kaggle days

Neural networks: Practical Guide

Pavel Ostyakov
Samsung AI

Ai EVERYTHING By GITEX

شرطة دبي DUBAI POLICE

مجلس علماء شرطة دبي DUBAI POLICE SCIENTISTS COUNCIL

LOGICAI

kaggle

Today’s topics

- How to speed up convergence of neural networks
- How to improve quality of predictions
- Useful tricks and tips for deep learning

General algorithm

1. Read the task carefully.
2. Think about the problem. Do you really need deep learning?
3. Think again, are not there any ways to solve it better?

General algorithm. Part 2

4. Collect as much data as you can;

5. Write basic pipeline or reuse a previous one;

6. Define a very simple model for experiments

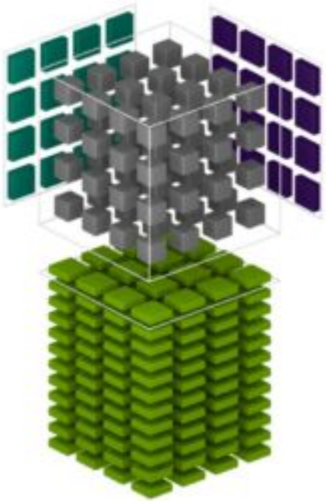
We need to speed up our experiments and improve quality, how to do it?

Data representation

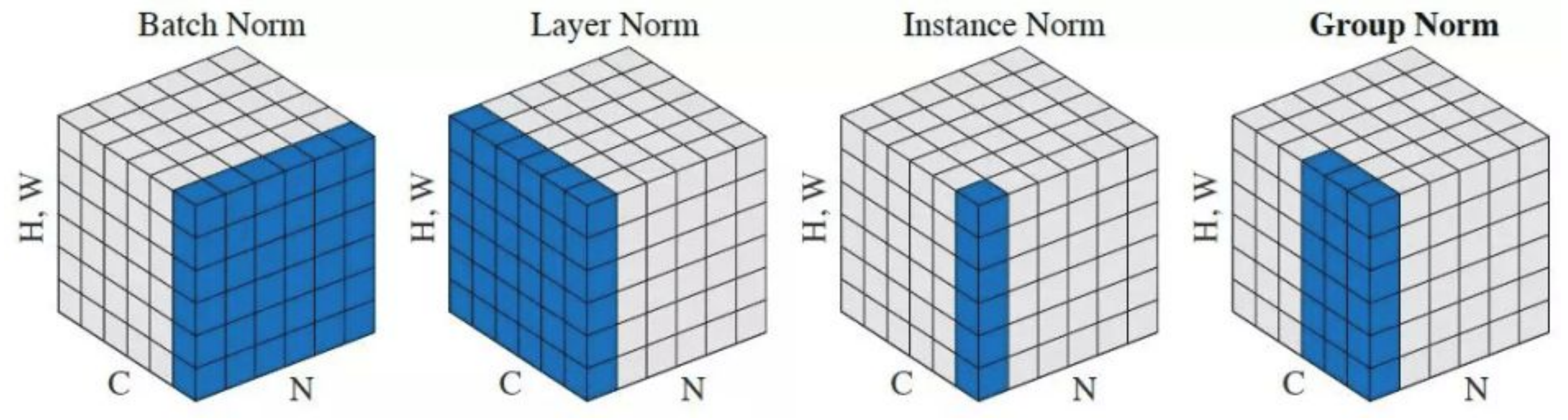
- Try to extract domain-specific features from your data;
- Normalize values. They should be similar to samples from the normal distribution;
- For first experiments, try to reduce the size of your data;

FP16 and mixed precision

- Use NVIDIA Apex if your GPU supports FP16;
- It can speed up training up to 2 times.

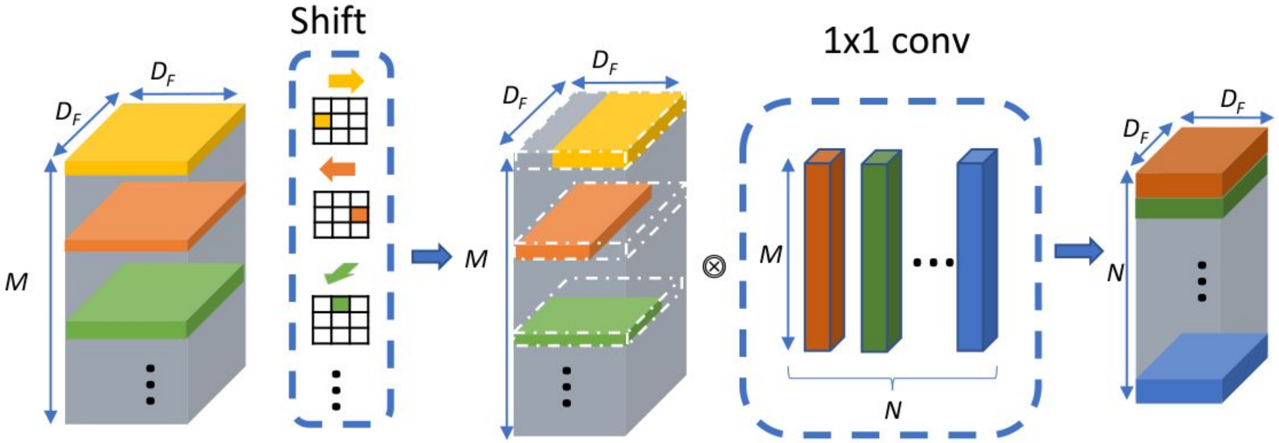


Normalization techniques



Modifications in architecture

Try different ways to extract non-local features

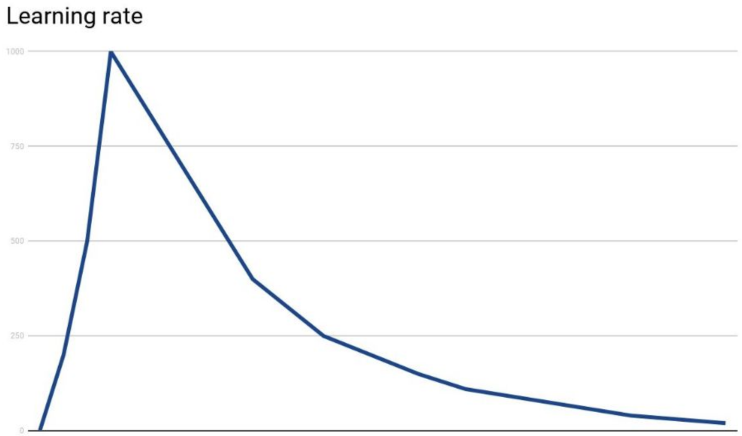


Optimization

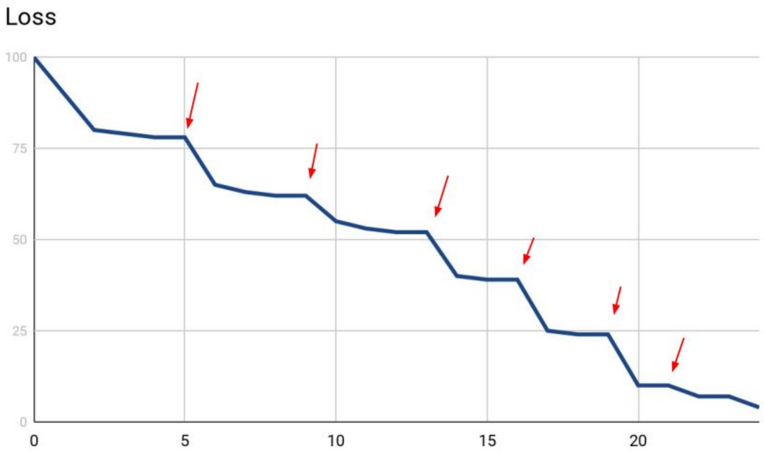
- Try different optimizers. You may start experiments with Adam;
- Try to set different learning rates for different layers;
- Freeze random layers on each step;
- Start from small batch size and increase it during training;

Learning rate scheduling

Warmup with linear decay



ReduceLROnPlateau



Data augmentation

- Use strong augmentations if you don’t have enough data;
- Increase probability of augmentations during training;
- Try to find additional data
- Data augmentation should be applied everywhere, even if it is not possible to augment data

Data augmentation. Part 2

- Use mixUp to decrease overfitting

$$\tilde{x} = \lambda x_i + (1 - \lambda) x_j,$$

$$\tilde{y} = \lambda y_i + (1 - \lambda) y_j,$$

$$\lambda \sim \mathrm{Beta}(\alpha, \alpha)$$

We already have a model, how to improve an accuracy of predictions?

Use test time augmentation (TTA)

- Use augmentations which were used during training
- Simply average all predictions
- TTA-10 is very fast but it can significantly improve accuracy
- TTA-128 still makes sense

Apply k-nearest neighbors search

- Extract features from the network for validation set
- Build index (use nmslib)
- During inference extract features for the test data and find their neighbors
- Use some heuristics to improve score

Adapt predictions for test distribution

- Tune the last layer to perform better on another target’s distribution;
- Fix predictions if you have some prior knowledge

Several tips

- Analyze mistakes of your models. Try to modify your algorithm;
- Always look at several metrics;
- Use correct weight initialization;
- Test ideas using small models;

Practical part

Your task

- Solve image classification task on CIFAR-10 dataset;
- Improve accuracy on validation set after 2 epochs;
- Do not do significant changes in model architecture;
- Use tips and tricks from slides;

Improve quality of classification

1. git clone
2. pip install -r requirements.txt
3. python bin/train.py cifar_pipeline/configs/simple_cnn.py

Slides: http://bit.ly/2IPHGIG

Link to our repo: https://bit.ly/2R9ZvC2