

## Aya の 2021UEC 杯アピール文章

- ・昨年までは LeelaZero の棋譜を使って学習させていましたが今回は KataGo の棋譜を使っています。Policy と Value のみを学習。

KataGo の棋譜

<https://katagoarchive.org/>

- ・Aya の playout も利用。探索は MCTS。

- ・ネットワークの構造

昨年は 192filter、15block の ResNet でしたが

今年は Tristan の論文(\*1),(\*2)を参考に MixNet を採用してみました。

ただし 3x3 と 5x5 の 2 つではなく、3x3 と 3x3 を 2 つ、です。

192x10block の ResNet との比較実験ではほぼ同じパラメータ数で +146 ELO 強いです。

192x10block(3x3)の ResNet に対する勝率。mb64 はミニバッチ 64 での比較。それ以外はミニバッチ 128

勝 引 負 局数 ELO	モデルのサイズ
358- 56-386 800 ( -12) mobile_20b の論文から	10MB 半分で互角。強い。Tristan
385- 44-371 800 ( 6) mobile_30b	15MB 1.5 倍でも棋力上昇なし
490- 75-235 800 ( 114) mobile_20b_1024	mb64 38MB 強いけどサイズが大きい
461- 90-249 800 ( 94) mix 512-128 3x3,5x5	mb64 16MB
441- 74-285 800 ( 68) mix 512-128 3x3,5x5,7x7	mb64 22MB
503- 85-212 800 ( 132) mix 768-196 3x3,5x5	mb64 33MB
500- 55-245 800 ( 114) mix 640-160 3x3,5x5	mb64 24MB
489-117-194 800 ( 134) mix 640-160 3x3,5x5	mb64 24MB swish
528- 63-209 800 ( 146) mix 640-160 3x3,3x3	mb64 23MB swish
492- 44-264 800 ( 101) mix 640-160 3x3,5x5	24MB
376- 72-352 800 ( 10) 256x20b 26MB)	89MB サイズが4倍(192x10bは 26MB)
299- 62-439 800 ( -61) 192x 5b	13MB

実験条件は

Aoba 駒落ちの棋譜を 1000 万棋譜から 1040 万棋譜まで、学習率 0.01、ミニバッチ 128 で 10 万回学習、での比較です。

局面数では 1280 万局面を学習、になります。Policy と Value を学習しています。

MixNet の構造は(\*2)の Tristan の論文で紹介されている bottleneck\_block を参考に丸ごと並列化しています。

DepthWise だけを並列、だとなぜか Caffe で動きませんでした。

filters = 512 -> 640

trunk = 128 -> 160 (3x3 が 80、5x5 が 80)

にそれぞれ変更して、これが 19 ブロックです。

大雑把に書くと下のよう感じです。

```
conv      160 1x1, bn,relu  1 層目のみ
----- bottleneck block, これを繰り返し
conv      640 1x1, bn,relu  <- input
dw_conv   640 3x3, bn,relu          dw...depthwise convolution
conv      80 1x1, bn      -> A
conv      640 1x1, bn,relu  <- input
dw_conv   640 3x3, bn,relu
conv      80 1x1, bn      -> B
concat                                <- A, <- B
eltwise <- input
-----
```

Caffe の Depthwise layer は下を使っています。

<https://github.com/yonghenglh6/DepthwiseConvolution>

MixNet を試してみました

<https://524.teacup.com/yss/bbs/3869>

(\*1) Cosine Annealing, Mixnet and Swish Activation for Computer Go

[https://icga.org/wp-content/uploads/2021/11/ACG\\_2021\\_paper\\_9.pdf](https://icga.org/wp-content/uploads/2021/11/ACG_2021_paper_9.pdf)

(\*2) Mobile Networks for Computer Go

[https://www.researchgate.net/publication/343848982\\_Mobile\\_Networks\\_for\\_Computer\\_Go](https://www.researchgate.net/publication/343848982_Mobile_Networks_for_Computer_Go)

-----  
Aya's 2021 UEC Cup appeal

Last year, I used LeelaZero's game records for training, but this time I use KataGo's game records.

Only Policy and Value are trained.

KataGo's game record

<https://katagoarchive.org/>

I also use Aya's playout. Ssearch is MCTS.

Network structure

Last year, I used ResNet with 192 filters and 15 blocks.

but this year, I used MixNet based on Tristan's paper(\*1),(\*2).

However, instead of 3x3 and 5x5, I used two 3x3.

In a comparison experiment with 192x10 block ResNet,

two 3x3 is +146 ELO is strong with almost the same number of parameters.

Win rate against baseline ResNet of 192x10block. Batchsize is 128. mb64 means 64.

Win	Draw	Loss	Games	ELO	model size
			( 0)	ResNet 192x10block	26MB
358-	56-	386	800 (-12)	mobile_20b	10MB
385-	44-	371	800 ( 6)	mobile_30b	15MB
490-	75-	235	800 (114)	mobile_20b_1024	mb64 38MB
461-	90-	249	800 ( 94)	mix 512-128 3x3,5x5	mb64 16MB
441-	74-	285	800 ( 68)	mix 512-128 3x3,5x5,7x7	mb64 22MB
503-	85-	212	800 (132)	mix 768-196 3x3,5x5	mb64 33MB
500-	55-	245	800 (114)	mix 640-160 3x3,5x5	mb64 24MB

489-117-194	800 ( 134) mix 640-160 3x3,5x5	mb64	24MB swish
528- 63-209	800 ( 146) mix 640-160 3x3,3x3	mb64	23MB swish
492- 44-264	800 ( 101) mix 640-160 3x3,5x5		24MB
376- 72-352	800 ( 10) 256x20b		89MB
299- 62-439	800 ( -61) 192x 5b		13MB

The experimental conditions are as follows:

10 million to 10.4 million AobaKomauchi games are trained. learning rate is fixed 0.01.

100,000 training step runs in 128 mini-batches.

Policy and Value have been learned.

The structure of MixNet is based on bottleneck\_block introduced in Tristan's paper (\*2).

For some reason, Caffe didn't work when I parallelized only DepthWise.

filters = 512 -> 640

trunk = 128 -> 160 (80 for 3x3, 80 for 3x3)

There is 19 blocks.

Roughly speaking, it looks like this

```

conv      160 1x1, bn,relu  only for 1st layer
-----
conv      640 1x1, bn,relu  <- input
dw_conv   640 3x3, bn,relu  dw...depthwise convolution
conv      80 1x1, bn        -> A
conv      640 1x1, bn,relu  <- input
dw_conv   640 3x3, bn,relu
conv      80 1x1, bn        -> B
concat                                <- A, <- B
eltwise <- input
-----

```

I use this Depthwise layer for Caffe.

<https://github.com/yonghenglh6/DepthwiseConvolution>

I tried MixNet( in Japanese)

<https://524.teacup.com/yss/bbs/3869>

(\*1) Cosine Annealing, Mixnet and Swish Activation for Computer Go

[https://icga.org/wp-content/uploads/2021/11/ACG\\_2021\\_paper\\_9.pdf](https://icga.org/wp-content/uploads/2021/11/ACG_2021_paper_9.pdf)

(\*2) Mobile Networks for Computer Go

[https://www.researchgate.net/publication/343848982\\_Mobile\\_Networks\\_for\\_Computer\\_Go](https://www.researchgate.net/publication/343848982_Mobile_Networks_for_Computer_Go)