Fastai Lesson 11a review

Joseph Catanzarite
Fastai Deep Learning From The Foundations
TWiML Study Group Meetup
Saturday, 10/12/2019

Overview

Today:

Start Lesson 11

Notebooks

- 07a_lsuv
- 08_data_block

Next meeting:

Continue in Lesson 11

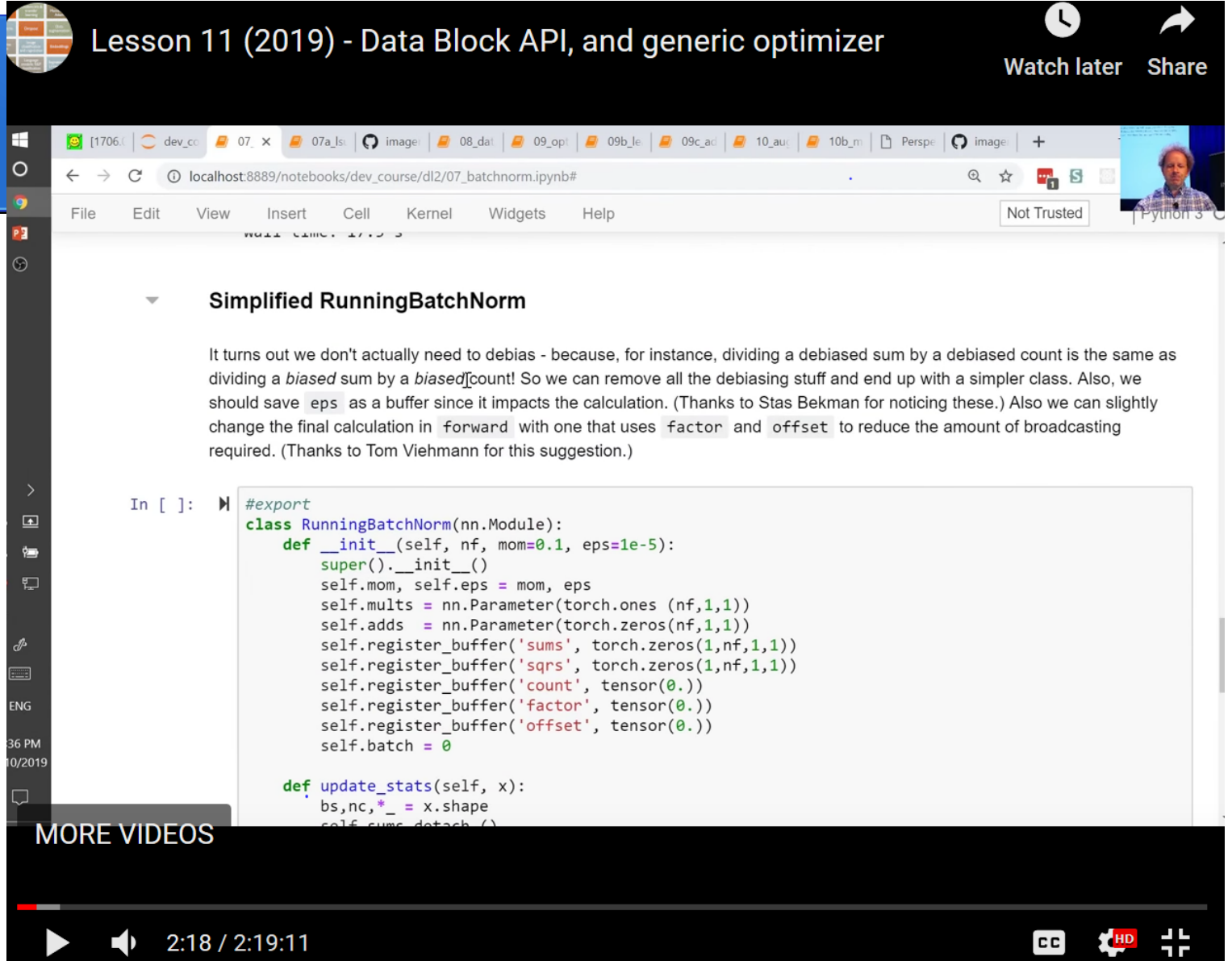
Listen to / watch at least the first half of the Lesson 11 video, through the DataBlocks API part.

Notebooks

- Finish 08_data_block
- Start 09_optimizers

Fastai Forum post: Questions about RunningBatchNorm class in Lesson 10

- https://forums.fast.ai/t/questions-about-runningbatchnorm-in-lesson-10/56331


Lesson 11 (2019) - Data Block API, and generic optimizer
Watch later
Share
localhost:8889/notebooks/dev_course/dl2/07_batchnorm.ipynb#
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Simplified RunningBatchNorm
It turns out we don't actually need to debias - because, for instance, dividing a debiased sum by a debiased count is the same as dividing a biased sum by a biased count! So we can remove all the debiasing stuff and end up with a simpler class. Also, we should save eps as a buffer since it impacts the calculation. (Thanks to Stas Bekman for noticing these.) Also we can slightly change the final calculation in forward with one that uses factor and offset to reduce the amount of broadcasting required. (Thanks to Tom Viehmann for this suggestion.)
In [ ]:
#export
class RunningBatchNorm(nn.Module):
def __init__(self, nf, mom=0.1, eps=1e-5):
super().__init__()
self.mom, self.eps = mom, eps
self.mults = nn.Parameter(torch.ones (nf,1,1))
self.adds = nn.Parameter(torch.zeros(nf,1,1))
self.register_buffer('sums', torch.zeros(1,nf,1,1))
self.register_buffer('sqrs', torch.zeros(1,nf,1,1))
self.register_buffer('count', tensor(0.))
self.register_buffer('factor', tensor(0.))
self.register_buffer('offset', tensor(0.))
self.batch = 0
def update_stats(self, x):
bs,nc,*_ = x.shape
MORE VIDEOS
2:18 / 2:19:11

Fastai Forum post re: Simplified RunningBatchNorm in Lesson 11

- https://forums.fast.ai/t/simplified-runningbatchnorm-in-lesson-11/56340

Fastai Forum post: Question on the Implementation of LSUV

- In the paper “All you need is a good init” the authors say that their Algorithm 1 is intended to follow a certain orthonormal initialization procedure: “We thus extend the orthonormal initialization Saxe et al. (2014) to an iterative procedure, described in Algorithm 1. Saxe et al. (2014) could be implemented in two steps. First, fill the weights with Gaussian noise with unit variance. Second, decompose them to orthonormal basis with QR or SVD decomposition and replace weights with one of the components.”
- Question @Sylvain, @stas or anyone else who knows: why did Fastai omit the Saxe et al. (2014) orthonormal initialization procedure?
- https://forums.fast.ai/t/lesson-11-discussion-and-wiki/43406/366

Note on a minor tweak that ‘fixes’ the LSUV normalization algorithm

- https://forums.fast.ai/t/lesson-11-discussion-and-wiki/43406/365

Running the Lesson 11 notebook 08_data_block.ipynb

- https://forums.fast.ai/t/running-the-lesson-11-notebook-08-data-block-ipynb/56346