NSML: Meet a Machine Learning as a Platform

Hanjoo Kim

NSML Team

Contents

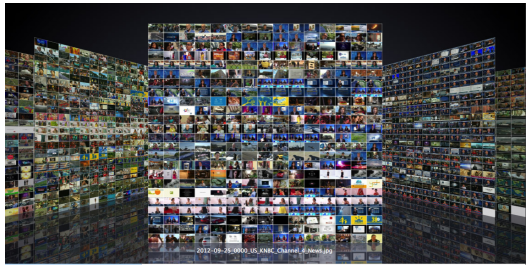
- Problem setup: Why NSML?
- NSML
- Key features
- Upcoming features
- Summary

You are angry even before doing experiments for real!

- You may build those things for your ML models…

Actually, you may spend most of your time to building environment ☹

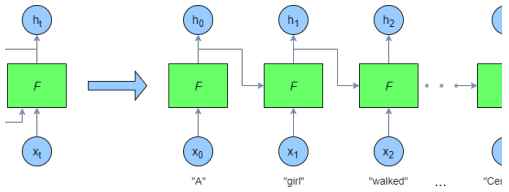
Resources & Infrastructure



Massive Dataset

Models & Training algorithm

Your real concern!

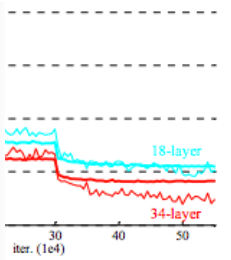


$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

$$\min_\theta J(\theta)$$

Why NSML?

- Der
  -
  -
  -
  -
  -



mm…,
t to see a
ing curve

A

t

Model C

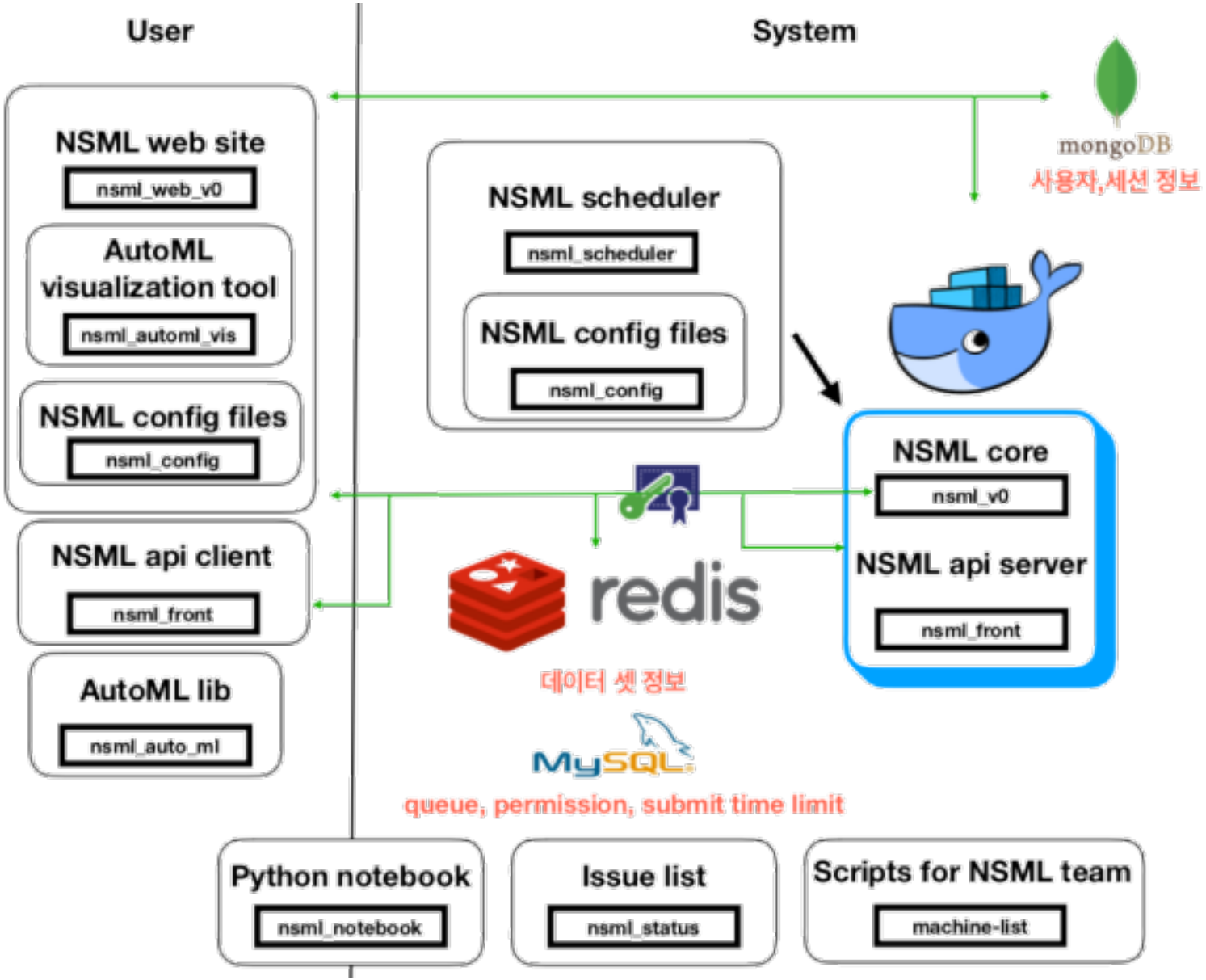
NSML: NAVER Smart Machine Learning platform

- We can provide …
  - Simple CLI environment and Web UI
  - NSML builds your experimental environment automatically.
  - Visualization tools
  - Managing multiple sessions simultaneously
  - Dataset management
  - Leaderboard for competition
- Forget about a server!

NSML: NAVER Smart Machine Learning platform

- NSML is not…
  - Deep learning library, itself
    - Use well-made Tensorflow or PyTorch
  - Automatic model designer
  - git-like repository
    - Use git ☺
  - Automatic parallelizer for your models
    - We can provide the infrastructure for parallelization, however you have to parallelize your code by yourself.
- NSML does not provided direct access point for a server that your session is running on.

NSML: NAVER Smart Machine Learning platform



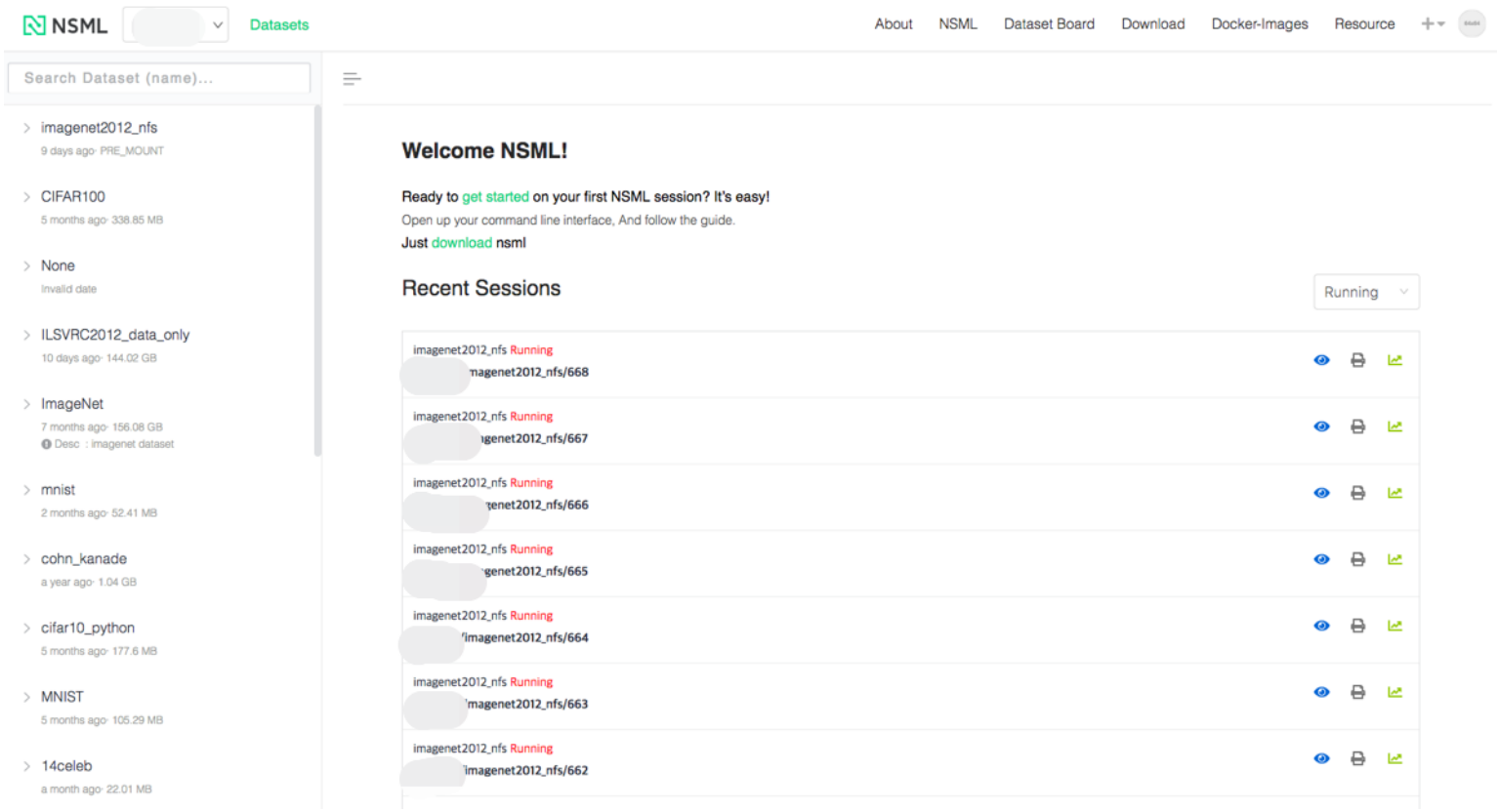
NSML Key features

- Basic topics
  - Datasets
  - Programming
  - Sessions
  - Visualization

- Advanced topics
  - Cooperating with teammates
  - Parallelization
  - Leaderboard
  - Hyperparameter tuning

Basic Topics – Datasets

- Public data: someone push, everyone can access
- Private data: someone push, permitted members only access



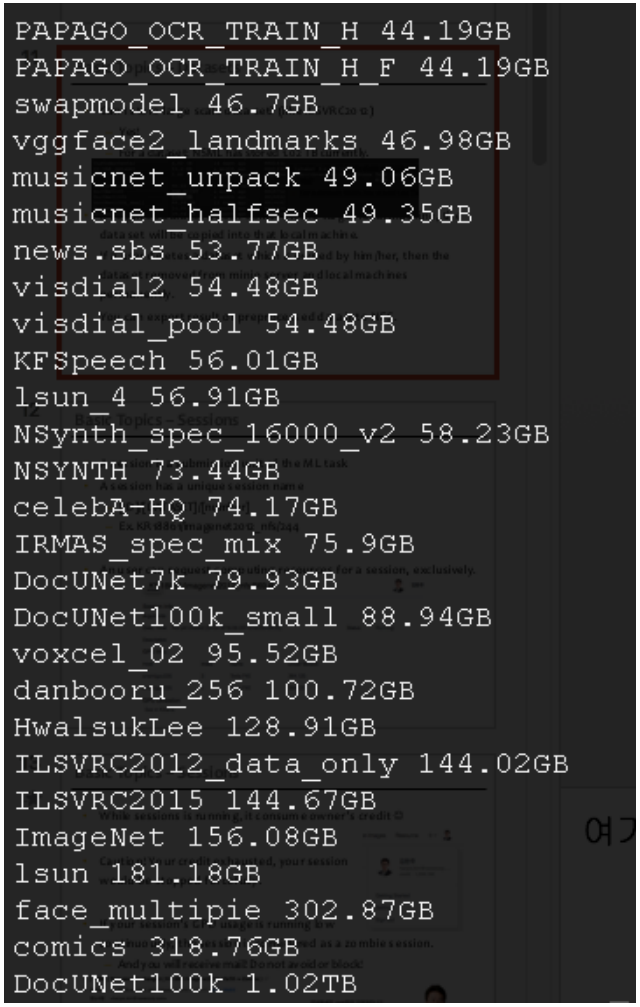
Basic Topics – Datasets

- Residence
  - dedicated public/private minio server
  - local NSML slaves
  - EXPERIMENTAL: shared file system (HDFS and NFS)

- If required dataset doesn't exist on the target local machine, the data set will be copied into that local machine.
- If a user deletes a dataset which is owned by him/her, then the dataset removed from minio server and local machines permanently.
- You can export the result or store preprocessed data into NFS.

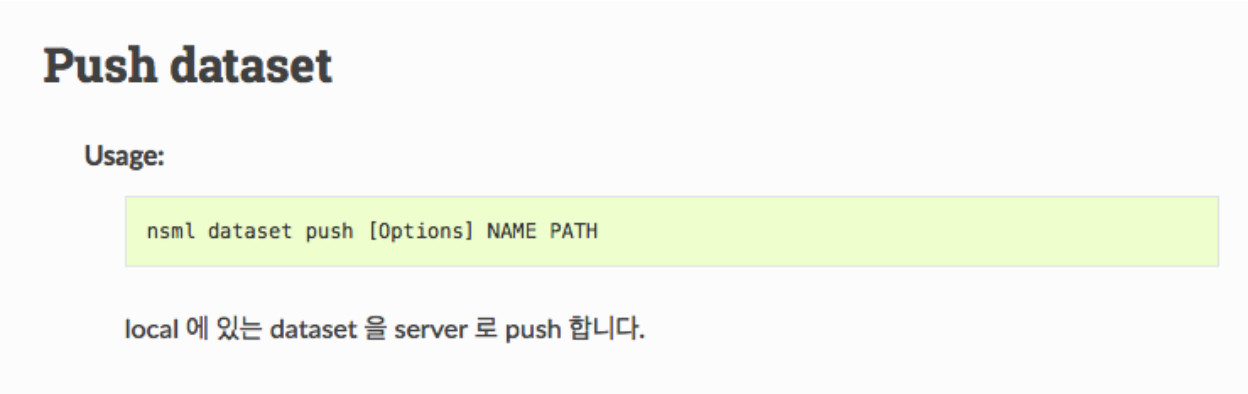
Basic Topics – Datasets

- Can I store large scale data set? (like ILSVRC2012)
  - Yes!
  - For a dataset: NSML has stored up to 1.02 TB.
- Data size concerning…
  - Need to preprocess?
  - I concern copy these data multiple times, does it spend much time??
- Don’t be afraid!
  - You can handle your dataset through NFS (available to read/write)



Basic Topics – Datasets

- How to push your dataset to NSML cluster?
  - Just use push command to upload the dataset in your computer, no matter how much the size of dataset is!



  - If you are in trouble due to too large size of data to hold on your local storage, please use NFS.
    - For using NFS shared file system, please feel free to contact us ☺

Basic Topics – Programming Models

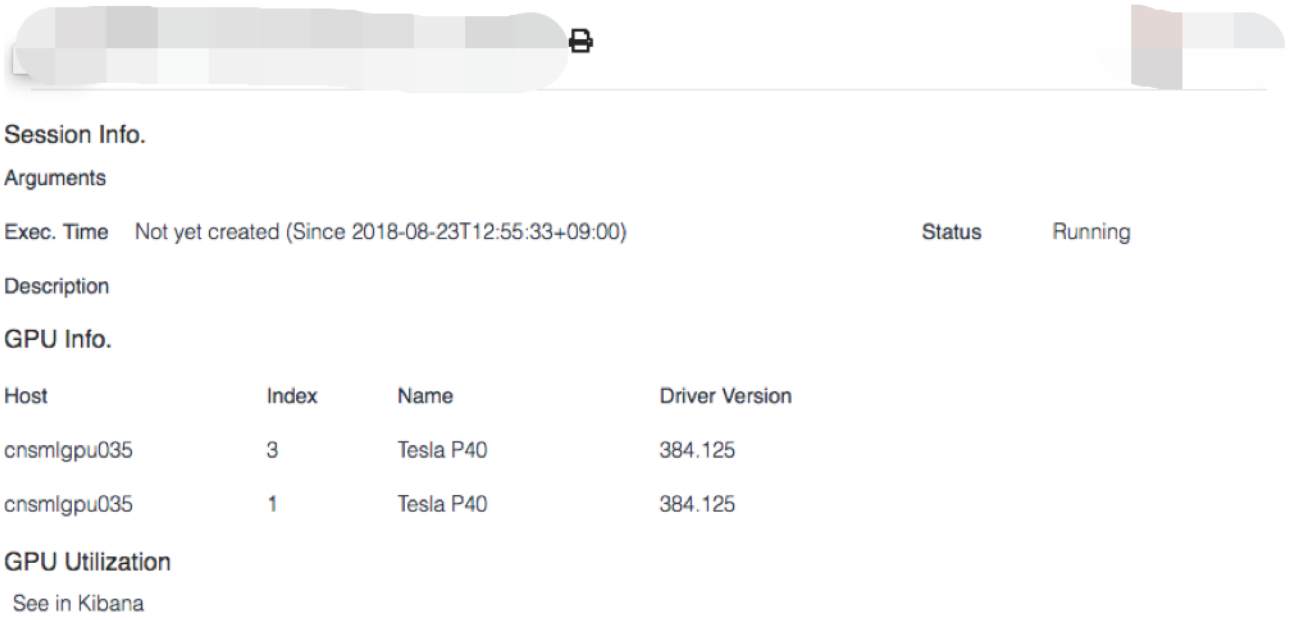
- Basically, python is supported as a programming language.
- Many existing users uses pytorch and tensorflow

- Write your own learning model
  - In a session, you can generate, train, save, and load the learning model.
  - The saved models can be accessed, submitted, and downloaded.
- If a specific version of library is needed, you can write your own requirements.txt.
- Please debugging your python code before reporting NSML bug ☺

Basic Topics – Sessions

- A session is a submission unit of the ML task
- A session has a unique session name
  - [ID]/[DATASET]/[number]
  - Ex. KR18861/imagenet2012_nfs/244

- An user can request computing resources for a session, exclusively.



Session Info.

Arguments

Exec. Time Not yet created (Since 2018-08-23T12:55:33+09:00) Status Running

Description

GPU Info.

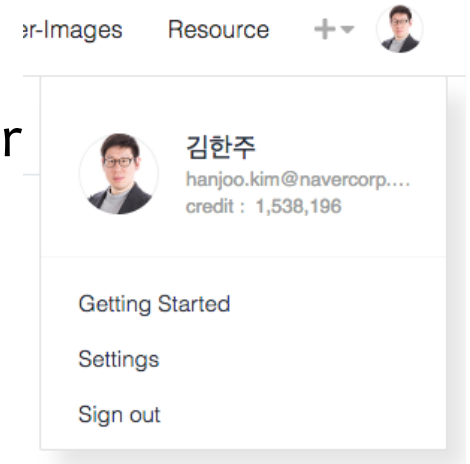
Host	Index	Name	Driver Version
cnsmlgpu035	3	Tesla P40	384.125
cnsmlgpu035	1	Tesla P40	384.125

GPU Utilization

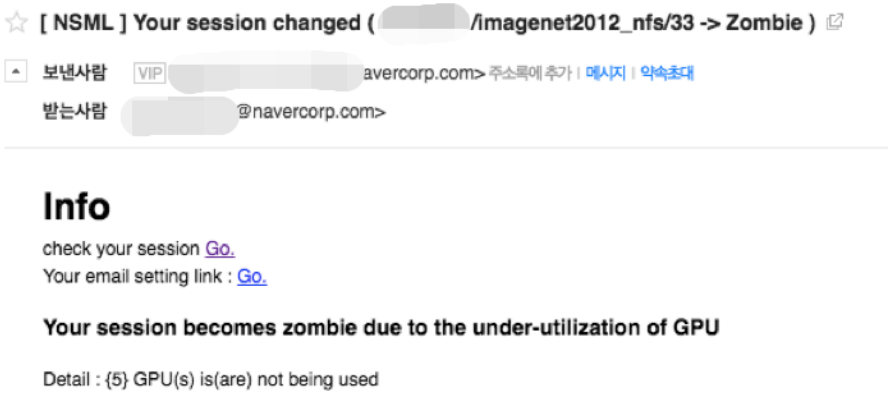
See in Kibana

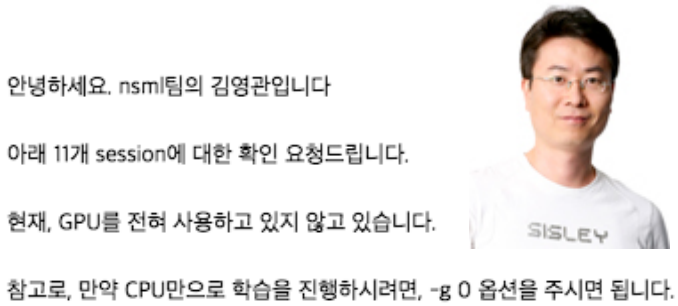
Basic Topics – Sessions

- While sessions is running, it consume owner's credit ☺



- Caution! If your credit has been exhausted, your session would be stopped forcefully.
- If your session's GPU usage is running low continuously, the session is considered as a zombie session.
  - And you will receive a mail! Do not avoid or block!



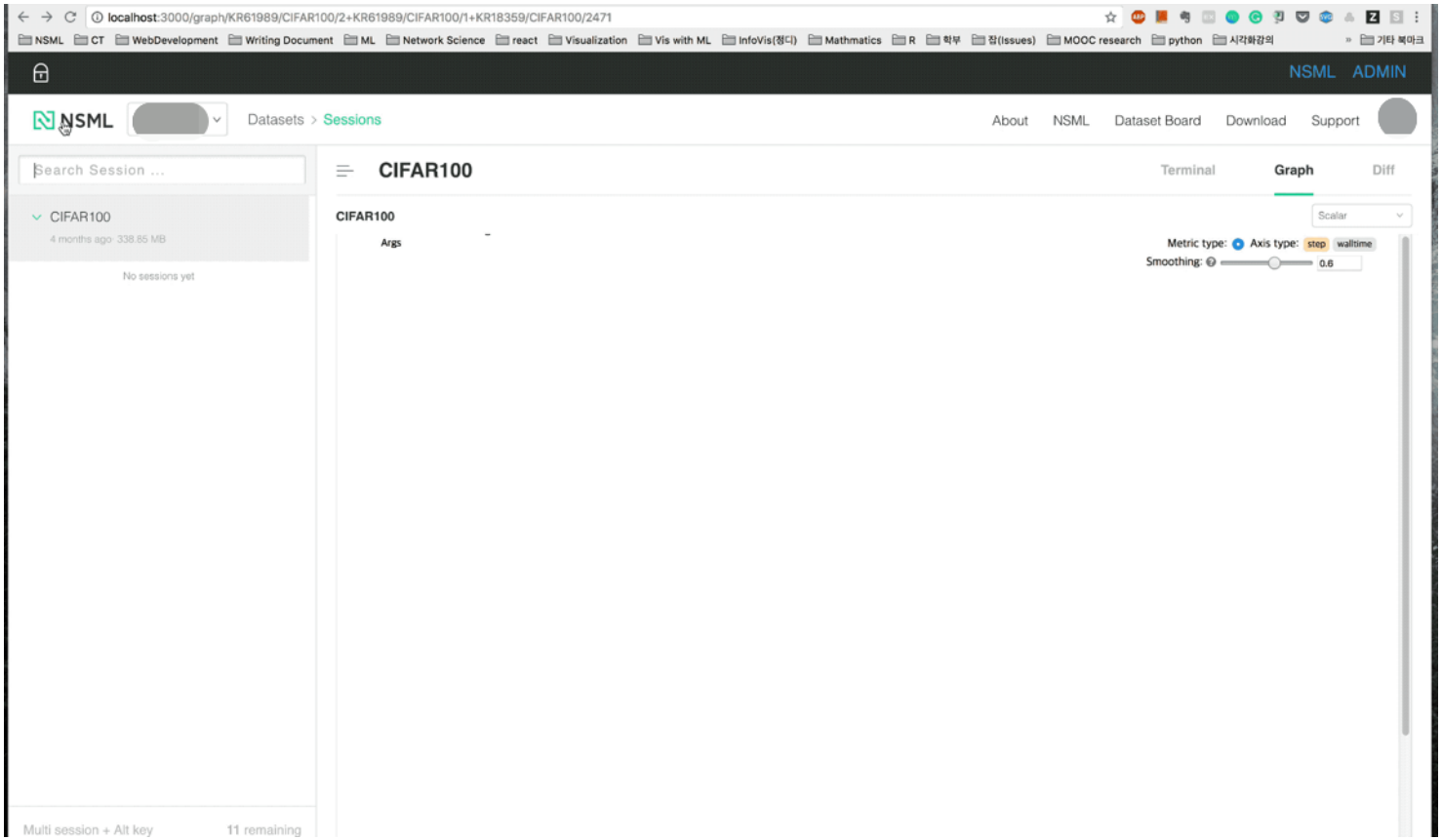


Basic Topics – Sessions

- Please refer to session monitor

Basic Topics – Visualization

- During a session running, you can report evaluation metrics to NSML event queue.
  - You don't have to open Excel or MATLAB for creating charts ☺



Advanced Topics – Cooperating with teammates

- Teamwork feature
  - Datasets: Some dataset should be shared with the teammate, but have to be hidden to the other users.
  - Models and Sessions: You want to reproduce or share the generated models and code that your teammate has tested on NSML.

- NSML provides a team feature for cooperating.
  - When you login to NSML, use `-t` option
  - Under team options is enabled, the sessions and models can be shared between teammates
  - Private dataset for a team is only displayed to the team.
  - For more details, please contact us ☺

Advanced Topics – Cooperating with teammates

- Actually, you might use the team feature
  - In Hackathons!

  - They can share their own models and codes for sessions.

- If you want to make your own team, or private datasets,
  - I repeat, please contact us!

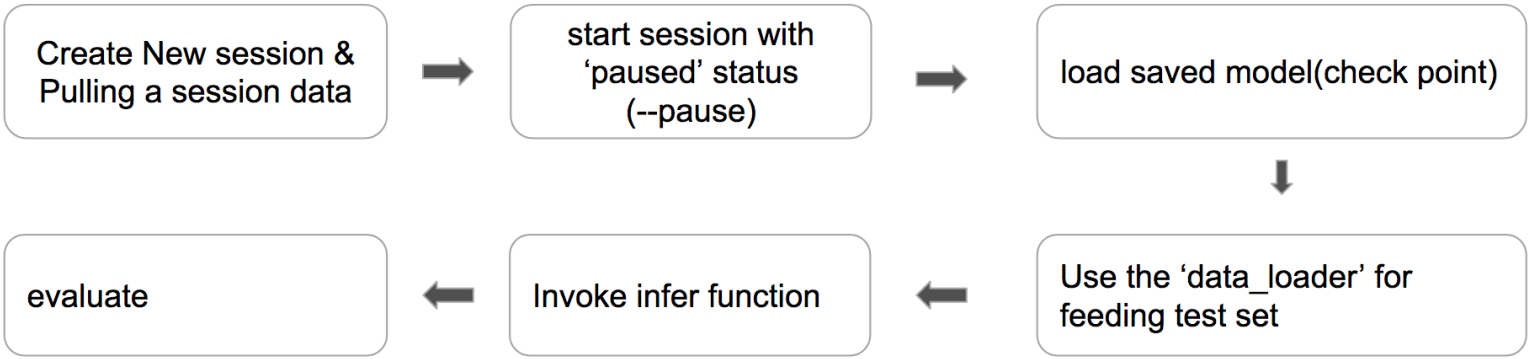
Advanced Topics – Parallelization

- NSML doesn't parallelize your code for multi GPUs or multi nodes.
  - NSML provides multiple resources only.

- You have to wrap your code with parallelization according to your ML libraries.
  - i.e) for pytorch:
    - `DataParallel` class for multi GPUs
    - `torch.distributed` package and `DistributedDataParallel` class for multi nodes (distributed learning)

Advanced Topics – Leaderboard

- Good for competition (did you see the hackathon?)
  - To build up leaderboard, your dataset and code should follow the specification of NSML. (refer to NSML manual)
  - The process of the leaderboard



Advanced Topics – Leaderboard

The End.

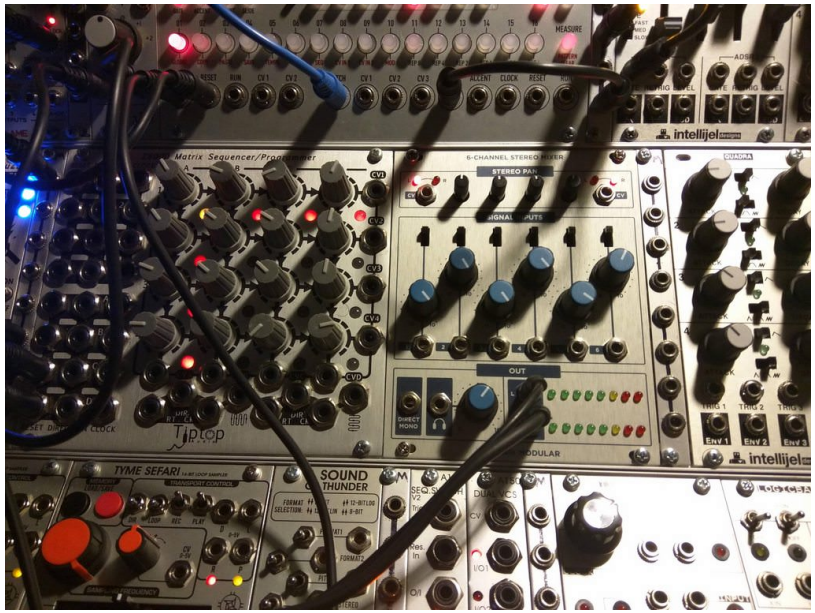
Custom · Descending

Public

순위	팀	모델	점수	제출시간	응원수
1	DoRoSY		0.52575	an hour ago	69
2	메인보드		0.45745	5 days ago	12
3	125		0.41825	2 days ago	4
4	ksj		0.41643	22 minutes ago	4
5	인간지능		0.41192	10 minutes ago	6
6	일이나잘하자		0.38326	6 hours ago	0
7	@bum_family		0.37879	15 days ago	2
8	골드핑거		0.37565	an hour ago	0
9	아빠둘		0.36193	7 hours ago	10
10	my mistake		0.35381	an hour ago	0

Advanced Topics – Hyperparameter tuning (Developing)

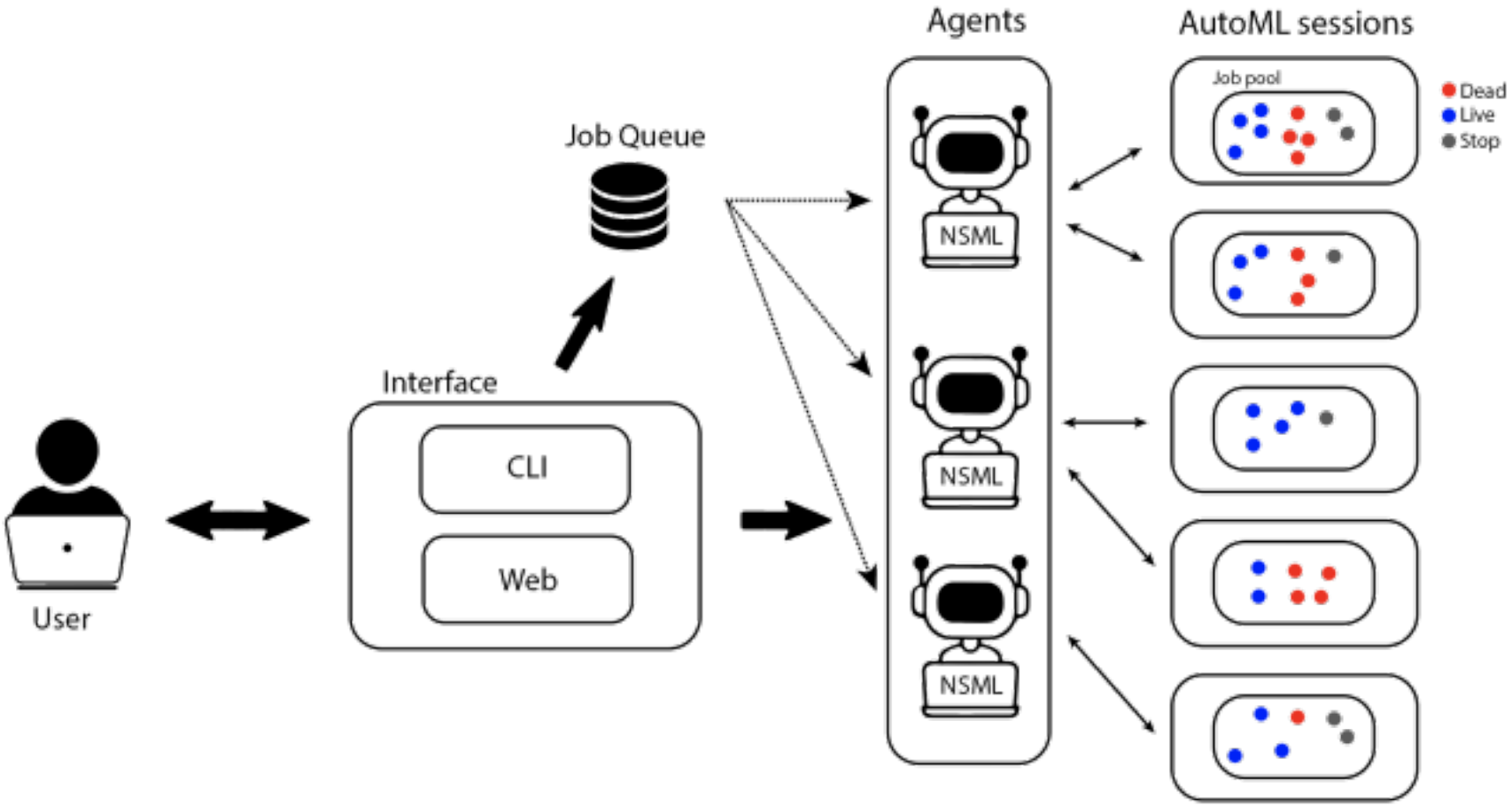
- In modern ML task, we have too many hyperparameters to …
- Hard to determine each parameter ☹
  - LR, weight decay, batch-size, …



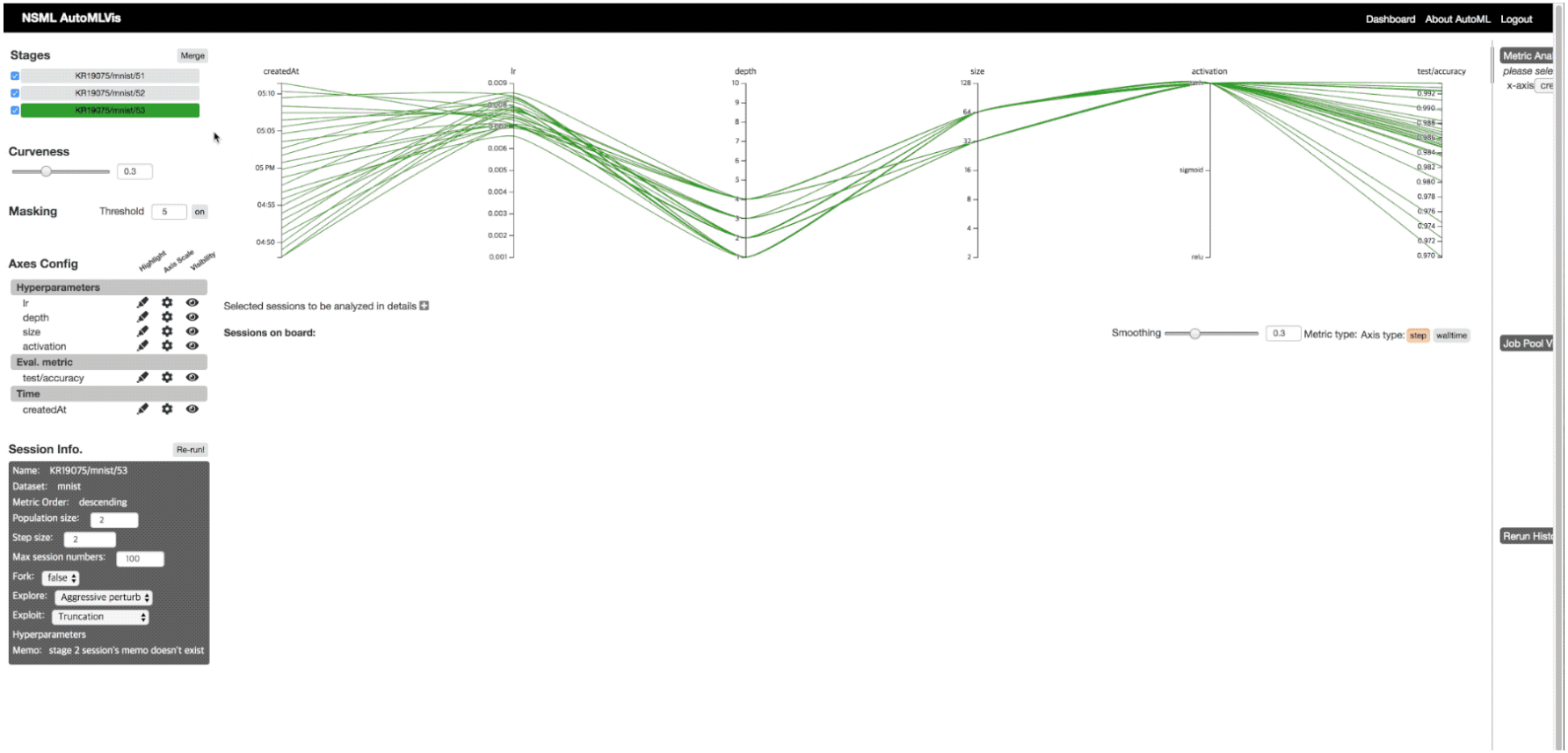
  - NSML's AutoML has been proposed to redeem you from tuning hell.

Advanced Topics – Hyperparameter tuning (Developing)

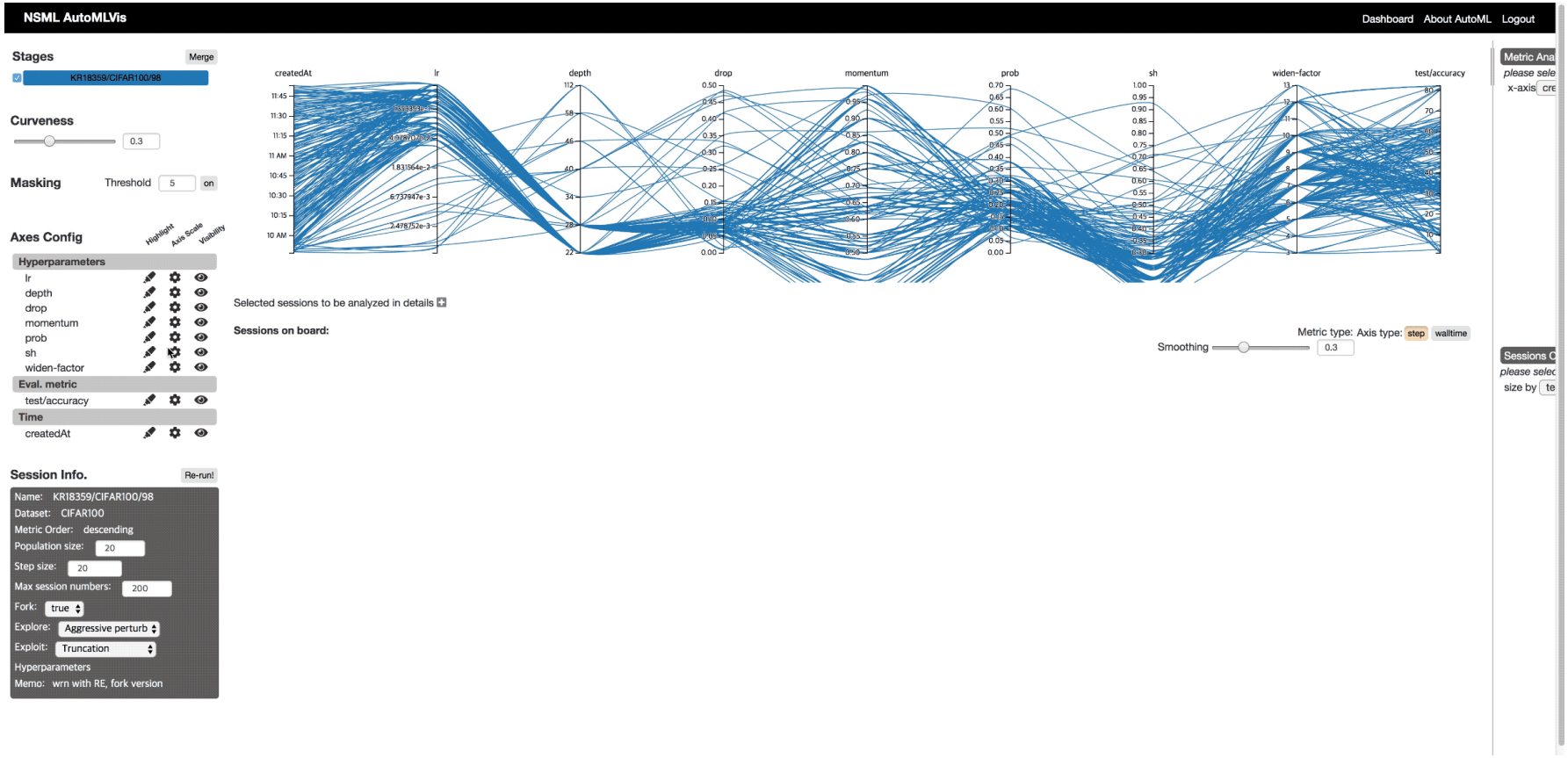
- NSML's AutoML architecture



Advanced Topics – Hyperparameter tuning (Developing)

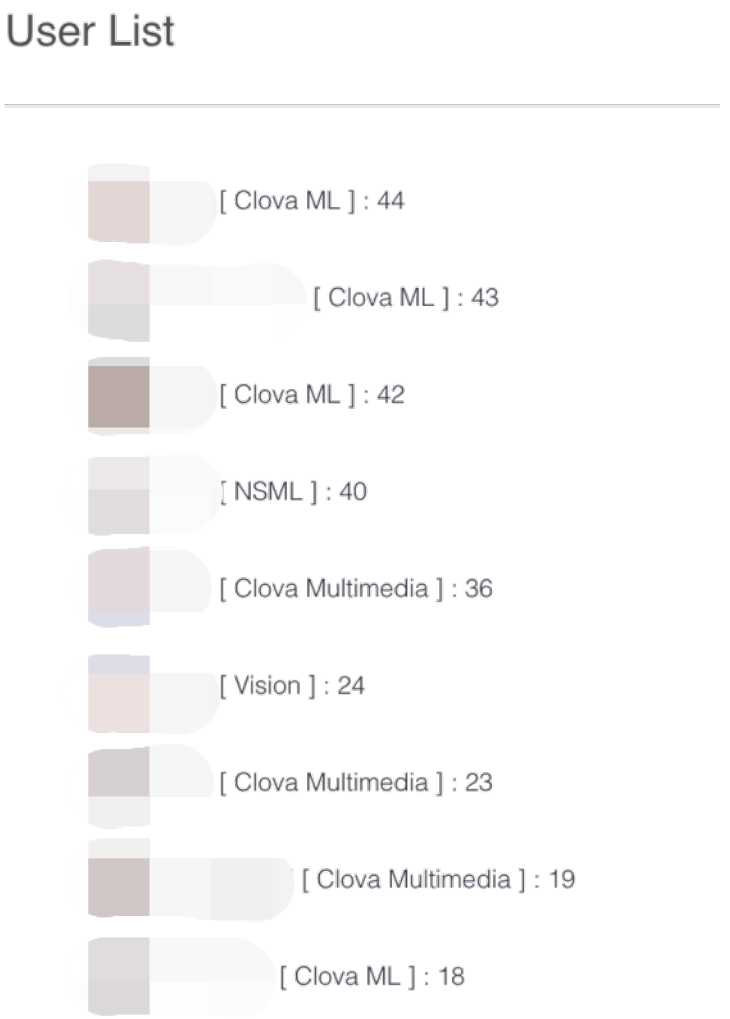


Advanced Topics – Hyperparameter tuning (Developing)



NSML – Use cases

- Many ML jobs are running on NSML.
- By the request from CLOVA Voice team, we synthesized 140K sentences for two different person with 70 x 2 GPUs. (for 4 days, used 947,076 credits)
- We also have held Hackathons three times.



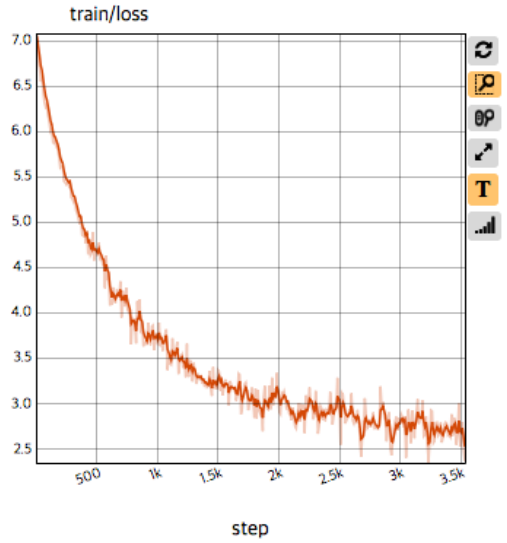
NSML – Use cases

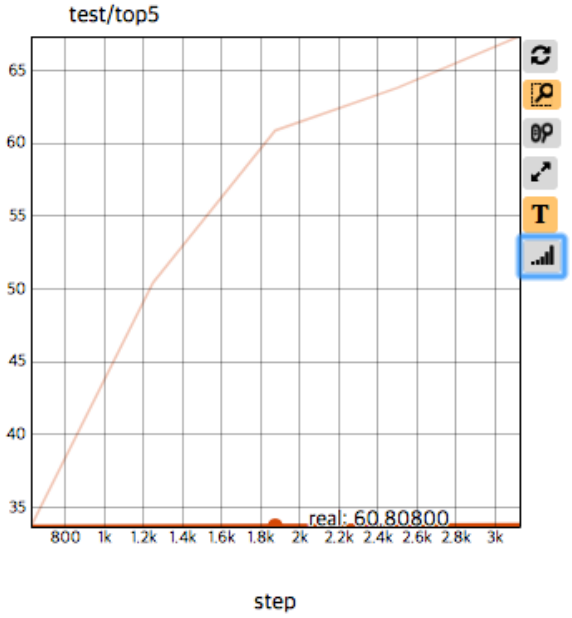
- ImageNet training…
  - ILSVRC 2012 dataset (over 144 GB)
  - ResNet implementation on PyTorch
  - Work well!

imagenet2012_nfs

KR18861/imagenet2012_nfs/957

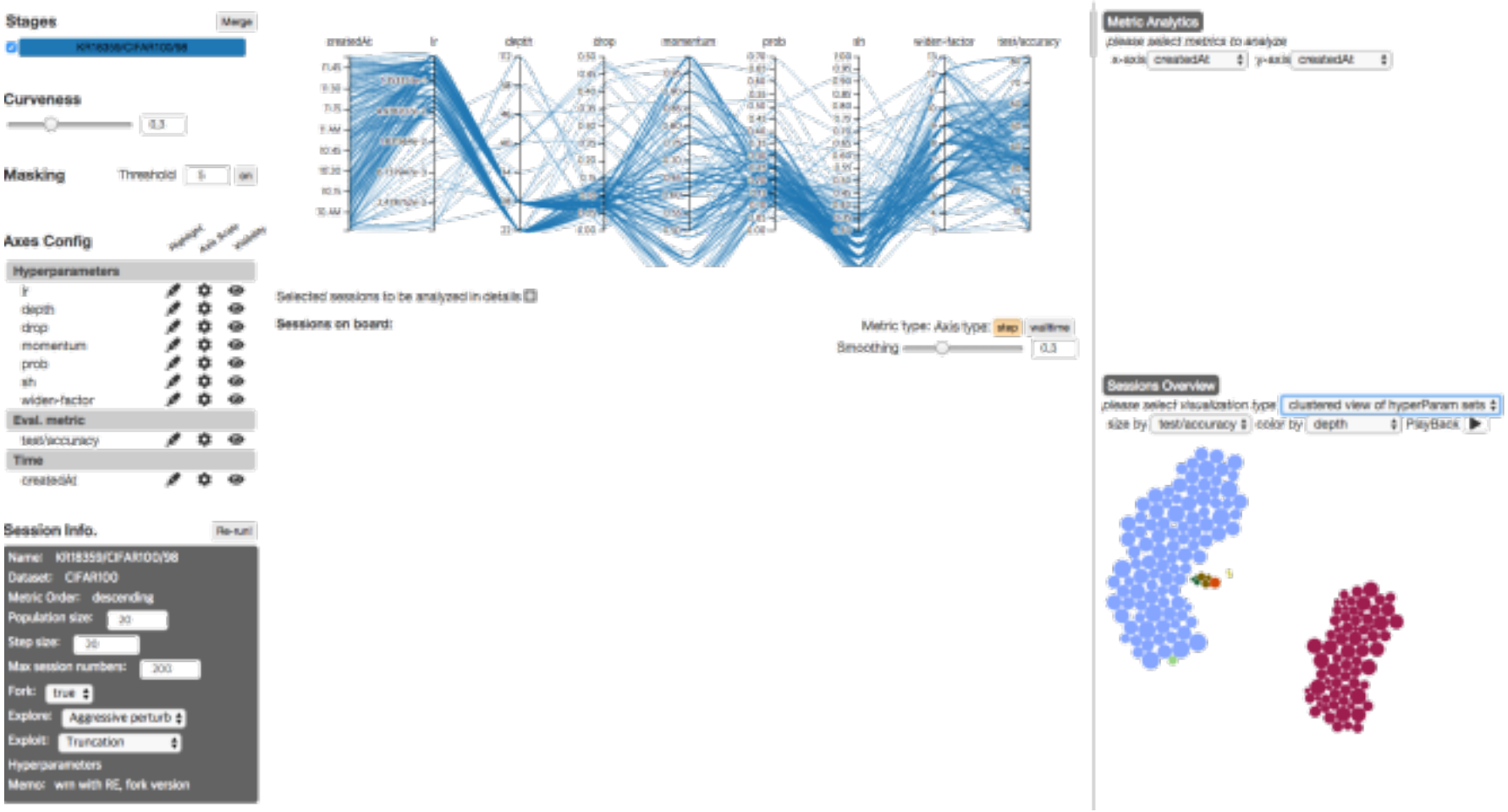
```
Epoch: [4][610/626]    Time 0.547 (2.690)      Data 0.000 (0.053)      Loss 2.7790 (2.7772)    Prec@1 43.359 (40.171)  Prec@5 64.844 (65.241)
Epoch: [4][620/626]    Time 8.962 (2.700)      Data 0.000 (0.052)      Loss 2.6075 (2.7770)    Prec@1 39.844 (40.158)  Prec@5 71.094 (65.260)
Test: [0/196]  Time 17.002 (17.002)    Loss 1.7871 (1.7871)    Prec@1 58.203 (58.203)  Prec@5 84.375 (84.375)
Test: [10/196] Time 0.178 (3.713)      Loss 2.8785 (2.2868)    Prec@1 39.062 (49.290)  Prec@5 62.891 (73.757)
Test: [20/196] Time 9.319 (3.456)      Loss 2.7195 (2.2771)    Prec@1 45.312 (49.219)  Prec@5 65.625 (73.977)
Test: [30/196] Time 0.771 (3.076)      Loss 1.9205 (2.2252)    Prec@1 50.781 (49.836)  Prec@5 81.250 (74.698)
Test: [40/196] Time 2.450 (2.973)      Loss 2.4679 (2.3199)    Prec@1 37.109 (46.456)  Prec@5 70.703 (73.190)
Test: [50/196] Time 0.171 (2.912)      Loss 1.9624 (2.2782)    Prec@1 47.266 (46.086)  Prec@5 79.297 (73.958)
Test: [60/196] Time 4.460 (2.896)      Loss 2.7698 (2.2444)    Prec@1 35.156 (46.356)  Prec@5 66.406 (74.686)
Test: [70/196] Time 0.126 (2.905)      Loss 2.6676 (2.2494)    Prec@1 41.406 (46.473)  Prec@5 67.969 (74.477)
Test: [80/196] Time 2.395 (2.875)      Loss 3.5014 (2.2664)    Prec@1 26.562 (46.267)  Prec@5 55.469 (74.166)
Test: [90/196] Time 0.159 (2.879)      Loss 4.4841 (2.3550)    Prec@1 17.969 (44.991)  Prec@5 41.797 (72.759)
Test: [100/196]        Time 1.312 (2.819)      Loss 3.6203 (2.4461)    Prec@1 29.297 (43.417)  Prec@5 50.781 (71.218)
Test: [110/196]        Time 0.162 (2.837)      Loss 2.7509 (2.4826)    Prec@1 43.750 (43.124)  Prec@5 64.844 (70.548)
Test: [120/196]        Time 4.536 (2.807)      Loss 3.3323 (2.5312)    Prec@1 30.859 (42.552)  Prec@5 57.812 (69.718)
Test: [130/196]        Time 0.931 (2.809)      Loss 2.4074 (2.5633)    Prec@1 48.828 (42.065)  Prec@5 69.141 (69.188)
Test: [140/196]        Time 2.310 (2.794)      Loss 2.9315 (2.5952)    Prec@1 35.938 (41.620)  Prec@5 62.891 (68.548)
Test: [150/196]        Time 4.832 (2.804)      Loss 3.2470 (2.6356)    Prec@1 35.547 (41.163)  Prec@5 61.328 (67.881)
Test: [160/196]        Time 0.523 (2.781)      Loss 2.6013 (2.6625)    Prec@1 44.141 (40.771)  Prec@5 67.578 (67.440)
Test: [170/196]        Time 6.809 (2.775)      Loss 2.0285 (2.6946)    Prec@1 50.781 (40.223)  Prec@5 78.516 (66.913)
Test: [180/196]        Time 0.653 (2.744)      Loss 1.8090 (2.7113)    Prec@1 51.562 (40.008)  Prec@5 80.078 (66.626)
Test: [190/196]        Time 7.550 (2.748)      Loss 2.1115 (2.6939)    Prec@1 37.891 (40.245)  Prec@5 82.812 (67.028)
 * Prec@1 40.586 Prec@5 67.268
Epoch: [5][0/626]      Time 25.403 (25.403)    Data 24.699 (24.699)    Loss 2.7878 (2.7878)    Prec@1 41.406 (41.406)  Prec@5 61.328 (61.328)
Epoch: [5][10/626]     Time 0.551 (4.638)      Data 0.000 (3.094)      Loss 2.8931 (2.7143)    Prec@1 41.406 (42.933)  Prec@5 66.016 (66.016)
Epoch: [5][20/626]     Time 9.379 (3.937)      Data 0.000 (1.675)      Loss 3.0382 (2.7291)    Prec@1 37.500 (42.374)  Prec@5 62.891 (65.885)
Epoch: [5][30/626]     Time 0.626 (3.403)      Data 0.000 (1.135)      Loss 2.7658 (2.7264)    Prec@1 42.578 (41.784)  Prec@5 64.844 (65.864)
Epoch: [5][40/626]     Time 8.984 (3.323)      Data 0.000 (0.858)      Loss 2.8173 (2.7268)    Prec@1 36.328 (41.568)  Prec@5 65.625 (66.082)
Epoch: [5][50/626]     Time 0.940 (3.095)      Data 0.000 (0.690)      Loss 2.8278 (2.7216)    Prec@1 36.719 (41.475)  Prec@5 62.109 (66.131)
```





NSML – Use cases

- Automatic hyperparameter tuning with the AutoML feature
  - CIFAR100 with Wide ResNet



NSML – Use cases

- Automatic hyperparameter tuning with the AutoML feature
  - AutoML beated the handy-tuned baseline!

Algorithm	Result by the author(s)	Configuration by the author(s)	Result by AutoML	Proposed configuration by AutoML
ResNet with RE	78.34%	depth: 110 momentum: 0.9 lr: 0.1 prob: 0.5 sh: 0.4 parameter: 1.73M	**78.92%**	depth: 110 momentum: 0.8065 lr: 0.1167 prob: 0.3044 sh: 0.4088 parameters: 1.73M
Wide ResNet with RE	82.31%	depth: 28 widen_factor: 10 prob: 0.5 sh: 0.4 momentum: 0.9 dropout: 0 lr: 0.1 parameter: 36.54M	**83.1%**	depth: 112 widen_factor: 10 prob: 0.52 sh: 0.8297 momentum: 0.615 dropout: 0.4817 lr: 0.2036 parameter: 172.07M
PyramidNet	82.11%	alpha: 200 bottleneck: True depth: 272 lr: 0.1 momentum: 0.9 weight_decay: 0.0001	**82.58%**	alpha: 275 bottleneck: True depth: 165 lr: 0.2325 momentum: 0.9491 weight_decay: 0.0001

Upcoming Features (available soon)

- Distributed learning
  - If you want to mobilize dozens of GPUs, you got the right place ☺
  - At this time, we just test on pytorch
  - The general guideline will be announced soon
- Hyperparameter tuning
  - For the meantime, only an authorized member per each team will test this feature
- Deploying the models for practical services
  - Your models can be deployed for commercial services through public accessible URL.
  - API server is under developing

Upcoming Features (voice of customers)

- Those features are requested or suggested by the existing users.
  - NSML is on-going project, so we listen to you attentively :)

- Job management
  - Job queueing when the cluster has insufficient resources
- User interface
  - Name aliasing or tagging for visualization (done)
  - UI view enhancement (e.g. freeze common info)
  - Session pulling with full meta-data
- Model deployment
  - Elastic resource allocation for heavy traffic

Upcoming Features (voice of customers)

- Dataset
  - Multiple datasets (e.g. corpus and embedding weights)
  - Dataset revision
- Visualization
  - More complicated visualization than the scalar plot
- Supporting for other platform rather than python
- Documentations and Announcements

Summary

- NSML has improved ML works efficiently with large scale GPU clusters .
- NSML provides many convenient UIs and visualization tools for analyzing your working models.
- By the existing usages and results, NSML demonstrated its usefulness for enhancing your ML works :)

References

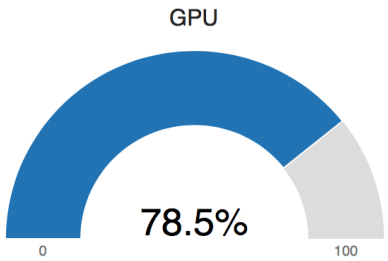
- https://nsml.navercorp.com/
- https://pages.oss.navercorp.com/nsml/docs.nsml/_build/html/index.html

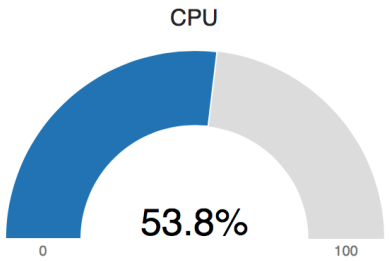
- https://pytorch.org/docs/stable/index.html
- https://www.tensorflow.org/api_docs/

- Contacts
  - https://oss.navercorp.com/nsml/nsml_QA/issues

Thank You!

- Even though the typhoon is coming…





Allocated GPU	Total GPU	Allocated CPU	Total CPU
446	568	1146	2132

Resource status

GPU	CPU	RAM (GB)
0	30	42.35
1	32	138.35
2	31	138.35
3	36	113.35
4	36	112.35
5	11	163.35
7	14	188.35

I tip my hat to you!