Assignment-1

CS 524-Deep Learning and Neural Networks

Submitted To : Prof. Bhaskar Sharma

Submitted by: Nelluri suresh kumar (MB19GID210)

Method: Problem-solution-Impact

Problem :

English has a millions of distinct words, so neural networks would have millions of neurons in the input layer. This high-dimensionality would prevent us from discovering the most interesting patterns with (deep) networks, and as well as model should concern about syntax and semantic structures.

Solution :

Instead of seeing a text as a sequence of words, we saw it as a sequence of characters. Since there are only a few dozen possible characters, that would make it possible to directly use deep learning systems for text classification. Character-level Convolutional Networks do not require the knowledge about the syntactic or semantic structure of a language. This simplification could be crucial for a single system that can work for different languages. Working on only characters also has the advantage that abnormal character combinations such as misspellings and emoticons may be naturally learnt.

Character-level Convolutional Networks :

1. Convolutional networks
2. Character quantization
3. Model design

Convolutional networks : As Text is 1-Dimensional so here we use 1-D convolution.

Why CNN :

Properties : Spatial invariance or parameter sharing.

How CNN works :

For images, a convolution is often used with a kernel to reveal certain properties of the image. kernel, filter, or feature detector multiply its values element-wise with the original matrix, then sum them up. To get the full convolution we do this for each element by sliding the filter over the whole matrix.

They use the same idea in this paper, each character is transformed into a binary vector ('a' is [1,0,…,0], 'b' is [0,1,0,…,0] and 'z' is [0,…,0,1]). Broadly speaking, a sequence of characters looks like an image, on which convolutions can be applied.

Character quantization:

Our models accept a sequence of encoded characters as input. Encoding Followed is one-hot encoding. The alphabet used in all of our models consists of 70 characters, including 26 english letters, 10 digits, 33 other characters and the new line character. The non-space characters are:

abcdefghijklmnopqrstuvwxyz0123456789 -,

;.!?:’”/\|_@#$%ˆ&*˜‘+-=<>()[]{}

FOR EXAMPLE :

sentence : I LOVE NLP .

split it into characters. One hot encoding for each character.

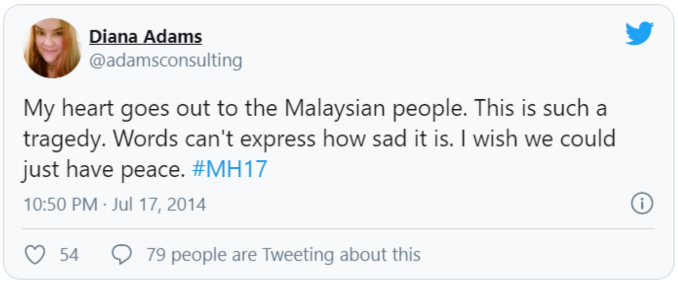
Alphabets

Of size 70(m)

I		L	O	V	E		N	L	P
1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1
0	0	1	0	0	0	0	0	1	0
….	…	…	…	…	…	…	…	…	…
0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0	0

$\longrightarrow$ Feature length (I_0)

The encoding is done by prescribing an alphabet of size m(max.length) for the input language, and then quantize each character using 1-of-m encoding (or “one-hot” encoding). Then, the sequence of characters is transformed to a sequence of such m sized vectors with fixed length I0. Any character exceeding length l0 is ignored, and any characters that are not in the alphabet including blank characters are quantized as all-zero vectors. The character quantization order is backward so that the latest reading on characters is always placed near the begin of the output, making it easy for fully connected layers to associate weights with the latest reading.

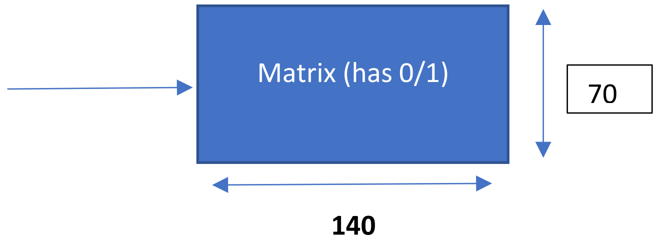


Diana Adams
@adamsconsulting

My heart goes out to the Malaysian people. This is such a tragedy. Words can't express how sad it is. I wish we could just have peace. #MH17

10:50 PM · Jul 17, 2014

54   79 people are Tweeting about this



Max .length =140 , alphabets = 70

How to select Max.Length :

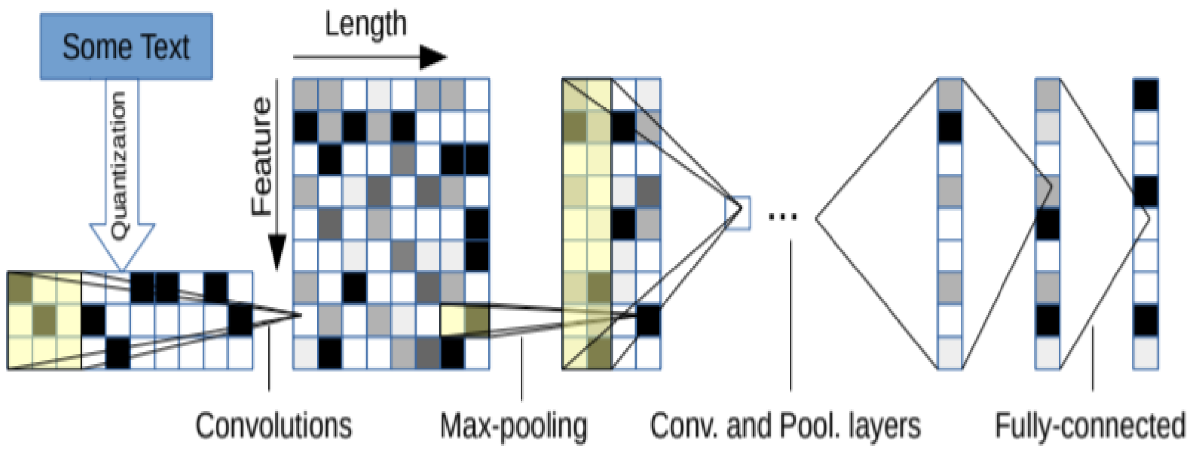
No. of characters that can capture most of the texts of interest.

Model design:

In this paper 2 ConvNets are built – one large and one small. They are both 9 layers deep with 6 convolutional layers and 3 fully-connected layers.

(Images taken from research paper for explaining purpose)

Architecture :



Properties:

Module: conv1D

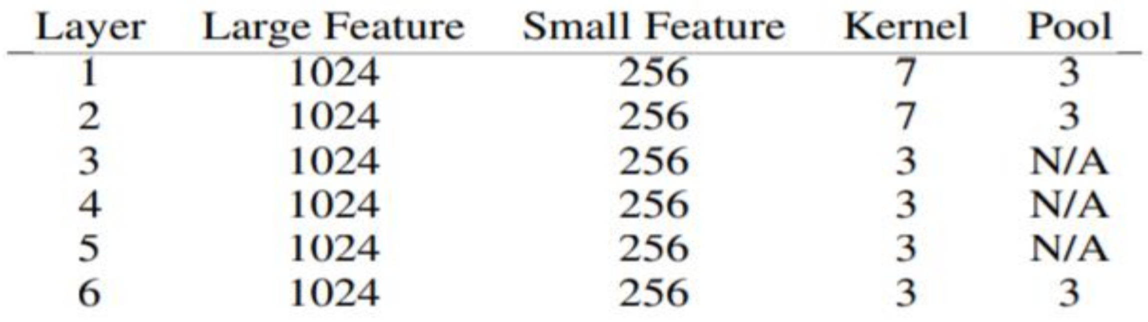
Pooling method: Max

Algorithm: stochastic gradient descent (SGD) with a minibatch of size 128, using momentum 0.9 and initial step size 0.01

Activation function: ReLUs (for introducing non-linearity)

How Character-level Convolution works :

1. Extract information from neighboring words via convolutional filter
2. kernel size ≈≈ text window from word2vec (analogy)
3. Filter moves in word-direction (y-axis)



Layer	Large Feature	Small Feature	Kernel	Pool
1	1024	256	7	3
2	1024	256	7	3
3	1024	256	3	N/A
4	1024	256	3	N/A
5	1024	256	3	N/A
6	1024	256	3	3

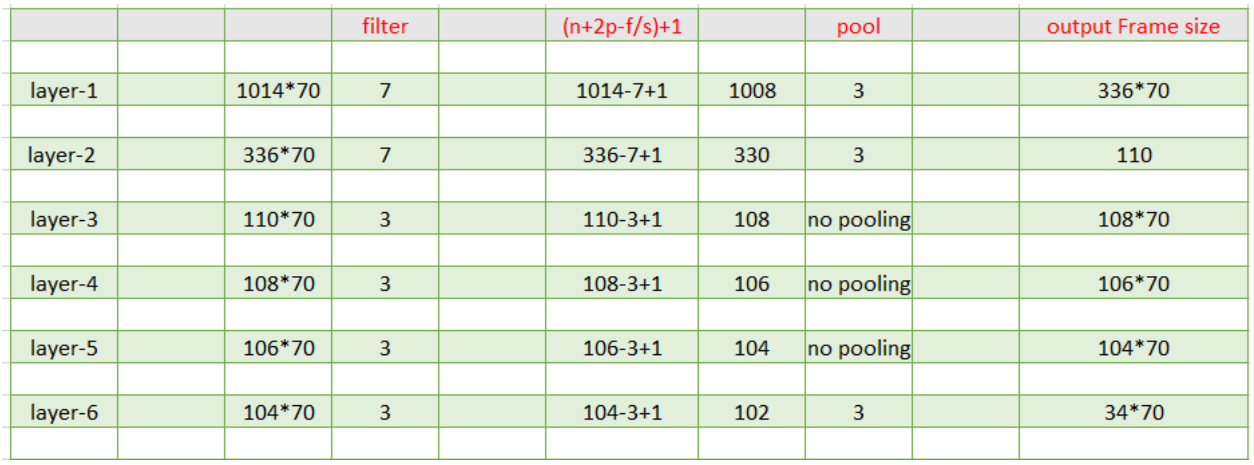
Kernel represents filter size

Pool represents shape of pooling.

The convolutional layers have stride 1 and pooling layers are all non-overlapping ones

Mathematical calculation :

(it is done in excel)



			filter		(n+2p-f/s)+1		pool		output Frame size
layer-1		1014*70	7		1014-7+1	1008	3		336*70
layer-2		336*70	7		336-7+1	330	3		110
layer-3		110*70	3		110-3+1	108	no pooling		108*70
layer-4		108*70	3		108-3+1	106	no pooling		106*70
layer-5		106*70	3		106-3+1	104	no pooling		104*70
layer-6		104*70	3		104-3+1	102	3		34*70

input dimension the first fully-connected layer has 34*70 .

generalizing the formula for input dimension first fully connected layer accepts is :

= (input feature length-96)/27 * #alphabhates(70)

= (1014-96)/27 = 34*70

Inserted 2 dropout modules in between the 3 fully-connected layers to regularize. They have dropout probability of 0.5.

Data Augmentation :

Purpose : To increase model's Generalization power.

Process followed :

1. For images : rotation, blur, cropping but in case of text the exact order of characters may form rigorous syntactic and semantic meaning. So rotation, blur, cropping not useful.
2. For text : replace the words or phrases with their synonyms.

Impact of Character-Level CNN :

- No text preprocessing (tokenization, lemmatization, stemming ...) is required.
- Its handles misspelled words and out-of-vocabulary tokens

- It handles large dataset (more than 1 million documents , it is evident from the plot provided in the research paper . for small datasets n-grams TF-IDF working better)
- No syntax and semantic information is required in prior.