TensorFlow

GDG NASHIK

Using TensorFlow Hub to make better AI applications

Rishit Dagli

10 STD, TEDX, Ted-Ed speaker|Google certified mobile site developer|Intel AI Scholar|2XGCP Champ|Mozilla Mumbai Lead

rishit.tech

Ideal Audience

- Devs who having worked on Deep Learning Models (Keras)
- Developers looking for ways to make better models

GDG NASHIK

Agenda

- Motivation behind TF Hub
- What is TF Hub?
- Why TF Hub?
- Using TF Hub
  - Transfer Learning
  - Image Feature Vectors
- Text based models and embeddings
- Demos!!
- Quick Recap
- Q & A

Motivation behind TF Hub

Motivation behind TF Hub

Motivation behind TF Hub

- Make using open source code easy

Motivation behind TF Hub

- Make using open source code easy
- Build applications faster

Motivation behind TF Hub

- Make using open source code easy
- Build applications faster
- Build better applications

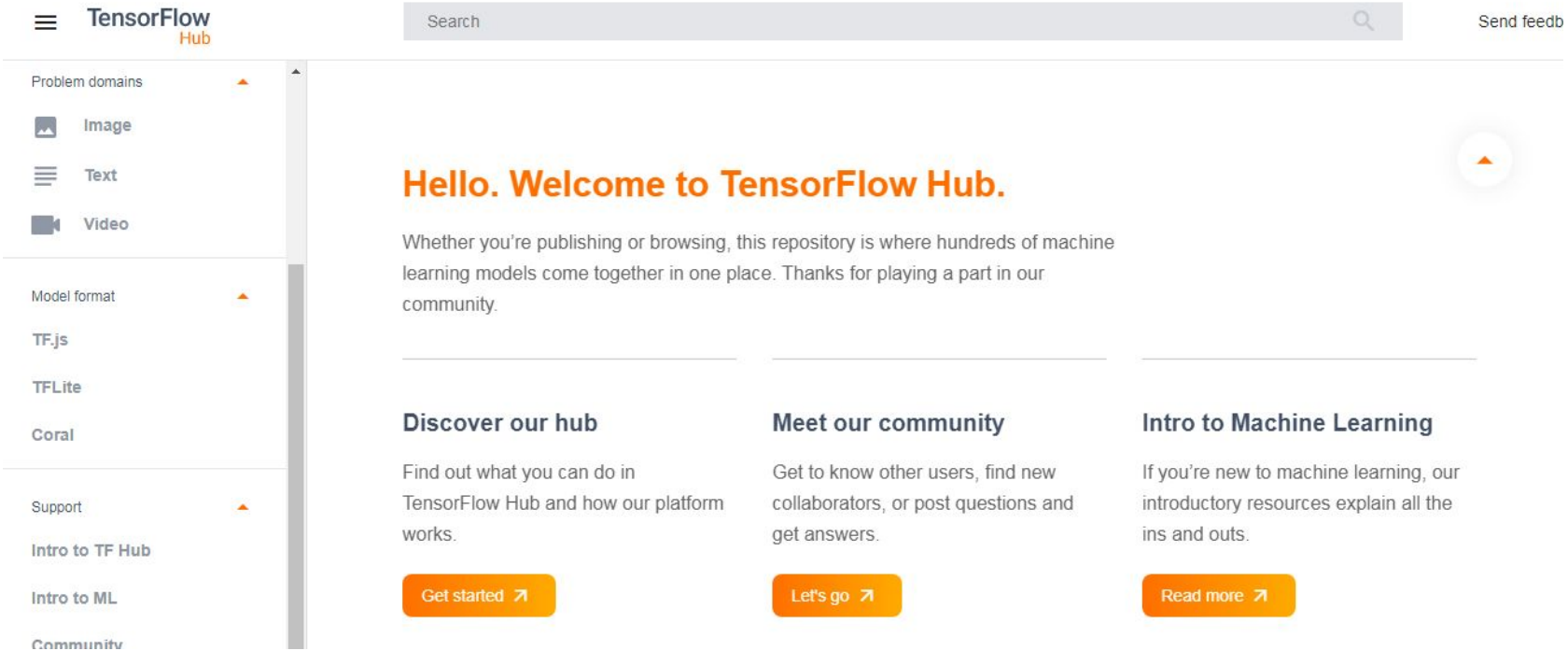
What is TF Hub?

TensorFLow Hub

- Modelling is an important part

TensorFLow Hub

tfhub.dev



TensorFLow Hub

- Modelling is an important part
- State of the art models

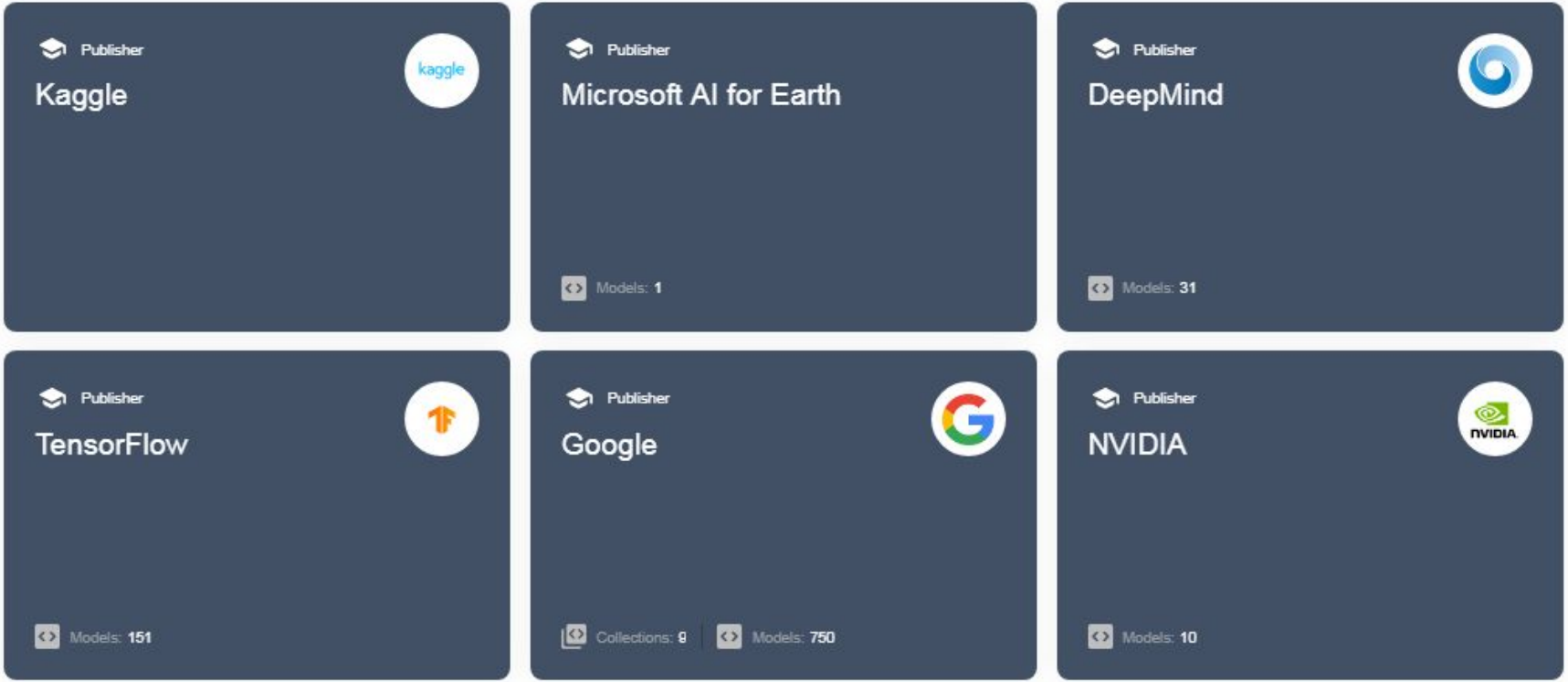
TensorFLow Hub

- Modelling is an important part
- State of the art models
- Ease in using and integrating with model APIs

TensorFLow Hub

- Modelling is an important part
- State of the art models
- Ease in using and integrating with model APIs
- A wide array of publishers

TensorFLow Hub



Why TF Hub?

Why TF Hub?

- Do you even need TF Hub?

Why TF Hub?

- Do you even need TF Hub?
- The API

Why TF Hub?

- Do you even need TF Hub?
- The API
- Transfer Learning

Why TF Hub?

- Do you even need TF Hub?
- The API
- Transfer Learning
  - Generalization

Why TF Hub?

- Do you even need TF Hub?
- The API
- Transfer Learning
  - Generalization
  - Less data

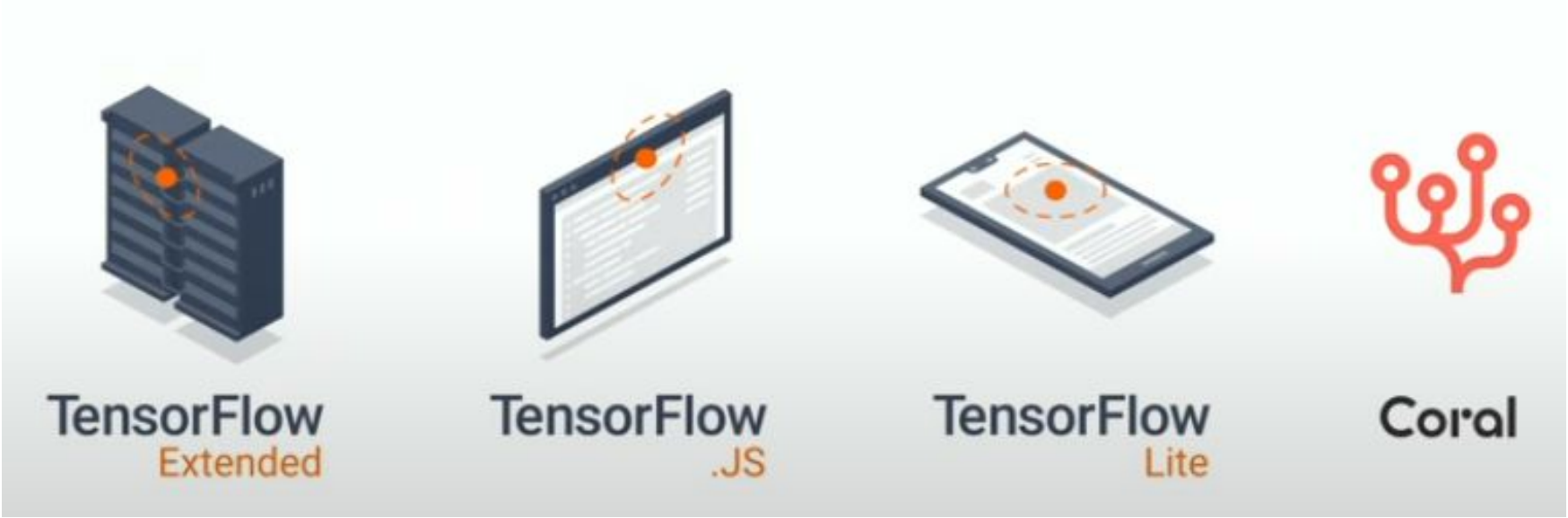
Why TF Hub?

- Do you even need TF Hub?
- The API
- Transfer Learning
  - Generalization
  - Less data
  - Training time

Why TF Hub?

- Do you even need TF Hub?
- The API
- Transfer Learning
  - Generalization
  - Less data
  - Training time
- Without code dependencies

Why TF Hub?

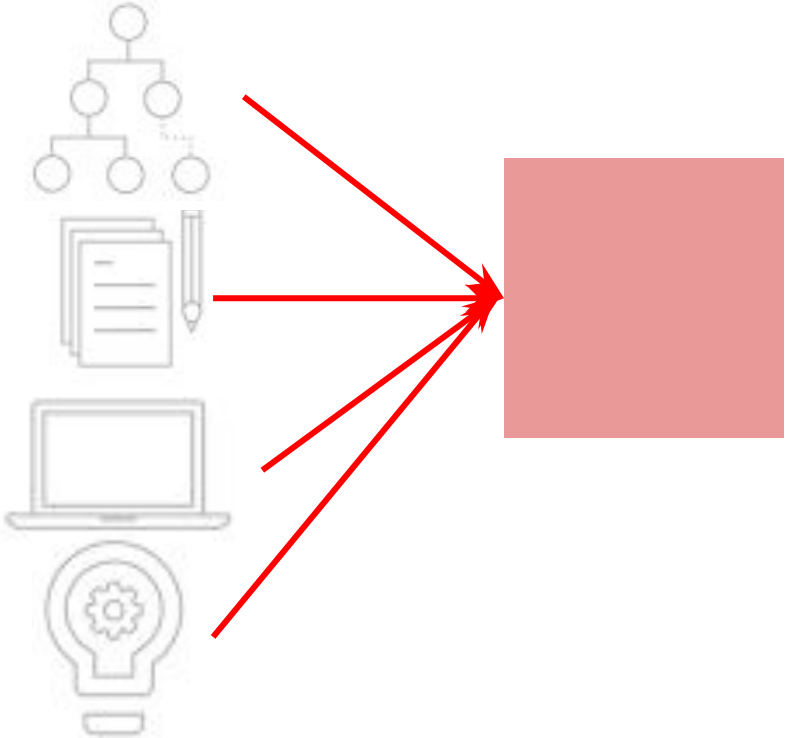


Credits: @TheSandeepGupta

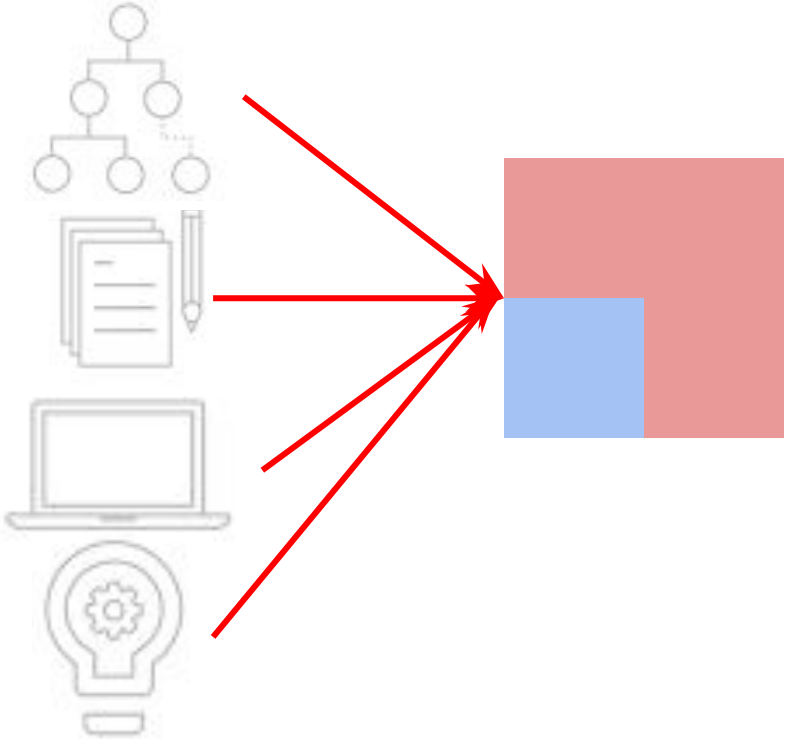
Using TensorFLow Hub

Transfer Learning

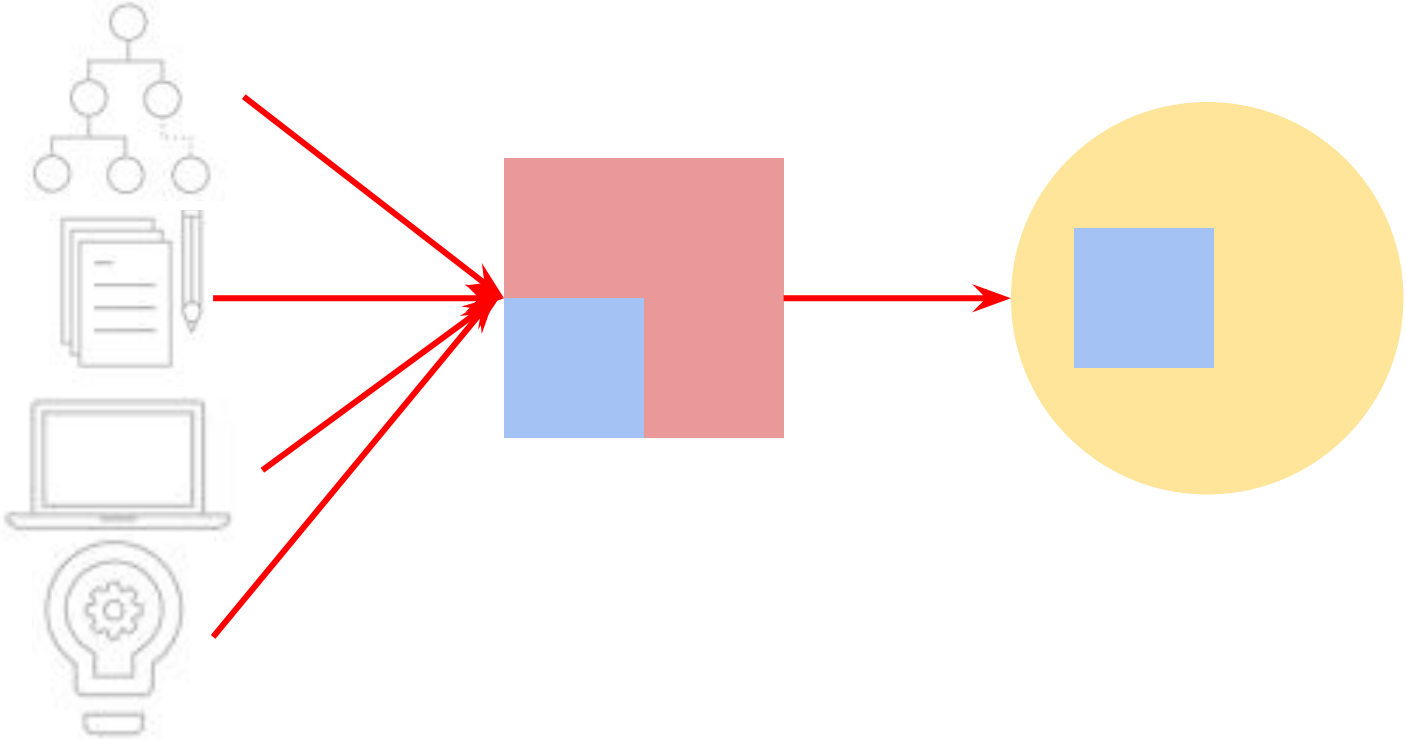
Transfer Learning



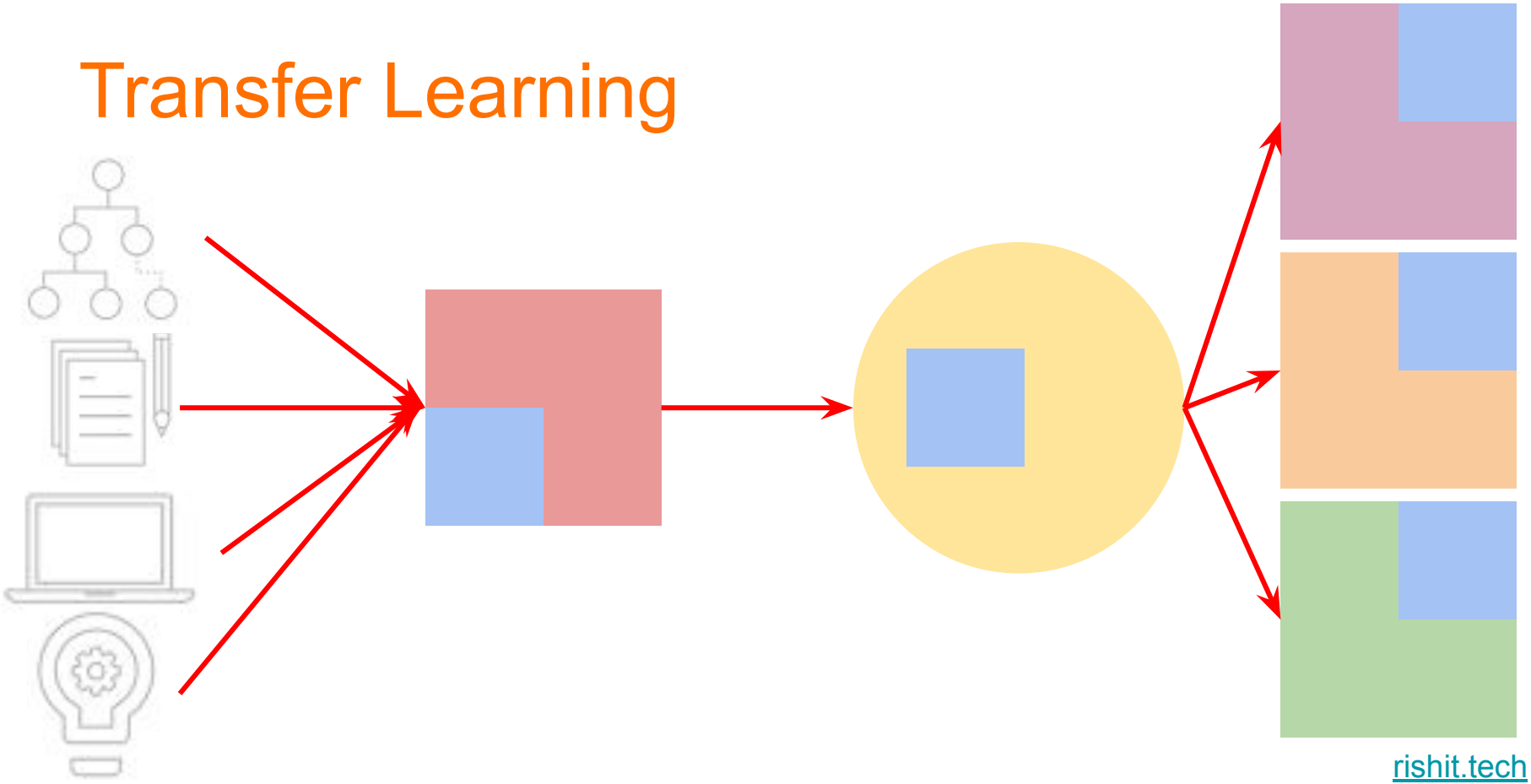
Transfer Learning



Transfer Learning



Transfer Learning



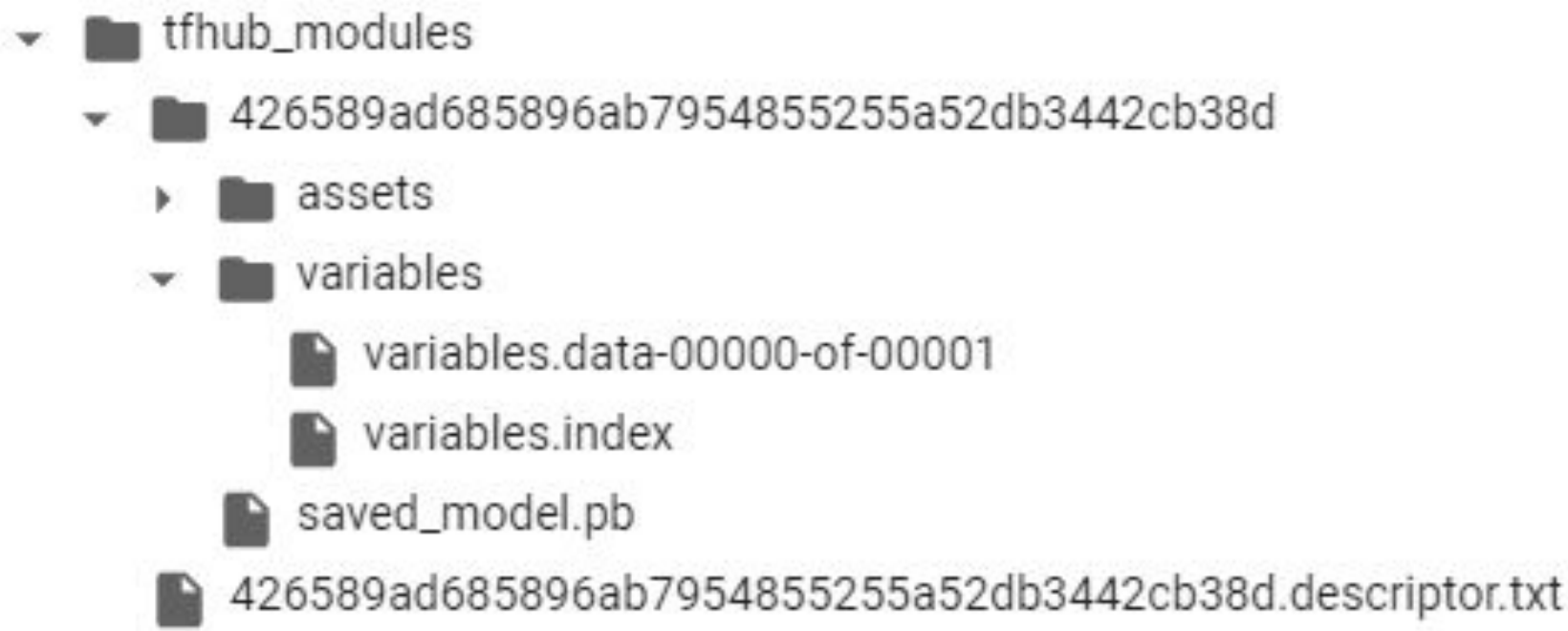
Installation

https://www.tensorflow.org/hub/installation

Loading models

```
MODULE_HANDLE =
'https://tfhub.dev/google/imagenet/inception_v3/classification/4'

module = hub.load(MODULE_URL)
```


tfhub_modules
426589ad685896ab7954855255a52db3442cb38d
assets
variables
variables.data-00000-of-00001
variables.index
saved_model.pb
426589ad685896ab7954855255a52db3442cb38d.descriptor.txt

Saved Model CLI

```
!saved_model_cli show --dir [DIR] --all
```

```
MetaGraphDef with tag-set: 'serve' contains the following SignatureDefs:

signature_def['__saved_model_init_op']:
  The given SavedModel SignatureDef contains the following input(s):
  The given SavedModel SignatureDef contains the following output(s):
    outputs['__saved_model_init_op'] tensor_info:
        dtype: DT_INVALID
        shape: unknown_rank
        name: NoOp
  Method name is:

Defined Functions:
  Function Name: '__call__'
    Option #1
      Callable with:
        Argument #1
          inputs: TensorSpec(shape=(None, 224, 224, 3), dtype=tf.float32, name=u'inputs')
        Argument #2
          DType: bool
          Value: False
        Argument #3
          DType: bool
          Value: False
        Argument #4
          batch_norm_momentum: TensorSpec(shape=(), dtype=tf.float32,
name=u'batch_norm_momentum')
...
...
```

Doing inference directly

```
tf.nn.softmax(module([img]))[0]
```

Doing inference directly (Bad Idea)

```
tf.nn.softmax(module([img]))[0]
```

The better approach

```
hub.KerasLayer(MODULE_HANDLE,
               input_shape=IMG_SIZE + (3,))
```

Image Feature Vector

- Remove the final classification layer

The better approach

```
model = tf.keras.Sequential([
        hub.KerasLayer(MODULE_HANDLE,
                       input_shape=IMG_SIZE + (3,),
                       output_shape=[FV_size],
                       trainable=True),
        tf.keras.layers.Dense(NUM_CLASSES,
                              activation='softmax')])
```

The API

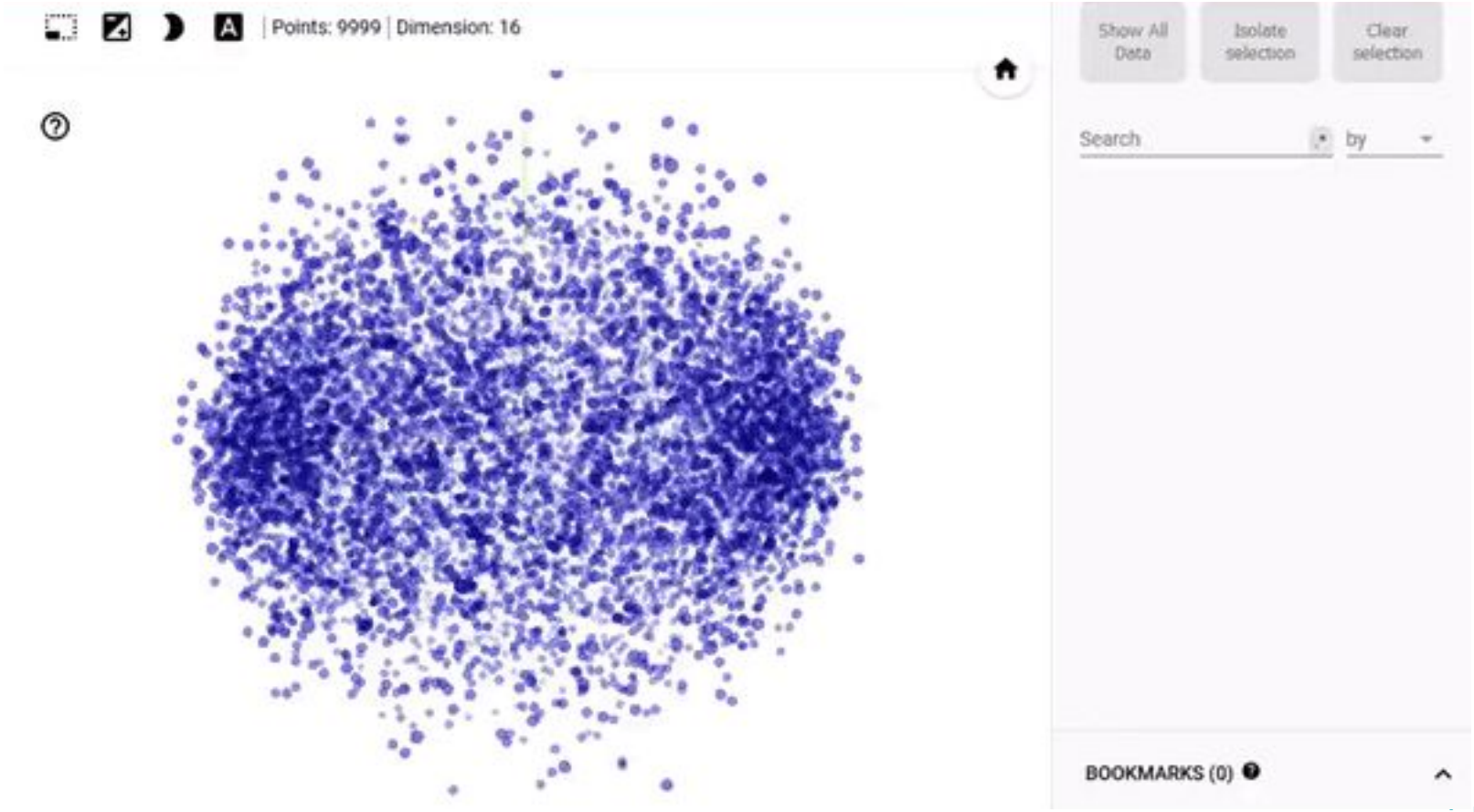
- Closely knit
- High level of integration

Text based models

Is text a problem?

Word embeddings

GDG NASHIK



Embeddings

- G News
- NNLM

```
embedding = "URL"

hub_layer = hub.KerasLayer(embedding,
                           input_shape=[],
                           dtype=tf.string,
                           trainable=True)
```

hub_layer(["Hello"])

```
<tf.Tensor: shape=(1, 20), dtype=float32, numpy=
array([[-0.52853537, -0.44937956,  0.9841924 , -0.1612744 ,
-1.2990013 , 1.0351968 , -1.0662416 ,  0.4302414 ,  1.1699632 ,
-0.3067501 , -1.6935859 ,  0.06937061,  0.36578923, -0.1971624 ,
0.4467874 , -0.857386  , -0.91542685,  0.6074831 ,  0.12608932,
-0.8414185 ]],
      dtype=float32)>
```

```
hub_layer(["This is a sample text",
           "Sample text2"])
```

```
<tf.Tensor: shape=(2, 20), dtype=float32, numpy=
array([[ 0.69290376, -0.55105245,  1.2560561 ,  1.2215823 ,
-1.3314309 , -0.3568075 , -0.14480753, -0.18133132,  0.65806144,
-1.551018  , -0.25053835, 0.5867856 , -0.4025703 , -0.3071912 ,
0.80824476, -0.7694559 , 0.36148477, -0.19313279,  0.2815062 ,
-0.55362564],
[-0.5478371 , 0.8583988 ,  0.07264819,  0.6364152 , -0.87926215,
0.7010501 , -0.05082453, -0.57450294,  0.42839637, -1.0226635 ,
0.3715258 , -1.1585846 ,  0.43829283,  0.29353622,  0.7442786 ,
-0.9790836 , -1.0114487 ,  0.3917291 ,  0.8044613 ,  0.5342001
]], dtype=float32)>
```

Demos!!

gdg-nashik.rishit.tech

Demos

Text Classification

Neural style Transfer

Quick Recap

- Motivation
- TF Hub
- Transfer Learning
- Seeing the process in code
- Feature Vectors and embeddings
- Demos

Q & A

TensorFlow

GDG NASHIK

THANK YOU