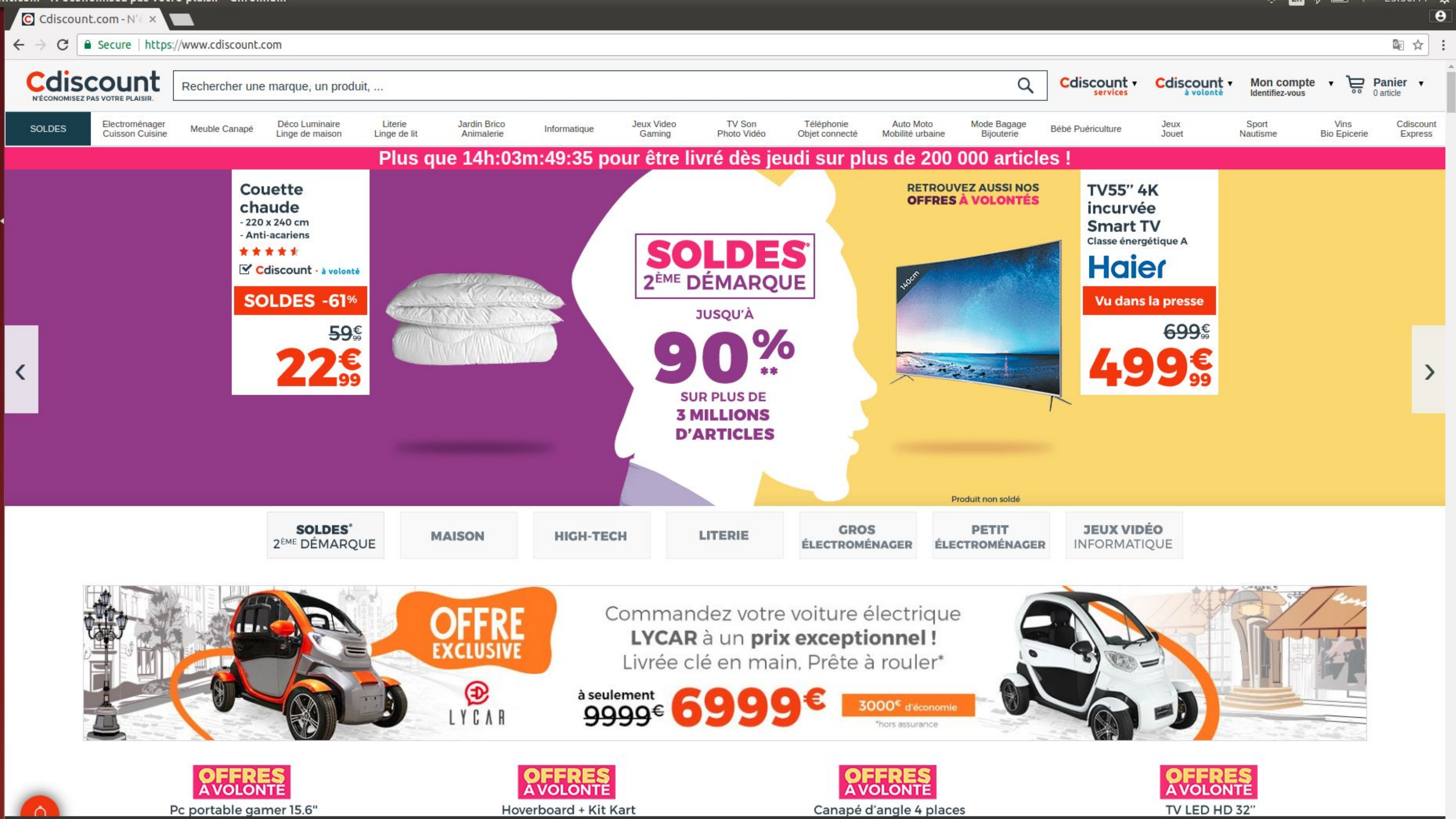
Cdiscount’s Image Classification Challenge

5th place solution

kaggle™ Cdiscount



Cdiscount
N'ÉCONOMISEZ PAS VOTRE PLAISIR.

Rechercher une marque, un produit, ...

Cdiscount services | Cdiscount à volonté | Mon compte Identifiez-vous | Panier 0 article

SOLDES | Electroménager Cuisson Cuisine | Meuble Canapé | Déco Luminaire Linge de maison | Literie Linge de lit | Jardin Brico Animalerie | Informatique | Jeux Video Gaming | TV Son Photo Vidéo | Téléphonie Objet connecté | Auto Moto Mobilité urbaine | Mode Bagage Bijouterie | Bébé Puériculture | Jeux Jouet | Sport Nautisme | Vins Bio Epicerie | Cdiscount Express

Plus que 14h:03m:49:35 pour être livré dès jeudi sur plus de 200 000 articles !

Couette chaude
- 220 x 240 cm
- Anti-acariens
★★★★★
☑ Cdiscount · à volonté
SOLDES -61%
~~59€99~~
22€99

SOLDES® 2ÈME DÉMARQUE
JUSQU'À
90%**
SUR PLUS DE 3 MILLIONS D'ARTICLES

RETROUVEZ AUSSI NOS OFFRES À VOLONTÉS

TV55" 4K incurvée Smart TV
Classe énergétique A
Haier
Vu dans la presse
~~699€99~~
499€99

Produit non soldé

SOLDES® 2ÈME DÉMARQUE | MAISON | HIGH-TECH | LITERIE | GROS ÉLECTROMÉNAGER | PETIT ÉLECTROMÉNAGER | JEUX VIDÉO INFORMATIQUE

OFFRE EXCLUSIVE
LYCAR
Commandez votre voiture électrique LYCAR à un prix exceptionnel !
Livrée clé en main. Prête à rouler*
à seulement ~~9999€~~ 6999€
3000€ d'économie
*hors assurance

OFFRES A VOLONTE
Pc portable gamer 15.6"

OFFRES A VOLONTE
Hoverboard + Kit Kart

OFFRES A VOLONTE
Canapé d'angle 4 places

OFFRES A VOLONTE
TV LED HD 32"

Data

- Almost 9 million products
- More than 12 million images at 180x180 resolution in the train set
- 3 million images in the test set
- 5270 categories
- 1-4 images per product

Some images

1000010653
TELEPHONIE - GPS
ACCESSOIRE TELEPHONE
COQUE TELEPHONE - BUMPER

1000004079
INFORMATIQUE
CONNECTIQUE - ALIMENTATIO
CHARGEUR - ADAPTATEUR SEC

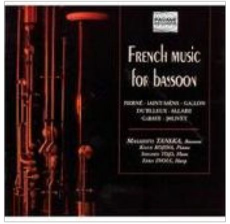
1000004141
INFORMATIQUE
PROTECTION - PERSONNALISA
COQUE - HOUSSE

1000015539
BRICOLAGE - OUTILLAGE - Q
SECURITE MAISON
ALARME AUTONOME

1000010653
TELEPHONIE - GPS
ACCESSOIRE TELEPHONE
COQUE TELEPHONE - BUMPER

1000005744
AUTO - MOTO
PIECES
BOBINE D'ALLUMAGE - BOBIN

1000018290
MUSIQUE
CD
CD MUSIQUE CLASSIQUE

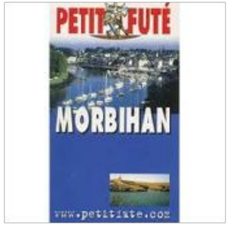


1000010653
TELEPHONIE - GPS
ACCESSOIRE TELEPHONE
COQUE TELEPHONE - BUMPER

1000018306
MUSIQUE
CD
CD VARIETE INTERNATIONALE

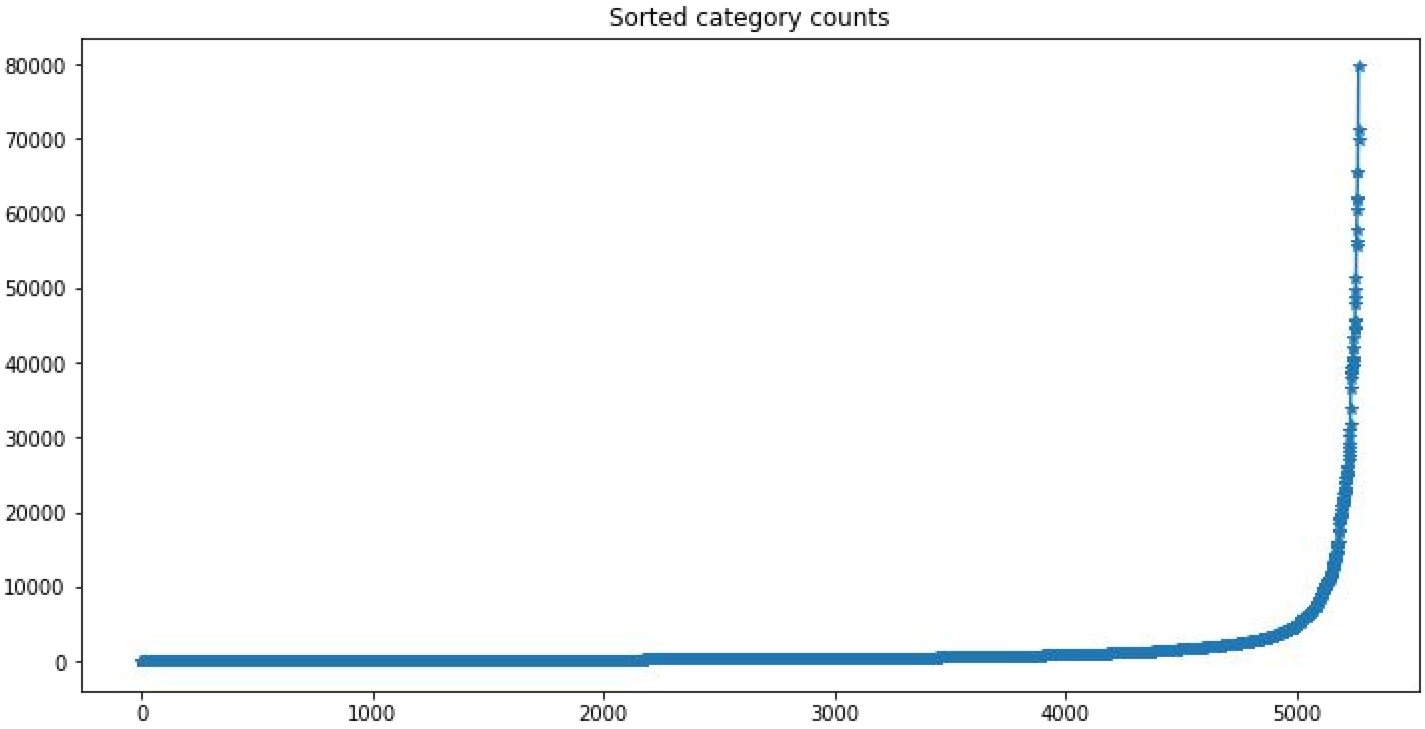
1000010961
TV - VIDEO - SON
CASQUE - MICROPHONE - DIC
CASQUE - ECOUTEUR - OREIL

1000015309
LIBRAIRIE
AUTRES LIVRES
AUTRES LIVRES



1000010653
TELEPHONIE - GPS
ACCESSOIRE TELEPHONE
COQUE TELEPHONE - BUMPER

Class imbalance



Validation split

- 150k examples for validation
- Strongly correlated with LB score
- Data distribution was the same in both train and test set

Convolutional neural networks

- Resnet models (Resnet50, Resnet101, Resnet151)
- Densenet 161
- InceptionResnetV2

We also tried

- Dual Path Networks
- Xception
- ResNeXt
- SE-Resnet
- VGG
- Nasnet

How to train a deep CNN model faster?

- Use smaller images (160x160 instead of 180x180)
- Don't Decay the Learning Rate, Increase the Batch Size
- Increase augmentation step by step
- Use pretrained weights for initialization

What did not work

- Dropout
- L2 regularization
- Multi-input CNN

How to ensemble several NNs?

- Using weighted mean
- Using weighted geometric mean
- Using 2nd layer model

Ensembling challenges

- Only one model was trained using KFold split (K=4)
- It is impossible to train XGBoost on 5270 classes
- 5270 * 4 features for each product

Hash trick

- One image could be present in the data multiple times
- However, one image could be relevant to several categories
- Finding the best way to use hashes is another challenge

Ensembling approach

- For each product and for each image of this product we extracted TOP-5 probabilities and 5 relevant classes
- We also computed MD5 hash for each picture and matched it with the most common class IDs

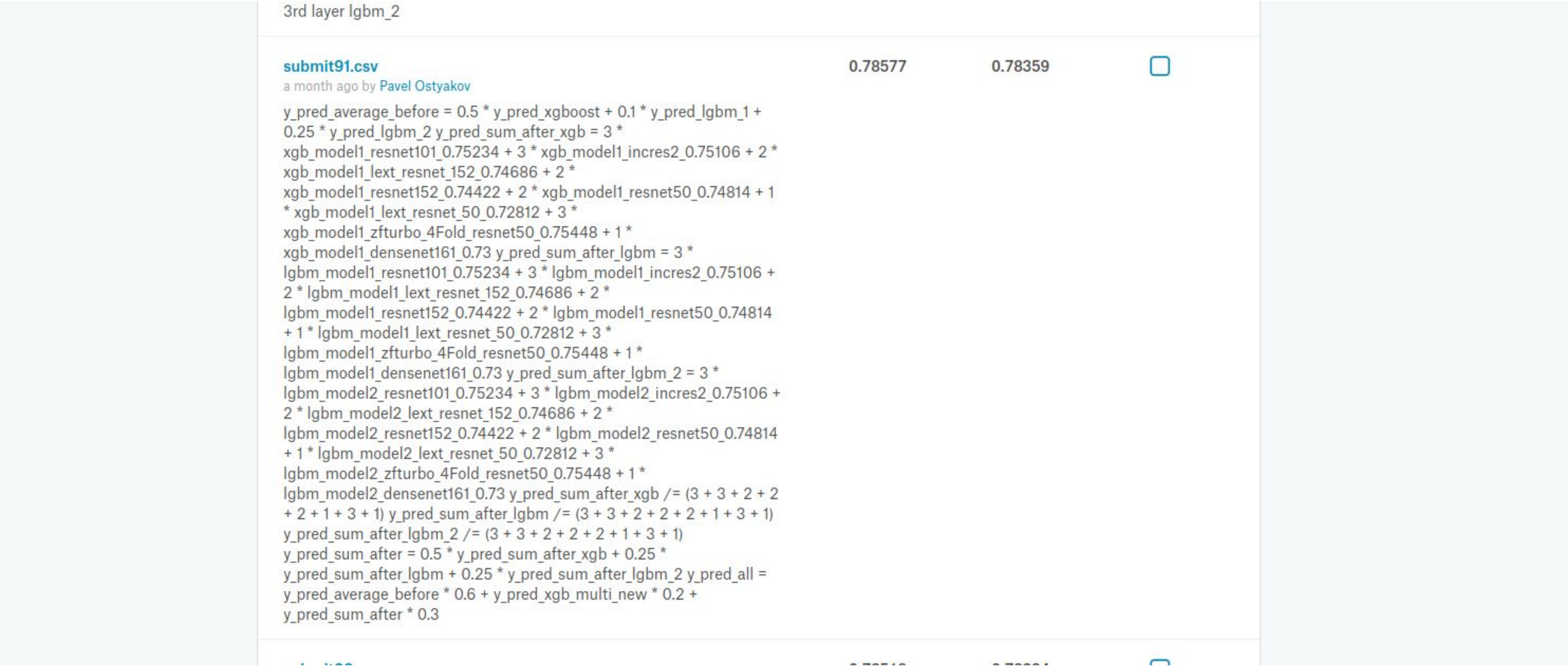
After this procedure, we had 80 raw features (40 numerical and 40 categorical)

Ensembling approach

- We added a new feature "possible_class_id" and changed the multi-classification problem to binary classification.
- Next, we were training a model which predicts 'is it true that this product has class_id which is equal to "possible_class_id"'
- We only used samples with possible_class_id which was one of the TOP5 predictions of CNN or one of the possible classes based on hash trick.

L2 models

- XGBoost
- LightGBM
- CatBoost
- Random Forest
- Extra Trees



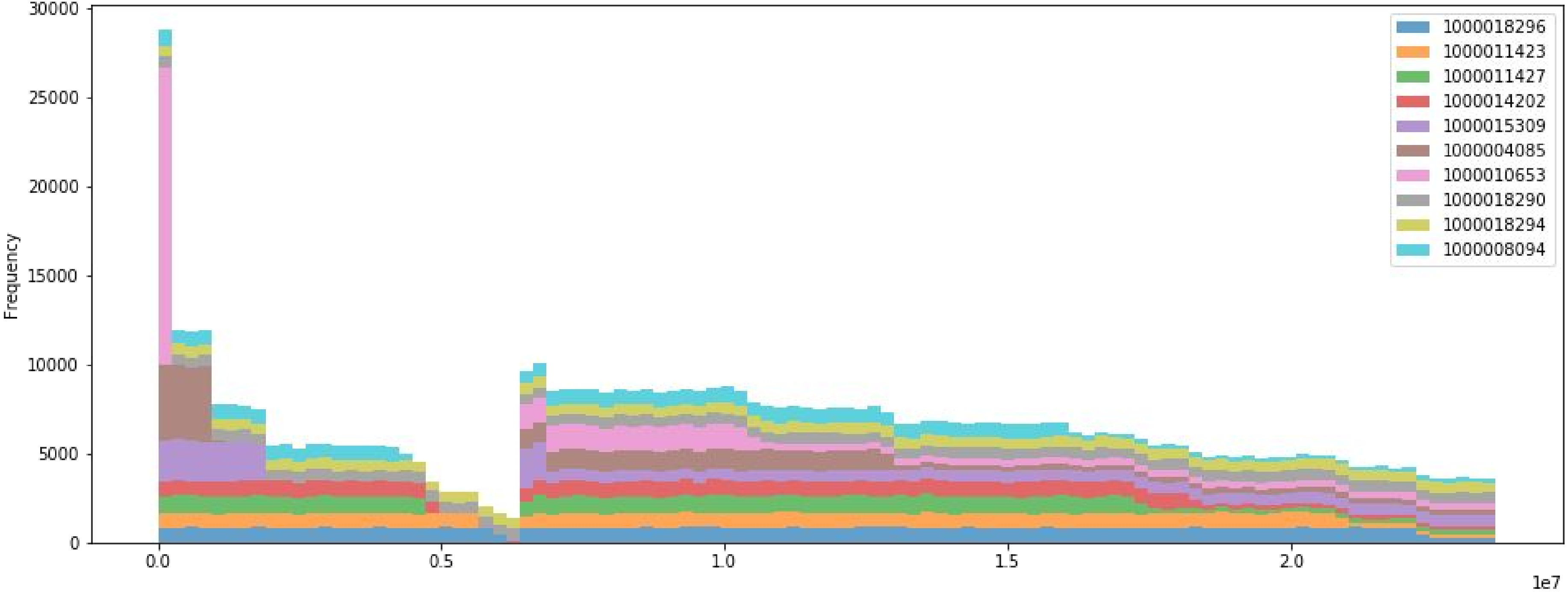
3rd layer lgbm_2

submit91.csv 0.78577 0.78359 □

a month ago by Pavel Ostyakov

y_pred_average_before = 0.5 * y_pred_xgboost + 0.1 * y_pred_lgbm_1 + 0.25 * y_pred_lgbm_2 y_pred_sum_after_xgb = 3 * xgb_model1_resnet101_0.75234 + 3 * xgb_model1_incres2_0.75106 + 2 * xgb_model1_lext_resnet_152_0.74686 + 2 * xgb_model1_resnet152_0.74422 + 2 * xgb_model1_resnet50_0.74814 + 1 * xgb_model1_lext_resnet_50_0.72812 + 3 * xgb_model1_zfturbo_4Fold_resnet50_0.75448 + 1 * xgb_model1_densenet161_0.73 y_pred_sum_after_lgbm = 3 * lgbm_model1_resnet101_0.75234 + 3 * lgbm_model1_incres2_0.75106 + 2 * lgbm_model1_lext_resnet_152_0.74686 + 2 * lgbm_model1_resnet152_0.74422 + 2 * lgbm_model1_resnet50_0.74814 + 1 * lgbm_model1_lext_resnet_50_0.72812 + 3 * lgbm_model1_zfturbo_4Fold_resnet50_0.75448 + 1 * lgbm_model1_densenet161_0.73 y_pred_sum_after_lgbm_2 = 3 * lgbm_model2_resnet101_0.75234 + 3 * lgbm_model2_incres2_0.75106 + 2 * lgbm_model2_lext_resnet_152_0.74686 + 2 * lgbm_model2_resnet152_0.74422 + 2 * lgbm_model2_resnet50_0.74814 + 1 * lgbm_model2_lext_resnet_50_0.72812 + 3 * lgbm_model2_zfturbo_4Fold_resnet50_0.75448 + 1 * lgbm_model2_densenet161_0.73 y_pred_sum_after_xgb /= (3 + 3 + 2 + 2 + 2 + 1 + 3 + 1) y_pred_sum_after_lgbm /= (3 + 3 + 2 + 2 + 2 + 1 + 3 + 1) y_pred_sum_after_lgbm_2 /= (3 + 3 + 2 + 2 + 2 + 1 + 3 + 1) y_pred_sum_after = 0.5 * y_pred_sum_after_xgb + 0.25 * y_pred_sum_after_lgbm + 0.25 * y_pred_sum_after_lgbm_2 y_pred_all = y_pred_average_before * 0.6 + y_pred_xgb_multi_new * 0.2 + y_pred_sum_after * 0.3

Product ID and category_id are related



Prior features

For each class_id:

- Frequency of this class in the whole dataset
- Frequency of this class among the closest products based on their IDs (50k closest products).

Third layer features

- Features from CNN
- Predictions of second-layer models
- Priors for categorical features

L3 models

- Three different LightGBM models
- One XGBoost model

It seems that we could improve our score by building a fourth layer model :)

L2 and L3 inference approaches

1. First, average probabilities extracted from CNNs, then build features and use them in second and third layer models.

2. Extract predictions of each CNN, build features, use it for predictions on second and third layers and only then merge predictions of different CNNs.

Semi-supervised learning

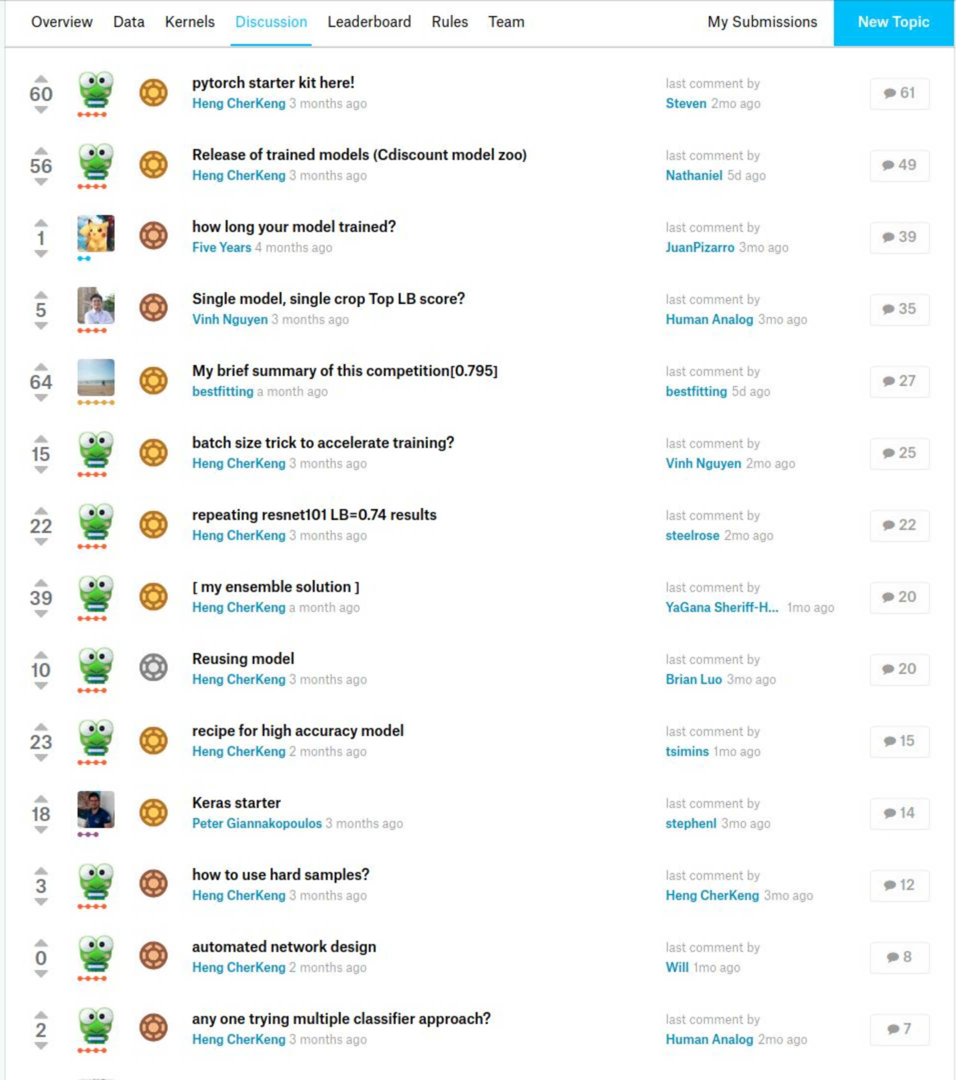
- We tried a simple pseudo-label method
- LB score of single Resnet-101 increased from 74.9 to 75.6
- However, it did not improve the score of our ensemble

Our mistakes

- Small validation set
- We did not use out-of-fold validation
- For experiments we trained very large models
- We did not pay enough attention to semi-supervised learning

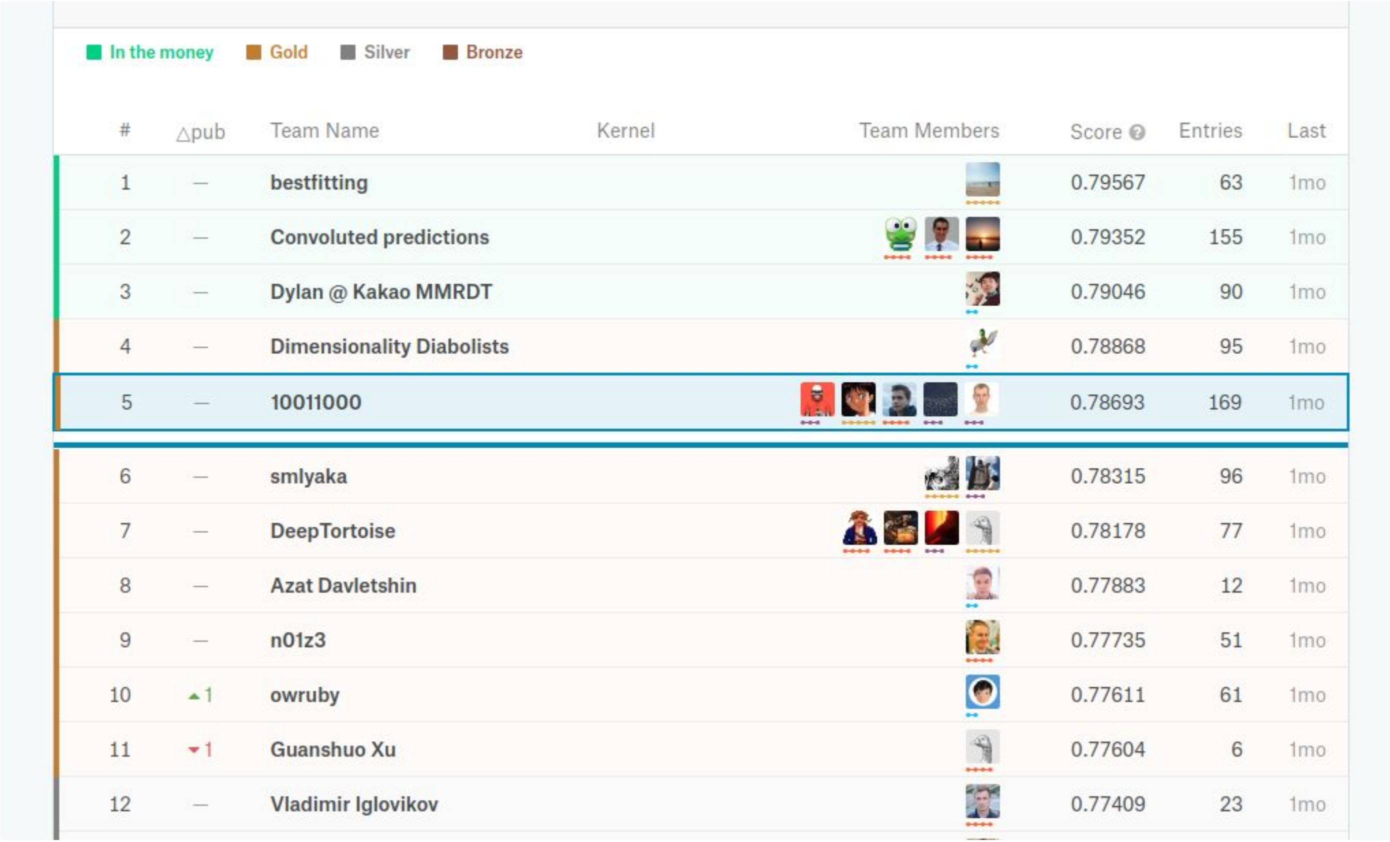
Software

- ~~Keras~~
- ~~Catboost~~
- Pytorch
- LightGBM
- XGBoost



60	pytorch starter kit here! Heng CherKeng 3 months ago	last comment by Steven 2mo ago	61
56	Release of trained models (Cdiscount model zoo) Heng CherKeng 3 months ago	last comment by Nathaniel 5d ago	49
1	how long your model trained? Five Years 4 months ago	last comment by JuanPizarro 3mo ago	39
5	Single model, single crop Top LB score? Vinh Nguyen 3 months ago	last comment by Human Analog 3mo ago	35
64	My brief summary of this competition[0.795] bestfitting a month ago	last comment by bestfitting 5d ago	27
15	batch size trick to accelerate training? Heng CherKeng 3 months ago	last comment by Vinh Nguyen 2mo ago	25
22	repeating resnet101 LB=0.74 results Heng CherKeng 3 months ago	last comment by steelrose 2mo ago	22
39	[my ensemble solution] Heng CherKeng a month ago	last comment by YaGana Sheriff-H... 1mo ago	20
10	Reusing model Heng CherKeng 3 months ago	last comment by Brian Luo 3mo ago	20
23	recipe for high accuracy model Heng CherKeng 2 months ago	last comment by tsimins 1mo ago	15
18	Keras starter Peter Giannakopoulos 3 months ago	last comment by stephenl 3mo ago	14
3	how to use hard samples? Heng CherKeng 3 months ago	last comment by Heng CherKeng 3mo ago	12
0	automated network design Heng CherKeng 2 months ago	last comment by Will 1mo ago	8
2	any one trying multiple classifier approach? Heng CherKeng 3 months ago	last comment by Human Analog 2mo ago	7

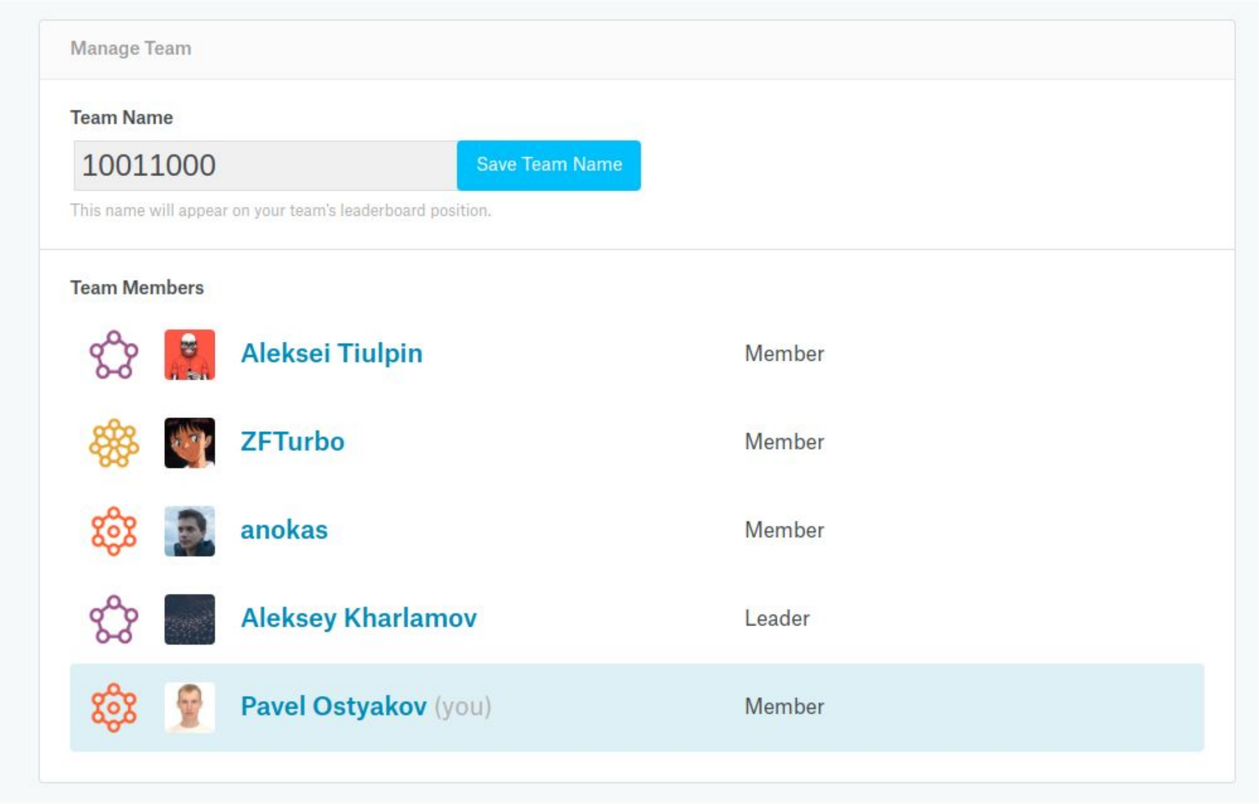
Private leaderboard



🟩 In the money ■ Gold ■ Silver ■ Bronze

#	△pub	Team Name	Kernel	Team Members	Score	Entries	Last
1	—	bestfitting			0.79567	63	1mo
2	—	Convoluted predictions			0.79352	155	1mo
3	—	Dylan @ Kakao MMRDT			0.79046	90	1mo
4	—	Dimensionality Diabolists			0.78868	95	1mo
5	—	10011000			0.78693	169	1mo
6	—	smlyaka			0.78315	96	1mo
7	—	DeepTortoise			0.78178	77	1mo
8	—	Azat Davletshin			0.77883	12	1mo
9	—	n01z3			0.77735	51	1mo
10	▲1	owruby			0.77611	61	1mo
11	▼1	Guanshuo Xu			0.77604	6	1mo
12	—	Vladimir Iglovikov			0.77409	23	1mo

Our team



Manage Team

Team Name

10011000

This name will appear on your team's leaderboard position.

Team Members

Aleksei Tiulpin	Member
ZFTurbo	Member
anokas	Member
Aleksey Kharlamov	Leader
Pavel Ostyakov (you)	Member

bestfitting(1st place)

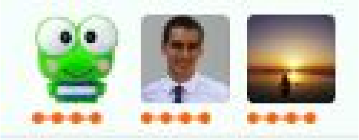
- Initial model → ResNet-34
- Representation bottleneck → 1x1 conv. to increase the channel from 512 to 5270
- x2 FC(5270)
- 11.5 epochs with 160x160 random crop, img. average, get a score >0.72
- ResNet-34 → ResNet-50
- GPU power + lack of time → ResNet-101,ResNet-152, Inception-ResNetV2 and InceptionV4

```
lr = 0.0003
if epoch > 7:
    lr = 0.0001
if epoch > 9:
    lr = 0.00005
if epoch > 11:
    lr = 0.00001
```

bestfitting part 2

- Full use of multi-images → concat. the images into a image
- Split the trainset into 4 parts → separate models
- Ensemble → 0.77 LB
- OCR(~0.14% to final sub.) → ResNet-50 Multi-Input
- Ensemble DenseNet-161,DenseNet-169,DPN-92 + TTA5 → 0.79 LB

Convoluted predictions(2nd place)

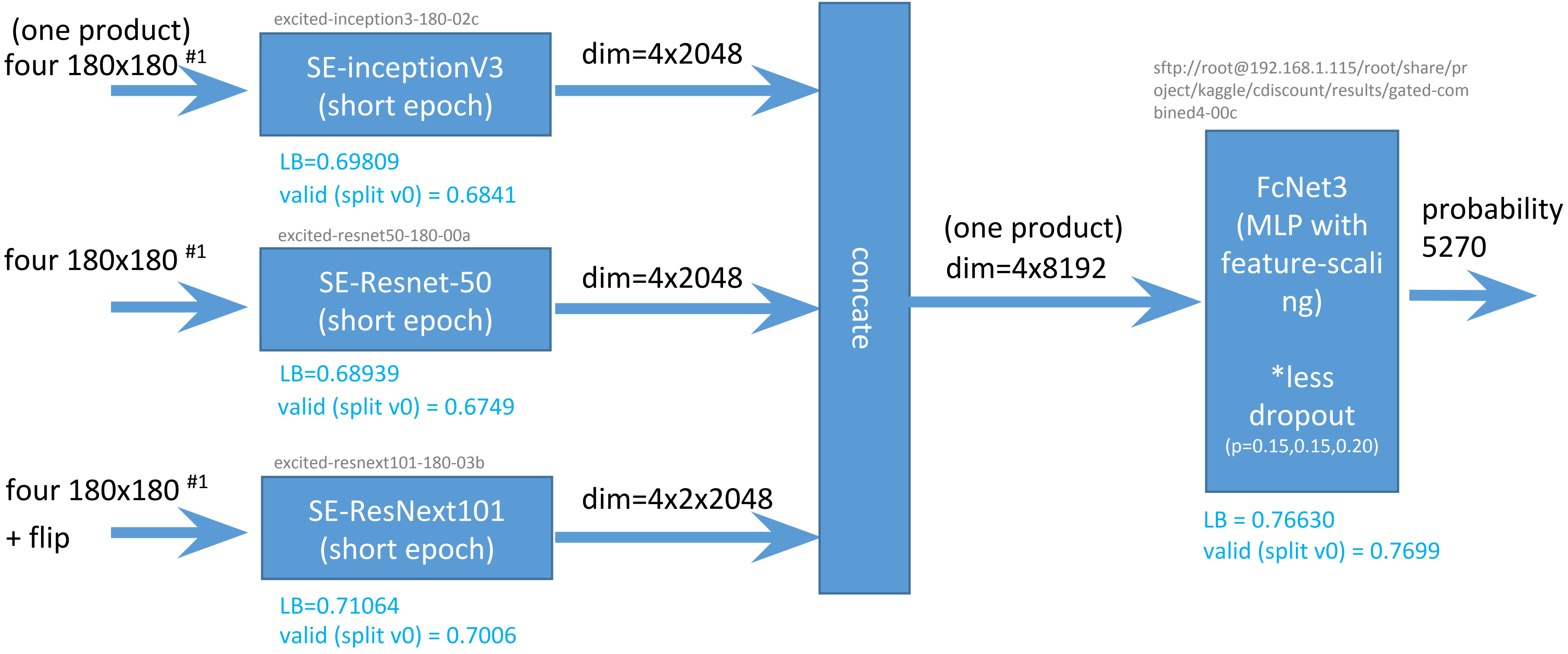


Tricks in training:

1. For base networks used dropout and augmentation=small shift, scale, rotation + flip + rotate90,180,270.Batch_size = 512 to 768, num_epoches = about 20. SDG/learning rate=0.01,0.001,0.0001 (manually tuned). Time to train (4x1080Ti) = 5 days per base networks
2. For Fcnet3 used dropout and only augmentation=flip (due to lack of space in SSD drive). Batch_size = 4096, num_epoches = about 100. SDG/learning rate=0.01,0.001,0.0001 (manually tuned). Time to train (4x1080Ti) = 1.5 days

[Model]

combine.0



(one product)
four 180x180 #1

excited-inception3-180-02c

SE-inceptionV3
(short epoch)

LB=0.69809
valid (split v0) = 0.6841

dim=4x2048

four 180x180 #1

excited-resnet50-180-00a

SE-Resnet-50
(short epoch)

LB=0.68939
valid (split v0) = 0.6749

dim=4x2048

four 180x180 #1
+ flip

excited-resnext101-180-03b

SE-ResNext101
(short epoch)

LB=0.71064
valid (split v0) = 0.7006

dim=4x2x2048

concate

(one product)
dim=4x8192

sftp://root@192.168.1.115/root/share/project/kaggle/cdiscount/results/gated-combined4-00c

FcNet3
(MLP with feature-scaling)

*less dropout
(p=0.15,0.15,0.20)

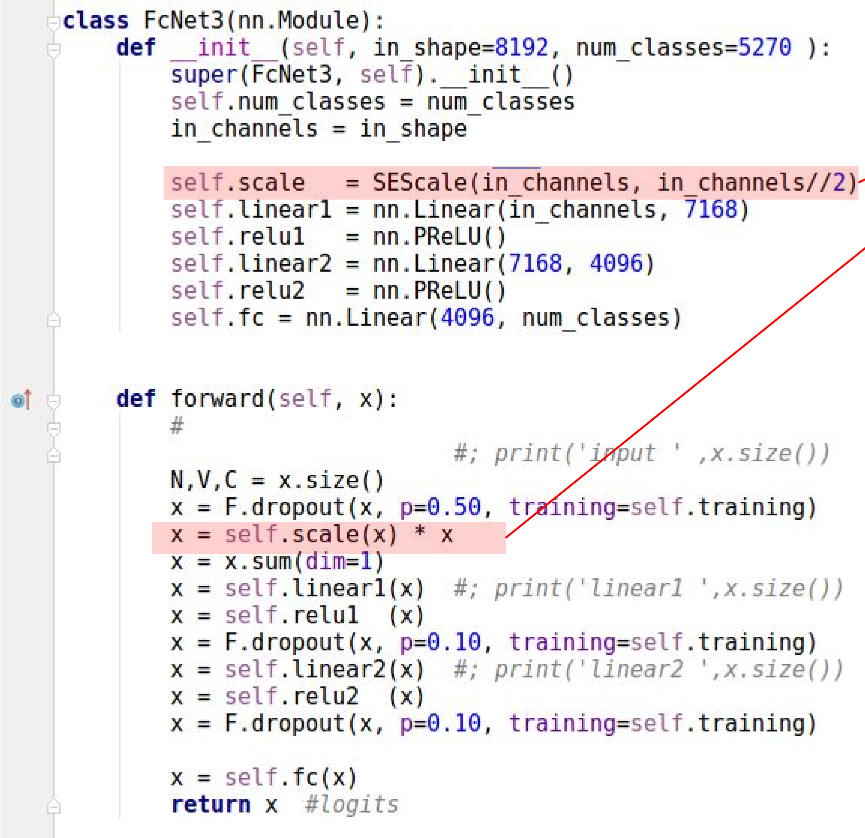
LB = 0.76630
valid (split v0) = 0.7699

probability
5270

#1 : if the product has less than 4 images, it is stuffed with zero images

[Fuse net]

FcNet3: view feature scaling (gating)



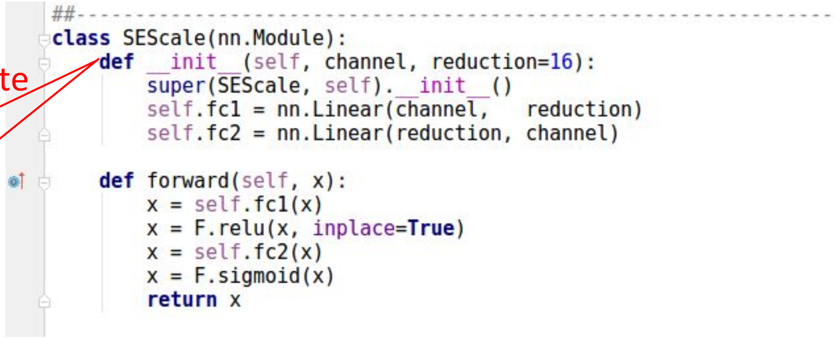
```
class FcNet3(nn.Module):
    def __init__(self, in_shape=8192, num_classes=5270 ):
        super(FcNet3, self).__init__()
        self.num_classes = num_classes
        in_channels = in_shape

        self.scale   = SEScale(in_channels, in_channels//2)
        self.linear1 = nn.Linear(in_channels, 7168)
        self.relu1   = nn.PReLU()
        self.linear2 = nn.Linear(7168, 4096)
        self.relu2   = nn.PReLU()
        self.fc = nn.Linear(4096, num_classes)


    def forward(self, x):
        #
                          #; print('input ' ,x.size())
        N,V,C = x.size()
        x = F.dropout(x, p=0.50, training=self.training)
        x = self.scale(x) * x
        x = x.sum(dim=1)
        x = self.linear1(x)  #; print('linear1 ',x.size())
        x = self.relu1  (x)
        x = F.dropout(x, p=0.10, training=self.training)
        x = self.linear2(x)  #; print('linear2 ',x.size())
        x = self.relu2  (x)
        x = F.dropout(x, p=0.10, training=self.training)

        x = self.fc(x)
        return x  #logits
```

gate

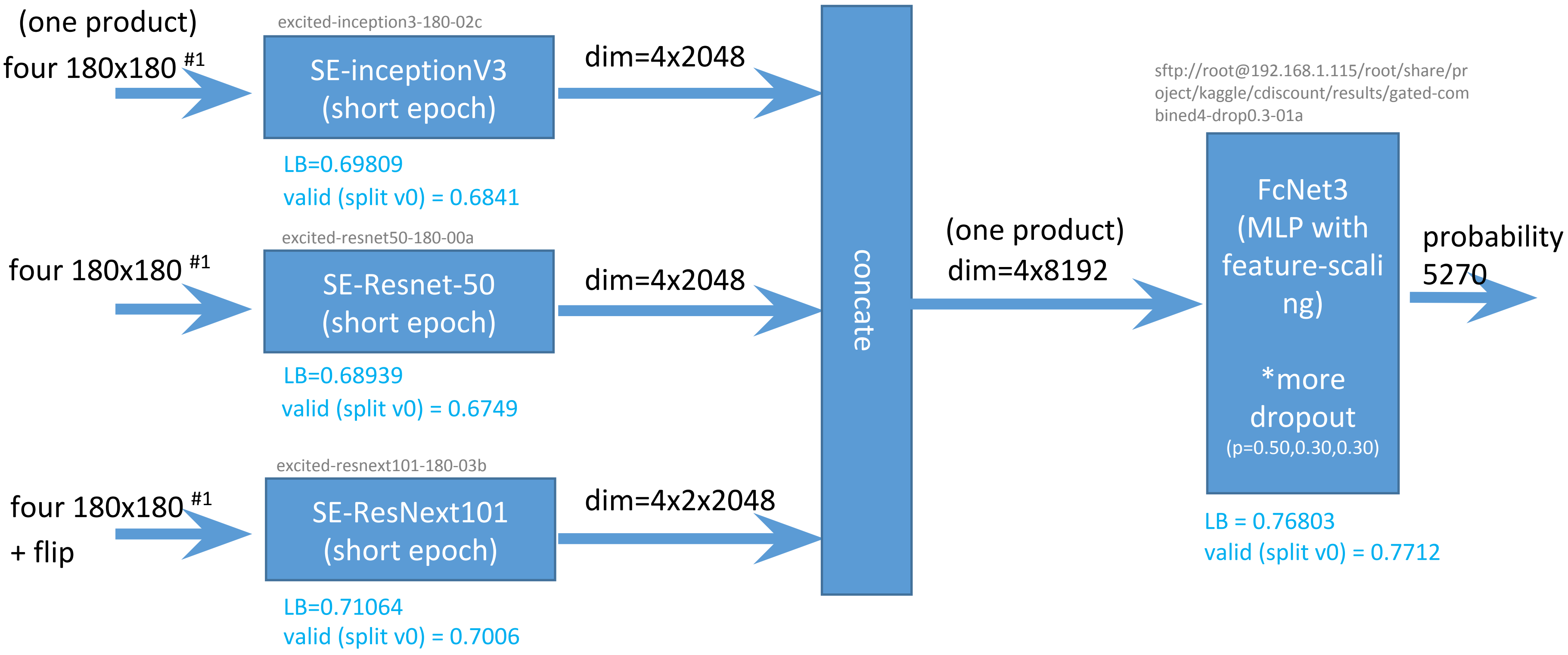


```
##------------------------------------------------------------------
class SEScale(nn.Module):
    def __init__(self, channel, reduction=16):
        super(SEScale, self).__init__()
        self.fc1 = nn.Linear(channel,   reduction)
        self.fc2 = nn.Linear(reduction, channel)

    def forward(self, x):
        x = self.fc1(x)
        x = F.relu(x, inplace=True)
        x = self.fc2(x)
        x = F.sigmoid(x)
        return x
```

N = batch size = 4096
V = num of images per product = 4
C = concat feature dim = 8192

combine.1



(one product)

four 180x180 #1

excited-inception3-180-02c

SE-inceptionV3 (short epoch)

LB=0.69809

valid (split v0) = 0.6841

dim=4x2048

four 180x180 #1

excited-resnet50-180-00a

SE-Resnet-50 (short epoch)

LB=0.68939

valid (split v0) = 0.6749

dim=4x2048

four 180x180 #1

+ flip

excited-resnext101-180-03b

SE-ResNext101 (short epoch)

LB=0.71064

valid (split v0) = 0.7006

dim=4x2x2048

concate

(one product)

dim=4x8192

sftp://root@192.168.1.115/root/share/project/kaggle/cdiscount/results/gated-combined4-drop0.3-01a

FcNet3 (MLP with feature-scaling)

*more dropout

(p=0.50,0.30,0.30)

LB = 0.76803

valid (split v0) = 0.7712

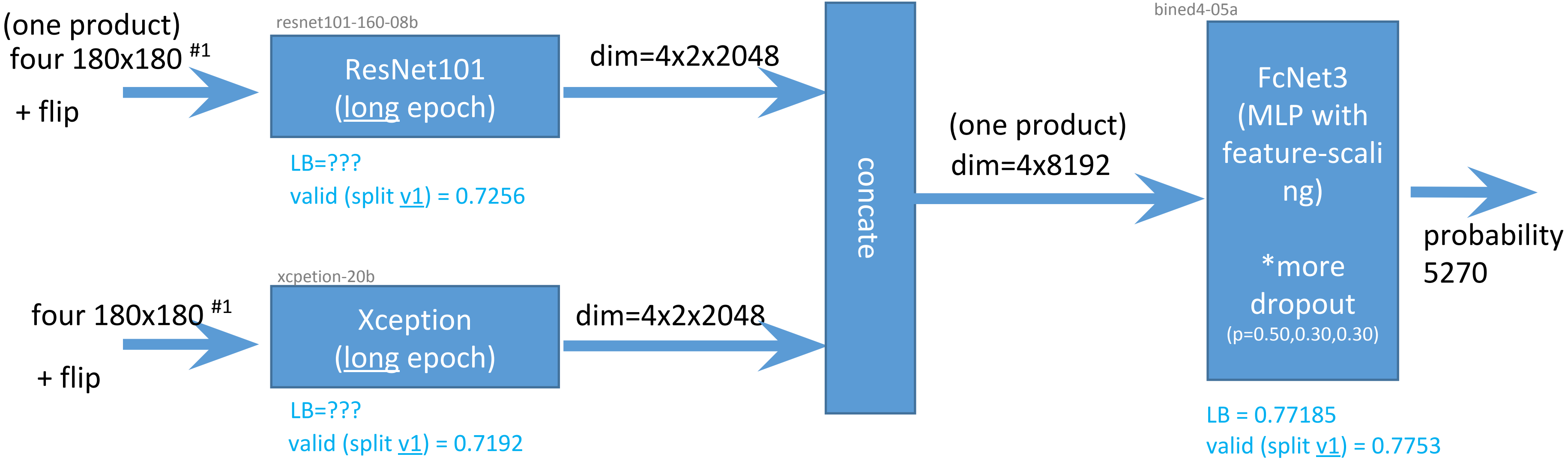
probability

5270

#1 : if the product has less than 4 images, it is stuffed with zero images

combine.2

sftp://root@192.168.1.115/root/share/project/kaggle/cdiscount/results/gated-combined4-05a



(one product)
four 180x180 [#1]
+ flip

resnet101-160-08b

ResNet101
(long epoch)

LB=???
valid (split v1) = 0.7256

dim=4x2x2048

four 180x180 [#1]
+ flip

xcpetion-20b

Xception
(long epoch)

LB=???
valid (split v1) = 0.7192

dim=4x2x2048

concate

(one product)
dim=4x8192

FcNet3
(MLP with feature-scaling)

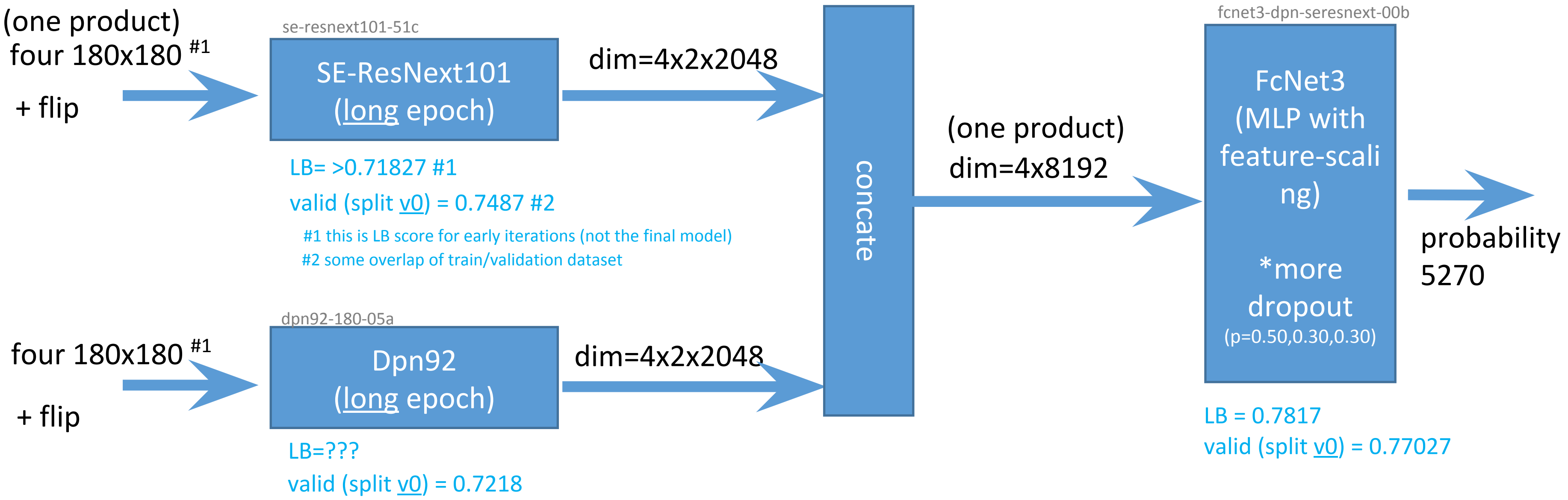
*more dropout
(p=0.50,0.30,0.30)

LB = 0.77185
valid (split v1) = 0.7753

probability
5270

#1 : if the product has less than 4 images, it is stuffed with zero images

combine.5



(one product)
four 180x180 [#1]
+ flip

se-resnext101-51c

SE-ResNext101
(long epoch)

LB= >0.71827 #1
valid (split v0) = 0.7487 #2

#1 this is LB score for early iterations (not the final model)
#2 some overlap of train/validation dataset

dim=4x2x2048

four 180x180 [#1]
+ flip

dpn92-180-05a

Dpn92
(long epoch)

LB=???
valid (split v0) = 0.7218

dim=4x2x2048

concate

(one product)
dim=4x8192

fcnet3-dpn-seresnext-00b

FcNet3
(MLP with feature-scaling)

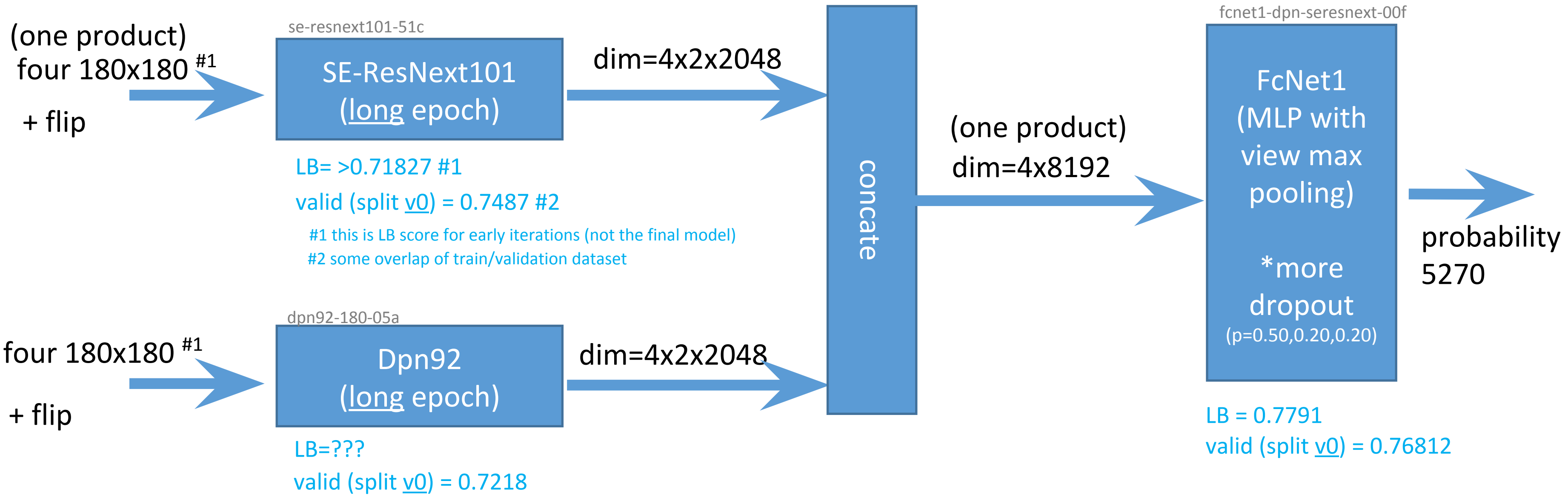
*more dropout
(p=0.50,0.30,0.30)

LB = 0.7817
valid (split v0) = 0.77027

probability
5270

#1 : if the product has less than 4 images, it is stuffed with zero images

combine.5max



(one product)
four 180x180 [#1]
+ flip

se-resnext101-51c

SE-ResNext101
(long epoch)

dim=4x2x2048

LB= >0.71827 #1

valid (split v0) = 0.7487 #2

#1 this is LB score for early iterations (not the final model)
#2 some overlap of train/validation dataset

four 180x180 [#1]
+ flip

dpn92-180-05a

Dpn92
(long epoch)

dim=4x2x2048

LB=???

valid (split v0) = 0.7218

concate

(one product)
dim=4x8192

fcnet1-dpn-seresnext-00f

FcNet1
(MLP with
view max
pooling)

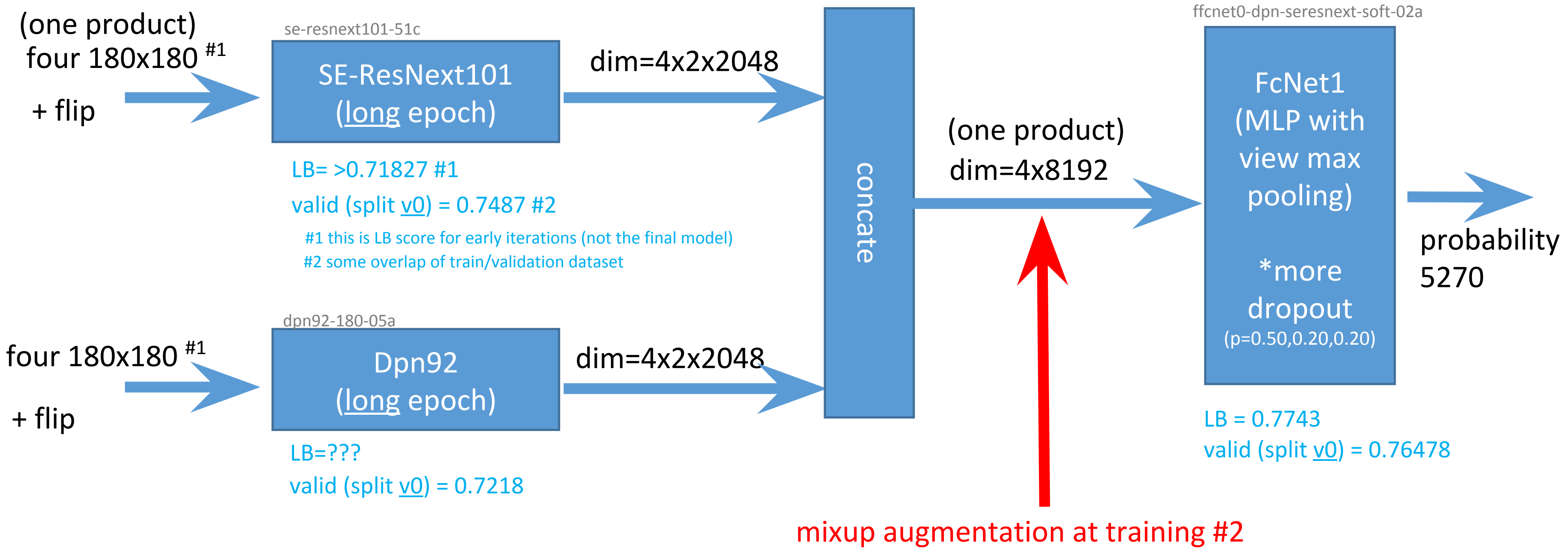
*more
dropout
(p=0.50,0.20,0.20)

probability
5270

LB = 0.7791
valid (split v0) = 0.76812

#1 : if the product has less than 4 images, it is stuffed with zero images

combine.5mix



(one product)
four 180x180 #1
+ flip

se-resnext101-51c

SE-ResNext101
(long epoch)

dim=4x2x2048

LB= >0.71827 #1

valid (split v0) = 0.7487 #2

#1 this is LB score for early iterations (not the final model)
#2 some overlap of train/validation dataset

four 180x180 #1
+ flip

dpn92-180-05a

Dpn92
(long epoch)

dim=4x2x2048

LB=???

valid (split v0) = 0.7218

concate

(one product)
dim=4x8192

mixup augmentation at training #2

ffcnet0-dpn-seresnext-soft-02a

FcNet1
(MLP with
view max
pooling)

*more
dropout
(p=0.50,0.20,0.20)

probability
5270

LB = 0.7743

valid (split v0) = 0.76478

#1 : if the product has less than 4 images, it is stuffed with zero images

#2 : "mixup: Beyond Empirical Risk Minimization" - Hongyi Zhang, ... arvix 2017 (https://arxiv.org/abs/1710.09412)

Azat Davletshin(8th place)

- Finetuned the pre-trained SE-ResNet-50 for about 7 epochs on 161x161 with random crops and horizontal flips
- Found 5 closest (by cosine) pictures from the train for each class
- scores[pic, label] = np.power((1 + search_scores[label, pic, :]) / 2, 35).sum()

  Got 0.752LB
- Finally: SE-ResNet-101(15 ep) + KNN + SE-ResNet-50 + TTA. Got 0.77883LB

n01z3(9th place)

- Train FC adam (LR 0.001-> 0.0001)
  Polishing, only FC SGD (LR 0.001 -> 0.0001)
  Train full net SGD (LR 0.001-> 0.0001)
  Manual annealing SGD (LR 0.01-> 0.001 -> 0.0001) 1-2 epochs at each stage, repeat twice
  Train without augs SGD LR 0.0001
- Resnext-101, Resnext-50, SE-Resnext-50, Resnet-101, Resnet-152 with 160 input size and TTA10
- Random sample from the item + hard negative sampling

Vladimir Iglovikov(12th place)

41

Single model performance

Vladimir Iglovikov 3 months ago

last comment by

steelrose 2mo ago

91

- Resnet 50, 101 and 152, on 160x160 crops, dropping learning rate on the plateau. At the first epoch, all layers except last are frozen
- TTA + geometric mean by model had 0.75 on LB
- Geometric mean of the previous step had 0.77 on LB

Reference

- A summary about our solution:
  https://www.kaggle.com/c/cdiscount-image-classification-challenge/discussion/45733

- The winner's approach:
  https://www.kaggle.com/c/cdiscount-image-classification-challenge/discussion/45863

Contacts

Pavel Ostyakov - pavelosta@gmail.com

Aleksey Kharlamov - axcel@me.com