Experiment report

Supervised set-up

Learning rate

To find good learning rate I was running "`lr finder`", namely, running for 200 steps with `lr` growing exponentially from 1e-7 to 1, and then eye-balling the plot and selecting the highest lr where loss was stable and somewhat decreasing.

Schedule

I started with CosineAnnealingLR. Then I noticed that initially training sometimes seems unstable, so added a linear warm-up for 5 epochs.

Number of epochs

I was running for 100 epochs, also tried more (stopped at 250). Was saving checkpoints after every epochs, selecting the best one by validation performance.

Weight averaging

To improve performance on held-out data I tried to version of weight averaging: EMA with span of 10 and Polyak (i.e. just arithmetic average).

Without averaging I got 0.703 score on Kaggle.

With Polyak averaging I got 0.707.

Then I understood that in my implementation batch norm buffers of Polyak-averaged model might be out-of-sync, and so did a couple more short rounds of training starting from that Polyak checkpoint, with very small lr (1e-7), to get less noisy versions of params and up-to-date batch norm buffers. This version got 0.751.

Pretext tasks

Metrics

Apart from pretext task loss I was tracking the classification performance of a linear probe based on the learned representations, and selecting checkpoints based on that. I was also observing confusion matrix, to see what kinds of errors happen due to representation.

Task

I chose rotation prediction, for its implementation simplicity. Model achieved 90%+ accuracy on rotation prediction (choosing among 4 rotations). Classification performance initially grew from 30% after 1st epoch to 50%, but then started declining. Confusion matrix showed that initially model was confusing cats and dogs (probably because predicting rotation for both of them is quite similar), and also sometimes dogs and horses. At peak (when there was 50% classiication performance), the confusion matrix started looking more or less diagonal.

But then I tried doing supervised learning starting from that checkpoint, and it never reached same performance as the networks trained from scratch. (Got something around 70%)

Warm start

To avoid breaking representations, I tried initializing the head using LogisticRegression, before starting the training. It allowed to get 45% after the first epoch, but still didn't get high accuracy in the end.