■訂正前

```
def predict(self, X_data, return_proba=False):
    preds = []
    with tf.Session(graph = self.g) as sess:
        self.saver.restore(sess, tf.train.latest_checkpoint('./model/'))
    test_state = sess.run(self.initial_state)
    for ii, batch_x in enumerate(create_batch_generator(
            X_data, None, batch_size=self.batch_size), 1):
        feed = {'tf_x:0': batch_x, 'tf_keepprob:0': 1.0,
                self.initial_state: test_state}
        if return_proba:
            pred, test_state = sess.run(
                ['probabilities:0', self.final_state],
                feed_dict=feed)
        else:
            pred, test_state = sess.run(
                ['labels:0', self.final_state],
                feed_dict=feed)
        preds.append(pred)
    return np.concatenate(preds)
```

■訂正後

```
def predict(self, X_data, return_proba=False):
    preds = []
    with tf.Session(graph = self.g) as sess:
        self.saver.restore(sess, tf.train.latest_checkpoint('./model/'))
        test_state = sess.run(self.initial_state)
        for ii, batch_x in enumerate(create_batch_generator(
                X_data, None, batch_size=self.batch_size), 1):
            feed = {'tf_x:0': batch_x, 'tf_keepprob:0': 1.0,
                    self.initial_state: test_state}
            if return_proba:
                pred, test_state = sess.run(
                    ['probabilities:0', self.final_state],
                    feed_dict=feed)
            else:
                pred, test_state = sess.run(
                    ['labels:0', self.final_state],
                    feed_dict=feed)
            preds.append(pred)
        return np.concatenate(preds)
```