**Отчет об исследовании возможности обработки флюорографий с помощью сверточных искусственных нейронных сетей.**

**Общее описание проблемы.**

Обработка медицинских данных средствами машинного обучения и искусственных нейронных сетей (ИНС) в частности, является одной из наиболее актуальных исследовательских задач на текущий момент. Тем не менее, данная область имеет ряд специфических проблем, которые не позволяют быстро добиться высоких результатов методами, отлично зарекомендовавшими себя при решении схожих задач из других направлений.

В данной работе, в частности, будет рассмотрена классификации флюорографий на предмет наличия, или отсутствия патологий.

Наиболее острой проблемой является количество и качество доступных данных. Получить массив обезличенной медицинской информации в России - нетривиальная задача, т.к. количество учреждений, заинтересованных в подобных исследованиях, не столь велико, а выгрузка и очистка данных из специализированного программного обеспечения требует участия в процессе специалиста-рентгенолога.

Выгруженные подобным способом изображения имеют высокое разрешение, что безусловно необходимо профильным специалистам для выявления незаметных на первый взгляд патологий, но требует значительных вычислительных ресурсов при обработке и приводит к существенной потере информации при уменьшении разрешения изображений.

Еще одной проблемой является специфичность самих изображений. Они не похожи на изображения большинства объектов реального мира и выдаются рентгеновским аппаратом в режиме grayscale. Это снижает шанс получения приемлемых результатов с использованием предобученных ИНС. В качестве примера, предобученная на датасете Imagenet сеть resnext-50, используемая в данной работе, выдает точность (accuracy) более 98% при [идентификации пород собак](https://www.kaggle.com/c/dog-breed-identification), но лишь 67% при попытке отличить флюорографию с наличием патологий от чистого снимка.

**Постановка задачи.**

Для постановки задачи были проведены консультации с двумя профильными специалистами из 9 ЛДЦ МО РФ.

Была выявлена следующая проблематика: молодые специалисты-рентгенологи не всегда могут выявить патологию на снимке, в то время как некоторые врачи привыкли полностью полагаться на описание снимка, сделанное рентгенологом, не проводя самостоятельный анализ снимка. Для снижения вероятности врачебной ошибки, в помощь молодым специалистам необходимо программное обеспечение, указывающее на отклонение снимка от нормы.

Было рассмотрено два основных сценария: модель, пытающаяся предсказать конкретное заболевание, или модель, указывающая на наличие любого отклонения. Оба специалиста сошлись во мнении, что второй вариант предпочтительней, при условии достаточно высокой точности программного продукта.

На второй итерации данного исследования, в связи с невозможностью обучения модели на четырнадцати патологиях, совмещенных в один класс, было решено обучать ИНС на 15 классах, а сравнение вероятностей наличия патологий, или их отсутствия, оставить на постобработку в готовом программном продукте.

Побочной целью данной работы являлось тестирование альфа-версии библиотеки fastai, по заявлению авторов, использующей самые современные методы и тенденции в DL.

**Обзор аналогичных решений.**

Самые высокие результаты на данный момент показывает 121-слойная сверточная ИНС CheXNet, созданная специалистами Стэнфордского университета на базе DenseNet, обученная на используемом в данной работе датасете ChestX-ray 14. Подробная [статья выложена на arxiv.org](https://arxiv.org/pdf/1711.05225.pdf). К сожалению, обученная модель, как и подробное описание процесса создания архитектуры, отсутствуют в открытом доступе. Остальные найденные в сети работы, или имеют [более низкую точность](https://github.com/gregwchase/nih-chest-xray/blob/master/README.md#model-structured-data), или используют [более старый датасет](https://openi.nlm.nih.gov/gridquery.php?q=&it=x,xg&sub=x&m=1&n=101), без четкой разметки, содержащий гораздо меньшее число снимков и, как следствие, тоже [показывают невысокую точность](https://github.com/bendidi/X-ray-classification).

**Данные.**

В работе был использован датасет [NIH Chest X-ray](https://www.kaggle.com/nih-chest-xrays/data) (так же известный как ChestX-ray 14), появившийся в сети в октябре 2017-го. В данном датасете содержится 112,120 снимков с разрешением 1024 x 1024 и файл с описанием. В файле каждый снимок имеет одну, или более метку из пятнадцати возможных, представляющих различные патологии, или отсутствие всех патологий. В аннотации указано, что метки были получены средствами NLP и могут содержать ошибки, но заявлена точность более 90%. Данное утверждение, однако, спорно. При визуальном анализе и сравнении с метками классов, отраслевой эксперт [указывает на низкое качество разметки](https://lukeoakdenrayner.wordpress.com/2017/12/18/the-chestxray14-dataset-problems/), наличие структурированного шума (structured noise), что приводит к неверному обучению ИНС.

В файле описания так же содержаться другие данные (пол, возраст и т.д.), что позволяет провести [эксплоративный анализ данных](https://www.kaggle.com/sbernadac/lung-deseases-data-analysis). Из него становится ясно, что классы не сбалансированы.

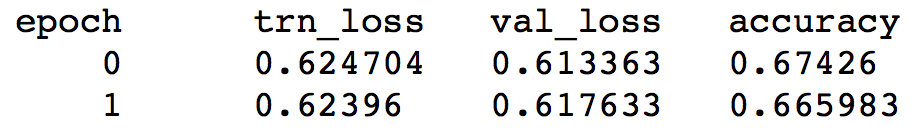
**Предобработка данных.**

Предобработка на первой итерации свелась к удалению лишних столбцов, а также объединению всех снимков, имеющих патологии в одну группу и чистых снимков в другую. На валидацию было выделено 22424 снимка (20%), выбранных случайным образом (см. раздел Data preprocessing, train/val split).

На второй итерации патологии не были объединены в одну группу, в остальном процесс прошел без изменений.

**Метрики**

В качестве функции потерь была использована categorical cross entropy. Используемая библиотека отображает потери на обучающей и тестовой выборке, а также точность (accuracy) на каждой эпохе, что позволяет динамически отслеживать процесс обучения. Данная методика позволяет вовремя выявить переобучение модели (overfitting), а также оценить общий прогресс обучения.

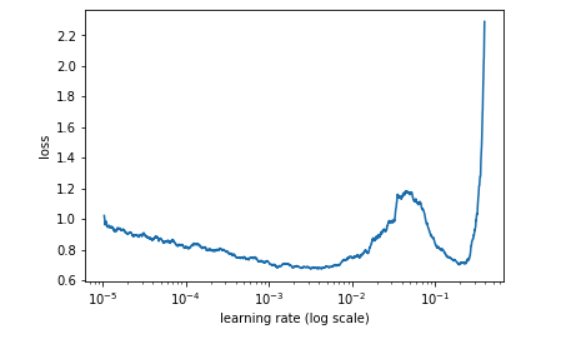
  
В качестве финальной метрики, на первой итерации предполагалось использование точности (accuracy) для бинарной классификации. На второй итерации, когда проблема была сформулирована как multilabel classification с пятнадцатью категориями, было принято решение использовать ROC AUC для каждой отдельной категории. Это позволило сравнить результат с аналогичными решениями, так как они использовали эту метрику.

**Методика решения**

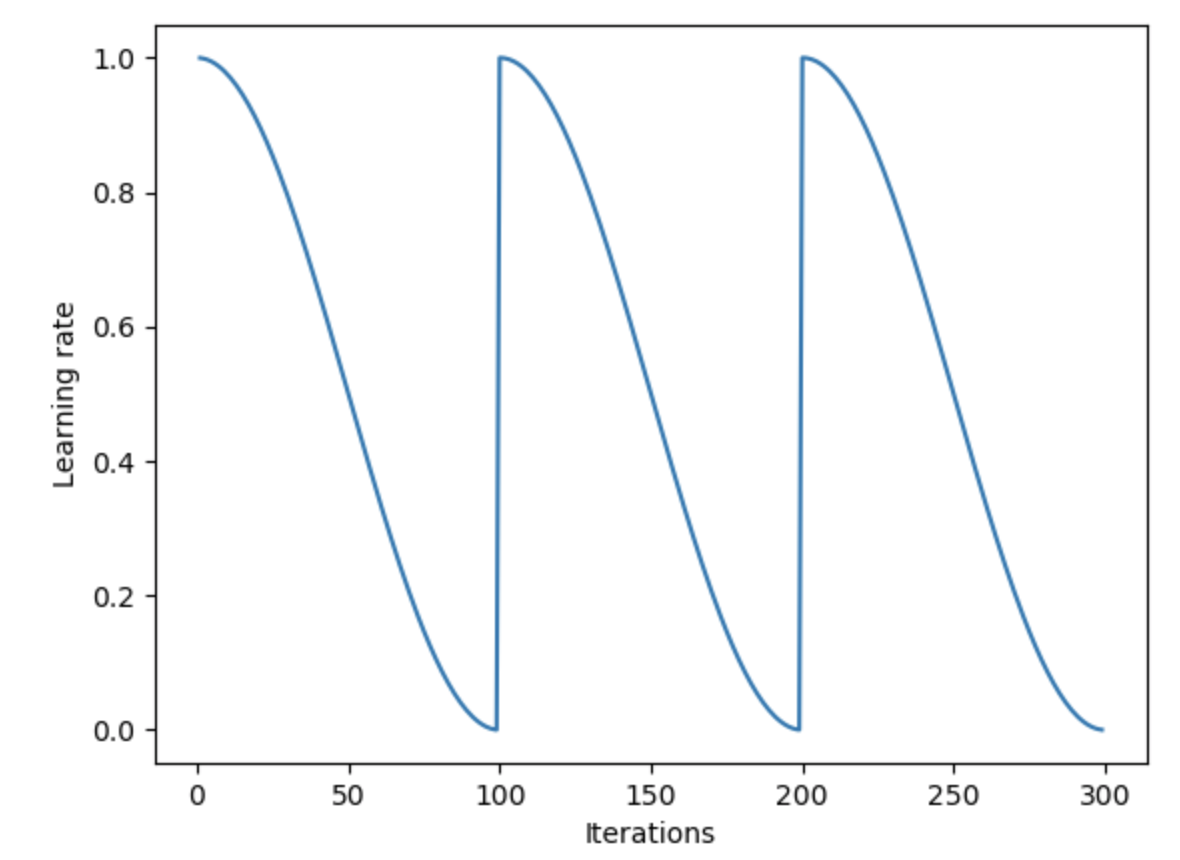
В качестве платформы был выбран AWS P2.xlarge. Это позволило обрабатывать изображения, уменьшенные до 256х256, с batch size размером 32. Уменьшение изображения до 128х128 давало ухудшение accuracy на 4-5%, а увеличение до 512х512 приводило к нехватке видеопамяти даже при уменьшении batch size.

Для создания классификатора была использована библиотека [fastai](https://github.com/fastai/fastai), являющаяся надстройкой над фреймворком [pytorch](http://pytorch.org/).

Это позволило протестировать несколько относительно новых техник, применяемых в Deep Learning.

 Данная библиотека значительно упрощает поиск оптимальной learning rate (LR) для модели, визуально показывая возрастание и убывание функции потерь при постепенном увеличении LR.

Так же, по умолчанию используется методика [Stochastic Gradient Descent with Warm Restarts](https://arxiv.org/abs/1608.03983), что позволяет автоматически уменьшать LR по мере обучения ИНС, но в то же время выходить из локальных минимумов.



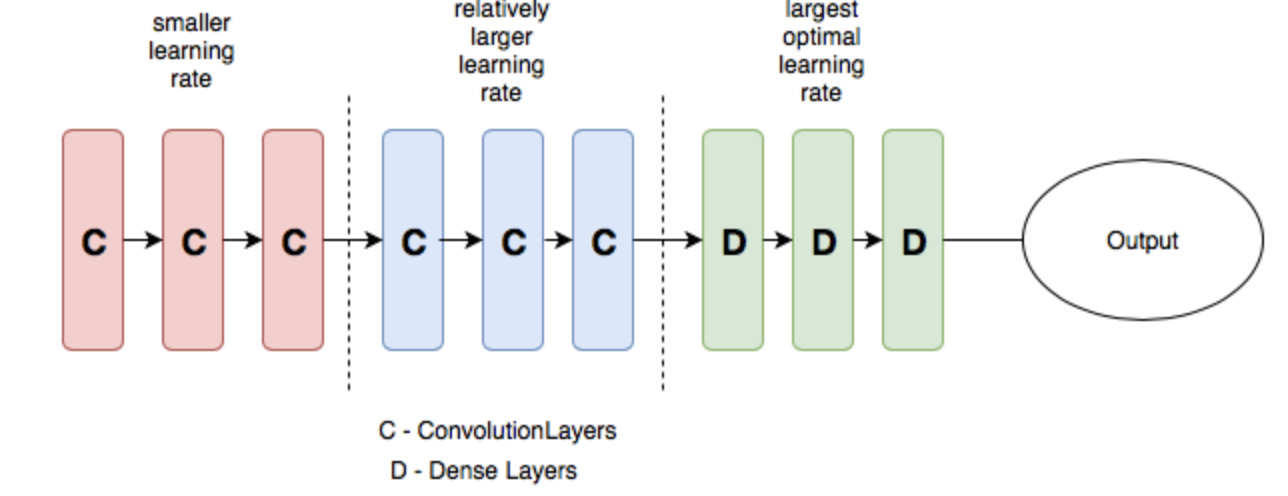
По аналогии с библиотекой Keras (applications), имеется некоторое количество «встроенных» моделей, мной были протестированы resnet34, resnext50 и resnext101.

На первой итерации исследования изменение модели, как и гиперпараметров, не давало ощутимого роста точности (accuracy). Data augmentation в виде увеличения изображения на 10%, наклонов в разные стороны на 5 градусов и смешения на 5% в разные стороны не дало ни прироста точности, ни уменьшения переобучения модели.

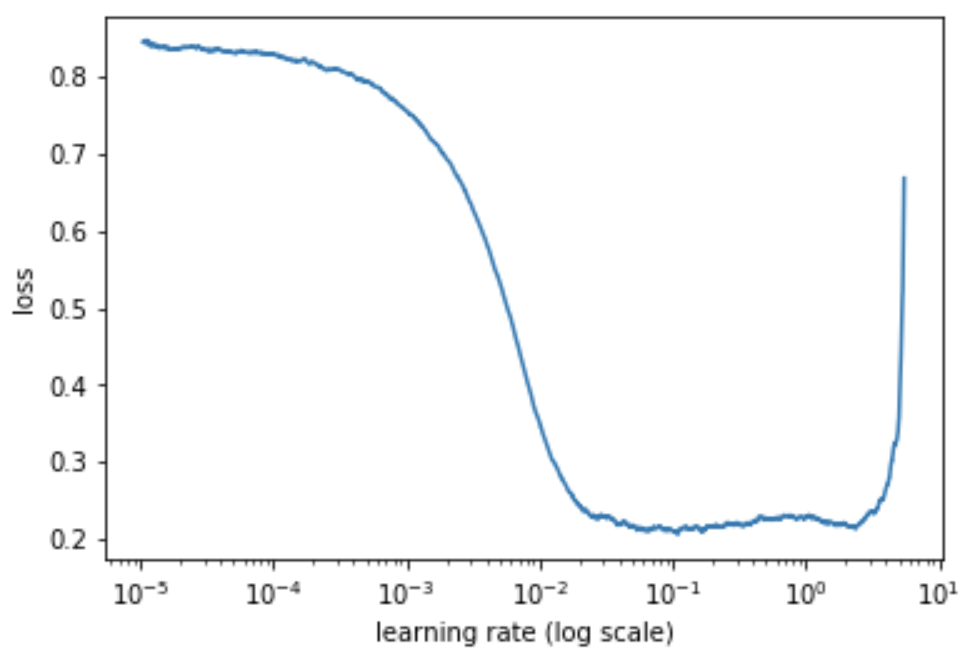
 В первой итерации, модель была основана на архитектуре [resnext50](https://github.com/facebookresearch/ResNeXt/blob/master/README.md#introduction). 6 эпох файнтюнинга модели дало accuracy 0.682. Следующим этапом обучения ИНС являлась техника differential learning rate, позволяющая обучать разные слои сети с разной скоростью. Как хорошо видно из статьи Matthew Zeiler [Visualizing and Understanding Convolutional Networks](https://arxiv.org/pdf/1311.2901.pdf), разные слои (группы слоев, в нашем случае) несут разную семантику. Если первые слои предобученной на датасете Imagenet ИНС позволяют находить линии, окружности, углы и несложные паттерны и будут не сильно изменены в процессе переобучения сети, то более поздние слои нуждаются в доработке. Библиотека fastai позволяет разделять ИНС на 3 группы

слоев, где первая группа – слои, отвечающие за базовые структуры изображения, а последняя – fully connected dense layer(s).

Таким образом, выставляя разный LR для разных групп, можно контролировать процесс обучения. В моем случае массив LR выглядел как [LR/9, LR/3, LR].



Поскольку результат 0.682% был очень далек от желаемого, на второй итерации было решено попробовать не объединять все патологии в один класс, а тренировать ИНС на 15 категориях, с возможностью присутствия нескольких категорий на одном снимке. Это дало резкий прирост качества обучения, точность составила 93%. Из этого можно сделать вывод, что ИНС не смогла обобщить патологии в одну категорию. Так же, стало заметно, что поиск learning rate выдал гораздо более стабильные результаты.

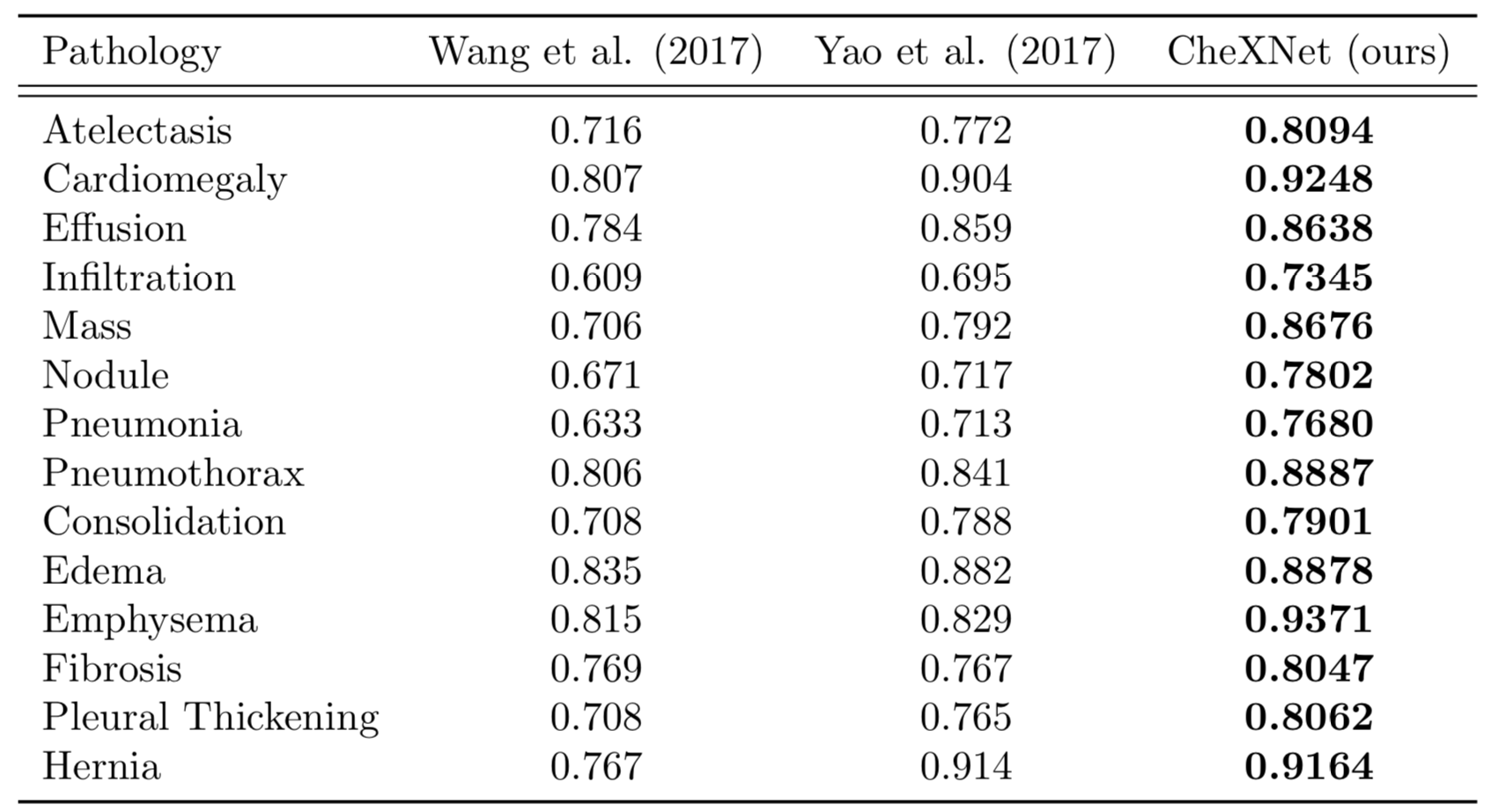
Обучение на второй итерации началось сразу с «разморозки» весов всех слоев, коэффициенты LR для групп слоев остались прежними, [LR/9, LR/3, LR]. В качестве архитектуры ИНС была выбрана resnet34 как компромисс между количеством обучаемых параметров и скоростью обучения. Обучение длилось шесть эпох и заняло чуть менее семи часов, но судя по соотношению training set loss и validation set loss, переобучения модели (overfitting) еще не наступило (что явилось следствием, в том числе, примененного data augmentation) и, при желании, модель можно было бы еще улучшить, увеличив количество эпох.

Так же, библиотека fastai позволяет использовать методику Test Time Augmentation (TTA), при которой предсказание класса тестового изображения берется, как среднее от пяти вариантов одного изображения, с примененным data augmentation.

**Оценка модели.**

Полученные результаты имеет смысл сравнивать с аналогичными решениями, приведенными в статье [CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning](https://arxiv.org/pdf/1711.05225.pdf)

Считаю важным подчеркнуть, что обученная мной ИНС гораздо проще по архитектуре, быстро обучается и не требует серьезных аппаратных ресурсов.



Результаты обученной мной ИНС:

Atelectasis: 0.800

Cardiomegaly: 0.895

Effusion: 0.875

Infiltration: 0.707

Mass: 0.815

Nodule: 0.724

Pneumonia: 0.724

Pneumothorax: 0.874

Consolidation: 0.782

Edema: 0.892

Emphysema: 0.876

Fibrosis: 0.809

Pleural Thickening: 0.77

Hernia: 0.822

No pathology: 0.773

**Выводы.**

Хотя результаты обученной мной ИНС и не позволяют сразу использовать ее в программных продуктах, изменение архитектуры сети на более сложную, использование изображений с большим разрешением, а также увеличение количества эпох вполне могут позволить создать продукт с высокой точностью выявления патологий.

Второй важный момент – комбинация использования облачных технологий и применения в современных библиотеках (fastai в частности) передовых разработок в области DL, позволяет добиться приемлемых результатов, не имея особой подготовки в области машинного обучения, не обладая экспертными навыками в отрасли применения программного продукта и не обладая серьезными финансовыми ресурсами на создание своей инфраструктуры.

**Источники информации**

[Fast.ai MOOC](http://course.fast.ai/)

[CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning](https://arxiv.org/pdf/1711.05225.pdf)

[Эксплоративный анализ датасета на Kaggle](https://www.kaggle.com/sbernadac/lung-deseases-data-analysis)

[Exploring the ChestXray14 dataset: problems](https://lukeoakdenrayner.wordpress.com/2017/12/18/the-chestxray14-dataset-problems/)

[Stochastic Gradient Descent with Warm Restarts](https://arxiv.org/abs/1608.03983)

[Visualizing and Understanding Convolutional Networks](https://arxiv.org/pdf/1311.2901.pdf)