

# Automated Poem Generator Using Recurrent Neural Network

ANSHUL GUPTA A20369831

PRATEEK AGRAWAL A20358932

# Introduction

- ▶ Poets Intuition behind writing a piece of work.
- ▶ Training a machine to write poem on it own.
- ▶ Interesting use cases for the problem.
- ▶ Training different RNNs on poems to generate automated text for poets such as Williams Wentworth, Williams Shakespeare.

# Approach

- ▶ Difference between feed forward neural network and Recurrent Neural Network.
- ▶ Difference between Vanilla RNNs and LSTM.

```
def cell(x,cPrev,hPrev):  
  
    x = tf.reshape(x,[nHiddenUnits,1])  
    hPrev = tf.reshape(hPrev,[nHiddenUnits,1])  
    cPrev = tf.reshape(cPrev,[nHiddenUnits,-1])  
  
    ic = tf.reshape(tf.concat([hPrev,x], axis = 0),[2*nHiddenUnits,-1])  
    ib = tf.reshape(biases['i'],[nHiddenUnits,-1])  
    i = tf.sigmoid(tf.matmul(weights['i'],ic) + ib)  
  
    fc = tf.reshape(tf.concat([hPrev,x], axis = 0),[2*nHiddenUnits,-1])  
    fb = tf.reshape(biases['f'],[nHiddenUnits,-1])  
    f = tf.sigmoid(tf.matmul(weights['f'],fc) + fb)  
  
    oc = tf.reshape(tf.concat([hPrev,x], axis = 0),[2*nHiddenUnits,-1])  
    ob = tf.reshape(biases['o'],[nHiddenUnits,-1])  
    o = tf.sigmoid(tf.matmul(weights['o'],oc) + ob)  
  
    gc = tf.reshape(tf.concat([hPrev,x], axis = 0),[2*nHiddenUnits,-1])  
    gb = tf.reshape(biases['g'],[nHiddenUnits,-1])  
    g = tf.tanh(tf.matmul(weights['g'],gc) + gb)  
  
    cCurrent = tf.add(tf.multiply(f,cPrev) , tf.multiply(i,g))  
    hCurrent = tf.multiply(o,tf.tanh(cCurrent))  
  
    return cCurrent,hCurrent
```

# Baseline Experiment

- ▶ Data has been collected by scrapping POETHUNTER.COM for poems and Limericks using Beautiful soup.
- ▶ Baseline model of n-gram character level language model.
- ▶ Accuracy Calculation.
- ▶ 14% Accurate for Base Line model.

fo nd?  
ARes sefor:  
Shor h sthid he, ftakevenor'sean't  
Erondu fomof: is s yoof  
LELIusait fre d orisif d bery ceourdw, meatilye te sewresw Bueweve my G he  
Youseead,  
  
Yot ait.  
NR:  
O, Eds saringucriseshe'Pr'dit tanin aint r theath thowhake hacede itho and, ain an tansoncono LI muga cofanthend Ishog blceme hesp wntousetho,  
HE: m.  
  
Wifotifombamy;  
Asthesthtroworanthn s buplataurengo prabl cato thee is, t d, hy s the and ano KERInoninong?  
LUToloukst s me y orthe.  
I ks rowsee f m rer  
Gorss, hane y we.

# RNN Experiments

- ▶ We had used Adam Optimizer with learning rate of 0.01.
- ▶ We use clipping to mitigate exploding and vanishing gradient problem.
- ▶ We use LSTM cells with 512 nodes in each unrolled for 128 time steps.

```
loss = tf.reduce_mean(tf.nn.softmax_cross_entropy_with_logits(logits = results, labels = y))
optimizer = tf.train.AdamOptimizer(lr)
dVar = optimizer.compute_gradients(loss)
dVarClipped = [(tf.clip_by_value(grad, -clipValue, clipValue), var) for grad, var in dVar]
train = optimizer.apply_gradients(dVarClipped)
```

# Three Hidden Layer

- ▶ We trained a three layered RNNs on Nvidia GPU for 160000 iteration.
- ▶ Text Sampled does not looks like anything close to a poem.
- ▶ Accuracy : 15.7%

```
rl,lesspt dhr : b c btl oni tn, bI s C h bANerN nutrl, l sc 't r nctrr,srr , Hfr r et t rh Gs 'N rl ,b,pterwsut brllm I lh, rrt br trrrlnn:Yrb, l,rl r,rc l,m rm
, t ret rrwu e rb btecrNne , rt,g rle n nl bGmn rrtrt g tnr rtNt
s trbr,t m,lr,s s tll' tlr bl r d d C r rc , tesltt GrMr r l h serllsr hl m cr l t:, ltderG b rtllbsnnlmw lllrhl' s strtreC rl tkrcpl tlrr t l trl lr d Ctrt -: rlte ls rttrbrn,rht
crl h' shet bs ll shlo m ?ri ltr l el'tu 'oteIs tlcu rg'N t er ' Crl,,trr lrrle l'Cc l tr r ,te I T,l tN ,nr:ln 'u :rblrp S rlmnp: nNr r, Ne r,n bfe r ,n bNptl rC 'lts Cret Nrrrmlr s
ch 'Nh e l c ss Nl,tn tr.Ut i Cth r rtl rll,r C utr ie CN m d ; t,tbclr smNt
tt :c R r're t r rrTs
l r rtrr rntl td bNi
irbt: et: m dcrN rr l rr rsr l Cr :lB Wrsu,l r n surrbl' rr WC? r ,htr ,: Ntr sn l l, tNnrstC r:s :N rn
r:e D rlcSlsrr rp ,b lt r NCl,rr 's t r tNlcSsrh: b?ittNr l'
lerN th, cr,
```

# Two Hidden Layer

- ▶ We trained a two layered RNNs on Nvidia GPU for 200000 iteration.
- ▶ Text Sampled does not looks like anything close to a poem but the model understands that every poem starts with a tilde which ends on a colon ( : )
- ▶ Accuracy : 22%

your the mised  
ow will we Jan.

Thy bod Sill, beat the nobe my ou aby therpamer fidame to bewit me glo;  
And non and ste ly ungle high arie suck.

Istrublif Warwick, fa eave, would in the by;  
Rice, like,  
Wh's thou brin nourst muich thinse dathavierge: dour uin farser;  
We hasth ueel:  
And hems ter inge agarat nawe swat mear we, ling e wedwir stralie!  
To ward's forow smpod sours will of hat se she, fnock whantrak's is Hun ffuske  
CLAnd meard uis for geid oursed ars, beeigghstlen.

GLOUCESTER:  
Wer EDWARD Wardereas you whr stariter, lostre benty welf a shake upeas welld,  
Cliart uncle-from ksee sens I whit swisentet, quid wild t wringe brows  
I Hreave  
Titeant went,

Nod muetend loves to Rich afrouble ee,  
And wis courtugenguent onwelss:  
Ther Jontoul, aurse liou thou pin, Warwive this dlaie!  
Foral no our with iflm ward he, isck, Call uspe ot ay bursaine at thin my avi

F0 Exly singe tall, we not thrn bless Ofoord noteet me I

# One Hidden Layer

- ▶ We trained a One layered RNNs on Nvidia GPU for 350000 iteration.
- ▶ Text Sampled looks similar to the poems it was trained on. The model has start understanding words and generating short poems.
- ▶ Accuracy : 40%

|whate of thrat that Romans  
As cown'd not to a eness no not been'ds  
Have sport, but of read given men: have  
Sprationiols  
And your gread Rome, and sent. Maught hereosion no his no stitar; and am it  
And you, notly not as in havals, infict's the my before powet to have beforde  
To thl-his pain fall your hout save  
Mospirtain'd,  
That I to bandity, no at highter's charius and  
To urterant Marcius!

BRUTUS:  
Thirds to IUS:  
Hath the not thy gate?

Like arm away with rough bounts and affection. We he sat commost your actren and good spate,  
And to said unlactiven and it,  
What hop.

SICINIUS:  
I noblehanced invesy genenen's the trink in proicit to laus hows  
That than tay,  
I charge, what cerns: senat: ned huries;' her;  
Hear cause the loved not and mourious lions,  
That yoursland.

Thir, the give mother of my end pale.

BRUTUS:  
How sword mocknost water aboutfection't, whom  
therefore divation!

CORIOLANUS:  
Like and for your engreat obbodess and yould  
He possius;  
That I havoldish'd shall  
able and of his th

# Conclusion

- ▶ Recurrent layer with one layer is more effective in generating new data.
- ▶ The reason behind this might be that in complex models with multiple layers each layer has its own cell state and hidden state. This limits the data transfer from layer to layer of the cell state resulting in a model that cannot remember data from long time dependency.
- ▶ Model with one hidden layer has the capability to remember data from infinity and hence generalize better on the unseen data.
- ▶ We can also experiment with different combination of learning Rate, Clipping Value and may use a different optimizer.



THANK YOU