Knowledge graph completion using Riemann optimization

NLA Final Project

Team: This is an ethical knot

Kamil Mardanshin
Marina Sheshukova
Ekaterina Filimoshina
Ignat Romanov
Ilona Basset

Github:

https://github.com/marina-shesha/NLA_project

Skolkovo
Institute of Science
and Technology

Skoltech

Problem and its relevance

Problem statement: Knowledge graph completion

Knowledge graph is a large graph-structured database that stores facts in a triple form and can be represented as a third-order binary tensor.

$$(\underbrace{e_s}_{\text{subject entity}}, \underbrace{r}_{\text{relation}}, \underbrace{e_o}_{\text{object entity}})$$

In practice, knowledge graphs consist of only a fraction of all possible relations. **The problem** is to use existing relations to complete the knowledge graph by predicting missing relations.

Consider a set of subjects S, objects O, relations R. Then a knowledge graph $E \subset S \times R \times O$. For a parameter set Θ we define a scoring function $\varphi : \Theta \times S \times R \times O \to \mathbb{R}$

$$\mathcal{L}(\theta, E) = \sum_{(s,*,o) \in E} \sum_{r \in R} L(\varphi(\theta, s, r, o), \mathbb{1}\{(s, r, o) \in E\}) \to \min_{\theta \in \Theta}$$

Relevance and applications

Zamini, M.; Reza, H.; Rabiei, M. **A Review of Knowledge Graph Completion**. Information 2022, 13, 396. https://doi.org/10.3390/ info13080396

Tong Shen, Fu Zhang, Jingwei Cheng, **A comprehensive overview of knowledge graph completion, Knowledge-Based Systems**, Volume 255, 2022, 109597, ISSN 0950-7051, https://doi.org/10.1016/j.knosys.2022.109597.

01. Knowledge base completion

Construction of comprehensive knowledge bases by predicting and adding missing relations between entities

02. Search engine enhancement

Improvement of search query relevance by predicting missing links, enhancing the depth and accuracy of search results

03. Recommender systems

Enhancement of recommendation algorithms by predicting latent connections, providing more personalized suggestions

04. Social network analysis

Complete missing connections in social graphs, facilitating more comprehensive analysis of community

02

Existing approaches

Skoltech

Some of the existing solutions

linear

RESCAL (Nickel et al., 2011):

- Optimizes a function with a bilinear product between subject and object entity vectors and a full-rank relation matrix.
- Overfits due to a large number of parameter.

M. Nickel, V. Tresp, and H.-P. Kriegel. ***A Three-Way Model for Collective Learning on Multi-Relational Data****. 2011*

DistMult (Yang et al., 2015):

- A special case of RESCAL with a diagonal matrix per relation, which reduces overfitting.
- Do not model asymmetric relations

B. Yang, W. Yih, X. He, J. Gao L. Deng. ***Embedding Entities and Relations for Learning and Inference in Knowledge Bases****. 2015.*

SimplE (Kazemi and Poole, 2018)

- Is based on Canonical Polyadic (CP) decomposition, in which subject and object entity embeddings for the same entity are independent.

Kazemi, Poole. ***SimplE Embedding for Link Prediction in Knowledge Graphs.*** *2018.*

ComplEx (Trouillon et al., 2016)

- Extends DistMult to the complex domain, where subject and object entity embeddings for the same entity are complex conjugates.
- Enables to model asymmetric relations.

Trouillon, Welbl, Riedel, Gaussier, Bouchard. ***Complex Embeddings for Simple Link Prediction.*** *2016.*

non-linear

ConvE (Dettmers et al., 2018):

- Performs a global 2D convolution operation on reshaped and concatenated subject entity and relation embedding vectors.
- Impressive results but involves unintuitive reshaping and concatenation.

T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel. ***Convolutional 2D Knowledge Graph Embeddings.*** *2018.*

HypER (Balazevic et al., 2019)

- A simplified convolutional model, that uses a hypernetwork to generate 1D convolutional filters for each relation, extracting relation-specific features from entity embeddings.
- Hard regularization.

I. Balazevic, C. Allen, T. Hospedales. ***Hypernetwork Knowledge Graph Embeddings****. 2019.*

The approach that we use

We introduce embeddings

$$\mathbf{e}_S : S \to \mathbb{R}^{d_s}, \mathbf{e}_R : R \to \mathbb{R}^{d_r}, \mathbf{e}_O : O \to \mathbb{R}^{d_o},$$

which effectively are matrices with $|S| = n_s,\ |R| = n_r,\ |O| = n_o$

$$\mathbf{e}_S \in \mathbb{R}^{d_s \times n_s}, \mathbf{e}_R \in \mathbb{R}^{d_r \times n_r}, \mathbf{e}_O \in \mathbb{R}^{d_o \times n_o}.$$

We complete knowledge graphs by predicting missing relations using **Tucker decomposition** [1] and **Riemann optimization** [2].

[1] – I. Balažević, C. Allen, T. M. Hospedales. TuckER: Tensor Factorization for Knowledge Graph Completion. 2019. https://arxiv.org/pdf/1901.09590.pdf
[2] – B. Vandereycken. Low-rank matrix completion by Riemannian optimization—extended version. 2012. https://arxiv.org/pdf/1209.3834.pdf

The approach that we use

Tucker decomposition: A tensor $X \in \mathbb{R}^{I \times J \times K}$ can be decomposed as

$$X \approx Z \times_1 A \times_2 B \times_3 C,$$

where $Z \in \mathbb{R}^{P \times Q \times R}$ is a core tensor, and $A \in \mathbb{R}^{I \times P}$, $B \in \mathbb{R}^{J \times Q}$, $C \in \mathbb{R}^{K \times R}$ are matrices.

Tucker decomposition for our problem:

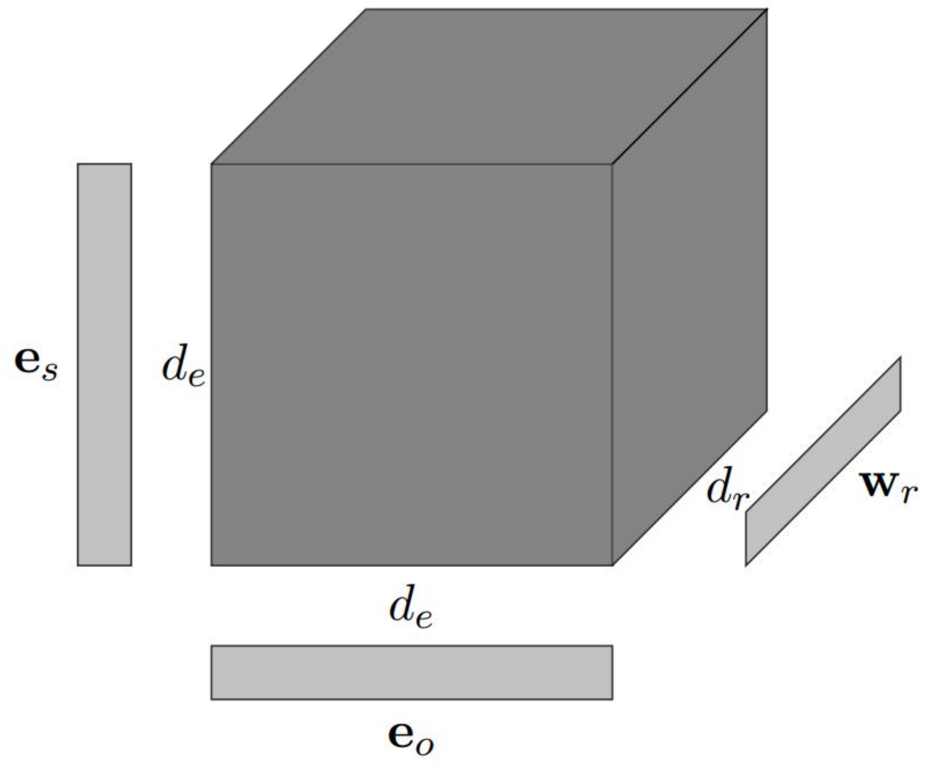
For a knowledge graph tensor $X \in \mathbb{R}^{n_e \times n_r \times n_e}$,

$$X \approx Z \times_1 E \times_2 R \times_3 E \,,$$

where $Z \in \mathbb{R}^{d_e \times d_r \times d_e}$ – core tensor,

$E \in \mathbb{R}^{n_e \times d_e}$ – entity embedding matrix,

$R \in \mathbb{R}^{n_r \times d_r}$ – relation embedding matrix,



n_e, n_r – number of entities and relations, d_e, d_r – dimensionality vectors

[1] – I. Balažević, C. Allen, T. M. Hospedales. TuckER: Tensor Factorization for Knowledge Graph Completion. 2019. https://arxiv.org/pdf/1901.09590.pdf

The approach that we use

Riemann optimization:

For a parameter set Θ we define a scoring function $\varphi : \Theta \times S \times R \times O \to \mathbb{R}$

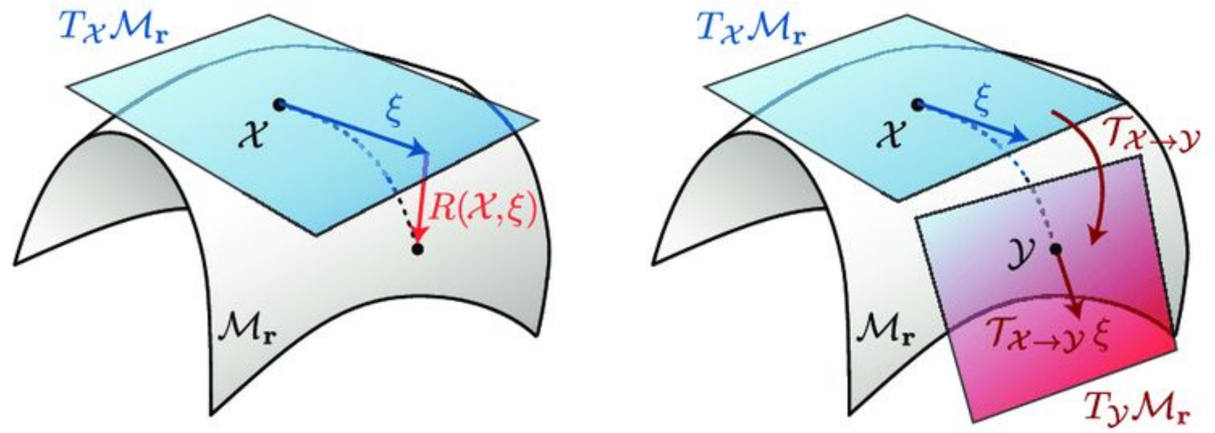
$$\mathcal{L}(\theta, E) = \sum_{(s,*,o)\in E} \sum_{r\in R} L(\varphi(\theta, s, r, o), \mathbb{1}\{(s, r, o) \in E\}) \to \min_{\theta\in\Theta}$$

Note that due to high order SVD (HOSVD) $W, \mathbf{e}_S, \mathbf{e}_R, \mathbf{e}_O$ locally parametrize the sub-manifold $\mathcal{M}_t$ of tensors of multilinear rank $t = (d_s, d_r, d_o)$. It is know than the Riemann stucture makes the optimization feasible.
So the iterate $x_k \in M_t$ and for $x_k \sim W_k \times_1 \mathbf{e}_{S,k} \times_2 \mathbf{e}_{R,k} \times_3 \mathbf{e}_{O,k}$ we define

$$f(x_k) := \mathcal{L}(W_k, \mathbf{e}_{S,k}, \mathbf{e}_{R,k}, \mathbf{e}_{O,k}, E),$$

Recall the vector transport and the retraction map

$$T_{a\to b} : T_a\mathcal{M}_t \to T_b\mathcal{M}_t$$
$$R : \mathbb{R}^{|S||R||O|} \to \mathcal{M}_t,$$



$T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}$
$\mathcal{X}$
ξ
$R(\mathcal{X},\xi)$
$\mathcal{M}_{\mathbf{r}}$
$T_{\mathcal{X}}\mathcal{M}_{\mathbf{r}}$
$\mathcal{X}$
ξ
$\mathcal{T}_{\mathcal{X}\to\mathcal{Y}}$
$\mathcal{Y}$
$\mathcal{M}_{\mathbf{r}}$
$\mathcal{T}_{\mathcal{X}\to\mathcal{Y}}\xi$
$T_{\mathcal{Y}}\mathcal{M}_{\mathbf{r}}$

constructed via the Riemann gradient of the contraction of tensor representaion of $v \in T_a\mathcal{M}_t$ with b for the vector transport and via HOSVD for the retraction map.

[2] – B. Vandereycken. Low-rank matrix completion by Riemannian optimization—extended version. 2012. https://arxiv.org/pdf/1209.3834.pdf

The approach that we use

In our project, as a base, we use the github **R-TuckER: Riemann optimization on manifolds of tensors for knowledge graph completion [1]** with the PyTorch implementation of Tucker model for knowledge graph link prediction task using:

- RGD,
- RSGD with Momentum,
- Adam

with Riemann optimization tools from github **Tucker Riemopt [2]**, in particular:

- Riemann gradient
- Vector transport
- Retraction map

[1] – https://github.com/johanDDC/R-TuckER/tree/master
[2] – https://github.com/johanDDC/tucker_riemopt/tree/master

Proposed solution

Proposed solution (1)

In our solution for knowledge graph completion problem, we **generalize and implement the following optimization methods** with Riemann gradient:

- RMSprop
- Nesterov momentum SGD
- AdamW
- AdaDelta

Loss:
We **add smooth L1 loss** following by Robust Low-Rank Matrix Completion by Riemannian Optimization

Project Github:

https://github.com/marina-shesha/NLA_project

Skoltech

Proposed solution (2)

RMSprop

RMSProp-like

$$\sigma_k^2 \leftarrow \beta\sigma_{k-1}^2 + (1-\beta)\|\nabla f(x_k)\|^2$$
$$x_k \leftarrow x_k - \frac{\gamma}{\sqrt{\sigma_k^2+\varepsilon}}\nabla f(x_k)$$
$$x_k \leftarrow R(x_k)$$

Nesterov momentum SGD

Nesterov momentum GD step

$$p_k \leftarrow \beta p_{k-1} + \nabla f(R(x_k + T_{x_{k-1}\to x_k}(\beta p_{k-1})))$$
$$x_k \leftarrow x_{k-1} - \gamma p_k$$
$$x_k \leftarrow R(x_k)$$
$$p_k \leftarrow T_{x_{k-1}\to x_k}(p_k)$$

Skoltech

Proposed solution (3)

AdamW

AdamW-like

$$
\begin{aligned}
p_k &\leftarrow \alpha T_{x_{k-1}\to x_k}(p_{k-1}) + (1-\alpha)\nabla f(x_k)\\
\sigma_k^2 &\leftarrow \beta\sigma_{k-1}^2 + (1-\beta)\|\nabla f(x_k)\|^2\\
\sigma_c^2 &\leftarrow \frac{\sigma_k^2}{(1-\beta)^{\frac{k}{v}}}\\
b_c &\leftarrow (1-\alpha^{\frac{k}{v}})\sqrt{\sigma_c^2} + \varepsilon\\
x_k &\leftarrow (1-\lambda)x_{k-1} - \frac{\gamma}{b_c}\nabla f(x_k)\\
x_k &\leftarrow R(x_k)
\end{aligned}
$$

AdaDelta

AdaDelta-like

$$
\begin{aligned}
g_k &\leftarrow \nabla f(x_k)\\
v_k &\leftarrow \beta v_{k-1} + (1-\beta)g_k^2\\
p_k &\leftarrow T_{x_k}\left(\frac{\sqrt{u_{k-1}+\varepsilon}}{\sqrt{v_k+\varepsilon}}g_k\right)\\
u_k &\leftarrow \alpha u_{k-1} + (1-\alpha)p_k^2\\
x_k &\leftarrow x_{k-1} - \gamma p_k\\
x_k &\leftarrow R(x_k)
\end{aligned}
$$

Skoltech

Datasets

We use the following datasets to evaluate smooth L1 and compare basic Riemannian SGD and our Riemannian optimizers RMSprop, SGD with Nesterov momentum, and AdamW:

FB15k

- is a subset of Freebase, a large database of real world facts
- 592,213 triplets with 14,951 entities and 1,345 relationships

FB15k-237

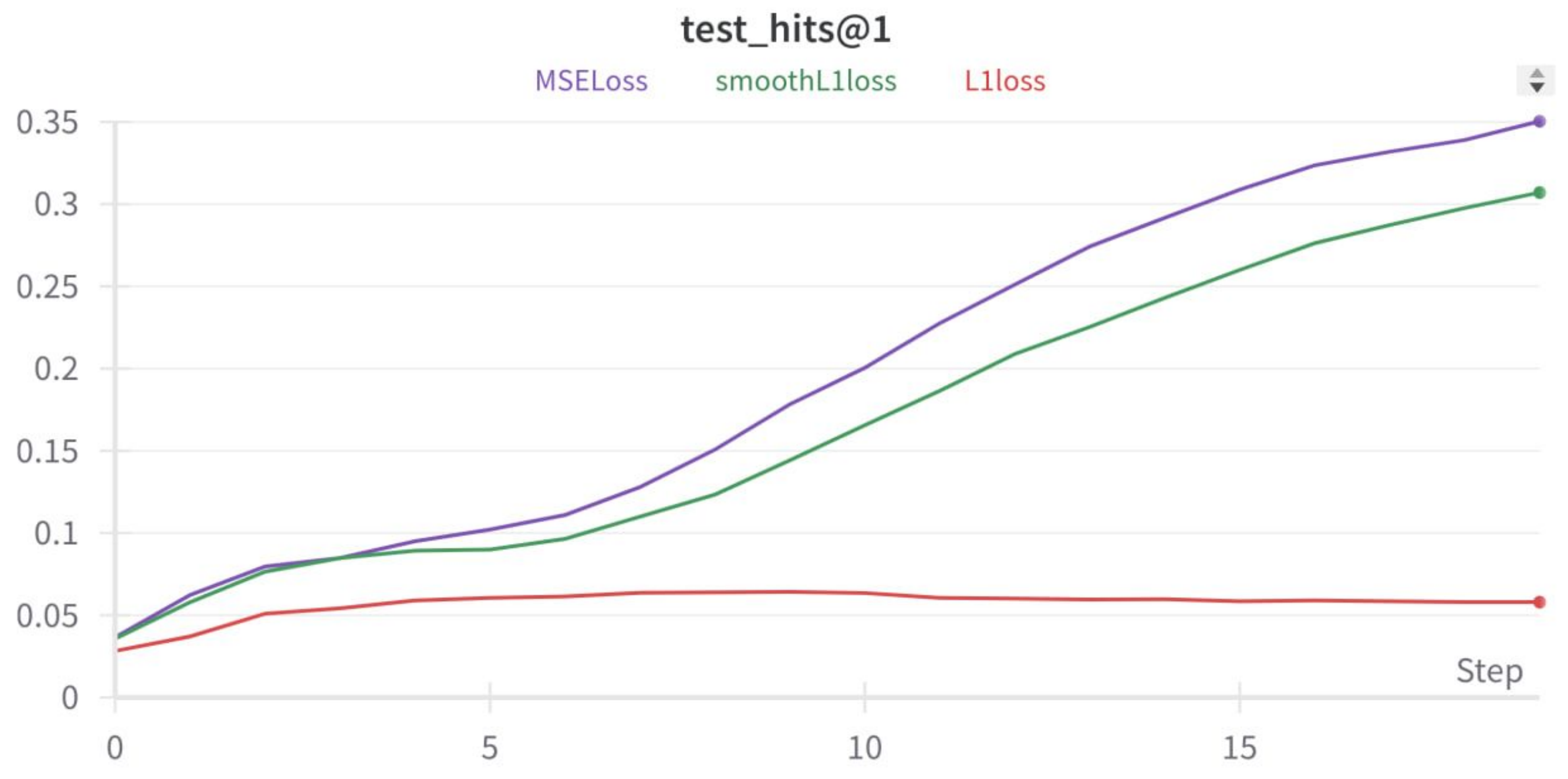
- was created from FB15k by removing the inverse of many relations that are present in the training set from validation and test sets, making it more difficult for simple models to do well
- 310,116 triplets with 14,541 entities and 237 relation types

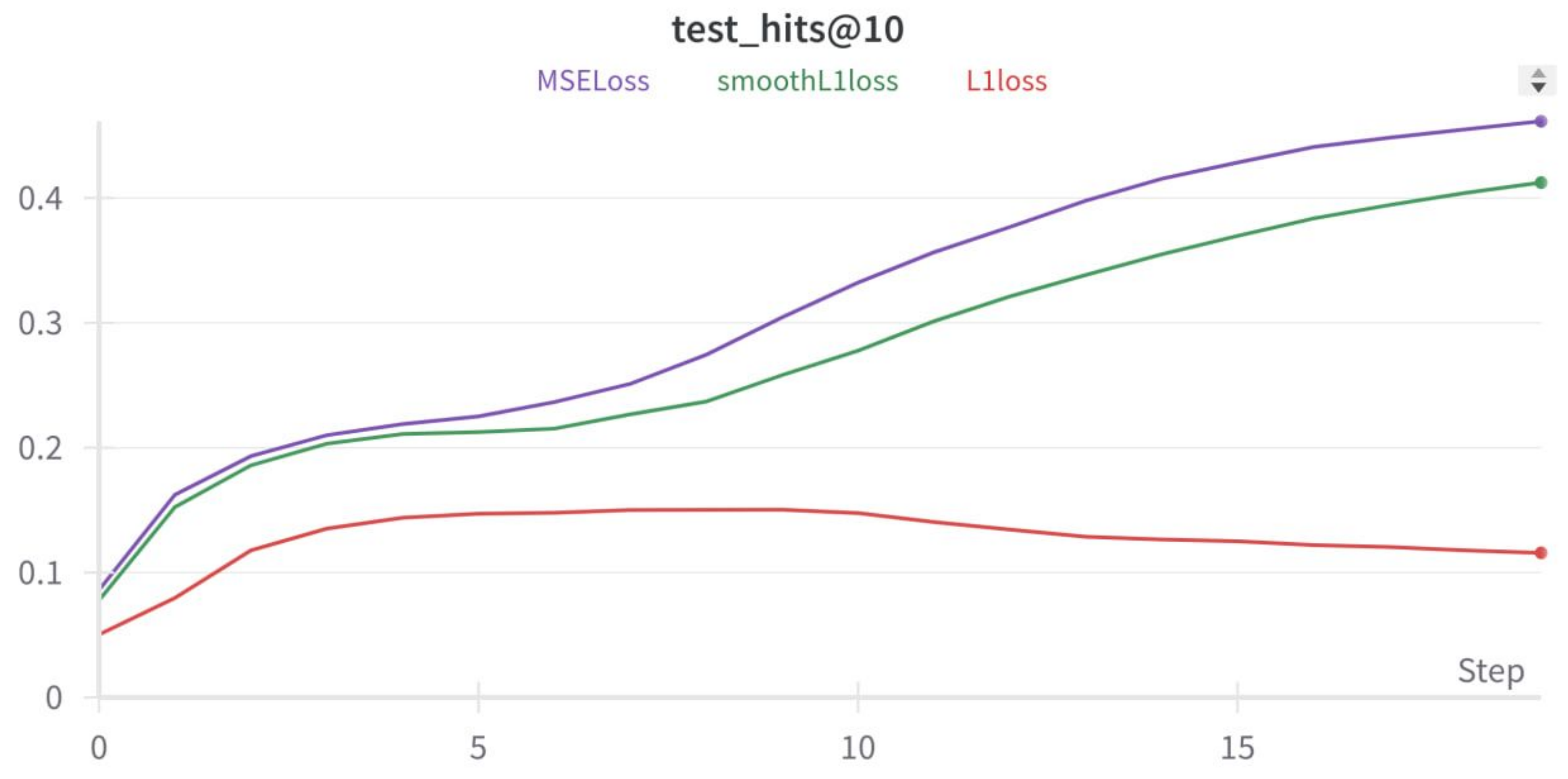
Skoltech

Proposed solution (4)

Loss We add smooth L1 loss [1]: $$f_\delta(x) = \sum_i \sqrt{\delta^2 + (x_i - y_i)^2}$$

FB15k





Results are available in Wandb

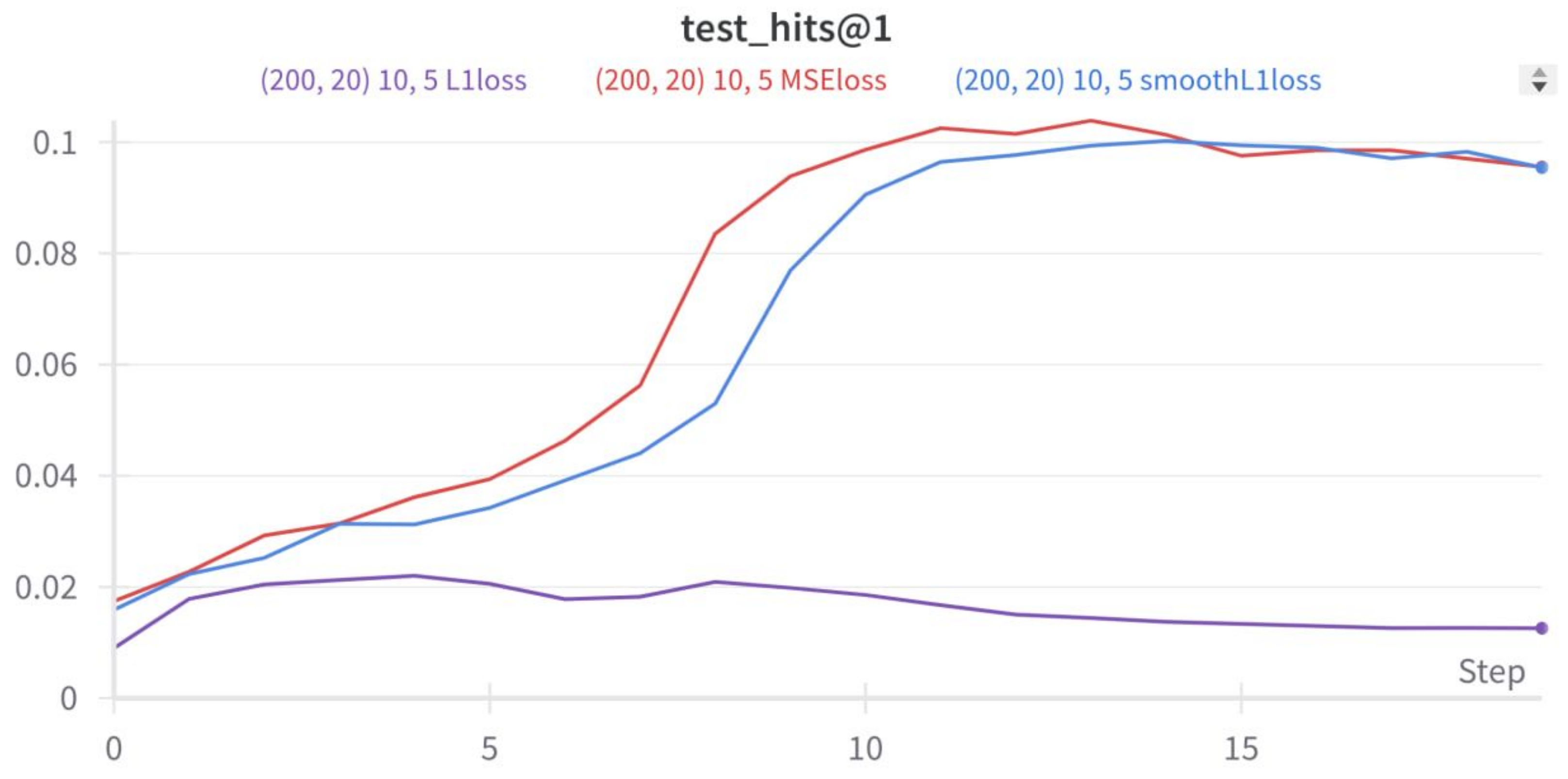
[1] – L. Cambier, P.-A. Absil. Robust Low-Rank Matrix Completion by Riemannian Optimization, 2016

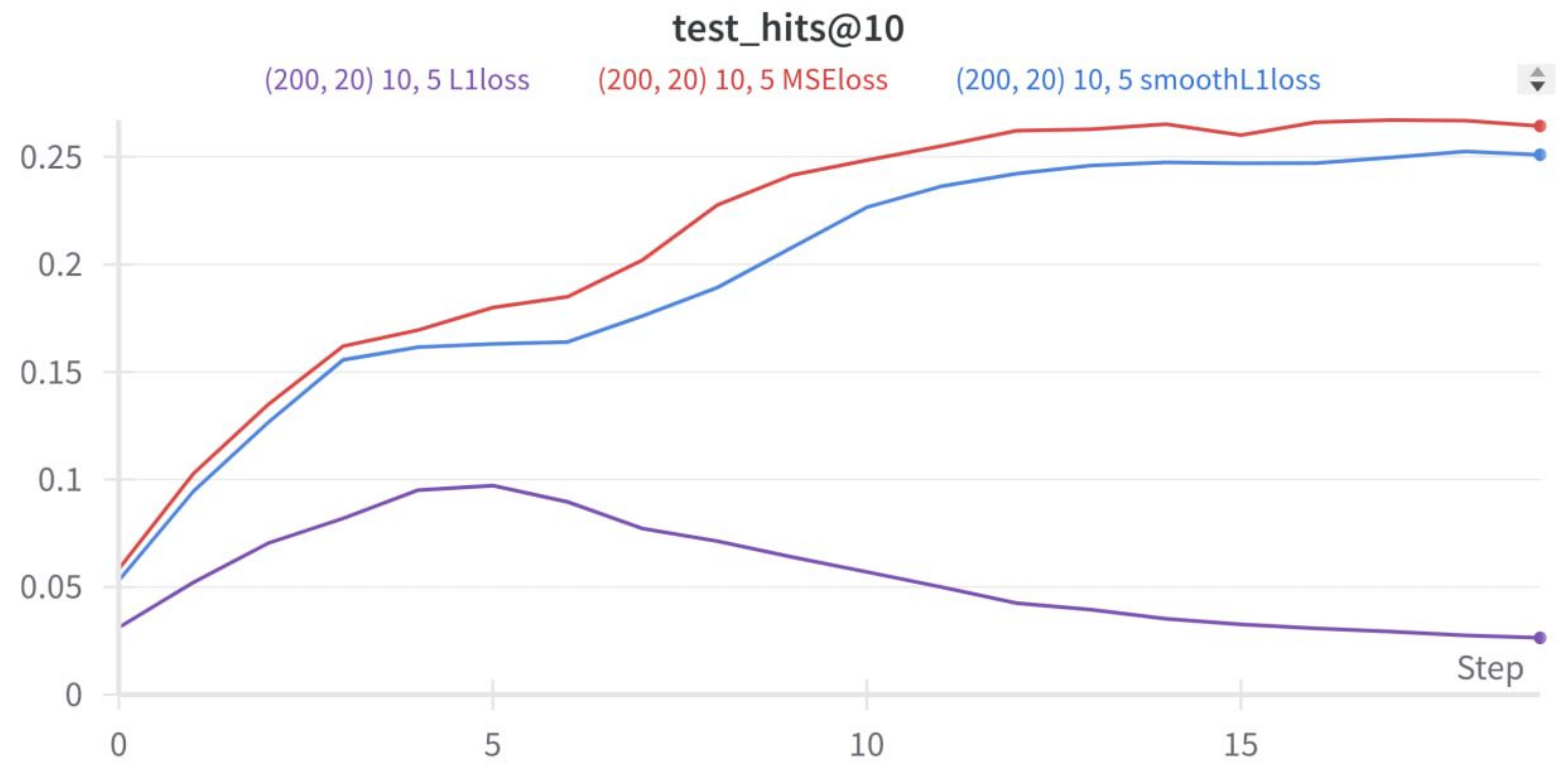
Skoltech

Proposed solution (5)

Loss We add smooth L1 loss [1]: $f_\delta(x) = \sum_i \sqrt{\delta^2 + (x_i - y_i)^2}$

FB15k-237





Results are available in Wandb

[1] – L. Cambier, P.-A. Absil. Robust Low-Rank Matrix Completion by Riemannian Optimization, 2016

Skoltech

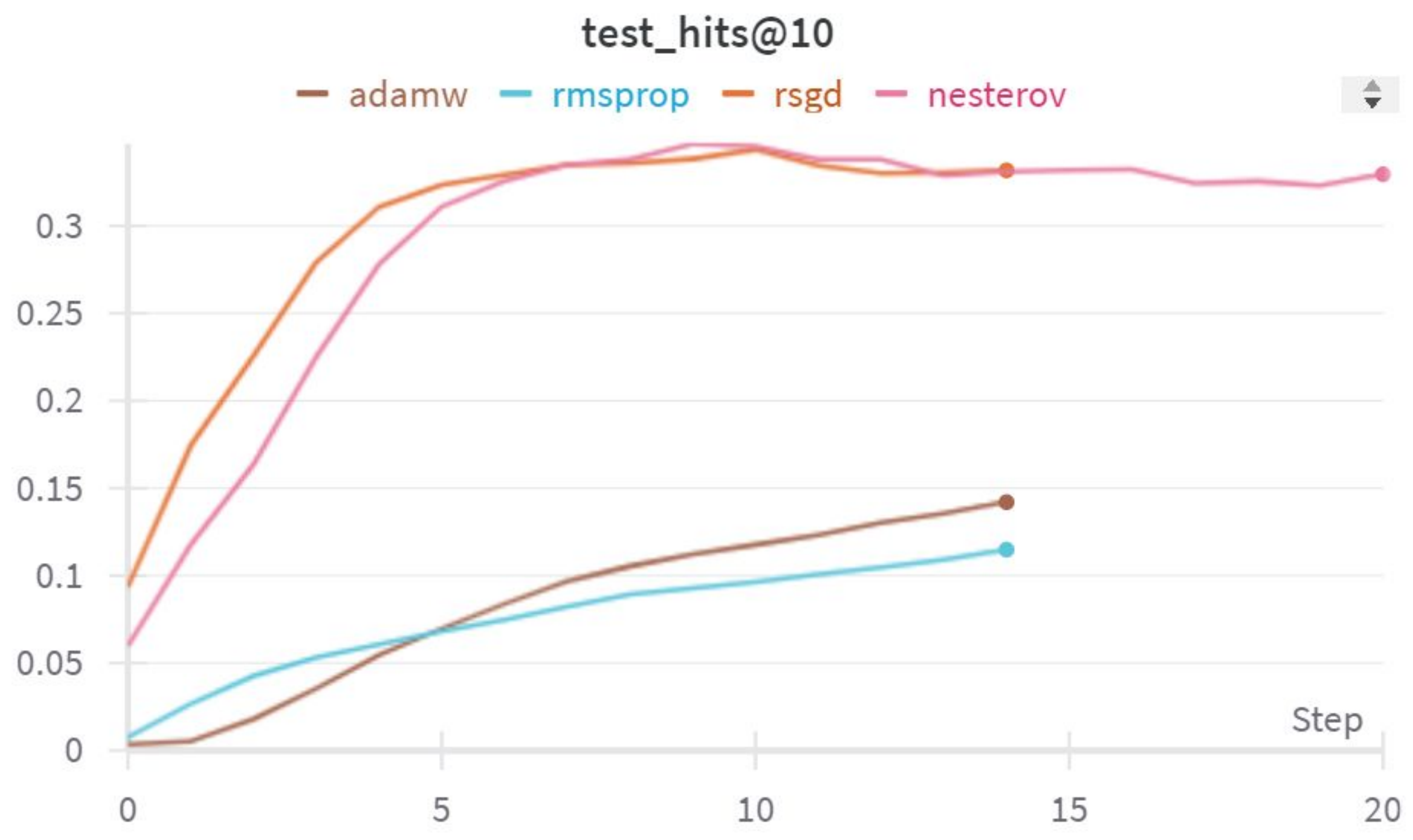
04

Results comparison

Skoltech

Results comparison (1)

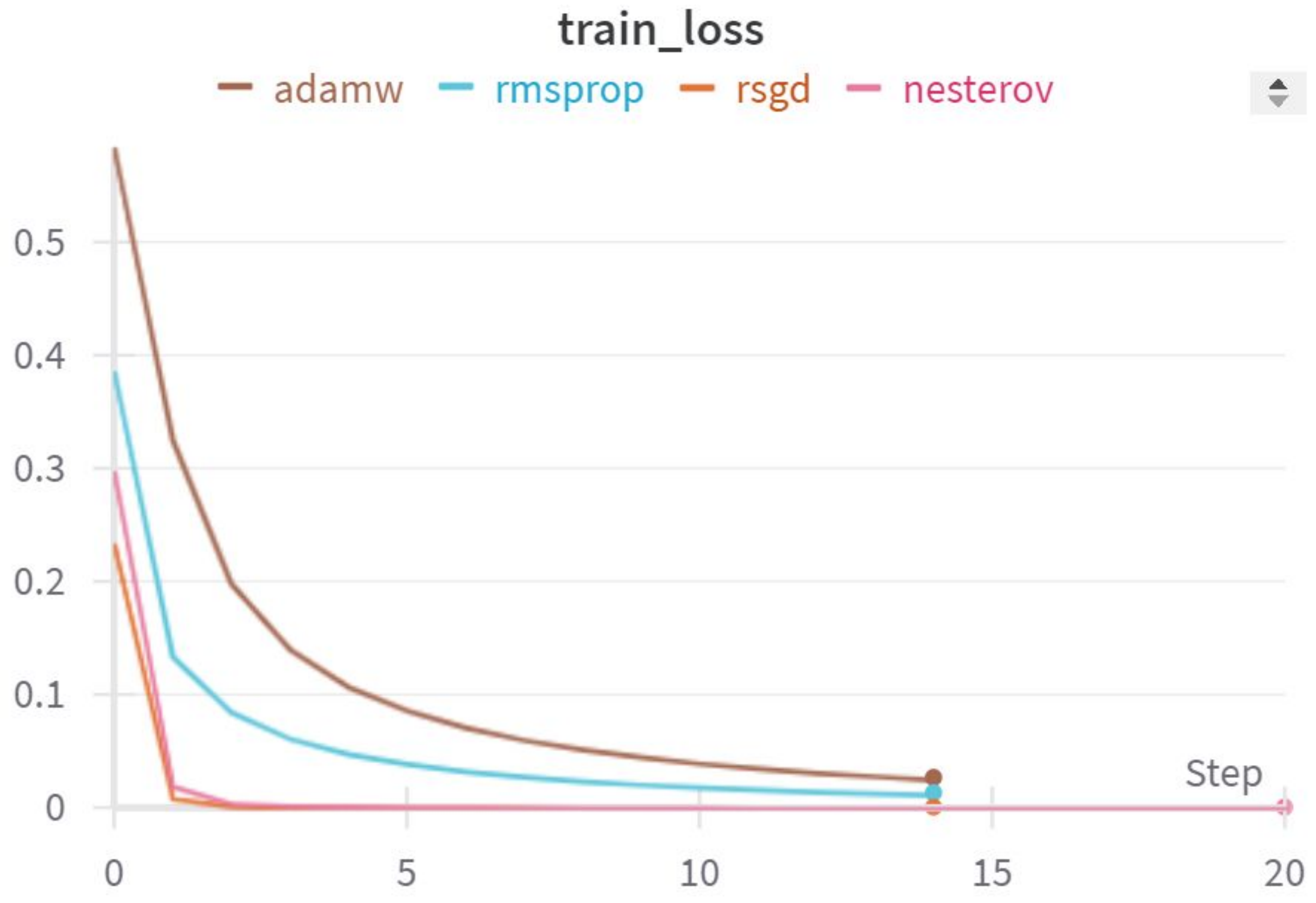
Hits@10 on test:



	test_hits@10
RSGD	0.3317
Nesterov	0.331
AdamW	0.142
RMSprop	0.1148

Results comparison (2)

Loss on train:



	train_loss (BCE)
RSGD	0.0004
Nesterov	0.0005
AdamW	0.02667
RMSprop	0.01318

Skoltech

Conclusions

- Trivial RGD and RSGD turned out to be more efficient and cheap in our datasets.
- Optimization methods that employ coordinate-wise operations are non-trivial for the Riemannian optimization because it is important to preserve the geometry.
- AdaDelta requires a lot of GPU RAM.
- Smoothness of loss function plays critical role. Quality with the vanilla L1 loss is inferior to its smoothed version.
- Outlier-robust smoothed L1 is conjectured to beat L2, which is also typically used in completion tasks.

Contribution of team members

Contribution of team members

Kamil Mardanshin

- Optimization problem statement
- AdaDelta
- Git management
- Presentation

Marina Sheshukova

- RMSprop
- Smooth L1 loss
- Experiments with losses
- Presentation

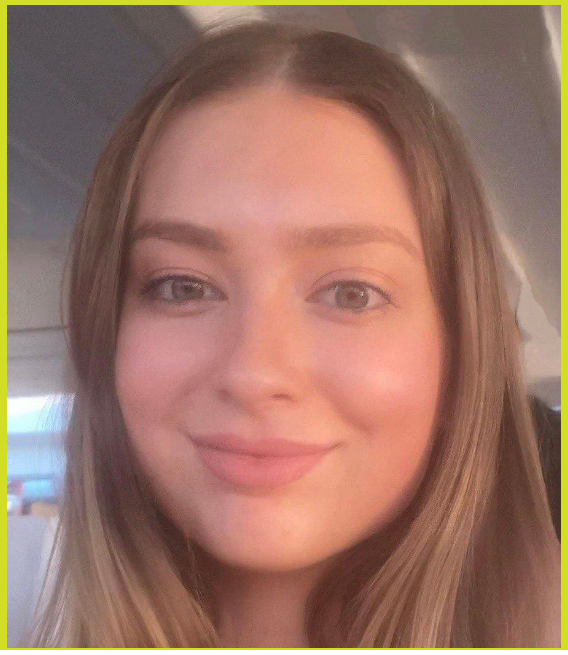
Ekaterina Filimoshina

- Nesterov momentum
- Existing approaches consideration
- Presentation

Ignat Romanov

- AdamW
- Repo formatting
- Presentation

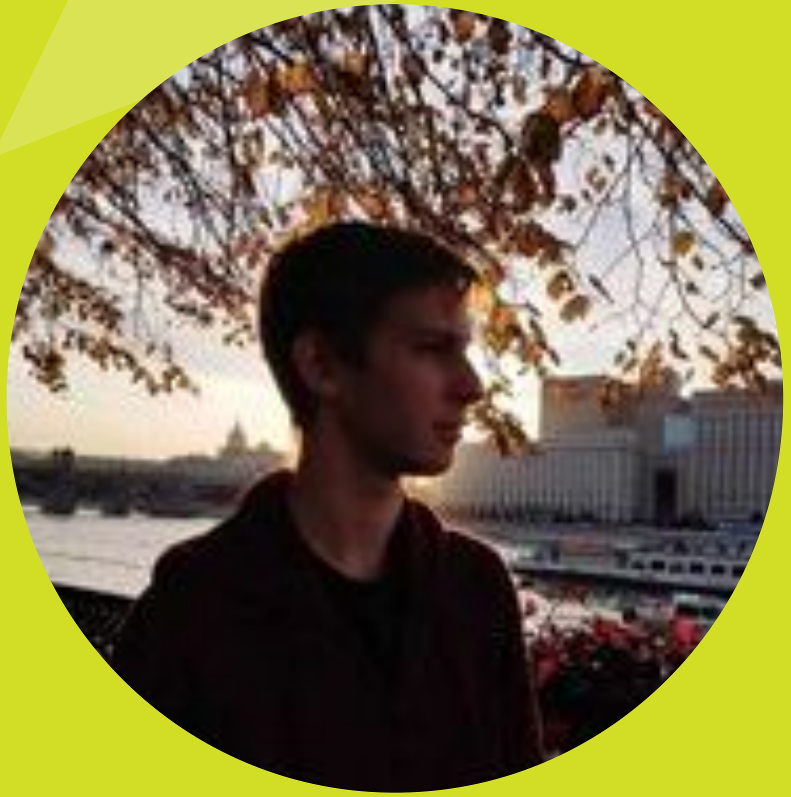
Ilona Basset



- Notebook with models tests
- Results comparison
- Presentation

Skoltech

Our Advisors



Maxim Rakhuba

Associate Professor, Senior Research Fellow and Laboratory Head at HSE's Faculty of Computer Science

Ivan Peshekhonov

Visiting Lecturer at HSE's Faculty of Computer Science

Skoltech

Thx

Skoltech