**參考資料**

**TensorFlow筆記-06-神經網絡優化-​​損失函數，自定義損失函數，交叉熵**

[**https://blog.csdn.net/qq\_40147863/article/details/82015360**](https://blog.csdn.net/qq_40147863/article/details/82015360)

**使用損失函數**

[**https://keras.io/losses/**](https://keras.io/losses/)

**延伸閱讀:自訂損失函數 1/2**

* **對於預測優酪乳日銷量問題，如果預測銷量大於實際銷量則會損失成本；如果預測銷量小於實際銷量則會損失利潤。**
* **在實際生活中，往往製造一盒優酪乳的成本和銷售一盒優酪乳的利潤不是等價的。因此，需要使用符合該問題的自訂損失函數**
* **自訂損失函數為：**
  + **loss = Σnf(y\_, y)**
  + **其中，損失函數成分段函數：損失函數表示若預測結果y小於標準答案y\_，損失函數為利潤乘以預測結果y與標準答案之差若預測結果y大於標準答案y\_，**
* **損失函數為成本乘以預測結果y與標準答案之差用Tensorflow函數表示為：**
  + **loss = tf.reduce\_sum(tf.where(tf.greater(y, y\_), COST\*(y-y\_), PROFIT\*(y\_-y)))**
* **第1種情況：若優酪乳成本為1元，優酪乳銷售利潤為9元，則製造成本小於優酪乳利潤，因此希望預測結果y多一些**
  + **# 定義損失函數使得預測少了的損失大，於是模型應該偏向多的放心預測**
    - **loss = tf.reduce\_sum(tf.where(tf.greater(y, y\_), COST\*(y-y\_), PROFIT\*(y\_-y)))**
    - **train\_step = tf.train.GradientDescentOptimizer(0.001).minimize(loss)**
* **第2種情況：若優酪乳成本為9元，優酪乳銷售利潤為1元，則製造利潤小於優酪乳成本，因此希望預# 重新定義損失函數使得預測多了的損失大，於是模型應該偏向少的方向預測**
  + **loss = tf.reduce\_sum(tf.where(tf.greater(y, y\_), COST\*(y-y\_), PROFIT\*(y\_-y)))**
  + **train\_step = tf.train.GradientDescentOptimizer(0.001).minimize(loss)**

**# 其他優化方法**

**# train\_step = tf.train.GMomentumOptimizer(0.001, 0.9).minimize(loss)**

**# train\_step = tf.train.AdamOptimizer(0.001).minimize(loss)**