

Loss of Plasticity in Deep Continual Learning

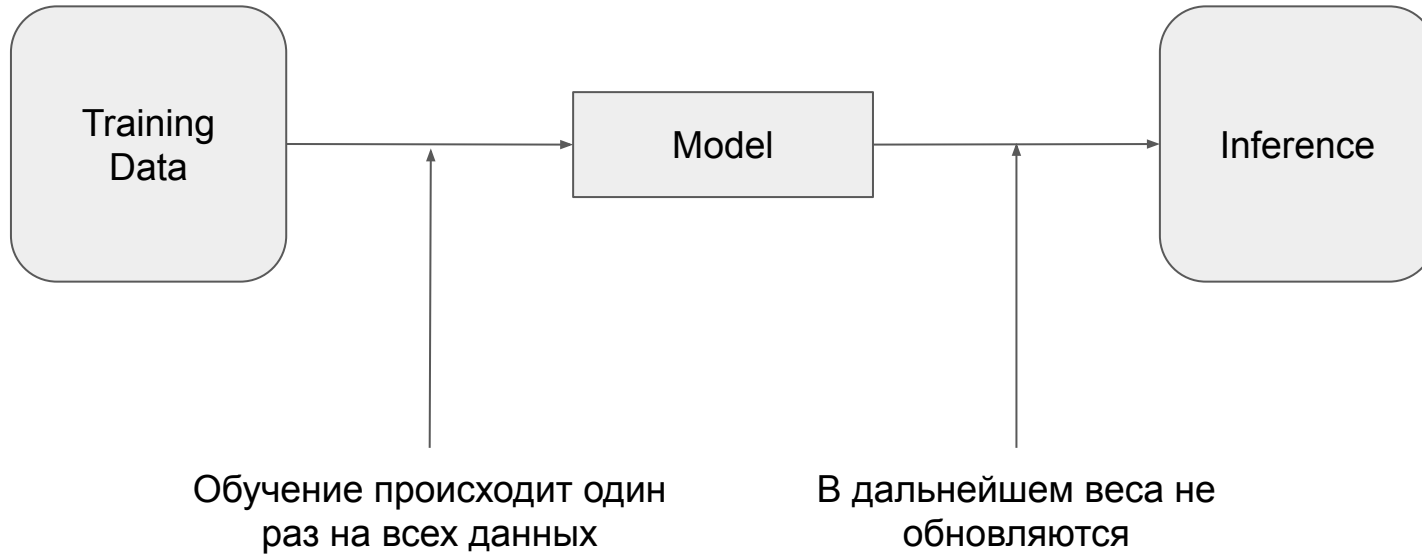
Выполнил:

Разин Арслан Дмитриевич, БПМИ202

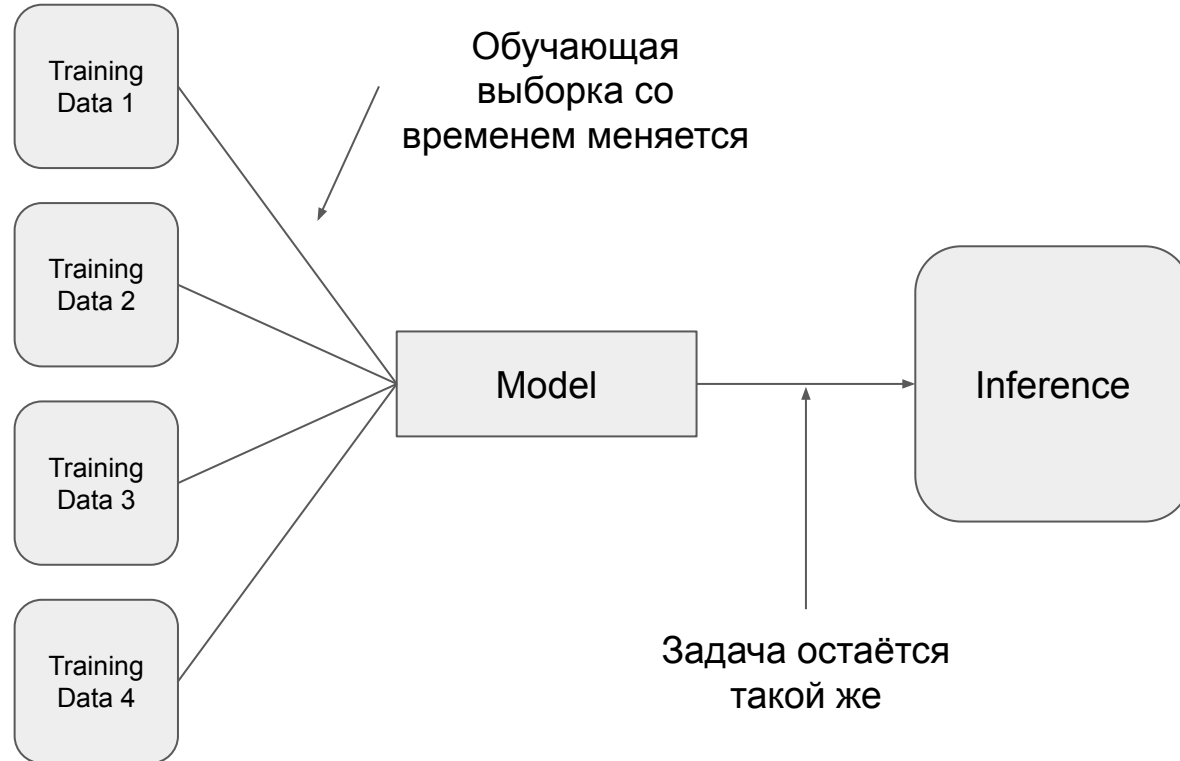
1. Введение

- 2. Continual ImageNet и Permuted MNIST
- 3. Метрики пластичности
- 4. Обзор существующих решений
- 5. Continual Backpropagation
- 6. Приложения и выводы

Classical Deep Learning (train-once)



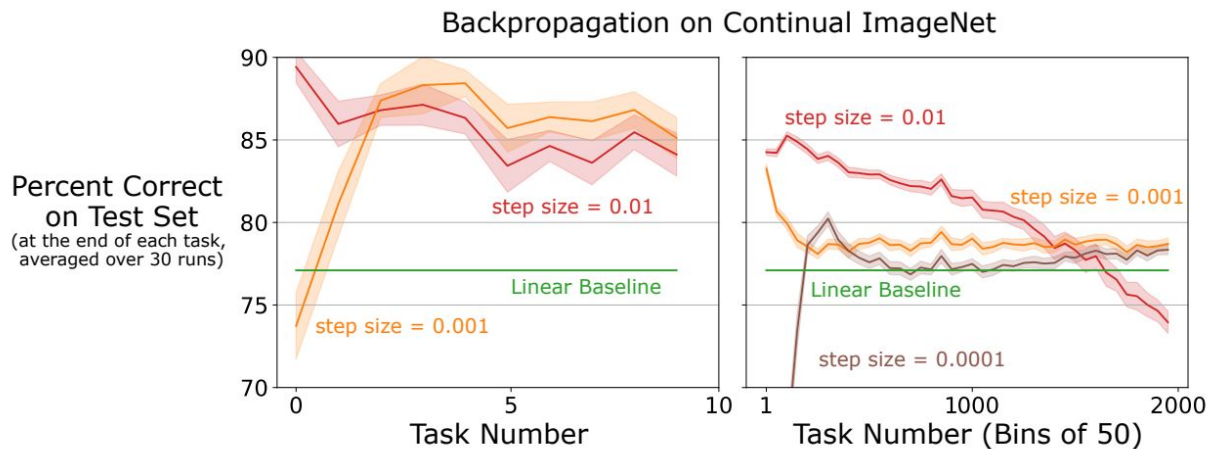
Continual Deep Learning



Проблема:
Классические модели
со временем
перестают обучаться
на новых данных и их
нужно заново обучать
с нуля

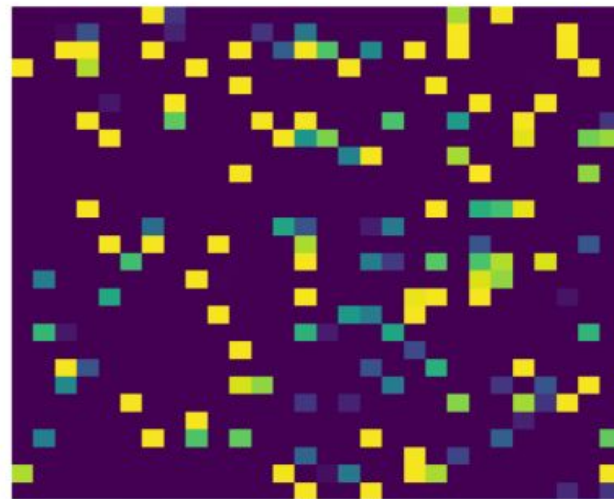
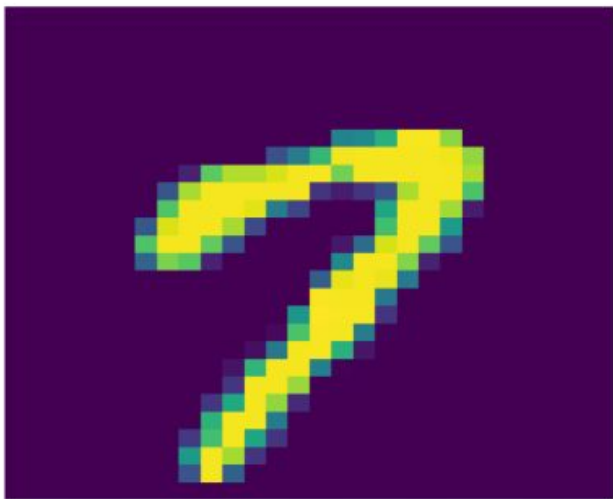
1. Введение
- 2. Continual ImageNet и Permuted MNIST**
3. Метрики пластичности
4. Обзор существующих решений
5. Continual Backpropagation
6. Приложения и выводы

Continual ImageNet



2000 задач бинарной классификации из двух различных классов ImageNet размера 32x32 пикселя

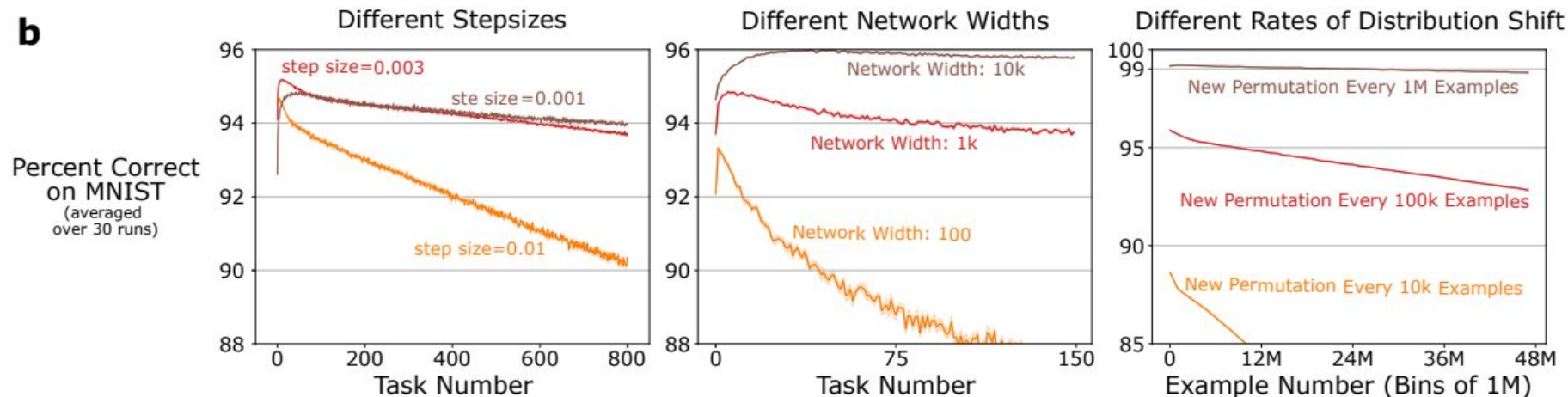
Permuted MNIST



Для каждой из 60000 картинок применяется одна из 800 перестановок пикселей

Permuted MNIST

b



Для каждой из 60000 картинок применяется одна из 800 перестановок пикселей

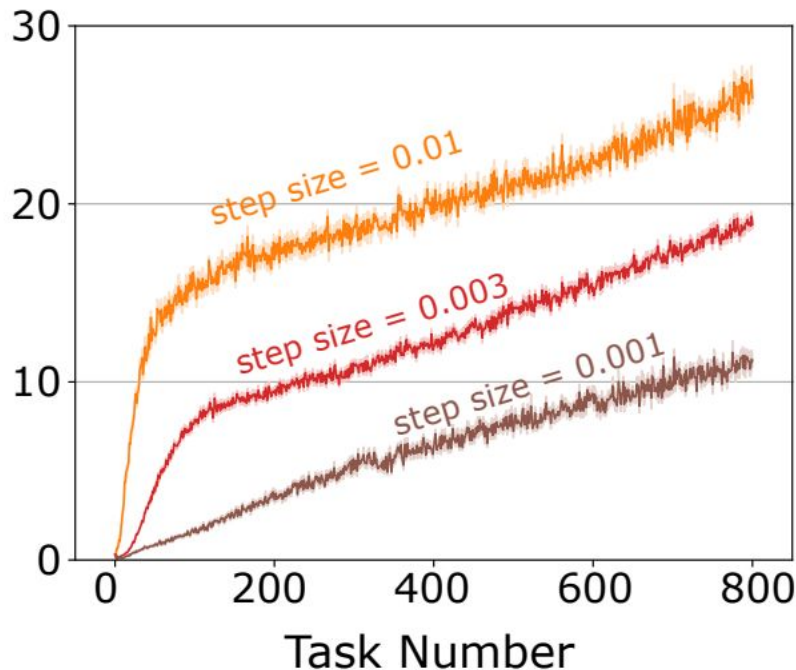
1. Введение
2. Continual ImageNet и Permuted MNIST
- 3. Метрики пластичности**
4. Обзор существующих решений
5. Continual Backpropagation
6. Приложения и выводы

Доля константных весов

Показывает долю весов модели, которые перестают меняться в процессе обучения. Увеличение этой метрики явно показывает, что модель перестает учиться.

Percent of Dead Units

(Computed before each task)

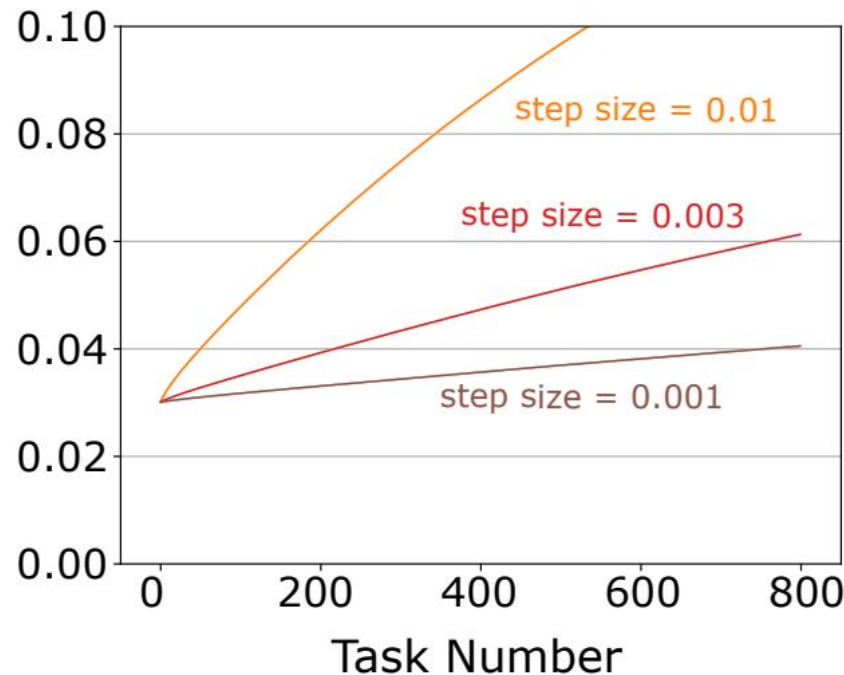


Средняя величина весов

Показывает среднее абсолютное значение весов модели. С ростом этого параметра растет неустойчивость обучения (взрыв градиентов, проблемы в SGD).

Weight Magnitude

(Average over all weights, binned over 60k examples)



Эффективный ранг

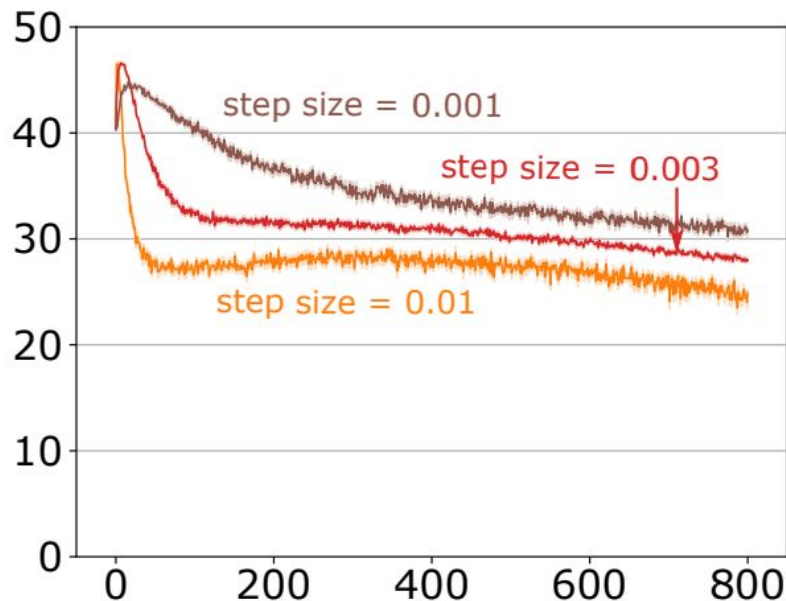
Для скрытых слоев была приближенно
рассчитана следующая метрика:

Formally, consider a matrix $\Phi \in \mathbb{R}^{n \times m}$ with singular values σ_k for $k = 1, 2, \dots, q$, and $q = \max(n, m)$. Let $p_k = \sigma_k / \|\sigma\|_1$, where σ is the vector containing all the singular values, and $\|\cdot\|_1$ is the ℓ^1 -norm. The effective rank of matrix Φ , or $\text{erank}(\Phi)$, is defined as

$$\text{erank}(\Phi) \doteq \exp \{H(p_1, p_2, \dots, p_q)\}, \text{ where } H(p_1, p_2, \dots, p_q) = - \sum_{k=1}^q p_k \log(p_k). \quad (1)$$

Effective Rank

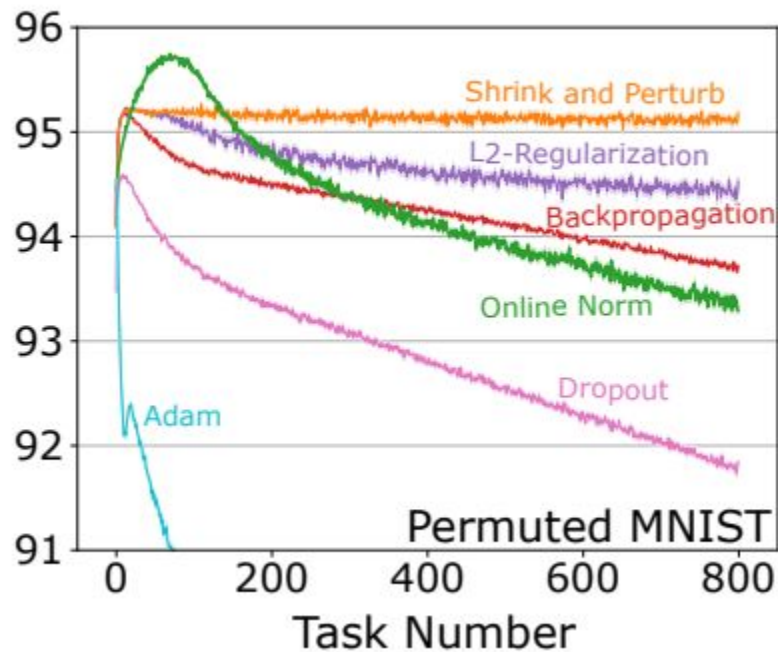
(Computed before each task, Scaled $\in [0, 100]$)



1. Введение
2. Continual ImageNet и Permuted MNIST
3. Метрики пластичности
- 4. Обзор существующих решений**
5. Continual Backpropagation
6. Приложения и выводы

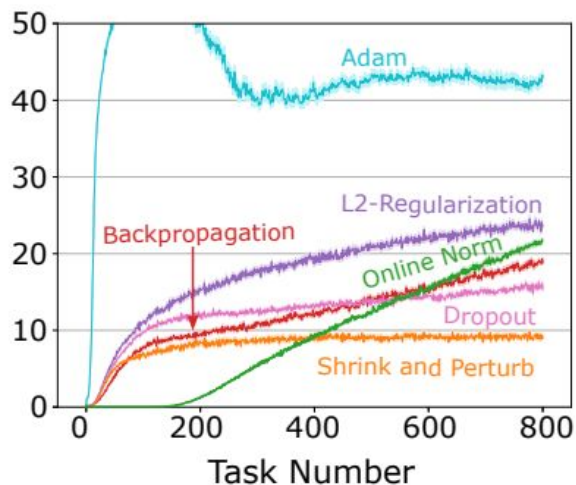
Качество с разными методами

Percent Correct on MNIST
(averaged over 30 runs)

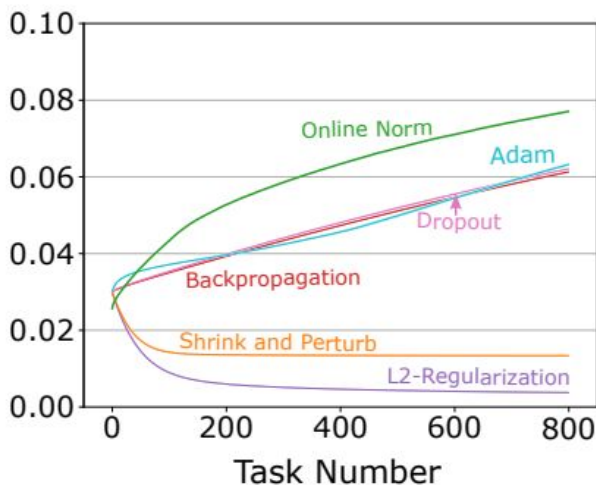


Метрики качества для классических методов

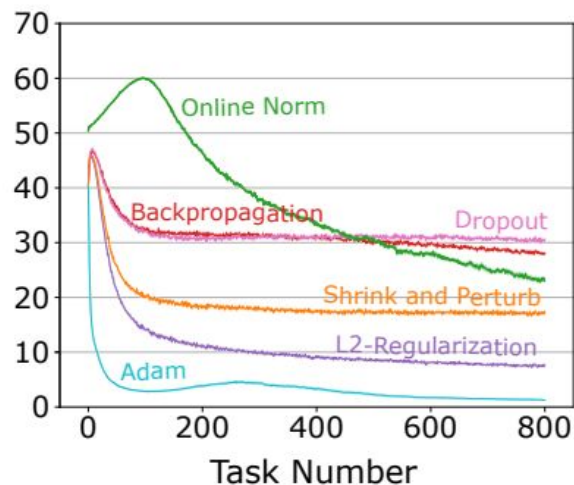
Percent of Dead Units
(Computed before each task)



Weight Magnitude
(Average over all weights)



Effective Rank
(Computed before each task, Scaled to [0,100])

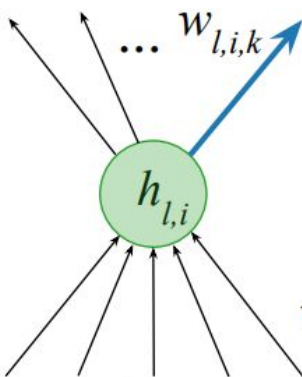


1. Введение
2. Continual ImageNet и Permuted MNIST
3. Метрики пластичности
4. Обзор существующих решений
- 5. Continual Backpropagation**
6. Приложения и выводы

Формулы

$$f_{l,i,t} = \eta * f_{l,i,t-1} + (1 - \eta) * h_{l,i,t},$$

$$\hat{f}_{l,i,t} = \frac{f_{l,i,t-1}}{1 - \eta^{a_{l,i,t}}},$$



$$y_{l,i,t} = \frac{|h_{l,i,t} - \hat{f}_{l,i,t}| * \sum_{k=1}^{n_{l+1}} |w_{l,i,k,t}|}{\sum_{j=1}^{n_{l-1}} |w_{l-1,j,i,t}|}$$

$$y_{l,i,t} = \frac{|h_{l,i,t} - \hat{f}_{l,i,t}| * \sum_{k=1}^{n_{l+1}} |w_{l,i,k,t}|}{\sum_{j=1}^{n_{l-1}} |w_{l-1,j,i,t}|}$$

$$u_{l,i,t} = \eta * u_{l,i,t-1} + (1 - \eta) * y_{l,i,t},$$

$$\hat{u}_{l,i,t} = \frac{u_{l,i,t-1}}{1 - \eta^{a_{l,i,t}}}.$$

Алгоритм

Algorithm 1: Continual backpropagation (CBP) for a feed-forward network with L hidden layers

Set: step size α , replacement rate ρ , decay rate η , and maturity threshold m (e.g. 10^{-4} , 10^{-4} , 0.99, and 100)

Initialize: Initialize the weights $\mathbf{w}_0, \dots, \mathbf{w}_L$. Let, \mathbf{w}_l be sampled from a distribution d_l

Initialize: Utilities $\mathbf{u}_1, \dots, \mathbf{u}_L$, average activation $\mathbf{f}_1, \dots, \mathbf{f}_l$, and ages $\mathbf{a}_1, \dots, \mathbf{a}_L$ to 0

for each input x_t **do**

Forward pass: pass input through the network, get the prediction, \hat{y}_t

Evaluate: Receive loss $l(x_t, \hat{y}_t)$

Backward pass: update the weights using stochastic gradient descent

for layer l in $1 : L$ **do**

Update age: $\mathbf{a}_l += 1$

Update unit utility: Using Equations 4, 5, and 6

Find eligible units: Units with age more than m

Units to reinitialize: $n_l * \rho$ of eligible units with the smallest utility, let their indices be \mathbf{r}

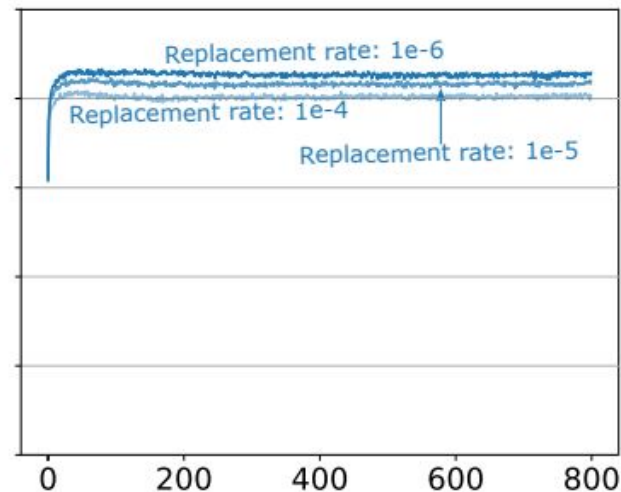
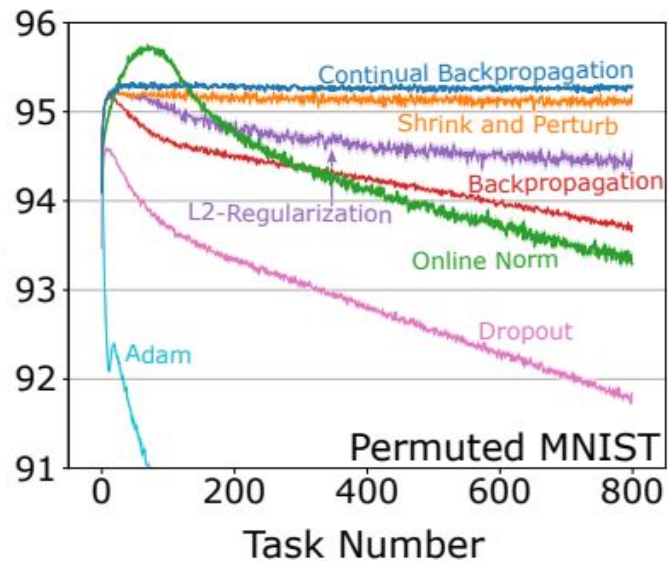
Initialize input weights: Reset the input weights $\mathbf{w}_{l-1}[\mathbf{r}]$ using samples from d_l

Initialize output weights: Set $\mathbf{w}_l[\mathbf{r}]$ to zero

Initialize utility, unit activation, and age: Set $\mathbf{u}_{l,\mathbf{r},t}$, $\mathbf{f}_{l,\mathbf{r},t}$, and $\mathbf{a}_{l,\mathbf{r},t}$ to 0

Итоговое качество

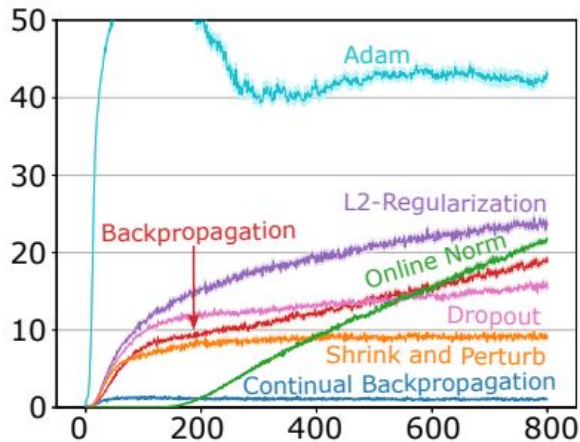
Percent Correct on MNIST
(averaged over 30 runs)



Итоговое качество

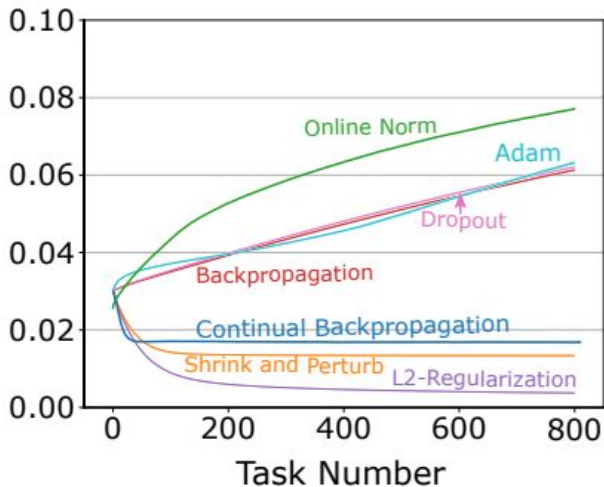
Percent of Dead Units

(Computed before each task)



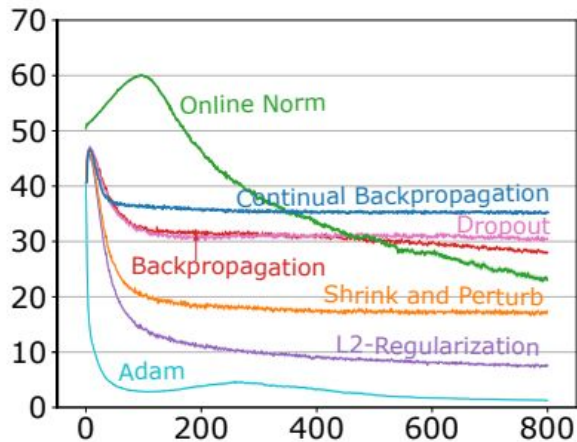
Weight Magnitude

(Average over all weights)



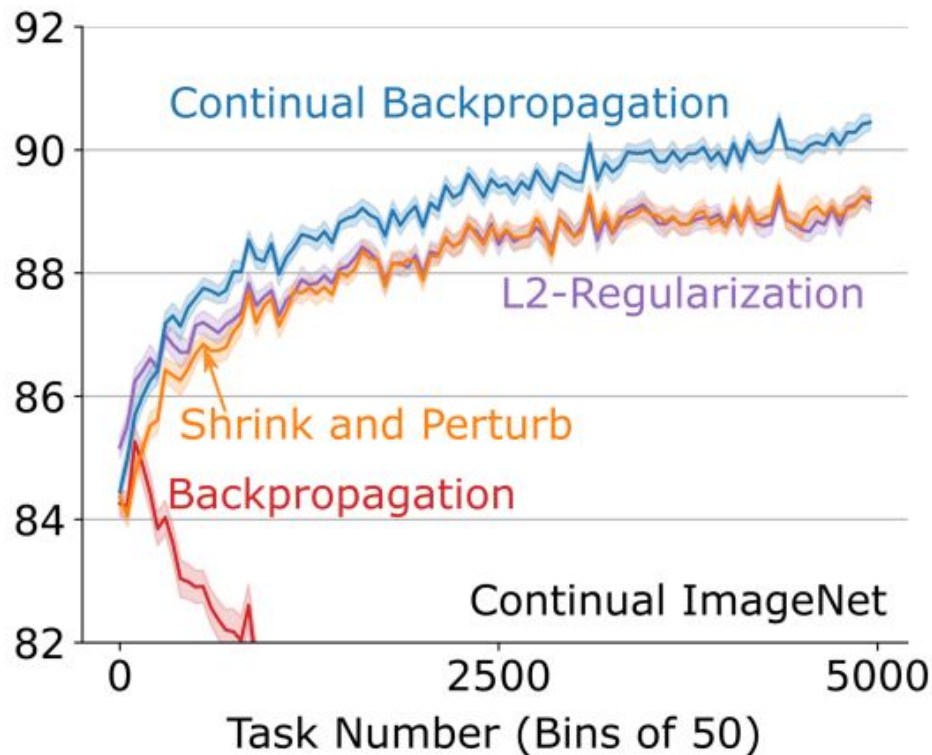
Effective Rank

(Computed before each task, Scaled to [0,100])



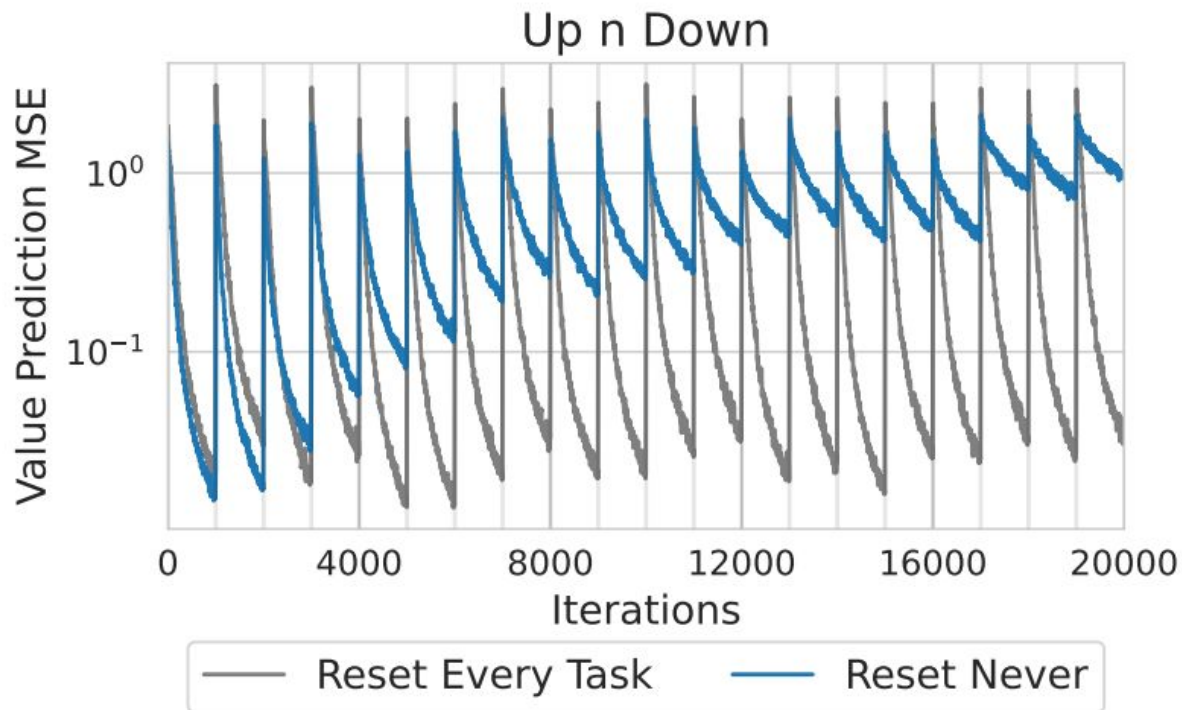
Итоговое качество

Percent Correct
on Test Set
(at the end of each task,
averaged over 30 runs)



1. Введение
2. Continual ImageNet и Permuted MNIST
3. Метрики пластичности
4. Обзор существующих решений
5. Continual Backpropagation
- 6. Приложения и выводы**

Где используется такой подход?



Сильные и слабые стороны

Сильные стороны:

- Универсальный метод непрерывного обучения
- Объяснение проблем существующих методов
- Представлено нескольких методов оценки пластичности

Слабые стороны:

- Обе рассмотренных в статье задачи слишком нестандартные, чтобы считать по ним метрики
- Не рассматривались ансамбли
- Никак не исследовались методы случайной инициализации

Источники:

1. <https://arxiv.org/pdf/2306.13812.pdf>
2. <https://arxiv.org/pdf/2305.15555.pdf>