

Deep Plankton Learning — road to the top

Рыжков Александр

ВМК МГУ

30 марта 2015 года

Содержание

- 1 Постановка задачи
- 2 Прикладные пакеты для Deep Learning
- 3 Amazon Web Services (AWS)
- 4 Построение решения
 - Предобработка данных
 - Обучение моделей
 - Комбинирование моделей
- 5 Выводы

Содержание

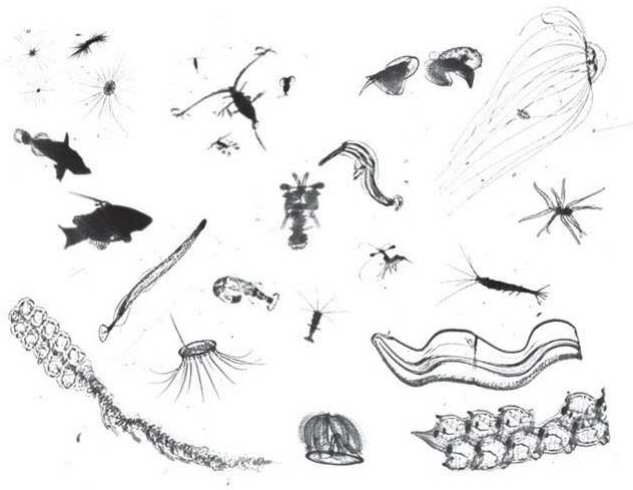
- 1 Постановка задачи
- 2 Прикладные пакеты для Deep Learning
- 3 Amazon Web Services (AWS)
- 4 Построение решения
 - Предобработка данных
 - Обучение моделей
 - Комбинирование моделей
- 5 Выводы

Постановка задачи

- Тренировочная выборка — 30336 изображений
- Тестовая выборка — 130400 изображений
- Таксономическая схема видов планктона
- Функционал — MultiLogLoss:

$$MultiLogLoss = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} \log(p_{ij}) \rightarrow \min_{P_{ij}}$$

Постановка задачи



Результат



Completed • \$175,000 • 1,049 teams

National Data Science Bowl

Mon 15 Dec 2014 – Mon 16 Mar 2015 (10 days ago)

Dashboard

Private Leaderboard - National Data Science Bowl

This competition has completed. This leaderboard reflects the final standings.

See someone using multiple accounts?
Let us know.

#	Rank	Team Name <small>↗ In the money</small>	Score <small>🏆</small>	Entries	Last Submission UTC (Best - Last Submission)
1	—	🇷🇺 Deep Sea 🇷🇺 🇺🇸 *	0.565971	96	Mon, 16 Mar 2015 17:32:03
2	—	Happy Lantern Festival 🇺🇸 *	0.580300	150	Mon, 16 Mar 2015 14:03:40 (-2.8h)
3	—	Poisson Process 🇺🇸 *	0.587967	134	Mon, 16 Mar 2015 23:54:04 (-0.9h)
4	🇺🇸	Junonia	0.604112	59	Mon, 16 Mar 2015 20:24:47 (-19.5h)
5	—	🇺🇸 Deepsea Challenger 🇺🇸 🇺🇸	0.606921	142	Mon, 16 Mar 2015 19:47:18 (-3.3h)
6	🇺🇸	AuroraXie	0.607333	32	Mon, 16 Mar 2015 22:15:07
7	🇺🇸	Maxim Milakov	0.610072	27	Sat, 14 Mar 2015 18:36:14 (-0.8h)
8	🇺🇸	Ilya Kostrikov	0.610964	51	Mon, 16 Mar 2015 20:47:09 (-0.6h)
9	—	old-ufo	0.613981	82	Mon, 16 Mar 2015 21:23:27
10	🇺🇸	🇺🇸 nagadomi	0.617123	31	Mon, 16 Mar 2015 05:57:34 (-0.4h)
11	🇺🇸	🇺🇸 zzspar	0.619674	174	Mon, 16 Mar 2015 18:47:59 (-0.2h)
12	🇺🇸	🇺🇸 Biolab 🇺🇸	0.619722	74	Mon, 16 Mar 2015 14:50:21 (-3.2d)
13	🇺🇸	Alexander Ryzhkov (MSU, Moscow, Russia)	0.620708	67	Mon, 16 Mar 2015 23:26:23 (-2.9h)
14	🇺🇸	belle	0.623529	28	Mon, 16 Mar 2015 21:01:15 (-0.2h)
15	🇺🇸	harmug	0.625620	28	Mon, 16 Mar 2015 20:42:44 (-4.9h)

Содержание

- 1 Постановка задачи
- 2 Прикладные пакеты для Deep Learning
- 3 Amazon Web Services (AWS)
- 4 Построение решения
 - Предобработка данных
 - Обучение моделей
 - Комбинирование моделей
- 5 Выводы

Прикладные пакеты

- Caffe
- CXXNET v.2.0
- Torch (Lua)
- Theano
- Matconvnet
- Cuda-convnet2
- Nolearn/Lasagne
- ...

Caffe

- Разработчик — Berkeley Vision and Learning Center (BVLC)
- Удобные утилиты для создания баз изображений, среднего изображений
- Классификации тестовых изображений:
 - `predict.py`
 - `classification.ipynb`
- Простое конфигурирование архитектуры сети и параметров обучающего алгоритма

CXXNET

- Разработчики — Bing Xu и Tianqi Chen
- Удобные утилиты для создания баз тренировочных и **тестовых** изображений
- Классификации тестовых изображений проходит сразу для всей базы
- Простое конфигурирование архитектуры сети и параметров обучающего алгоритма

Содержание

- 1 Постановка задачи
- 2 Прикладные пакеты для Deep Learning
- 3 Amazon Web Services (AWS)**
- 4 Построение решения
 - Предобработка данных
 - Обучение моделей
 - Комбинирование моделей
- 5 Выводы

Amazon Web Services (AWS)



- Сервис с мощными серверами за небольшие деньги
- Можно сразу взять созданный и настроенный образ (AMI)
- Меньшая нагрузка на свое железо
- Есть некоторые риски :)

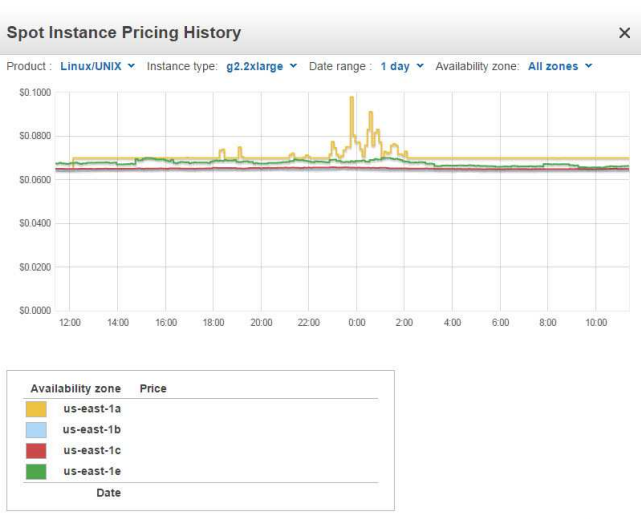
Типы инстансов

- T2 — с повышаемой производительностью
- M3 — сбалансированный вариант
- C4(3) — вычислительные нагрузки
- R3 — больше памяти
- G2 — есть GPU
- I2 — оптимизированные диски (SSD)
- HS1 — высокая плотность хранения данных

Spot vs. Standard инстанс

- Spot:
 - Меньше стоят
 - Могут быть отключены в ЛЮБОЙ момент по цене
- Standard:
 - Стоят ≈ 10 раз больше
 - Отключаются пользователем
 - I2 есть только Standard

Spot история цен



Tips & Tricks

- Чем больше заявок — тем больше серверов
- Поставим цену больше изначально
- Стоит выбирать наиболее «спокойную» зону
- Сохранение диска при отключении сервера!
- Запуск задач на сервере:
 - Putty
 - Pytty + nohup + &

Содержание

- 1 Постановка задачи
- 2 Прикладные пакеты для Deep Learning
- 3 Amazon Web Services (AWS)
- 4 Построение решения**
 - Предобработка данных
 - Обучение моделей
 - Комбинирование моделей
- 5 Выводы

Предобработка данных

Объекты — изображения:

- Цвет
- Геометрия
- Морфология
- Комбинирование изображений

Варианты обработки:

- Online
- Offline

Моя предобработка

- Offline-трансформация
- Один размер с сохранением AR и без
- Повороты: 45° , 30° , 15° каждый раз к исходному файлу
- Морфология (эрозия + дилатация в каналах G и B)

Предобработка: победители

- Realtime-трансформация
- Повороты
- Сдвиги
- 3D трансформации
- Отражения
- Масштабирование

Предобработка: 10-е место

- Realtime-трансформация
- Переход к негативу
- Повороты
- Сдвиги
- 3D трансформации
- Отражения
- Масштабирование

Обучение моделей

- Обязательное использование мощных GPU
- Победители: GTX 980, GTX 680 и Tesla K40 (12GB)
- 10-е место: x20 GRID K520 AWS
- Мои GPU: x2 GRID K520 AWS, GTX 650M

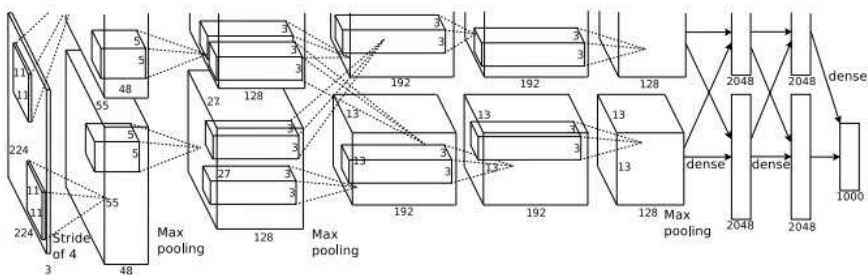
Мои модели

- 8–10 различных моделей CXXNET (96×96)
- 8–10 моделей типа AlexNet (256×256)
- Последовательное добавление слоев Dropout с увеличением threshold
- Уменьшение LearningRate для максимально близкого подхода к минимуму функционала

Модель CXXNET

- conv4-96
- maxpool
- conv3-192
- conv3-256
- maxpool
- flatten
- FC-2048
- FC-1024
- FC-121
- Soft-max

AlexNet





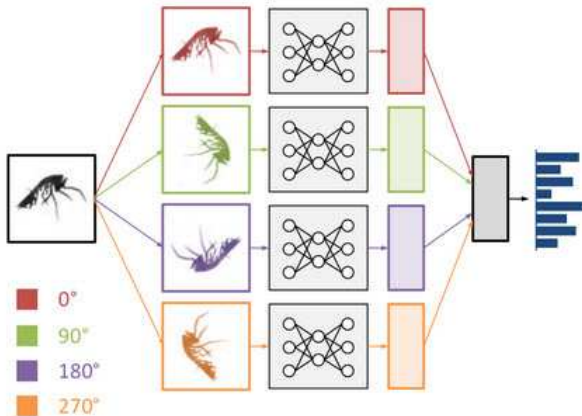
Модели победителей

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 x 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

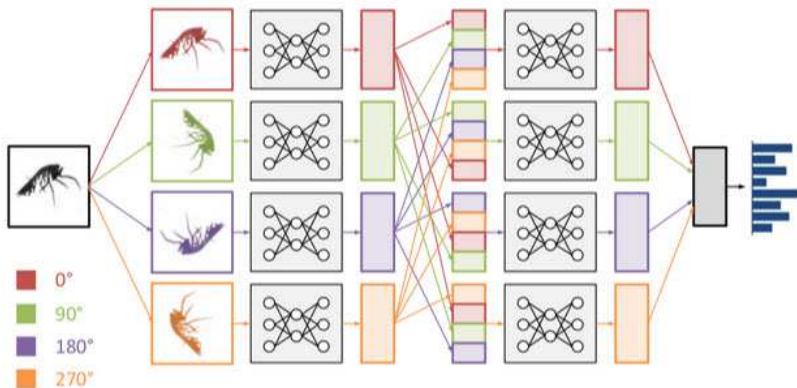
Table 2: Number of parameters (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Модели победителей



Модели победителей



Модели победителей

- Leaky ReLU:

$$y = \max(x, ax)$$

- Замена MaxPool на RootMeanSquarePool
- Разбиение выборки 90/10
- Использование связки CPU+GPU
- 3 стадии обучения с понижающимся LearningRate
- Попытка использования Pseudo-Labels
- Добавление признаков изображения внутрь сети

Модели победителей

Layer type	Size	Output shape
cyclic slice		(128, 1, 95, 95)
convolution	32 3x3 filters	(128, 32, 95, 95)
convolution	16 3x3 filters	(128, 16, 95, 95)
max pooling	3x3, stride 2	(128, 16, 47, 47)
cyclic roll		(128, 64, 47, 47)
convolution	64 3x3 filters	(128, 64, 47, 47)
convolution	32 3x3 filters	(128, 32, 47, 47)
max pooling	3x3, stride 2	(128, 32, 23, 23)
cyclic roll		(128, 128, 23, 23)
convolution	128 3x3 filters	(128, 128, 23, 23)
convolution	128 3x3 filters	(128, 128, 23, 23)
convolution	64 3x3 filters	(128, 64, 23, 23)
max pooling	3x3, stride 2	(128, 64, 11, 11)
cyclic roll		(128, 256, 11, 11)
convolution	256 3x3 filters	(128, 256, 11, 11)
convolution	256 3x3 filters	(128, 256, 11, 11)
convolution	128 3x3 filters	(128, 128, 11, 11)
max pooling	3x3, stride 2	(128, 128, 5, 5)
cyclic roll		(128, 512, 5, 5)
fully connected	512 2-piece maxout units	(128, 512)
cyclic pooling (rms)		(32, 512)
fully connected	512 2-piece maxout units	(32, 512)
fully connected	121-way softmax	(32, 121)

Комбинирование моделей

- Среднее арифметическое:

$$Y_{answer} = \frac{y_1 + y_2 + \dots + y_n}{n}$$

- Среднее геометрическое (теряется нормировка):

$$Y_{answer} = \sqrt[n]{y_1 \cdot y_2 \cdot \dots \cdot y_n}$$

Комбинирование моделей

- У меня:
 - Начало — среднее арифметическое (2–5 моделей)
 - Середина — среднее геометрическое (6–14 моделей)
 - Конец — среднее арифметическое (15–20 моделей)
- Победители и 10-е место использовали взвешенную линейную комбинацию вида

$$Y_{answer} = w_1y_1 + w_2y_2 + \dots + w_ny_n + Const,$$

посчитанные по небольшой валидационной выборке (5% от всех данных)

Комбинирование моделей

Еще больше идей:

- Использование комбинации моделей, обученных на данных с сохранением AR и без него
- Иерархическая структура комбинации:
 - 4 уровня иерархии
 - + Тренировка сетей-специалистов на меньшем числе меток
 - Простота качества сопоставим с обычными методами комбинирования
 - Плохое взаимодействие с классами Unknown и Others

Содержание

- 1 Постановка задачи
- 2 Прикладные пакеты для Deep Learning
- 3 Amazon Web Services (AWS)
- 4 Построение решения
 - Предобработка данных
 - Обучение моделей
 - Комбинирование моделей
- 5 Выводы**

Выводы

- Участвовать весело и интересно, но затратно по времени + \$\$\$
- Необходимо пересмотреть все видео с лекциями от Hinton & Fergus
- Осилить множество статей с возможно-полезными результатами
- Dropout is awesome!

Выводы



Completed • \$175,000 • 1,049 teams

National Data Science Bowl

Mon 15 Dec 2014 – Mon 16 Mar 2015 (10 days ago)

Dashboard

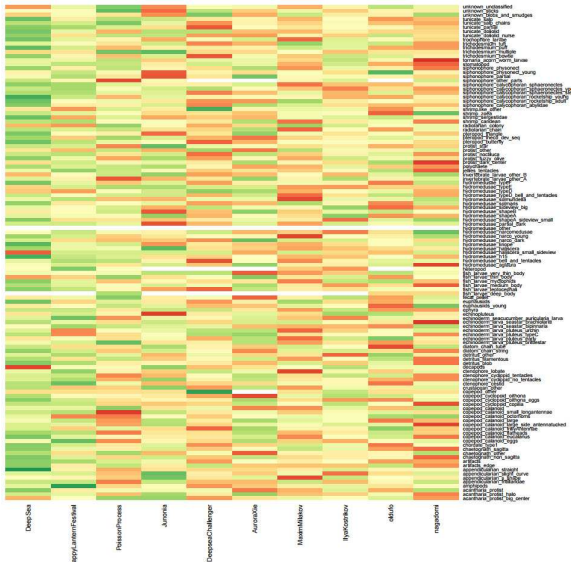
Private Leaderboard - National Data Science Bowl

This competition has completed. This leaderboard reflects the final standings.

See someone using multiple accounts?
Let us know.

#	Rank	Team Name <small>↗ in the money</small>	Score <small>🏆</small>	Entries	Last Submission UTC (Best - Last Submission)
1	—	🇷🇺 Deep Sea 🇷🇺 🇺🇸 *	0.565971	96	Mon, 16 Mar 2015 17:32:03
2	—	Happy Lantern Festival 🇺🇸 *	0.580300	150	Mon, 16 Mar 2015 14:03:40 (-2.8h)
3	—	Poisson Process 🇺🇸 *	0.587967	134	Mon, 16 Mar 2015 23:54:04 (-0.9h)
4	🇺🇸	Junonia	0.604112	59	Mon, 16 Mar 2015 20:24:47 (-19.5h)
5	—	🇮🇳 Deepsea Challenger 🇮🇳	0.606921	142	Mon, 16 Mar 2015 19:47:18 (-3.3h)
6	🇺🇸	AuroraXie	0.607333	32	Mon, 16 Mar 2015 22:15:07
7	🇺🇸	Maxim Milakov	0.610072	27	Sat, 14 Mar 2015 18:36:14 (-0.8h)
8	🇺🇸	Ilya Kostrikov	0.610964	51	Mon, 16 Mar 2015 20:47:09 (-0.6h)
9	—	old-ifo	0.613981	82	Mon, 16 Mar 2015 21:23:27
10	🇺🇸	🇮🇳 nagadomi	0.617123	31	Mon, 16 Mar 2015 05:57:34 (-0.4h)
11	🇺🇸	🇮🇳 zzspar	0.619674	174	Mon, 16 Mar 2015 18:47:59 (-0.2h)
12	🇺🇸	🇮🇳 Biolab 🇮🇳	0.619722	74	Mon, 16 Mar 2015 14:50:21 (-3.2d)
13	🇺🇸	Alexander Ryzhkov (MSU, Moscow, Russia)	0.620708	67	Mon, 16 Mar 2015 23:26:23 (-2.9h)
14	🇺🇸	belle	0.623529	28	Mon, 16 Mar 2015 21:01:15 (-0.2h)
15	🇺🇸	harmug	0.625620	28	Mon, 16 Mar 2015 20:42:44 (-4.9h)

Выводы



Ссылки

- Блог победителей:
<http://benanne.github.io/2015/03/17/plankton.html>
- 10-е место Nagadomi:
<https://github.com/nagadomi/kaggle-ndsb>
- Другие идеи и решения:
<http://www.kaggle.com/c/datasciencebowl/forums/t/12930>

Спасибо за внимание!!