High Performance Machine Learning Lab 3

Name: Vishwajeet Kulkarni
NYU ID: vk2630

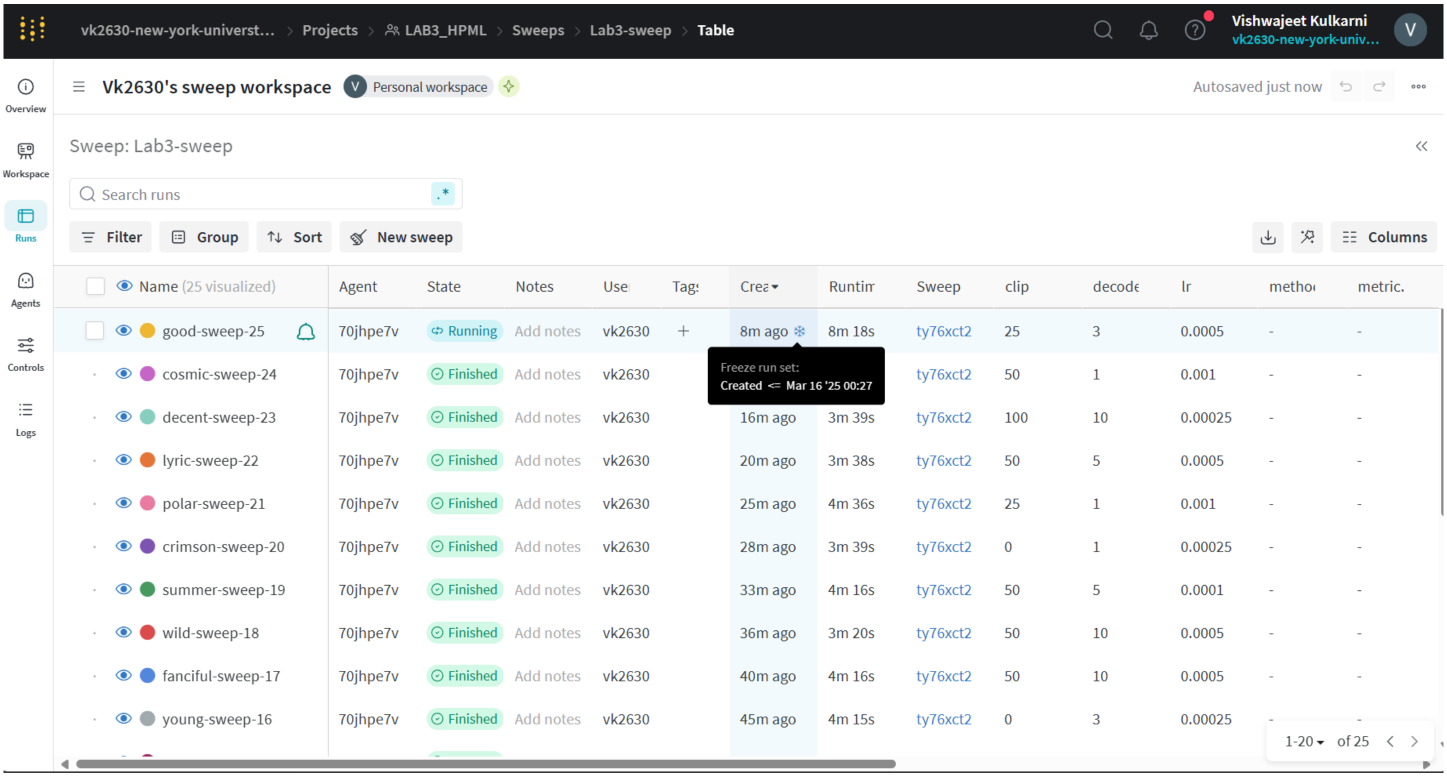
Prob 1: Chatbot Seq-2-Seq Model

1. Wandb Project:

https://wandb.ai/vk2630-new-york-universtity/LAB3_HPML?nw=nwuservk2630

2. Hyperparameter Sweeps:

Strategy: Random Search
Total Runs: 25

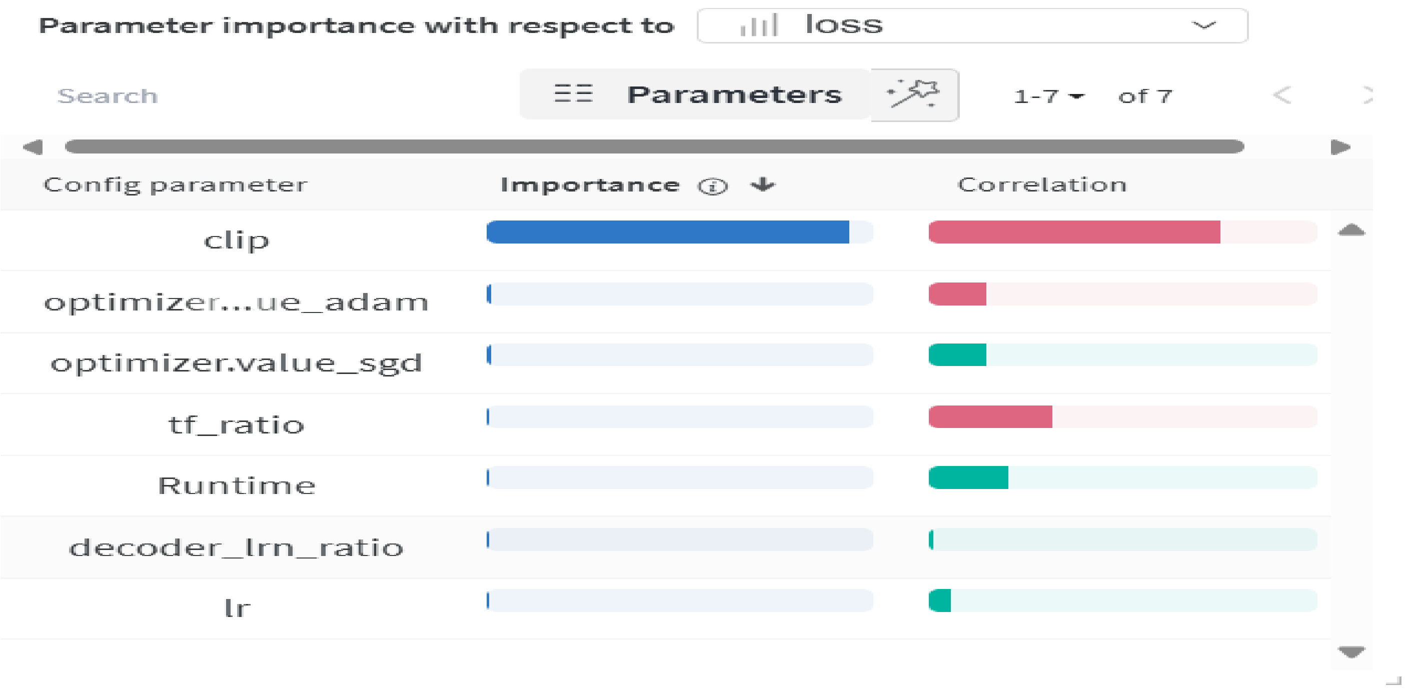


Name (25 visualized)	Agent	State	Notes	Use	Tag	Crea ▾	Runtir	Sweep	clip	decode	lr	metho	metric.
good-sweep-25	70jhpe7v	Running	Add notes	vk2630	+	8m ago	8m 18s	ty76xct2	25	3	0.0005	-	-
cosmic-sweep-24	70jhpe7v	Finished	Add notes	vk2630				ty76xct2	50	1	0.001	-	-
decent-sweep-23	70jhpe7v	Finished	Add notes	vk2630		16m ago	3m 39s	ty76xct2	100	10	0.00025	-	-
lyric-sweep-22	70jhpe7v	Finished	Add notes	vk2630		20m ago	3m 38s	ty76xct2	50	5	0.0005	-	-
polar-sweep-21	70jhpe7v	Finished	Add notes	vk2630		25m ago	4m 36s	ty76xct2	25	1	0.001	-	-
crimson-sweep-20	70jhpe7v	Finished	Add notes	vk2630		28m ago	3m 39s	ty76xct2	0	1	0.00025	-	-
summer-sweep-19	70jhpe7v	Finished	Add notes	vk2630		33m ago	4m 16s	ty76xct2	50	5	0.0001	-	-
wild-sweep-18	70jhpe7v	Finished	Add notes	vk2630		36m ago	3m 20s	ty76xct2	50	10	0.0005	-	-
fanciful-sweep-17	70jhpe7v	Finished	Add notes	vk2630		40m ago	4m 16s	ty76xct2	50	10	0.0005	-	-
young-sweep-16	70jhpe7v	Finished	Add notes	vk2630		45m ago	4m 15s	ty76xct2	0	3	0.00025	-	-

I observed that run with id vk2630 has least loss of . Hyperparameters value of above model:

1. Learning Rate: 0.00025
2. Clip: 100.0
3. Teacher Forcing Ratio: 1
4. Decoder Learning Ratio: 10
5. Optimizer: adam

3. Feature Importance:



Parameter importance with respect to loss

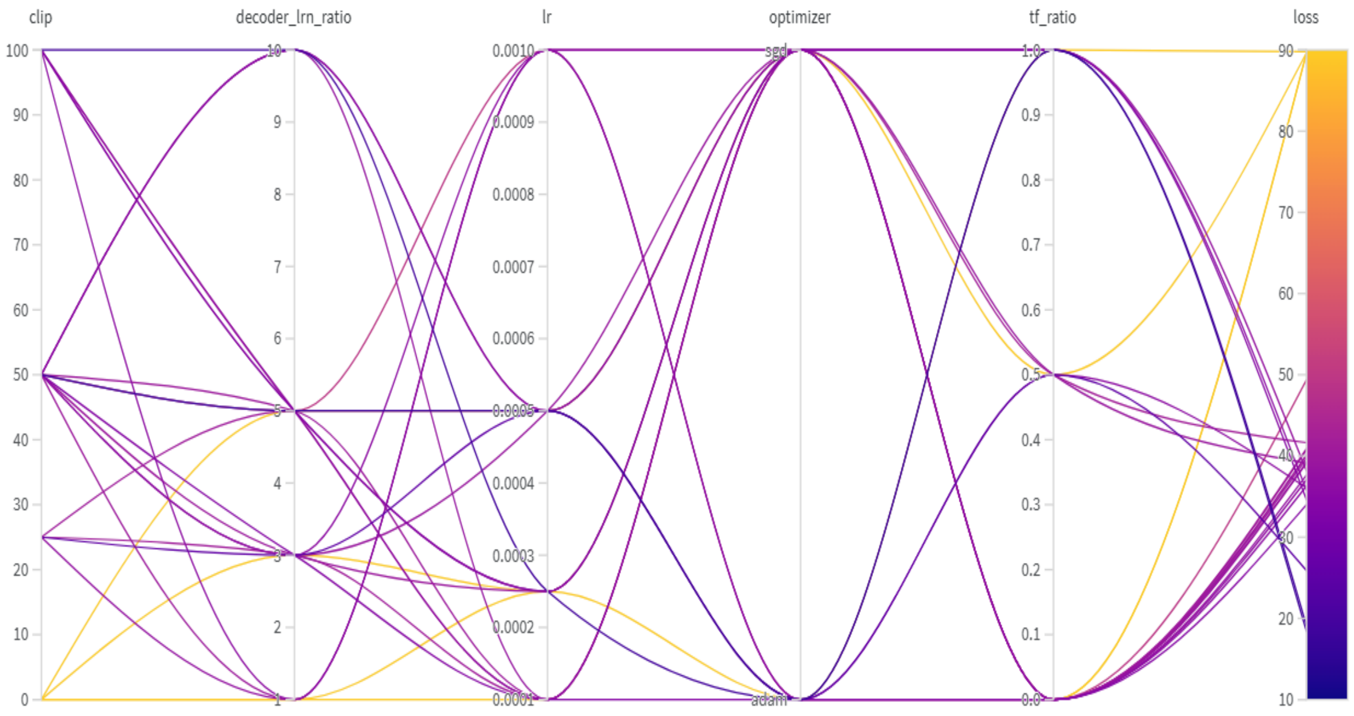
Search

Parameters

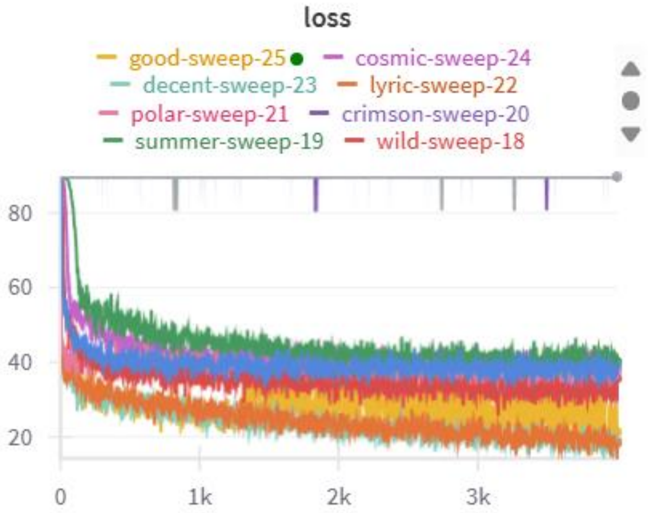
1-7 of 7

Config parameter	Importance ⓘ ↓	Correlation
clip		
optimizer...ue_adam		
optimizer.value_sgd		
tf_ratio		
Runtime		
decoder_lrn_ratio		
lr		

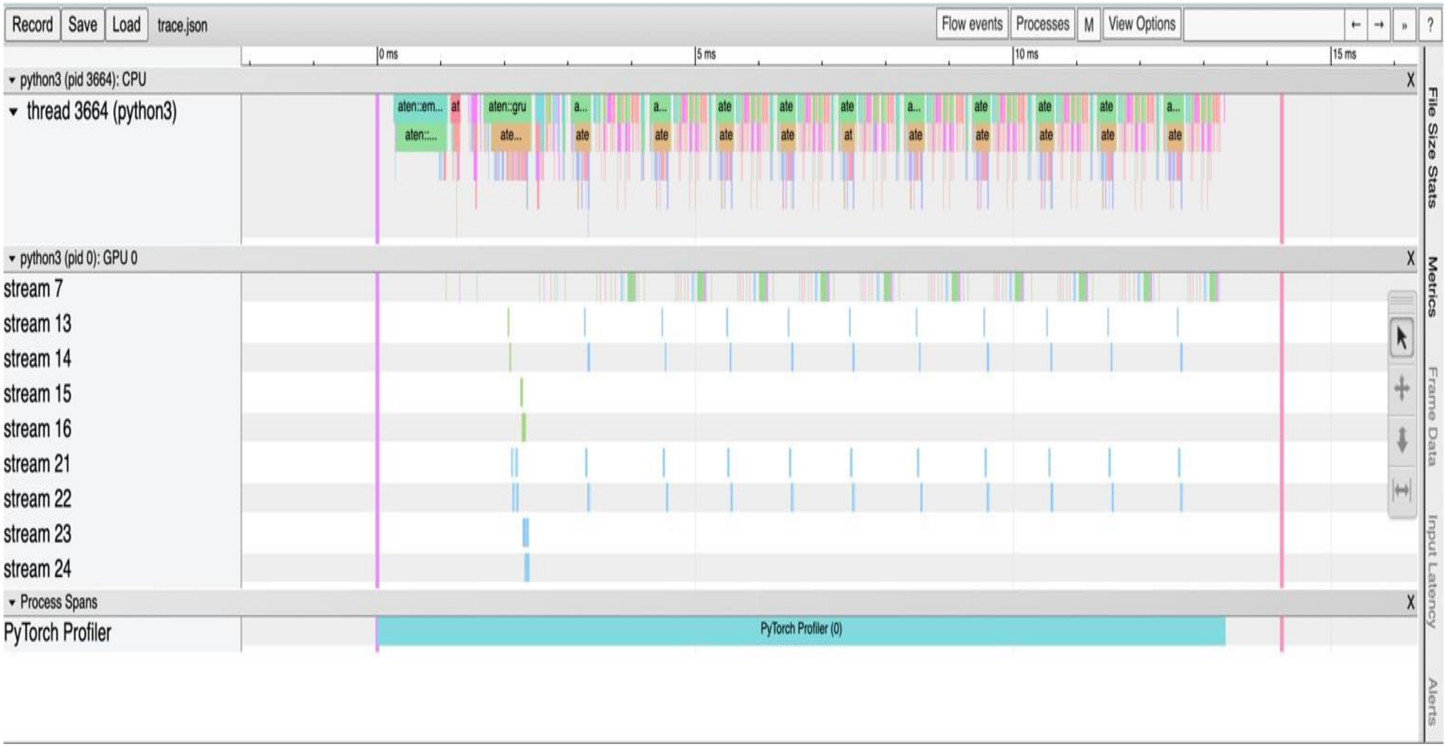
From above panel, we can observe that value of clip has very high effect on the loss i.e if value of clip is high, loss will be less (Value of correlation is negative). Moreover, we can see that both “ADAM” and “SGD” has high correlation, but ADAM has positive correlation and hence ADAM will be better choice.



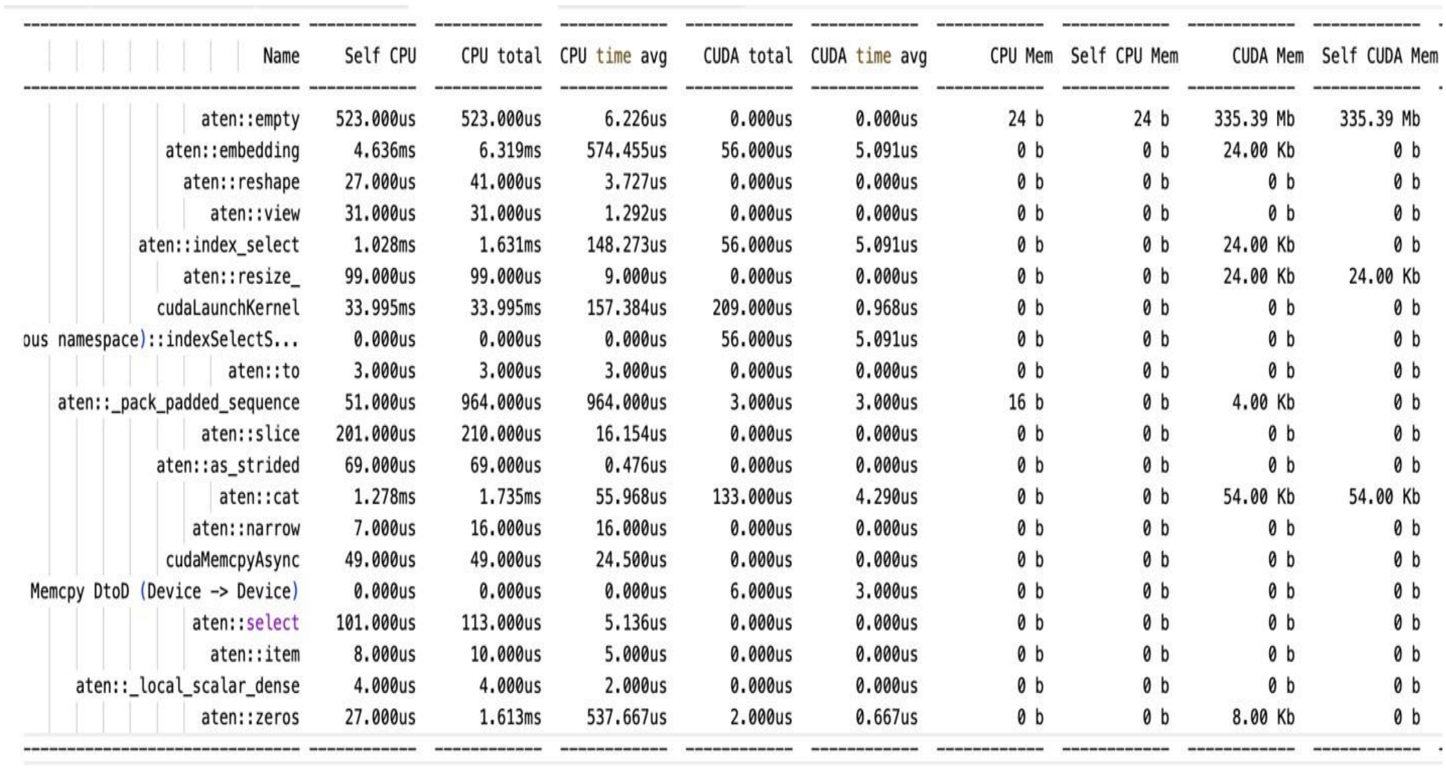
4. Variation of loss with iterations



5. PyTorch Profiler Tracing



6. Measurement of time and memory Consumption of model’s operators



```
------------------------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------
| | | | | | | |           Name     Self CPU    CPU total CPU time avg   CUDA total CUDA time avg    CPU Mem Self CPU Mem     CUDA Mem Self CUDA Mem
------------------------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------
| | | | | |        aten::empty    523.000us    523.000us      6.226us      0.000us      0.000us         24 b         24 b    335.39 Mb    335.39 Mb
               aten::embedding      4.636ms      6.319ms    574.455us     56.000us      5.091us          0 b          0 b     24.00 Kb          0 b
| | | | |        aten::reshape     27.000us     41.000us      3.727us      0.000us      0.000us          0 b          0 b          0 b          0 b
| | | | | |          aten::view     31.000us     31.000us      1.292us      0.000us      0.000us          0 b          0 b          0 b          0 b
            aten::index_select      1.028ms      1.631ms    148.273us     56.000us      5.091us          0 b          0 b     24.00 Kb          0 b
| | | | |        aten::resize_     99.000us     99.000us      9.000us      0.000us      0.000us          0 b          0 b     24.00 Kb     24.00 Kb
| | | |       cudaLaunchKernel     33.995ms     33.995ms    157.384us    209.000us      0.968us          0 b          0 b          0 b          0 b
ous namespace)::indexSelectS...      0.000us      0.000us      0.000us     56.000us      5.091us          0 b          0 b          0 b          0 b
| | | | | | |           aten::to      3.000us      3.000us      3.000us      0.000us      0.000us          0 b          0 b          0 b          0 b
    aten::_pack_padded_sequence     51.000us    964.000us    964.000us      3.000us      3.000us         16 b          0 b      4.00 Kb          0 b
| | | | | |         aten::slice    201.000us    210.000us     16.154us      0.000us      0.000us          0 b          0 b          0 b          0 b
| | | |       aten::as_strided     69.000us     69.000us      0.476us      0.000us      0.000us          0 b          0 b          0 b          0 b
| | | | | | |          aten::cat      1.278ms      1.735ms     55.968us    133.000us      4.290us          0 b          0 b     54.00 Kb     54.00 Kb
| | | | |         aten::narrow      7.000us     16.000us     16.000us      0.000us      0.000us          0 b          0 b          0 b          0 b
| | | |        cudaMemcpyAsync     49.000us     49.000us     24.500us      0.000us      0.000us          0 b          0 b          0 b          0 b
 Memcpy DtoD (Device -> Device)      0.000us      0.000us      0.000us      6.000us      3.000us          0 b          0 b          0 b          0 b
| | | | |         aten::select    101.000us    113.000us      5.136us      0.000us      0.000us          0 b          0 b          0 b          0 b
| | | | | |          aten::item      8.000us     10.000us      5.000us      0.000us      0.000us          0 b          0 b          0 b          0 b
| |   aten::_local_scalar_dense      4.000us      4.000us      2.000us      0.000us      0.000us          0 b          0 b          0 b          0 b
| | | | | |         aten::zeros     27.000us      1.613ms    537.667us      2.000us      0.667us          0 b          0 b      8.00 Kb          0 b
------------------------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------ ------------
```

Prob 2: TorchScript Seq-2-Seq Model

Q1. Explain the differences between tracing and scripting and how they are used in TorchScript?

Solution:

In **tracing**, both the model and sample input data are required to record computations and generate a graph-based function. However, it only logs operations specific to the provided input, making it ineffective for handling data-dependent control flow. In **scripting**, the model alone is converted into TorchScript without requiring sample data. This process translates the model code into a subset of Python that retains all control flow structures. For models without control flow dependencies, torch.jit.trace() can be used directly without modifications, as it automatically converts the model into TorchScript. However, scripting may require adjustments to the model code to ensure compatibility with TorchScript syntax before applying torch.jit.script(). When using tracing, the model's device and dropout layers must be set to test mode beforehand, as the traced model does not manage these operations internally. In contrast, scripting allows these settings to be adjusted just before inference, similar to standard eager execution.

Q2. Explain the changes needed in the chatbot model to allow for scripting.

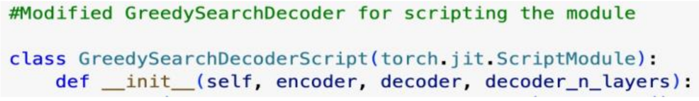
Solution:

In the given model, there are three sub-modules: Encoder, Decoder, and *GreedySearchDecoder*.

Changes required:

1. **Explicit Type Annotations for Forward Method Arguments:** By default, TorchScript assume all parameters of function as tensor. Hence, in case we need to pass any argument with different type like int in our case, we need to specify the type in python function.

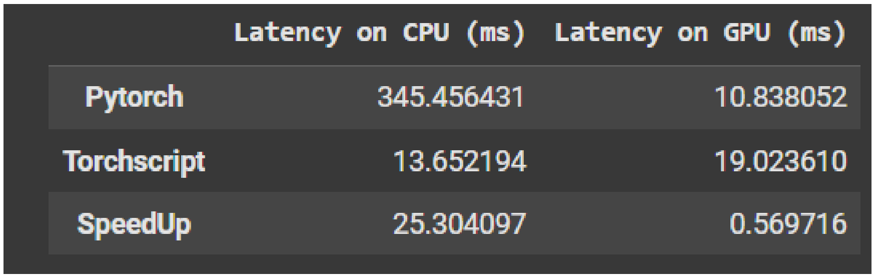
2. **Passing decoder_n_layers as a Constructor Argument:** Earlier, it was using fetching this value from decoder but since we are using traced version of decoder we will not be able to access that value anymore and hence we need to pass this to its constructor for it to use.



```
#Modified GreedySearchDecoder for scripting the module

class GreedySearchDecoderScript(torch.jit.ScriptModule):
    def __init__(self, encoder, decoder, decoder_n_layers):
```

Q3: Comparing Latency:



	Latency on CPU (ms)	Latency on GPU (ms)
Pytorch	345.456431	10.838052
Torchscript	13.652194	19.023610
SpeedUp	25.304097	0.569716