use image augmentation due to the task of image captioning

CNN for Feature Extraction

- Uses EfficientNetB0
  - Without top layer
  - Using imagenet weights
  - Using average pooling
- Primarily chosen due to how lightweight the model is
  - When processing hundreds of thousands of images we looked for a solid mix between speed and accuracy

Model Architecture

- We used an LSTM to take in the results of the CNN, along with the captions
  - We used an LSTM specifically because it excels at sequences and has the short term memory
    - We thought this would be best for generating our captions
- We used layers such as embedding, dropout and dense layers to drive the model
  - Embedding to reduce dimensions of input while keeping the same meanings and mappings
  - Dropout to regularize
  - And dense layer to select proper words



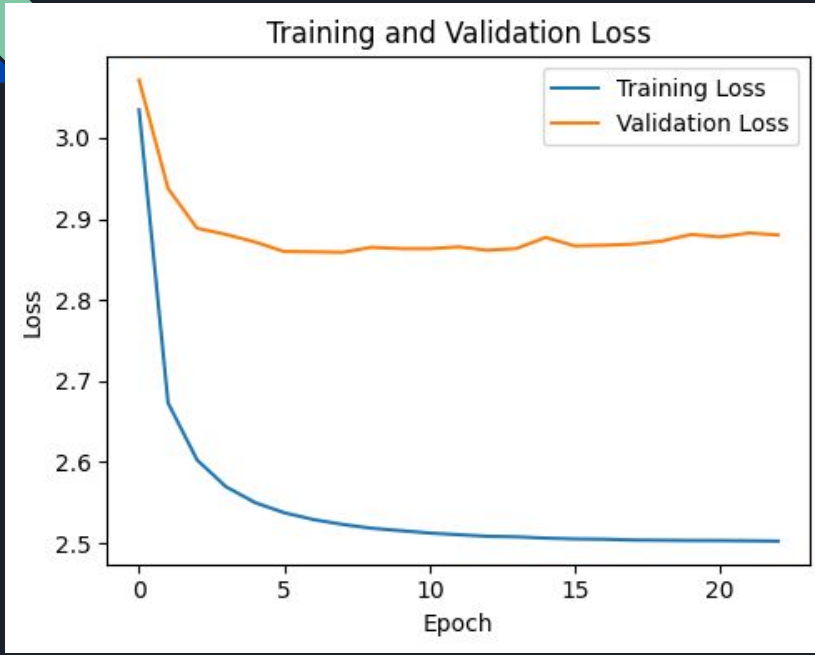
```
Model: "model_1"
__________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
==================================================================================================
 input_4 (InputLayer)        [(None, 52)]              0         []

 input_3 (InputLayer)        [(None, 1280)]            0         []

 embedding (Embedding)       (None, 52, 128)           3588224   ['input_4[0][0]']

 dropout (Dropout)           (None, 1280)              0         ['input_3[0][0]']

 dropout_1 (Dropout)         (None, 52, 128)           0         ['embedding[0][0]']

 dense (Dense)               (None, 128)               163968    ['dropout[0][0]']

 lstm (LSTM)                 (None, 128)               131584    ['dropout_1[0][0]']

 add (Add)                   (None, 128)               0         ['dense[0][0]',
                                                                  'lstm[0][0]']

 dense_1 (Dense)             (None, 128)               16512     ['add[0][0]']

 dense_2 (Dense)             (None, 28033)             3616257   ['dense_1[0][0]']

==================================================================================================
Total params: 7,516,545
Trainable params: 7,516,545
Non-trainable params: 0
__________________________________________________________________________________________________
```

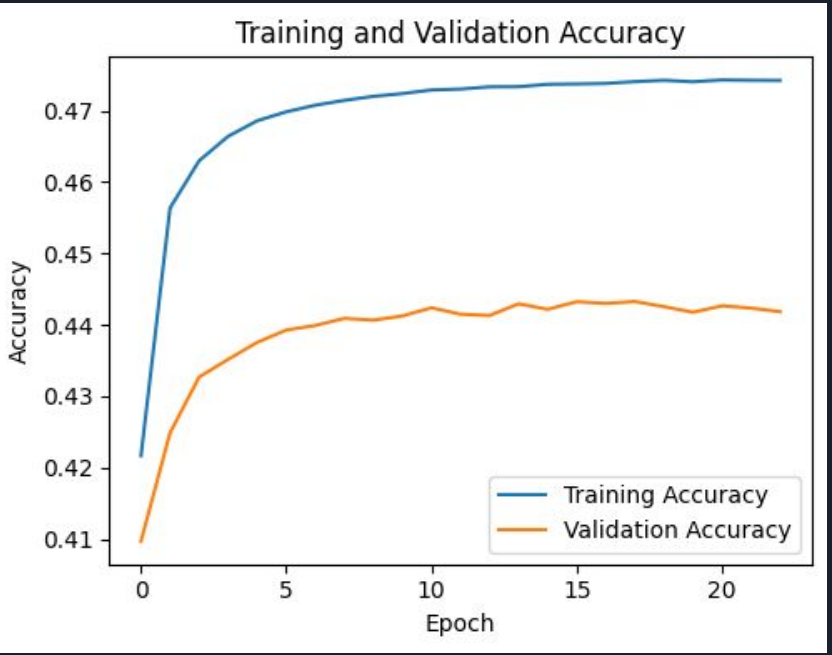
```
Epoch 15: early stopping
Model: "model_1"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_4 (InputLayer)           [(None, 52)]         0           []

 input_3 (InputLayer)           [(None, 1280)]       0           []

 embedding (Embedding)          (None, 52, 128)      3588224     ['input_4[0][0]']

 dropout (Dropout)              (None, 1280)         0           ['input_3[0][0]']

 dropout_1 (Dropout)            (None, 52, 128)      0           ['embedding[0][0]']

 dense (Dense)                  (None, 128)          163968      ['dropout[0][0]']

 lstm (LSTM)                    (None, 128)          131584      ['dropout_1[0][0]']

 add (Add)                      (None, 128)          0           ['dense[0][0]',
                                                                  'lstm[0][0]']

 dense_1 (Dense)                (None, 128)          16512       ['add[0][0]']

 dense_2 (Dense)                (None, 28033)        3616257     ['dense_1[0][0]']

==================================================================================================
Total params: 7,516,545
Trainable params: 7,516,545
Non-trainable params: 0
__________________________________________________________________________________________________
```

Training

- Loss Function - Categorical Cross Entropy
- Optimizer - Adam with default initial learning rate
- Metrics - accuracy
- Batch Size of 10
- Vocab Size of 15,000
- Sequence Max Length of 52
- Utilizing Early Stopping with Patience of 5
- Train for 100 Epochs
- Stopped at 22 Epochs

Results


Training and Validation Loss
Training Loss
Validation Loss
Loss
3.0
2.9
2.8
2.7
2.6
2.5
0
5
10
15
20
Epoch


Training and Validation Accuracy
Accuracy
0.47
0.46
0.45
0.44
0.43
0.42
0.41
Training Accuracy
Validation Accuracy
0
5
10
15
20
Epoch

Generated Captions

a cat is laying on a white table

a large bear is sitting on a rock

a pizza that is on a table with a fork

man is holding a large white teddy bear

a man is holding a pair of scissors

a large white and white clock sitting on top of a car

Proposed Enhancements

- GradCAM or possibly n-Best/n-Worst for Model Interpretability
- Use a performance metrics better suited like BLEU, METEOR, or CIDEr
- Experiment with other base models for CNN portion
- Experiment with different regularization methods for slight overfitting
- Experimenting with different architecture layers for Caption Generation Model
- Hyperparameter Tuning via Keras Tuner, etc
- Adding object detection to avoid false mentions of objects
- Figure out why certain words seem to show up in almost every second or third caption
    - Words such as scissors or clocks/clock tower seem to appear a lot despite being irrelevant in most images