

人工神经网络 HW4 实验报告

计 52 路橙 2015010137

Part 1：综述

本次实验中，我用 TensorFlow 实现了基本的 RNNCell，其中包含 BasicRNNCell、GRUCell 与 BasicLSTMCell。并通过初步的调参，比较了三者之间的区别，得到了一个较优性能的结果模型。

最终，使用 tensorboard 绘制了训练过程中的变化。

Part 2：不同 RNNCell 的训练过程

选择相同超参数的三种 RNNCell 进行训练，共计训练 30 个 epoch，结果如下。

参数选择：

- Learning Rate:0.001
- Units:512
- Embed_units:300
- Layers:1
- Batch_size:16

其余超参数为网络默认参数。

① Optimizer 为 GradientDescentOptimizer:



Figure 1. 每个 epoch 网络在 dev 集与 test 集的准确率

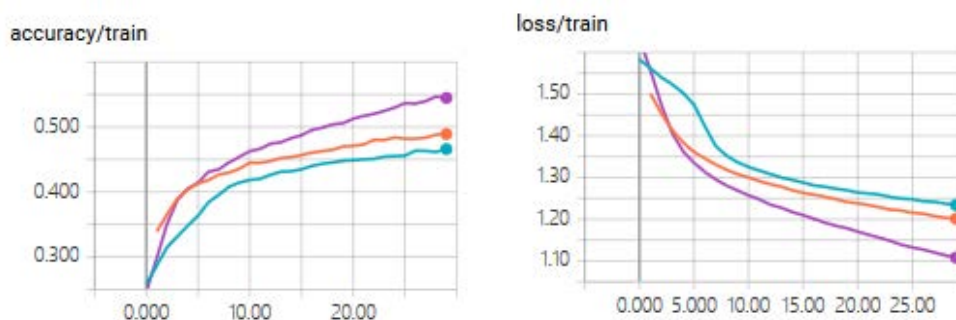


Figure 2. 每个 epoch 网络在训练集上的 accuracy 与 loss

由上图可知，当 `Optimizer` 为 `GradientDescentOptimizer` 时，三种网络有如下特点：

1. `BasicRNNCell` 具有收敛快、性能较好的优点，甚至在 `dev` 集上的准确率达到了 0.44 以上，超过了其余两个网络。这也说明了基础的 RNN 就可以做到较高的水平，RNN 在与序列有关的问题中可以有较优的表现；

但另一方面，RNN 的过拟合问题比较严重，当在 `test` 集上的准确率已处于徘徊波动状态时，在 `train` 集上依然可以观察到 `loss` 在稳步快速下降，`accuracy` 在快速上升，说明 `BasicRNNCell` 的过拟合问题是三种网络里最明显的。

2. `GRUCell` 的收敛速度最慢，但性能较优，几乎不存在过拟合问题。一方面，由于 `GRUCell` 比 `BasicRNNCell` 的参数多很多，因而收敛速度慢有一定的道理；而另一方面，在 `test` 集最终 `accuracy` 的曲线上看，`GRUCell` 依然可以有继续爬升的空间，说明 `GRUCell` 的收敛速度慢，但最终可以达到一个较高的准确率水平。

3. `BasicLSTMCell` 收敛速度较快，且过拟合问题不严重，最终性能可以达到很高水平。这也从另一个角度说明，LSTM 作为 RNN 的改进版本，拥有很强的处理序列问题的能力。

一方面，在 `test` 集与 `dev` 集上，`BasicLSTMCell` 都可以做到收敛速度很快且最终效果较好；另一方面，在 `train` 集上，`BasicLSTMCell` 的过拟合程度较轻，`loss` 下降也先快后慢最终趋于平缓。这说明，`BasicLSTMCell` 集成了 `BasicRNNCell` 与 `GRUCell` 的优点，同时一定程度上规避了两者的缺陷，是处理序列问题中选用 RNN 模型时的最佳 `Cell` 选择。

② Optimizer 为 AdamOptimizer:



Figure 3. 每个 epoch 网络在 dev 集与 test 集的准确率

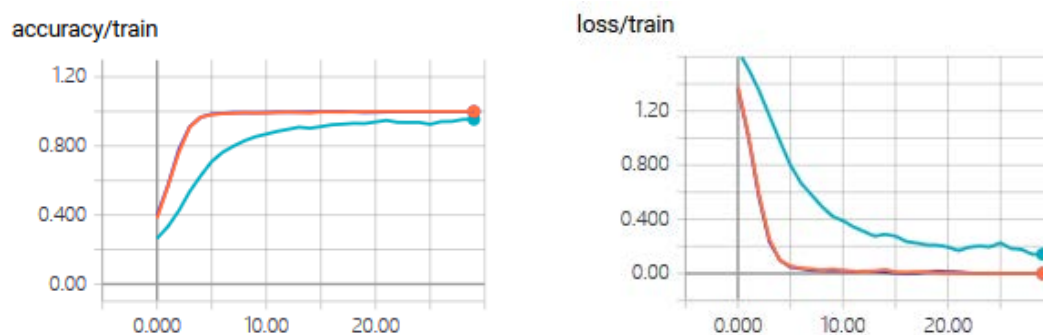


Figure 4. 每个 epoch 网络在训练集上的 accuracy 与 loss

由上图可知，当 Optimizer 为 AdamOptimizer 时，三种网络都会有严重的过拟合问题（甚至最终可以在 train 集上达到 1.0 的准确率，但在 test 集上的准确率较低）。

此外，三种网络还有如下特点：

1. BasicRNNCell 性能欠佳，在 test 集上的准确率处于徘徊状态，难以达到其最优性能，且其在 train 集上的 loss 下降也较慢。由于很快产生了过拟合，因此在 test 集上的性能表现无法再提高，从一开始便决定了性能较差。
2. GRUCell 在 test 集与 dev 集中都有很高的表现，甚至在 test 集上可以有超过 0.46 的准确率。

一方面，AdamOptimizer 可以使网络在最初阶段达到很高的 test 集准确率，从而保存当前网络后便会得到一个极优的性能；另一方面，由于 AdamOptimizer 的收敛速度较快，使得网络在达到较优结果后会超着向 train 集收敛的方向迈进，从而

使得网络出现严重的过拟合，在 `test` 集与 `dev` 集上的表现也相应下降，最终稳定的结果并不如 `GradientDescentOptimizer`。

3. `BasicLSTMCell` 也有与 `GRUCell` 十分类似的特征，即最初准确率极高，然而随着训练的进行网络会向 `train` 集过拟合，导致最终稳定的结果较差。且 `BasicLSTMCell` 的结果比 `GRUCell` 的结果差一些。

③ 总结

通过①与②的比较可知，在 `RNNCell` 的训练过程中，若为了追求最高的测试集准确率，则应该选用 `AdamOptimizer` 并训练 10 个以内的 `epoch` 即停止；而若为了使训练稳健增长且尽量避免过拟合问题，则应该选用 `GradientDescentOptimizer`。

Part 3：BasicLSTMCell 的简单调参

① Dropout 的选择：

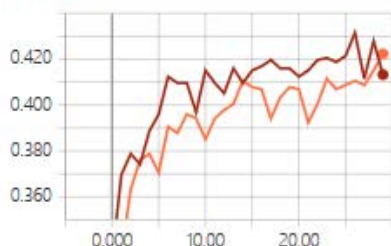
通过对 LSTM 网络增加 `DropoutWrapper` 来减少过拟合。所用的 `Optimizer` 为 `GradientDescentOptimizer`。分别进行 `keep_prob` 为 1.0 与 0.5 的对照实验，结果如下图所示。

参数选择：

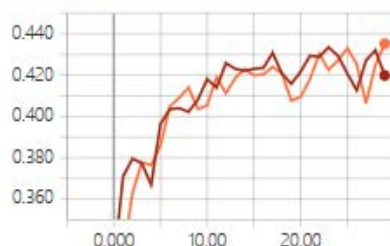
- **Learning Rate:0.001**
 - **Units:512**
 - **Embed_units:300**
 - **Layers:1**
 - **Batch_size:16**
- 其余超参数为网络默认参数。

■ LSTM_1.0_dropout
■ LSTM_0.5_dropout

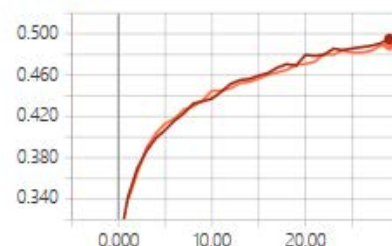
accuracy/dev



accuracy/test



accuracy/train



由上图可知，增加 dropout 后的 LSTM 网络具有更快的收敛速度、更高的测试集准确率。因此，适当增加 dropout 的确会使得 LSTM 网络减小过拟合，使得网络有更稳定、更高效的性能。

② Units_size 的选择（讨论在 AdamOptimizer 下过拟合的主要因素）：

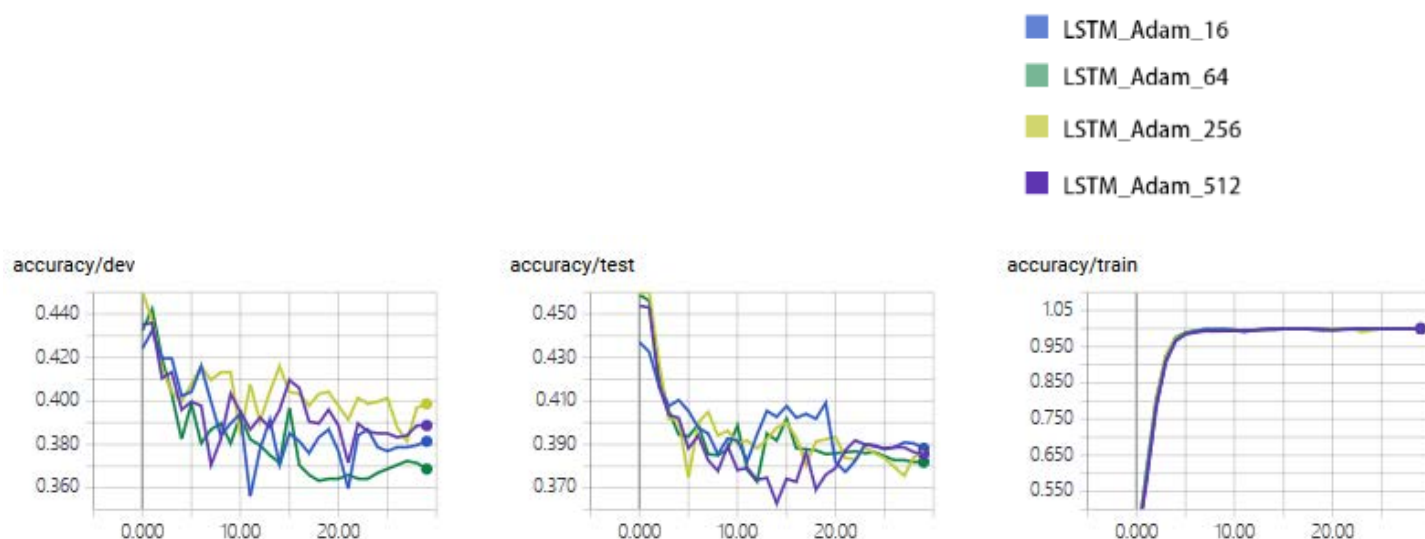
通过对 LSTM 网络内部 gate 的 units_size 进行调整，分别选取 size 为 512、256、64、16，结果如下图所示。

参数选择：

- Learning Rate:0.001
- Embed_units:300
- Layers:1
- Batch_size:16

其余超参数为网络默认参数。

选用的 Optimizer 为 AdamOptimizer。



由上图可知，从宏观而言，units_size 即使为 16 时也可以得到较好的效果，说明在 Optimizer 为 AdamOptimizer 时，网络超参数的数量已经不是过拟合的主要因素，而恰恰是 Optimizer 的收敛速度导致过拟合的出现。

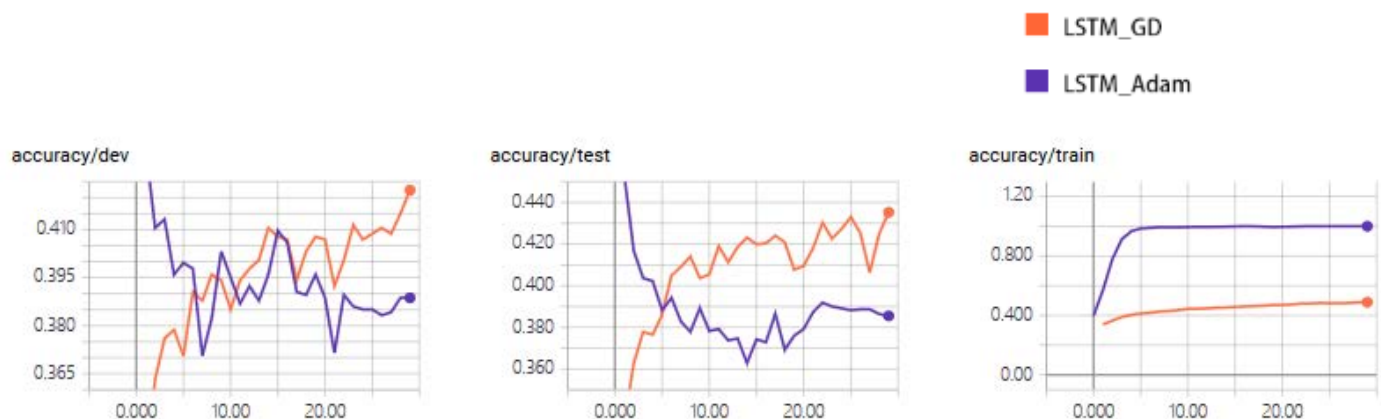
而当该参数为 256 时，网络的效果最稳定，甚至可以出现 0.46 以上的超高测试集准确率。

③ Optimizer 的选择：

分别选取 AdamOptimizer 与 GradientDescentOptimizer，结果如下图所示。

参数选择：

- **Learning Rate:0.001**
 - **Units:512**
 - **Embed_units:300**
 - **Layers:1**
 - **Batch_size:16**
- 其余超参数为网络默认参数。



由上图可知，Adam 可以让网络在最初时拥有超过的测试集准确率，但随着过拟合的出现而逐渐下降；但 GD 可以使网络缓慢稳步上升，最终的性能甚至可以达到 Adam 的性能，且过拟合的程度小很多。因此，可以得出如下结论：

在 LSTM 的训练过程中，若为了追求最高的测试集准确率，则应该选用 AdamOptimizer 并训练 10 个以内的 epoch 即停止，并选用保存的最佳性能的网络；而若为了使训练稳健增长且尽量避免过拟合问题，则应该选用 GradientDescentOptimizer。

Part 4：最优网络

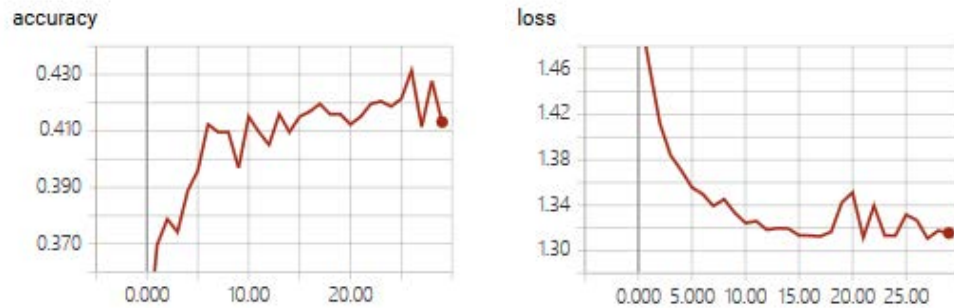
通过 Part 3 的讨论，我最终选取的最优网络有两种，分别对应两种 Optimizer。

① 最优准确率最高：



选用 `BasicLSTMCell`，且 `Optimizer` 为 `AdamOptimizer`。由 Part 3 的讨论知，此时 `units_size` 选择 256 最佳；而最终网络可以有超过 0.46 的测试集准确率。

② 最终准确率最高，且尽可能减少过拟合：



选用 `BasicLSTMCell`，且 `Optimizer` 为 `GradientDescentOptimizer`。由 Part 3 的讨论知，此时需要添加 0.5 的 `keep_prob` 的 `dropout`；而最终网络可以有超过 0.43 的测试集准确率。