TensorFlow

Курс “Практическое применение по TensorFlow”

Шигапова Фирюза Зинатуллаевна

1-й семестр, 2019 г.

https://github.com/Firyuza/TensorFlowPractice

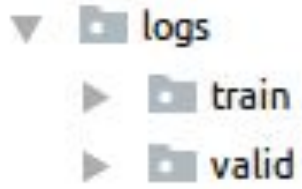
Quiz. Gradient Tape

Will differentiates loss_fn() **AND** other_loss_fn()? Explain.

```
with tf.GradientTape() as t:
  loss = loss_fn()
with tf.GradientTape() as t:
  loss += other_loss_fn()
t.gradient(loss, ...)
```

Create TensorBoard writer

```
self.valid_file_writer = tf.summary.create_file_writer(cfg.train.logs_base_dir + "/valid")
self.train_file_writer = tf.summary.create_file_writer(cfg.train.logs_base_dir + "/train")
```



Log scalar

```
with self.train_file_writer.as_default():
    tf.summary.scalar('reg_loss', reg_loss, global_step)
```

Name (tag)

Value

Step

Log text (table)

```
tf.summary.text(name, text_data, step=step)
```

Table

```
hyperparameters = [tf.convert_to_tensor(row) for row in rows]
with summary_writer.as_default():
    tf.summary.text(name, tf.stack(hyperparameters), step=step)
```

Create rows as in table

Each row: `rows = [['key1', 'value1'], ['key2', 'value2']]`

GES TEXT HPARAMS

Filter tags (regular expressions supported)

config

config
tag: config

step 0

dataset.nrof_classes	10
train.nrof_classes	10
train.image_size	32
train.image_channels	3
train.batch_size	32
train.epoch_size	5000

Log hyperparameters

Adding hyperparameters of the model to the tensorboard

```
from tensorboard.plugins.hparams import api as hp
```

Log hyperparameters

```
with self.train_file_writer.as_default():
    hp.hparams({'image_size': cfg.train.image_size,
                'optimizer': cfg.train.optimizer,
                'batch_size': 32, 'lr': cfg.train.learning_rate})
```

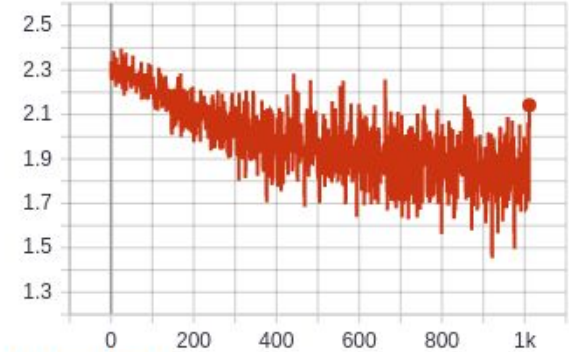
HPARAMS

INACTIVE

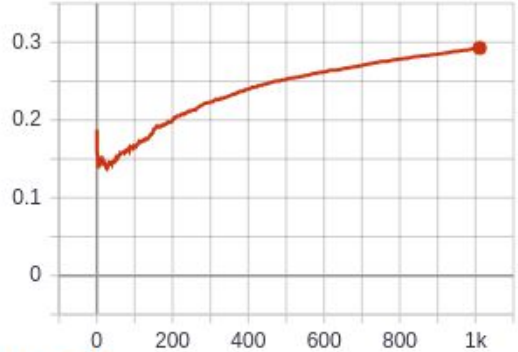
TABLE VIEW | PARALLEL COORDINATES VIEW | SCATTER PLOT MATRIX VIEW

Trial ID	Show Metrics	image_size	lr	optimizer	batch_size	CE_loss
15981fea47efc52...	☑	32.000	0.000030000	ADAM	32.000	2.1418

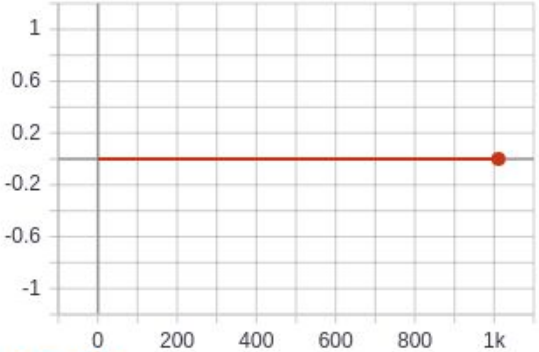
CE_loss



category_accuracy



learning_rate



reg_loss

total_loss

Log image

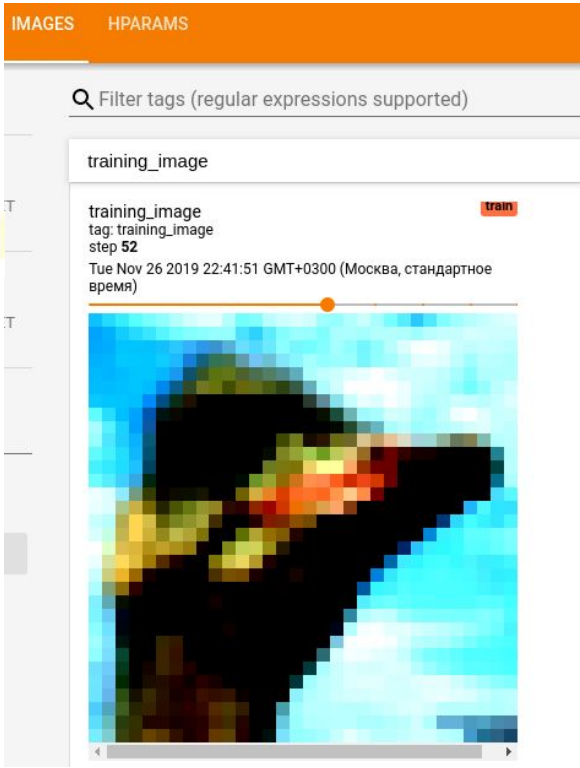
```
tf.summary.image('training_image', [images[0]], step=global_step)
```

One image tensor

```
tf.summary.image('training_images', images[:4], step=global_step)
```

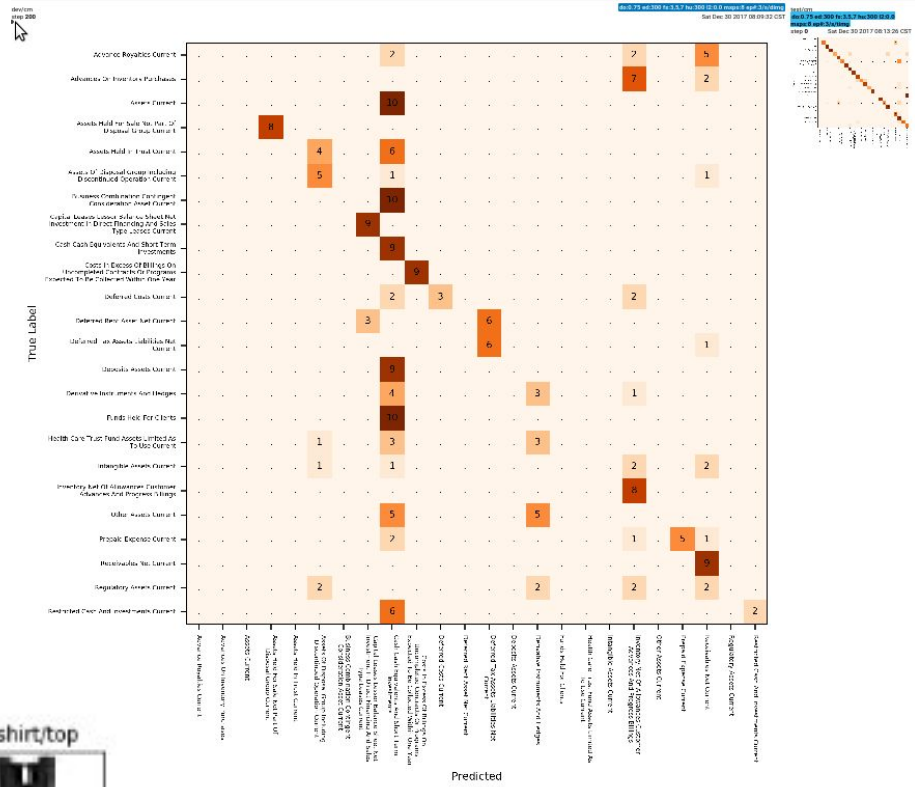
Multiple image tensors

```
self.custom_category_metric = CustomSparseCategoricalAccuracy('custom_category_metric')
```



Log image

- Log Confusion Matrix
- Log plots



Source: https://www.tensorflow.org/tensorboard/image_summaries; https://stackoverflow.com/questions/41617463/tensorflow-confusion-matrix-in-tensorboard

Log image. Plot

Create figure

```
def image_grid(images, class_labels, predicted_class_labels, class_name_map):
    # Create a figure to contain the plot.
    figure = plt.figure(figsize=(10, 10))
    for i in range(len(images)):
        # Start next subplot.
        plt.subplot(5, 5, i + 1, title='%s_%s' %
                                        (class_name_map[class_labels[i]],
                                         class_name_map[predicted_class_labels[i]]))
        plt.xticks([])
        plt.yticks([])
        plt.grid(False)
        plt.imshow(images[i], cmap=plt.cm.binary)

    return figure
```

Log image. Plot

Convert figure into PNG image and save in memory

```
def plot_to_image(figure):
    """Converts the matplotlib plot specified by 'figure' to a PNG image and
    returns it. The supplied figure is closed and inaccessible after this call."""
    # Save the plot to a PNG in memory.
    buf = io.BytesIO()
    plt.savefig(buf, format='png')
    # Closing the figure prevents it from being displayed directly inside
    # the notebook.
    plt.close(figure)
    buf.seek(0)
    # Convert PNG buffer to TF image
    image = tf.image.decode_png(buf.getvalue(), channels=4)
    # Add the batch dimension
    image = tf.expand_dims(image, 0)

    return image
```

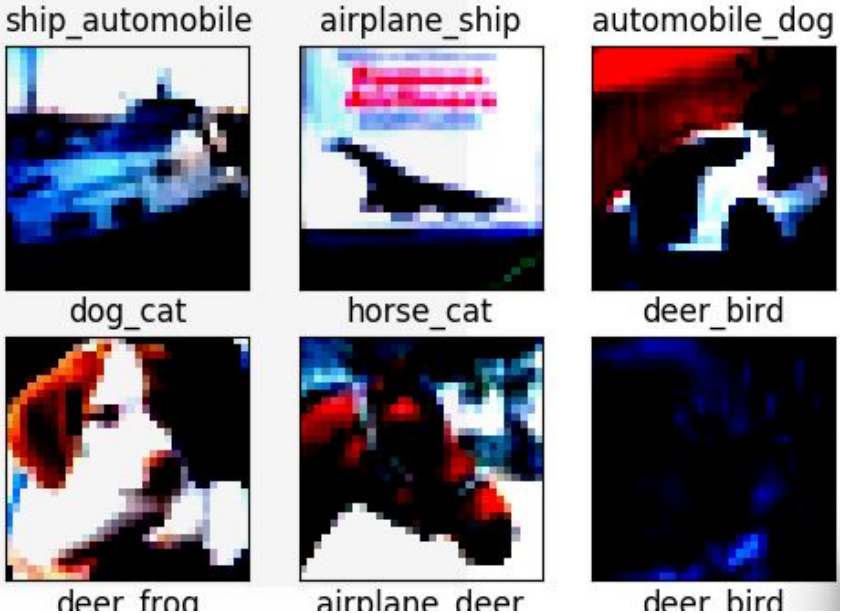
valid_images
tag: valid_images
step **12 504**

Wed Dec 04 2019 00:12:53 GMT+0300 (Москва, стандартное время)

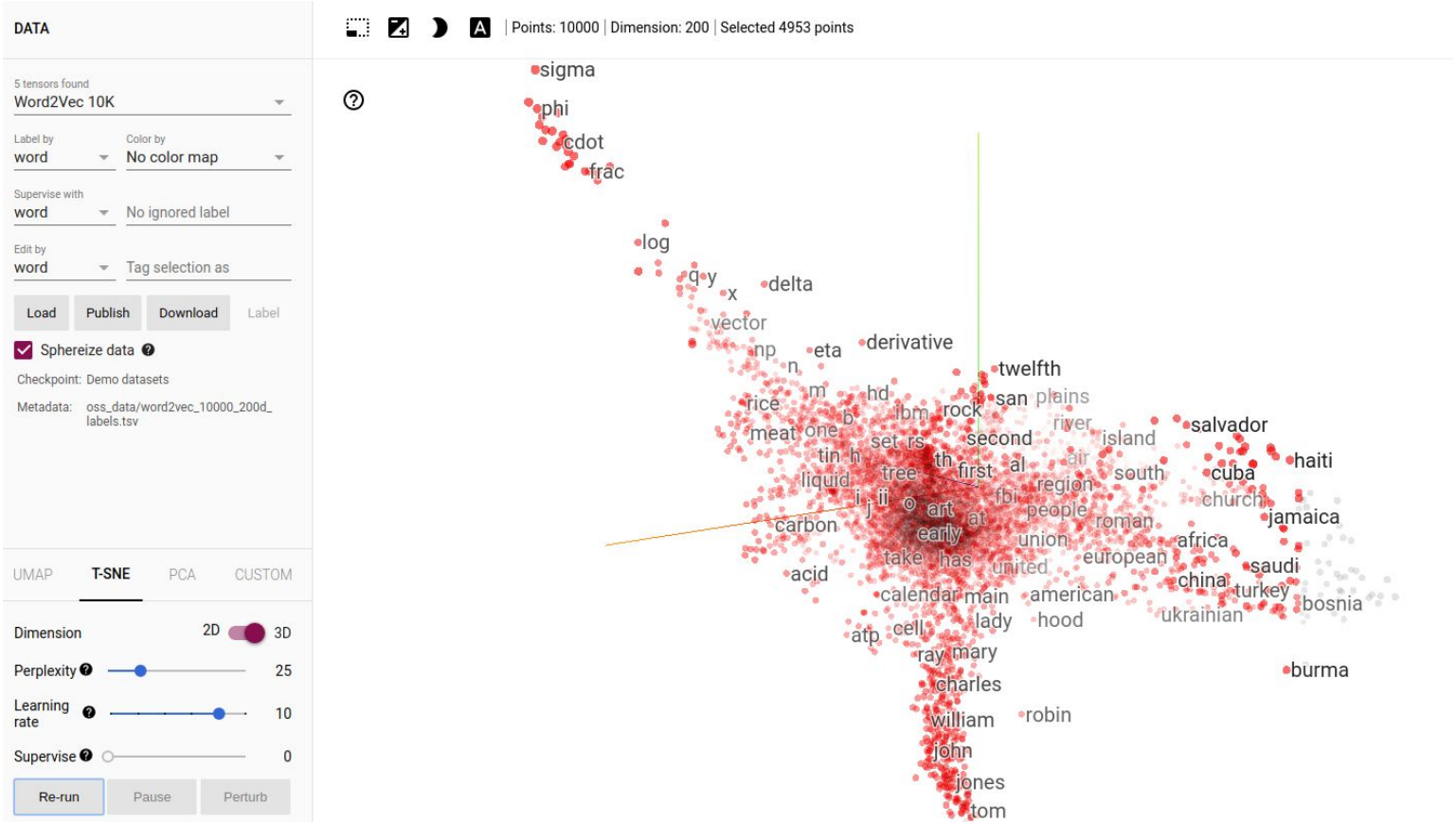
valid

valid_images
tag: valid_images
step **12 504**

Wed Dec



Visualize embeddings



TensorBoard.dev

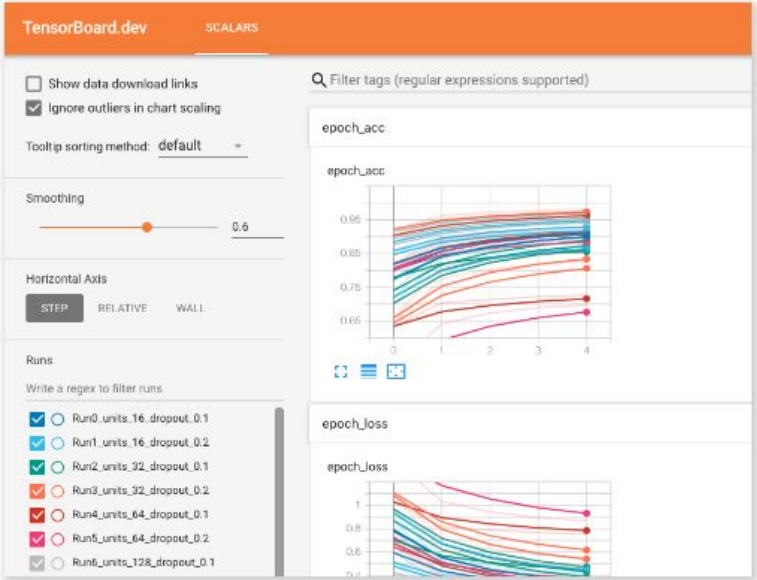
TensorBoard.dev TensorBoard TensorFlow

TensorBoard.dev PREVIEW

Easily host, track, and share your ML experiments for free.

A managed TensorBoard experience that lets you upload and share your ML experiment results with anyone.

Get started Example Colab



TensorBoard.dev

Install the latest TensorBoard. You may need to first uninstall other TensorBoard versions.

```
$ pip install -U tensorboard
```

For help, run "tensorboard dev --help" or "tensorboard dev COMMAND --help"

```
$ tensorboard dev upload --logdir logs
```

```
Upload started and will continue reading any new data as it's added
to the logdir. To stop uploading, press Ctrl-C.
View your TensorBoard live at: https://tensorboard.dev/experiment/7nZx2lcBSWq1sw
3JIXUNqw/
```

tensorboard.dev TensorBoard

TensorBoard.dev SCALARS SEND FEEDBACK

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

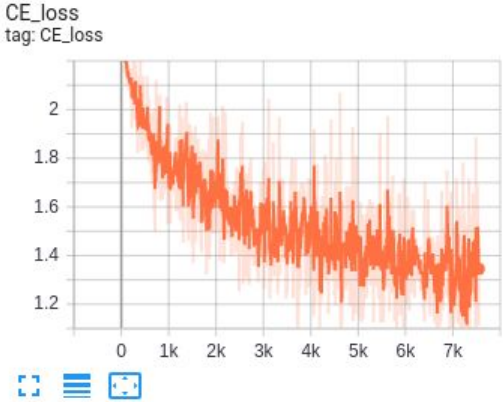
train

TOGGLE ALL RUNS

experiment 7nZx2lcBSWq1sw3JIXUNqw

Filter tags (regular expressions supported)

CE_loss



category_accuracy

learning_rate

reg_loss

total_loss

Корзина

MLflow

Open-source platform for managing the end-to-end machine learning lifecycle.

```
$ pip install mlflow
```

Source: https://mlflow.org/docs/latest/index.html

MLflow

- **MLflow Tracking**: log parameters, metrics etc.
- **MLflow Projects**: save model
- **MLflow Models**: managing and deploying

MLflow Tracking

- Code Version
- Start & End Time
- Source
- Parameters
- Metrics
- Artifacts

MLflow

Experiment name as global scope for runs

```
import mlflow
mlflow.set_experiment(cfg.train.experiment_name)
```

Experiments

Default

ImageClassification

ImageClassification

Experiment ID : 1

Artifact Location : file:///home/firiuza/PycharmProjects/TensorflowPractice1/Lesson11/mlruns/1

Description:

Search Runs: metrics.rmse < 1 and params.model = "tree"

State: Active

Search

Filter Params: alpha, lr

Filter Metrics: rmse, r2

Clear

Showing 1 matching run

Compare

Delete

Download CSV

Date	User	Run Name	Source	Version	Tags	Parameters
2019-11-29 02:14:01	firiuza	First Train	image...	5ac9ba		

MLflow Tracking. Start

```
with mlflow.start_run():
    img_net = ImageClassification()
    img_net.run_train()
```

Using context manager in Python
run code within MLflow scope

MLflow Tracking. Start

```
with mlflow.start_run(run_name=cfg.train.run_name):
    mlflow.log_artifact(os.path.abspath(__file__))

    img_net = ImageClassification()
    img_net.run_train()
```

- Start with defining **run_name**
- Log code as **artifact**

mlflow

GitHub Docs

Experiments

Default

Default

Experiment ID: 0

Artifact Location: file:///home/firiuza/PycharmProjects/TensorflowPractice1/Lesson11/mlruns/0

Description:

Search Runs: metrics.rmse < 1 and params.model = "tree"

State: Active

Search

Filter Params: alpha, lr

Filter Metrics: rmse, r2

Clear

Showing 1 matching run

Compare

Delete

Download CSV

	Date	User	Run Name	Source	Version	Tags	Parameter
	2019-11-27 11:17:08	firiuza	ImageClass...	image...	da6432		

Your run

MLflow Tracking. Log artifact

Log local files or directory as artifact

```
mlflow.log_artifact(os.path.abspath(__file__))
```

MLflow Tracking. Log param

```
mlflow.log_param('lr', cfg.train.learning_rate)
mlflow.log_param('optimizer', cfg.train.optimizer)
```

MLflow Tracking. Log metric

```
from mlflow import log_metric, log_artifact

log_metric('validation_category_accuracy', category_metric.result().numpy(), step)
```

MLflow. How to run?

In terminal:

```
cd folder/where/mlruns/exist
```

```
mlflow ui
```

```
[2019-12-05 23:02:18 +0300] [9231] [INFO] Starting gunicorn 20.0.2
[2019-12-05 23:02:18 +0300] [9231] [INFO] Listening at: http://127.0.0.1:5000 (9231)
[2019-12-05 23:02:18 +0300] [9231] [INFO] Using worker: sync
[2019-12-05 23:02:18 +0300] [9235] [INFO] Booting worker with pid: 9235
```

ImageClassification > First Train

Date: 2019-12-05 21:59:35

Source: image_classification.py

Git Commit: 0fbe352

User: firiuza

Duration: 58.9min

Notes

None

Parameters

Name	Value
lr	3e-05
optimizer	ADAM

Metrics

Name	Value
validation_category_accuracy	0.681

Tags

Name	Value	Actions

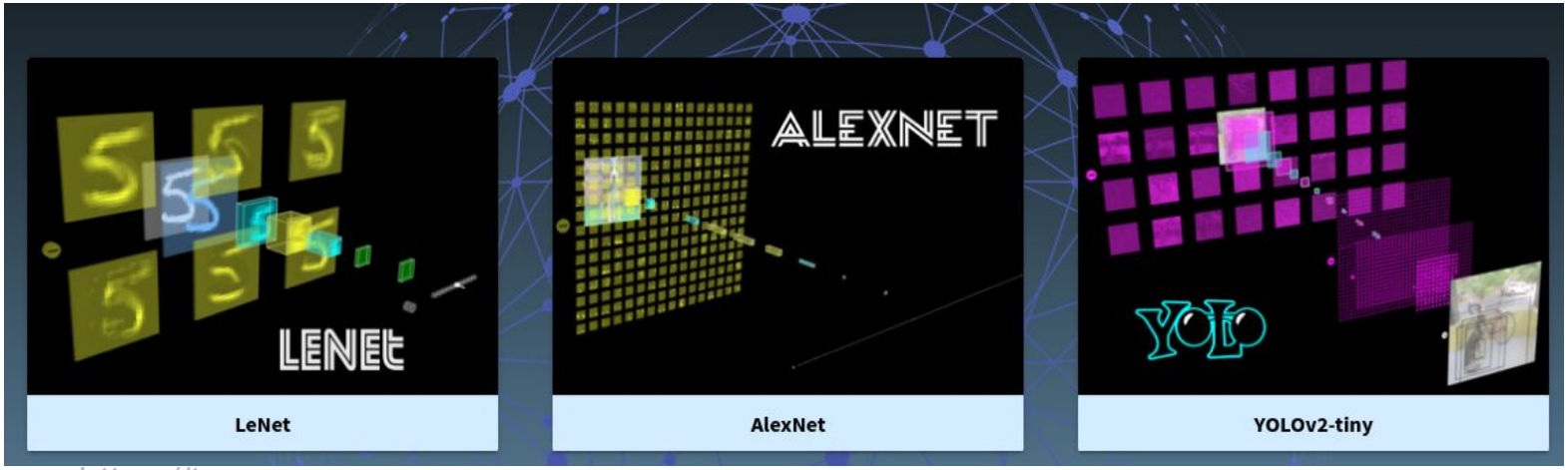
MLflow Models

Save model and allow to deploy on a local machine and to several production environments.

MLflow Projects

Packaging data science code.

Tensorspace js



Source: https://tensorspace.org

Tensorspace js. Installation

```
$ conda create -n envname python=3.6
$ source activate envname
$ pip install tensorspacejs
```

Important step:

```
$ tensorspacejs_converter -init
```

Source: https://github.com/tensorspace-team/tensorspace-converter/blob/master/README.md

Tensorspace js. Convert model

```
tensorspacejs_converter \
    --input_model_from="tensorflow" \
    --input_model_format="tf_keras" \
    --output_layer_names="conv_1,maxpool_1,conv_2,maxpool_2,dense_1,dense_2,softmax" \
    ./rawModel/tf_keras_model.h5 \
    ./convertedModel
```

Assignment

1. Add TensorBoard logging into train and validation pipeline:
   - scalar
   - text (config file)
   - hyperparameters
   - images
2. Add MLflow into train and validation pipeline:
   - artifact (code files)
   - params
   - metrics
3. Commit screenshots with code.