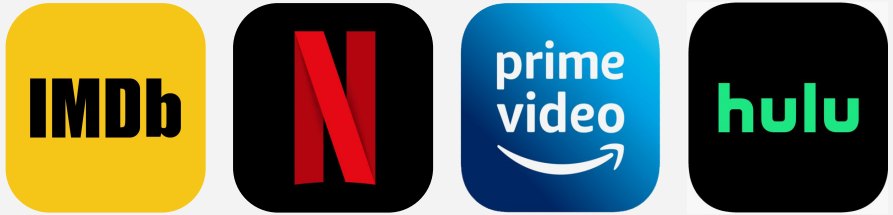
PREDICTING SENTIMENT FROM MOVIE REVIEWS

ECON 187 FINAL PROJECT

Jake Brophy, Sai Priyanka Iragavarapu,
Summer Le, Christine Tseng



OUTLINE

1. Project Goal
2. Dataset
3. Methodology
4. Results
5. Conclusions

01.
PROJECT GOAL

- Use bag of words (BoW) from reviews to classify viewers' sentiment about a movie
- Implement classification algorithms to find the best for this dataset
- Advise production companies on their upcoming project based on results

02.
DATASET

- Collection of 50,000 reviews from IMDB, allowing no more than 30 reviews per movie
- Contains an even number of positive and negative reviews, so randomly guessing yields 50% accuracy.

03. METHODOLOGY

1. Data Preprocessing
2. Implementation Procedure of Classification Algorithms
3. Brief Comments on Classification Algorithms NOT Used

DATA PREPROCESSING

Converting Words to Numbers

Stemming, Lemmatizing, and Converting to Matrix

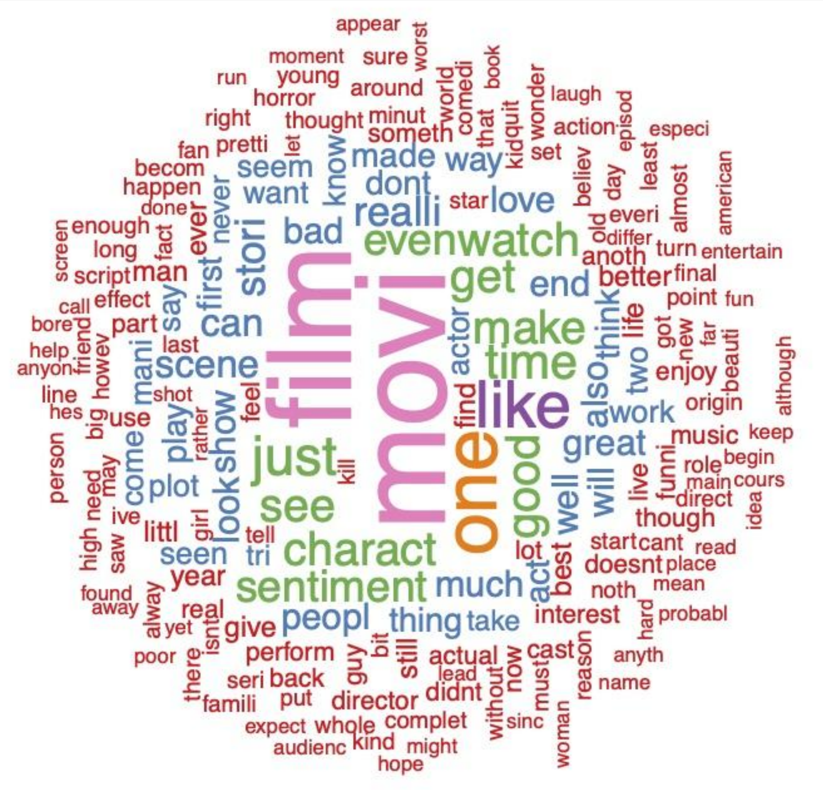
- In its raw form, statistical models can't be applied to our dataset so it needs to be converted into a useable format
- Steps:
  - Create a corpus, or collection of all the words present in our dataset
  - Clean up the corpus by converting every word to lowercase, removing punctuation, and removing stopwords (words that appear a disproportionate amount but don't add any meaning to the sentence such as a, an, the)
  - Conduct stemming on the corpus
    - This reduces the total number of words to just the stems (i.e. the word bake, baked, and bakes would all be replaced by the stem bake)
  - Lemmatize the corpus
    - This serves a similar purpose in terms of reducing the number of different word forms in our corpus, but it takes into account the meaning behind the word (i.e. the word studying, studies, and studied all collapse to the word study)
  - The final step is to create a Term Document Matrix

Term Document Matrix

	I	love	football	Messi	is	a	great	player	has	won	s
I love football	1	1	1	0	0	0	0	0	0	0	(
Messi is a great football player	0	0	1	1	1	1	1	1	0	0	(
Messi has won seven Ballon d'Or awards	0	0	0	1	0	0	0	0	1	1	1

- This is an example of a different dataset, the original sentence is not actually kept in the new matrix
- This is a very basic representation of an NLP problem and more complex representations exist
- Source: https://analyticsindiamag.com/a-guide-to-term-document-matrix-with-its-implementation-in-r-and-python/

WORD CLOUD


movi
film
one
like
evenwatch
make
time
get
just
see
charact
sentiment
good
great
well
realli
made
dont
way
love
end
stori
bad
can
scene
show
look
play
plot
seen
peopl
much
thing
take
seem
want
know
first
say
mani
come
tri
act
best
actor
think
also
two
work
will
find
appear
moment
sure
worst
young
run
horror
around
world
comedi
book
wonder
laugh
right
thought
minut
someth
that
kidquit
action
episod
especi
fan
pretti
let
set
believ
day
least
becom
happen
done
star
everi
almost
american
enough
screen
long
fact
ever
never
old
differ
anoth
turn
entertain
script
man
better
final
point
fun
call
effect
part
life
got
new
far
beauti
bore
last
friend
howev
help
anyon
enjoy
although
line
shot
origin
hes
use
feel
big
rather
music
keep
person
may
need
kill
funni
role
begin
main
cours
live
direct
idea
though
high
ive
girl
tell
littl
lot
start
cant
read
saw
year
doesnt
place
found
away
alway
real
noth
mean
yet
isnt
give
interest
hard
probabl
poor
perform
bit
still
actual
cast
reason
anyth
there
guy
now
must
seri
back
lead
name
famili
put
director
didnt
without
sinc
expect
whole
complet
woman
audienc
kind
might
hope

IMPLEMENTATION OVERVIEW

Procedure, Classification Algorithms Used,
Classification Algorithms NOT Used

CLASSIFICATION ALGORITHMS USED

LOGISTIC

NAIVE BAYES

TREE-BASED METHODS

Classification Tree

Random Forest

eXtreme Gradient Boosting

SUPPORT VECTOR MACHINE

Linear

Radial

Polynomial

NEURAL NETWORK

Long Short-Term Memory

Convolutional

IMPLEMENTATION PROCEDURE

Split data into train and test sets: 35k v. 15k observations

If computationally necessary, train on random subset of train set

For each method…

1. Tune model parameters using cross-validation
2. Implement model on test set
3. Report test set classification accuracy

IMPLEMENTATION PROCEDURE

Variable Selection/Regularization is not used.

Why? Natural choice in corpus data analysis:

- The removal of predictors (i.e. words) is a loss of information
- Construction of Term Document Matrix identifies most "important" words

CLASSIFICATION ALGORITHMS NOT USED

DISCRIMINANT ANALYSIS

Linear

Quadratic

- Implementation errors indicate collinearity is present in dataset
- Expectation of collinearity: associated words are similarly used in reviews
    - e.g. “terrible” and “awful”
- Cannot use method because cannot remove collinearity

CLASSIFICATION ALGORITHMS **NOT** USED

COMPUTATIONALLY-EXPENSIVE

Bagging

- We have 2,593 predictors
- Even with a subset of 20%, requires a lot of power
- “Bootstrap Aggregation” (averaging across samples)
- Favor Random Forest and XGBoost instead

KNN

- Curse of dimensionality

04.
RESULTS

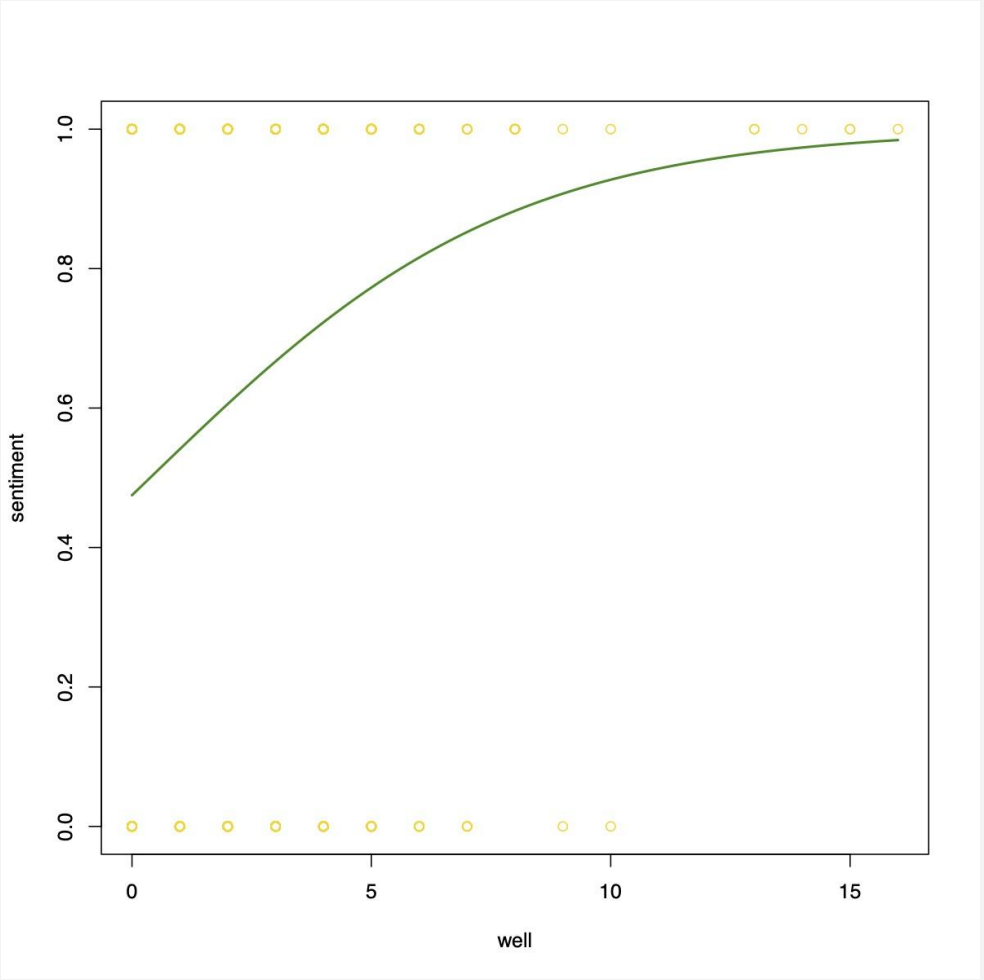
LOGISTIC REGRESSION

- Predict probabilities of the terms

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}{1+e^{\beta_0+\beta_1 X_1+\cdots+\beta_p X_p}}$$

  - X: predictors
  - Assign P(Term) > 0.5 to Sentiment = 1
  - Example: P(well) = 0.65 -> Sentiment = 1
- Logistic regression does well when the outcome is binary
- **Validation Set Accuracy: 87%**

LOGISTIC REGRESSION



NAIVE BAYES

- For a given new record to be classified, find other records that have similar predictors
- Classify the new record to the dominant class of the existing record
- For a new review, find overlapping words in previous reviews, then classify to positive/negative
- **Validation Set Accuracy: 76%**

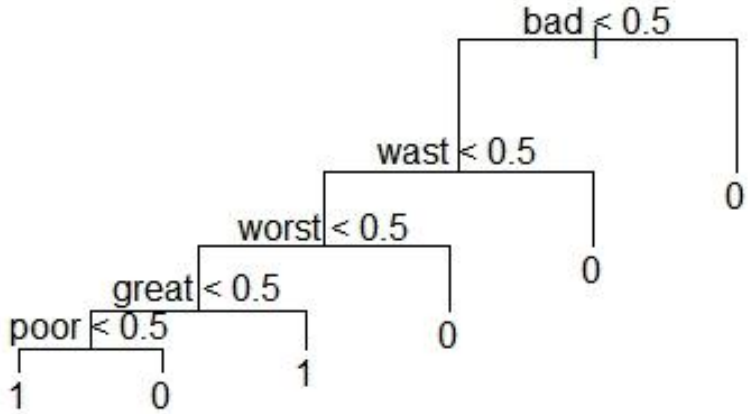
TREE-BASED METHODS

Classification Tree, Random Forest, eXtreme Gradient Boosting (XGBoost)

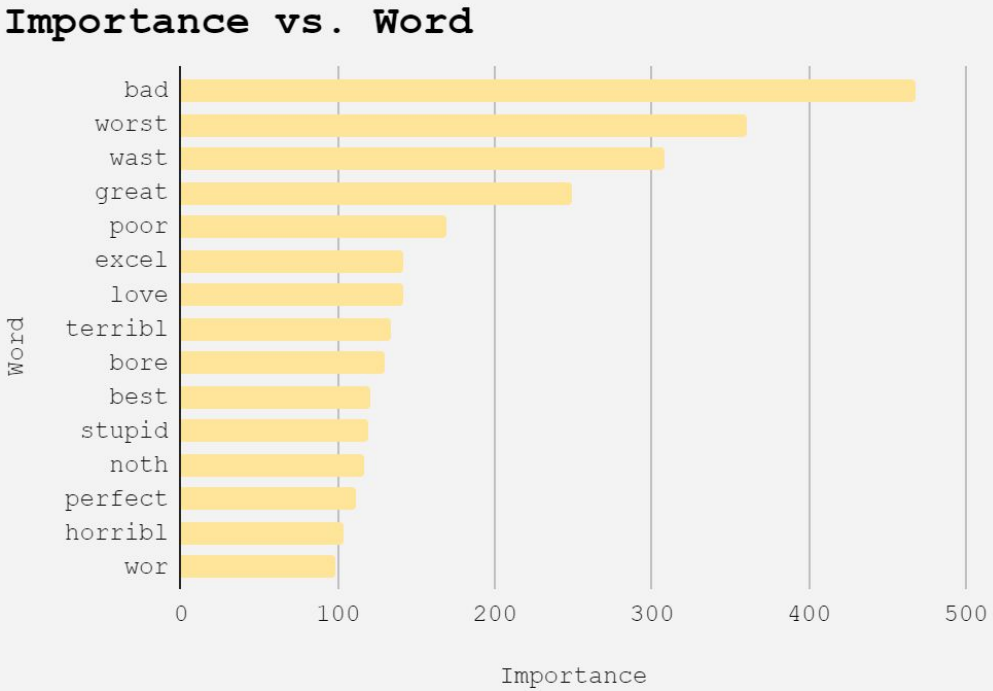
CLASSIFICATION TREE

Validation Set Accuracy: 69.89%

- 6 terminal nodes
- 5/2593 predictors used
- Didn't prune



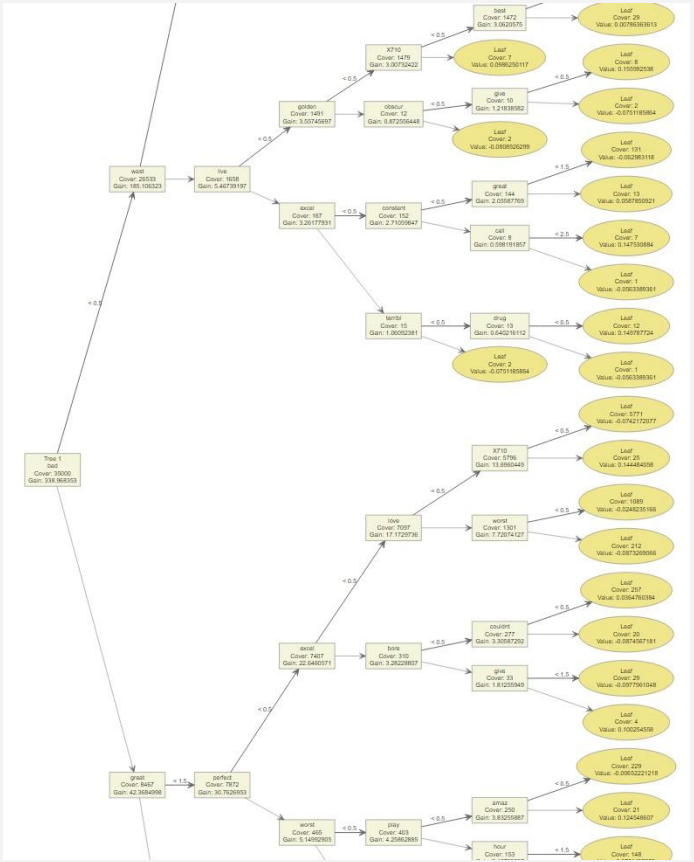
RANDOM FOREST (RANGER)



Validation Set Accuracy: 84.91%

(best among tree methods)

XGBOOST



Important variables:

bad, worst, wast, great, poor

Validation Set Accuracy: 84.6%

69.89%

Classification Tree Test Accuracy

84.91%

Random Forest Test Accuracy

84.6%

XGBoost Test Accuracy

SUPPORT VECTOR MACHINE

With Linear, Radial, and Polynomial Kernels
(run on subsetted training set)

Tuning with Cross-Validation

cost <dbl>	error <dbl>	dispersion <dbl>
1e-03	0.319	0.05466057
1e-02	0.194	0.04926121
1e-01	0.222	0.04685676
1e+00	0.228	0.03823901
1e+01	0.237	0.04191261
1e+02	0.237	0.04191261
1e+03	0.237	0.04191261

7 rows

Minimum Training Error:

Linear 0.194

Tuning with Cross-Validation

cost <dbl>	gamma <dbl>	error <dbl>	dispersion <dbl>
1e-01	0.5	0.531	0.03414023
1e+00	0.5	0.530	0.03299832
1e+01	0.5	0.530	0.03299832
1e+02	0.5	0.530	0.03299832
1e+03	0.5	0.530	0.03299832
1e-01	1.0	0.531	0.03414023
1e+00	1.0	0.531	0.03414023
1e+01	1.0	0.531	0.03414023
1e+02	1.0	0.531	0.03414023
1e+03	1.0	0.531	0.03414023

1–10 of 25 rows Previous 1 2 3 Next

Minimum Training Error:

Linear	0.194
Radial	0.530

Tuning with Cross-Validation

cost <dbl>	degree <dbl>	error <dbl>	dispersion <dbl>
1e-01	1	0.531	0.03414023
1e+00	1	0.420	0.07102425
1e+01	1	0.206	0.05295701
1e+02	1	0.211	0.04532598
1e+03	1	0.226	0.04299871
1e-01	2	0.531	0.03414023
1e+00	2	0.531	0.03414023
1e+01	2	0.508	0.08202980
1e+02	2	0.385	0.06916165
1e+03	2	0.229	0.04724640

1–10 of 25 rows

Previous 1 2 3 Next

Minimum Training Error:

Linear	0.194
Radial	0.530
Polynomial	0.206

Results

SUPPORT VECTOR MACHINE

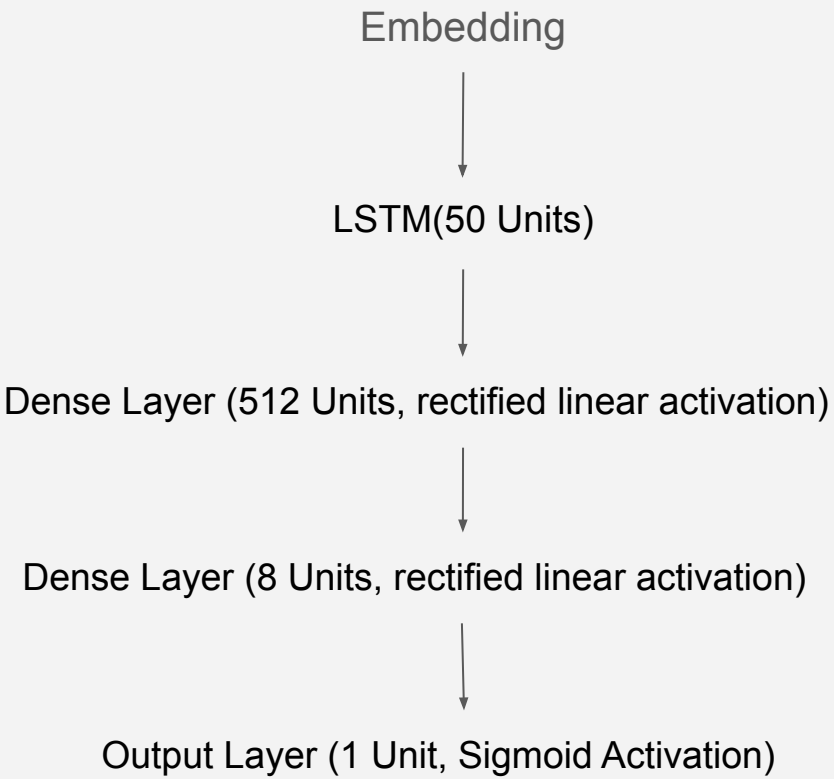
	Linear	Radial	Polynomial
Best Model:	cost = 0.01	cost = 1 gamma = 0.5	cost = 10 degree = 1
Validation Set Accuracy:	**80%** *up to 84% if increase training set	**50%** *predicting all reviews to be negative	**79%**

Seems like the decision boundary is linear.

NEURAL NETWORK

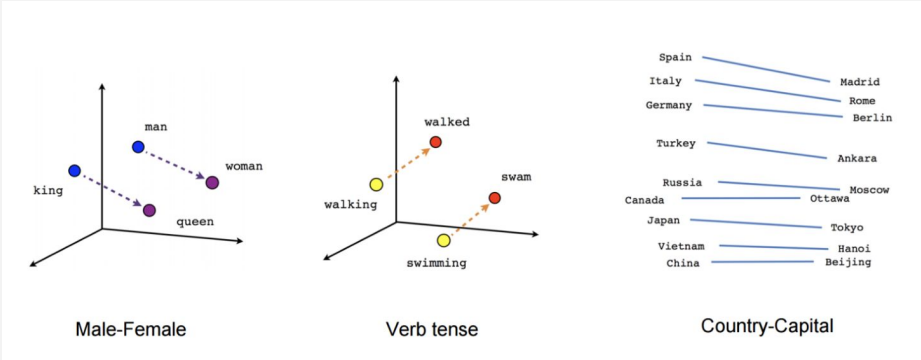
Long Short-Term Memory (LSTM NN) and Convolutional (CNN)

LSTM Neural Network Architecture



Embedding Layer

- An alternative to one-hot-encoding a corpus or making a term frequency matrix
- Instead of a vector of ones and zeros, a dense vector is produced with real values that better represent meaning behind the word.
- Words with similar meanings have closer vectors than words with very different meanings.
- Embedding layer can be trained along with the neural network



Source: https://towardsdatascience.com/deep-learning-4-embedding-layers-f9a02d55ac12

LSTM

A More complicated Recurrent Neural Net

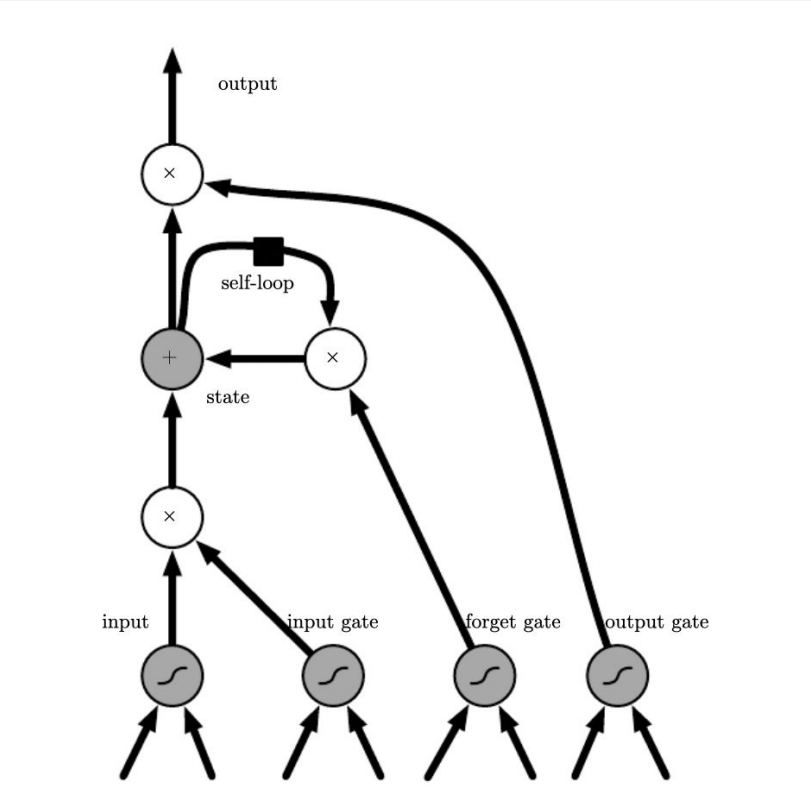
Input comes from a regular artificial neuron, and is regulated by a sigmoid input gate before it can enter the state unit

The state unit is a self-loop, but its weight is regulated by a sigmoid forget gate

Output is also regulated by a sigmoid gate.

LSTM has the advantage of not being vulnerable to the vanishing gradient problem

Source: Deep Learning, GoodFellow et al., 2016



Results

- Run on 10 epochs with a batch size of 128
- Early Stopping implemented
- Evaluated on an 80/20 split, the LSTM achieved a **Validation Set Accuracy of 86.2%**

```
Model: "sequential_13"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_13 (Embedding)     (None, 150, 50)           50000
_________________________________________________________________
dropout_8 (Dropout)          (None, 150, 50)           0
_________________________________________________________________
lstm (LSTM)                  (None, 150, 50)           20200
_________________________________________________________________
flatten_13 (Flatten)         (None, 7500)              0
_________________________________________________________________
dense_35 (Dense)             (None, 512)               3840512
_________________________________________________________________
dropout_9 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_36 (Dense)             (None, 8)                 4104
_________________________________________________________________
dense_37 (Dense)             (None, 1)                 9
=================================================================
Total params: 3,914,825
Trainable params: 3,914,825
Non-trainable params: 0
_________________________________________________________________
```

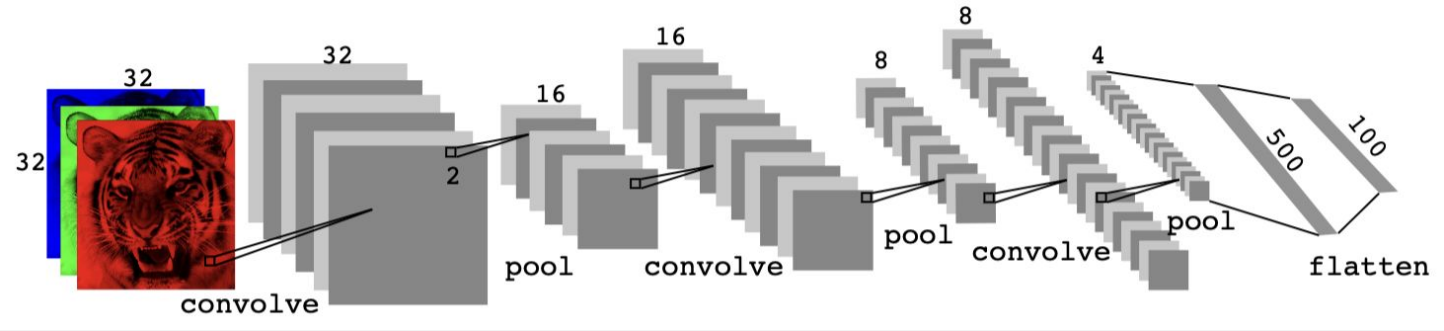
CNN

Convolutional Neural Network is a neural network that is well suited to grid data or data with spatial dependencies

The Convolutional layer performs the dot product between a matrix of learnable parameters and a portion of the grid/perception field

The window then slides (convolves) across the entire dataframe

Useful in NLP where words have an impact on the words around them



CNN

- Combining layers of CNN allows for dimension reduction and more complex feature engineering
- Resulting **Validation Set Accuracy was 87.4%** evaluated on an 80/20 split
- 12 epochs were used
- 256 batch size
- Note: CNN is faster to train than LSTM because of fewer parameters

```
Model: "sequential_12"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_12 (Embedding)     (None, 150, 50)           50000
_________________________________________________________________
conv1d_36 (Conv1D)           (None, 150, 128)          25728
_________________________________________________________________
max_pooling1d_36 (MaxPooling (None, 75, 128)           0
_________________________________________________________________
conv1d_37 (Conv1D)           (None, 75, 64)            32832
_________________________________________________________________
max_pooling1d_37 (MaxPooling (None, 37, 64)            0
_________________________________________________________________
conv1d_38 (Conv1D)           (None, 37, 32)            8224
_________________________________________________________________
max_pooling1d_38 (MaxPooling (None, 18, 32)            0
_________________________________________________________________
flatten_12 (Flatten)         (None, 576)               0
_________________________________________________________________
dense_32 (Dense)             (None, 256)               147712
_________________________________________________________________
dropout_7 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_33 (Dense)             (None, 8)                 2056
_________________________________________________________________
dense_34 (Dense)             (None, 1)                 9
=================================================================
Total params: 266,561
Trainable params: 266,561
Non-trainable params: 0
```

05.
CONCLUSIONS

BEST AND WORST MODELS

Best model:

Convolutional Neural Network with 87.4% accuracy

(Logistic was comparable with 87%)

Worst model:

SVM with Radial Kernel with 50% accuracy

Test set contains 15k reviews:

0.1% increase in accuracy = 15 more reviews are correctly classified

- CNN outperforms Logistic by 0.4% = 60 reviews
- CNN outperforms SVM with Radial Kernel by 37.4% = 5,610 reviews

FUTURE WORK

- Applications
  - Consult Netflix on upcoming projects: which categories of movies/TV shows to produce, continue, or cancel, based on category keywords (e.g. comedy, drama, thriller, etc.)
  - Based on positive sentiment, online retailers (e.g. Hulu, Amazon Prime Video) can make recommendations for movie purchases
- Improvements
  - If given increased computational bandwidth, test if methods (e.g. SVM, Trees, NN) can obtain higher accuracies
  - Use more complex nlp representations (e.g. Term Frequency-Inverse Document Frequency Matrix)