Moritz Eck
05.11.2018

Exercise 03: Evaluating different Optimizers in a Recurrent Neural Network

I have chosen to evaluate: **Adam**, **RMSProp** and **Gradient Descent**. The models were trained on the full dataset for min. four epochs (average 18 min per training).

Initial Evaluation

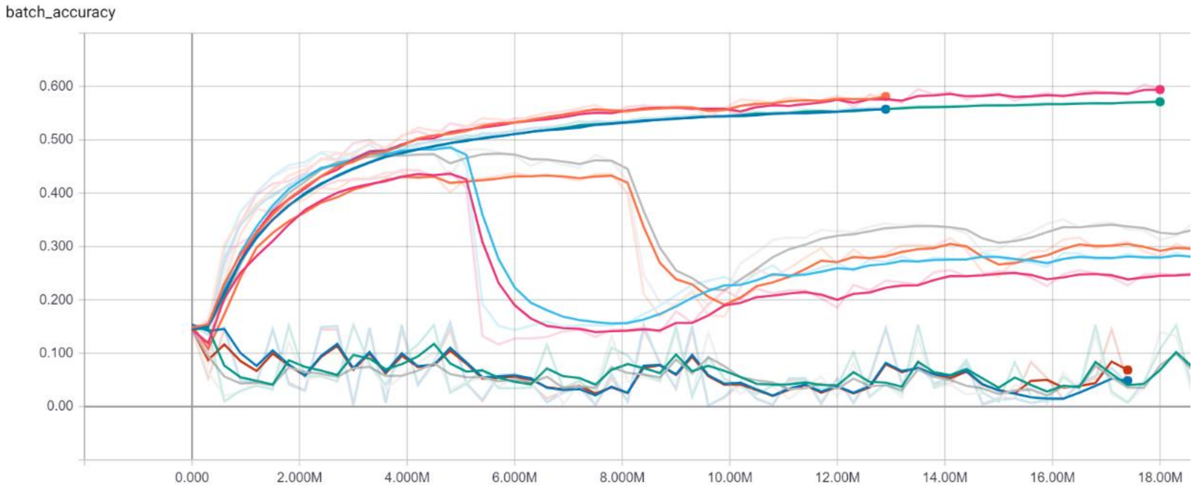

Figure 1: Adam, RMSProp, Gradient Descent (GD) Evaluation

In Figure 1, **Adam** (top-4-curves) performs the best and most consistent. Initially, the performance of Adam and **RMSProp** (middle-4-curves) are approximately the same (until 1.5 epochs). Afterwards, **Adam** continues to improve approximating 0.6 for both training and validation accuracy.

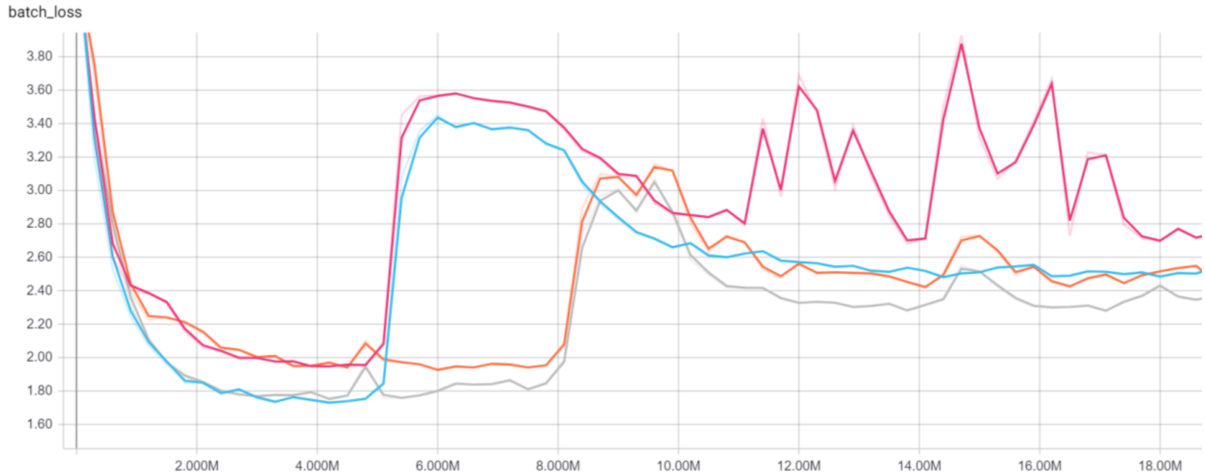

Figure 2: Increase in validation and training loss in RMSProp

The **RMSProp's accuracy** drops down from approx. 0.42/0.47 to below 0.2. The decrease in accuracy is due to a spike in training/validation loss (Figure 2). One iteration decreases the loss again afterwards. In the other one, the loss oscillates back and forth. **RMSProp** can train the model to a certain extent before starting to overfit (see below the single-layer model performs best).

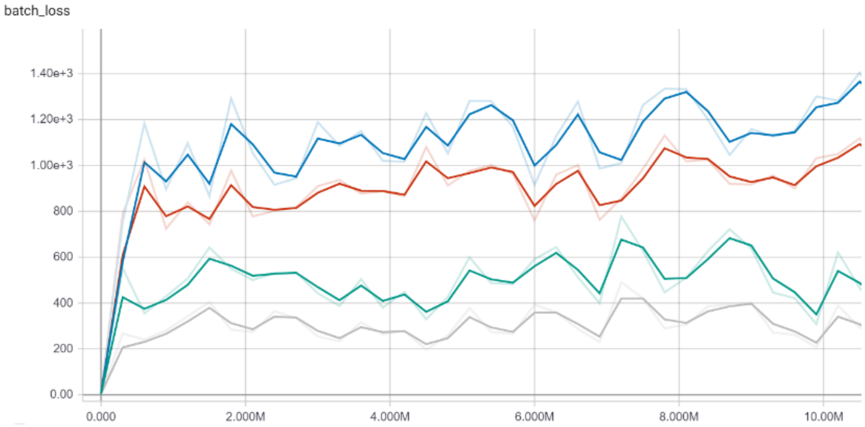

Figure 3: Gradient Descent Loss

Gradient Descent performs the worst and is unable to training the model (bottom-4-curves in Figure 1). The accuracy oscillates between 0.05 and 0.15. The loss increases from the beginning, never decreases, only oscillates (Figure 3). Even tuning the learning rate made no difference. GD is unable to train the model => most likely due to nature of GD and not using: ***learning rate decay*** (RMSProp, Adam) and ***momentum*** (Adam). Using both delivers the best results.

Layer Size Evaluation

I have evaluated one, three, and five recurrent layers. Training with one recurrent layer is approx. 5x faster than with three-layers (five-layers is 2x slower than three-layers).

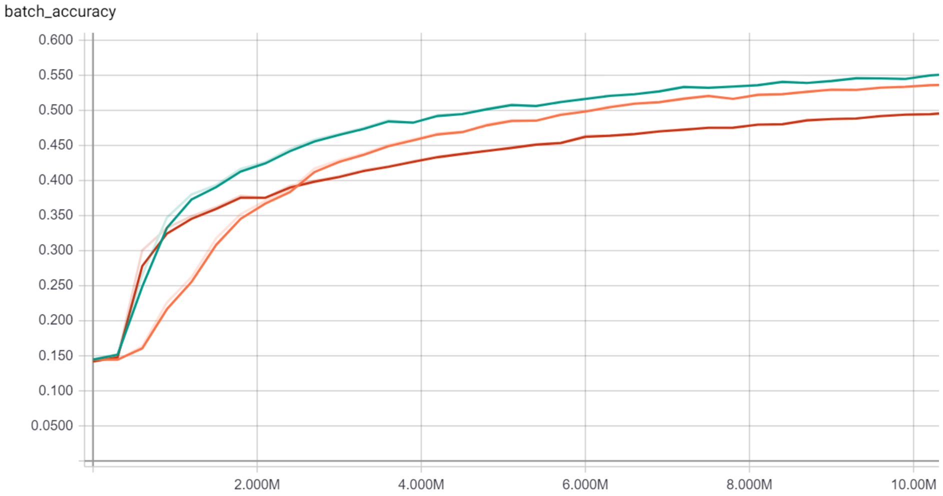

Figure 4: Adam results with different layer sizes

In Figure 4, the performance of the **Adam optimizers** is approx. the same for all layer sizes. The model lags at the start with more layers (orange-curve=five-layers, green-curve=three-layers, red-curve=single-layer) and is slower to improve. The single-layer model learns the fastest from the start but the three/five-layer models overtake after 0.3/1.3-epochs since it is most likely not powerful enough to learn more details.

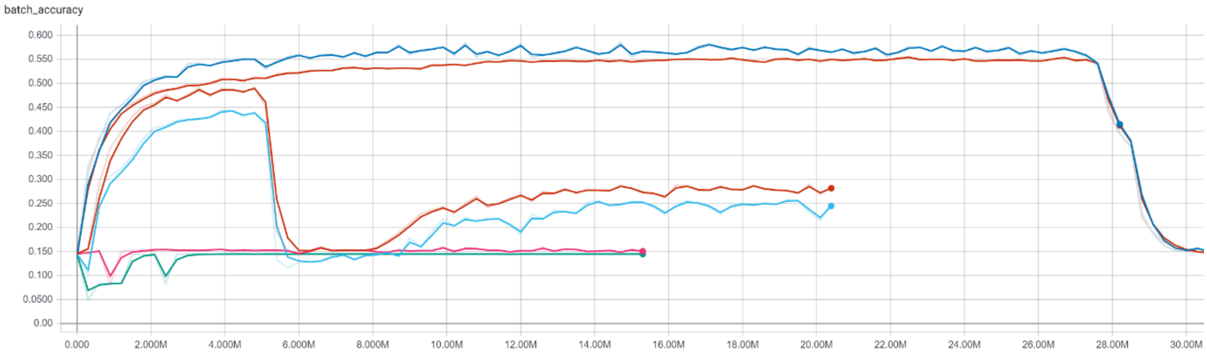

Figure 5: RMSProp results with different layers

In Figure 5, the performance of the **RMSProp** with a single-layer model (top-2-curves) outperforms the three-layer models (middle-2-curves) and the five-layer model (bottom-2-curves). **Assumption:** The optimizer is unable to train the model => underfitting to the data. Additionally, the model with more layers experiences the drop in accuracy earlier (=> overfitting? since every model's accuracy decreases after some epochs).

Gradient Descent remained unable to train the model independent from the number of layers.

Dropout Evaluation

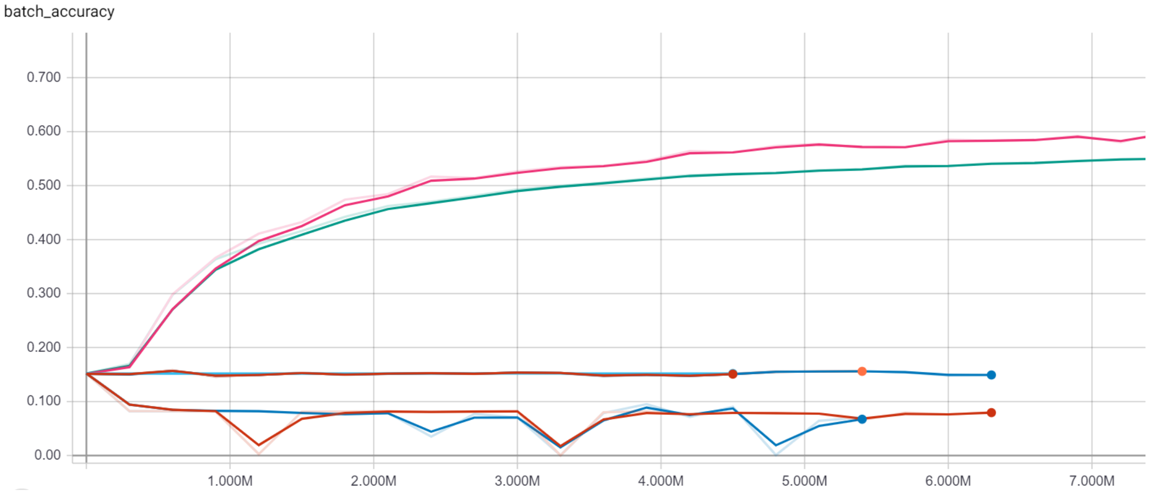

Figure 6: Dropout Evaluation (Adam & RMSProp)

dropout_pkeep = 0.1 (very strong regularization): Both **Adam** and **RMSProp** are unable to train the model, since the regularization is too strong (middle-4-curves).

dropout_pkeep = 1.0 (no dropout regularization): The training and validation accuracy of **Adam** diverge the longer the training continues but not very strongly (top-2-curves). For the **RMSProp** the results are confusing since both training/validation accuracy are worse without regularization than when using almost maximal regularization (bottom-2-curves).