h2o deep learning different results per run

Asked 3 years, 9 months ago Active 3 years, 9 months ago Viewed 613 times

0

I use the h2o deep learning using python on a data of 2 balanced classes "0" and "1", and adjusted the parameters to be as follows:

```
prostate_dl = H2ODeepLearningEstimator(
     activation=,"Tanh"
     hidden=[50,50,50],
     distribution="multinomial",
    score_interval=10,
    epochs=1000,
    input_dropout_ratio=0.2
    ,adaptive_rate=True
    , rho=0.998, epsilon = 1e-8
    )

prostate_dl .train(
x=x,
y=y,
training_frame =train,
validation_frame = test)
```

Each time the program runs gives different confusion matric and accuarcy results, can anyway explain that? how can the results can be reliable?

Also, all of the runs gives the majority prediction as class "1" not "0" , is their any suggestion?

deep-learning h2o Edit tags

Share Edit Follow Close Flag

asked Nov 4 '17 at 2:47
user8883441
3 3

Please move "Also, all of the runs gives the majority prediction as class "1" not "0" , is their any suggestion?" to a separate question (and provide a reproducible example). – Erin LeDell Nov 4 '17 at 3:14

1 Answer

Active Oldest Votes

1

This question has already been answered here, but you need to set `reproducible=TRUE` when you initialize the `H2ODeepLearningEstimator` in Python (or in `h2o.deeplearning()` in R).

Even after setting `reproducible=TRUE`, the H2O Deep Learning results are only reproducible when using a single core; in other words, when `h2o.init(nthreads = 1)`. The reasons behind this are outlined here.

Also, per the H2O Deep Learning user guide:

> **Does each Mapper task work on a separate neural-net model that is combined during reduction, or is each Mapper manipulating a shared object that's persistent across nodes?**
>
> Neither; there's one model per compute node, so multiple Mappers/threads share one model, which is why H2O is not reproducible unless a small dataset is used and force_load_balance=F or reproducible=T, which effectively rebalances to a single chunk and leads to only one thread to launch a map(). The current behavior is simple model averaging; between-node model averaging via "Elastic Averaging" is currently in progress.

Share Edit Follow Flag

edited Jun 20 '20 at 9:12
Community ♦
1 1

answered Nov 4 '17 at 2:58
Branden Murray
476 2 10

Thank you. Yes it works when I set reproducible=TRUE and seed=1. – user8883441 Nov 5 '17 at 0:38

Can anyone suggest how can I let the results not biased to a certain class, although the 2 classes are balanced in training phase? – user8883441 Nov 5 '17 at 1:22