Lesson 6. Transfer Learing

6.1. tensorflow_hub

- **Import**

```
import tensorflow_hub as hub
```

Get the model

- You can get a pretrained model from TF Hub Doc
- The magic spell is `hub.KerasLayer(URL, input_shape)`

```
URL = "https://tfhub.dev/google/tf2-preview/mobilenet_v2
feature_extractor = hub.KerasLayer(
                            URL,
                            input_shape=(224, 224, 3))
```

- **Freeze the parameters**

```
feature_extractor.trainable = False
```

- **Usage**

```
model = tf.keras.Sequential([
  feature_extractor,
  layers.Dense(num_classes)
])

model.summary()
```

6.2. Honey tips

1. split a tfds

- When the dataset consists of only training set.

- If split the dataset, tfds **returns a tuple** with meta info. There are two ways.

```
# return a tuple
# first method
(training_set, validation_set), meta = tfds.load(
    'tf_flowers',
    split=["train[:80%]","train[80%:]"],
    with_info=True,
    as_supervised=True
)
```

2. Inside tfds meta data

- **Get number of classes**

```
num_classes = meta.features['label'].num_classess
```

- **Get label names**

```
meta.features['label'].names
```

3. Get Real data for training and validation from tfds

- `.prefetch()` : Most dataset input pipelines should end with a call to prefetch. This allows later elements to be prepared while the current element is being processed. This often improves latency and throughput, at the cost of using additional memory to store prefetched elements.
- `.shuffle()` : more information

```
train_batches  =
   train_examples.cache().shuffle(num_examples//4)
                 .map(format_image).batch(BATCH_SIZE)
                 .prefetch(1)
```

4. Resize image

- Resize

```
image = tf.image.resize(image, (224, 224))
```