CMap: DPeak Challenge v1.0 4th place report

Author: Roman Chernenko (TopCoder nickname: Ramzes2)
Country of residence: Ukraine
Degree: bachelor in computer science, Cherkasy National University
Motivation: solving of real problem with prize moneys

High-level description

The experiment is processed in two phases. In the first phase, I collect statistic for every gene pairs from the whole experiment. With this information, I estimate boundaries of informative values just by dropping parts with a low amount of measurements.
In the second phase, every set of measurements clipped with these boundaries. Then I calculate 32-bins histogram of these measurements. And then simple unet-like neural network predicts positions of low value and high value in the [0;1] interval. Later these values recalculated to absolute values with the known clipping boundaries.

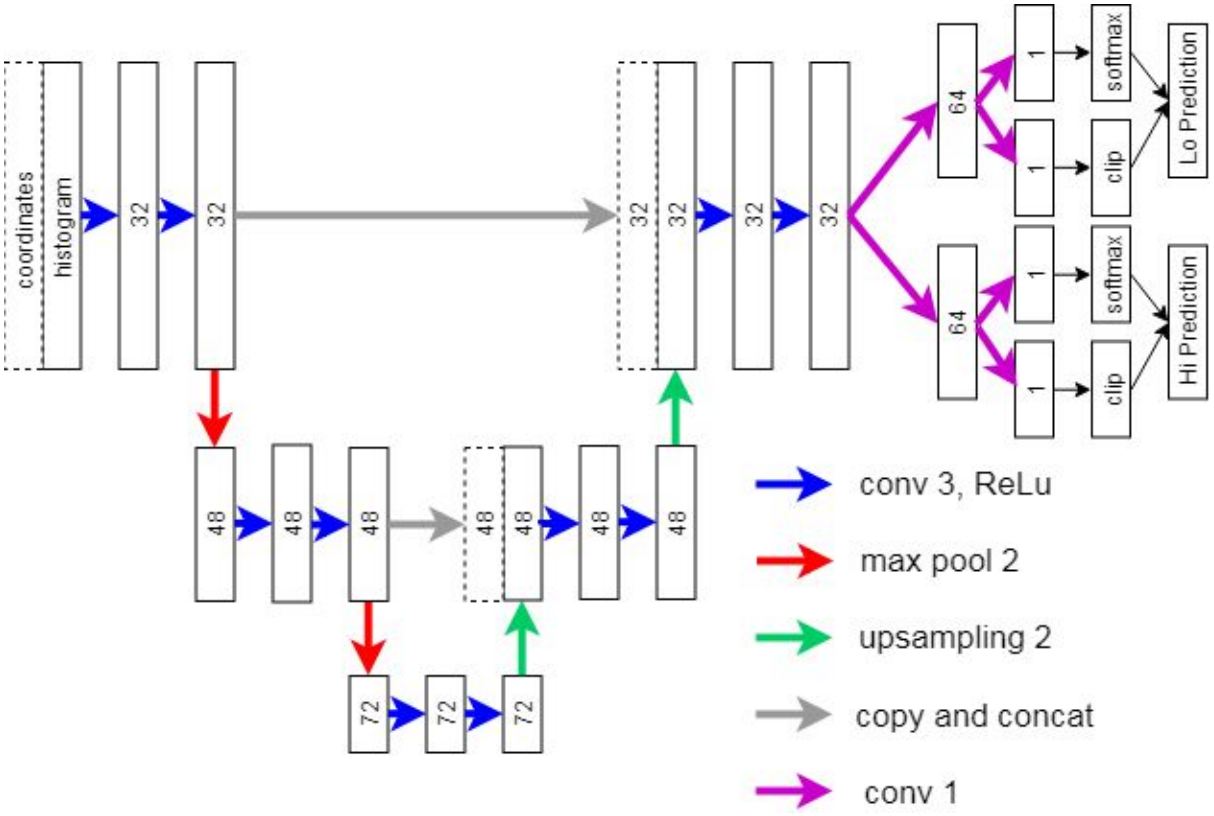
I also implemented small C++ application for reducing of data preprocessing time. The application loaded all experiment's files and generated one csv-file with 32-bins histogram.

Neural network architecture

As the input for the neural network, I used 32 bins histogram of the measured values. As it was described before, measurements are filtered with the precalculated boundaries. Additionally, I used an approach known as CoordConv - added one more input layer with the bins coordinates in the [0;1] interval.

The neural network contains 2 parts. The first part is the classic U-net architecture with the depth 3. For filters counts see illustration. The output of this network is used for predicting low and high values separately. For this prediction, I have 2 subnetworks with the same architecture but weights are trained separately. The architecture of the prediction part is very simple, but has 2 outputs:

1. bins classification (softmax activation) - trained to predict bin where is low/high gene is located
2. prediction of exact value shift if gene located in this exact bin (32 values from 0 to 1) - trained to predict the accurate location of the gene inside bin for the correct bin, 0 for all bins to the right, 1 for all bins to the left from the correct bin



Having predicted bin and shift of the value inside this bin I can recover exact gene position. That's was important to implement this calculation in a differentiable manner, so I can make training end-to-end.

Even with the ability to train network end-to-end with the simple MSE loss, I used supplementary losses for the separate outputs of the low and high genes. For the classification output, the categorical cross-entropy loss is used. And for the shift values, simple MSE loss is used. So, in the end, I have 3 losses - MSE on the low and high values, CCE on bins classifications and MSE on the bins shifts. these losses are weighted with the weights 1, 0.001 and 0.001.
For training I used Adam optimizer with the cosine annealing of learning rate 10^-4 and 10^-8.

Training

The solution archive already contains CNN weights file, so training process is optional. The archive also contains input and ground-truth data from the competition pack.

Requirements:

- Linux x64 OS (tested on Ubuntu 17.10)

- CUDA GPU (optional, but highly recommended, for training process only)
- anaconda with python 3.6
- zip compress tool
- python packages: tensorflow-gpu, keras, numpy, pandas, matplotlib, sklearn, tqdm, joblib

start training:
python ./train_cnn.py

Deployment

Requirements:

Linux x64 OS with installed docker

I prepared a deployment script for collecting all needed files into docker. Just start ./deploy.sh and you will receive “cmap_docker.zip” archive in the same format as required for submission.

Running

1. Copy “cmap_docker.zip” archive to the target PC and unpack it.
2. Run next commands for docker image installation:
   “docker build -t cmap .”
   “docker tag cmap cmap/solution”
3. Download and unpack “competitor_pack_v2” into separate folder
4. Run “run.sh” script from “competitor_pack_v2” folder

Feedback

I already posted my short feedback on the TopCoder forum here:
https://apps.topcoder.com/forums/?module=Thread&threadID=930240&start=0