High Performance Machine Learning
Lab 3

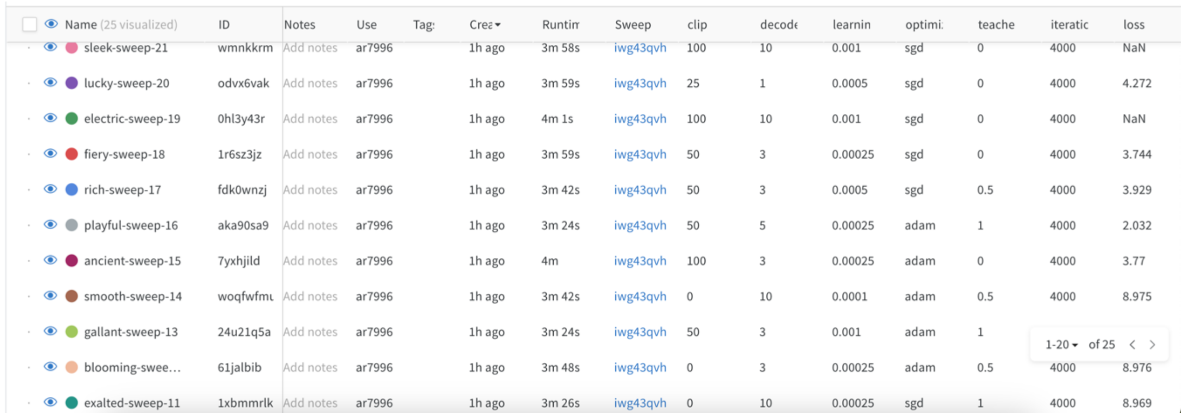
Ankit Rajvanshi
ar7996

Chatbot Seq-2-Seq Model

Wandb Project: https://wandb.ai/ar7996/hpml-lab-3?nw=nwuserar7996

Hyperparameter Sweeps:

Strategy: Random Search
Total Runs: 25

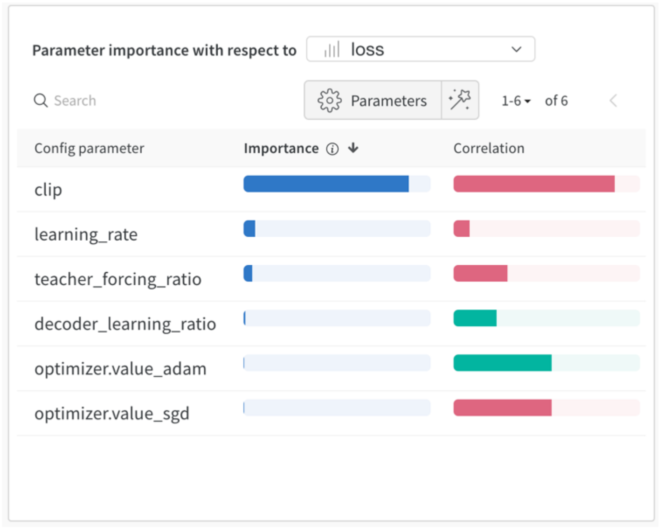


Name (25 visualized)	ID	Notes	Use	Tag:	Crea	Runtir	Sweep	clip	decode	learnin	optimi:	teache	iteratic	loss
sleek-sweep-21	wmnkkrm	Add notes	ar7996		1h ago	3m 58s	iwg43qvh	100	10	0.001	sgd	0	4000	NaN
lucky-sweep-20	odvx6vak	Add notes	ar7996		1h ago	3m 59s	iwg43qvh	25	1	0.0005	sgd	0	4000	4.272
electric-sweep-19	0hl3y43r	Add notes	ar7996		1h ago	4m 1s	iwg43qvh	100	10	0.001	sgd	0	4000	NaN
fiery-sweep-18	1r6sz3jz	Add notes	ar7996		1h ago	3m 59s	iwg43qvh	50	3	0.00025	sgd	0	4000	3.744
rich-sweep-17	fdk0wnzj	Add notes	ar7996		1h ago	3m 42s	iwg43qvh	50	3	0.0005	sgd	0.5	4000	3.929
playful-sweep-16	aka90sa9	Add notes	ar7996		1h ago	3m 24s	iwg43qvh	50	5	0.00025	adam	1	4000	2.032
ancient-sweep-15	7yxhjild	Add notes	ar7996		1h ago	4m	iwg43qvh	100	3	0.00025	adam	0	4000	3.77
smooth-sweep-14	woqfwfmu	Add notes	ar7996		1h ago	3m 42s	iwg43qvh	0	10	0.0001	adam	0.5	4000	8.975
gallant-sweep-13	24u21q5a	Add notes	ar7996		1h ago	3m 24s	iwg43qvh	50	3	0.001	adam	1		
blooming-swee…	61jalbib	Add notes	ar7996		1h ago	3m 48s	iwg43qvh	0	3	0.00025	adam	0.5	4000	8.976
exalted-sweep-11	1xbmmrlk	Add notes	ar7996		1h ago	3m 26s	iwg43qvh	0	10	0.00025	sgd	1	4000	8.969

From above results, we can observer that run with id ***aka90sa9*** has least loss of ***2.032***.
Hyperparameters value of above model:

1. Optimizer: ADAM
2. Learning Rate: 0.00025
3. Clip: 50
4. Teacher Forcing Ratio: 1
5. Decoder Learning Ratio: 5

Feature Importance:


Parameter importance with respect to loss

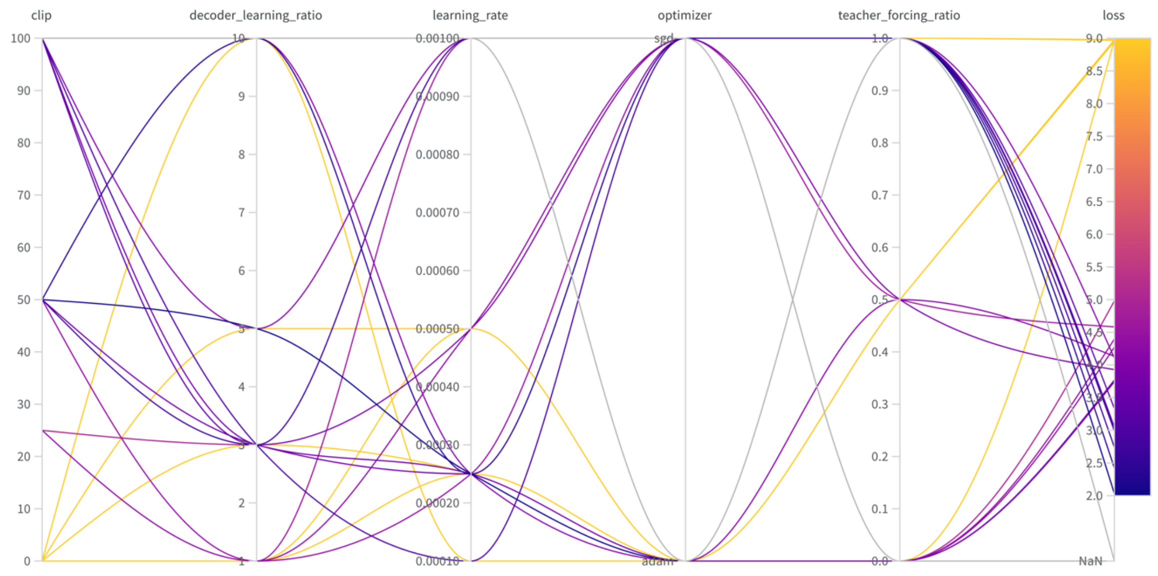
Search | Parameters | 1-6 of 6

Config parameter	Importance	Correlation
clip		
learning_rate		
teacher_forcing_ratio		
decoder_learning_ratio		
optimizer.value_adam		
optimizer.value_sgd		

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).

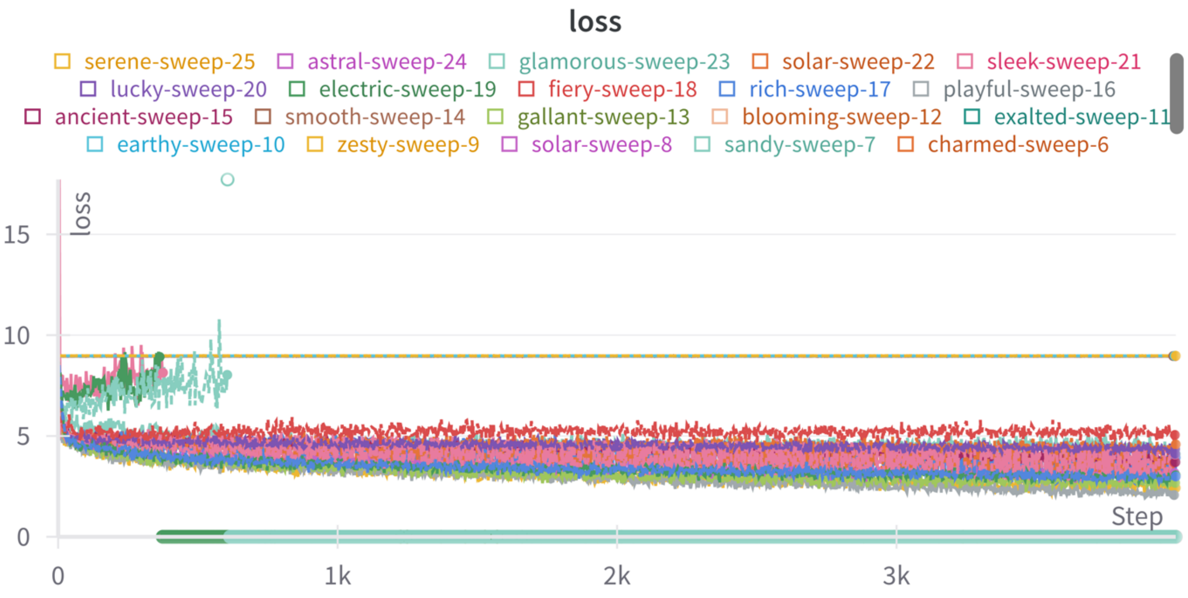
Moreover, we can see that both "ADAM" and "SGD" has high correlation, but ADAM has positive correlation and hence ADAM will be better choice.

The importance metric is very high for clip with respect to other hyperparameters which shows that if clip value is very less, no matter what the value of other parameters is, loss will be greater. It is evident by following diagram.



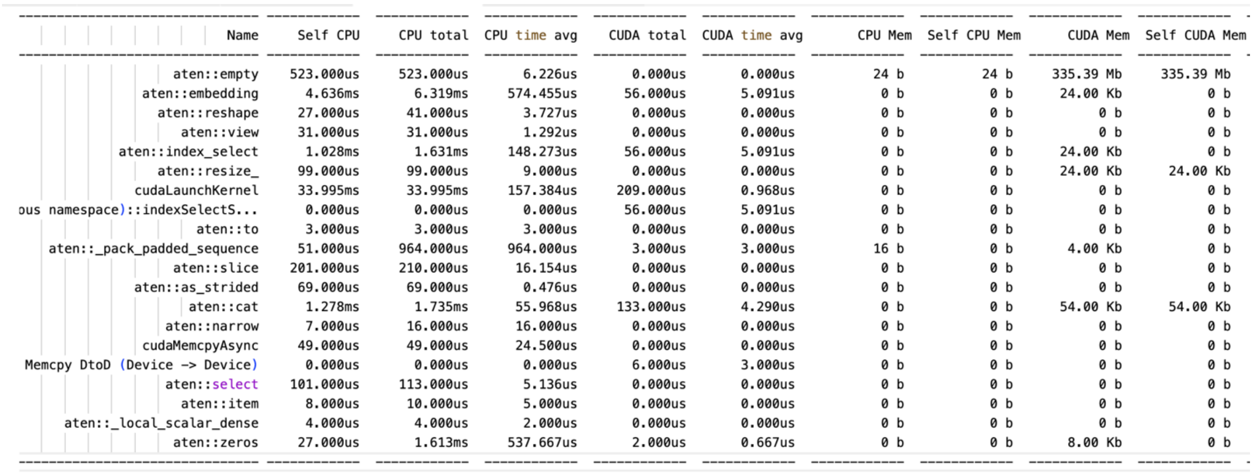
Variation of loss with iterations

For each run, there were 4000 iterations.



Measurement of time and memory Consumption of model's operators

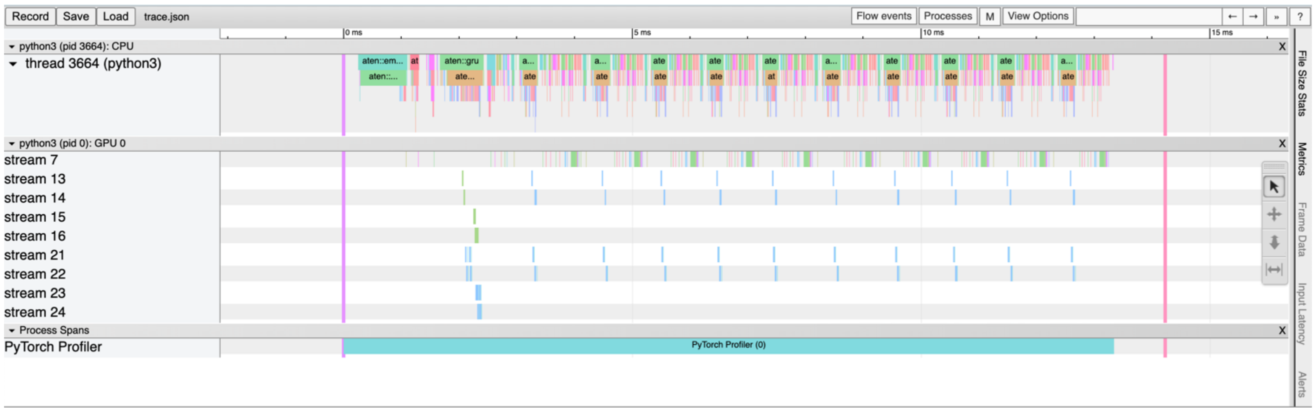
Full measurement is present in *profiler.txt* file.



```
-------------------------------  ------------  ------------  ------------  ------------  ------------  ------------  ------------  ------------  ------------  -
                           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
-------------------------------  ------------  ------------  ------------  ------------  ------------  ------------  ------------  ------------  ------------  -
```

PyTorch Profiler Tracing

Trace file has been saved as *trace.json*



TorchScript Seq-2-Seq Model

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

Answer:
Tracing: In case of tracing, function takes model and sample data as inputs and records the computations and builds graph-based function. Hence, only computations which were occurred for given data will be recorded. It is incapable of capturing data-driven control flow.

Scripting: In case of scripting, functions require only model without sample data. It converts model code into TorchScript (which is a subset of python language), including all control flows.

For a simple model without any control flow in its logic, we can easily use *torch.jit.trace()* without any modification to the model to trace the model and it will be converted to TorchScript. Whereas in case of *scripting*, we may need to rewrite the code to make sure it fits the TorchScript syntax. Once the code is rewritten adhere to TorchScript requirements, we can use *torch.jit.script()* to convert to TorchScript.

In case of *trace*, we need to set the device and dropout layers of model to test mode before tracing the model because traced version of model does not have functionality to perform these two operations. But in case of *scripting*, we can set the device and dropout layers to test mode just before the inferencing as we do in case of normal eager mode.

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

Answer:
In the given model, there are three sub modules: *Encoder*, *Decoder* and *GreedySearchDecoder*. Third sub module *GreedySearchDecoder* requires scripting as it has input based control flow in its logic.

Changes required in *GreedySearchDecoder*:

1. **Add additional arguments in its constructor for *decoder_n_layers*.** 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):
```

2. **Need to add more attributes**. Earlier, we were accessing various variable available in global scope of our python environment, but TorchScript version do not have access to those variables, and we need to save value of those variables from global scope to attributes of the model class. *(_SOS_token, _device, _decoder_n_layers)*

```
self._device = device
self._SOS_token = SOS_token
self._decoder_n_layers = decoder_n_layers
```

3. **Store above attributes as constants.** We need to add above attributes to special list called *__constants__* so that they can be used as literal values when constructing the graph in the forward method.

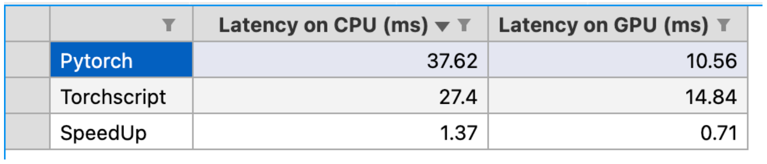
```
__constants__ = ['_device', '_SOS_token', '_decoder_n_layers']
```

4. **Enforce types of *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.

```
@torch.jit.script_method
def forward(self, input_seq : torch.Tensor, input_length : torch.Tensor, max_length : int):
```

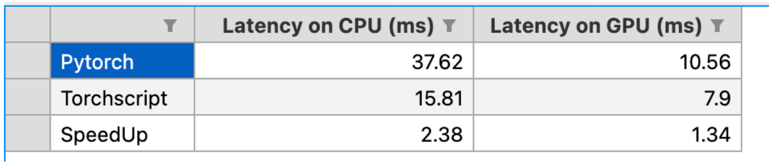
Q3: Compare Latency

Initial Run:



	Latency on CPU (ms)	Latency on GPU (ms)
Pytorch	37.62	10.56
Torchscript	27.4	14.84
SpeedUp	1.37	0.71

Second Run:



	Latency on CPU (ms)	Latency on GPU (ms)
Pytorch	37.62	10.56
Torchscript	15.81	7.9
SpeedUp	2.38	1.34