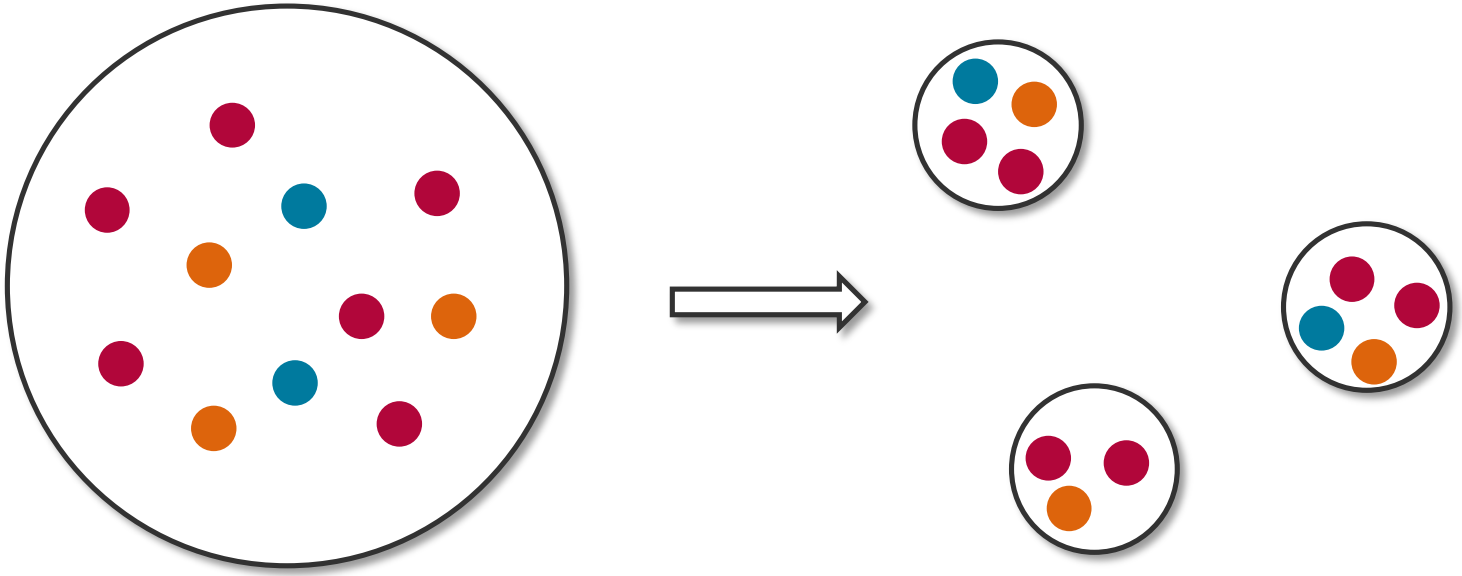


Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Machine Intelligence with Deep Learning
Importance batching for improved training of neural networks

Samik Real, Jonathan Gadea Harder, Johannes Hötter

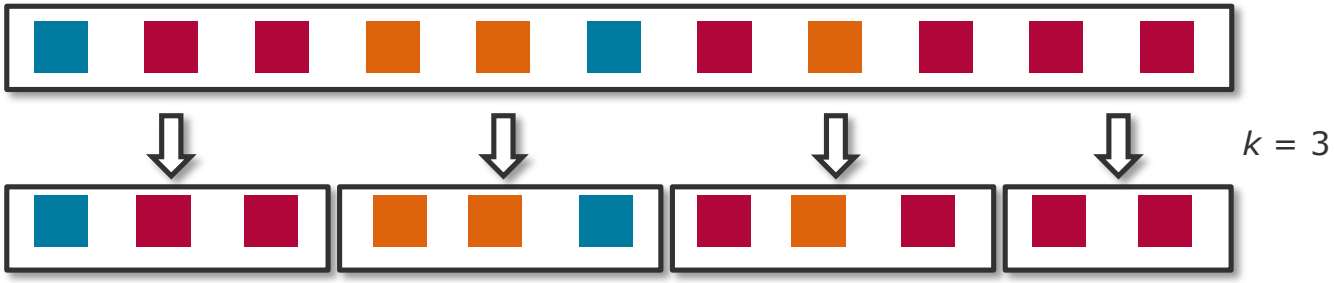
02.12.2019

Agenda

1. Introduction to the problem
2. Our current results
   - Fashion-MNIST
   - CIFAR-10
   - CIFAR-100
3. Next steps

Introduction to the problem

- Stochastic learning splits data into multiple batches of a maximum size k
  - For instance: 11 images split into 4 batches with at most 3 samples each



  - Necessary if dataset is too large for RAM
  - Speeds up training and helps model to converge due to added noise
- Usually batches are created randomly

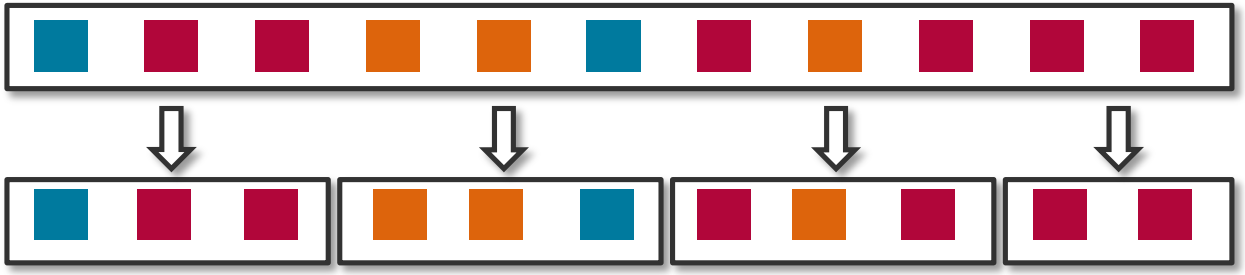
Is there a better approach ?!

Importance batching for improved training of neural networks

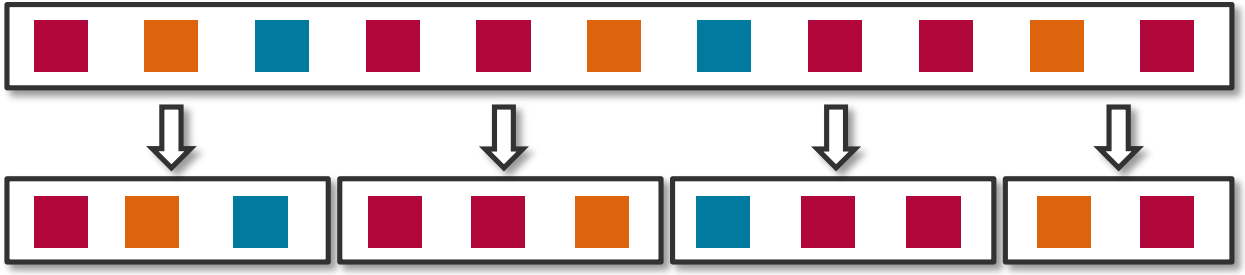
Samik Real,
Jonathan Gadea Harder,
Johannes Hötter

Introduction to the problem

- How we created batches so far:
  - Freezing the data

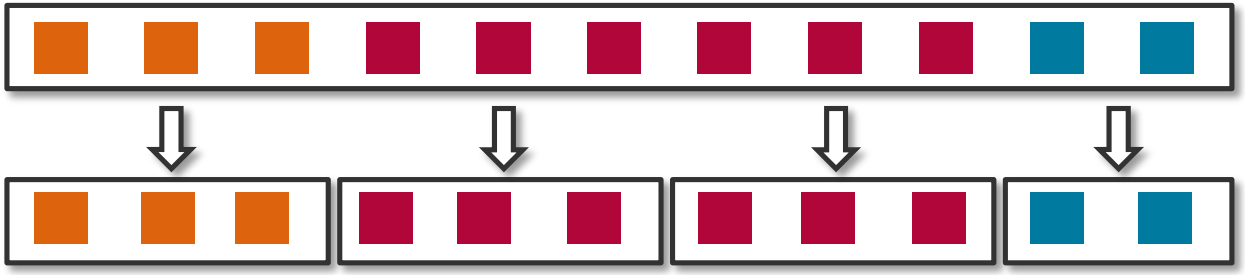


  - Shuffling the data

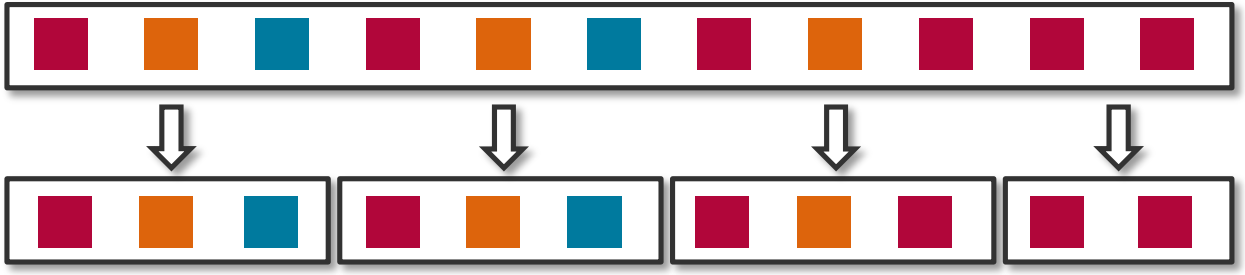


Introduction to the problem

- How we created batches so far:
  - Sorting the data homogeneously



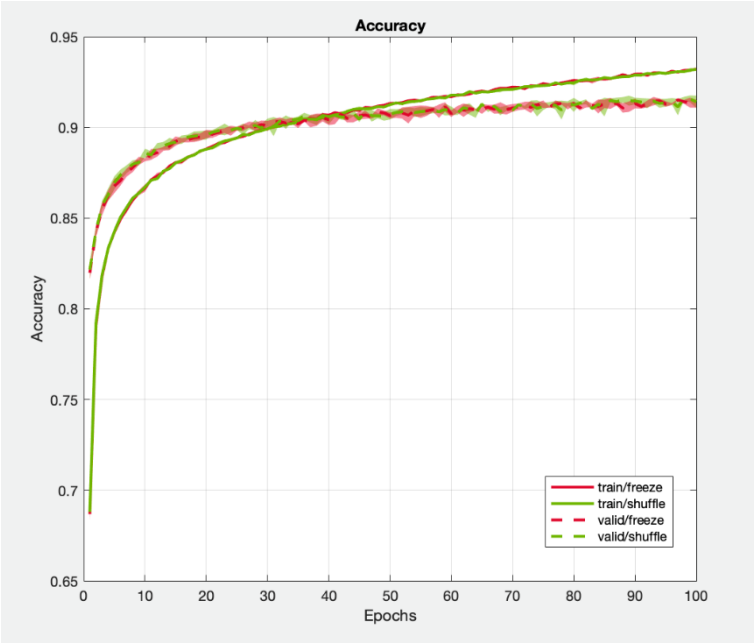
  - Sorting the data heterogeneously

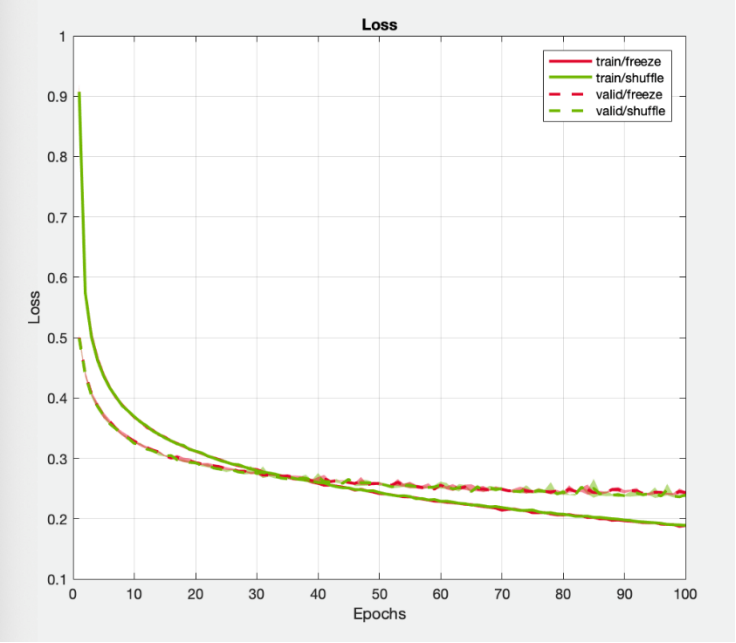


Our current results

- 1st experiment: Fashion-MNIST
  - 70,000 gray-scale 28x28 images (6:1 train/test split) with 10 classes
- Model used: Custom convolutional neural network
  - 4 layers: 3 conv layers + 1 FC layer
  - Conv layers: 64, 128, 64
  - filter sizes: 2x2
  - activation: PReLU
  - Max Pooling (2x2) after every conv layer
  - Dropout (0.5) used on FC layer

Our current results



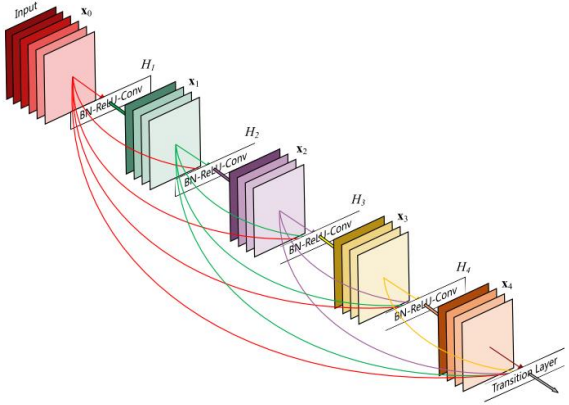


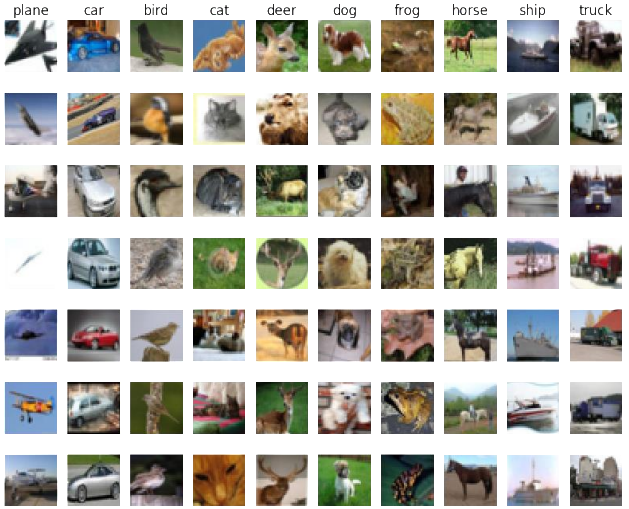
Importance batching for improved training of neural networks

Samik Real,
Jonathan Gadea Harder,
Johannes Hötter

Our current results

- 2nd experiment: CIFAR-10
  - 60,000 32x32 rgb images (i.e. 3x32x32 tensors) with 10 classes, uniformly distributed along all images
  - Model used for classification: ResNet with 18 layers



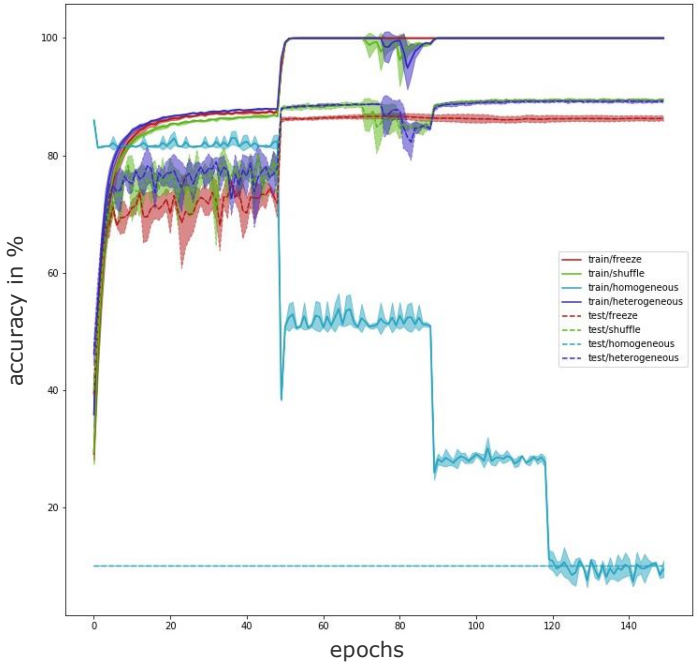


Importance batching for improved training of neural networks

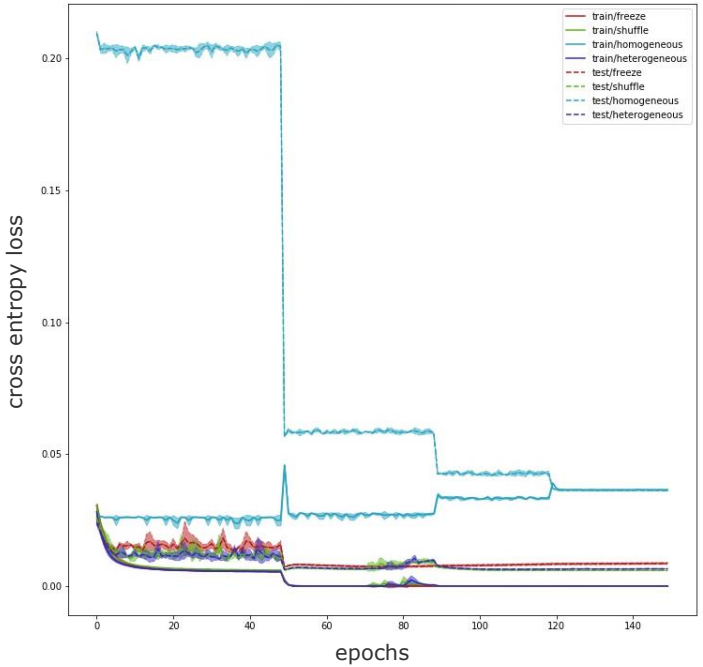
Samik Real,
Jonathan Gadea Harder,
Johannes Hötter

Our current results

Change of accuracy over epochs



Change of loss over epochs

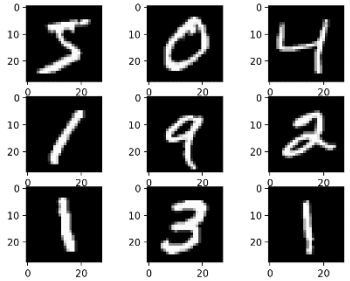


Importance batching for improved training of neural networks

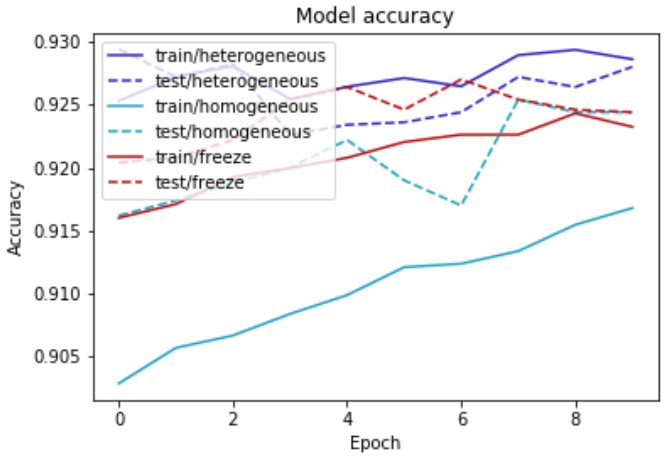
Samik Real,
Jonathan Gadea Harder,
Johannes Hötter

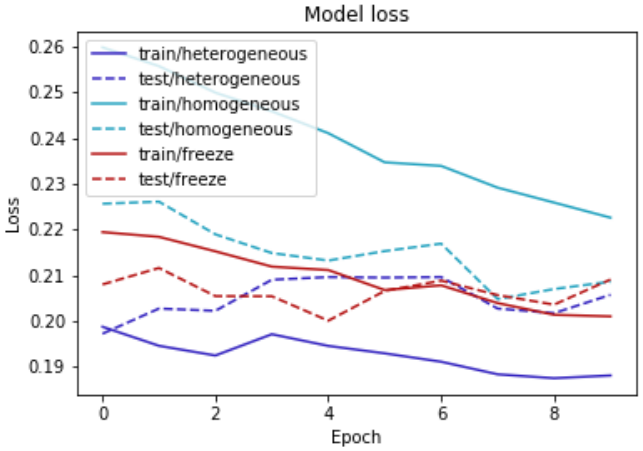
Our current results

- 3rd experiment: MNIST
  - 70,000 gray-scale 28x28 images (6:1 train/test split) with 10 classes
- Model used: Custom convolutional neural network
  - 3 layers: 2 conv layers + 1 FC layer
  - Conv layers: 64, 32
  - filter sizes: 2x2
  - activation: ReLU in conv layers+ softmax in FC layer
  - Max Pooling (2x2) after every conv layer
  - Dropout (0.3) used on conv layer, Dropout (0.5) used on FC layer



Our current results





Importance batching for improved training of neural networks

Samik Real,
Jonathan Gadea Harder,
Johannes Hötter

Next steps

- Implementing the weighted random sampling method

Algorithm D, a definition of WRS

Input : A population V of n weighted items
Output : A set S with a WRS of size m
1 : For $k = 1$ to m do
2 : Let $p_i(k) = w_i / \sum_{s_j \in V-S} w_j$ be the probability of item v_i to be selected in round k
3 : Randomly select an item $v_i \in V - S$ and insert it into S
4 : End-For

- Experimenting with different ideas to sort the data for training
  - Sorting the data by the computed loss of each input globally
  - Sorting the data by the computed loss of each input per class
- Creating batches of dynamic sizes (i.e. increasing/decreasing *k*)

Thank you
for your attention!

Samik Real, Jonathan Gadea Harder, Johannes Hötter

02.12.2019