What’s in an Image?

Michael Mulder

michaelhmulder@gmail.com
linkedin.com/in/mhmulder
github.com/mhmulder

Objective and Motivation

Many large companies are exploring augmented reality and using images in new ways. Walmart in particular is considering using this technology to enhance their home furnishings business. One goal is to be able to have a user upload an image from a room in their house and use that image to make recommendations for furniture they should add or swap out. **The goal of this project is to simply start by describing the object(s) in an image.** I accomplished this goal by using a pre-trained convolutional neural net (VGG16) and concatenating it to a long short term memory (LSTM) recurrent neural net that fed into another bidirectional LSTM in a seq2seq fashion.

Discussion

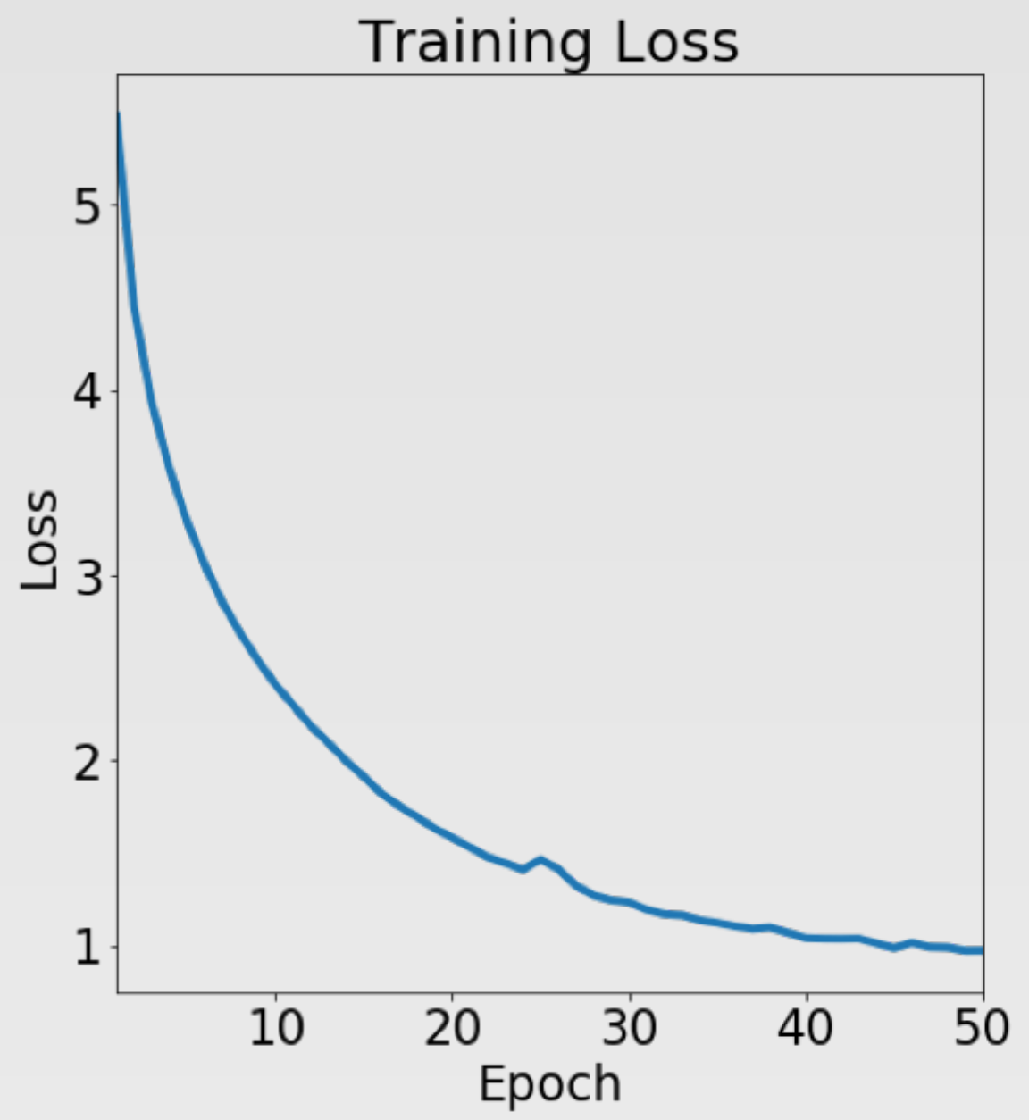
All data was scraped off of Walmart’s website. The scraped data contains 296,000 images, of more than 100,000 product, with over 5,500 unique words. For training the net, I randomly sampled 20,000 of the images. Using the net architecture and process described below I achieved successful results. **Actual predictions from the test set are shown at the bottom.**

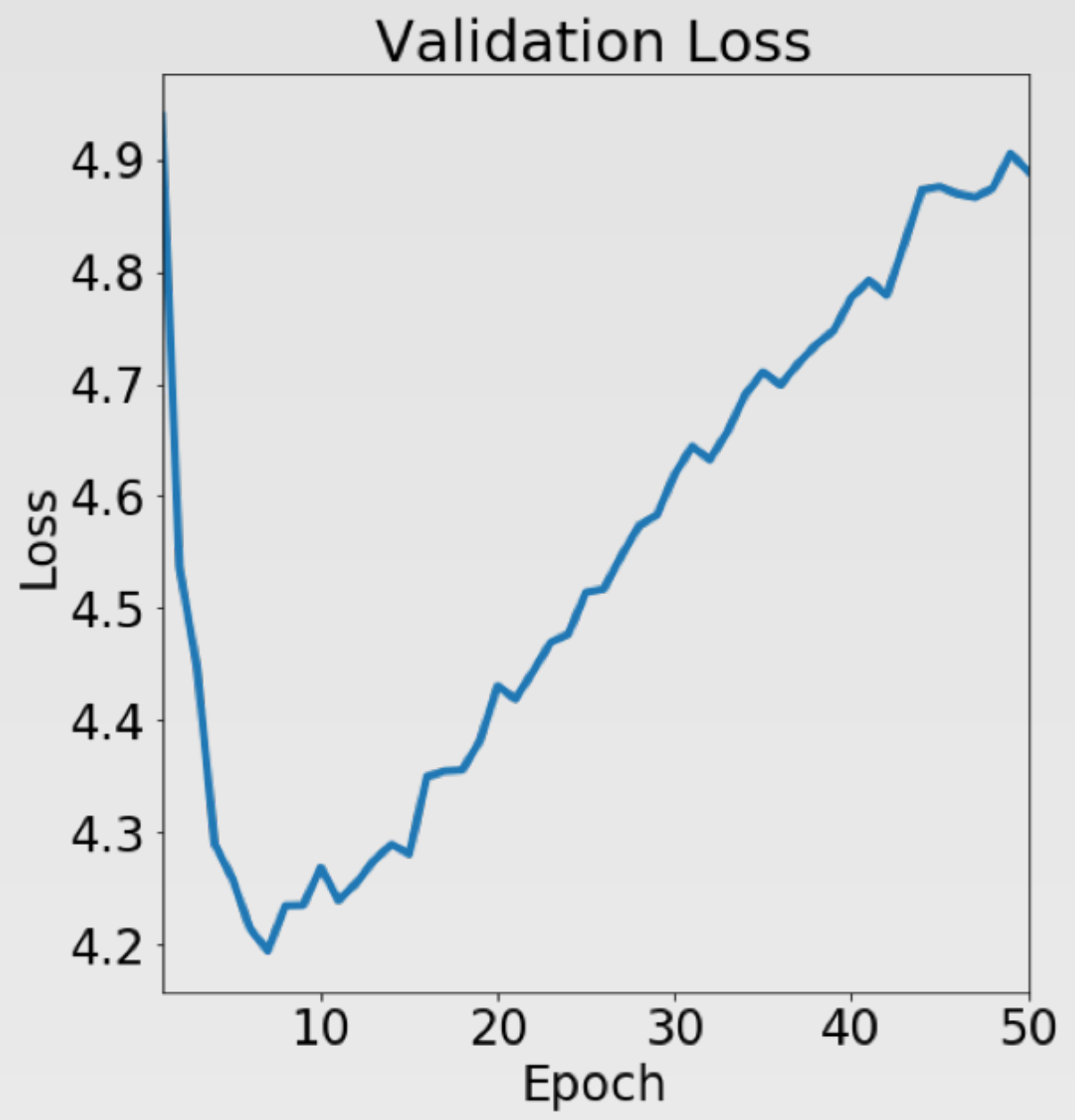
Results and Evaluations

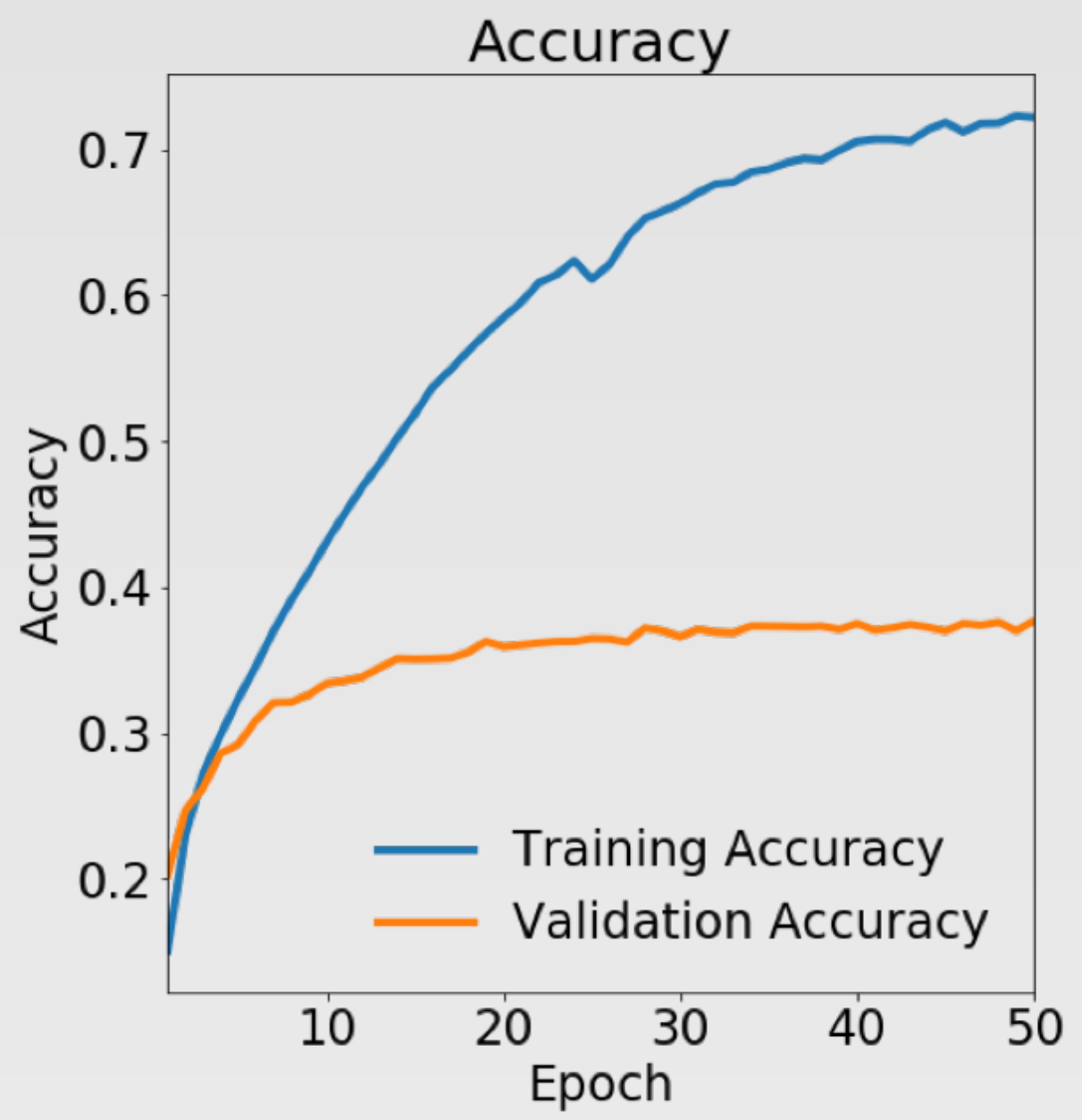
- **Evaluating the model results was the principle challenge**
- Used SpaCy (NLP engine) to clean and lemmatize words in descriptions and applied cosine similarity to score predictions
- Created a human click through evaluation to score quality of predicted description

Results and Evaluations Continued

After each epoch categorical cross-entropy loss and accuracy were calculated for both a training and validation set, as seen below. After about 9 epochs, the model began to overfit. Upon a secondary human evaluation, I decided to end the training there. **The final predictions were about 54% good descriptions, 22% ‘I can see it’, and 24% poor descriptions.**

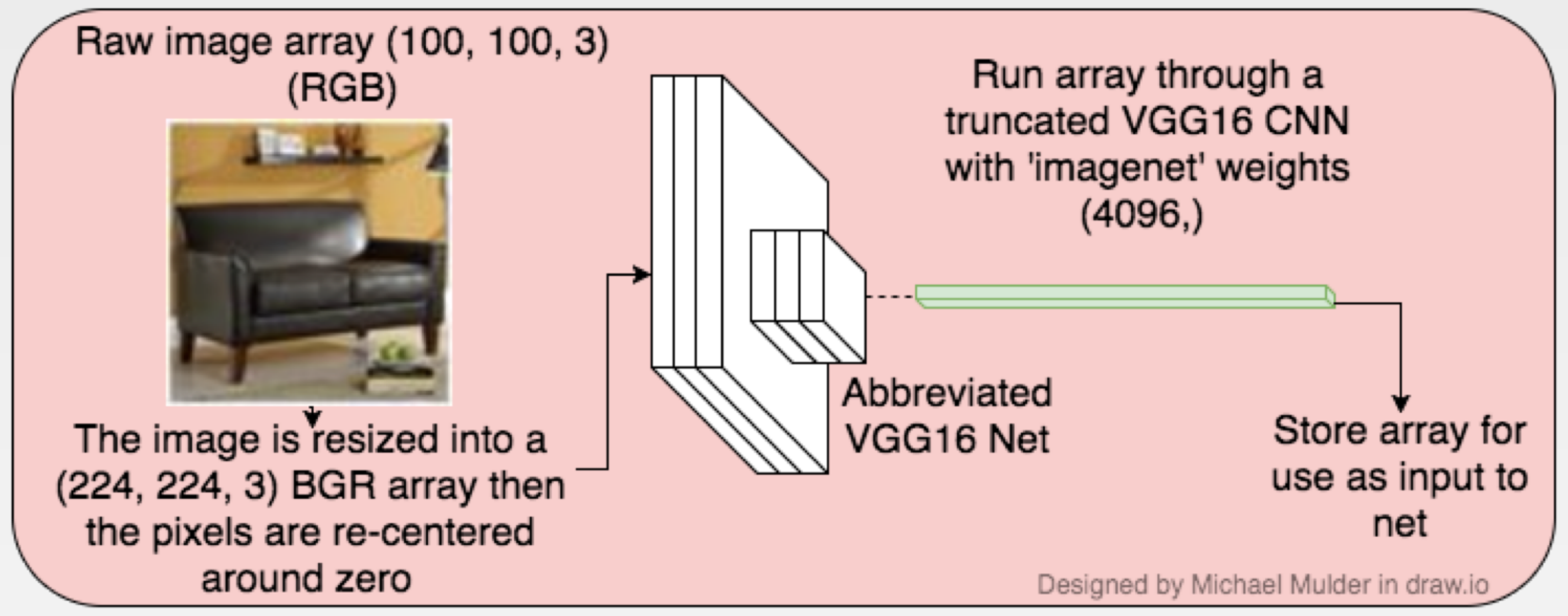






Processing the Data

Image Preprocessing



Raw image array (100, 100, 3) (RGB)

The image is resized into a (224, 224, 3) BGR array then the pixels are re-centered around zero

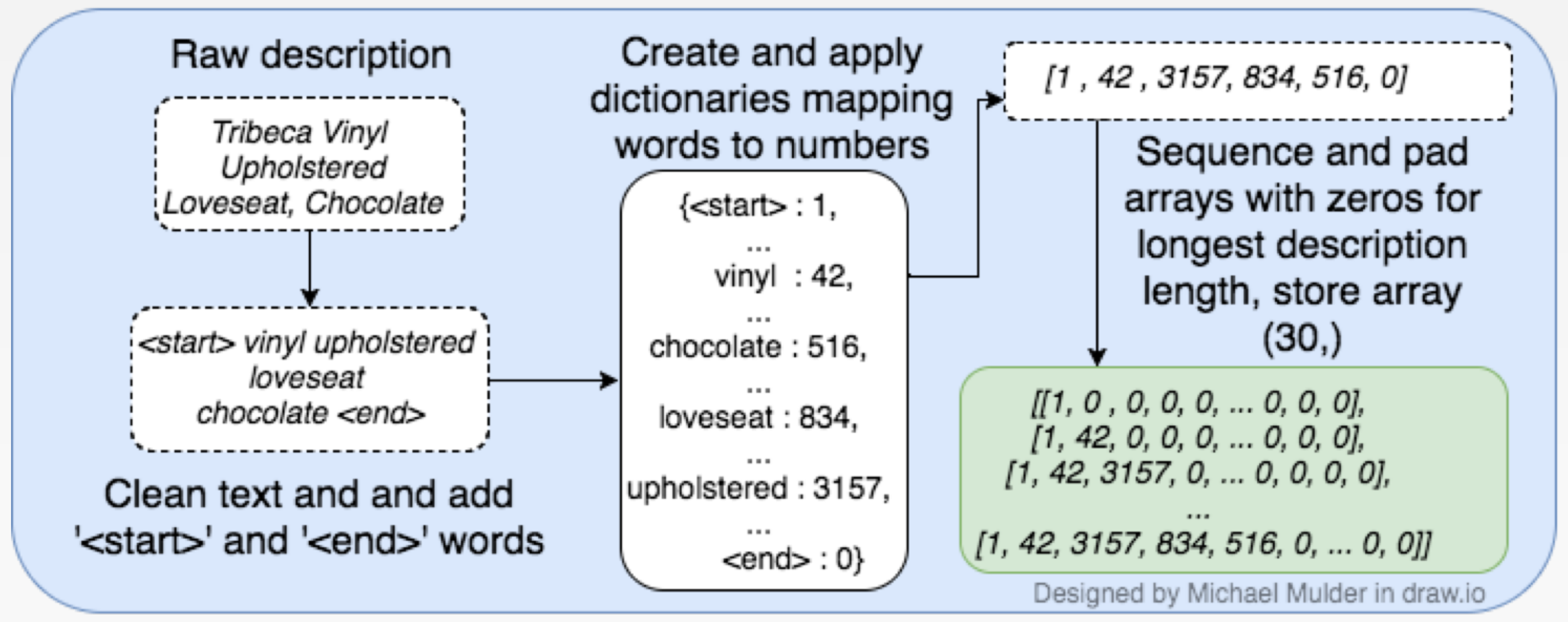
Run array through a truncated VGG16 CNN with 'imagenet' weights (4096,)

Abbreviated VGG16 Net

Store array for use as input to net

Designed by Michael Mulder in draw.io

Text Preprocessing



Raw description

Tribeca Vinyl Upholstered Loveseat, Chocolate

<start> vinyl upholstered loveseat chocolate <end>

Clean text and and add '<start>' and '<end>' words

Create and apply dictionaries mapping words to numbers

{<start> : 1,
...
vinyl : 42,
...
chocolate : 516,
...
loveseat : 834,
...
upholstered : 3157,
...
<end> : 0}

[1 , 42 , 3157, 834, 516, 0]

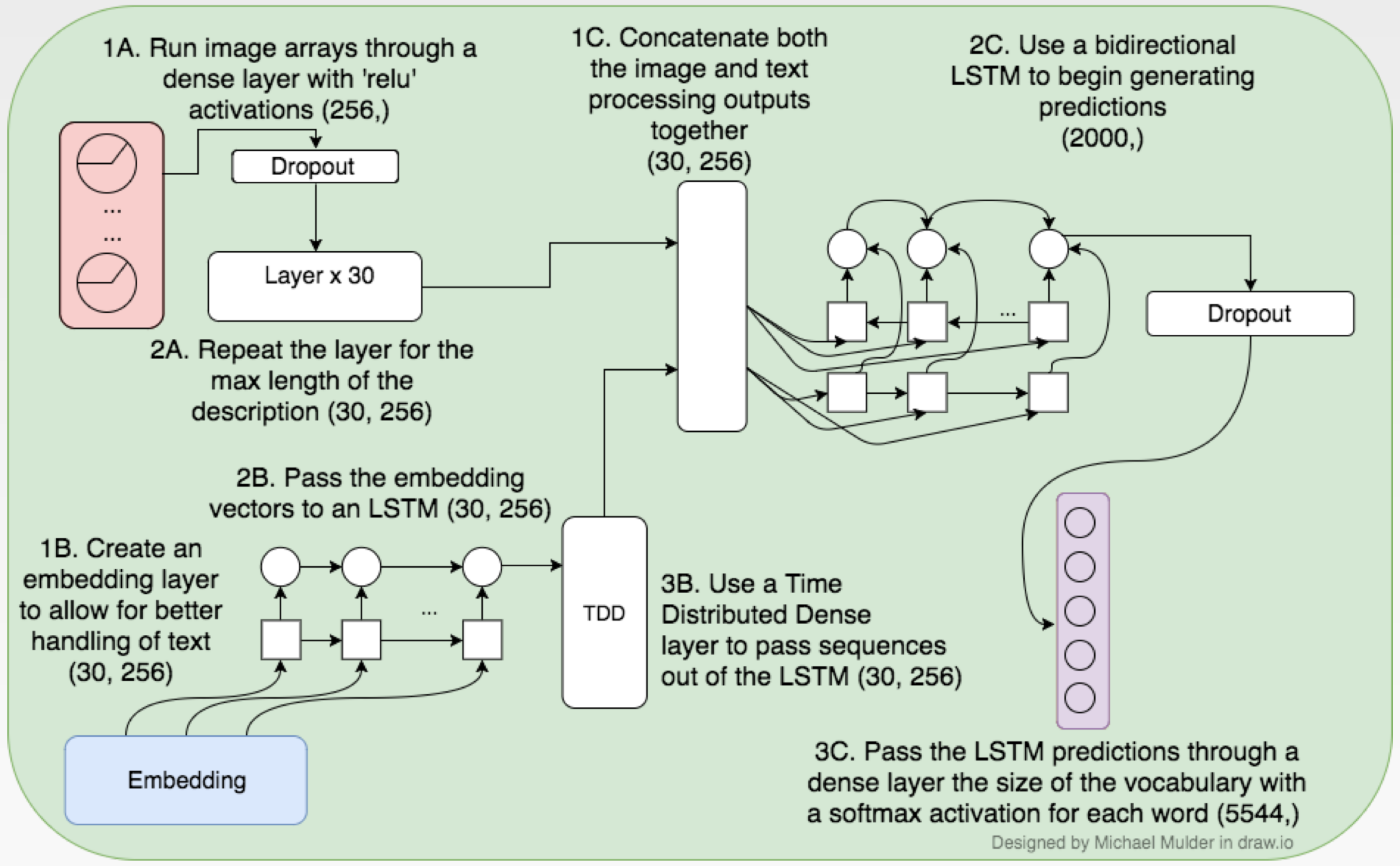
Sequence and pad arrays with zeros for longest description length, store array (30,)

*[[1, 0 , 0, 0, 0, ... 0, 0, 0],
[1, 42, 0, 0, 0, ... 0, 0, 0],
[1, 42, 3157, 0, ... 0, 0, 0, 0],
...
[1, 42, 3157, 834, 516, 0, ... 0, 0]]*

Designed by Michael Mulder in draw.io

Main Net Architecture

Net Architecture



1A. Run image arrays through a dense layer with 'relu' activations (256,)

Dropout

Layer x 30

2A. Repeat the layer for the max length of the description (30, 256)

1B. Create an embedding layer to allow for better handling of text (30, 256)

Embedding

2B. Pass the embedding vectors to an LSTM (30, 256)

1C. Concatenate both the image and text processing outputs together (30, 256)

2C. Use a bidirectional LSTM to begin generating predictions (2000,)

Dropout

TDD

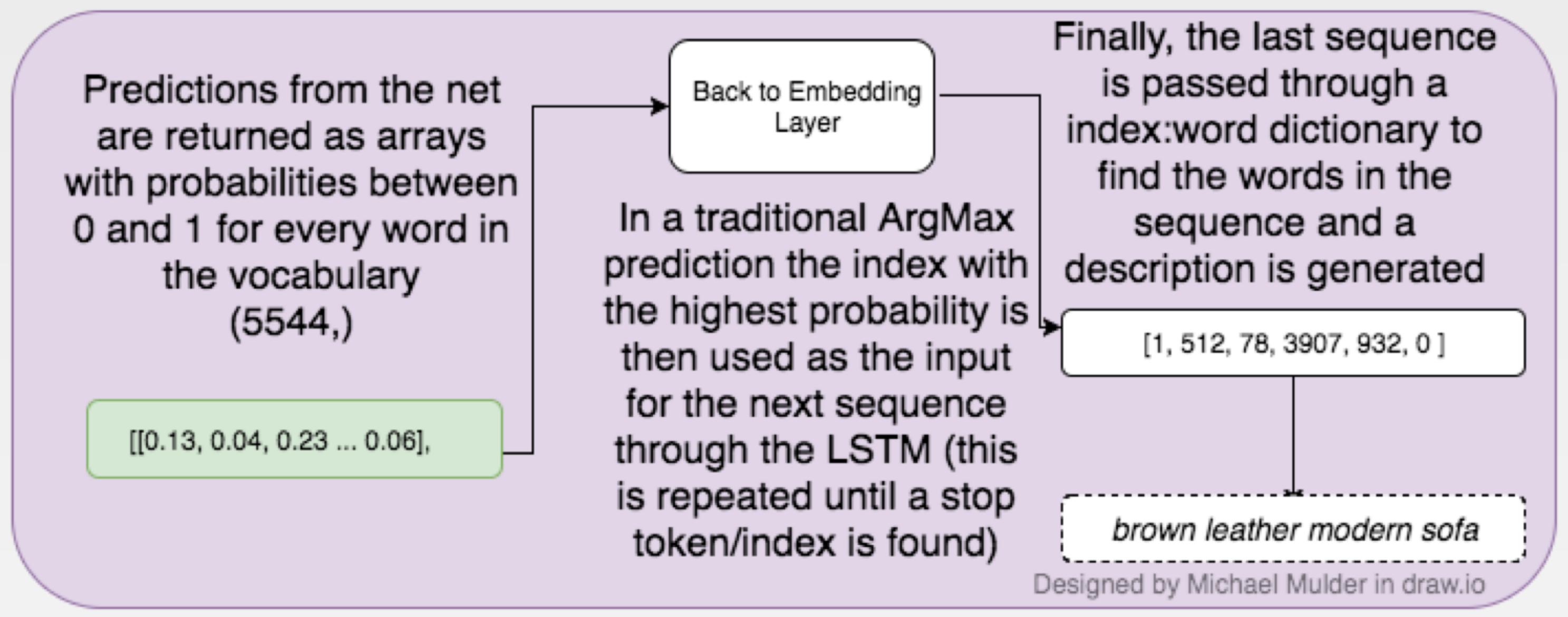
3B. Use a Time Distributed Dense layer to pass sequences out of the LSTM (30, 256)

3C. Pass the LSTM predictions through a dense layer the size of the vocabulary with a softmax activation for each word (5544,)

Designed by Michael Mulder in draw.io

Making Predictions

Prediction Generation



Predictions from the net are returned as arrays with probabilities between 0 and 1 for every word in the vocabulary (5544,)

[[0.13, 0.04, 0.23 ... 0.06],

Back to Embedding Layer

In a traditional ArgMax prediction the index with the highest probability is then used as the input for the next sequence through the LSTM (this is repeated until a stop token/index is found)

Finally, the last sequence is passed through a index:word dictionary to find the words in the sequence and a description is generated

[1, 512, 78, 3907, 932, 0]

brown leather modern sofa

Designed by Michael Mulder in draw.io

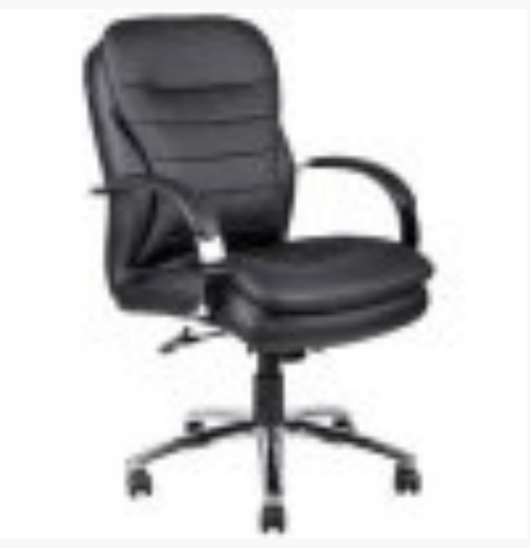
Other Applications

- Redefine how we search for and through images
- Object identification in videos
- Aid for the visually impaired
- Better sales recommendations
- Create new data from images

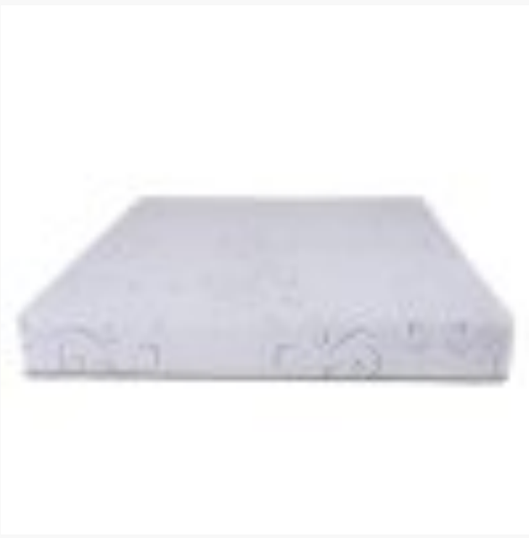
References

- Very Deep Convolutional Networks for Large-Scale Image Recognition, K. Simonyan, A. Zisserman, arXiv:1409.1556
- The Unreasonable Effectiveness of Recurrent Neural Networks, A. Karpathy
- Translation Modeling with Bidirectional Recurrent Neural Networks, M. Sundermeyer, T. Alkhouli
- Bidirectional Recurrent Neural Networks as Generative Models - Reconstructing Gaps in Time Series, M. Berglund, T. Raiko

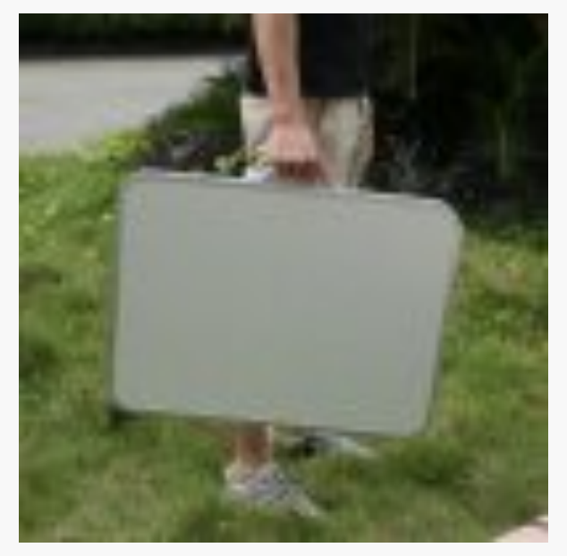
Great Predictions



black leather executive office chair brown

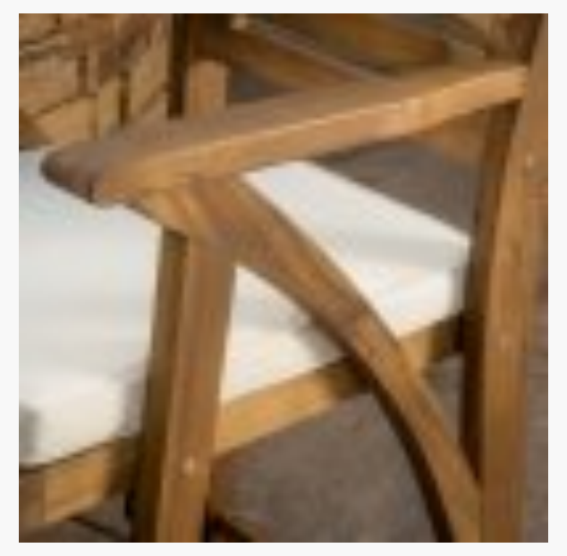


gel memory foam mattress multiple sizes

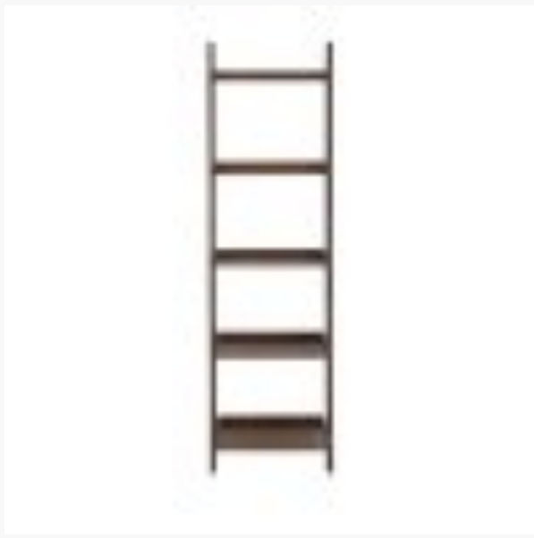


lightweight picnic table dining portable party indooroutdoor folding

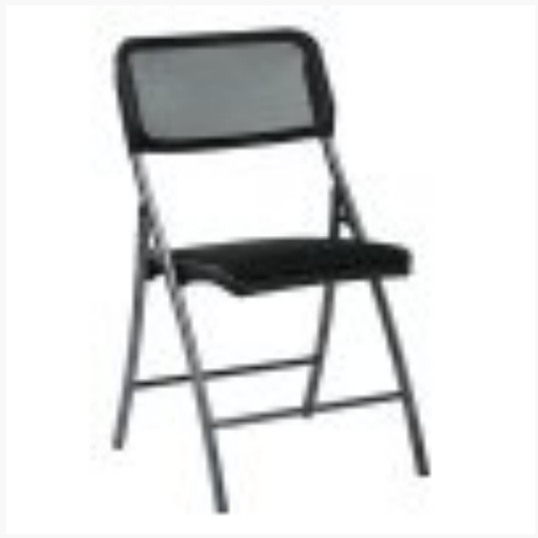
Interesting Predictions



abordale wood bar stool

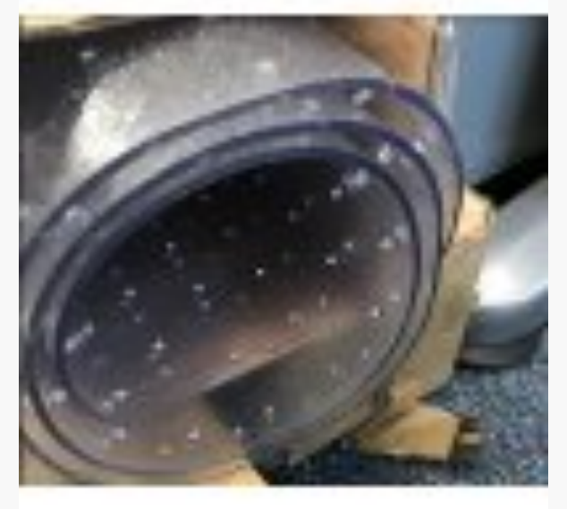


iron inch rectangular decmode

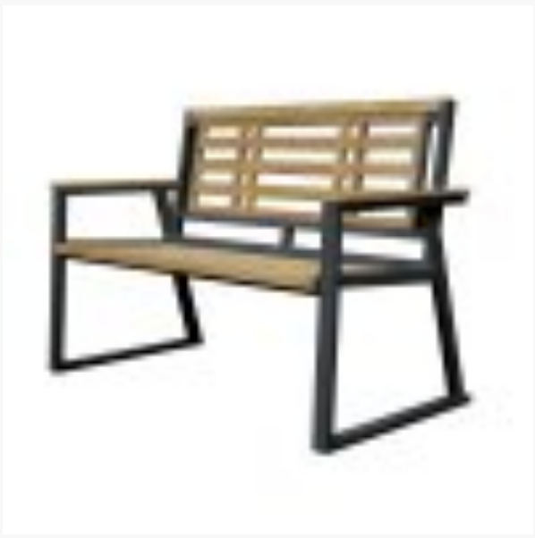


hercules triple series triple triple braced hinged hinged

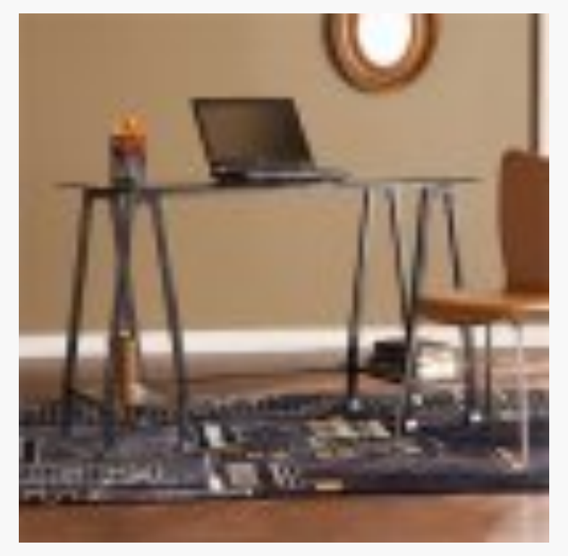
Bad Predictions



nba office chair brown



lancashire coffee table



heavy duty coffee table multiple colors