KARAN THUKRAL - 20460691

ENGINEERING LOGBOOK

SYDE 461

Contents

Week 1: Sept. 11 - 17

- Discussion with team members to clarify the idea and discuss the potential interest
- Discussions with Lyrical Security[1] regarding the data access policy along with NDA and IP agreements.
- Discussion around potential supervisor

[1] `http://lyricalsecurity.com/`

Week 2: Sept. 18 - 24

- Team met with and signed Alex Wong as our supervisor
- Back and forth with Lyrical Security about details of NDA and IP agreement
- Finished team and advisor contracts
- Talked with Alex Wong regarding the details of the NDA

Sept 23: Speed Dating Round 1

Team 7: Snow Removal

- No good fully automated way
- Risk of injury while removing snow
- Need more affordable options
- Has a good mix of skills in the team members to pull it off

Team 10: International Shipping

- Using US + Mexico border as example
- lost of illicit goods being shipped
- Not sure how to tackle it yet
- Looking into modelling the physical steps in shipping and then find ways to improve it.
- One idea is a tamper proof seal

Team 5: Heat Exhaustion

- Early detection of heat exhaustions
- #2 cause of death for athletes in US

- Health damage or death can happen from heat exhaustion
- Need to measure a good approximation of internal body temp to be able to detect it
- Prof Stashuk as advisor. Good choice

Team 1: Real Time Wait Time

- How busy is the restaurant I want to go to?
- How long will be the wait time?
- One challenge is to take into consideration crowds inside and outside the location
- Potential for ML

Team 2: Women's Health

- Uncomfortable topic to talk about
- Lot of stigma
- Chat bot to make this conversation easier
- Source info and knowledge from doctors
- Need to scrape existing forums etc to train the NLP model. This will be challenging
- Should not use conversations to train the model since can lead to mis-training and ruin the purpose - example Microsoft bot

Team 4: Pressure Ulcers

- Why does it happen?
- How do you prevent it?
- How do you take proactive action towards it?

Team 8: Understand Products

- Understand existing info about products by using forums, social media reviews etc
- structure this unstructured data somehow - NLP problem

Feedback for Us

- Try and narrow the scope
- Cannot process each packet in realtime without adding overhead
- Have very clear ways of testing and validating it
- Look at Cloudflare. Potentially have a contact there through someone in class

Tensorflow Example

- Started work on a tensorflow example to learn details about neural nets

TF Softmax Regression Saturday, 24, 2016

- Mnist
- each image is 28×28

~~can cb r~~

28×28

- 55K images = [55000, 784] array

each bixel

- lables ar One hot

→ eg 3 = [0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

→ lables = [55000, 10] array

- Soft max is a simple model for when the alg might want to assign probabilities of classification.

Gives a # b/w 0, &1 to each class & adds up to 1

- step 1 → Add evidence of our input being in certain classes ,

step 2 → convert into probabilities

- Mnist

evidence → weighted sum of pixel intensities . is -ve if in favour of not being in class , +ve otherwise

↙ input img

$$\text{evidence}_i = \sum_j W_{i,j} \, x_j + b_i$$

↳ bias

weights ↳ summing index

$$y = \text{prob} = \text{softmax}(\text{evidence})$$

↳ find out → softmax turns evidence into prob dist at the output layer

AThukral
24/09/16

Week 3: Sept. 15 - Oct. 1

Tensorflow Example Continued

Sunday, 25, 2016

- Since softmax turns evidence into prob dist its used as the last layer even in complicated models.

- ~~Soft~~ i = class
  j = summing index over all classes

  x = input image

- for this softmax is serving as a link func ~~out~~ shaping the output of our linear func into the form we want!!
- its tallying the evidence into prob for our input img

$$\text{Softmax}(x) = \text{normalize}(\exp(x))$$

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\Sigma_j \exp(x_j)}$$

↳ exponentiating its inputs & then normalizing them

- No hypothesis has 0 or -ve weights
- The exp inc the weight for one unit of 4+ve evidence & reduce multiplicatively similarly for -ve evidence

- $y = \text{softmax}(Wx + b)$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax}\left(\begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right)$$

Regression

- Numpy ~~comp~~ does computations in C & thus cares about data types. There is overhead in swoitching b/w the two though.
- Tensor flow does heavy lifting outside Python. You describe the graph in py & execute outsid of py.

Trainning

- Define what it means for the model to be bad?
  ↳ cost func, loss func etc
- minimize the error
- Cross - entropy

$$H_{y'}(y) = -\sum_i y'_i \log(y_i)$$

y = predicted prob ~~dits~~ dist

y' = true dist

measure ~~th~~ how inefficient our prediction is

→

Back Propogation

→ Reverse mode differentiation

→ get → ~~How a~~ gives the derivate of the output with respect to every single node

25/09/16

Back Propagation

- Reverse mode differentiation
- Regular chain rule gives you the derviative of the output with respect to one input/node
- Doing that for all nodes is intractable
- To extend this to find $\frac{\partial output}{\partial}$ with respect to all nodes and inputs in the neural net/graph you start using the chain rule from the other end (output) and go till the input layer. This is essential for neural networks. [2]

Multiple Instance Learning Paper

[2] Christopher Olah. Calculus on computational graphs: Backpropagation. https://colah.github.io/posts/2015-08-Backprop/, August 15, 2015

Back Propogation

→ Reverse mode differentiation

→ get → ~~How a~~ gives the derivate of the output with respect to every single node

25/09/16

Chiyam Meeting Sept 26

Todo

- intro to topic, context
- ~~ne~~ Problem def
- Needs, Prior art
  - ↳ users

- Project objectives & outcomes
  - ↳ specific objectives
  - ↳ realistic in time frame
  - ↳ how will they solve the problem
  - ↳ major problems

Problem Def

ML Paper (Sept 26, 2016)

- detection is based on:
  - URL
  - flow duration
  - number of bytes trasferred from client to server & other way
  - user agent
  - referer
  - MIME - type
  - HTTP Status
- The n-dimentional feature vector represents each proxy log & used to differentiate b/w legit & malicious traffic
- Paper model only analysis single proxy log, & skips temporal features
- Attacking domains change frequenly but behaviour doesn't.

How

- The proxy logs originating at a particular user machine are grouped into bags based on the domain in the URL.
- The bags are labeled according to the domain.

  ↳ if domain is in any blacklist, the bag has a +ve label

  ↳ if not, bag has a -ve label

MIL

- flow is described by a vector of features

  $x \in X \subseteq \mathbb{R}^d$ & a label $y \in Y = \{\underset{\text{malicious}}{+1}, \underset{\text{not}}{-1}\}$

- Network traffic monitored in a given period is fully described by the completed annotated data

$$D_{cmp} = \{(x_1, y_1) \ldots (x_m, y_m)\} \in (X \times Y)^m$$

independant, identicaly distributed

assumed to be generated from i.i.d. random vars with unknown dist

$$p(x, y)$$

- Annotating everything is expensive, thus we use bags of flows

- The weakly annotated data

$$D_{bag} = \{\underbrace{x_1, \ldots, x_m}_{\text{features}}, \underbrace{(B_1, z_1), \ldots (B_n, z_n)}_{\text{assignment to labled bags}}\}$$

$$\{(B_1, z_1), \ldots (B_n, z_n)\} \in (P \times y)^m$$

P = set of all partitions of indeces $\{1, \ldots m\}$.

- The ith bag is a set of flow features $\{x_j \mid j \in B_i\}$ label by $z_i \in Y$.

- D_{bag} carries partial info about D_{imp}.

Assumptions:

1) Flow features $\{x_1, ..., x_m\}$ are the same in both.

2) Negative bag contains a single instance, & the label is correct.

$\Rightarrow z_i = -1$ implies $|B_i| = 1$ & $y_i = -1$

3) +ve bags have a variable size & at least 1 instence is positive

$\Rightarrow z_i = +1$ implies $\exists j \in B_i$ s.t. $y_j = +1$

Bibliography

Christopher Olah. Calculus on computational graphs: Backpropagation. `https://colah.github.io/posts/2015-08-Backprop/`, August 15, 2015.