MindSpore

MindSpore TSC Meeting
May 20 2021

Antitrust Policy Notice

- MindSpore community meetings involve participation by industry competitors, and it is the intention of the MindSpore community to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable antitrust and competition laws in the member representative's nation or state.

MindSpore Useful Information

- Website: www.mindspore.cn (Chinese/English Display)
- Gitee: https://gitee.com/mindspore GitHub: https://github.com/mindspore-ai

  iHub:https://code.ihub.org.cn/companies/4vioxkz2
- Mailing Lists: https://mailweb.mindspore.cn/postorius/lists/mindspore-tsc.mindspore.cn/
- Logo:
  - https://gitee.com/mindspore/community/blob/master/MindSpore-logo.png
  - https://github.com/mindspore-ai/community/blob/master/MindSpore-logo.png
- Presentation Template:
  - https://gitee.com/mindspore/community/tree/master/slides
  - https://github.com/mindspore-ai/community/tree/master/slides
- Charter:
  - https://gitee.com/mindspore/community/blob/master/governance.md
  - https://github.com/mindspore-ai/community/blob/master/governance.md

Agenda

- Roll Call and Approval for previous minutes
- Community Progress Update
- SIGs/WGs Update
- Release Plan Review
- Operational Matters

Roll Call
(First name alphabetically ordered)

MindSpore

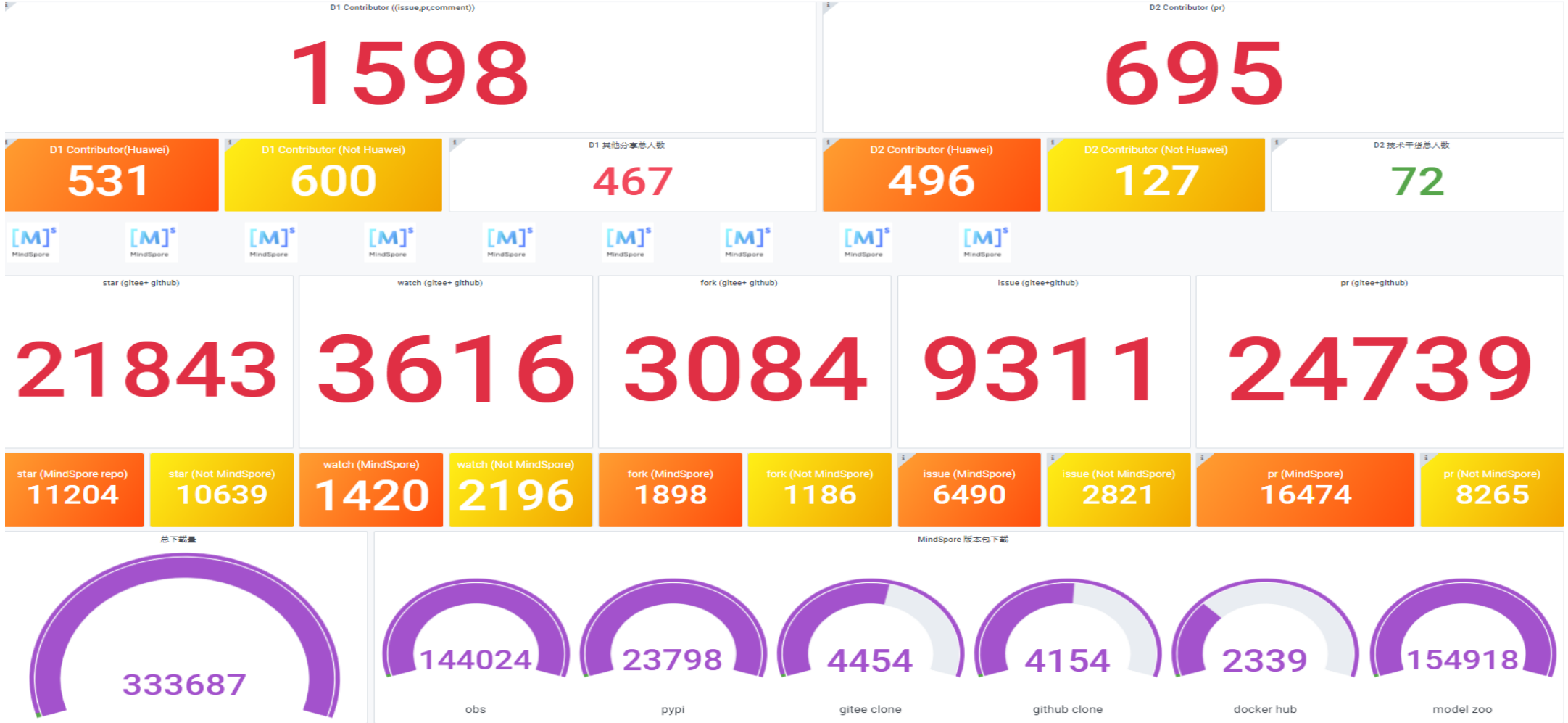
Affiliation	**TSC Member**
University of Edinburgh	Amos Storkey
Conic AI Technology	Han Xiao
ICBC's Big Data and Artificial Intelligence Lab	Jianjun Chen
Tsinghua University	Jun Zhu
University Paris-Sacla	Joel Falcou
Apulis Technology	Jin Li
Huawei	Lei Chen **(Chair)**
Xidian University	Maoguo Gong
Imperial College London	Peter Pietzuch
Key Lab of Intelligent Information Processing of the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS)	Shiguang Shan
University of Muenster	Sergei Gorlatch
Harbin Institute of Technology	Tonghua Su
University of Science and Technology of China	Xiangyang Li
Peking University/Pengcheng Lab	Yonghong Tian

Approval of previous minutes

- All the meeting notes and slides could be found at:
  - ❑ https://github.com/mindspore-ai/community/tree/master/tsc/meeting-notes
  - ❑ https://github.com/mindspore-ai/community/tree/master/tsc/slides

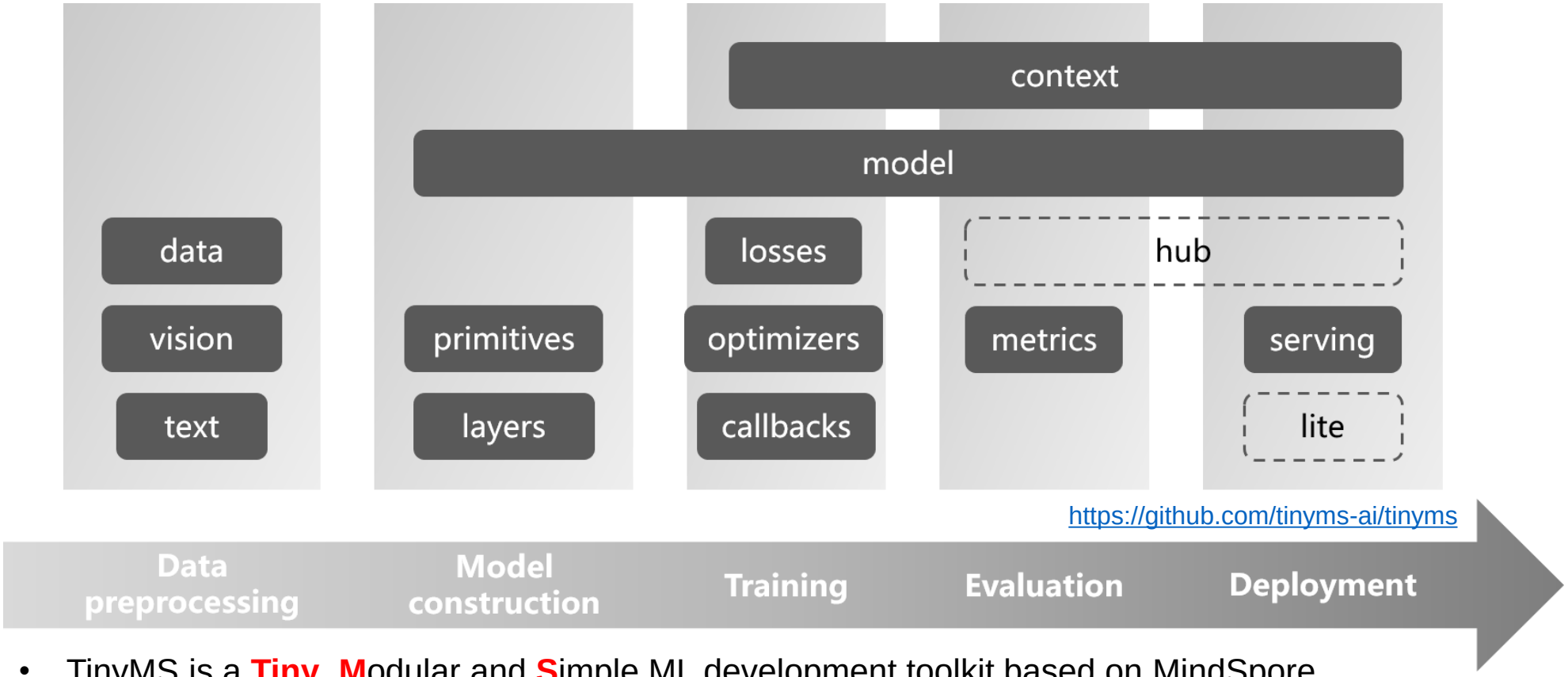
- 2021 Apr TSC meeting recording:
  - ❑ https://www.bilibili.com/video/BV1v5411g7px

Community Progress Update



2021 Apr: **26.7w** download, **25%** growth

Community Progress Update: TinyMS Development Toolkit



- TinyMS is a **Tiny**, **M**odular and **S**imple ML development toolkit based on MindSpore.
- Enable users to experience AI apps development and deployment processes in **1min**.
- With nanny-level deep learning online tutorials, especially friendly to **machine learning beginners**.

Community Progress Update: TinyMS Development Toolkit

MindSpore

Data preprocessing

Step 1: data download

TinyMS supports users to complete the download, decompression and format adjustment operations of the data set with one click.

```
from tinyms.data import download_dataset

mnist_path = download_dataset('mnist', local_path='./')
```

```
from tinyms.data import MnistDataset

mnist_ds = MnistDataset(mnist_path, shuffle=True)
```

Step 2: data loading

By completely inheriting MindSpore's native API, users can use the xxxDataset interface to instantiate different data sets very conveniently.

Step 3: data processing

TinyMS directly corresponds to the processing of the dataset itself through the DatasetTransform interface, allowing users to utilize a single piece of data or the entire dataset with just one line of code.

```
from PIL import Image
from tinyms.vision import mnist_transform

# Preprocessing a single one picture
img = mnist_transform(Image.open('picture.jpg'))
# Apply preprocessing to MnistDataset class instance
mnist_ds = mnist_transform.apply_ds(mnist_ds)
```

TinyMS

Community Progress Update: TinyMS Development Toolkit

MindSpore

Model construction

Key features:

- Super easy to instantiate a network
- Provide a high-level Model API with the consistent development experience for TinyMS users

TinyMS

```
from tinyms.model import lenet5

net = lenet5(class_num=10)
```

AI Frameworks (e.g. MindSpore)

```
import mindspore.nn as nn

class LeNet5(nn.Cell):
    def __init__(self, num_class=10, num_channel=1):
        super(LeNet5, self).__init__()
        self.conv1 = nn.Conv2d(num_channel, 6, 5, pad_mode='valid')
        self.conv2 = nn.Conv2d(6, 16, 5, pad_mode='valid')
        self.relu = nn.ReLU()
        self.max_pool2d = nn.MaxPool2d(kernel_size=2, stride=2)
        self.flatten = nn.Flatten()
        self.fc1 = nn.Dense(16 * 5 * 5, 120, weight_init=Normal(0.02))
        self.fc2 = nn.Dense(120, 84, weight_init=Normal(0.02))
        self.fc3 = nn.Dense(84, num_class, weight_init=Normal(0.02))

    def construct(self, x):
        x = self.max_pool2d(self.relu(self.conv1(x)))
        x = self.max_pool2d(self.relu(self.conv2(x)))
        x = self.flatten(x)
        x = self.relu(self.fc1(x))
        x = self.relu(self.fc2(x))
        x = self.fc3(x)
        return x

net = LeNet(class_num=10)
```

TinyMS

MindSpore

Community Progress Update: TinyMS Development Toolkit

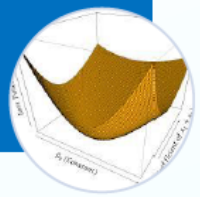
Training

- TinyMS high-level API encapsulates the loss functions at the network level.

Loss function

- No need to know how optimizer works and use it with a single line of code.

Optimizer



- Automatically configure the training callback strategy.

Callback function

```
from tinyms.losses import SoftmaxCrossEntropyWithLogits
from tinyms.optimizers import Momentum

net_loss = SoftmaxCrossEntropyWithLogits(sparse=True, reduction='mean')
net_opt = Momentum(net.trainable_params(), 0.01, 0.9)
```

```
from tinyms.callbacks import mobilenetv2_cb

net_cb = mobilenetv2_cb()
```

TinyMS

Community Progress Update: TinyMS Development Toolkit

MindSpore

Evaluation

Metrics class	Introduction	Metrics class	Introduction
Accuracy	Calculates the accuracy for classification and multilabel data.	Metric	Base class of metric.
MAE	Calculates the mean absolute error.	Recall	Calculates recall for classification and multilabel data.
MSE	Measures the mean squared error.	Fbeta	Calculates the fbeta score.
Precision	Calculates precision for classification and multilabel data.	F1	Calculates the F1 score.
TopKCategoricalAccuracy	Calculates the top-k categorical accuracy.	Loss	Calculates the average of the loss.

How to use?

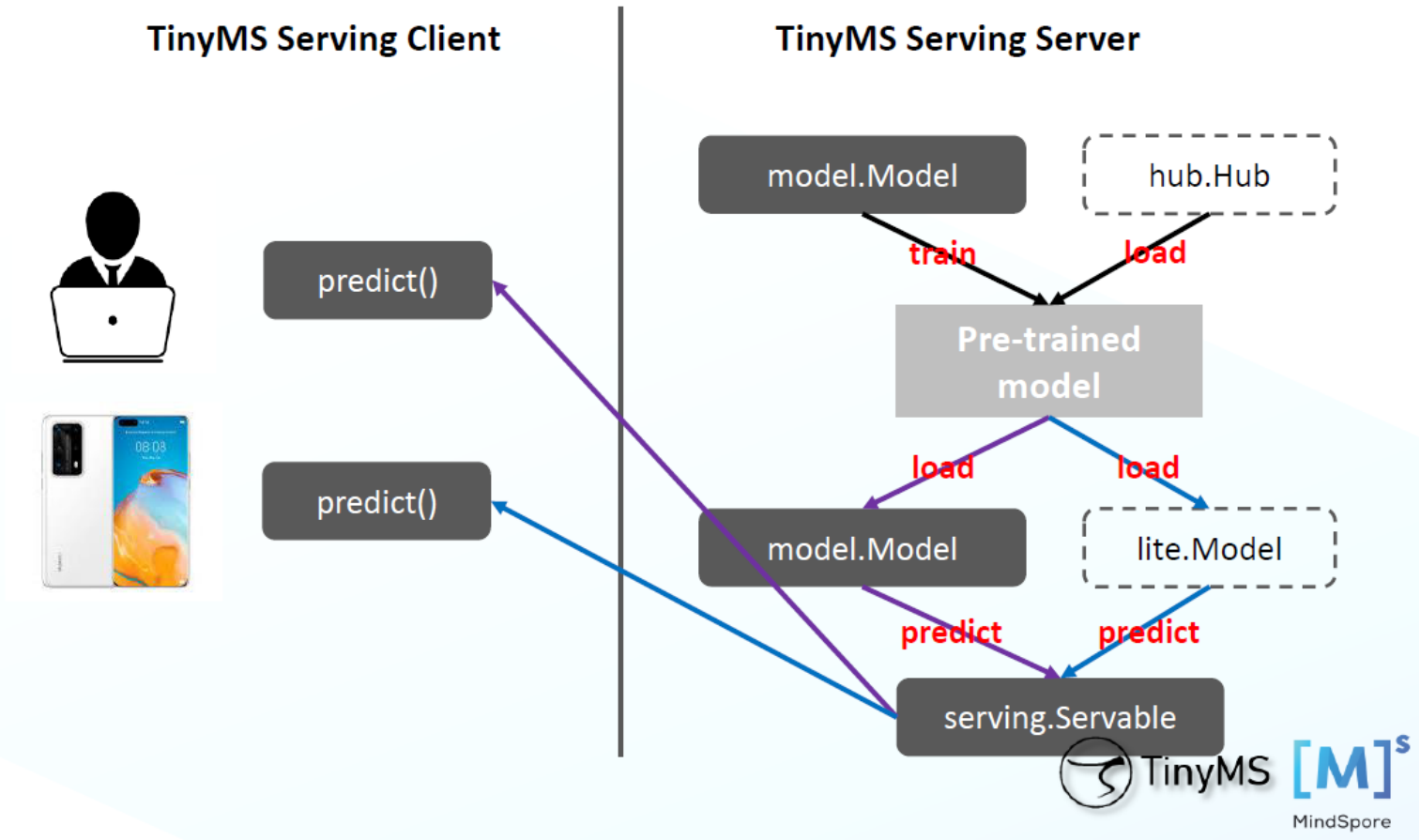
```
from tinyms.model import Model
from tinyms.metrics import Accuracy

model = Model(net)
model.compile(metrics={"Accuracy": Accuracy()})
model.eval(ds_eval)
```

TinyMS

Community Progress Update: TinyMS Development Toolkit

Deployment

Serving is a module that supports performing model deploy and inference on both computers and mobile phones, with a **unified predict interface** to users.



SIGs/WGs Update

- None

Next Step

- MindSpore v1.3.0 Release Roadmap
- MindSpore Developer Conference

THANK YOU