LSTM Tagger

Philipp Windischhofer

February 13, 2017

The Workflow

Training

- match tracks to their associated jets (contained in different ROOT trees)
  - done *ahead* of training
- save matched tracks / jets to disk (HDF5)
- for each track in the jet, feed the track parameters into the classifier network during training
  - (may) use p_T ordering, i.e. hardest track first
- supervised training: provide a binary (0/1) output value for each jet (from MC truth)

Results so far (work in progress)

trained a number of LSTM networks, scanned the hyperparameters:

- number of nodes in each layer
- number of layers
- number of training epochs

Details of the training:

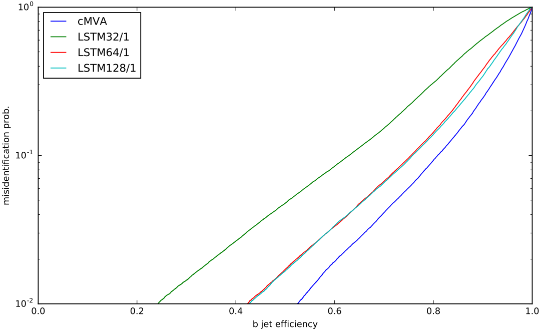
- training dataset contains 100-150k jets
- training/validation split 80:20
- batch size: 1k jets (also tried unbatched training)

Progress w.r.t. last time:

- made code (much) more flexible
- ready to efficiently ramp up training & try different network types

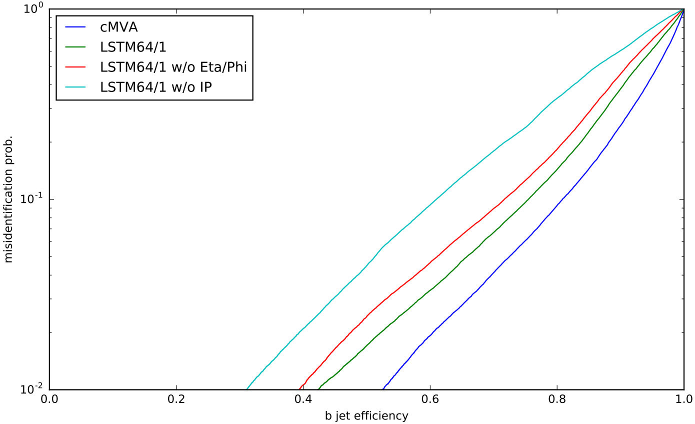
different network shapes (unbatched)

- batch size = 1
- labelling: *number nodes / number layers*



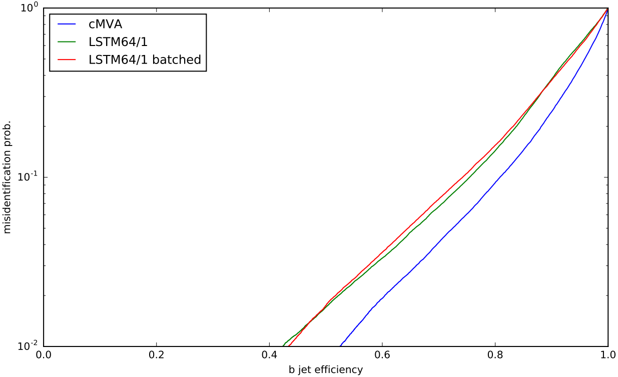
3-layer networks very similar...

using less information



batched training

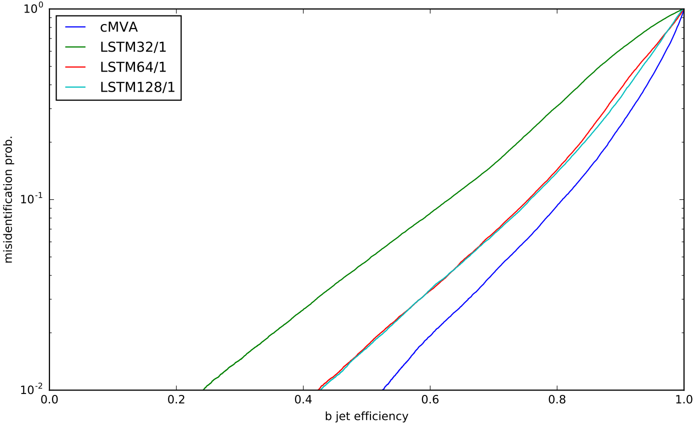
batch size = 1k jets



What is a reasonable batch size to use?

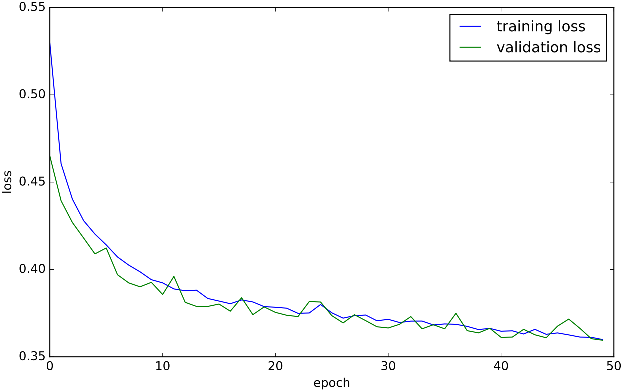
different network shapes (batched)

- batch size $= 1$
- labelling: *number nodes / number layers*



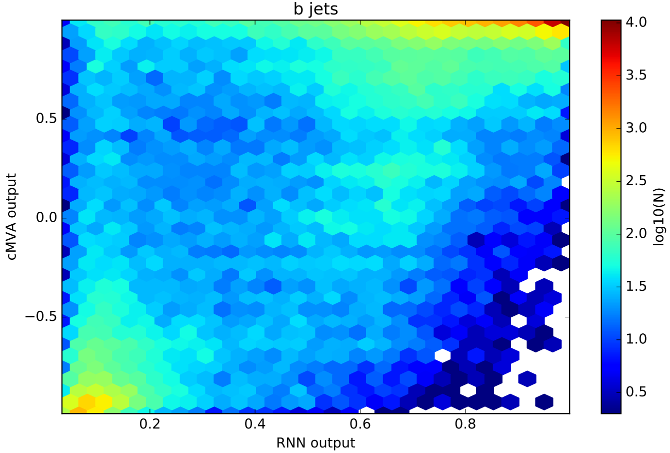
LSTM64/1: batched training

batch size = 1k jets → much smoother descent

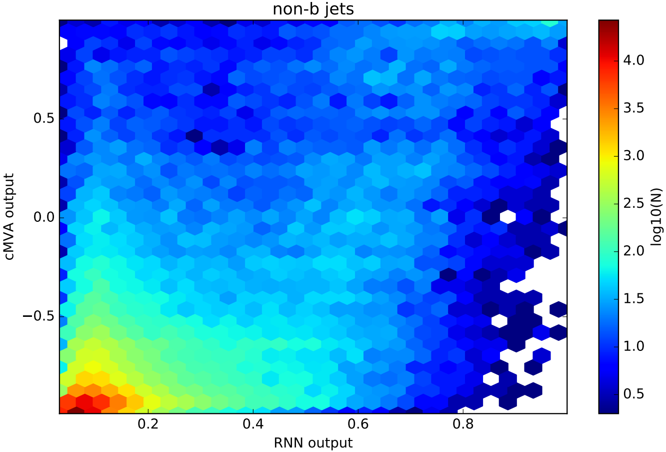


TODO: add callback for terminating the training once saturated!

LSTM64/1: batched training



LSTM64/1: batched training



Conclusions