9th International Conference "Distributed Computing and Grid Technologies in Science and Education" (GRID'2021),
Dubna, Russia, July 5-9 2021

Using distributed computing systems to solve the problem of image classification using deep neural networks

Ilya I. Kurochkin

IITP RAS, Moscow, Russia

Использование распределенных вычислительных систем для решения задачи классификации изображений с помощью глубоких нейронных сетей

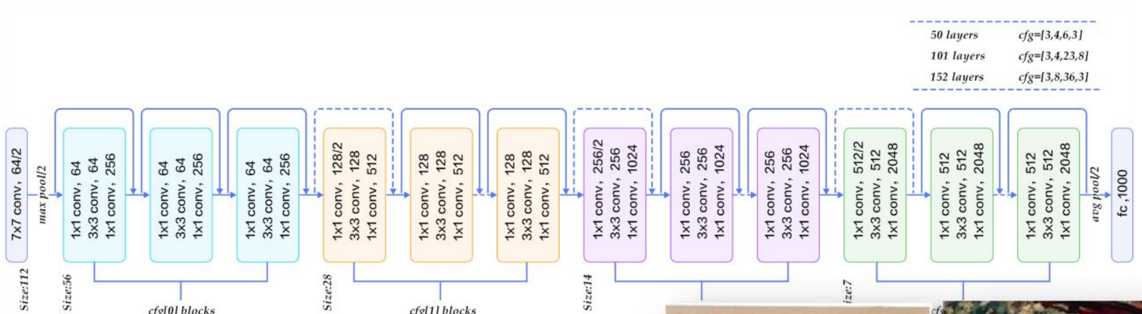
Курочкин И.И.

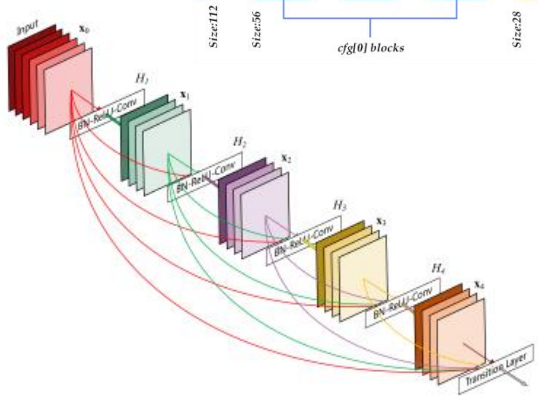
ЦРВ ИППИ РАН, Москва, Россия

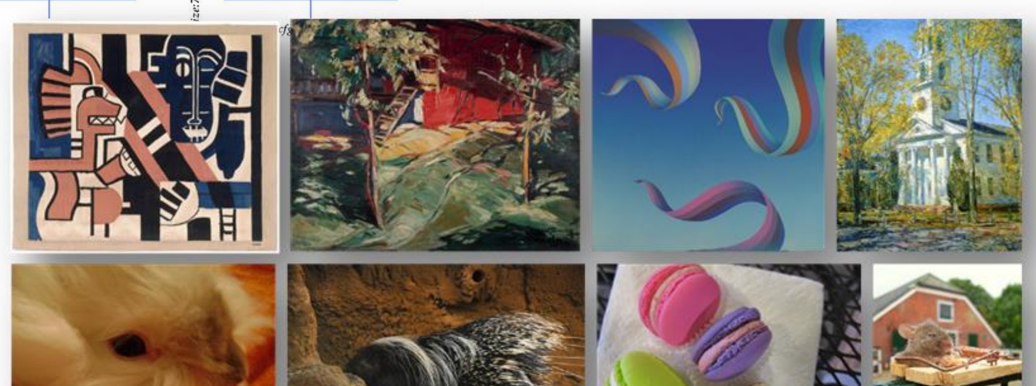
This work was funded by RFBR according to the research projects No. 18-29-03264 and No. 19-07-00802

Deep neural networks

- The tendency to increase the depth of neural networks (from 8 layers for LeNet to 152 for ResNet)
- A huge number of customizable parameters (ResNet-152-60.2 million)
- Operating with large data arrays (ImageNet – 14.5 million images)

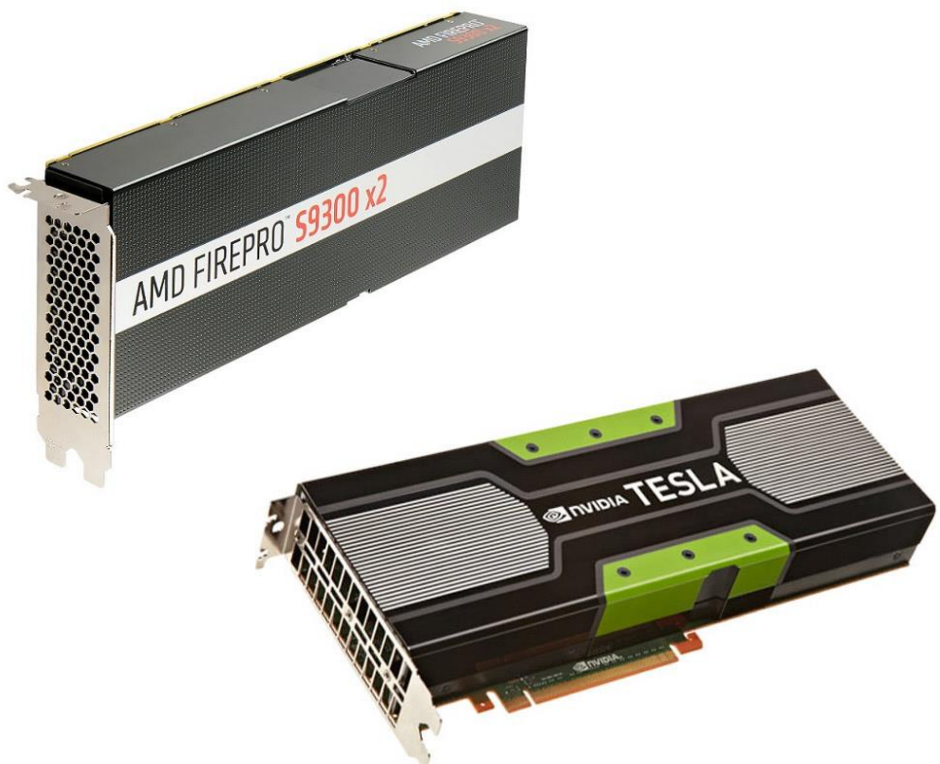


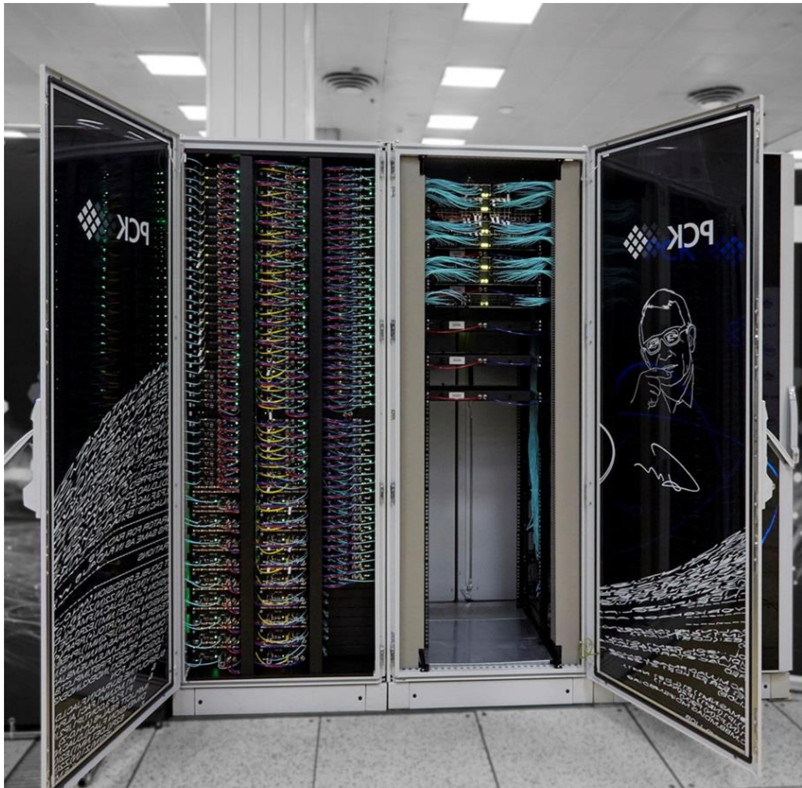




Deep learning bottlenecks

- Large dataset size
- The need to use the entire dataset during training
- Neural network training time



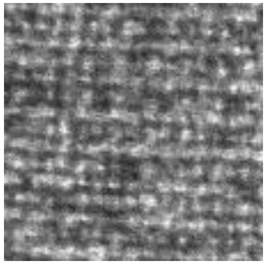


Texture Image Datasets

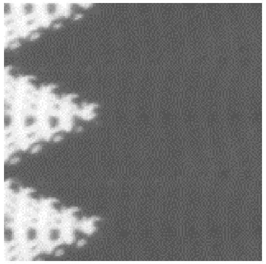
A dataset is processed and structured information in a tabular form.

Brodatz (112 classes, 1 channel, 512x512 dots)
KTH-TIPS (11 classes of 81 images, 3 channels, 200x200 dots)
KTH-TIPS 2 (11 classes of 108 images, 3 channels, 200x200 dots)
Used in the work – The Kylberg Texture Dataset.
28 classes of 160 images per class, 1 channel, 576x576 pixels

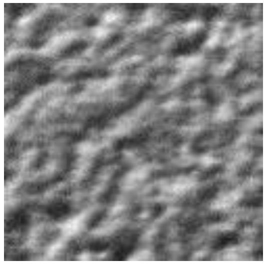
Image examples:



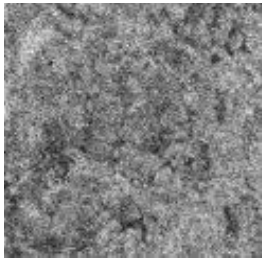
Class 1



Class 2



Class 3



Class 4

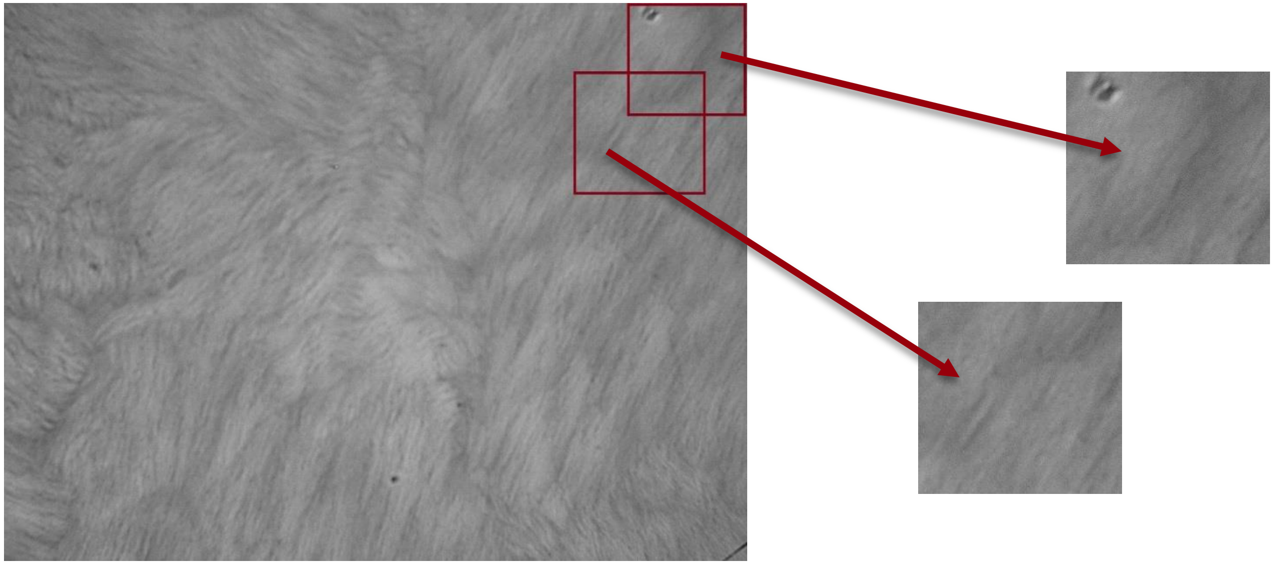
Classification images of PVA cryogels

Set of images:

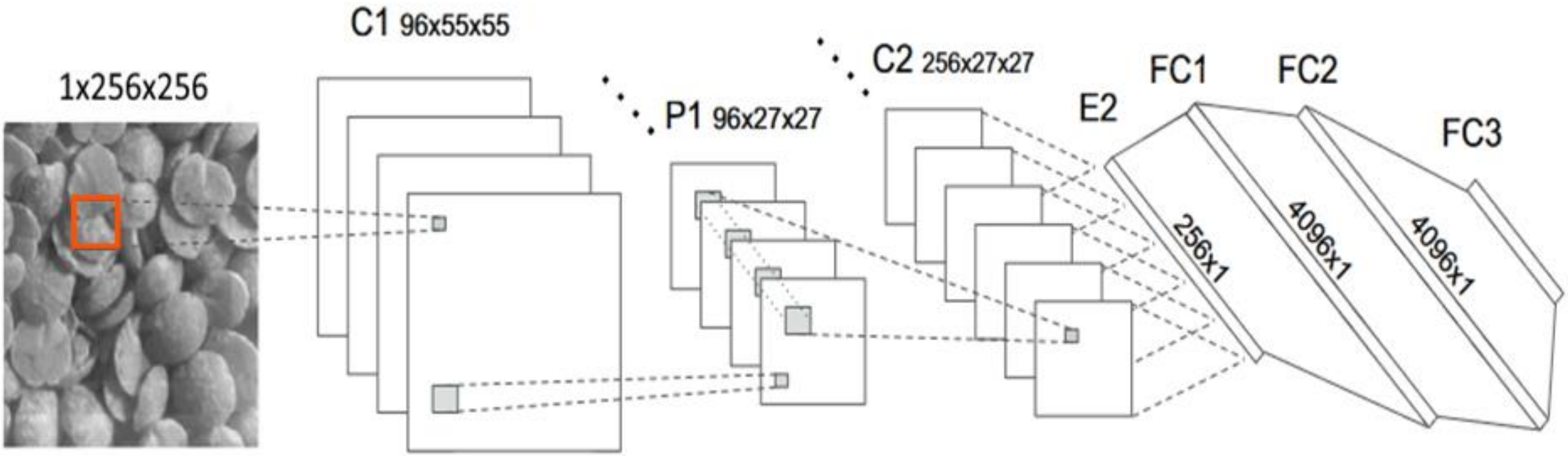
- 654 images (2272 x 1074 pixels)
- 3 classes

Dataset:

- 6972 fragments for each image (256 x 256 pixels)
- 654 x 6972 ≈ **4.55 million image fragments**

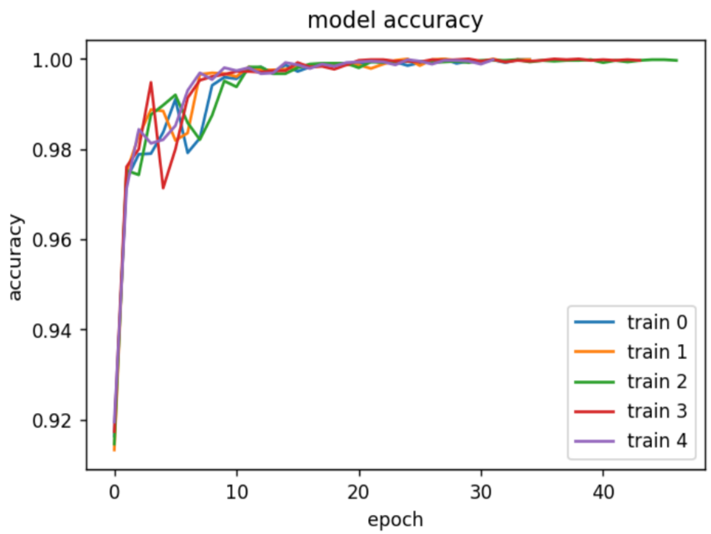


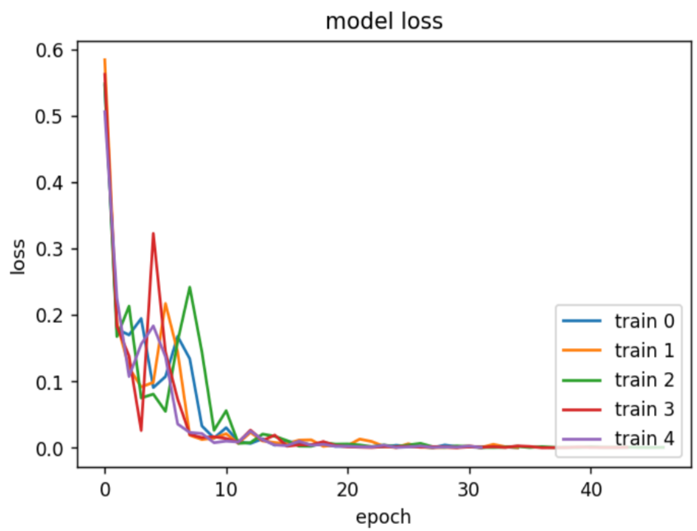
Texture-CNN-2



Andrearczyk, Vincent, and Paul F. Whelan.
"Using filter banks in convolutional neural networks for texture classification."
Pattern Recognition Letters 84 (2016): 63-69.
https://arxiv.org/pdf/1601.02919.pdf

Testing on a reference Kylberg dataset





12 classes

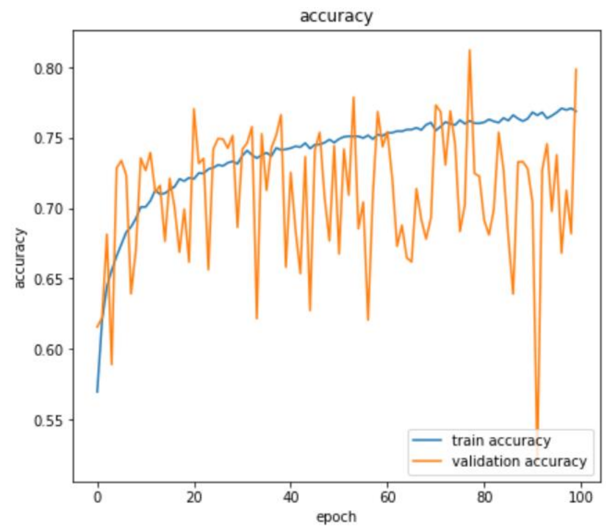
```
Cross-validation (5-fold) results
Accuracy: 100.00% (+/- 0.00%)
Loss: 0.01% (+/- 0.01%)
```

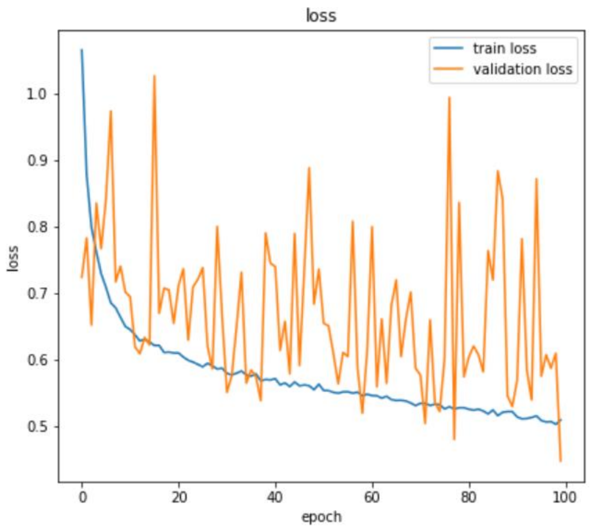
28 classes

	Kylberg
T-CNN-1 (20.8)	89.5 ± 1.0
T-CNN-2 (22.1)	$\mathbf{99.2 \pm 0.3}$

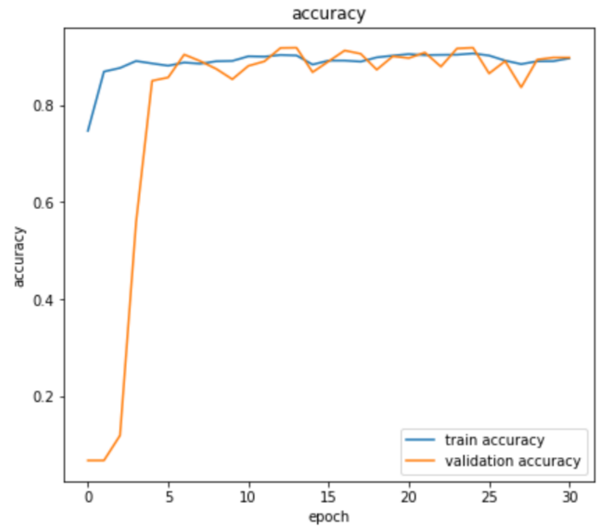
Testing on a PVACG dataset

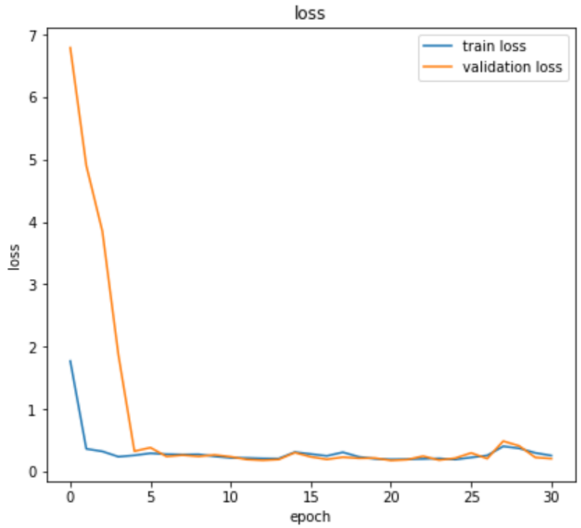
- 3 classes PVACG
- 16 000 fragments per class for train
- 4 800 fragments per class for test





- 3 classes PVACG + 12 Kylberg
- 480 fragments per class for train
- 80 fragments per class for test





Bag of tasks

The task can be divided into many independent subtasks. Each subtask will be calculated on a separate computing node of the distributed system.

For different subtasks, different sets of input data and a single algorithm for processing them are used. This type of task in the literature is called "bag of tasks".

As an example of such tasks, we can give the following tasks:

- Combinatorial problems and complete search;
- SAT problems;
- Machine learning;
- Simulation mathematical modeling;

and etc.

Desktop grid features

- Heterogeneity of nodes;
- The autonomy of calculations at different nodes and the impossibility of constant coordination of calculations between nodes;
- Unreliability of connections and possible disconnection of computing nodes;
- Unstable time of continuous operation of the node and the difficulty of calculating long-term tasks;
- The presence of errors and delays in calculations.

Software platforms for organizing distributed computing

- HTCondor
- Globus
- BOINC
- Oracle Grid Engine

BOINC software

BOINC – Berkeley Open Infrastructure for Network Computing

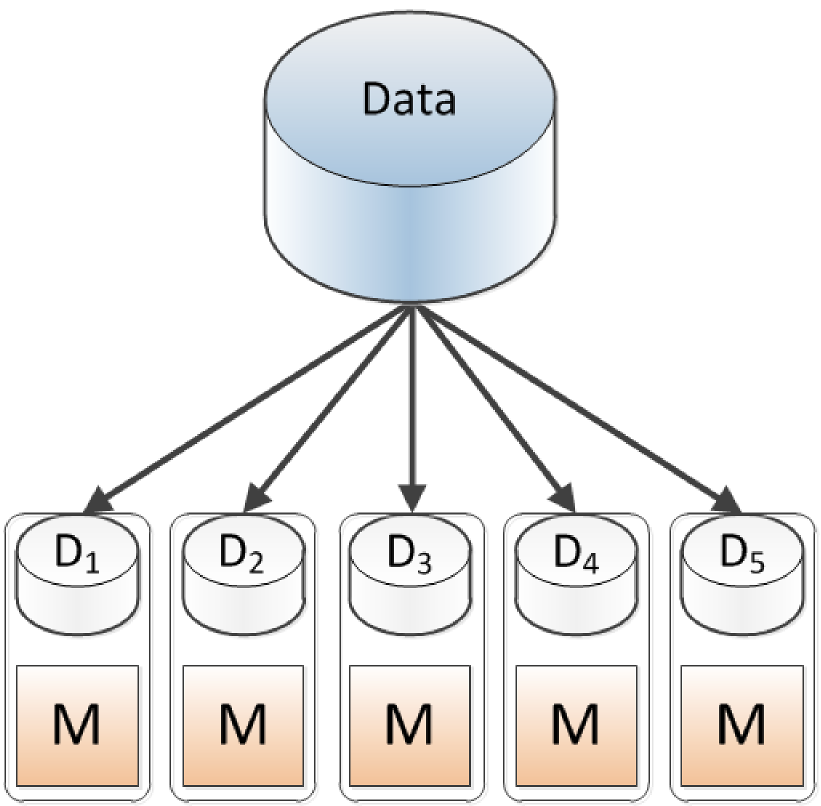
- A platform for organizing voluntary distributed computing:
- it consists of a server and a client part;
- makes it possible to use the computing power of personal computers (PCs) and other personal devices;
- Cross-platform client part;
- Flexible configuration of the client part for efficient use of PC resources.

Distributed learning parameters

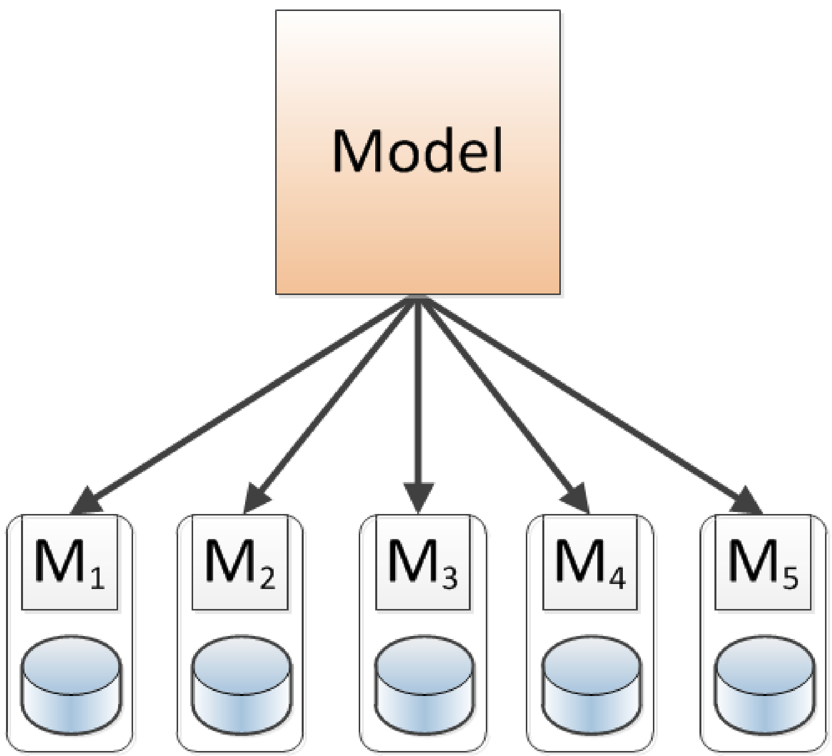
- Distributed learning method with data parallelism;
- The distributed learning process is iterative with synchronous updating of neural network parameters;
- During the execution of one iteration of distributed learning on a computing node, the neural network model is trained for several epochs;
- When forming a training set on a computing node, it is proposed to use a certain number of random images. For a desktop grid system the local training set is 4500-6000 images;
- To eliminate the disadvantages of synchronous updating and reduce the downtime of nodes, the concept of a minimum quorum (minimum number of results) for completing an iteration (75% -90%.)
- is introduced. The formation of tasks for the next iteration is based on the current parameters of the global model.
- The results that came to the server after the aggregation of the iteration results are not taken into account and are discarded.

Distributed learning. 2 approaches

Separation by data

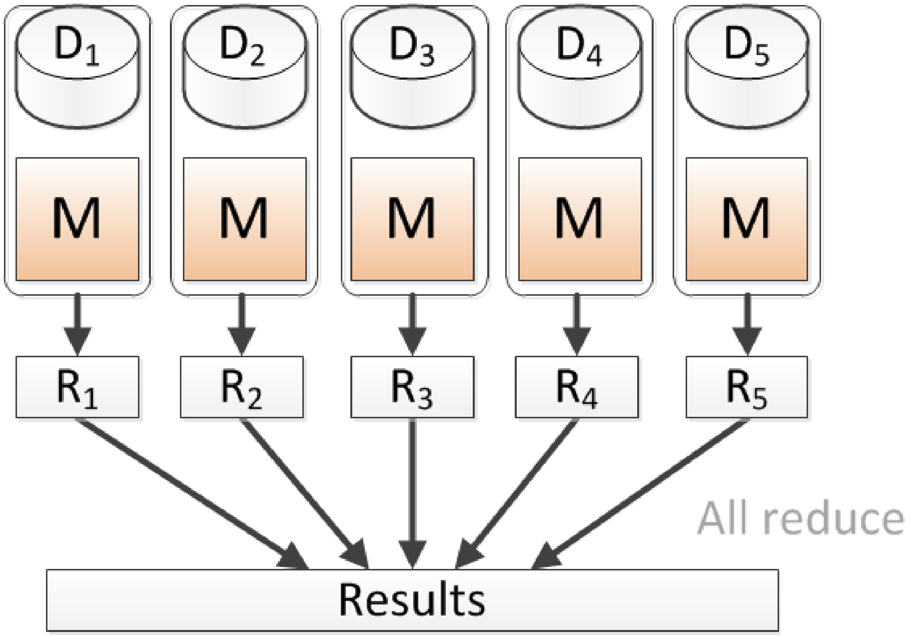


Separation by model

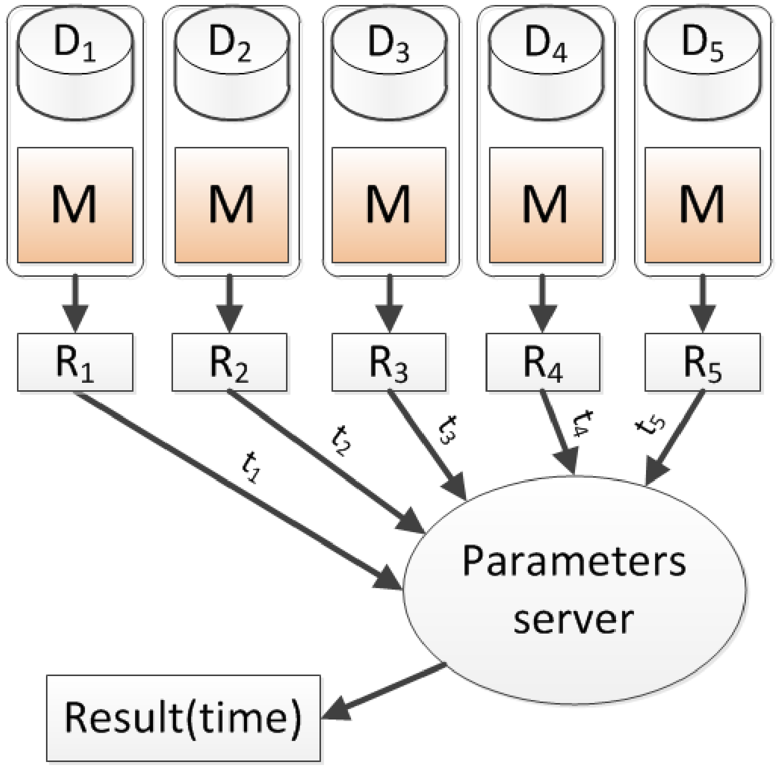


Separation by data approach

Synchronous approach



Asynchronous approach



Asynchronous and synchronous approaches

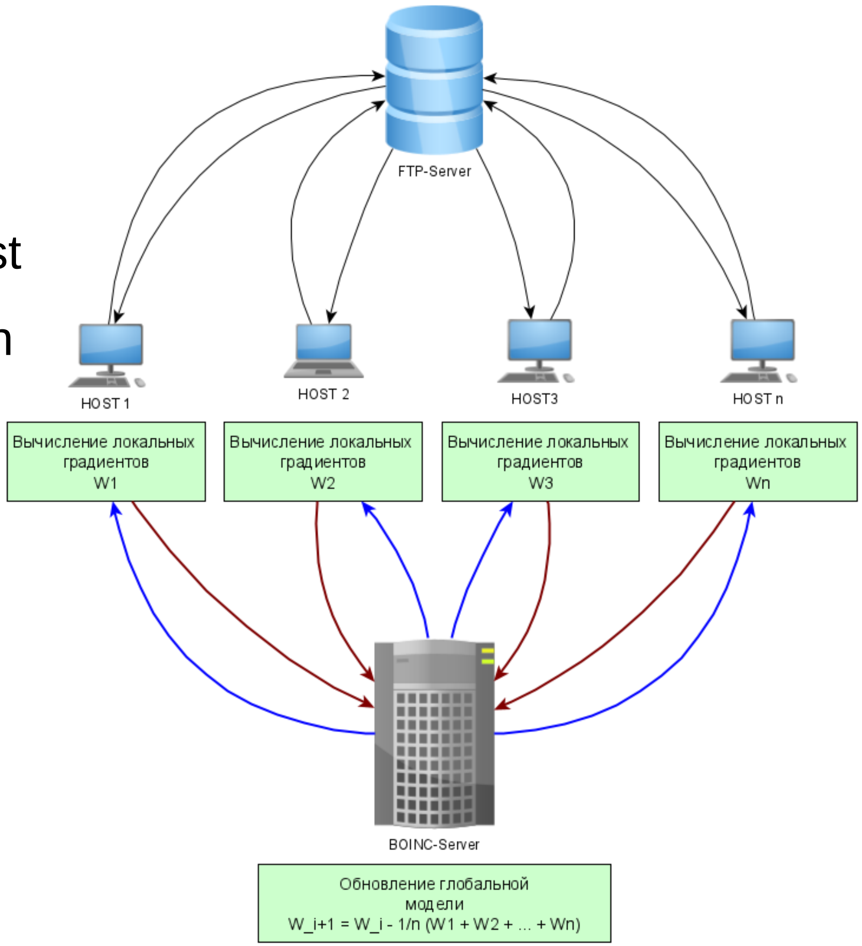
In synchronous approach, the replicas of the model are synchronized every computational cycle.
The advantage is the relevance of the resulting gradients, which guarantees better convergence.
The disadvantage is global/general synchronization, which creates the problem of lagging performers and seriously reduces its effectiveness in heterogeneous systems or systems with failures. It is also worth noting that general synchronization has significant communication costs.

In asynchronous approach, replicas are synchronized independently, which solves the problem of the synchronous case, communication is more uniform throughout the entire learning process.
But there is a problem of obsolescence of gradients, since during their calculation, local models are continuously updated, which can significantly affect the convergence of training.

Parameters for the experiment (synchronous approach)

Parameters of the computational experiment:

- 12-15 hosts
- One iteration: 10 training epochs on each host
- each host downloads 6000 fragments from an FTP-server for local irradiation
- 18 iterations of updating the global model
- Average execution time
- The computational task on one host takes about 3 hours.



Updating the model

The formula for updating weights for simple stochastic descent (SGD):

$$W_{i+1} = W_i - learning\ rate * Grad_i$$

W – the value of the model parameters;

i – is the iteration number of the parameter update;

$learning\ rate$ – the learning rate of the model;

$Grad$ – the value of the calculated gradients.

Changing the parameters of the local trained model:

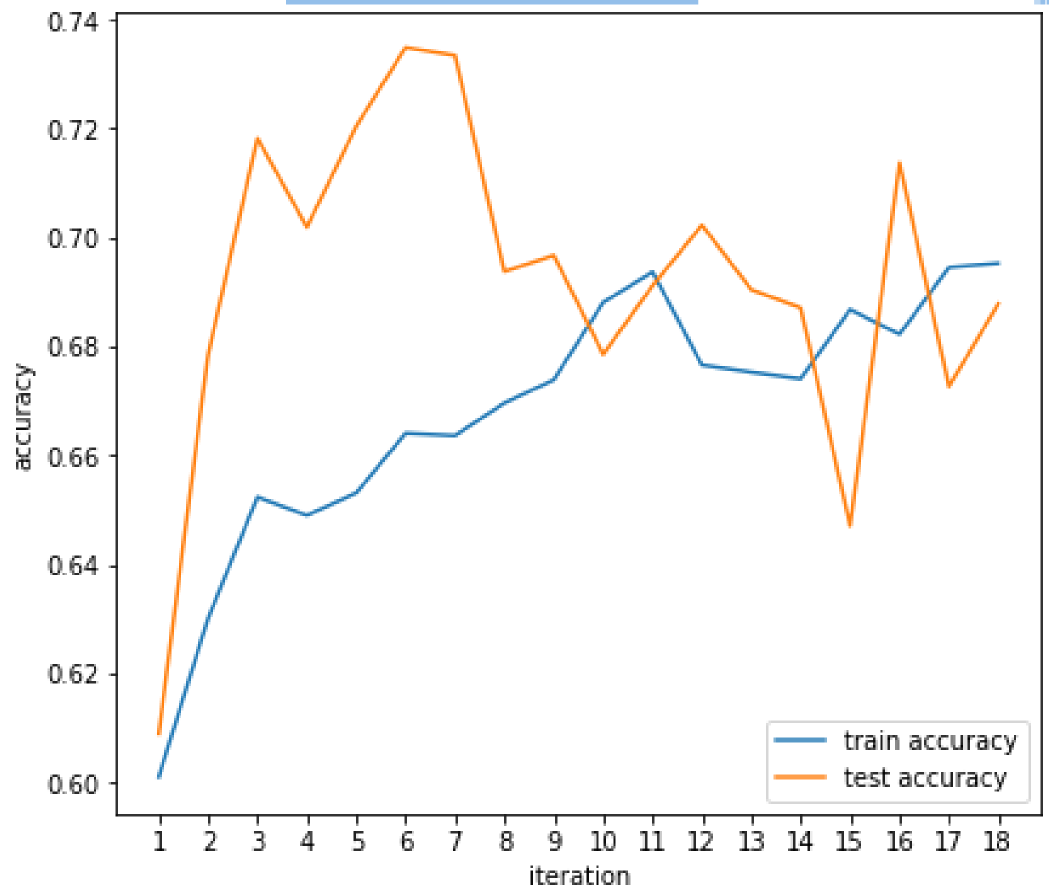
$$learning\ rate * Grad_i = W_i - W_{i+1}$$

Updated global model parameters:

$$W_{i+1}^{global} = W_i^{global} - \frac{1}{n}\sum_{j=1}^{n}(learning\ rate * Grad_i)$$

n – is the number of locally trained models.

Results (sync.ap.)

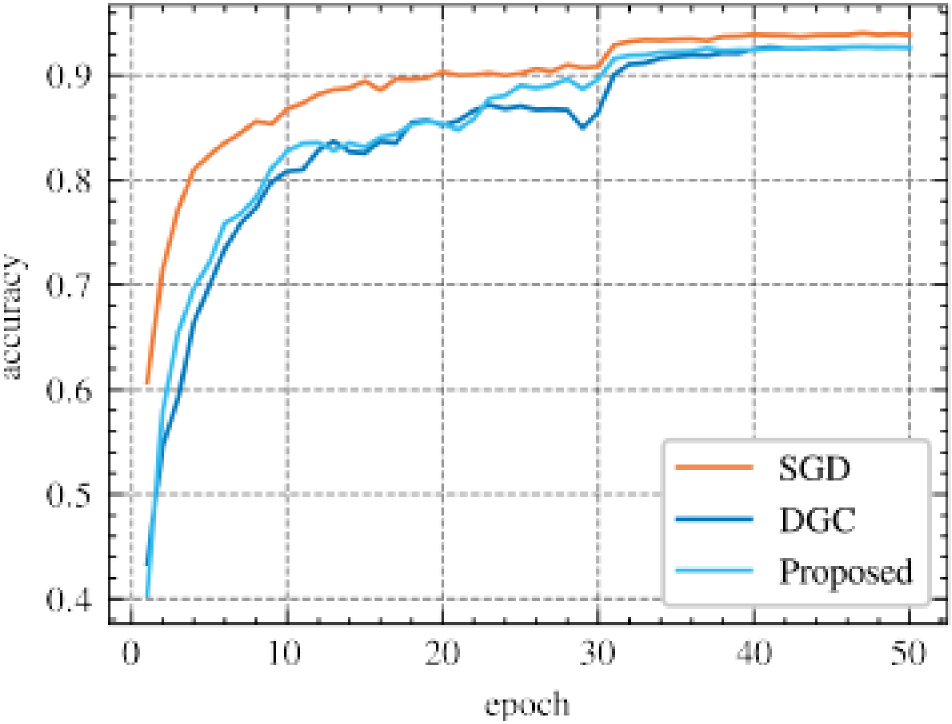


Name	**Class 1**	**Class 2**	**Class 3**
Accuracy	0.874	0.756	0.664
Precision	0.81	0.65	0.50
Recall	0.82	0.58	0.54
F1-measure	0.81	0.61	0.52

Results (asynchronous approach)

Parameters of the computational experiment:

- Reference dataset (ImageNet)
- Images 512 x 512 pixels (3 channels)
- 4 hosts
- each host downloads 50000 fragments for train and 10000 fragments for test
- 50 iterations of updating the global model
- The computational task on one host takes about 1.7 hours.



Conclusions

- As a result of the experiment, the possibility of neural networks distributed learning on a desktop grid system on the BOINC platform was confirmed.
- The location of the image dataset on a separate FTP server allowed to unload the BOINC server and showed the possibility of using large datasets in BOINC projects.
- The method of random formation of local training sets was tested.
- Using a minimum quorum allowed us to reduce the waiting time when aggregating results.

Thank you for attention

Centre for distributed computing
Institute for information transmission problem,
Russian academy of sciences
(IITP RAS)

web: distributed-computing.ru

e-mail: kurochkin@iitp.ru, qurochkin@gmail.com