**Experimentation report: parameter optimization**

Wenqian Cui, 2021/08/04

**Overview**

This is the report for the parameter optimization of the GCN and GAT model. For a deep learning model, a better way to do the parameter optimization is to randomly sample the value for these parameters. In my experiments, I used two sample methods in total, which called “uniform sample” and “exponential sample”.

**Uniform sample:** Sample the value uniformly within a scale.

**Exponential sample:** This is a special case when doing the parameter optimization. The reason is that some parameters need to be paid attention to different scales of values. For example, I believe the value search for the learning rate within (0.0001, 0.001) and within (0.001, 0.01) is equally important. Therefore, here’s what I do: I want to search the optimal value for learning rate between 0.0001 and 0.01, so I first perform the uniform sample **“n”** between **-4** and **-2**, and then use the sampled value to calculate **10n** to derive the sample value for learning rate.

The optimized parameters include the **learning rate**, **hidden layer size**, **dropout rate**, **weight decay rate**.

**Learning rate:** Through my careful investigation, the optimal learning rate should be a value between 0.0001 and 0.01. (exponential sample)

**Hidden layer size:** Between 10 and 500. (uniform sample)

**Dropout rate:** A value sample from [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9].

**Weight decay rate:** Between 0.0001 and 0.01. (exponential sample)

**Experiments & Results**

Best result: (GCN)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accuracy | Learning rate | Hidden layer size | Dropout rate | Weight decay rate |
| 0.835 | 0.009698537 | 249 | 0.9 | 0.000103214 |

Best result: (GAT)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Accuracy | Learning rate | Hidden layer size | Dropout rate | Weight decay rate |
| 0.851 | 0.004084 | 189 | 0.8 | 0.000936 |

**Analysis & Questions**

The results seem ok from here, but I found that the accuracies under these parameters are unstable. For all the results, when I perform the repeated experiments, the accuracy varies for approximately 1%. Therefore, the result can become unconvincing if we want to show that the data augmentation method can improve the performance of the models, especially for the cases when the performance improvement are within 1%.

No matter how I adjust the parameters, the problem stays the same. I also tried to force the parameter initialization to use “xaiver” method, but the accuracy drops after doing that. Do you have any better ideas?