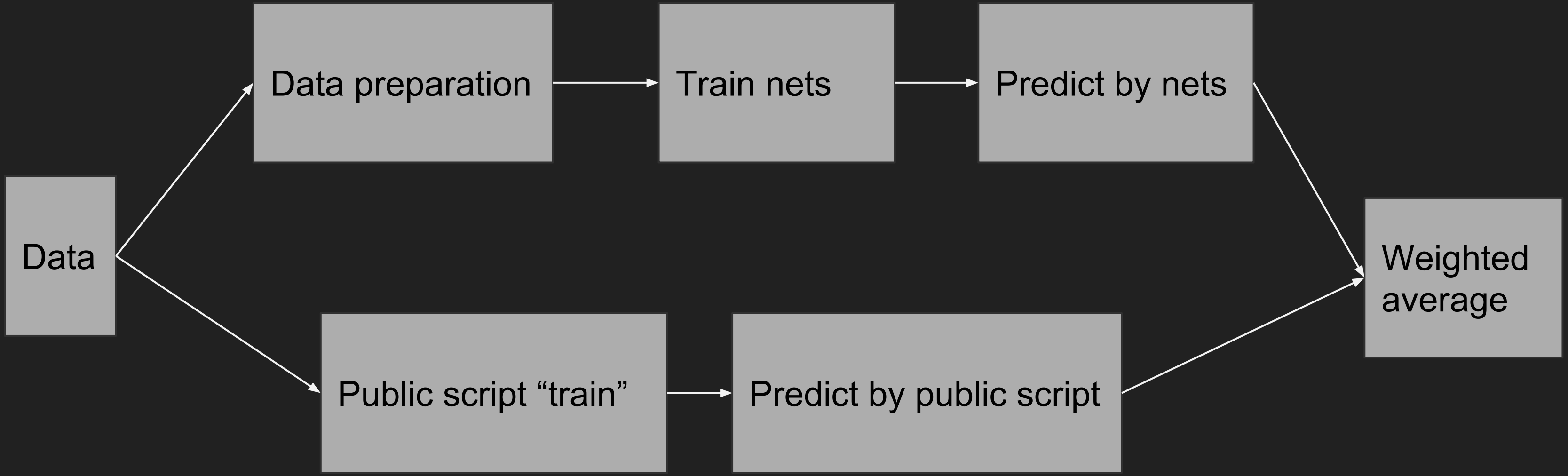
Kaggle Expedia competition: 18th place solution summary

Vitalii Moshkivskyi
kaggle: Sparse Woodman

Hardware

- 16GB RAM
- 240GB SSD
- GTX 980
- CPU does not matter

General overview



Public script idea: “train”

Compute weighted (closer to 2015 - bigger weight) hotel_cluster distribution by following keys:

- Data leak:
    - (user_location_city, orig_destination_distance)
- Find user’s booking preferences by keys:
    - (user_id, user_location_city, srch_destination_id, hotel_country, hotel_market)
    - (user_id, srch_destination_id, hotel_country, hotel_market)
- Local preferences:
    - (srch_destination_id,hotel_country,hotel_market,is_package)
    - (hotel_market)
- Global preferences

Public script idea: prediction

Until we have 5 hotel_clusters:

- If data leak have data for this instance - add
- If user booked something similar - add
- If we have local preferences for this instance - add
- If still less that 5 hotel_clusters - add globally best hotel_clusers

Data preparation

Constraint: RAM

- Fix dates:
  - train data have empty(“”) and non valid dates(2663-01-30)
  - replace them with something meaningful
- Split years:
  - split train csv to 4 parts: 2014/2013, click/booking
- Split to pandas h5 chunks
  - load every part into pandas(DTYPES!!!)
  - shuffle
  - split into 2^16 rows
  - chunks save as hdf5 (pickle is bugged)
  - split test.csv into chunks and save as hdf5

Data preparation: train/val split

Constraint: RAM

Validation(1 chunk from every part, bookings only):

- random 65k bookings from 2013
- random 65k bookings from 2014

Important! Validation data are excluded from encoding creation

Data preparation: encoding creation   Constraint: RAM

- Compute **dates difference in days** and save them as 3 new features.
- Save **date_time_hour and date_time_month** as 2 new features
- Drop original dates, cnt, is_booking

Got: ['site_name', 'posa_continent', 'user_location_country', 'user_location_region', 'user_location_city', 'orig_destination_distance', 'user_id', 'is_mobile', 'is_package', 'channel', 'srch_adults_cnt', 'srch_children_cnt', 'srch_rm_cnt', 'srch_destination_id', 'srch_destination_type_id', 'hotel_continent', 'hotel_country', 'hotel_market', 'srch_ci_minus_date_time', 'srch_co_minus_srch_ci', 'date_time_minus_srch_co', 'date_time_hour', 'date_time_month’]

Total: 23 features and 1 label - hotel_cluster (hc)

Data preparation: encoding creation Constraint: RAM

```
for data in [2013click, 2013booking, 2014click, 2014booking]:
  for every feature $f[j]$:
    Compute $P'(hc = i \mid f[j] = k)$ - normalized empirical conditional probability distribution for hotel_cluster/variable
```

Normalized: add fake event for every (hc,f[j]) pair # ensure every event happened at least once

$P(hc = i \mid f[j] = k) = 100 \cdot P'(hc = i \mid f[j] = k) - 1$ # mean and std normalization

Data preparation: encode data

Constraint: SSD

```
Something encoded as 2013(2014) means:
    For every row R:
      Explode every feature f[ j ] into 2 len100 features:
        P(hc | f[ j ] = R[ j ]), for 2013click data and
        P(hc | f[ j ] = R[ j ]), for 2013booking data
```

Got 46x100 matrix(image)

Data preparation: encode data

Constraint: SSD

Train:

- 2013booking encoded as 2014 (47GB)
- 2014booking encoded as 2013 (23GB)

Test(for submission creation):

- 2015booking encoded as 2014 (62GB)

Total: 132GB

Validation:

- 2013booking encoded as 2013
- 2013booking encoded as 2014
- 2014booking encoded as 2013
- 2014booking encoded as 2014

Total: 6GB

Networks: general

Framework: caffe (https://github.com/BVLC/caffe)

Loss: softmax log loss

Nonlinearity: VLReLU, negative_slope = 0.2

Initialization: OrthonormalLSUV init [All you need is a good init] (arXiv:1511.06422)]

Networks: general

Batch size: 128 2013bookings + 128 2014bookings, 256 total

Solver: SGD with 0.9 momentum

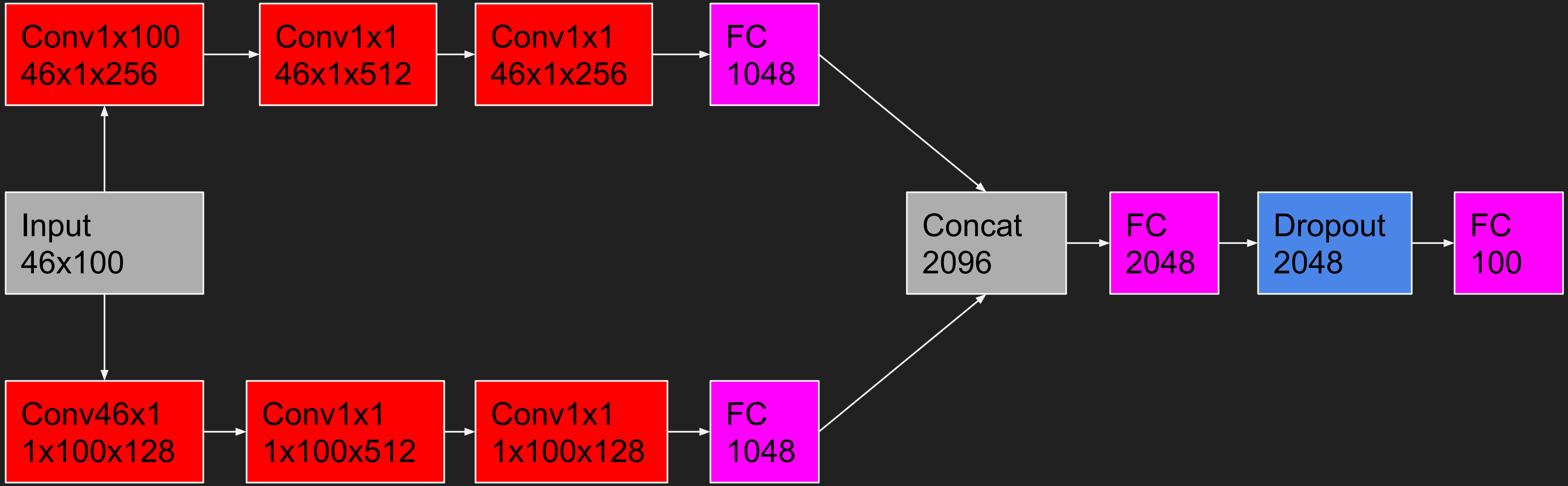
Base learning rate: 0.02.
Rule of thumb: maximum lr for which net starts to converge, divided by 2

Gamma: 0.3, 7 steps.
In the end of training learning rate is 5000 time smaller that base learning rate

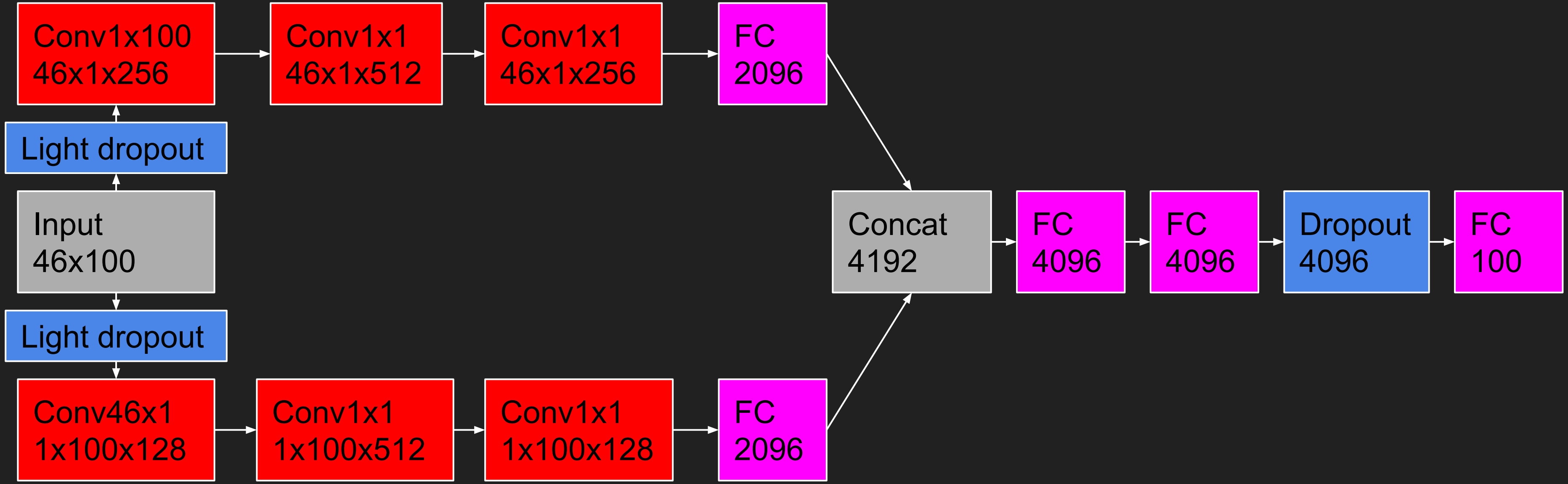
Max iterations: ~80K == 20M rows

Traintime: 1-4 hours

Networks general architecture publicLB 0.49566

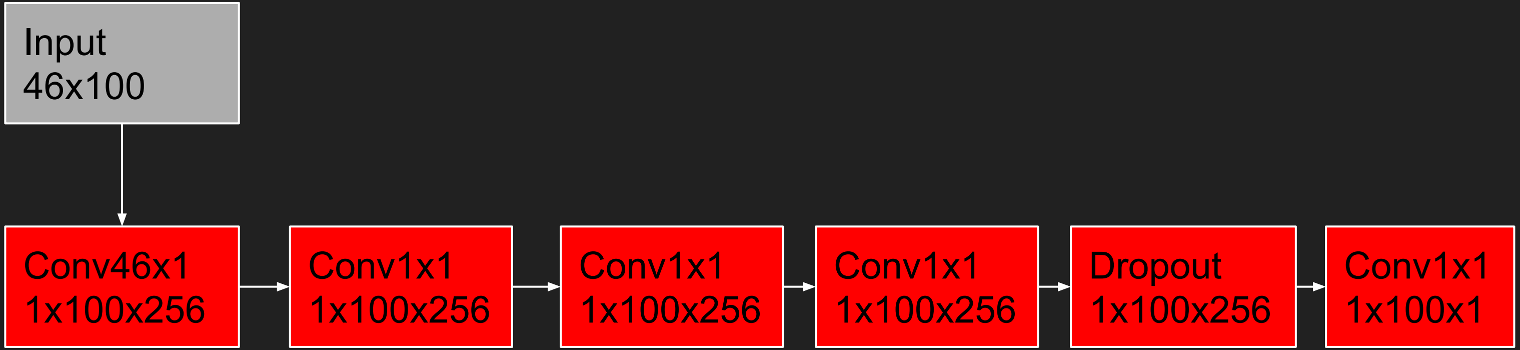


Networks varians: deeper wider more dropout publicLB 0.48754



Networks varians: no FC

publicLB 0.48389



Input
46x100

Conv46x1
1x100x256

Conv1x1
1x100x256

Conv1x1
1x100x256

Conv1x1
1x100x256

Dropout
1x100x256

Conv1x1
1x100x1

Local validation

We have 2 x 2 x 2 validation scores:

[MAP@5, logloss] x

[2013data, 2014data] x

[encoded as 2013, encoded as 2013]

Local validation to public leaderboard mapping is nonlinear =(

Local validation: example

MAP@5	2013 bookings	2014 bookings
Encoded as 2013	0.819388	0.554535
Encoded as 2014	0.626688	0.825403

Public lb

MAP@5:

0.49566

logloss	2013 bookings	2014 bookings
Encoded as 2013	1.07566	2.20843
Encoded as 2014	1.90397	1.02978

Leaderboard

	Public score	**Private score**	**Public place**	**Private place**
Best single net	0.49566	0.49304	1126	1146
4nets ensemble	0.50041	0.49747	784	784
Public script	0.50182	0.49914	~333	~310
4nets ensemble + Public script	0.50855	0.50558	26	24
4nets ensemble + (Public script / 2)	**0.51028**	**0.50719**	**18**	**18**

Ways to improve this solution

Data encoding scheme can’t capture multivariable interaction, possible solutions:

1)Use current scheme, but:

- Add feature pairs (triplets) to features. Cons: 12x (89x) more memory usage
- Same but use only some pair(triplets). Cons: I’m lazy =(

2)Use different coding scheme:

- One-hot (embedding) encoding. Cons: slow? memory?