Adgen Documentation

Machine learning and tensorflow

David Cheng

**Research Question**

How might we use deep learning to create ads or content?

**Abstract**

In the world of personalized digital advertisement, extensive research has been conducted around grouping and matching customers to various user segments. Current algorithms are able to classify users based on behavior, history, and demographics, and as a result, firms are able to drill down to an incredible level of detail on the characteristics and patterns of a customer. However this granularity in segmentation is often wasted as firms usually only a small set of advertising content to choose from, creating blanket ads with reduced impact and efficacy.

For digital advertisers, this gap highlights an unmet demand for the growing number of identifiable segment types . Through the use of deep learning, this research will seek to develop a means to provide input for a content generation algorithm such that, given characteristics of a personalization segment, the algorithm can generate personalized ad content. This will enable advertisers to meet this demand through quickly and affordably generating contents tailored towards each segment.

**Advertisement Background**

Ad types:

While there are several different flavors of advertisements, they can be grouped into three primary buckets:

* Text
* Pictures (display, banner)
* Video Ads

However, from a conceptual standpoint, text ads serve as the primary building block for the other two.

Personalization:

* Create (not a lot of work has been done here)
* Segment
* Match

The technology for content creation still greatly lags behind the content matching technology.

Segments:

Identify what the segments are and what data we can produce for each segment:

* Segment Dimensions
  + Deliberative vs. impulsive
  + Holistic vs. analytical
  + Direct response ad versus brand awareness ad (different messaging)
* How do we generate ads that can be designed for deliberative or impulsive when we have all this user generated content data?
* Reviews are not labeled - pre-labeled data would allow us to throw into CNN
  + Once CNN trained, it could identify if they are impulsive or deliberative

**Overall Approach:**

Design a system that incorporates machine learning and “human in the loop” (for human creativity), that generates ads to be shown for various personalization segments.

How to use deep learning and neural networks

* Apply neural nets and deep learning to do content generation
* Generating Ads and text
* Generate text for ads
* Exploration of obstacles
  + Still very early
  + Write a type of corpus to train the neural net on
* Started on AirBnB data set
  + Given all airbnb in nyc
  + Can we train neural net based on all these ads
  + Generate new ads?
  + Quality of the text was not very good
    - How to improve this?
  + LSTM - long short term memory
* Looking for data sets
  + Read customer needs for users
    - Training data quality isnt very good
    - How to get RNN to forget some data
      * Hard to do just for queens
    - So much variation in product that is offered
    - Where to get Annals of data to train?
    - Kaggle for data to train neural net?
    - Perhaps use user reviews for corpus?
  + What are challenges for trying to train?
    - Document - if you are to use ads
      * You will run into these 3-5 challenges
        + Ex. Repetition (page 3 in timshenko)
  + CHART ON PAGE 9
    - Clarence really likes this architecture diagram
    - Last three boxes are the "main contribution"
    - For us, identify content step is different, RNN is going to generate text, not classify.
      * When they sample sentences from different clusters, they are just picking out sentences that are part of the corpus (the UGC) and showing that, rather than generating new sentences
  + Very simple LSTM – pre-written code
    - Play with it and understand it
    - Defining structure
      * Separate title from product description
    - When adds are turned into RNNs -
      * Vector, series of 0,1,0,0
      * If there is a "This" in the first series, what is the probability of a "is" in the next spot?
  + GPUs - training data set using ; add admin users on linux

**Three Approaches**

1. Use mTurk to start writing and have turkers select generated recommendations while writing
2. Take model and have it generate different ads, have mTurks fix the grammar for the generated ads.
3. Use Turk to help generate text (focus on benefits, focus on features, focus on x), when you when you read it, it has preserved the style from ads.

**Approach #1: Predictive typing for ads**

Background thought: When you are writing an essay, what is the most challenging part? What is a tool that would help an individual write better/faster?

The amount of time spent writing is heavily front-loaded (research, developing ideas, thesis). But there is also much challenge as individuals spend time identifying what phrase or sentence construction they need to use while writing. It is faster to edit some writing than to start from scratch (as long as the writing is above a certain quality).

What if the adgen played a role in the ad generation process by serving as an IDE suggestion-type tool (autofill methods, generate getters/setters, etc etc).

Use RNN to generate the style, and then in instructions, tell turks to write according to core needs (e.g., include these product features); use RNN to generate

Turker - have a screen: design an ad for a deliberative person;

* I'm going to go write maybe 10 words and this will be a text ad shown to customers about cell phones
  + Needs to know product attributes
* Identify which attributes are strongest
* How does this specific attribute give customer value
* Specific construct that is more effective for deliberative style
* Use RNN to give me a word
  + Autocomplete at the word level
  + Constructs?
  + Tell turker to generate sentences in that way

What are the practices for designing search ads

* What are the best practices that practitioners are talking about?

<http://imnmarketer.com/textads/>

<https://www.thebalance.com/how-to-write-effective-print-ads-39151>

**Approach #2: Generate ads from RNN, fix grammar with mTurks**

Architecture would look like this:

1. UGC (user generated content, i.e. product reviews) used for training RNN
2. RNN generates sloppy clumps of words (with style of ads preserved)
3. Sloppy word clump is sent to mTurks to make the clump of words make some sort of sense
4. Somewhat sensical word clump sent to copywriter (editor); the editor will now have a bunch of ads that are okay, with the ability to modify (for that final bit of creativity, now that a base foundation has been written)
5. These ads can be deployed and A/B testing can be run to identify the best ads
6. Turks go through and weed out and augment the data to refine?

For Hauser, he uses CNN to generate product attributes.

Find fix verify algorithm --> summarize paragraphs written in this word processor

* Sends out paragraphs that you would write into a summary
* Sends out to a crowdsourcing system

Two worlds

* Personalization - may be cost prohibitive to create different segments
* Copywriters creating all this (still need to test ads)

Break up the creation of the title from the body into two steps

Time series predictions

Given yt = yt-1

Add covariance, don’t just give avg seq of stock prices, but given you’ve seen ibm prices for the last x days, give me the likely seq of ibms price, given that I tell you that its IBM

Yt = f(yt-1, x) x is covariate

X for given subset of the ys that ive seen, spit out a condition

Modify the RNN in a way, that if we tell it x is title or paragraph (binary variable)

* Generate 5 characters from the title you’ve seen
* Generate 8 words from the paragraphs you've seen

Very likely that 10 bodies generated have nothing to do with

How to add this covariate.

Title dataset and paragraph dataset

(how to figure out correlation)

AI:

Look at Udacity and geoffrey henson (figured out how to estimate first NN)

Run two sets on titles and ads

Configuration of words that shows up in the ads. And what is the optimal ad you want to show the copywriter.

Clarify verify step - won't crowdsourced version always win?

What are the goals?

* Improve purchases
* Improve click through rate (cost and time saved)

**Approach 2 Test Results**

**Trial #1**

Setup: Trained RNN with all amazon smart phone reviews (ran for about 8 hrs.), use ad generator python script to run the ad 500 times and print the result into output.txt

Result: large set of very colloquial with many negatively-connoted ads

Takeaways: style is preserved very well, but unfortunately, many of the reviews are not written in a great style nor are they very positive

**Trial #2**

Setup: Train RNN with only 5-star reviews

Result: Large set of very colloquial ads with more positive keywords and while many are no longer negative, they are not very positive either (not ad-worthy because they just make neutral statements)  
Takeaway: Data set of product reviews is good but not the best; need to identify a way to make ads more positive

**Trial #3**

Setup: Train RNN with only 5-star reviews plus smartphone ad text scraped from google and bing; use a primer (anchor word) from a set of words that were calculated as having the most positive correlation with “positive statements” (i.e., “good”, “works”, “great”)

Result: Generated ads were all very positive but did not provide many product features (previous ads did not either)

Takeaway: Anchoring seems to be a potential strategy that could help improve the quality of the generated ads. However, review text is still a problem as the style (colloquial) is still preserved and few product features are highlighted; need to consider scraping text from articles to get a larger, better corpus of data than just user reviews

**New strategy developed from lessons learned in previous trials**

Rather than building out one giant RNN and then trying to impart different features, perhaps consider separate RNNs for each product feature, e.g., batter life, screen size?

Topic modeling with a stop word

Use top of funneling to conglomerate comments or paragraphs; then gather thematically similar things to generate orthogonal sets of data.

* Combine copora in various ways, then separate out (topic model, or anchor words), then to feed and generate 10 RNNs for various stop words.

How to identify content to feed into each RNN?

* Hierarchical topical separating
* Naïve search – if the sentence contains the word, take that sentence and throw it in
* Topic modeling - give LDA (latent duration allocation) a bunch of text, and the output will tell you it is about these topics

Grammar scoring

Use to filter out quality of sentences to send to mTurks/Copywriter – perhaps develop a score?

1. Pull out sentence
2. Entropy inside sentence will look different from RNN sentence
   1. Train CNN to
3. CNN and slam words together

**Grammar**

<https://nlp.stanford.edu/software/lex-parser.shtml>

You can try using the parser "confidence". Each probabilistic parser calculates probabilities of different tags and assigns the most probable sequence. I've tried this with a part of speech tagger (<http://www.ark.cs.cmu.edu/TweetNLP/>), where each tag is assigned with some confidence (0.93, 0.45, etc.), I calculate the average confidence of all tags in a sentence and compare it to some confidence threshold (based on other sentences in the corpus).

Better approach (although naïve) is to check for how many grammar recommendations the parser will provide – just perform a hard count.