

Week2 - Parameters Tuning and DL Frameworks

Sonntag, 19. Juli 2020 10:11

Parameters Tuning Process

Choose random uniformed values to tune parameters

```
r = -4 * n.random.randn()  
Learning_rate = 10r
```

This method is useful when the two ends of range are on different scales.

Batch Normalization

Speed up learning process by normalize the 'input' of each layer -> normalize $z()$

for certain layer l , $z^{(1)}, z^{(2)}, \dots, z^{(m)}$

$$\mu = \frac{1}{m} \sum_{i=1}^m z^{(i)} \quad \sigma^2 = \frac{1}{m} \sum_{i=1}^m (z^{(i)} - \mu)^2$$

$$z_{norm}^{(i)} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \epsilon}}$$

$$\tilde{z}^{(i)} = \gamma z_{norm}^{(i)} + \beta$$

learned, which control the mean and variance
so that mean / variance always $[0, 1]$

Batch norm has also a slight regularization effect.

Batch norm at test time: using exponential weighted average through all mini-batches to estimate μ and σ^2 for test time

Multiple classification

Softmax layer as the last layer

activation: $t = e^{z^{(L)}}$

$$a^{(L)} = \frac{e^{z^{(L)}}}{\sum_{i=1}^n t_i} \quad (4.4) \quad , \quad \hat{a}_i^{(L)} = \frac{t_i}{\sum t_i}$$

$$a^{L-1} = \frac{0}{\sum_{i=1}^n t_i} \quad (4.4) , \quad a_i = \frac{1}{\sum t_i}$$

Take away from programming assignment:

TensorFlow 1: general idea of running a TensorFlow program:

- Create Tensors (variables) that are not yet executed/evaluated.
- Write operations between those Tensors.
- Initialize your Tensors.
- Create a Session.
- Run the Session. This will run the operations you'd written above.

Note that, a-d have only created the calculation graph, step e runs it.

A placeholder is an object whose value you can specify only later. To specify values for a placeholder, you can pass in values by using a "feed dictionary" (feed_dict variable).

for multi-class:

LINEAR -> RELU -> LINEAR -> RELU -> LINEAR -> SOFTMAX. The SIGMOID output layer has been converted to a SOFTMAX. A SOFTMAX layer generalizes SIGMOID to when there are more than two classes. for the `tf.reduce_mean()` to calculate the cost: It is important to know that the "logits" and "labels" inputs of `tf.nn.softmax_cross_entropy_with_logits` are expected to be of shape (number of examples, num_classes).