TensorFlow

Курс “Практическое применение по TensorFlow”
Шигапова Фирюза Зинатуллаевна
1-й семестр, 2019 г.

https://github.com/Firyuza/TensorFlowPractice

Quiz. tf.data API

1.
   - dataset**.map(map_fn, num_parallel_calls=4).batch(**batch_size=4**)**
   - dataset**.batch(**batch_size=4**).map(map_fn, num_parallel_calls=4)**

Do these two pipelines return the same batches?

2. **What should be first map or batch when map_fn is cheap function?**

TensorFlow 2.0. Network architecture

- Conv2D (5 x 5, 36)
- MaxPool2D (2 x 2, 2)
- Conv2D (5 x 5, 64)
- MaxPool2D (2 x 2, 2)
- Flatten
- Dense

TensorFlow 2.0. Loss function and accuracy

- tf.losses.**SparseCategoricalCrossentropy**

- tf.metrics.**SparseCategoricalAccuracy**

TensorFlow 2.0. build Model

```
def build(self, inputs_shape):
    self.seq_model.add(tf.keras.layers.Conv2D(32, [5, 5],
                                              activation=tf.nn.relu, name='conv1'))
    self.seq_model.add(tf.keras.layers.MaxPool2D([2, 2], 2,
                                                 name='pool1'))
    self.seq_model.add(tf.keras.layers.Conv2D(64, [5, 5],
                                              activation=tf.nn.relu, name='conv2'))
    self.seq_model.add(tf.keras.layers.MaxPool2D([2, 2], 2,
                                                 name='pool2'))
    self.seq_model.add(tf.keras.layers.Flatten())
    self.seq_model.add(tf.keras.layers.Dense(self.nrof_classes,
                                             activation=None, name='fc1'))

    super(Network, self).build(inputs_shape)

    self.built = True

    return
```

Recall: Custom Accuracy

1. **Inherit** from **tf.keras.metrics.Metric**
2. In constructor add variable for tracking custom accuracy value

```
class BinaryTruePositives(tf.keras.metrics.Metric):
  def __init__(self, name='binary_true_positives', **kwargs):
    super(BinaryTruePositives, self).__init__(name=name, **kwargs)
    self.true_positives = self.add_weight(name='tp', initializer='zeros')
```

3. Update accuracy value using **update_state** method

```
def update_state(self, y_true, y_pred, sample_weight=None):
  y_true = tf.cast(y_true, tf.bool)
  y_pred = tf.cast(y_pred, tf.bool)

  values = tf.logical_and(tf.equal(y_true, True), tf.equal(y_pred, True))
  values = tf.cast(values, self.dtype)

  self.true_positives.assign_add(tf.reduce_sum(values))
```

Recall: Custom Accuracy

4. Override result method for getting current accuracy value

```
def result(self):
  return self.true_positives
```

5. Work in pipeline as with pre-made keras accuracies

Custom accuracy

Implement ***update_state*** method:

```
def update_state(self, true_labels, predicted_labels):
    # TODO
    value = None

    self.nrof_elements.assign_add(len(true_labels))

    return self.categorical_accuracy.assign_add(value)
```

Create in pipeline custom metric object:

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

TensorFlow 2.0. Assignment

- Add validation