Adadelta のlrPの最適化

Adadeltaは非常にたちが悪い。学習率が低すぎても高すぎても振動が発生する。

rd.BezierCurve.debugmode=False

rd.BezierCurve.mloop\_itt=3

rd.BezierCurve.convg\_coe = 1e-5

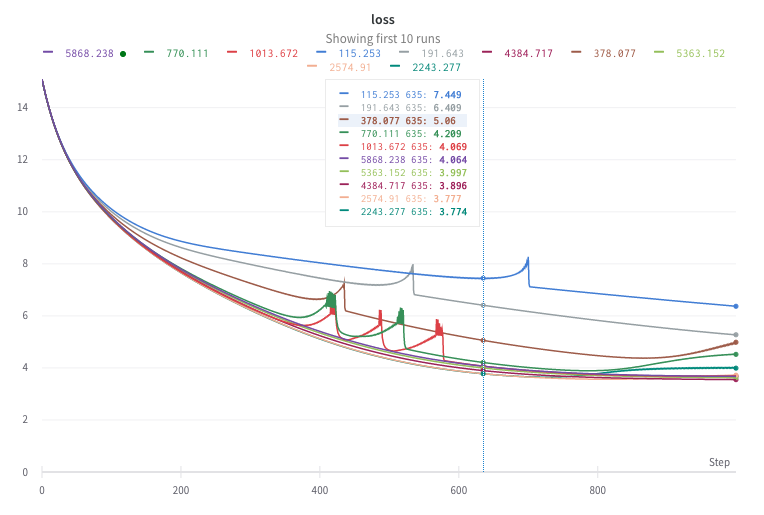
rd.BezierCurve.swing\_penalty = 300

rd.BezierCurve.smoothness\_coe = 4e-6

err,etime = check1T(4,mode=0,N=9,pat=500,maxTry=1000,err\_th=0.65,optimizer\_name='Adadelta',lr=lr,lrP=lrP,test=obj,coe=lr,wandb=wandb)

fitT mode 0 image4 N=9 の例

lr = 0.013 に固定、lrP を Optunaで試行錯誤



1000以下では、1000ステップまでに突然エラーが大きくなり、大きなノコギリ状の波形となる。2000以上になると1000ステップまでにはのこぎり歯状変動は見られないが、右端を見るとエラーが上昇し始めているのが見て取れる。

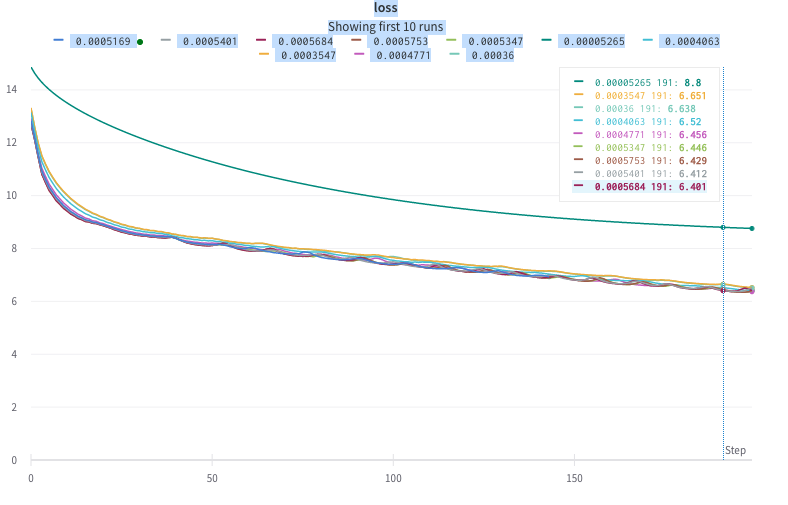
RMSprop の lrとlrPの最適化

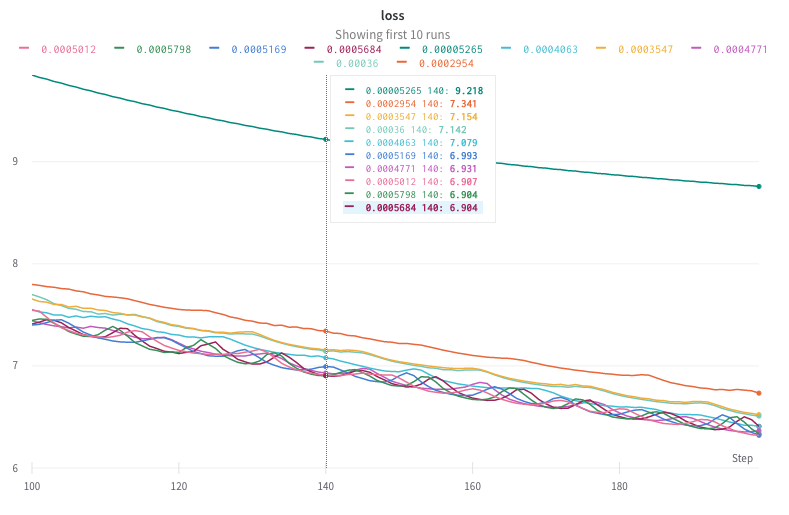
image 4 N=9 mode 0 の fitT

rd.BezierCurve.mloop\_itt=1

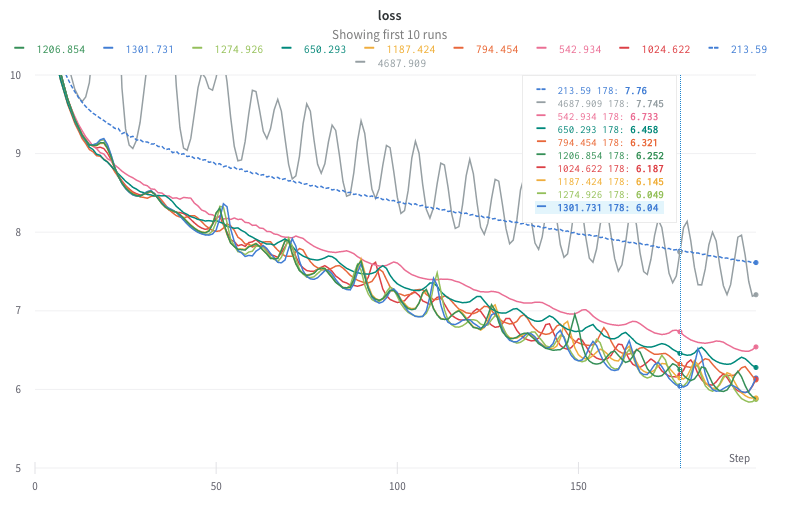
lrP = 600

Adamとその兄弟の場合、0.005 あたりが良好な結果をもたらすのだが、RMSpropは一桁程度小さくしないと振動が激しい。



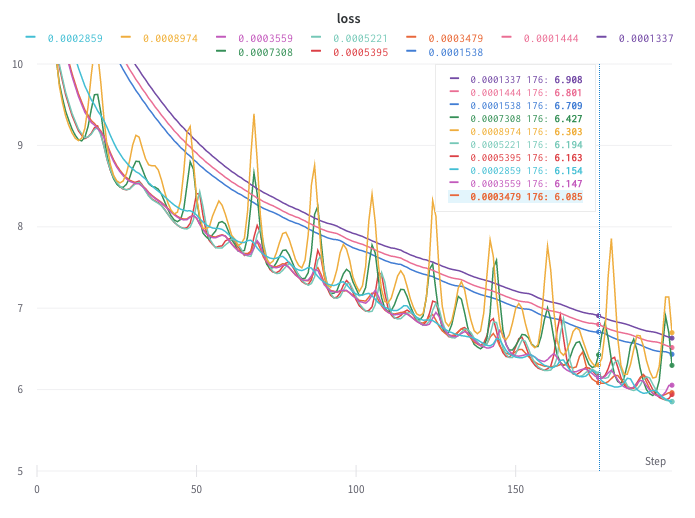


この結果lr=0.0005ぐらいが妥当と判断。この値に固定し、lrP の最適化を試みる



lrP の値は小さすぎても大きすぎてもだめで、 小さい方から増やしていくと、lossは下がっていくが、徐々に振幅が大きくなってきて、1300あたりが平均勾配が最大となる。1000〜1300は大きな差はない。

こんどは lrP を 1300に固定し、lr の最適値を探る



lr = 0.0001から0.00035 あたりまでは lr が大きいほど同じステップ数でのエラーは低くなるが、0.00035 を超えて大きくすると逆にエラーが多くなってしまう。

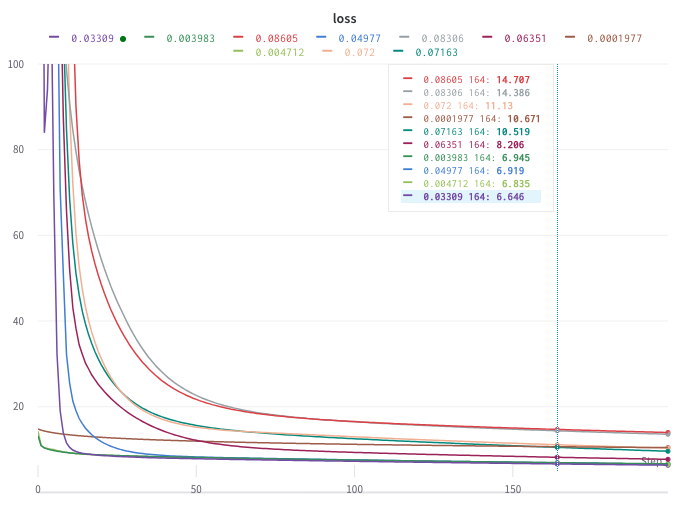
lrP = 1300 の場合は lr = 0.00035 ぐらいが最適　と言っても、0.0002 〜 0.0005 はあまり違いがない。lr が大きいと振幅が大きくなるので、lr = 0.0003ぐらいがよいのかも。

Adagradのlr(learning\_rate)の最適化

３回繰り返しの場合で実験

lr=0.02

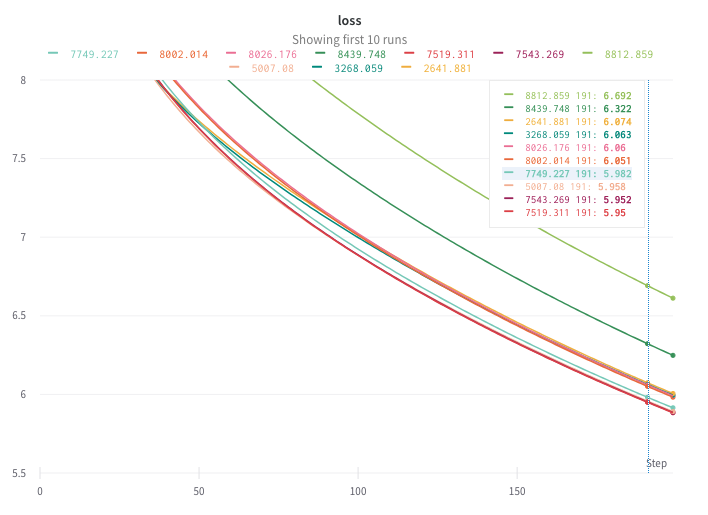
lrP=250 が最適と判断



lr=0.03 より大きい値、0.0002 より小さい値だとエラーは大きくなる。

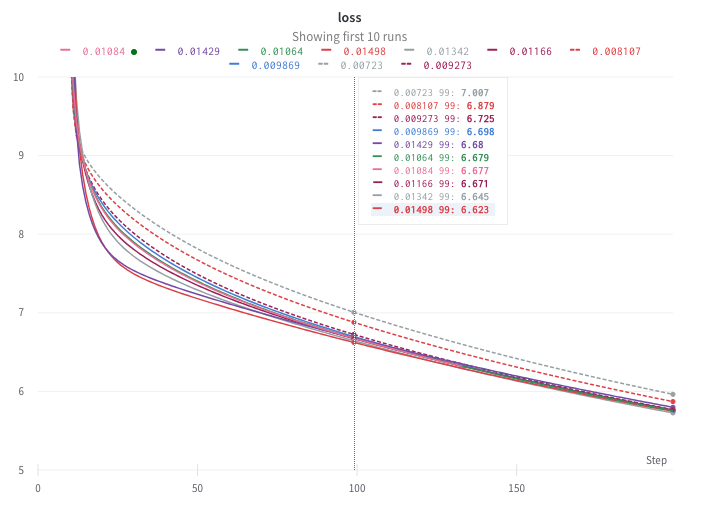
0.0002と0.03の間で実験　→　0.006 以上はあまり変わらない

r = 0.008 で固定し、lrP を最適化



lrP = 2600〜8000は大きな差はなく、7500あたりが最適。これより小さくても大きくてもエラーが増加する。

lrP = 7500で固定して、lr を最適化

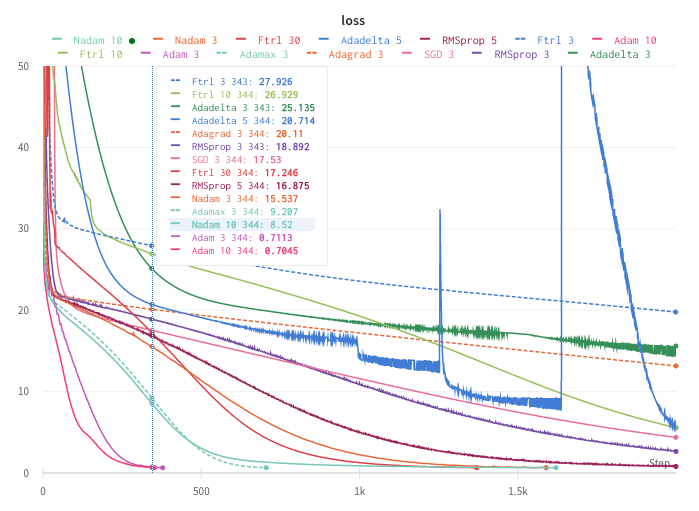


lr = 0.01から0.015 まで大きな違いがない。0.013 ぐらいを最適値とする。

Optimizers

| 手法 | lr（媒介変数の学習係数） | lrP（制御点の学習係数の倍率） | 備考 |
| --- | --- | --- | --- |
| SGD | 5e-6(4.5e-6〜5.5e-6) | 2e6(1.85〜2.1e6) |  |
| Adam | 0.0015(0.001〜0.007) | 1140 | mloop\_ittが３未満だと不安定。 |
| Adamax | 0.0075(0.005-0.0075） | 1000(700〜1300) | （mloop\_itt=3の場合）lr が 0.0078になると急にエラー上昇 |
| AMSgrad | 0.005(0.003〜0.006) | 650 | lr 0.007 になると急にエラー上昇 |
| Adagrad | 0.02 | 250 | （mloop\_itt=3の場合） |
| Adadelta | 0.013（0.012〜0.014） | 3500（3000-4000） | mloop\_itt （媒介変数の修正をn実行した後、制御点の修正をn回実行という繰り返しによる n ）の変更に敏感。もともとlr が大きい激しく発散収束を繰り返すが、mloop\_itt を少しでも大きくすると振幅が大きくなる。３ぐらいが限界。 |
| RMSprop | 0.0003 | 1300 | mloop\_ittを大きくできない。５が限界 |
| Nadam | 0.001 | 500 |  |
| Ftrl | 0.12 | 1000 | mloop\_itt をかなり大きく（たとえばn=30とか）にしても発散せず、収束曲線はほとんど変わらない。 |

<https://wandb.ai/aquapathos/keep-fit1t-0-0/reports/-21-05-25-07-05-32--Vmlldzo3MjE3NzY>

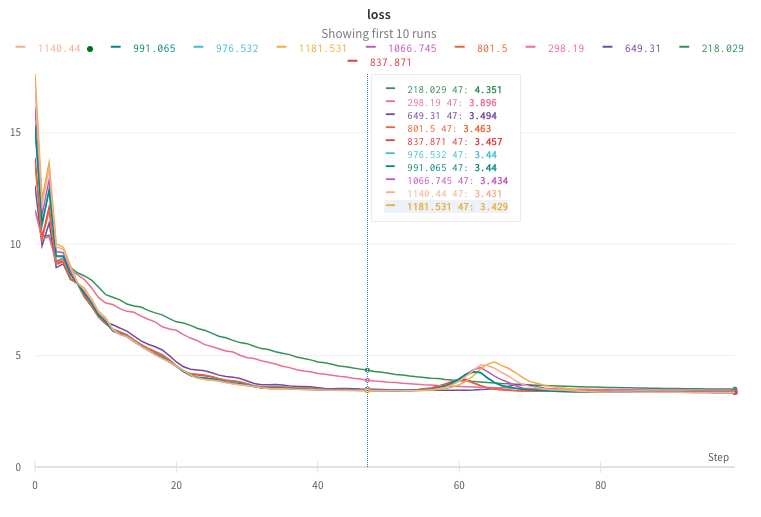


https://user-images.githubusercontent.com/5820803/119416690-2e219f00-bd2f-11eb-8439-d6c92e640a8f.png

adam, ,adamax,

adagrad, rmsprop, sgd, adadelta, ftrl

Adam



lrPを６５０より大きくするほど誤差は小さくなる傾向にあるが、あまり変わらず、不安定さがますので、650でよい

BezierCurve.mloop\_ittの効果

Ftrl と Adam で mloop\_itt（１回の繰り返しで実行するminimize の回数）と収束時間の関係を調べた。

対象 image1 fit1T mode 0 N=7

| mloop\_itt | Adam cpu time | Adam steps | Ftrl cpu time（2000ステップ相当時間） | Ftrl steps |
| --- | --- | --- | --- | --- |
| 1 | 604.08322 （未収束） | 2000 | 417.15907 | 2000 |
| 2 | 1097.56240（548.7812） | 2000 | 757.11568（378.55784) | 2000 |
| 3 | 283.71916 | 359（1077）BEST | 1097（365.67） | 2000 |
| 5 | 523.87891 | 412（2060） | 1768.11160（353.62232） | 2000 |
| 10 | 1000.18566 | 401（4010） | 3475.97312（347.597312） | 2000 |
| 20 | 1457.08230 | 296（5920） | 6897.00488（344.850244） | 2000 |
| 30 | 1897.60998 | 259 (7770) | 10312.71470（343.75716） | 2000 |