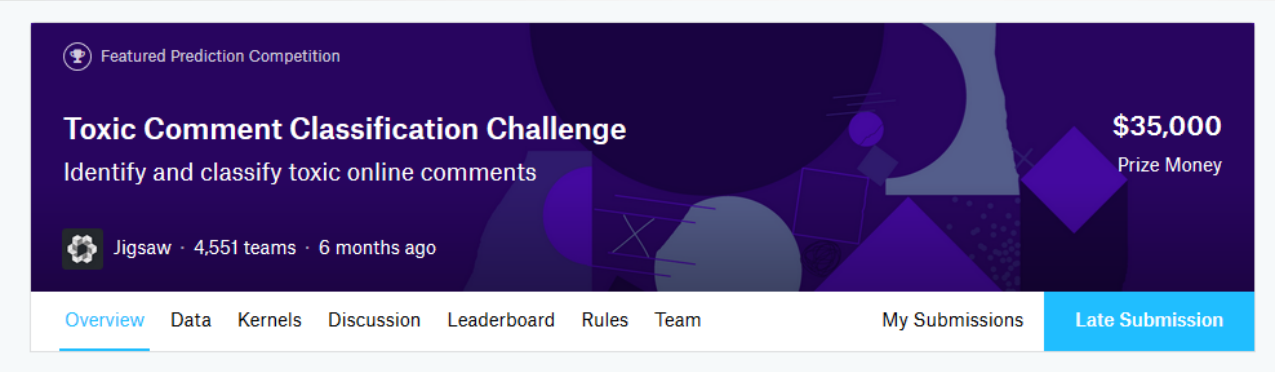
Toxic Comment Classification

Tensorflow Recurrent Neural Net

Overview



- Kaggle competition to classify comments from a Wikipedia forum
- Classifications:
  - "Toxic", "Severe Toxic", "Obscene", "Threat", "Insult" and "Identity Hate"
- 160,000 Training Records – hand labeled by competition host
- 150,000 Testing Records – to be labeled by our algorithm!

Workflow

- 1) Load data into script
- 2) Define the Network Architecture
- 3) Convert data into Tensorflow usable format
- 4) Train the Model
- 5) Make Predictions

Training Snippet

	A	B	C	D	E	F	G	H
1	id	comment_text	toxic	severe_toxic	obscene	threat	insult	identity_hate
2	0000997932d777bf	Explanation Why the edits made under my username Hardcore Metallica Fan were reverted? They weren't vandalisms, just closure on some GAs after I voted at New York Dolls FAC. And please don't remove the template from the talk page since I'm retired now.89.205.38.27	0	0	0	0	0	0
3	000103f0d9cfb60f	D'aww! He matches this background colour I'm seeming	0	0	0	0	0	0
4	000113f07ec002fd	Hey man, I'm really not trying to edit war. It's just that th	0	0	0	0	0	0
5	0001b41b1c6bb37e	"	0	0	0	0	0	0
6	0001d958c54c6e35	You, sir, are my hero. Any chance you remember what p	0	0	0	0	0	0
7	00025465d4725e87	"	0	0	0	0	0	0
8	0002bcb3da6cb337	[redacted]BEFORE YOU[redacted]AROUND ON MY WORK	1	1	1	0	1	0
9	00031b1e95af7921	Your vandalism to the Matt Shirvington article has been	0	0	0	0	0	0
10	00037261f536c51d	Sorry if the word 'nonsense' was offensive to you. Anyw	0	0	0	0	0	0
11	00040093b2687caa	alignment on this subject and which are contrary to thos	0	0	0	0	0	0
12	0005300084f90edc	"	0	0	0	0	0	0
13	00054a5e18b50dd4	bbq	0	0	0	0	0	0
14	0005c987bdfc9d4b	Hey... what is it..	1	0	0	0	0	0

Testing Snippet

	A	B	C	D	E	F
1	id	comment_text				
2	00001cee341fdb12	Yo [redacted] Ja Rule is more succesful then you'll ever be				
3	0000247867823ef7	== From				
4	00013b17ad220c46	"				
5	00017563c3f7919a	:If you have a look back at the source, the informatio				
6	00017695ad8997eb	I don't anonymously edit articles at all.				
7	0001ea8717f6de06	Thank you for understanding. I think very highly of y				
8	00024115d4cbde0f	Please do not add nonsense to Wikipedia. Such edits				
9	000247e83dcc1211	:Dear god this site is horrible.				
10	00025358d4737918	"				
11	00026d1092fe71cc	==				
12	0002eadc3b301559	I think its crap that the link to roggenbier is to this ar				
13	0002f87b16116a7f	"::::				
14	0003806b11932181	, 25				
15	0003e1cccfd5a40a	"				
16	00059ace3e3e9a53	"				
17	000634272d0d44eb	==Curren				
18	000663aff0fffc80	this other one from 1897				
19	000689dd34e20979	==				
20	000834769115370c	:: Wallamoose was changing the cited material to say				
21	000844b52dee5f3f	|blocked]] from editing Wikipedia. |				

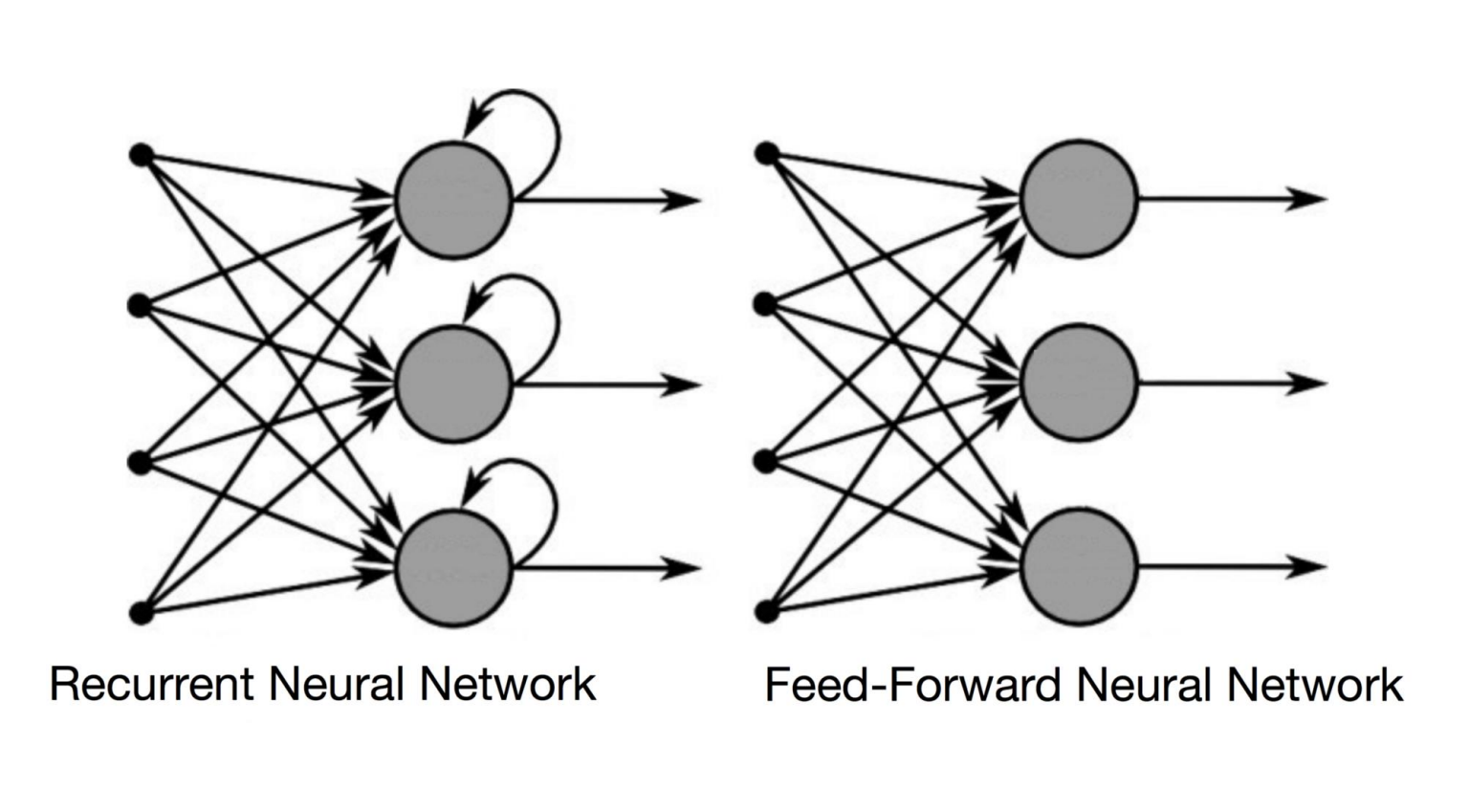
Loading Data

- Use Pandas library to load in .csv
- “Comment” column becomes the Inputs
- “Toxic”, “Severe Toxic”, “Obscene”, “Threat”, “Insult” and “Identity Hate” become the Labels
  - This is converted into a list format 6 indices wide that correspond to each label column – [1, 0, 0, 1, 0, 1]
- After loading data, text cleaning and enrichment measures are implemented to amplify significance of meaningful words

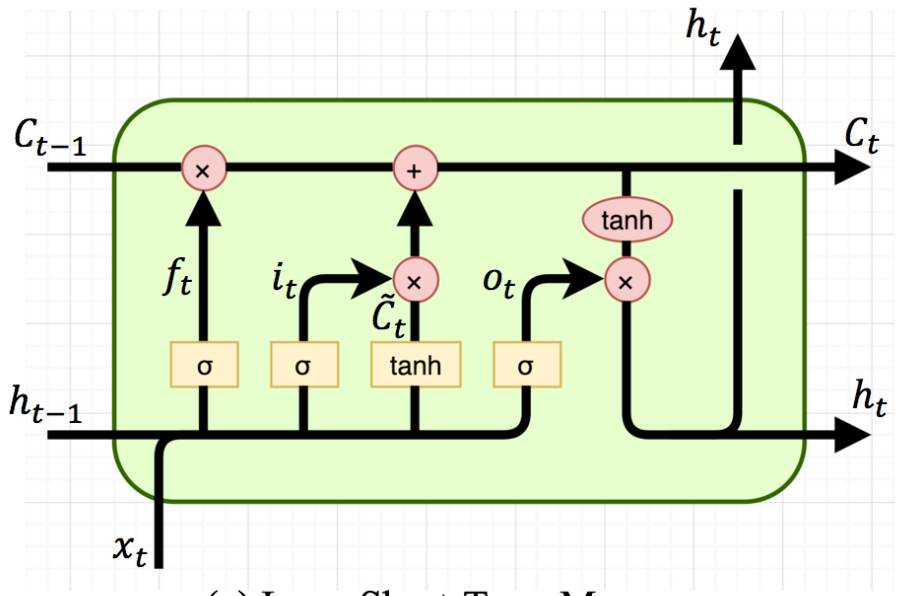
Defining the Network

- Use the Tensorflow library, we will be implementing a Recurrent Neural Net
  - Tensorflow is a great library for implementing all kinds of Networks
  - Utilizes GPU processing for expedited training
- Our network will have:
  - 1 Input Layer
  - 1 Hidden Layer, with 512 LSTM Cells
  - 1 Output Layer

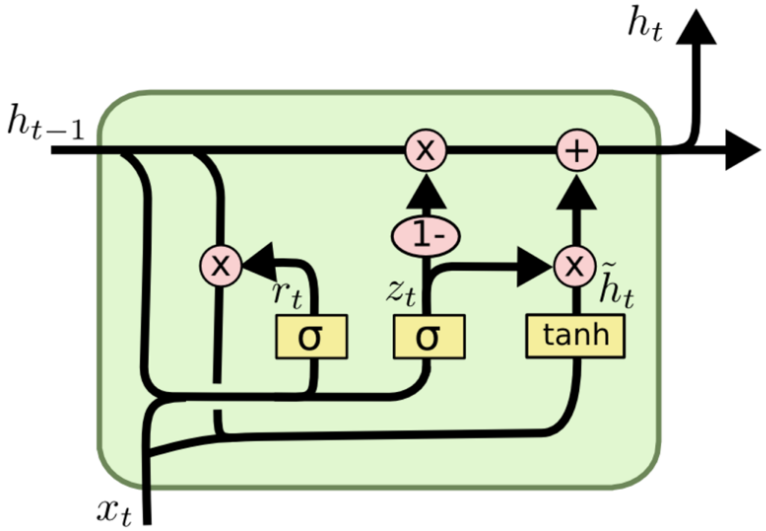
Why Recurrent Neural Net vs. FFN?



Cell Options: LSTM vs GRU



(a) Long Short-Term Memory

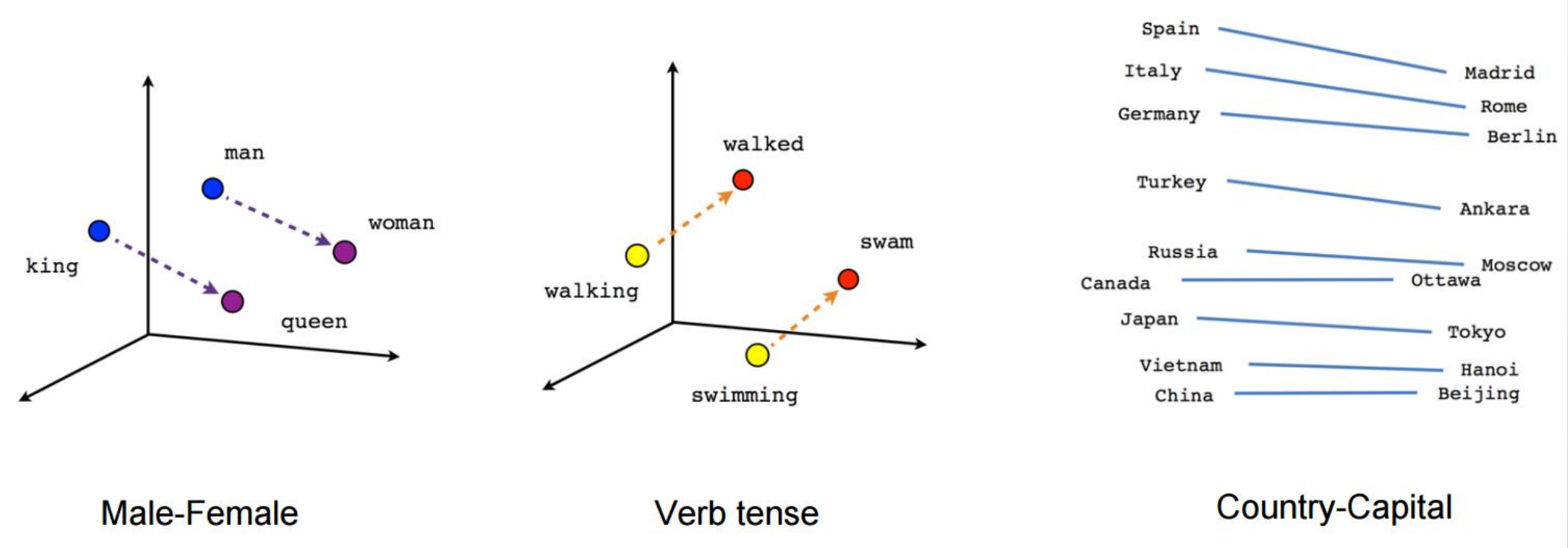


(b) Gated Recurrent Unit

What does the network need?

- Properly formatted data!
- Tensorflow operates around data structures called Tensors
  - These are essentially matrices, or lists of lists, depending on your background
- In this instance, we will be working around 3-D Tensor Inputs
- Level 1 is the Batch Size
  - Level 2 is the Number of Words in a Comment
    - Level 3 is a Vector Representation of each word
- Outputs will be a 1-D Tensor 6 wide (same size as number of Classes)

Making Tensors – Word Vector



Making Tensors – Word Vector

- Two options for attaining Word Vectors
  - 1) Training a Word2Vec model using your own text data
  - 2) Using a pretrained model of Vectors
- Here, I opted to use pretrained Vectors from Spacy.io
- Their Vectors are 384 indices wide
- This makes up the "Lowest" level of our Input Tensor

Making Tensors – Comment Word Stream

- Because conversation and text is very contextual, Recurrent Neural Nets are the optimal network choice
  - In order to preserve context, we need to feed sequences of text, one word after another
- Tensorflow needs to know the Max Sequence length
- To do this:
  - Find average of the number of words (tokens)
  - Length = avg + 2*stddev
  - Came out to 300 for this data set

Making Tensors – Comment Word Stream

- Every Comment needs to be exactly 300 words long
- For shorter comments, we pad the back
  - Appending 384 long vectors of Zeros until reaching 300 words
- For longer comments, several schools of thought
  - In this case, I summed enough word vectors in the center of the sequence to get down to 300 words
  - Can also just remove the long ones

Making Tensors – Batch Size

- Batch size is the number of comments to look at, at a single time
  - Not unique to RNNs – All Networks have this as Top Level dimension
- Large batch sizes – 256 to 1024
  - Slower processing
  - Less in depth optimization – good for generalization, each sample matters less
  - Memory issues though, not good for image processing
- Small batch sizes – 1 to 128
  - Faster processing
  - More in depth look at each sample
  - Overfitting possible

Making Tensors – Final Dimensions

- After all this setup, the final Input dimension size is:
  - 128 Batch Size X 300 Words X 384 Index Word Vector
- Output dimension is:
  - 6 Classifications

6) Training Process – Overall Idea

- Want the network to see as much data as possible
  - Need to avoid Overfitting – aka memorizing the input data
- To combat overfitting, using random comments out of order
- Need to watch network train
  - Accuracy – want to see accuracy rising, as long as Loss is dropping
  - Loss – (amount incorrect) should decrease, until reach a bottoming out
    - At this bottom, stop training

Training Process - Validation

- To help gage Training and Combat overfitting, use technique called Validation
- This is doing fake “predictions”
  - Performed on data split from the Training set
  - We already know the answers
  - Helps to ballpark how well will perform on actual testing set

```
[1 1 1 0 1 1] [0.96382385 0.13392922 0.4735796  0.03125597 0.48971945 0.13643862]
[1 0 0 0 1 0] [0.9620943  0.07468704 0.38714013 0.02189665 0.48987272 0.11311764]
[1 0 0 0 1 0] [0.96189266 0.07069163 0.38193834 0.02161204 0.5126006  0.1119339 ]
[1 0 1 0 1 0] [0.9582799  0.126599   0.4871138  0.02861618 0.42941508 0.12185494]
[1 0 0 0 0 0] [0.96130735 0.1205037  0.48559585 0.02659003 0.43373612 0.11495809]
92/256 R-init RAN: Accuracy at 92: 93.75000
[1 0 1 0 1 0] [0.9544784  0.07052454 0.37121403 0.02286721 0.5155425  0.10856833]
[1 0 0 0 0 0] [0.9480141  0.114159   0.43101346 0.03049614 0.38347125 0.11320248]
[1 0 0 0 0 0] [0.95450836 0.0652789  0.3589456  0.02216885 0.49114412 0.10316646]
[0 0 1 0 1 0] [0.9538031  0.06705999 0.36517638 0.02245401 0.51031685 0.10414129]
[1 0 1 0 1 0] [0.93769896 0.1145888  0.40481415 0.03547747 0.4158516  0.1323241 ]
93/256 R-init RAN: Accuracy at 93: 81.25000
[1 0 1 0 0 1] [0.94820493 0.1206248  0.45292658 0.03247375 0.4131245  0.13030931]
[1 0 0 0 0 0] [0.9510548  0.06470874 0.37692657 0.02197044 0.4913866  0.09800299]
[1 0 1 0 1 0] [0.95035374 0.06622162 0.37151507 0.02283605 0.5113628  0.10092932]
[1 0 1 0 1 0] [0.9327345  0.06732211 0.39520496 0.02333705 0.41599715 0.09394737]
[1 0 1 0 1 0] [0.94916904 0.06774069 0.3868644  0.02275481 0.5120323  0.10168304]
```

```
[1 0 0 0 1 0] [9.99974608e-01 5.97091712e-13 9.39326839e-09 1.29482665e-08
 9.97994304e-01 1.82949496e-08]
[1 0 1 0 0 0] [9.9252880e-01 1.6130975e-09 9.9979299e-01 5.2329296e-09 3.0174468e-07
 2.0117438e-10]
[1 0 0 0 0 0] [9.9999690e-01 7.7366503e-04 1.2612701e-04 2.1297124e-08 6.4585358e-04
 3.3271126e-06]
[1 0 1 0 0 0] [9.9999690e-01 1.8770416e-07 9.9997377e-01 1.3800580e-13 6.5128249e-04
 1.9924218e-10]
[1 0 0 0 1 0] [9.9987757e-01 5.6142613e-10 6.4447920e-07 3.8407442e-09 9.9929130e-01
 1.0162222e-08]
7380/12675 R-init RAN: Accuracy at 7380: 79.68750
[1 0 0 0 0 0] [9.9999976e-01 2.2377308e-06 9.9432737e-06 2.7383038e-09 1.1509101e-06
 1.5234678e-09]
[1 0 0 0 0 1] [9.9988794e-01 4.1497068e-04 1.4125850e-05 3.7155487e-03 7.7030709e-06
 9.9996245e-01]
[1 0 1 0 1 0] [9.9999976e-01 1.4951650e-05 9.9994528e-01 4.0773473e-05 9.9934250e-01
 4.1645922e-07]
[1 0 1 0 0 0] [9.9999237e-01 4.0337042e-04 9.9970406e-01 1.1213967e-05 1.4627581e-06
 2.0160282e-04]
[0 0 1 0 0 0] [4.4152883e-05 1.0196491e-08 9.9999988e-01 1.8764578e-06 4.9514011e-03
 1.3983101e-04]
7390/12675 R-init RAN: Accuracy at 7390: 70.31250
```

Making Predictions

- Format Test comments exactly the same as Training comments
- Capture the output 1-D tensor and convert into a prediction
- B/c each comment belongs to multiple classes, use Sigmoid activation
  - Sigmoid allows each index to scale 0-1 independent of other indices
  - As opposed to Softmax, where sum of all indices = 1 – aka only one class possible
- Convert these outputs into a list of list, and write to .csv for submission

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
rms_512_layer.csv	3 months ago	4 seconds	5 seconds	0.9705

Complete

Jump to your position on the leaderboard

Public Leaderboard Private Leaderboard

The private leaderboard is calculated with approximately 75% of the test data.

This competition has completed. This leaderboard reflects the final standings.

Refresh

In the money Gold Silver Bronze

#	△pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	—	Toxic Crusaders			0.9885	171	6mo
2	—	neongen & Computer says no			0.9882	129	6mo
3	▲3	Adversarial Autoencoder			0.9880	451	6mo
4	▲1	Leyantech			0.9878	164	6mo
5	▲2	TPMPM			0.9878	299	6mo
6	▼3	Mike			0.9878	182	6mo
7	▲1	GL Team			0.9878	247	6mo

Resources

- Competition Home Page:
  - https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge
- RNN LSTM Explanation:
  - https://colah.github.io/posts/2015-08-Understanding-LSTMs/