Overcoming catastrophic forgetting in NeuralNetworks

山田真徳

目的

Catastrophic forgetting(taskAの後にtaskBを学習するとtaskAを忘れてしまうこと)を回避する方法を提案

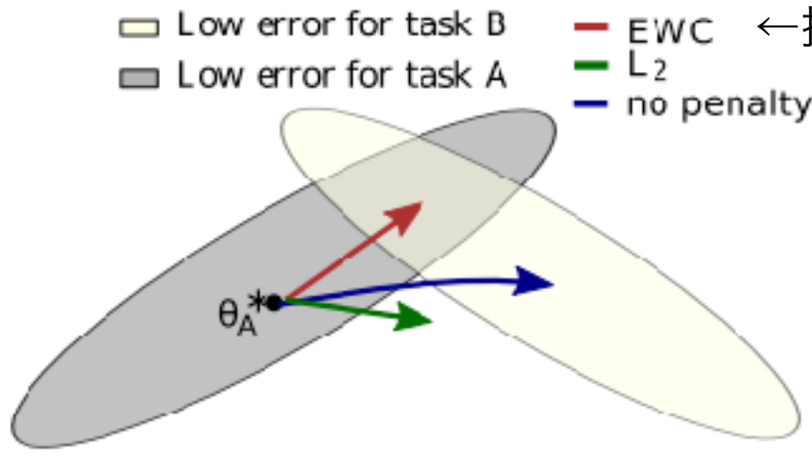
taskAとtaskBのデータをシャッフルさせて学習しないことを仮定

戦略：Fisher情報量が大きいパラメータは重要だと思って忘れないように学習する

実験：Catastrophic forgettingを回避できることをmnistの分類問題とAtari gamesで試した

目的関数

小さいlossを出すパラメータの組み合わせはたくさんあるので、
taskAとtaskB同時にlossを小さくするパラメータを探したい



←提案手法(Elastic weight consolidation)

統合

L2

$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \frac{\lambda}{2}\sum_i \left(\theta_i - \theta^*_{A,i}\right)^2$$

EWC

$$\mathcal{L}(\theta) = \mathcal{L}_B(\theta) + \frac{\lambda}{2}\sum_i F_i \left(\theta_i - \theta^*_{A,i}\right)^2$$

↑

Fisher information matrix

理論

EWCの導出

$$\log p(\theta|D) = \log \frac{p(D|\theta)\,p(\theta)}{p(D)}$$

$$= \log \frac{p(D_A|\theta)\,p(D_B|\theta)\,p(\theta)}{p(D_A)\,p(D_B)}$$

$$= \log p(D_B|\theta) - \log p(D_B) + \log \frac{p(D_A|\theta)\,p(\theta)}{p(D_A)}$$

$$= \log p(D_B|\theta) - \log p(D_B) + \log p(\theta|D_A)$$

Bの尤度関数　　最適化に無関係

仮定

$$p(D_A, D_B|\theta) = p(D_A|\theta)\,p(D_B|\theta)$$

$$p(D_A, D_B) = p(D_A)\,p(D_B)$$

taskAのすべての情報が事後分布に含まれている

真の事後分布 $p(\theta|D_A)$ の計算は無理なのでMacKayのLaplace近似を使う

↑

物理でいうsaddle point 近似の実数版

Laplace近似(1次元)

分布のピークx_0で展開する

ピークなので1階微分は消える

$$\log p(x) \simeq \log p(x_0) - \frac{c}{2}(x - x_0)^2 + \cdots$$

$$c \equiv -\frac{\partial^2}{\partial x^2} \log p(x)|_{x=x_0}$$

2次の項までをGauss分布で近似する

$$Q^*(x) \equiv p(x_0) \exp\left(-\frac{c}{2}(x - x_0)^2\right)$$

normalizationを行って全体の確率が1になるようにする(漸近展開をぶった切ったので)

$$Q(x) \equiv \frac{1}{Z_Q} p(x_0) \exp\left(-\frac{c}{2}(x - x_0)^2\right)$$

$$Z_Q \equiv p(x_0)\sqrt{\frac{2\pi}{c}}$$

MacKayのLaplace近似(多次元版)

k次元だと

$$\log p\left(x\right) \simeq \log p\left(x_0\right) - \frac{1}{2}\left(x - x_0\right)_i A_{ij} \left(x - x_0\right)_j + \cdots$$

$$A_{ij} \equiv -\frac{\partial}{\partial x_i}\frac{\partial}{\partial x_j} \log p\left(x\right)|_{x=x_0}$$

normalizationを行って全体の確率が1になるようにする

$$Q\left(x\right) \equiv \frac{1}{Z_Q} p\left(x_0\right) \exp\left(\frac{1}{2}\left(x - x_0\right)_i A_{ij} \left(x - x_0\right)_j\right)$$

$$Z_Q \equiv p\left(x_0\right)\sqrt{\frac{\left(2\pi\right)^K}{\det A}}$$

Fisher情報行列になっている

Fisher情報行列

$$F_{ij} = E_x\left[-\frac{\partial}{\partial\theta_i}\frac{\partial}{\partial\theta_j}\log p\left(\theta|x\right)\right]$$

$$\downarrow \quad E_x\left[\frac{\partial^2}{\partial\theta^2}p\left(\theta|x\right)\right] = 0$$

$$F_{ij} \equiv E_x\left[\frac{\partial}{\partial\theta_i}\log p\left(\theta|x\right)\frac{\partial}{\partial\theta_j}\log p\left(\theta|x\right)\right]$$

$$\begin{aligned}
F_{ij} &= E_x\left[-\frac{\partial}{\partial\theta_i}\frac{\partial}{\partial\theta_j}\log p\left(\theta|x\right)\right] \\
&= E_x\left[-\frac{\partial}{\partial\theta_i}\frac{1}{p\left(\theta|x\right)}\frac{\partial}{\partial\theta_j}p\left(\theta|x\right)\right] \\
&= E_x\left[\left(\frac{1}{p\left(\theta|x\right)}\frac{\partial}{\partial\theta_i}p\left(\theta|x\right)\right)^2\right] \\
&= E_x\left[\left(\frac{\partial}{\partial\theta_i}\log p\left(\theta|x\right)\right)^2\right]
\end{aligned}$$

計算コストを低くするため対角成分のみ計算

Fisher情報行列を対角化することも可能なので対角化して計算すれば完璧な計算が可能、
ただしθの基底が混ざるので毎回補正する必要がある

$$\log p\left(\theta|D\right) = \log p\left(D_B|\theta\right) - \log p\left(D_B\right) + \frac{\lambda}{2}\sum_i F_i\left(\theta_i - \theta^*_{A,i}\right)^2$$

$$\mathcal{L}\left(\theta\right) = \mathcal{L}_B\left(\theta\right) + \frac{\lambda}{2}\sum_i F_i\left(\theta_i - \theta^*_{A,i}\right)^2$$

実験

実験1:mnistの分類問題

Data

taskA:普通のmnistの分類問題
taskB:画素をランダムシャッフルしたmnist(※シャッフルの規則は全ての画像で同じ)
taskC:画素をランダムシャッフルしたmnist(※シャッフルの規則は全ての画像で同じ)

Algorithm

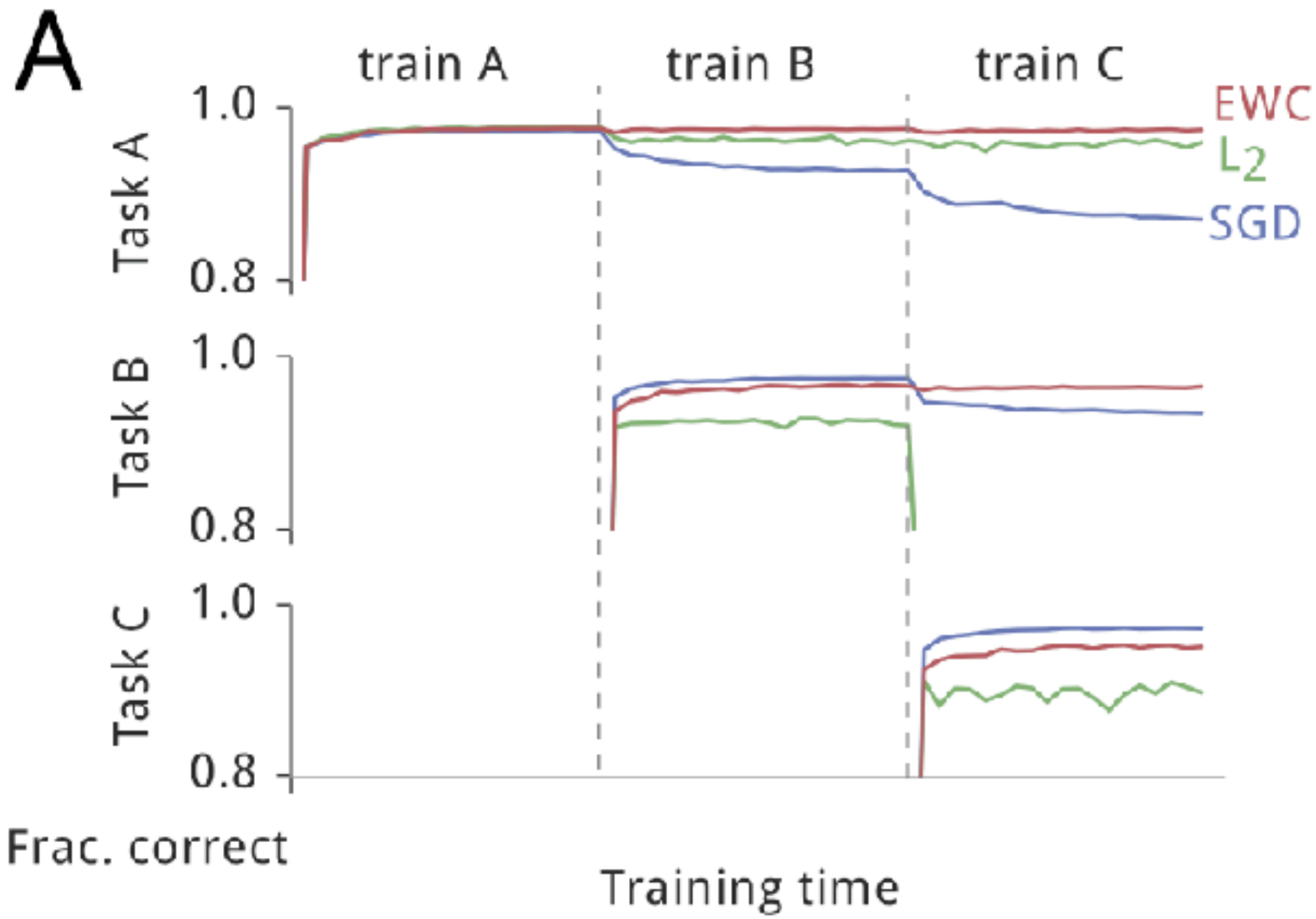
全結合のnetwork

optimizer:SGD

Hyperparameter	Reference figure		
	3A	3B	3C
learning rate	10^{-3}	10^{-5}-10^{-3}	10^{-3}
dropout	no	yes	no
early stopping	no	yes	no
n. hidden layers	2	2	6
width hidden layers	400	400-2000	100
epochs / dataset	20	100	100

Table 1: Hyperparameters for each of the MNIST figures

validationデータに対する精度



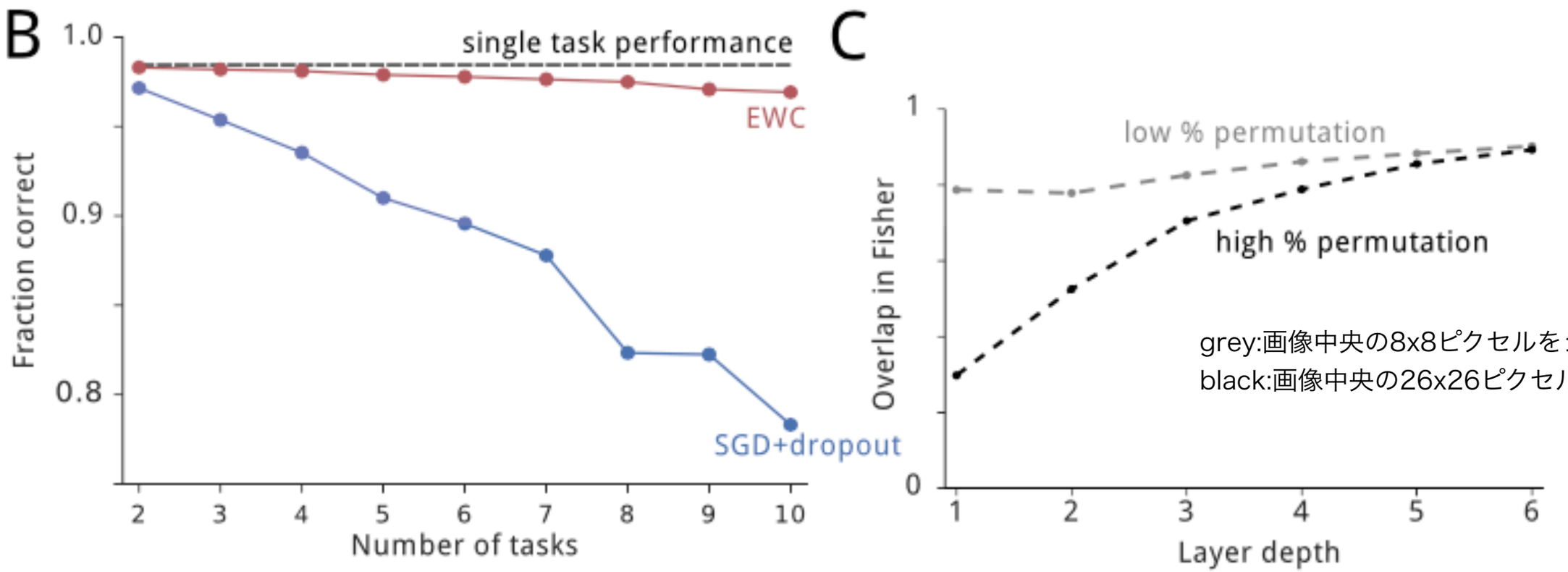
EWC:忘れず楽ちゃんと学べている

L2:他のタスクを学びづらい

SGD:忘却する

精度/タスク数

Fisher情報量の類似度/layerの場所



grey:画像中央の8x8ピクセルをシャッフル
black:画像中央の26x26ピクセルをシャッフル

Taskが増えてもEWCは忘れにくい

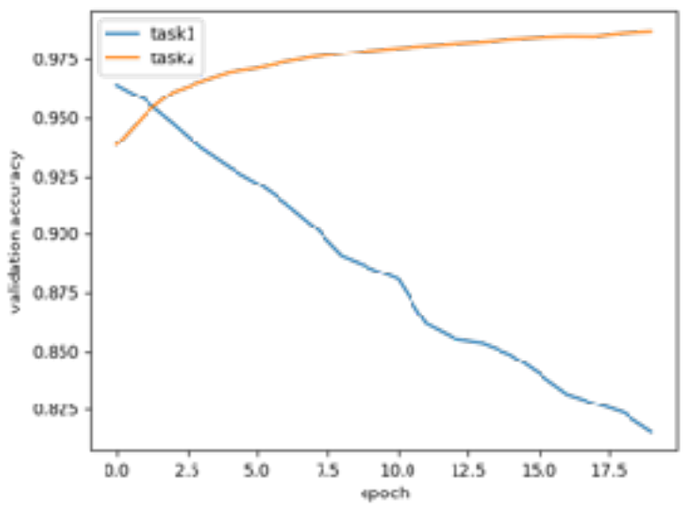
taskが違うと浅い層での
fisher情報行列に違いが出る

overlap fisher: Fréchet distance

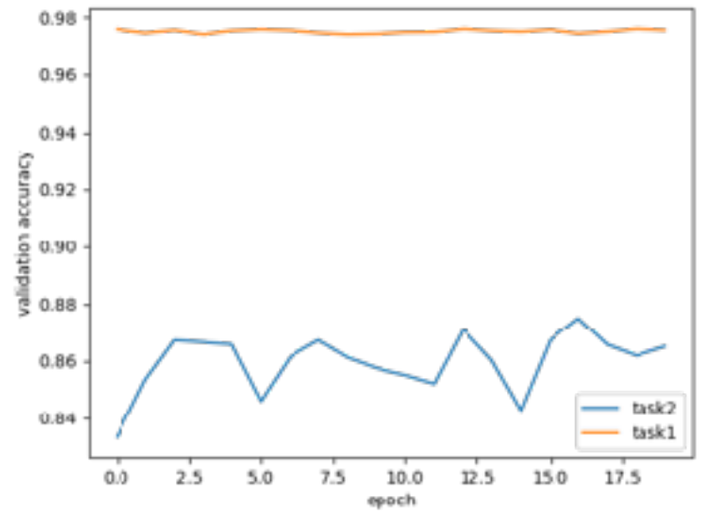
$$d^2(\hat{F}_1, \hat{F}_2) = \frac{1}{2}\mathrm{tr}\left(\hat{F}_1 - \hat{F}_2 - 2(\hat{F}_1\hat{F}_2)^{1/2}\right)$$
$$= \frac{1}{2}\|\hat{F}_1^{1/2} - \hat{F}_2^{1/2}\|_F$$

フロベニウスノルム

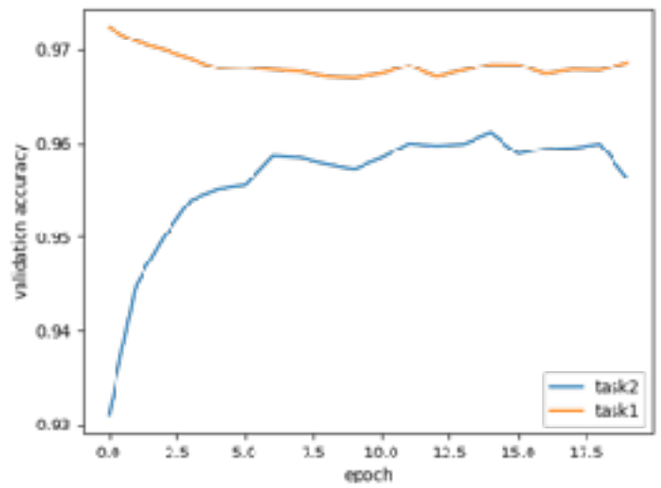
mnist自分でもやってみた
task1学習後にtask2を学習



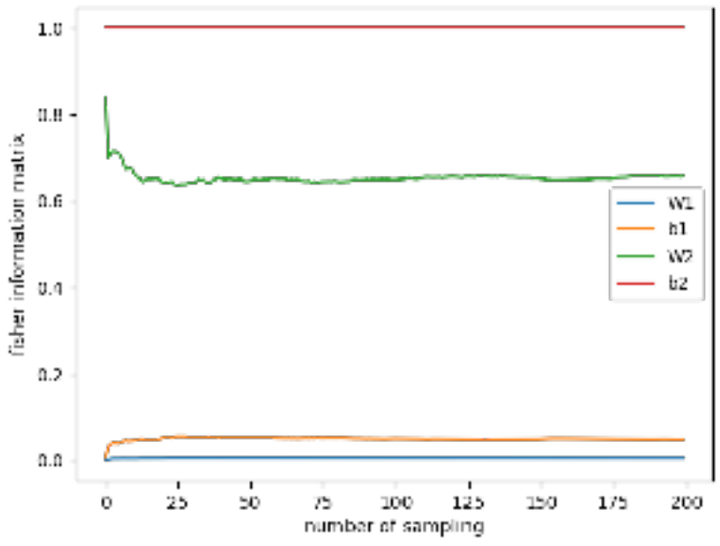
SGD

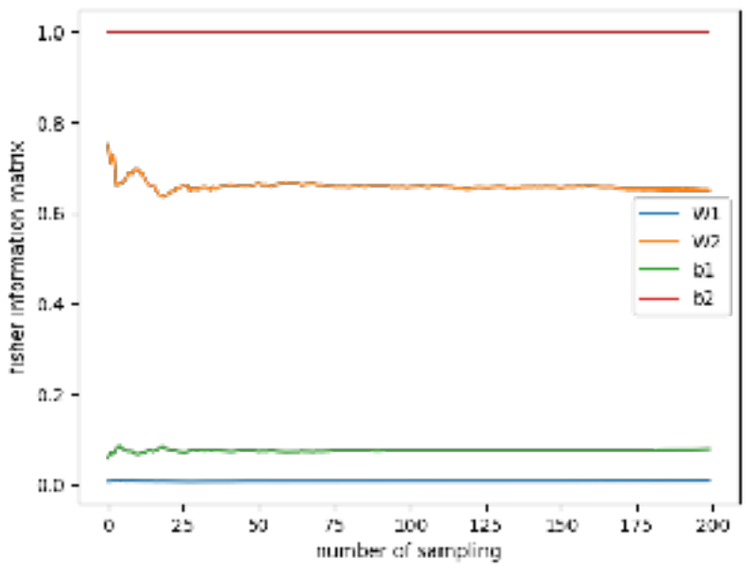


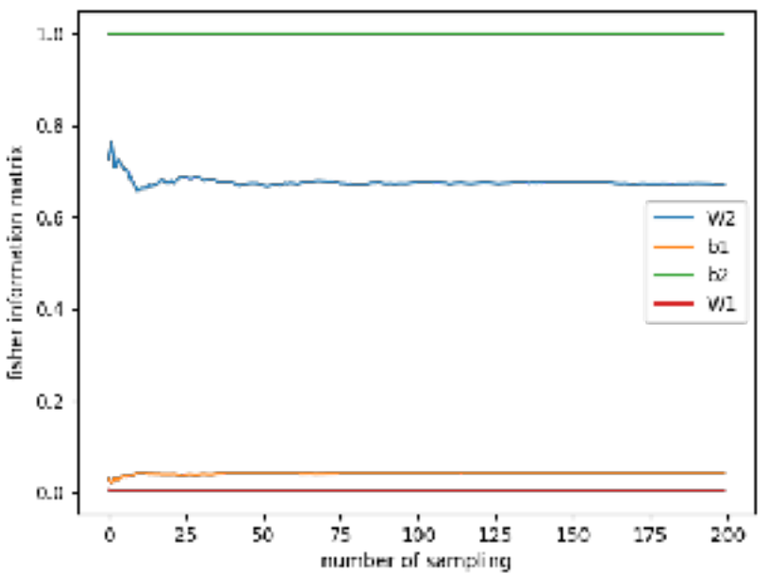
L2



EWC







Fisher情報行列の色がそろってなくてすいません・・・

実験2:Atari gameの強化学習

Task

18のゲーム(standard dqnで人間のレベルをこえたもの)
からランダムに10個のゲーム選びプレイ
これを10セット行う
異なるシードで4回プレイしてそのスコアの平均を測定

Algorithm

Double DQNの論文と一緒
conv→conv→conv→fc

画像の前処理はNatrueのDQNと一緒
210x160(x3)→84x84(x1)

actionの数は18
(Atari gameは全部18以下のアクションでプレイできる)
replay memoryはgameごとに持つ

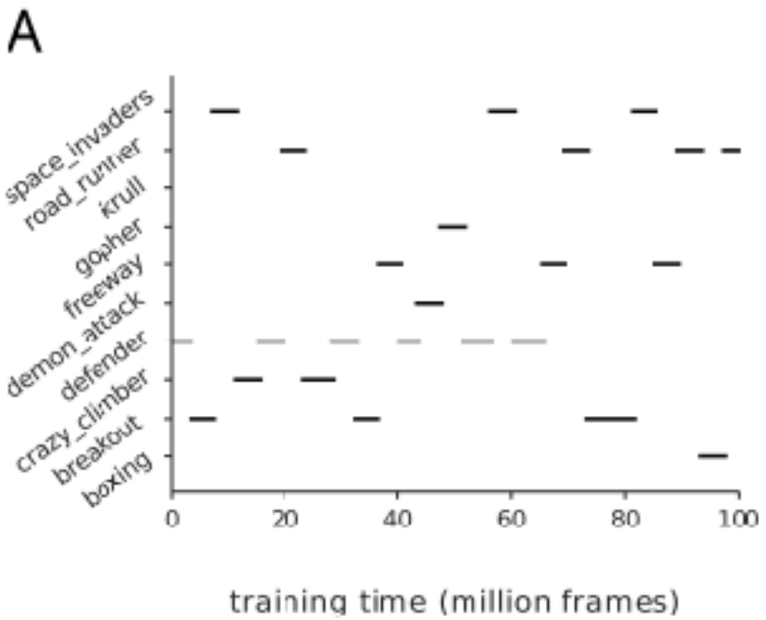
λ
ペナルティーを始めるタイミング

Hyperparameter	value	brief description
action repeat	4	Repeat the same action for four frames. Each agent step will occur every fourth frame.
discount factor	0.99	Discount factor used in the Q-learning algorithm.
no-op max	30	Maximum number of do nothing operations carried out at the beginning of each training episode to provide a varied training set.
max. reward	1	Rewards are clipped to 1.
scaled input	84x84	Input images are scaled to 84x84 with bilinear interpolation.
optimization algorithm	RMSprop	Optimization algorithm used.
learning rate	0.00025	The learning rate in RMSprop.
max. learning rate	0.0025	The maximum learning rate that RMSprop will apply.
momentum	0.	The momentum used in RMSprop.
decay	0.95	The decay used in RMSProp.
clipδ	1.	Each gradient from Q-learning is clipped to $\pm$ 1.
max. norm	50.	After clipping, if the norm of the gradient is greater than 50., the gradient is renormalized to 50.
history length	4	The four most recently experienced frames are taken to form a state for Q-learning
minibatch size	32	The number of elements taken from the replay buffer to form a mini-batch training example.
replay period	4	A mini-batch is loaded from the replay buffer every 4 steps (16 frames including action repeat).
memory size	50000	The replay memory stores the last fifty thousand transitions experienced.
target update period	7500	The target network in Q-learning is updated to the policy network every 7500 step.
min. history	50000	The agent will only start learning after fifty thousand transitions have been stored into memory.
initial exploration	1.	The value of the initial exploration rate.
exploration decay start	50000	The exploration rate will start decaying after fifty thousand frames.
exploration decay end	1050000	The exploration rate will decay over one million frames.
final exploration	0.01	The value of the final exploration rate.
model update period	4	The Dirichlet model is updated every fourth step.
model downscaling	2	The Dirichlet model is downscaled by a factor of 2, that is an image of size 42x42 is being modelled.
size window	4	The size of the window for the task recognition model learning.
num. samples Fisher	100	Whenever the diagonal of the Fisher is recomputed for a task, one hundred mini-batches are drawn from the replay buffer.
Fisher multiplier	400	The Fisher is scaled by this number to form the EWC penalty.
start EWC	20E6	The EWC penalty is only applied after 5 million steps (20 million frames).

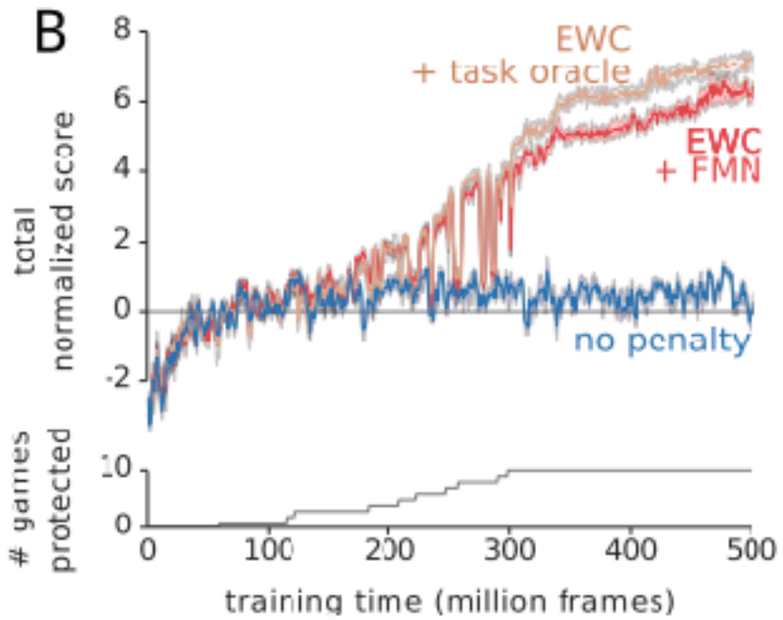
Table 2: Hyperparameters for each of the MNIST figures

終盤で各ゲーム固有の情報を学ぶと仮定して途中からペナルティーを入れる

学習のスケージュリング



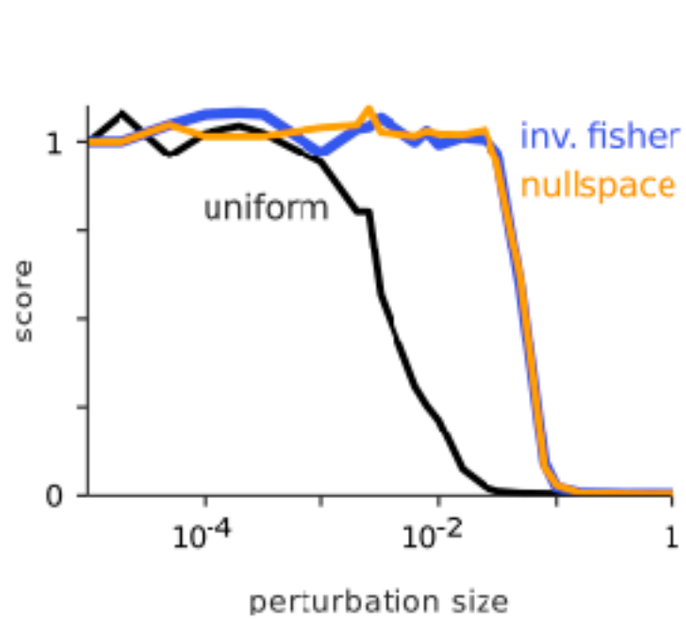
taskのスコアの平均



赤：タスクのラベルをForget Me Not Algorithmで与える
ブラウン：タスクのラベルが与えられる

強化学習においてもEWCは忘却を防いでいる

single gameのweightにノイズを
加えたときのスコアの性能



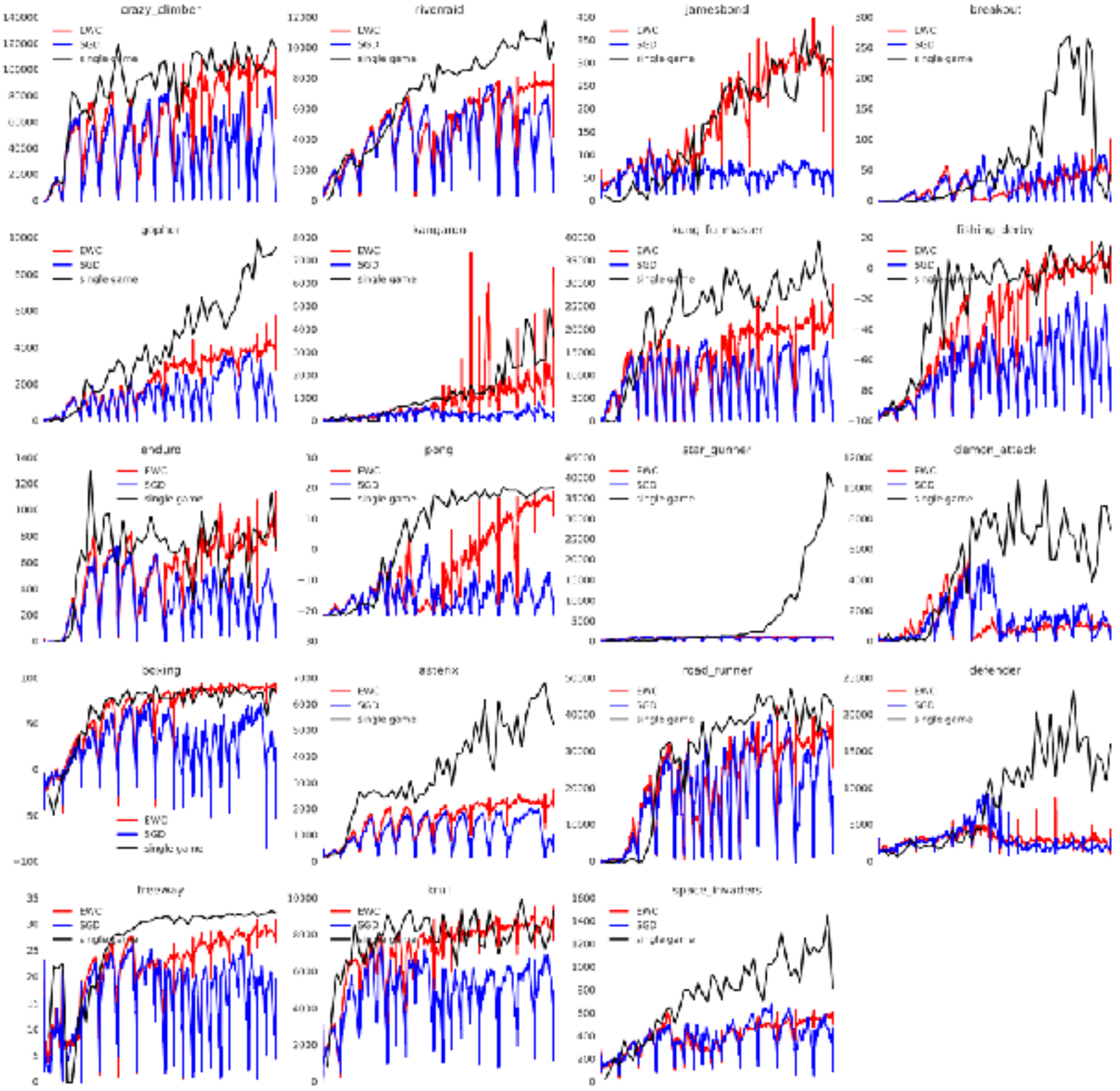
Breakout学習済み

黒：Gaussノイズを入れた場合のスコア
青：Gaussノイズ*Fisher情報行列の逆数
橙：Fihser情報行列が0のところにノイズ

EWCはスコアに影響する大切な重みを
探してきている

Forget me not Algorithm@Nips2016；http://papers.nips.cc/paper/6055-the-forget-me-not-process

各ゲームごとのスコアの比較



ゲームによっては事後分布をガウス分布で近似するのが上手くいっていない