



Методы Машинного Обучения в Высокоэнергитической Физике

Для защиты магистерской диссертации,
Долгопрудный, Июнь 2017

Научный руководитель - Андрей Устюжанин

Михаил Усвяцов

Московский Физико-Технический Институт
Skolkovo Institute of Science and Technology

Эксперимент CRAYFIS

Cosmic RAYs с помощью смартфонов

CRAYFIS предлагает использовать смартфоны для наблюдения за Ультра Высоэнергитическим Космическим Излучением (УВКИ):

- › Высокие энергии: $> 10^{18}$ eV;
- › Распределенная обсерватория;
- › Камера - детектор космического излучения;
- › Скопления мобильных телефонов в качестве детектора душ.

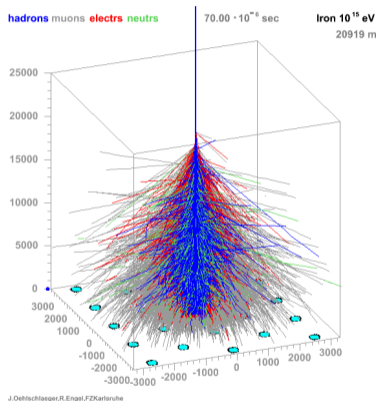


Иллюстрация душа порождённого ионом железа с энергией 1 PeV, CORSIKA симуляция, J. Oehlschläger and R. Engel.

Трудности

CRAYFIS:

- › Ожидается низкая частота событий:
 - › Шумовое космическое излучение: менее раза в минуту на km^2 ;
 - › УВКИ: менее раза в год на km^2 ;
- › Длительность душа от УФКИ менее микросекунд;
- › Необходима высокая частота съёмки ($\sim 10 \text{ Hz}$).

Мобильные телефоны:

- › Малые вычислительные ресурсы;
- › Ограниченные размеры хранилища.

Примеры фотонных треков

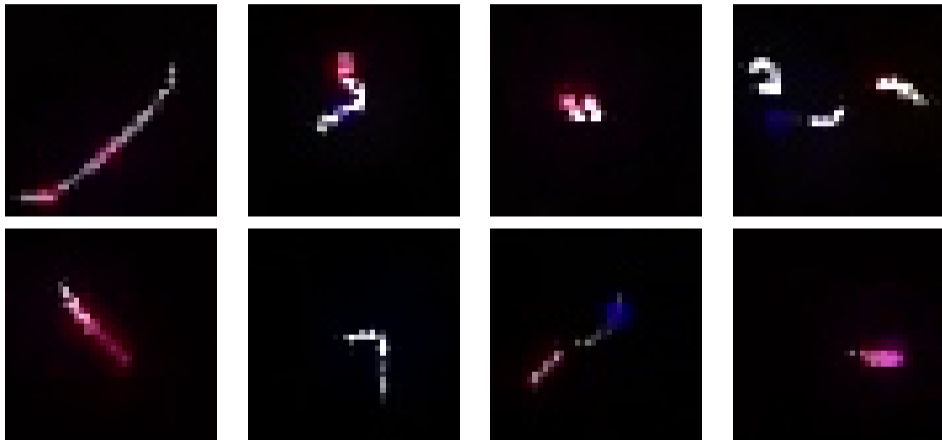


Рис.: Изображения получены облучением сенсора мобильного телефона пучком радиоактивных частиц. Изображения 40×40 пикселей.

Мюоны

Известно что:

Мюоны оставляют треки на уровне сравнимом с собственным шумом матрицы.

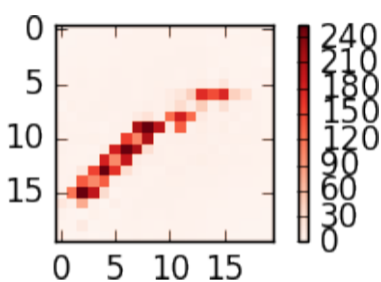
Проблема:

- › Простая стратегия (порог яркости) нечувствительна к мюонам, то есть:
 - › Небольшая точность: большое потребление памяти и сетевых ресурсов
 - › Небольшая чувствительность: потери сигнала

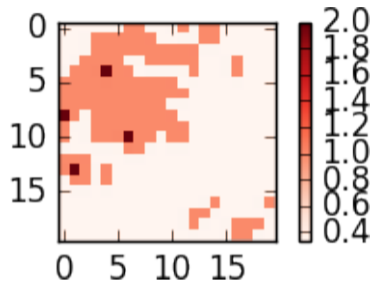
Предлагается:

Распознавание формы.

Пример шума



(a) Фотон



(b) Шум

Рис.: Примеры фотонного и шумового событий. Изображения получены облучением сенсора мобильного телефона пучком радиоактивных частиц.

Формальная постановка

Обучающая выборка

- › \mathcal{X} – пространство изображений $N \times M$
- › $\{(X^k, Y^k) \mid k \in [1, K]\} \in \mathcal{X} \times \{0, 1\}^{N \times M}$ – K тренировочных пар
- › $\gamma := \sum_{kij} y_{ij}^k / (kNM) \ll 1$ – среднее количество 1 в метках Y^k

Обучение

функция: $f : \mathcal{X} \rightarrow [0, 1]^{N \times M}$,
решающее правило: $1[f_{ij}(X) > \tau]$

Loss function

$$\mathcal{L}(f, X, Y) = \sum_{ij} (\gamma(1 - y_{ij})\log(1 - f_{ij}(X)) + y_{ij}\log(f_{ij}(X)))$$

Литература

-  Daniel Whiteson, Michael Mulhearn, Chase Shimmin, Kyle Cranmer, Kyle Brodie, and Dustin Burns.
Searching for ultra-high energy cosmic rays with smartphones.
Astroparticle Physics, 79:1–9, 2016.
-  Paul Viola and Michael J Jones.
Robust real-time face detection.
International journal of computer vision, 57(2):137–154, 2004.
-  Yann LeCun, LD Jackel, Léon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, UA Muller, E Sackinger, Patrice Simard, et al.
Learning algorithms for classification: A comparison on handwritten digit recognition.
Neural networks: the statistical mechanics perspective, 261:276, 1995.
-  Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu.
Deeply-supervised nets.
In *AISTATS*, volume 2, page 6, 2015.

Мюонный триггер

Распознавание образов

Сверточные нейронные сети:

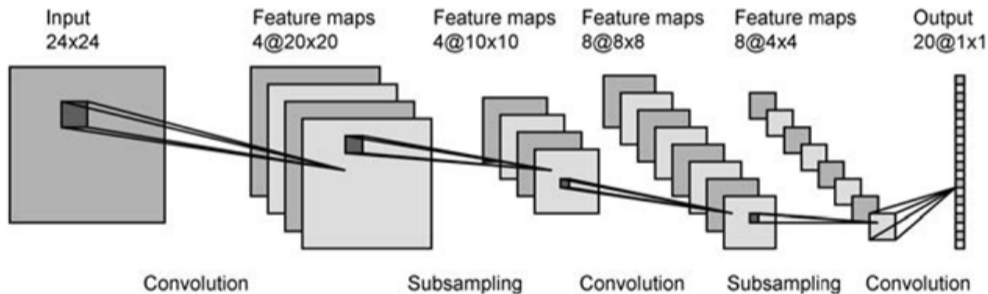
- › Лидируют в различных задачах, связанных с распознаванием образов на изображениях;
- › Вычислительно дорого (обычно, мощные GPUs).

Альтернативные подходы:

- › Алгоритмы, заточенные под конкретную задачу;
- › Viola-Jones каскады [2].

Сверточные нейронные сети

лучший (по качеству) выбор для распознавания образов



Сверточная нейронная сеть это набор слоёв, каждый из которых представлен набором сверточных фильтров. Каждый фильтр сворачивается с выходом предыдущего слоя для получения более высокоуровневого представления изображения .

Каскадная архитектура для сверточных нейронных сетей

Предлагается:

Адаптировать каскадную архитектуру для сверточных нейронных сетей.

Каскадная архитектура:

- › каскад эквивалентен последовательности триггеров ;
- › первые каскады отсекают очевидный шум на ранних этапах;
- › последние каскады улучшают качество поиска треков.

Сверточные нейронные сети:

- › back-propagation;
- › представления передаются от каскада к каскаду.

Схема работы одного каскада

- › Вычисление сверточных операций только в активированных зонах ('ленивое' выполнение)
- › Уменьшение размера изображения (subsampling)
- › Вычисление обновлённой активационной маски

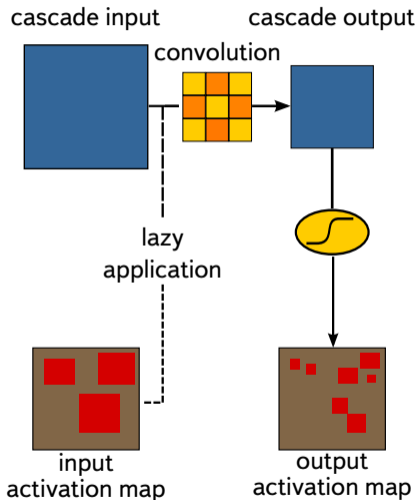
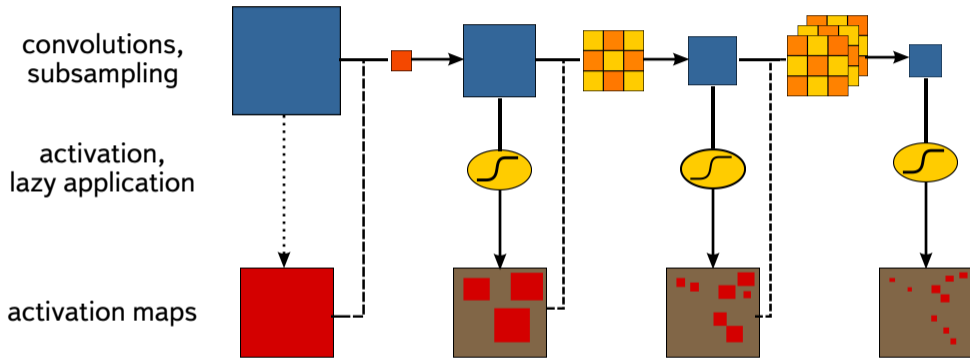


Схема работы каскадной сети



Первая строка представляет сверточную нейронную сеть: последовательность сверточных слоёв, синие квадраты представляют этапы обработки изображения. Красные области в квадратах ниже обозначают активированные области, пунктирные линии это 'ленивое' выполнение.

Формальное ленивое выполнение

$$I^i = h^i(I^{i-1}), \quad (1)$$

$$\hat{A}_{x,y}^i = \sigma^i(I_{x,y}^i), \quad (2)$$

$$A^i = \hat{A}^i \odot A^{i-1}, \quad (3)$$

I^i промежуточное представление входного изображения I^0 после последовательного применения сверточных слоёв

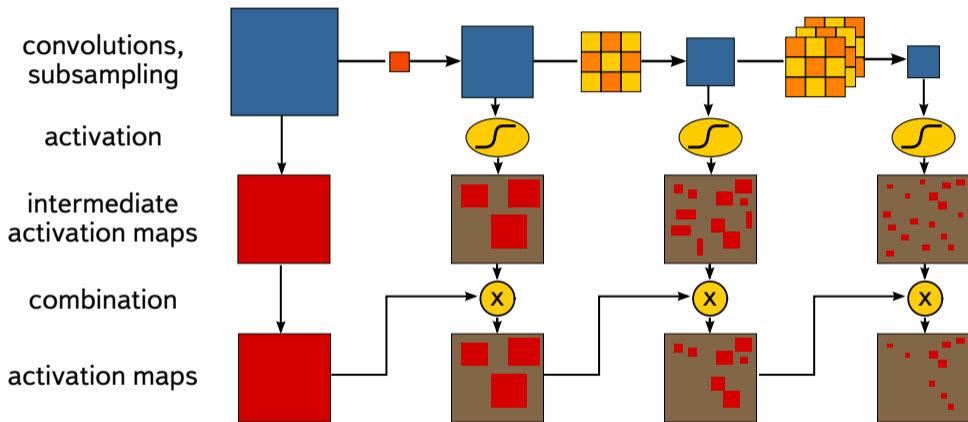
h^i преобразования i -го слоя сети

σ^i нелинейная функция

\hat{A}^i промежуточная маска активаций i -го каскада

A^i дифференцируемая активация каскада i

Тренировка каскадной сети



Каскадная сверточная сеть при тренировке похожа на обычную сверточную сеть. Вместо 'ленивого' вычисления сверточные фильтры применяются ко всему входному изображению, хотя, активационные карты все ещё учитывают информацию с предыдущих каскадов.

Функция потерь

$$\mathcal{L}(H, X, Y) = \sum_i \alpha^i \mathcal{L}_{\text{cascade}}^i(A^i, Y) + \beta \sum_i \mathcal{L}_{\text{complexity}}^i(A^i, Y)$$

› $\mathcal{L}_{\text{cascade}}^i$ — функция потерь i -го каскада, взвешенная кросс-энтропия

$$\mathcal{L}_{\text{cascade}}^i(A^i, Y) = \sum_{pq} (\gamma(1 - y_{pq}) \log(1 - A_{pq}^i) + y_{pq} \log(A_{pq}^i))$$

› $\mathcal{L}_{\text{complexity}}^i$ — оценка вычислительных затрат i -го каскада

$$\mathcal{L}_{\text{complexity}}^i(A^i, Y) = \sum_{pq} F^i(1 - y_{pq}) A_{pq}^i$$

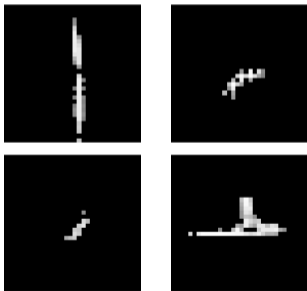
Коэффициенты

- › Наказания за активацию на шуме увеличиваются от каскада к каскаду
- › Наказания за пропуск сигнала выше, чем за активацию на шуме

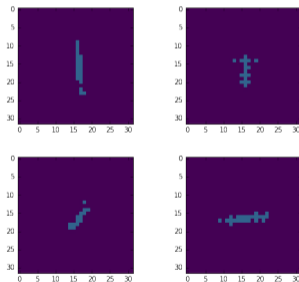
Вычислительный эксперимент

Данные

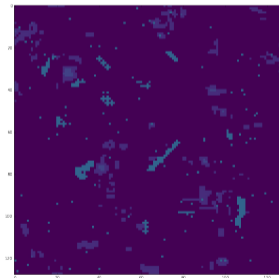
1. Изображения с матрицы облучаемого телефона были взяты за основу
2. Треки были выделены и яркость была снижена до уровня шума
3. Результат был помещён на реальный шум



1. Оригинальные треки



2. Треки с заниженным уровнем



3. Треки помещены на шум

Показательный пример

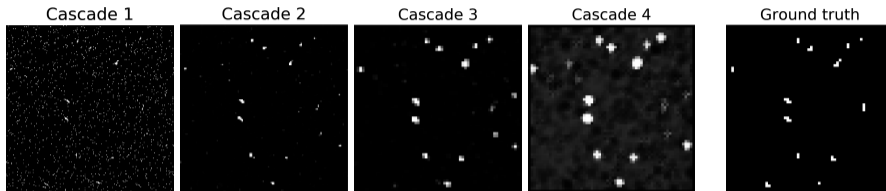


Рис.: Промежуточные активационные маски и маска правильного ответа

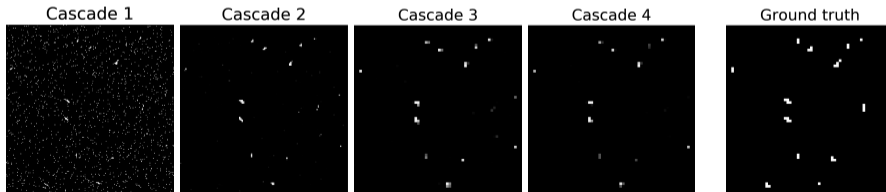


Рис.: Активационные маски и маска правильного ответа

Результаты

- › Baseline: порог яркости
- › 4 каскада:

1 filter $1 \times 1 \rightarrow$ 1 filter $3 \times 3 \rightarrow$ 3 filters $3 \times 3 \rightarrow$ 6 filters 3×3

Суммарная сложность $\approx 0.04 \times$ сложности традиционной сверточной сети

$$\text{Recall} = \frac{TP}{TP+TN}, \text{Precision} = \frac{TP}{TP+FP}$$

Model \ Recall	Baseline(1)	CNNTrig(1.4)	CNNTrig(2.0)
0.90	-	0.60	0.65
0.95	-	0.39	0.44
0.99	-	0.12	0.15
1.00	0.01	-	-

Таблица: Сравнение precision моделей

Заключение

Заключение

CRAYFIS experiment:

- › нужен для изучения УВКИ;
- › эффективный триггер на каждом мини-детекторе играет ключевую роль.

Каскадная сверточная нейронная сеть:

- › каскадная архитектура позволяет отбрасывать шум на ранних стадиях;
- › существенное уменьшение вычислительных затрат, по сравнению с традиционными сверточными нейронными сетями.

Результаты эксперимента:

- › суммарное количество вычислений: $1.4 \times$ baseline, 0.04 of CNN;
- › signal efficiency (recall): 0.9
- › background rejection (precision): 0.6.

Мой вклад

- › Предложил каскадную сверточную нейронную сеть для детектирования мюонов
- › Экспериментально доказал применимость предложенного метода для поставленной задачи
- › Сравнил качество/вычислительные затраты для своего метода и традиционных сверточных нейронных сетей
- › Экспериментально показал применимость своего метода для традиционных задач, связанных с распознаванием образов

CRAYFIS collaboration

University of California Irvine:

- › Daniel Whiteson;
- › Chase Shimmin;
- › Homer Strong;
- › Jay Karimi;
- › Kyle Brodie;
- › Rob Porter;
- › Eric Albin;
- › Emma He;
- › Zichao Ziliver Yuan;
- › Jeff Swaney;

University of California Davis:

- › Michael Mulhearn;
- › Dustin Burns;
- › Brandon Buonacarsi;

New-York University:

- › Kyle Cranmer;

Yandex School of Data

Analysis:

- › Andrey Ustyuzhanin;
- › Maxim Borisyak;
- › Mikhail Usvyatsov;

National Astronomical Observatory of China:

- › Jianrong Deng;

Unaffiliated:

- › Danielle Cranmer;
- › Jodi Goddard.

Backup

Cascade Network Loss Function

$$\mathcal{L} = \sum_i \alpha^i \mathcal{L}_{\text{cascade}}^i + \beta \sum_i \mathcal{L}_{\text{complexity}}^i$$

$$\mathcal{L}_{\text{cascade}}^i = \sum_j \gamma^i (1 - y_j) \log(1 - p_j^i) + y_j \log p_j^i$$

$$\mathcal{L}_{\text{complexity}}^i = \sum_j F^i (1 - y_j) p_j^i$$

- › γ^i — penalty for passing noise;
- › p_j^i — i -th cascade activation on j -th window;
- › y_j — presence of particle trace in j -th window (target);
- › F^i — total size of filters in i -th cascade (computational cost).

Experiment

