**Report for NLP - CSE 628 - HW3 - Gourab Bhattacharyya - 170048888**

# 1 Configuration and experiment details (Code in DependencyParser.py in ‘forward_pass()’ module):

| Configuration | Updated files | Loss values | Accuracy |
|---|---|---|---|
| Default configuration with cube non-linearity (Best Configuration) | DependencyParser.py<br>ParsingSystem.py | Average loss at step 0 : 4.51575613022 | |
| | | Average loss at step 100 : 2.2349341011 | |
| | | Average loss at step 200 : 1.11413907409 | Testing on dev set at step 200<br>UAS: 43.7295909465<br>UASnoPunc: 47.0553326174<br>LAS: 35.1895705063<br>LASnoPunc: 38.3484994065<br>UEM: 1.47058823529<br>UEMnoPunc: 1.47058823529<br>ROOT: 8.23529411765 |
| | | Average loss at step 300 : 0.786763839722 | |
| | | Average loss at step 400 : 0.621368857622 | Testing on dev set at step 400<br>UAS: 53.5683126854<br>UASnoPunc: 57.3164528345<br>LAS: 47.3066281128<br>LASnoPunc: 50.8619227943<br>UEM: 2.82352941176<br>UEMnoPunc: 2.88235294118<br>ROOT: 19.0 |
| | | Average loss at step 500 : 0.527416317463 | |
| | | Average loss at step 600 : 0.463749989867 | Testing on dev set at step 600<br>UAS: 67.4950769001<br>UASnoPunc: 70.6409314418<br>LAS: 62.5246154997<br>LASnoPunc: 65.4411349121<br>UEM: 8.35294117647<br>UEMnoPunc: 8.88235294118<br>ROOT: 53.8823529412 |
| | | Average loss at step 700 : 0.425270348489 | |

| | | | |
|---|---|---|---|
| | | | |
| | | Average loss at step  800 :<br>0.389575449526 | Testing on dev set at step  800<br>UAS: 69.9753221826<br>UASnoPunc: 73.0373594077<br>LAS: 65.4859535858<br>LASnoPunc: 68.2699372633<br><br>UEM: 10.4117647059<br>UEMnoPunc: 11.1764705882<br>ROOT: 58.3529411765 |
| | | Average loss at step  900 :<br>0.372784814835 | |
| | | Average loss at step  1000<br>:  0.346725863516 | Testing on dev set at step  1000<br>UAS: 71.4136151756<br>UASnoPunc: 74.4475216187<br>LAS: 67.2682403968<br>LASnoPunc: 69.9683490646<br><br>UEM: 11.5882352941<br>UEMnoPunc: 12.4117647059<br>ROOT: 59.9411764706 |
| | | | |
| With sigmoid non-linearity | DependencyParser.py | Average loss at step  0 :<br>5.18698453903 | |
| | | Average loss at step  100 :<br>2.39680196524 | |
| | | Average loss at step  200 :<br>2.15802170992 | Testing on dev set at step 200<br>UAS: 16.4668345091<br>UASnoPunc: 16.2889278246<br>LAS: 7.12914724431<br>LASnoPunc: 7.69230769231<br>UEM: 0.588235294118<br>UEMnoPunc: 0.588235294118<br>ROOT: 2.58823529412 |
| | | Average loss at step  300 :<br>1.95886603832 | |
| | | Average loss at step  400 :<br>1.76847328544 | Testing on dev set at step 400<br>UAS: 18.1992671436<br>UASnoPunc: 18.4479737749<br>LAS: 8.64222150211<br>LASnoPunc: 9.58853783982<br>UEM: 0.647058823529<br>UEMnoPunc: 0.647058823529<br>ROOT: 2.64705882353 |
| | | Average loss at step  500 :<br>1.62100197911 | |
| | | Average loss at step  600 :<br>1.49606019855 | Testing on dev set at step 600<br>UAS: 30.6129571005 |

| | | | |
|---|---|---|---|
| | | | UASnoPunc: 32.7332843497<br>LAS: 18.6903307825<br>LASnoPunc: 21.0224382524<br>UEM: 1.05882352941<br>UEMnoPunc: 1.05882352941<br>ROOT: 3.17647058824 |
| | | Average loss at step 700 : 1.40109686017 | |
| | | Average loss at step 800 : 1.31042621255 | Testing on dev set at step 800<br>UAS: 40.3444923598<br>UASnoPunc: 43.8845871249<br>LAS: 27.2228730962<br>LASnoPunc: 30.4385915334<br>UEM: 1.41176470588<br>UEMnoPunc: 1.41176470588<br>ROOT: 5.35294117647 |
| | | Average loss at step 900 : 1.23885253072 | |
| | | Average loss at step 1000 : 1.16462132931 | Testing on dev set at step 1000<br>UAS: 43.0216616397<br>UASnoPunc: 46.8999039168<br>LAS: 31.8119500461<br>LASnoPunc: 35.3275306618<br>UEM: 1.47058823529<br>UEMnoPunc: 1.47058823529<br>ROOT: 6.11764705882 |
| | | | |
| With Relu non-linearity | DependencyParser.py | Average loss at step 0 : 4.48135995865 | |
| | | Average loss at step 100 : 2.1972854805 | |
| | | Average loss at step 200 : 1.54496123672 | Testing on dev set at step 200<br>UAS: 31.6225041753<br>UASnoPunc: 33.6093370259<br>LAS: 21.0284916619<br>LASnoPunc: 23.447125982<br>UEM: 1.11764705882<br>UEMnoPunc: 1.11764705882<br>ROOT: 3.52941176471 |
| | | Average loss at step 300 : 1.22418649316 | |
| | | Average loss at step 400 : 1.01173090756 | Testing on dev set at step 400<br>UAS: 43.9688909938<br>UASnoPunc: 47.3492341604<br>LAS: 33.7338285515<br>LASnoPunc: 36.949641101 |

| | | | |
|---|---|---|---|
| | | | UEM: 1.82352941176<br>UEMnoPunc: 1.94117647059<br>ROOT: 9.05882352941 |
| | | Average loss at step  500 : 0.881880100965 | |
| | | Average loss at step  600 : 0.771893889308 | Testing on dev set at step  600<br>UAS: 51.9605154922<br>UASnoPunc: 55.3043576556<br>LAS: 43.550115911<br>LASnoPunc: 46.3997060985<br><br>UEM: 3.35294117647<br>UEMnoPunc: 3.41176470588<br>ROOT: 27.5882352941 |
| | | Average loss at step  700 : 0.703143923283 | |
| | | Average loss at step  800 : 0.63053144753 | Testing on dev set at step  800<br>UAS: 57.5391978463<br>UASnoPunc: 60.7980557283<br>LAS: 50.4449485256<br>LASnoPunc: 53.3035663822<br><br>UEM: 4.11764705882<br>UEMnoPunc: 4.35294117647<br>ROOT: 37.8235294118 |
| | | Average loss at step  900 : 0.592157555819 | |
| | | Average loss at step  1000 :  0.540369702578 | Testing on dev set at step  1000<br>UAS: 62.0335518608<br>UASnoPunc: 65.3902673374<br>LAS: 56.0360944238<br>LASnoPunc: 58.9894308484<br><br>UEM: 5.47058823529<br>UEMnoPunc: 5.47058823529<br>ROOT: 44.6470588235 |
| | | | |
| With tanh non-linearity | DependencyParser.py | Average loss at step  0 : 4.58066225052 | |
| | | Average loss at step  100 : 2.03260846138 | |
| | | Average loss at step  200 : 1.46153076768 | Testing on dev set at step  200<br>UAS: 39.4296682205<br>UASnoPunc: 42.6891991183<br>LAS: 25.9939676446 |

| | | | |
|---|---|---|---|
| | | | LASnoPunc: 28.3643248742<br>UEM: 1.52941176471<br>UEMnoPunc: 1.58823529412<br>ROOT: 11.2941176471 |
| | | Average loss at step  300 : 1.17652178168 | |
| | | Average loss at step  400 : 0.990951049924 | Testing on dev set at step  400<br>UAS: 49.7569608894<br>UASnoPunc: 52.4359916351<br>LAS: 39.2576713114<br>LASnoPunc: 40.9483976714<br>UEM: 3.05882352941<br>UEMnoPunc: 3.23529411765<br>ROOT: 39.0 |
| | | Average loss at step  500 : 0.864987668991 | |
| | | Average loss at step  600 : 0.770465891957 | Testing on dev set at step  600<br>UAS: 55.1262557021<br>UASnoPunc: 58.2122873453<br>LAS: 46.5812498442<br>LASnoPunc: 49.0137342452<br>UEM: 3.76470588235<br>UEMnoPunc: 4.0<br>ROOT: 38.6470588235 |
| | | Average loss at step  700 : 0.698352285624 | |
| | | Average loss at step  800 : 0.640565508008 | Testing on dev set at step  800<br>UAS: 59.2068200513<br>UASnoPunc: 62.4455999548<br>LAS: 52.207293666<br>LASnoPunc: 54.9963262307<br>UEM: 4.47058823529<br>UEMnoPunc: 4.58823529412<br>ROOT: 43.2352941176 |
| | | Average loss at step  900 : 0.59782643795 | |
| | | Average loss at step  1000 :  0.552508433759 | Testing on dev set at step  1000<br>UAS: 59.9471545729<br>UASnoPunc: 63.3612163002<br>LAS: 53.8001346063<br>LASnoPunc: 56.8812524727<br>UEM: 4.76470588235 |

| | | | |
|---|---|---|---|
| | | | UEMnoPunc: 4.82352941176<br>ROOT: 40.3529411765 |
| | | | |
| With 2 hidden layers.<br>1 - With cube and 2 – with tanh non-linearity | DependencyParser.py | Average loss at step  0 : 4.51270771027 | |
| | | Average loss at step  100 : 2.00896693587 | |
| | | Average loss at step  200 : 1.08066220343 | Testing on dev set at step  200<br>UAS: 20.0912331431<br>UASnoPunc: 21.78827785<br>LAS: 0.902360595259<br>LASnoPunc: 0.997569660318<br><br>UEM: 0.470588235294<br>UEMnoPunc: 0.470588235294<br>ROOT: 1.17647058824 |
| | | Average loss at step  300 : 0.766031338573 | |
| | | Average loss at step  400 : 0.610648165345 | Testing on dev set at step  400<br>UAS: 16.9254929332<br>UASnoPunc: 18.1879839485<br>LAS: 0.897375177606<br>LASnoPunc: 1.00887356582<br><br>UEM: 0.588235294118<br>UEMnoPunc: 0.588235294118<br>ROOT: 2.52941176471 |
| | | Average loss at step  500 : 0.52376113236 | |
| | | Average loss at step  600 : 0.464983358085 | Testing on dev set at step  600<br>UAS: 15.3849988783<br>UASnoPunc: 16.4302266433<br>LAS: 0.987112695366<br>LASnoPunc: 1.10778273894<br><br>UEM: 0.529411764706<br>UEMnoPunc: 0.529411764706<br>ROOT: 1.88235294118 |
| | | Average loss at step  700 : 0.426696145535 | |
| | | Average loss at step  800 : 0.393317946196 | Testing on dev set at step  800<br>UAS: 15.4722436872<br>UASnoPunc: 16.3963149268<br>LAS: 1.03447416307<br>LASnoPunc: 1.13604250268 |

| | | | |
|---|---|---|---|
| | | | UEM: 0.823529411765<br>UEMnoPunc: 0.823529411765<br>ROOT: 1.58823529412 |
| | | Average loss at step 900 : 0.376706490517 | |
| | | Average loss at step 1000 : 0.352690368891 | Testing on dev set at step 1000<br>UAS: 14.8166612658<br>UASnoPunc: 15.6728649748<br>LAS: 1.10427001022<br>LASnoPunc: 1.22082179393<br><br>UEM: 0.823529411765<br>UEMnoPunc: 0.823529411765<br>ROOT: 1.94117647059 |
| | | | |
| With 3 hidden layers.<br>1 - With cube, 2 – with relu and 3 - with tanh non-linearity | DependencyParser.py | Average loss at step 0 : 4.44890880585 | |
| | | Average loss at step 100 : 2.08809595823 | |
| | | Average loss at step 200 : 1.13550832093 | Testing on dev set at step 200<br>UAS: 14.8789789865<br>UASnoPunc: 15.2828802351<br>LAS: 0.620684497844<br>LASnoPunc: 0.703668117335<br><br>UEM: 0.705882352941<br>UEMnoPunc: 0.705882352941<br>ROOT: 8.52941176471 |
| | | Average loss at step 300 : 0.7579912889 | |
| | | Average loss at step 400 : 0.595848079324 | Testing on dev set at step 400<br>UAS: 16.1402896528<br>UASnoPunc: 16.5404397219<br>LAS: 0.418775082883<br>LASnoPunc: 0.474764030973<br><br>UEM: 0.823529411765<br>UEMnoPunc: 0.882352941176<br>ROOT: 5.05882352941 |
| | | Average loss at step 500 : 0.511559298038 | |
| | | Average loss at step 600 : 0.450352653563 | Testing on dev set at step 600<br>UAS: 17.5835680634<br>UASnoPunc: 18.0834228226<br>LAS: 0.438716753496<br>LASnoPunc: 0.488893912847 |

| | | | |
|---|---|---|---|
| | | | UEM: 0.764705882353<br>UEMnoPunc: 0.823529411765<br>ROOT: 5.76470588235 |
| | | Average loss at step  700 : 0.415178941488 | |
| | | Average loss at step  800 : 0.382003716528 | Testing on dev set at step  800<br>UAS: 18.9495725004<br>UASnoPunc: 19.677273498<br>LAS: 0.51848343595<br>LASnoPunc: 0.584977109591<br><br>UEM: 0.705882352941<br>UEMnoPunc: 0.823529411765<br>ROOT: 6.70588235294 |
| | | Average loss at step  900 : 0.364776985049 | |
| | | Average loss at step  1000 :  0.343047633469 | Testing on dev set at step  1000<br>UAS: 20.2657227609<br>UASnoPunc: 21.1919968349<br>LAS: 0.53343968891<br>LASnoPunc: 0.593455038716<br><br>UEM: 0.823529411765<br>UEMnoPunc: 0.941176470588<br>ROOT: 5.64705882353 |
| | | | |
| Without using gradient clipping | DependencyParser.py | Average loss at step  0 : 4.61847877502 | |
| | | Average loss at step  100 : 2.21099841356 | |
| | | Average loss at step  200 : 1.58685922503 | Testing on dev set at step  200<br>UAS: 35.3914799212<br>UASnoPunc: 38.3626292884<br>LAS: 22.4518284019<br>LASnoPunc: 24.4814333352<br><br>UEM: 1.35294117647<br>UEMnoPunc: 1.35294117647<br>ROOT: 11.1176470588 |
| | | Average loss at step  300 : 1.26422503352 | |
| | | Average loss at step  400 : 1.04455502331 | Testing on dev set at step  400<br>UAS: 48.0968168108<br>UASnoPunc: 50.8647487707<br>LAS: 37.2460552883 |

| | | | |
|---|---|---|---|
| | | | LASnoPunc: 39.0295597129<br>UEM: 2.70588235294<br>UEMnoPunc: 2.94117647059<br>ROOT: 34.0 |
| | | Average loss at step 500 : 0.905504993796 | |
| | | Average loss at step 600 : 0.791361171007 | Testing on dev set at step 600<br>UAS: 52.7556896079<br>UASnoPunc: 55.5106539309<br>LAS: 44.5521848593<br>LASnoPunc: 46.6031763974<br>UEM: 3.35294117647<br>UEMnoPunc: 3.82352941176<br>ROOT: 42.7647058824 |
| | | Average loss at step 700 : 0.717938861847 | |
| | | Average loss at step 800 : 0.645456998348 | Testing on dev set at step 800<br>UAS: 59.3464117456<br>UASnoPunc: 62.4512519075<br>LAS: 52.7357479373<br>LASnoPunc: 55.3580512067<br>UEM: 4.41176470588<br>UEMnoPunc: 4.76470588235<br>ROOT: 47.0588235294 |
| | | Average loss at step 900 : 0.609607232809 | |
| | | Average loss at step 1000 : 0.553272984326 | Testing on dev set at step 1000<br>UAS: 62.203056061<br>UASnoPunc: 65.2263607076<br>LAS: 56.6493007952<br>LASnoPunc: 59.2748544622<br>UEM: 5.17647058824<br>UEMnoPunc: 5.52941176471<br>ROOT: 52.0588235294 |
| | | | |

## 2 What is Gradient Clipping and why it is required? :

In case of multi-layered neural network, the propagation happens in forward and backward mode. In case of backward propagation, if the gradients of the loss function get multiplied with numbers less than one then there is a chance of the gradients to be vanished this is called vanishing gradient problem. Also, if the gradients get multiplied with numbers greater than one then there is possibility for exploding gradients also. This is when they get exponentially large and create a situation called exploding gradients. Gradient clipping helps to clip the gradients between two numbers to prevent them from getting too large.

I have configured the layered neural network and captured the result which can be seen in the above table (last row).

## 3 Findings and observations:

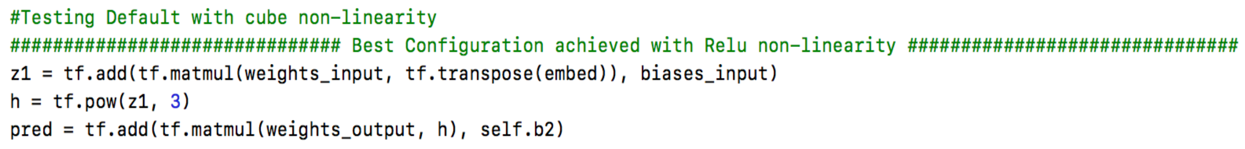
From the table above with experiment result with different configuration its observable that we got higher accuracy and lower loss with cube, Relu and tanh non-linearity and with one single hidden layer. But with sigmoid non-linearity and single hidden layer, we didn't observe much increment of the accuracy value.

If we increase the #of hidden layers with and use any non-linearity like cube or Relu or tanh, we start seeing a significant decrease in the accuracy even though the loss is not much increasing.

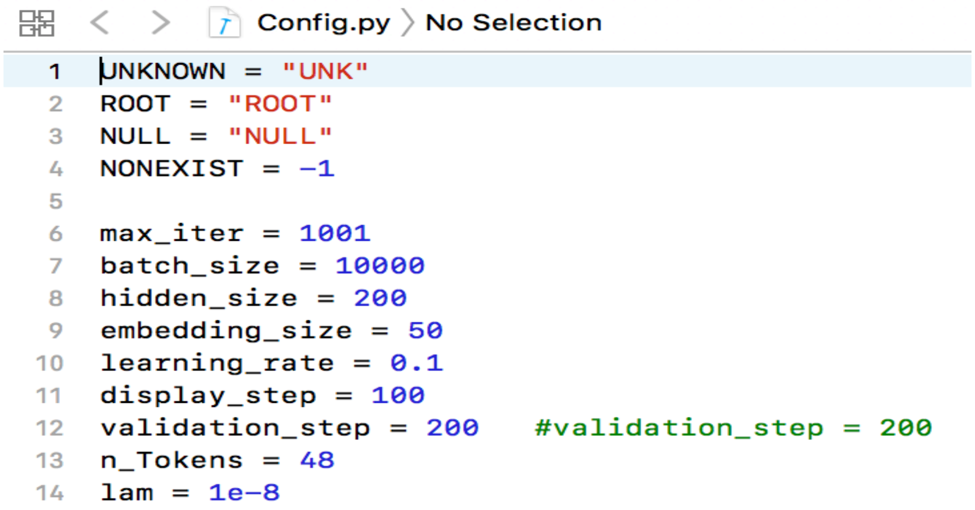
Without gradient clipping, I observed moderate loss value and accuracy.

## 4 Best Configuration:

Out of all the experiments, I have seen the best result in terms of less loss value and more accuracy with **cube non-linearity and with single hidden layer (can be found in DependencyParser.py in forward_pass() module).** I have used the default configuration given in paper and used the configuration parameters provided in the Config.py



```
#Testing Default with cube non-linearity
############################### Best Configuration achieved with Relu non-linearity ###############################
z1 = tf.add(tf.matmul(weights_input, tf.transpose(embed)), biases_input)
h = tf.pow(z1, 3)
pred = tf.add(tf.matmul(weights_output, h), self.b2)
```



```
UNKNOWN = "UNK"
ROOT = "ROOT"
NULL = "NULL"
NONEXIST = -1

max_iter = 1001
batch_size = 10000
hidden_size = 200
embedding_size = 50
learning_rate = 0.1
display_step = 100
validation_step = 200   #validation_step = 200
n_Tokens = 48
lam = 1e-8
```

## Loss Computed using Best Configuration:

```
Average loss at step  0 :  4.51575613022
Average loss at step  100 :  2.2349341011
Average loss at step  200 :  1.11413907409
Average loss at step  300 :  0.786763839722
Average loss at step  400 :  0.621368857622
Average loss at step  500 :  0.527416317463
Average loss at step  600 :  0.463749989867
Average loss at step  700 :  0.425270348489
Average loss at step  800 :  0.389575449526
Average loss at step  900 :  0.372784814835
Average loss at step  1000 :  0.346725863516
```

## Accuracy Computed using Best Configuration:

```
Testing on dev set at step  1000
UAS: 71.4136151756
UASnoPunc: 74.4475216187
LAS: 67.2682403968
LASnoPunc: 69.9683490646

UEM: 11.5882352941
UEMnoPunc: 12.4117647059
ROOT: 59.9411764706
```