Model Pruning: Weekly Report 6

Patricia Gschoßmann

1. Weekly Progress

In this week I continued working on the unfinished classification task from last week. Last week's experiments maintained a stable performance until $\alpha = 0.015$ (i.e. a validation accuracy of $\approx 88\%$). The first larger drop happend at $\alpha = 0.0125$, which reached a validation accuracy of 84% with a learning rate of 0.0001. Last week's main observation was, that local minima can be exceeded by resuming the training and increasing the initial learning rate. Based on this, two experiments were executed, starting from $\alpha = 0.0125$:

1. Resume training while iteratively increasing the learning rate. Reduce α as soon as the local minima are surpassed.
2. Same as above, but additionally enable weight updates for the pretrained model.

I additionally started another experiment, where weight updates for the pretrained model are enabled from the beginning. A final validation accuracy of 88% was achieved, moreover, it was sufficient to decay α by 0.1 at each iteration to reach this performance.

2. Results

2.1. Without weight updates

With a learning rate of 0.0001 the model was able to reach a validation accuracy of $\approx 84\%$ at $\alpha = 0.0125$. To overcome this local minima, the initial learning rate was increased - once with 0.001 and once with 0.01. How However, both approached worsened the performance (see fig. 1).
Based on these results, I continued reducing α to 0.01. However, at this step, the model failed to achieve a higher validation accuracy than 70%, even with the learning rate approach - increasing the learning rate worsened the results as before.
I decided to reduce α even further to 0.0, based on the fact, that in last week's experiments a validation accuracy of 85% was reached for this α. An initial learning rate of 0.001 lead to a validation accuracy of 48%; resuming training for this result with a learning rate equal to 0.01 increased it to 75%. Currently a further iteration with 0.1 is running, to overcome the 75% accuracy bound (see fig. 2).

(a) Training accuracy

(b) Validation accuracy

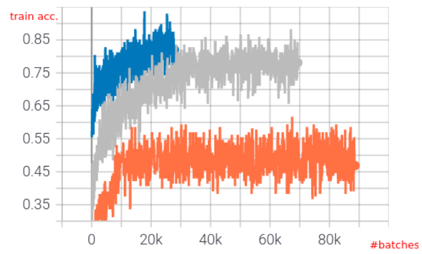
Figure 1: Train- and validation accuracy for $\alpha = 0.0125$, based on the previously trained model with $\alpha = 0.0125$ (green). Pink: $\alpha = 0.0125$, $lr = 0.001$, based on green. Orange: $\alpha = 0.0125$, $lr = 0.01$, based on green.

2.2. With weight updates

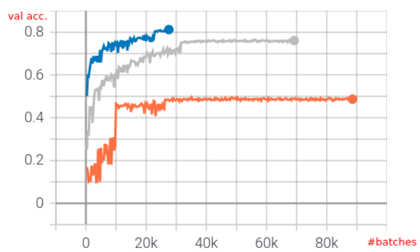
For the following experiments, weight updates for the pretained model were enabled.

2.2.1 Starting from $\alpha = 0.0125$

I started this experiment by resuming the training for the model with $\alpha = 0.0125$, without changing the learning rate ($= 0.0001$). Enabeling weight updates only improved the performance significantly - the validation accuracy increased from 84% to $91{,}5\%$. I continued reducing α in 0.00025 steps, without changing the learning rate. Until $\alpha = 0.0075$ an accuracy of $\approx 91\%$ could be maintained. At $\alpha = 0.005$ it decreased by 2%. The first significant drop happend at $\alpha = 0.0025$, where the validation accuracy was



(a) Training accuracy



(b) Validation accuracy

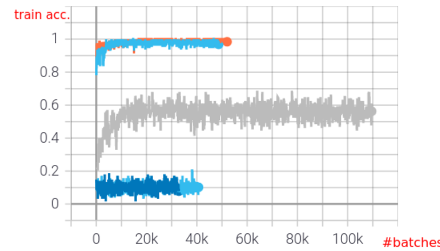
Figure 2: Train- and validation accuracy for $\alpha = 0.0$, based on the previously trained model with $\alpha = 0.0125$ (green curve in fig. 1).
Orange: $\alpha = 0.0$, $lr = 0.001$. Gray: $\alpha = 0.0$, $lr = 0.01$, based on orange. Blue: $\alpha = 0.0$, $lr = 0.1$, based on gray.

only 55%. To overcome this local minima, the initial learning rate was increased - once with 0.001 and once with 0.01. However, both times, the model failed to learn anything (see fig. 3).
Based on these results, I decided to set α equal 0.0, since 0.005 is already very small. Learning rates equal to 0.0001 and 0.00005 failed, however further iterations with 0.001 and 0.01 are running.

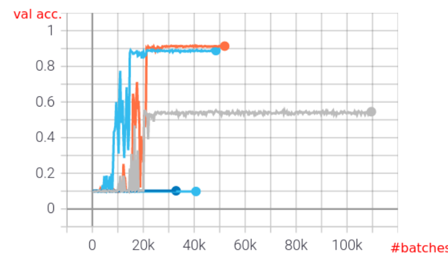
2.2.2 From the beginning

This experiment was executed starting from $\alpha = 1.0$ and with a step-deacy of 0.1. With an initial learning rate of 0.001 for each iteration, the model was able to maintain a validation accuracy of 90% throughout the whole training, until $\alpha = 0.0$, where the accuracy dropped by 6%. Resuming training two times, first with a learning rate of 0.01, second with a learing rate of 0.1, lead to a final validation accuracy of 88% (see fig. 4).

After the final model reached the desired accuracy, its parallel branches were pruned to obtain the smaller version of it. The resulting pruned model was again tested on the test data, to ensure that the pruning was done correctly. It maintained the same accuracy as during the validation.



(a) Training accuracy

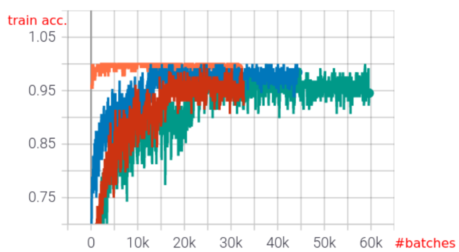


(b) Validation accuracy

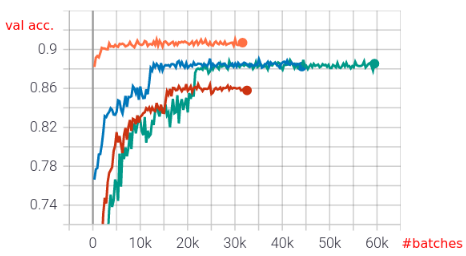
Figure 3: Train- and validation accuracy for different α-values with weight updates of the pretrained model starting from $\alpha = 0.0125$ (green curve in fig. 1).
Orange: $\alpha = 0.0075$. Light blue (high acc.): $\alpha = 0.005$. Gray: $\alpha = 0.0025$, $lr = 0.0001$. Dark blue: $\alpha = 0.0025$, $lr = 0.001$, based on gray. Light blue (low acc.): $\alpha = 0.0025$, $lr = 0.01$, based on gray.

3. Plan

- Try to improve performance of unfinished models (see section 2.1, and 2.2.1).
- Execute additional tests regarding time and average memory consumption for finished models and visualize results to further compare the performances of the original and pruned models.
- Test pruning approach on skip connections.



(a) Training accuracy



(b) Validation accuracy

Figure 4: Train- and validation accuracy for different α-values with weight updates of the pretrained model starting from $\alpha = 1.0$.
Orange: $\alpha = 0.1$, $lr = 0.001$. Red: $\alpha = 0.0$, $lr = 0.001$. Blue: $\alpha = 0.0$, $lr = 0.01$, based on red. Green: $\alpha = 0.0$, $lr = 0.1$, based on blue.