Homework-3

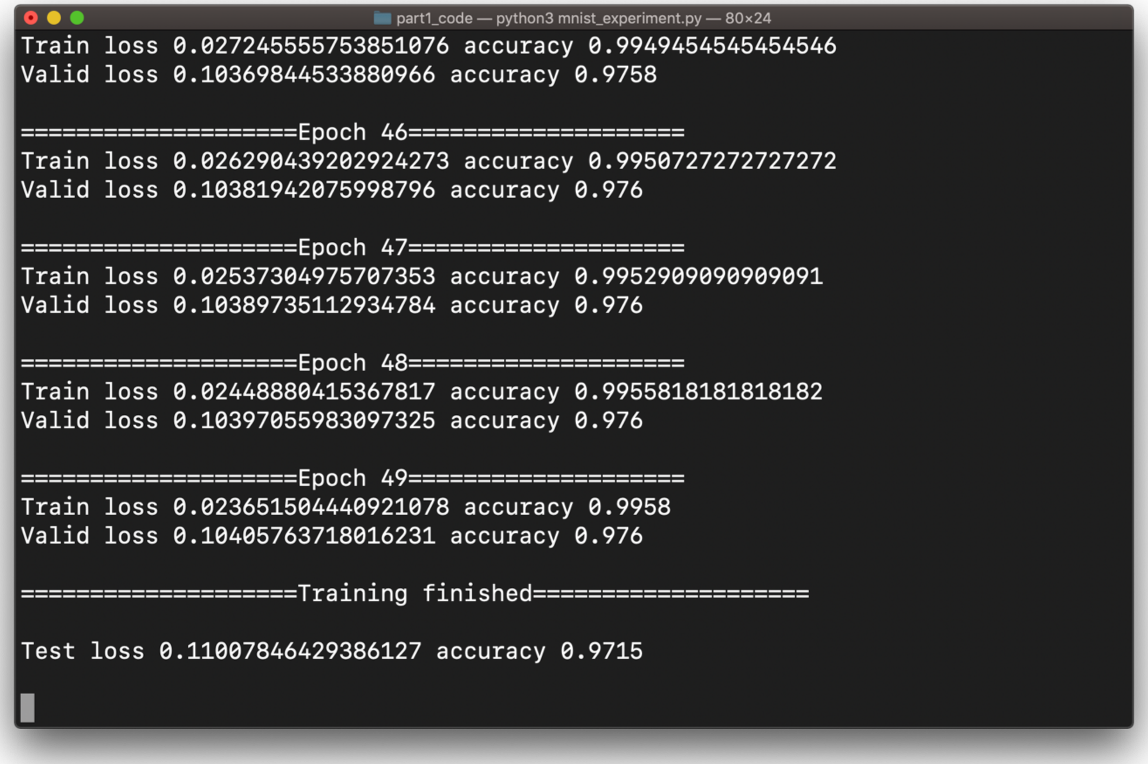
Amit Adate

Part 01:

Check attached part1_code folder, with modified fnn.py (comments added on top of every line that is changed, total lines changed – 05)

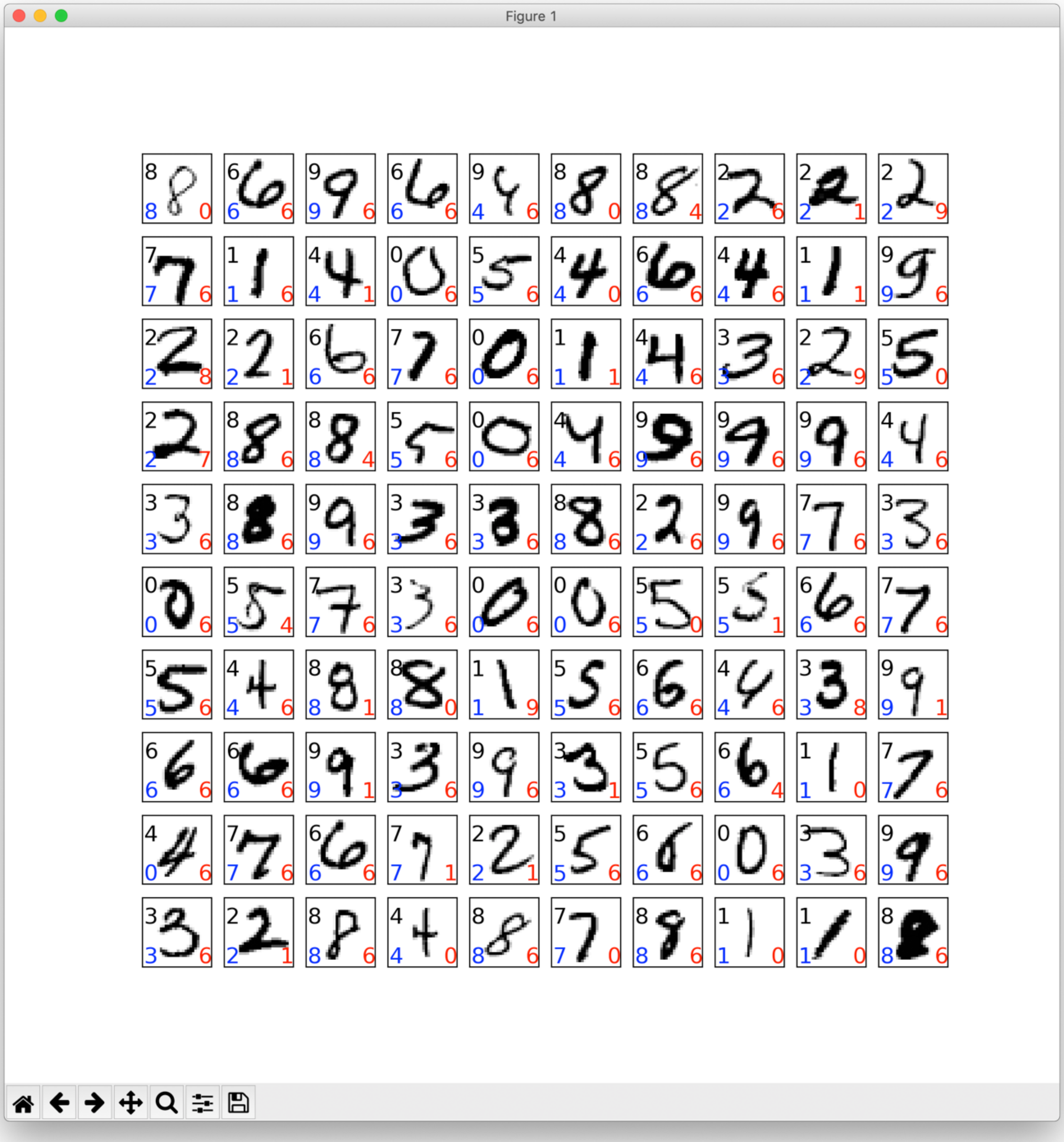
Results of running mnist_experiment.py:

Output Log:



```
Train loss 0.027245555753851076 accuracy 0.9949454545454546
Valid loss 0.10369844533880966 accuracy 0.9758

====================Epoch 46====================
Train loss 0.026290439202924273 accuracy 0.9950727272727272
Valid loss 0.10381942075998796 accuracy 0.976

====================Epoch 47====================
Train loss 0.02537304975707353 accuracy 0.9952909090909091
Valid loss 0.10389735112934784 accuracy 0.976

====================Epoch 48====================
Train loss 0.02448880415367817 accuracy 0.9955818181818182
Valid loss 0.10397055983097325 accuracy 0.976

====================Epoch 49====================
Train loss 0.023651504440921078 accuracy 0.9958
Valid loss 0.10405763718016231 accuracy 0.976

====================Training finished====================

Test loss 0.11007846429386127 accuracy 0.9715
```

Output Generated:



Part 02:

Question 01:

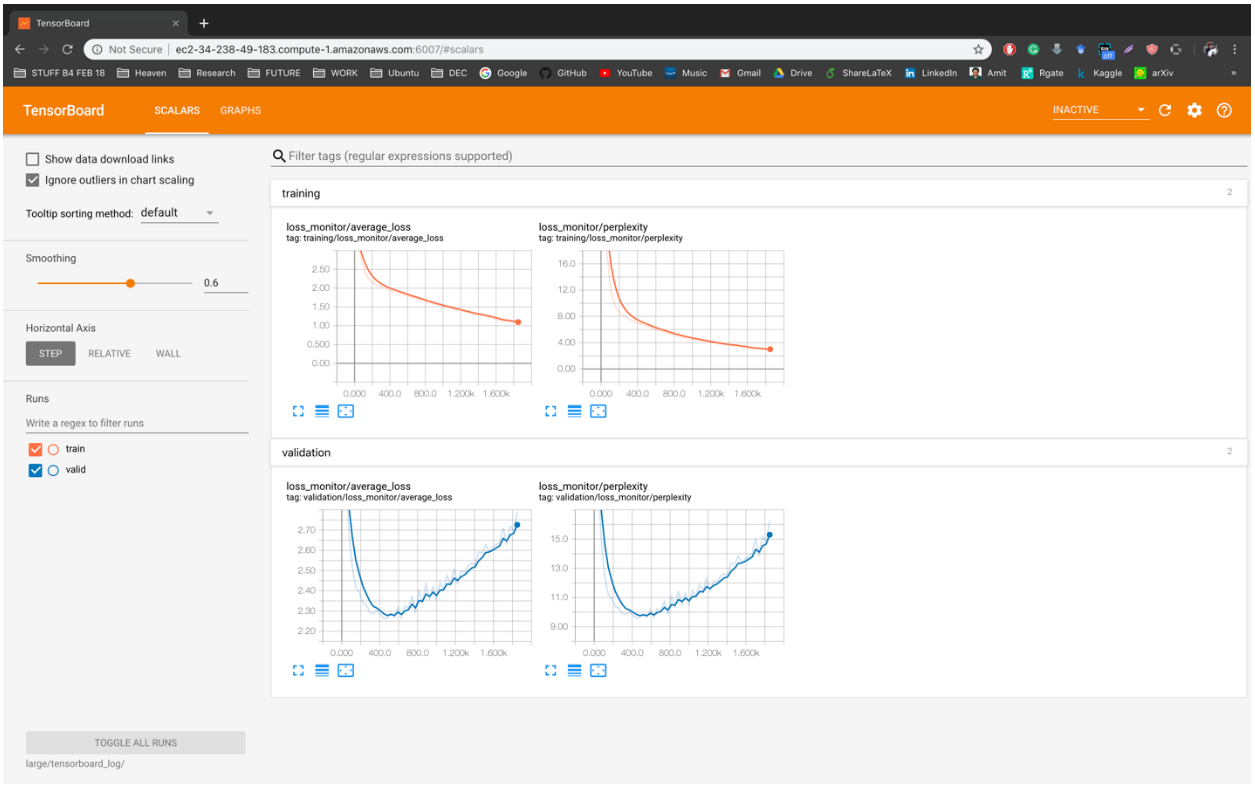
what is the difference between the curves of the two recurrent neural networks, and why does this difference make sense?

Refer the images Model-Large-Vanilla and Model-Small-Vanilla in the directory.

The key difference between the two graphs is that within the hidden size of 8, the perplexity and validation loss decreases and they tend to stabilize with the increment in the number of training steps.

Whereas in the hidden size of 256, the perplexity and validation loss initially decreases and then further increases as the vector size of 256, is too large for the data and it over fits the data.

Model-Large-Vanilla



Not Secure | ec2-34-238-49-183.compute-1.amazonaws.com:6007/#scalars

□ Show data download links
☑ Ignore outliers in chart scaling

Tooltip sorting method: default

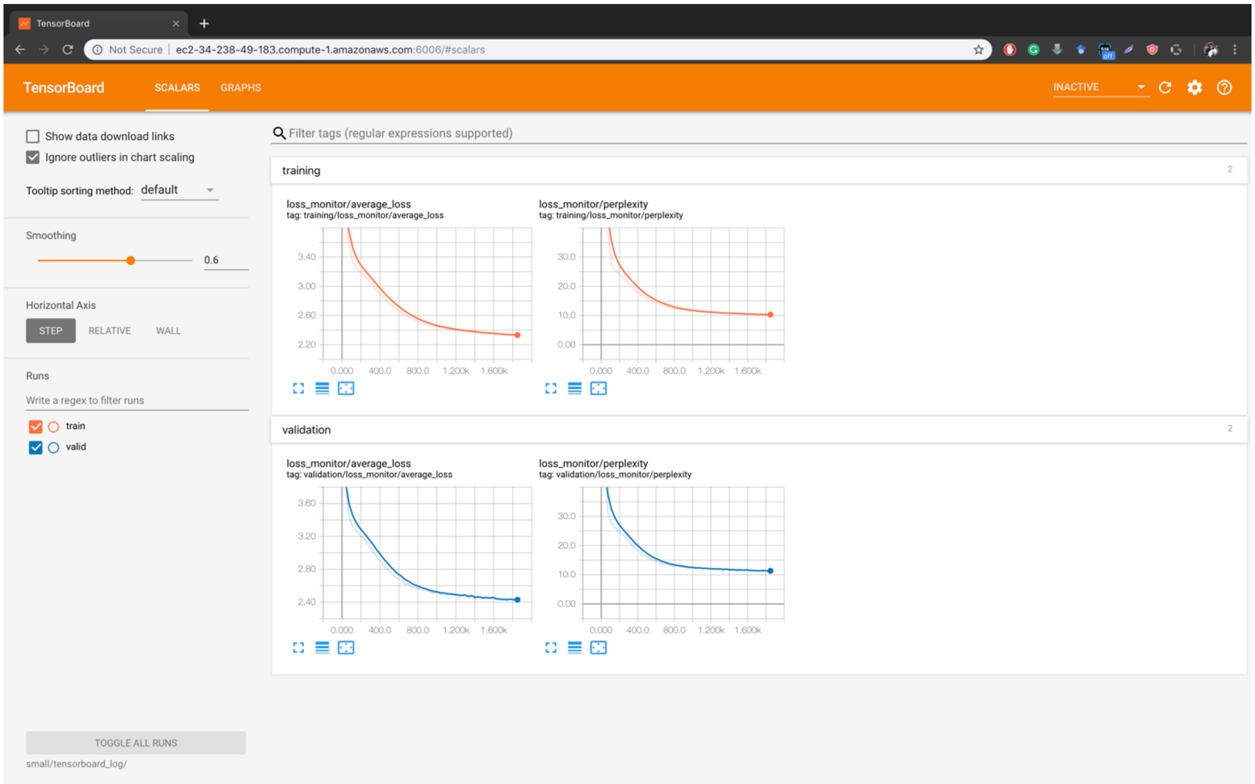
Smoothing 0.6

Horizontal Axis: STEP

Runs
☑ train
☑ valid

large/tensorboard_log/

Model-Small-Vanilla



Not Secure | ec2-34-238-49-183.compute-1.amazonaws.com:6006/#scalars

☐ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis: STEP

Runs: ☑ train ☑ valid

small/tensorboard_log/

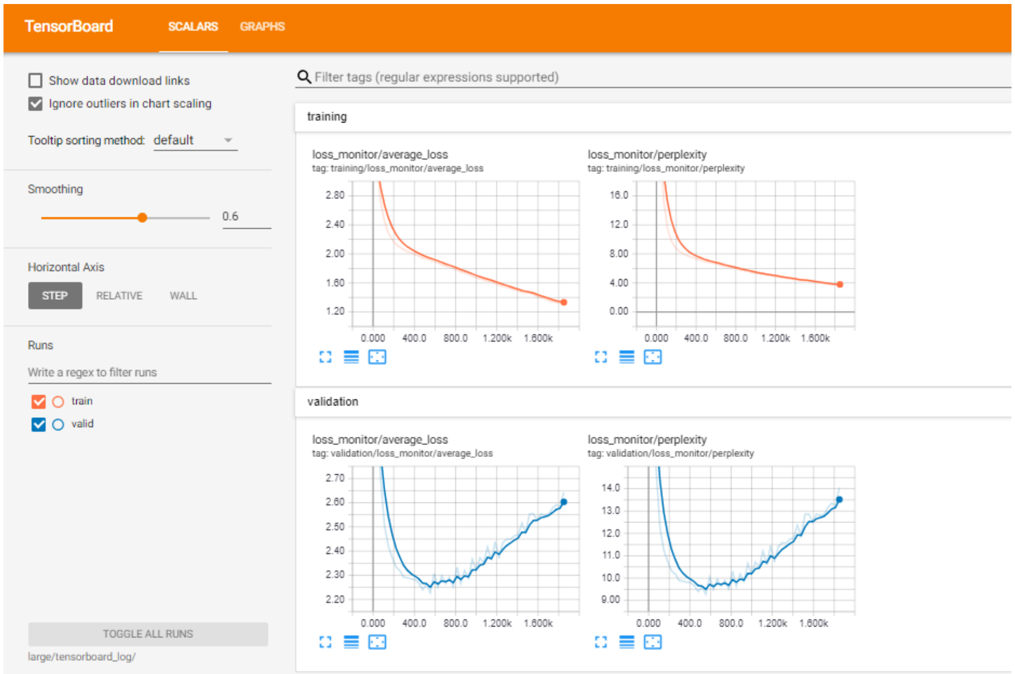
Question 02:

@Dropout: What is the difference between their learning traces, and why?

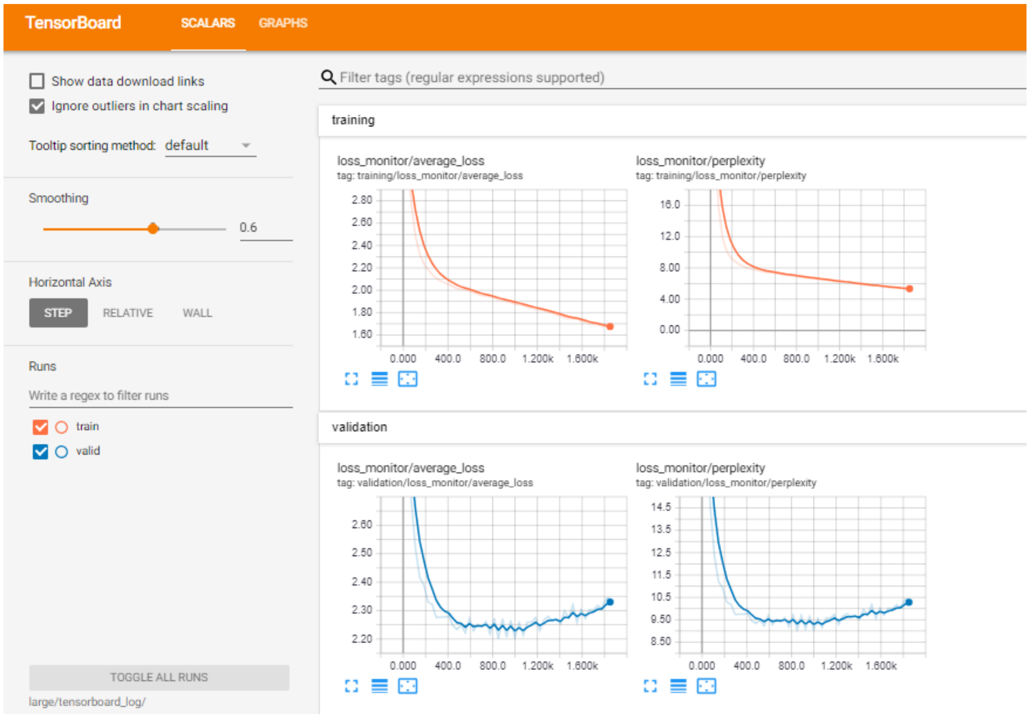
Refer the images dropout-0.1-log-final, dropout-0.1-log-final and dropout-0.1-log-final.

Dropout – Parameter	Validation - Perplexity	Test - Perplexity
0.1	9.538	8.959
0.3	9.1759	8.68049
0.5	9.0169	8.4156

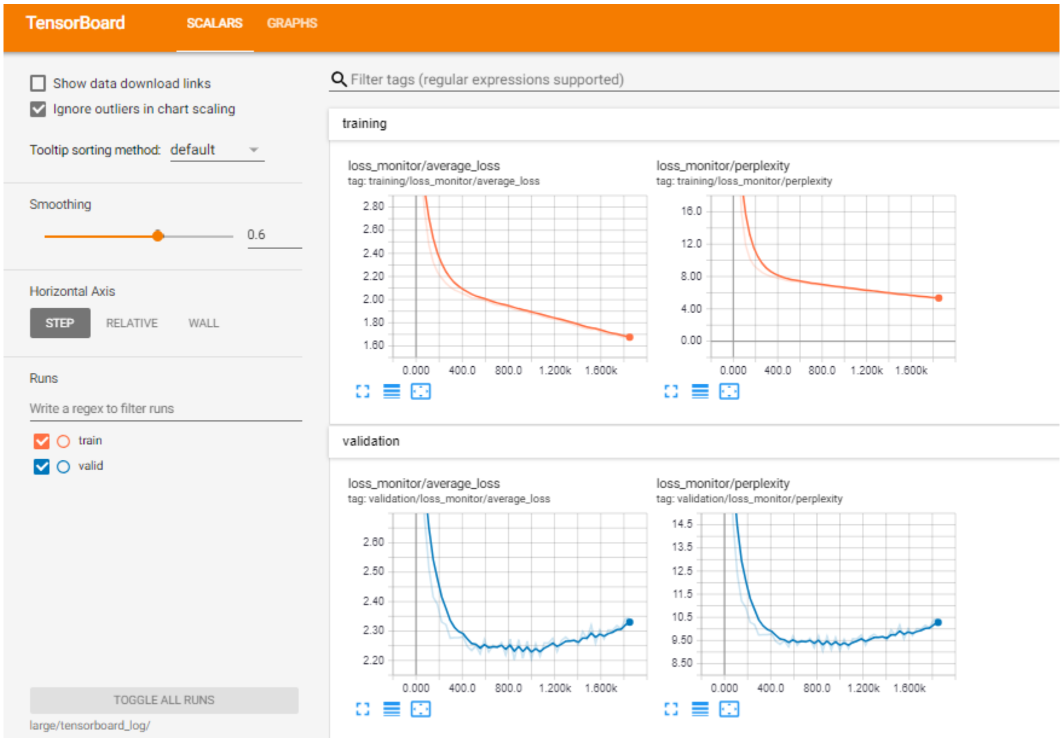
The observed difference from the learning curves is that the validation error does up after a initial low kink for a lower dropout. Hence, the experimental observation is that to attain a better / higher validation accuracy, it is advised to implement a higher dropout. From reading a bit about dropout, I have understood that it essentially provides a way to avoid overfitting. In our implementation, choosing a high dropout rate provides with preventing overfitting during training.



TensorBoard
SCALARS
GRAPHS
□ Show data download links
☑ Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
☑ ○ train
☑ ○ valid
TOGGLE ALL RUNS
large/tensorboard_log/
Filter tags (regular expressions supported)
training
loss_monitor/average_loss
tag: training/loss_monitor/average_loss
2.80
2.40
2.00
1.60
1.20
0.000
400.0
800.0
1.200k
1.600k
loss_monitor/perplexity
tag: training/loss_monitor/perplexity
16.0
12.0
8.00
4.00
0.00
0.000
400.0
800.0
1.200k
1.600k
validation
loss_monitor/average_loss
tag: validation/loss_monitor/average_loss
2.70
2.60
2.50
2.40
2.30
2.20
0.000
400.0
800.0
1.200k
1.600k
loss_monitor/perplexity
tag: validation/loss_monitor/perplexity
14.0
13.0
12.0
11.0
10.0
9.00
0.000
400.0
800.0
1.200k
1.600k



TensorBoard
SCALARS
GRAPHS
□ Show data download links
☑ Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
☑ ○ train
☑ ○ valid
TOGGLE ALL RUNS
large/tensorboard_log/
Filter tags (regular expressions supported)
training
loss_monitor/average_loss
tag: training/loss_monitor/average_loss
2.80
2.60
2.40
2.20
2.00
1.80
1.60
0.000
400.0
800.0
1.200k
1.600k
loss_monitor/perplexity
tag: training/loss_monitor/perplexity
16.0
12.0
8.00
4.00
0.00
0.000
400.0
800.0
1.200k
1.600k
validation
loss_monitor/average_loss
tag: validation/loss_monitor/average_loss
2.60
2.50
2.40
2.30
2.20
0.000
400.0
800.0
1.200k
1.600k
loss_monitor/perplexity
tag: validation/loss_monitor/perplexity
14.5
13.5
12.5
11.5
10.5
9.50
8.50
0.000
400.0
800.0
1.200k
1.600k



TensorBoard SCALARS GRAPHS

Filter tags (regular expressions supported)

☐ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.6

Horizontal Axis: STEP RELATIVE WALL

Runs

Write a regex to filter runs

☑ ○ train

☑ ○ valid

TOGGLE ALL RUNS

large/tensorboard_log/

training

loss_monitor/average_loss
tag: training/loss_monitor/average_loss

loss_monitor/perplexity
tag: training/loss_monitor/perplexity

validation

loss_monitor/average_loss
tag: validation/loss_monitor/average_loss

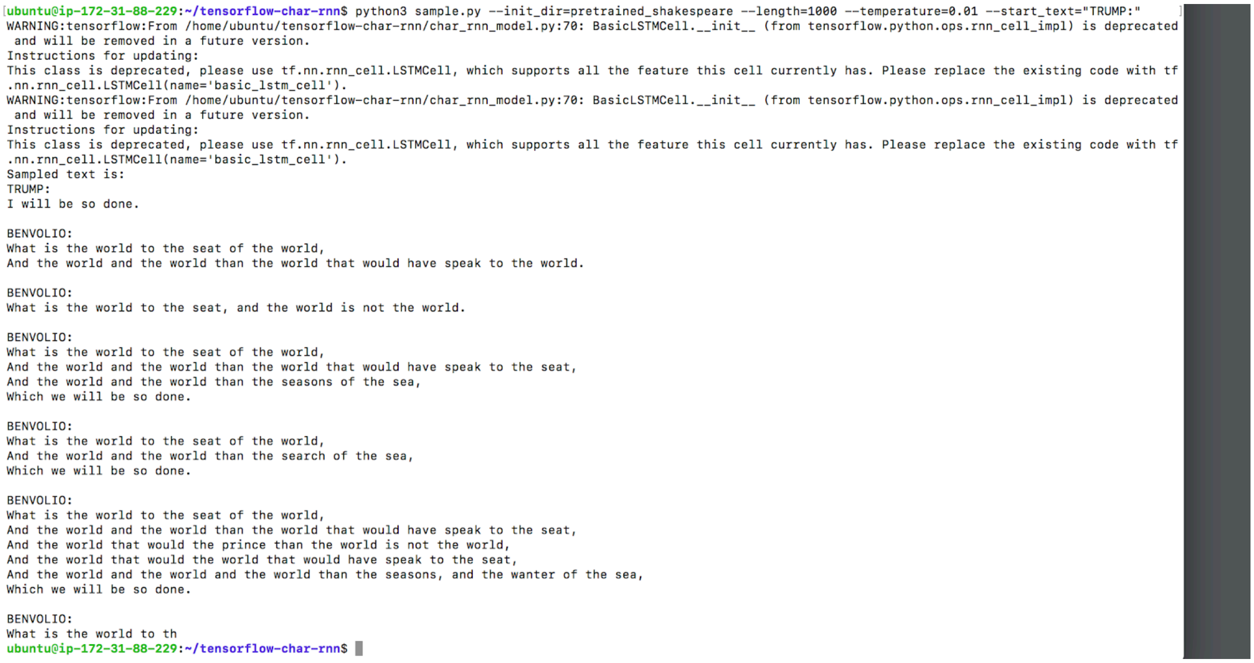
loss_monitor/perplexity
tag: validation/loss_monitor/perplexity

Question 03:

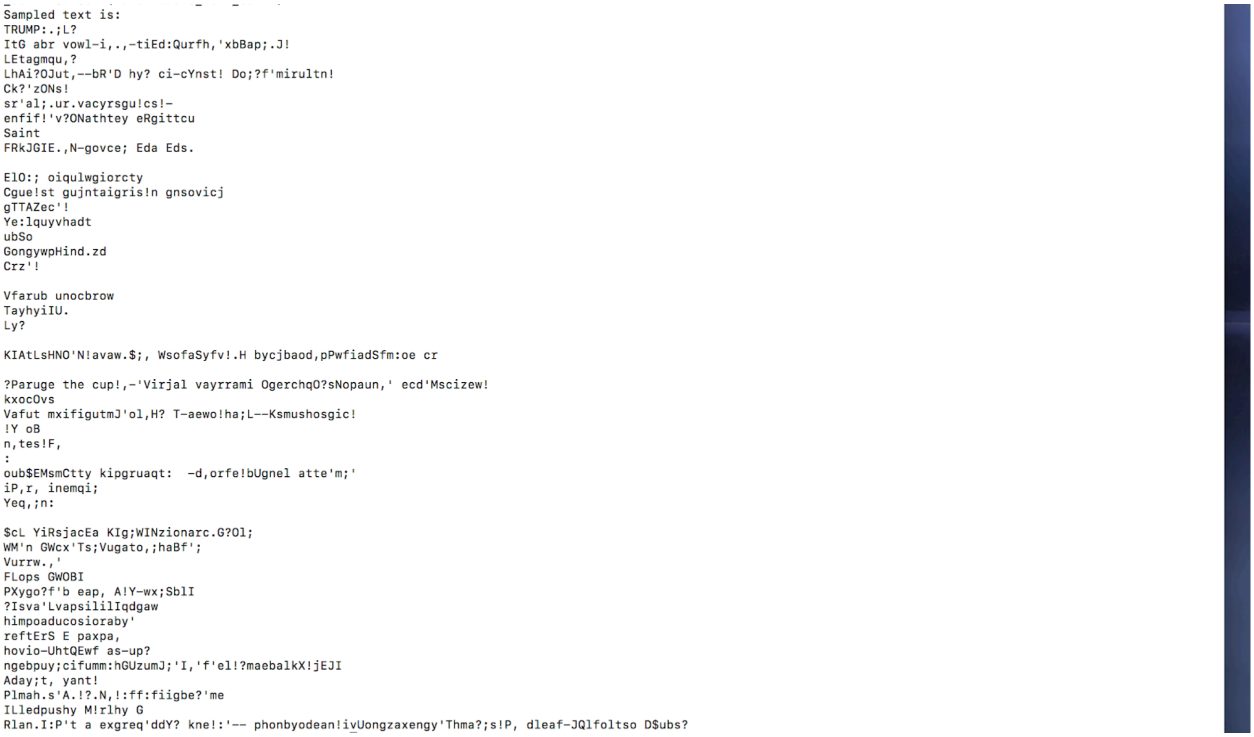
How are the samples different from the previous one (with temperature=0.5) and why? (think about how the temperature would change the shape of the distribution, and perhaps try some simple mathematical examples.)

Refer the images temperature-0.01-log-final, temperature 0.5-log-final and temperature -5.0-log-final in the directory.

As observed the samples with temperature 0.01 produced the best results, the ones with temperature 0.5 producing moderately good results and the ones with temperature 5 producing ambiguous results. After reading a bit about Temperature within LSTM, in my understanding as the temperature gets higher the probability density over the classes decreases. Hence the RNN gets fairly quickly excited by the samples, resulting in an increase in sensitivity and making more errors. I think that a lesser temperature can produce a more firm prediction.



```
ubuntu@ip-172-31-88-229:~/tensorflow-char-rnn$ python3 sample.py --init_dir=pretrained_shakespeare --length=1000 --temperature=0.01 --start_text="TRUMP:"
WARNING:tensorflow:From /home/ubuntu/tensorflow-char-rnn/char_rnn_model.py:70: BasicLSTMCell.__init__ (from tensorflow.python.ops.rnn_cell_impl) is deprecated
 and will be removed in a future version.
Instructions for updating:
This class is deprecated, please use tf.nn.rnn_cell.LSTMCell, which supports all the feature this cell currently has. Please replace the existing code with tf
.nn.rnn_cell.LSTMCell(name='basic_lstm_cell').
WARNING:tensorflow:From /home/ubuntu/tensorflow-char-rnn/char_rnn_model.py:70: BasicLSTMCell.__init__ (from tensorflow.python.ops.rnn_cell_impl) is deprecated
 and will be removed in a future version.
Instructions for updating:
This class is deprecated, please use tf.nn.rnn_cell.LSTMCell, which supports all the feature this cell currently has. Please replace the existing code with tf
.nn.rnn_cell.LSTMCell(name='basic_lstm_cell').
Sampled text is:
TRUMP:
I will be so done.

BENVOLIO:
What is the world to the seat of the world,
And the world and the world than the world that would have speak to the world.

BENVOLIO:
What is the world to the seat, and the world is not the world.

BENVOLIO:
What is the world to the seat of the world,
And the world and the world than the world that would have speak to the seat,
And the world and the world than the seasons of the sea,
Which we will be so done.

BENVOLIO:
What is the world to the seat of the world,
And the world and the world than the search of the sea,
Which we will be so done.

BENVOLIO:
What is the world to the seat of the world,
And the world and the world than the world that would have speak to the seat,
And the world that would the prince than the world is not the world,
And the world that would the world that would have speak to the seat,
And the world and the world and the world than the seasons, and the wanter of the sea,
Which we will be so done.

BENVOLIO:
What is the world to th
ubuntu@ip-172-31-88-229:~/tensorflow-char-rnn$
```



```
Sampled text is:
TRUMP:.;L?
ItG abr vowl-i,.,-tiEd:Qurfh,'xbBap;.J!
LEtagmqu,?
LhAi?OJut,--bR'D hy? ci-cYnst! Do;?f'miraltn!
Ck?'zONs!
sr'al;.ur.vacyrsgu!cs!-
enfif!'v?ONathtey eRgittcu
Saint
FRkJGIE.,N-govce; Eda Eds.

ElO:; oiqulwgiorcty
Cgue!st gujntaigris!n gnsovicj
gTTAZec'!
Ye:lquyvhadt
ubSo
GongywpHind.zd
Crz'!

Vfarub unocbrow
TayhyiIU.
Ly?

KIAtLsHNO'N!avaw.$;, WsofaSyfv!.H bycjbaod,pPwfiadSfm:oe cr

?Paruge the cup!,-'Virjal vayrrami OgerchqO?sNopaun,' ecd'Mscizew!
kxocOvs
Vafut mxifigutmJ'ol,H? T-aewo!ha;L--Ksmushosgic!
!Y oB
n,tes!F,
:
oub$EMsmCtty kipgruaqt:  -d,orfe!bUgnel atte'm;'
iP,r, inemqi;
Yeq,;n:

$cL YiRsjacEa KIg;WINzionarc.G?Ol;
WM'n GWcx'Ts;Vugato,;haBf';
Vurrw.,'
FLops GWOBI
PXygo?f'b eap, A!Y-wx;SblI
?Isva'LvapsililIqdgaw
himpoaducosioraby'
reftErS E paxpa,
hovio-UhtQEwf as-up?
ngebpuy;cifumm:hGUzumJ;'I,'f'el!?maebalkX!jEJI
Aday;t, yant!
Plmah.s'A.!?.N,!:ff:fiigbe?'me
ILledpushy M!rlhy G
Rlan.I:P't a exgreq'ddY? kne!:'-- phonbyodean!ivUongzaxengy'Thma?;s!P, dleaf-JQlfoltso D$ubs?
```

Question 04:

Describe the dataset you used for training. Include screenshots (Figure 2) of your learning curves, the result.json file in your output folder, and some of your favorite samples?

Dataset Used: Pride and Prejudice – A classic novel by Jane Austen
Dataset Size: 395KB
Encoding: utf-8

Default Settings:

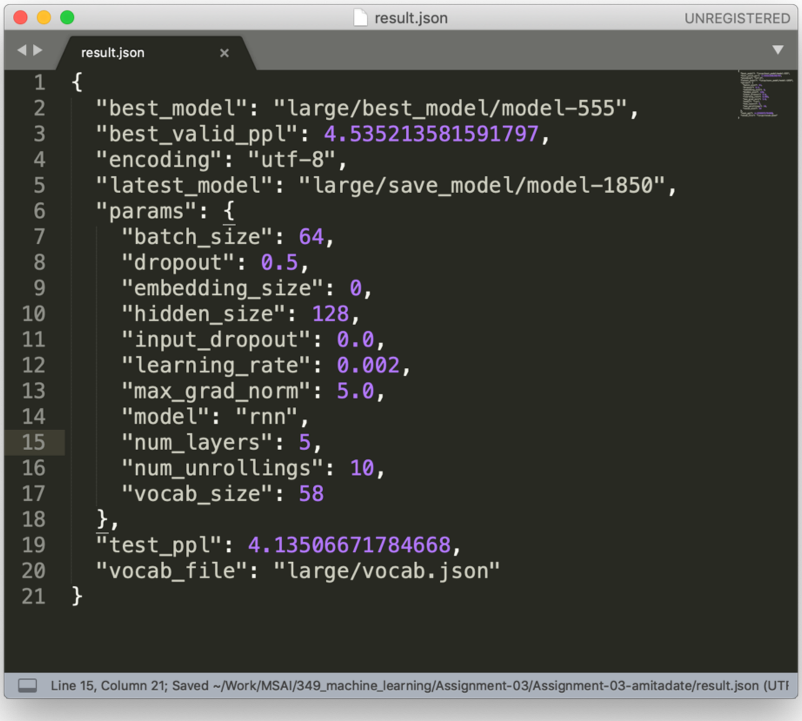
Best_valid_ppl : 3.873207685
Test_ppl: 3.58127001

After Optimization:

Best_valid_ppl : 4.5352135
Test_ppl: 4.1350667

Kindly Check the Attached Result.Json File

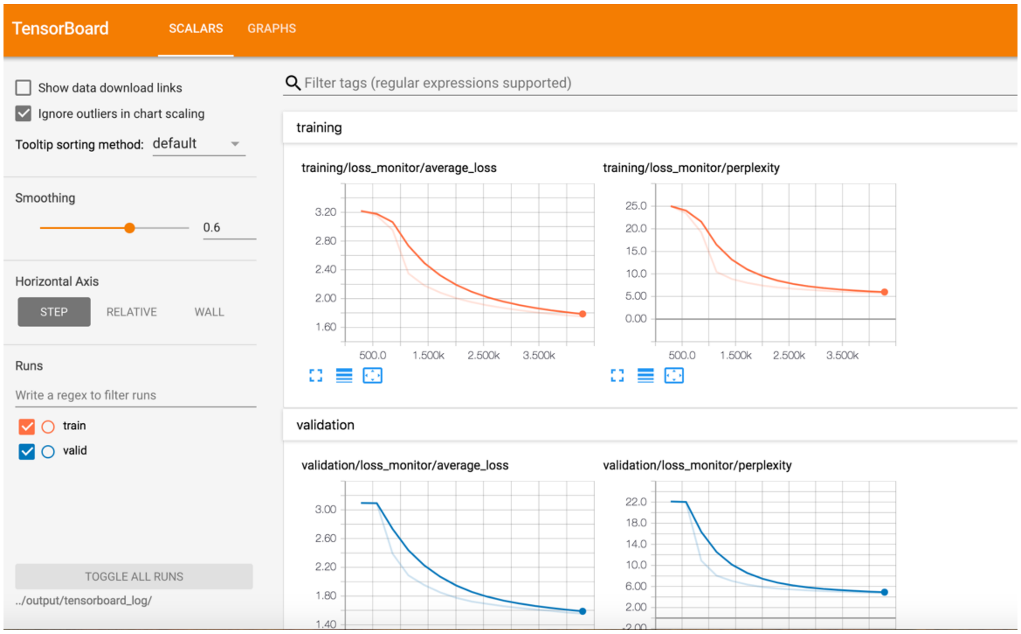
Result_Json:



```
{
  "best_model": "large/best_model/model-555",
  "best_valid_ppl": 4.535213581591797,
  "encoding": "utf-8",
  "latest_model": "large/save_model/model-1850",
  "params": {
    "batch_size": 64,
    "dropout": 0.5,
    "embedding_size": 0,
    "hidden_size": 128,
    "input_dropout": 0.0,
    "learning_rate": 0.002,
    "max_grad_norm": 5.0,
    "model": "rnn",
    "num_layers": 5,
    "num_unrollings": 10,
    "vocab_size": 58
  },
  "test_ppl": 4.13506671784668,
  "vocab_file": "large/vocab.json"
}
```

Line 15, Column 21; Saved ~/Work/MSAI/349_machine_learning/Assignment-03/Assignment-03-amitadate/result.json (UTI

Optimized Curve:



□ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing: 0.6

Horizontal Axis: [✓ STEP] RELATIVE WALL

Runs

Write a regex to filter runs

☑ ○ train

☑ ○ valid

../output/tensorboard_log/