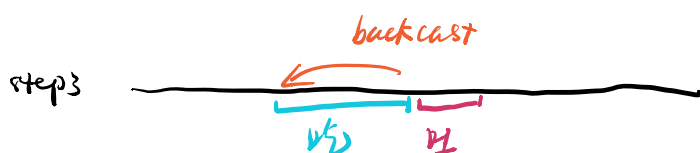


① 用 $x - \hat{x}$ 代替 $y - \hat{y}$ 作残差的优势



step 4 ~~~~~

第三次吃入后 backcast 的那一段 $x_3 - \hat{x}_3$

刚好覆盖前两次

吐预测的那两段

Concatenate ($y_1 - \hat{y}_1, y_2 - \hat{y}_2$)

所以用 $x - \hat{x}$ 不仅在 block 间可以实践类似 Boosting 的效果。

其在 stack 间的作用亦类似传统上直接用 $y - \hat{y}$ 作残差的方式 (其实还是不一样)

② 这个优势实际在作者给的程序中不存在

training samples 抽取方式

training.csv 共 99 行数据。

从每一行随机切一段

这么大的数据。作为 (x, y) 一个 (sample, target)

取满全部 99 行, 得 99 组 pairs, 作一个 training - batch.

smape 输出方式

每训 plot_result 个 batches 输出一次 smape

save_model ()

prediction 画图

问题

1° 作者使用了 10000 个 batch.

然而 model 里有 104046 个 trainable parameters.

2° 抽取 batch 时的 Resampling 过程破坏了 ① 中构造的对称使用 $x-\hat{x}$ 合理性的解释

③ 作者关于自己 doubly Residual Stacking 结构

合理性自圆其说的 experiments 在 Appendix B.3.

1. parallel Backward Residual \rightarrow disabled

每个 block 独都吃原始数据 x
 \hat{y}_i 求和.

2. no-residual Backward Residual \rightarrow disabled

每个 block 独都吃 \pm 一个的 backward \hat{x}
 \hat{y}_i 求和.

3 last-forward Backward Residual \rightarrow activated.

只用本 stack 内最后一个 block 作本 stack 输出

4. no-residual

last-forward Backward Residual \rightarrow disabled

每个 block 独都吃 \pm 一个的 backward \hat{x}

只用本 stack 内最后一个 block 作本 stack 输出

其实

应该验证 backward Residual disabled

用 $y_i - \hat{y}_i$ 喂下一个 block

partial forward activated.

但作者没写