# Categorizing Amazon Products

A Categorization (and Prediction) Problem

# The Sparkforce Team

Harish Gurram

Markus Anderle

Dilip Patel

Mallesh Sunkara

Igal Levy

Brijesh Tyagi

# Our Goals, Our Data, Our Tools

- Predict the Number of Stars in a User Review
- Guess a Product's Category



**CNN**

OUR GUIDE

**UNDERSTANDING CONVOLUTIONAL NEURAL NETWORKS FOR NL**

By Denny Britz

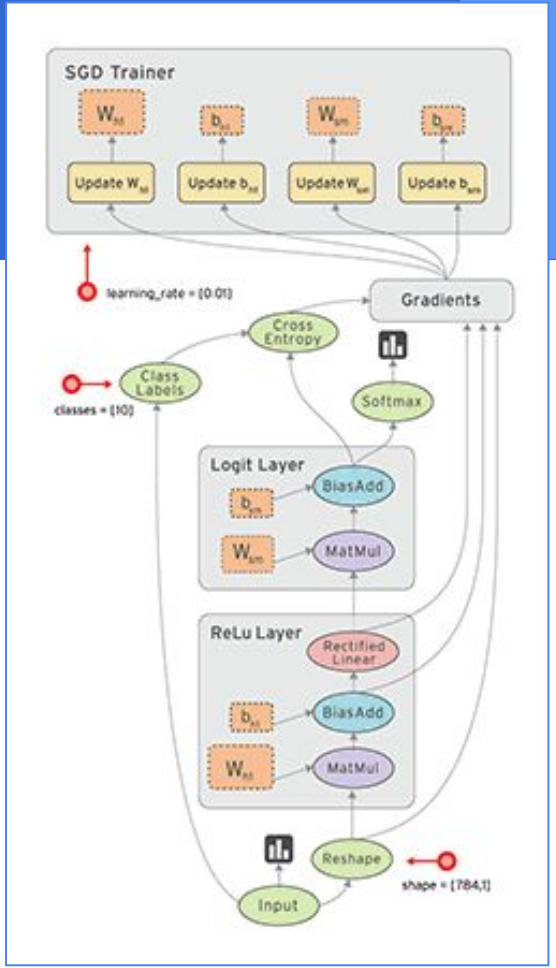
http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

# What's TensorFlow?

An open-source ML library from Google for numerical computations.

**Tensor**: A type of multidimensional data array taken from mathematics.

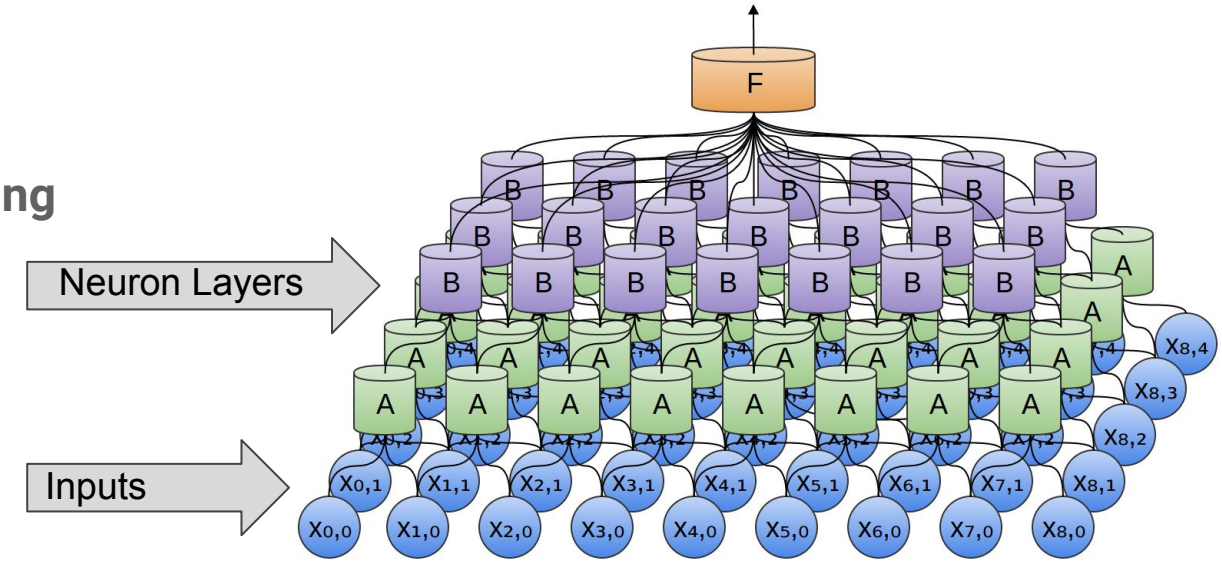
**Flow:** Data flows through computational nodes.



# CNN For NLP

**C**onvolutional **N**eural **N**etwork (NOT a television news network)

- **Embedding**
- **Padding**
- **Vocabulary Preprocessing**
- **Types of NLP**



# Amazon Product Data

- **Source:** **Julian McAuley**, UCSD, http://jmcauley.ucsd.edu/data/amazon/

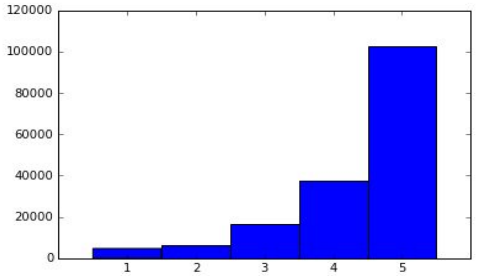
**Sample review:**

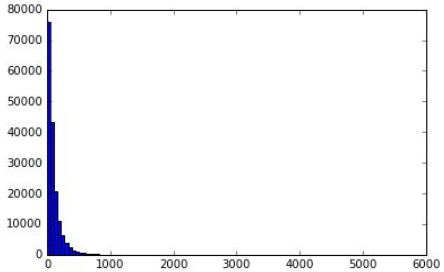

```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the piano.  He is having a wonderful time playing these old hymns.  The music  is at times hard to read because we think the book was published for singing from more than playing from.  Great purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```

# Review Observations

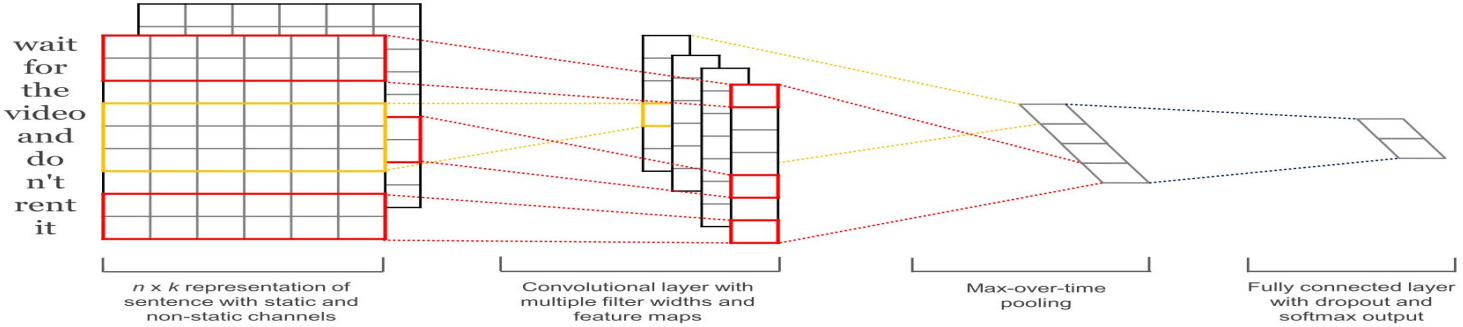
- Stars review 1 -5
- Toys and Games category of reviews
- Highly imbalanced star distribution
- “Borrowed” from “Games and Video” review to fill up difference for 1 & 2

- Word length of reviews has”long tail”
- Outlier exist
- Leads to  sparse representation
- Used cut-off at 100 words



# Network Architecture & Data Flow



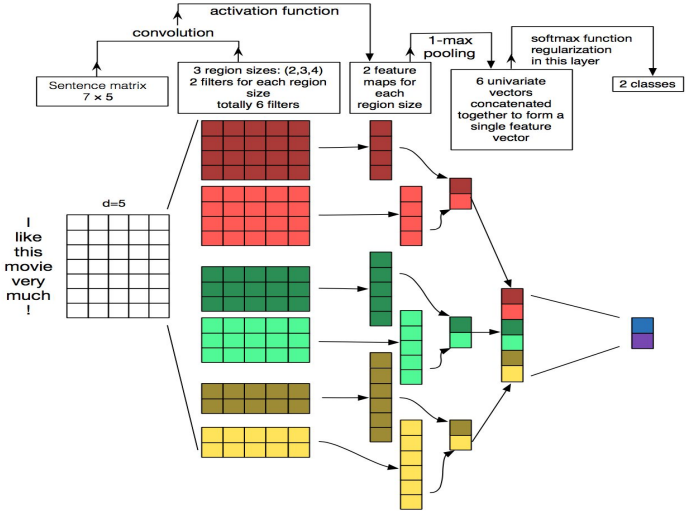

Explains what happens in each layer of the network.

i/p - word embeddings

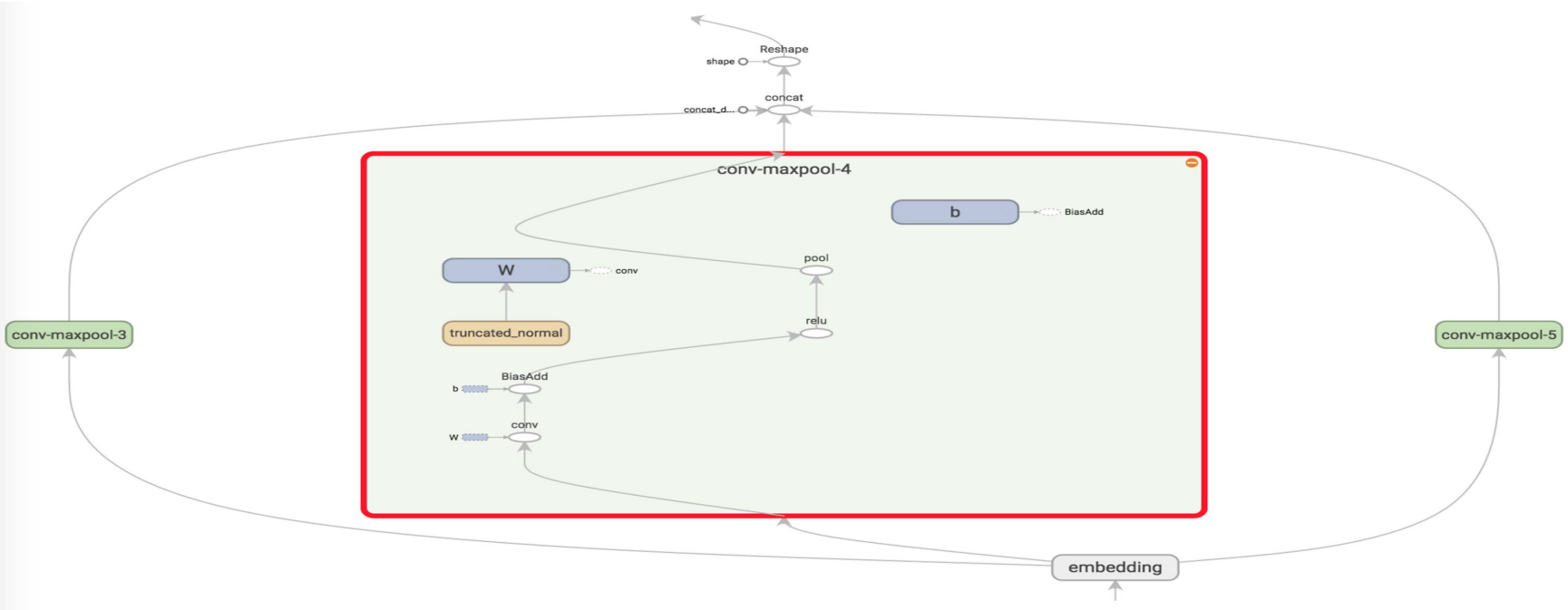
L1- Convolution Filters with activation function with
3 strides/ 3 region sizes

L2 - Pooling

L3 - Softmax Logistic classification



# Tensorflow Network Graph



# Hyper-and Training Parameters

Convolution Type: Narrow/wide, number and sizes of convolution filters, **default : Narrow,[3, 4, 5], 128**

Stride Size: defining by how much the filter moves at each step.(mostly **1 word at a time for NLP**)

Pooling Layers: Max/Avg pooling, Pooling over windows/pooling on complete convolution o/p. **Default: Max pooling**

Channels: Separate channel for each word embedding (WordProcess/Word2Vec/Glove), **static**/dynamic **word embeddings**.

Activation Function: ReLu, tanh, Softmax, **Default**: **ReLu**

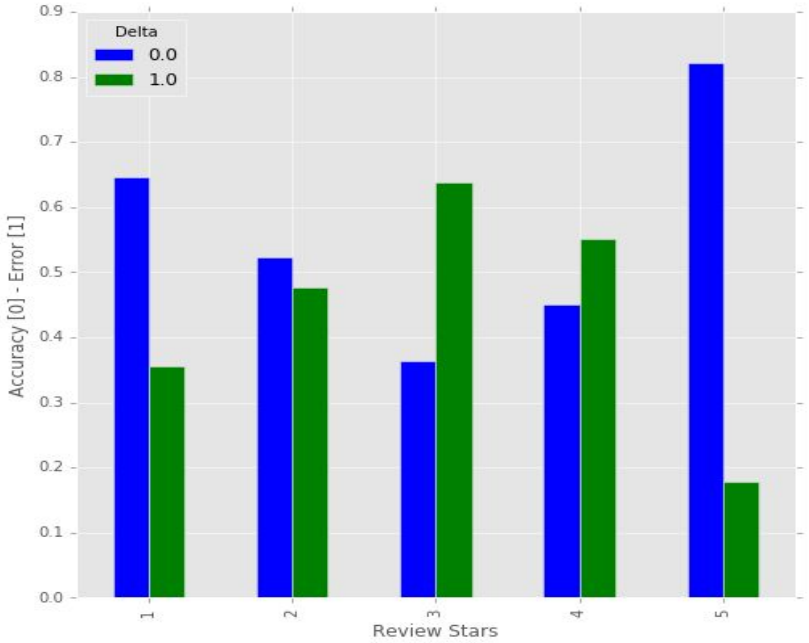
Regularization: L2/L1/Maxnorm/Dropout: **dropout rate: 0.5**

Sensitivity Analysis of CNN for classification tasks - **https://arxiv.org/abs/1510.03820**

# Results

Training:
Vocabulary Size: 42839
Train/Dev split: 31680/1000
Equally distributed

Test:
5000 reviews, equally distributed



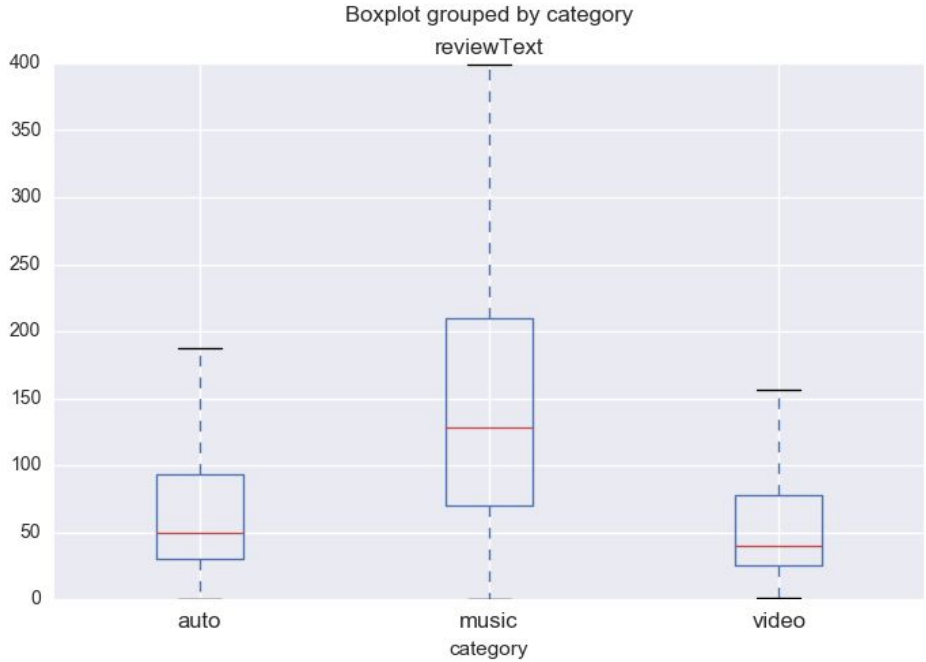
# Categorization

**Training Data:**

Due to limitations in computing power, three sets of review data is considered (Digital Music, Instant Video and Automotive).

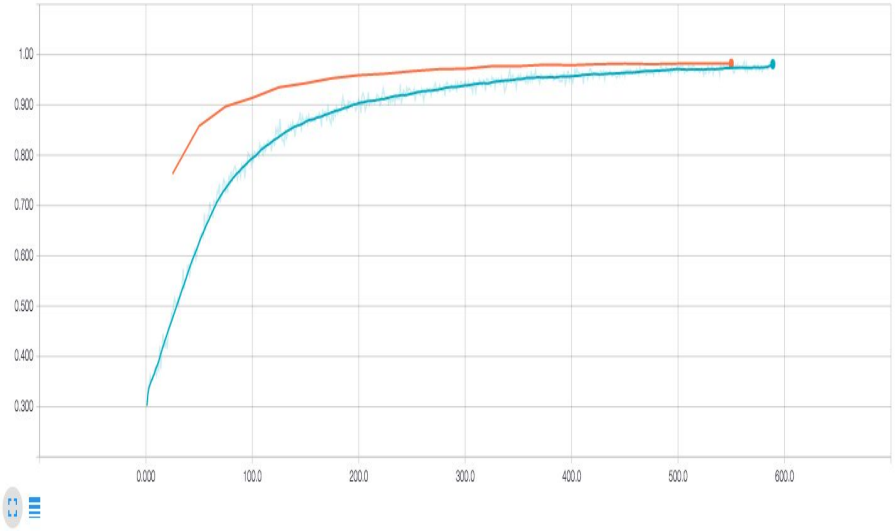
To keep the training data balanced, the dataset size is reduced to 20K records each (smallest dataset has ~20K records)

Vectorize the output to feed into the model.

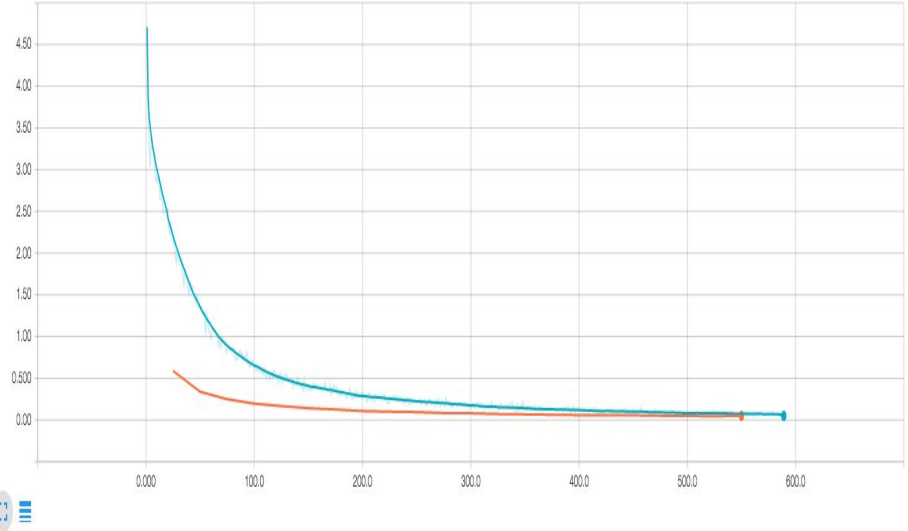


# Training results

## Accuracy




## Loss



# DEMO!

In [101]:
```
#Pick a random text from Amazon reviews (from Instant video, Digital Music or Automotive categories)
x_random = [clean_str("Remember when Star Trek used to be science fiction rather \
                       than science fantasy? These new one get further and further \
                       away from 'believable' to just being mindless action movies \
                       This movie had some glimmers of hope...but flushed it down the toilet in the finale.")]
x_st = np.array(list(vocab_processor.fit_transform(x_random))) # use x_sample of x_random based on what needs tobe pred
feed_test = {
              cnn.input_x: x_st,
              cnn.dropout_keep_prob: 1.0
            }
```

In [102]:
```
#Run the predictions

pred = sess.run(cnn.predictions,feed_test)
predicted_values = pd.DataFrame({'Reviewtext': x_random, 'category': pred })
```

In [105]:
```
predictions_text = predicted_values.apply(lambda x : [x[0],'video' if x[1] == 0 else 'music' if x[1] == 1 else 'auto'],
predictions_text
```

Out[105]:

| | Reviewtext | category |
|---|---|---|
| 0 | remember when star trek used to be science fic... | video |

# Questions?