Анализ клеток

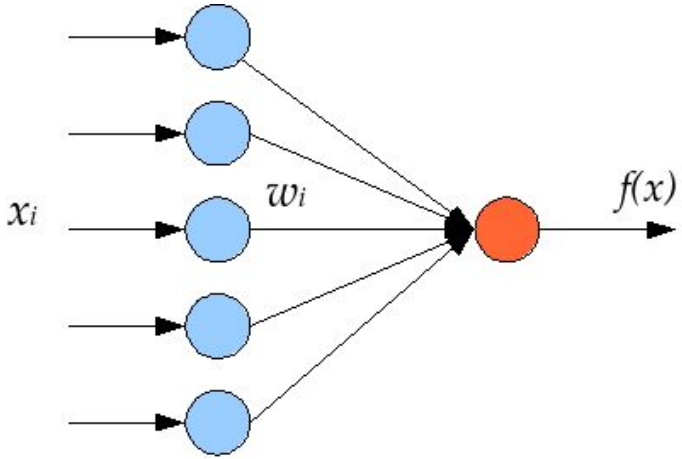
Автор: Валерий Черепанов
Научный руководитель: Алексей Шпильман

Классификация изображений

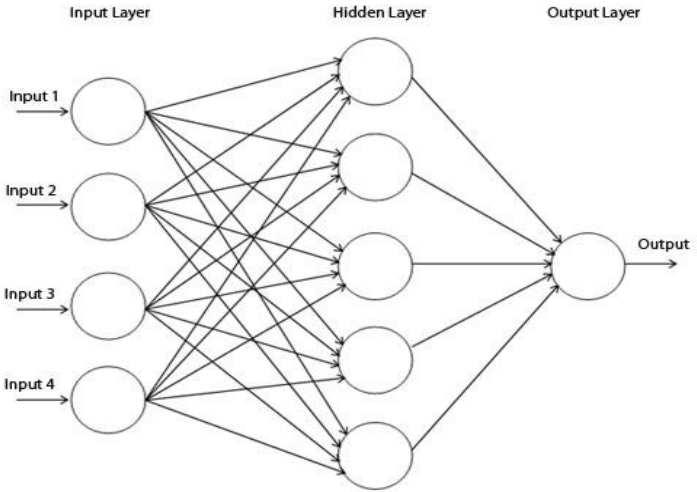
- Эвристики (hand-crafted features)
  - Часто плохо масштабируются на другие задачи
  - Существуют в огромном разнообразии (HOG, SIFT, тысячи их)
- Нейронные сети
  - Появились еще в шестидесятых годах, имели большой потенциал
  - К началу девяностых почти все от них отказались
  - Теперь это снова the next big thing

Teach Yourself Machine Learning in 24 seconds

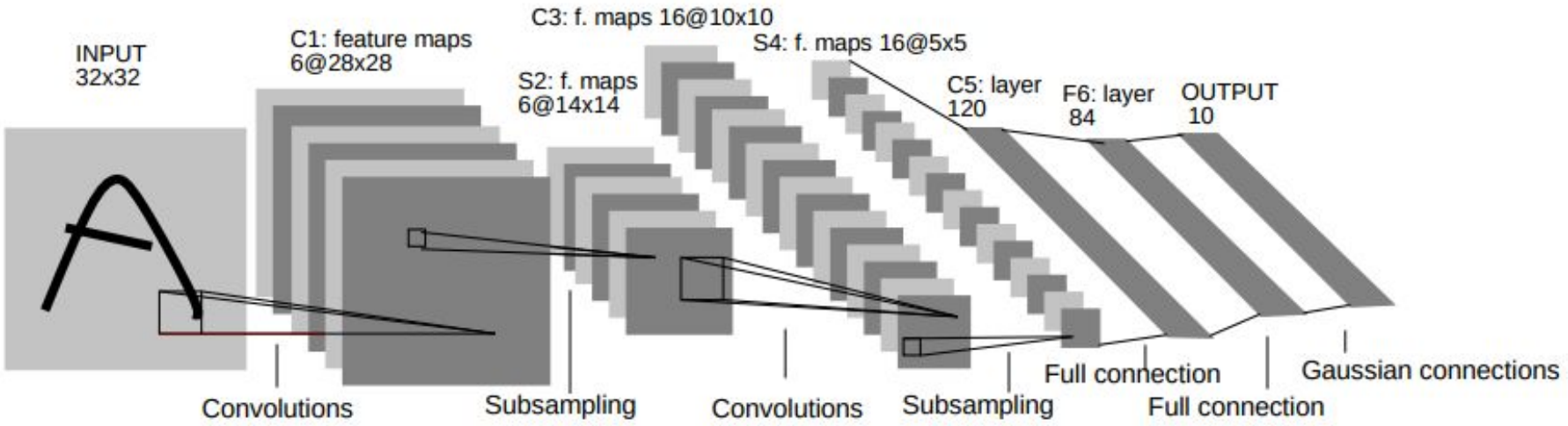
Перцептрон



Многослойный перцептрон



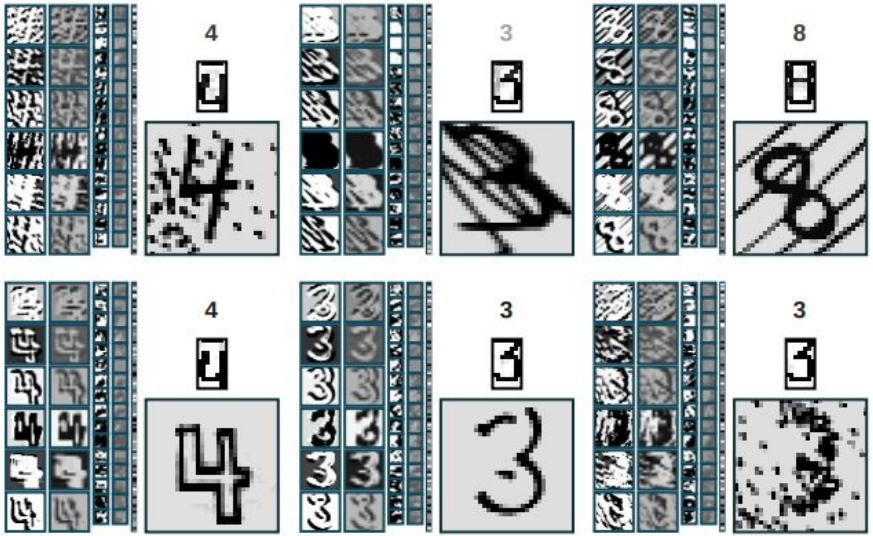
Сверточная сеть

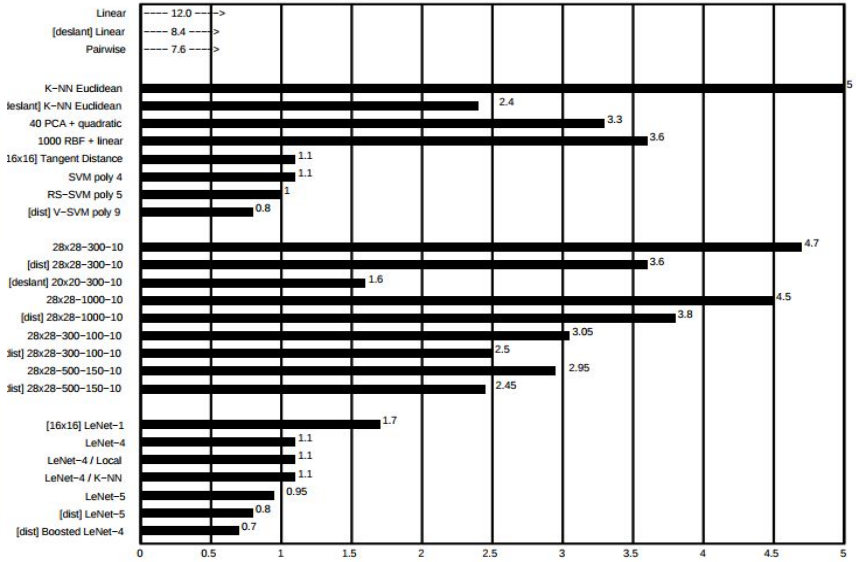


В современных реализациях часто бывают и другие слои, например, dropout

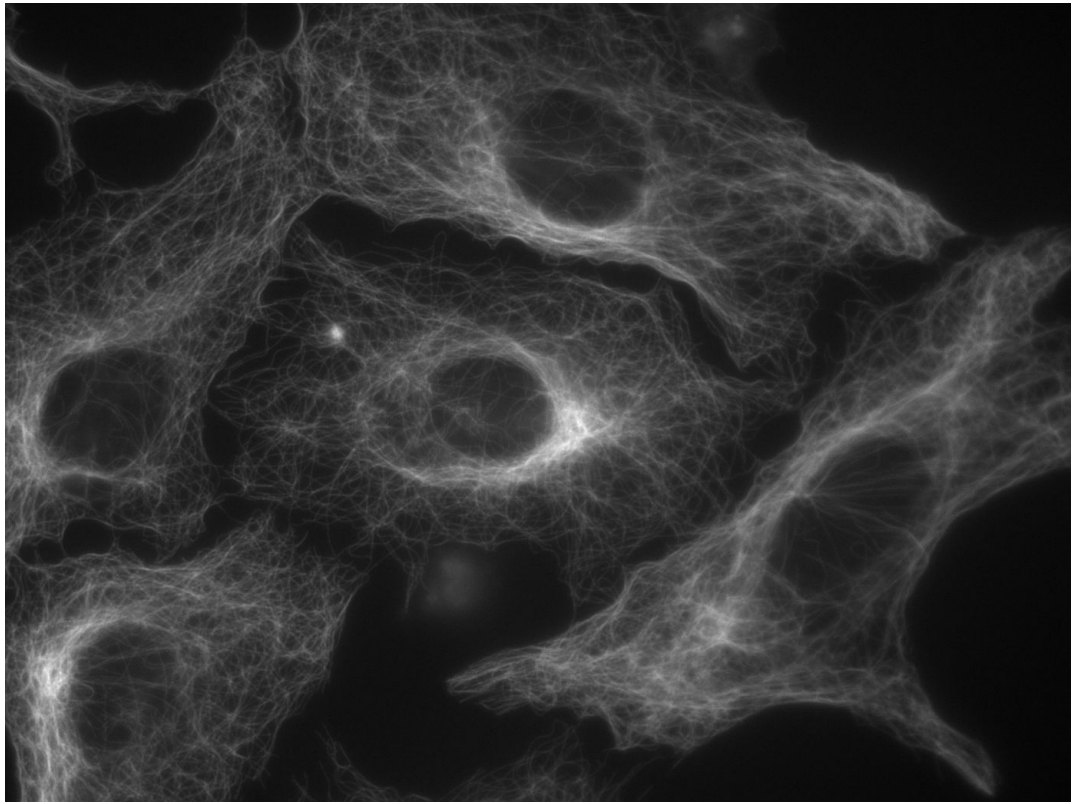
Сверточная сеть

- MNIST — одна из первых успешно решенных задач (LeCun et al., 1998)



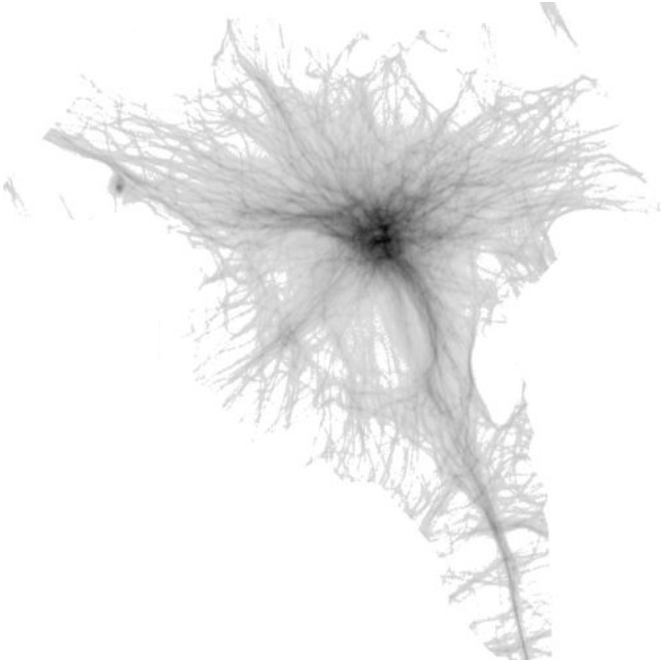


Исходные данные

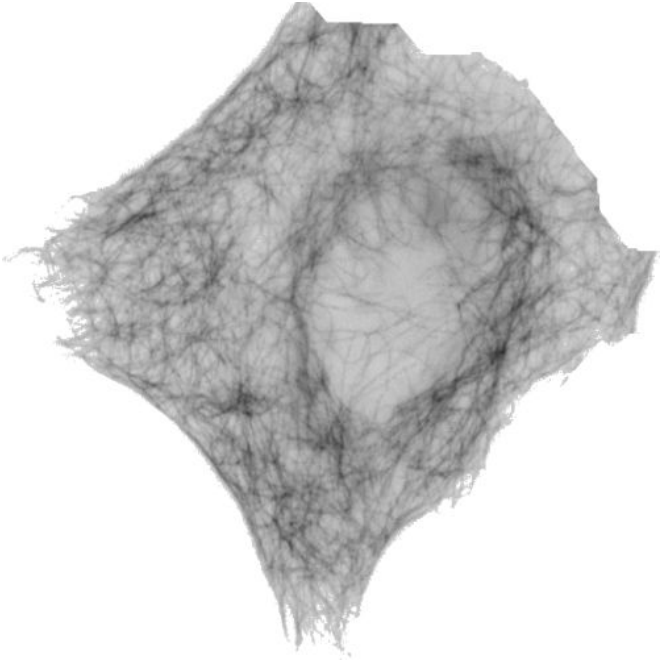


Постановка задачи:

Нужно научится разделять клетки на 4 класса в зависимости от концентрации препарата



Обычная клетка



Клетка под воздействием Таксола

Технологии

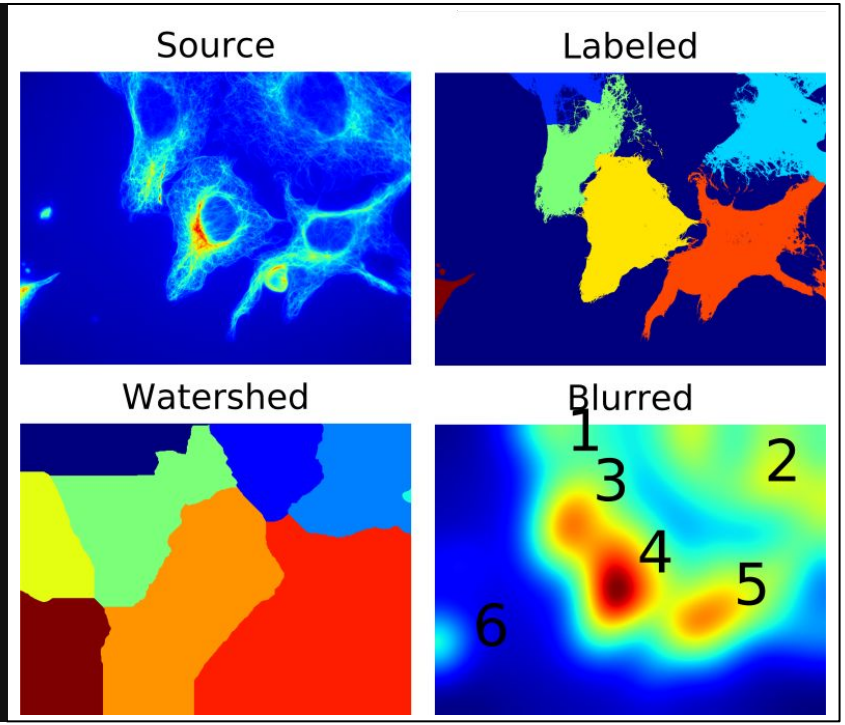
- Python
- Numpy
- Mahotas
- Keras (with Theano backend)
- Matplotlib

Этапы

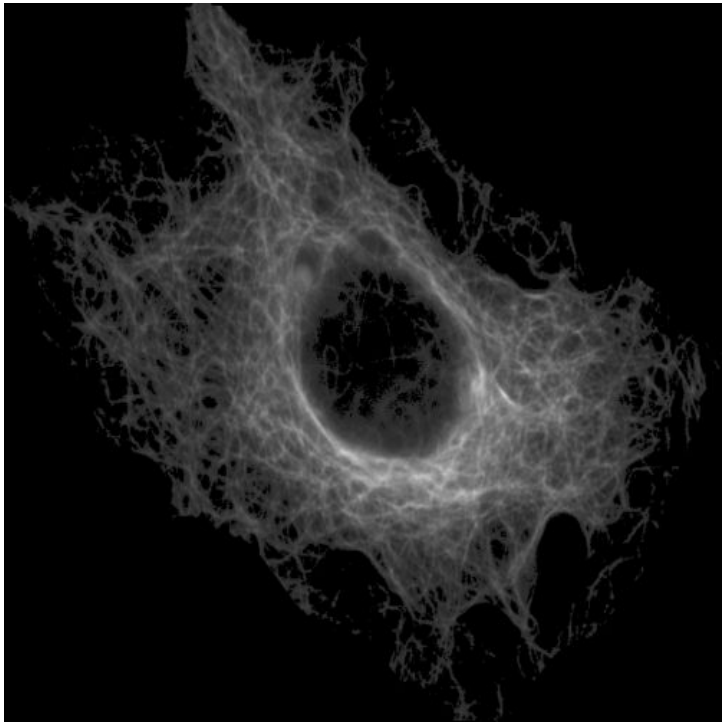
1. Обработка изображений
   - Можно немного улучшить результат, удалив руками неудачные варианты (их <5%)
2. Расширение выборки (data augmentation)
   - Можно генерировать в реальном времени на CPU, пока сеть обучается на GPU
3. Обучение

Обработка изображений

```
blurred = gaussian_filter(image)
bin_image = img > blurred.mean()

maxima = regmax(stretch(blurred))

dist = distance(bin_image)

watershed = cwatershed(dist, maxima)

remove_small_regions(img, watershed*bin_image)
remove_bordering(img)
```



Результат



Были опробованы и другие варианты, в частности:

- Обрабатывать всю картинку сразу
- Нарезать на много случайных кусков
- Всякие преобразования, вроде эквализации гистограммы

Все они по тем или иным причинам не подошли.

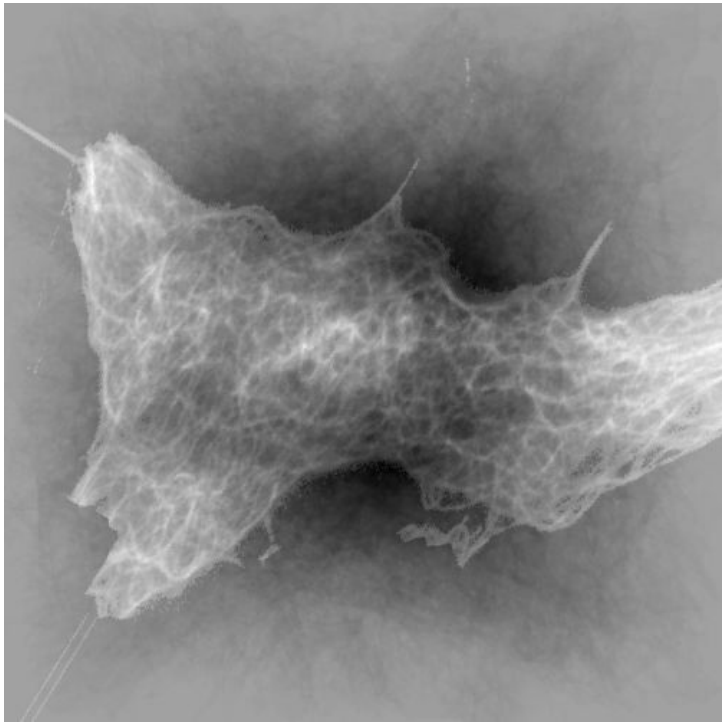
Data augmentation

- Поворот на случайный градус
- Выравнивание яркости
- Отражение относительно осей

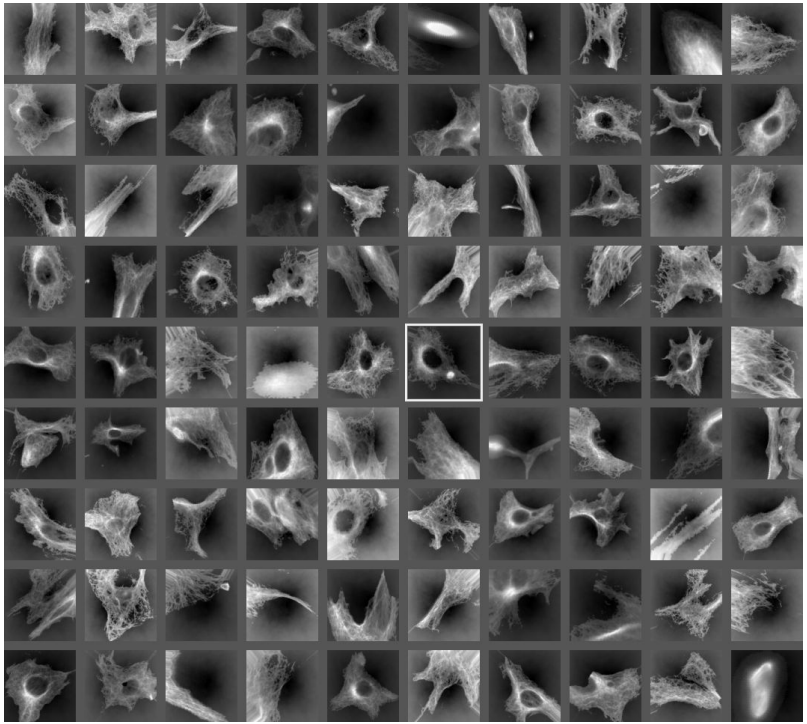
Опционально:

- Разные сдвиги
- Масштабирование
- Нормализация стандартного отклонения (судя по всему, неправильно работает в Keras :( )

Результат

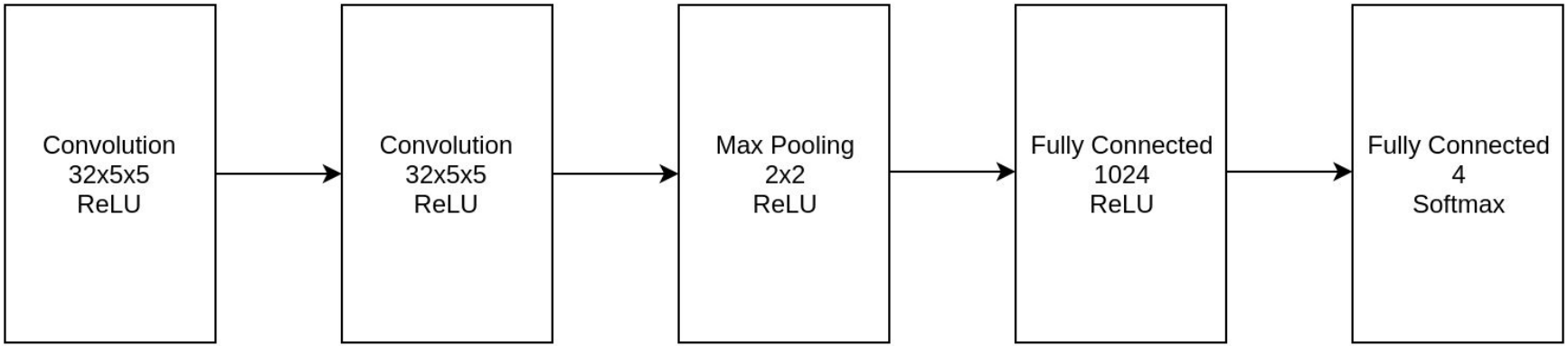


С этим уже можно работать



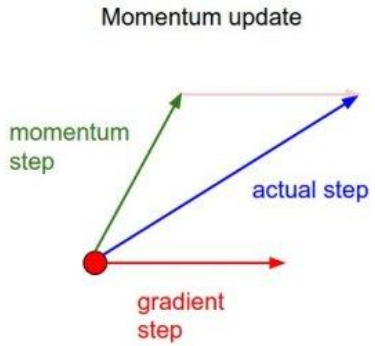
Архитектура

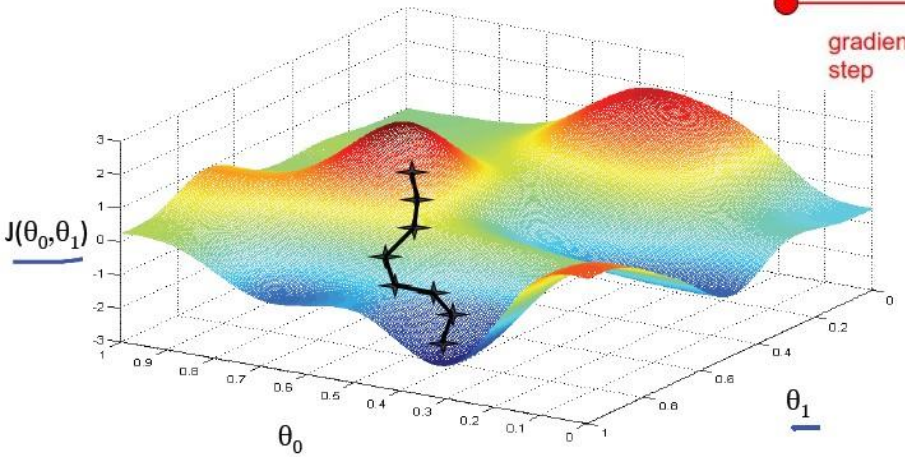
- Оптимизатор: Стохастический градиентный спуск (+момент Нестерова)
- Функция активации: Rectified Linear Unit (ReLU)
- Функция потерь: Перекрестная энтропия
- Размер входа: 78x78



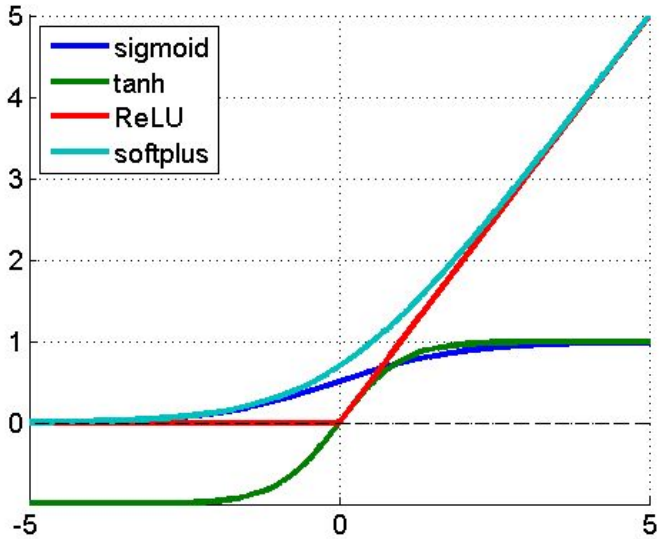
Стохастический градиентный спуск

Достаточно быстро,
достаточно надежно





Rectified Linear Unit

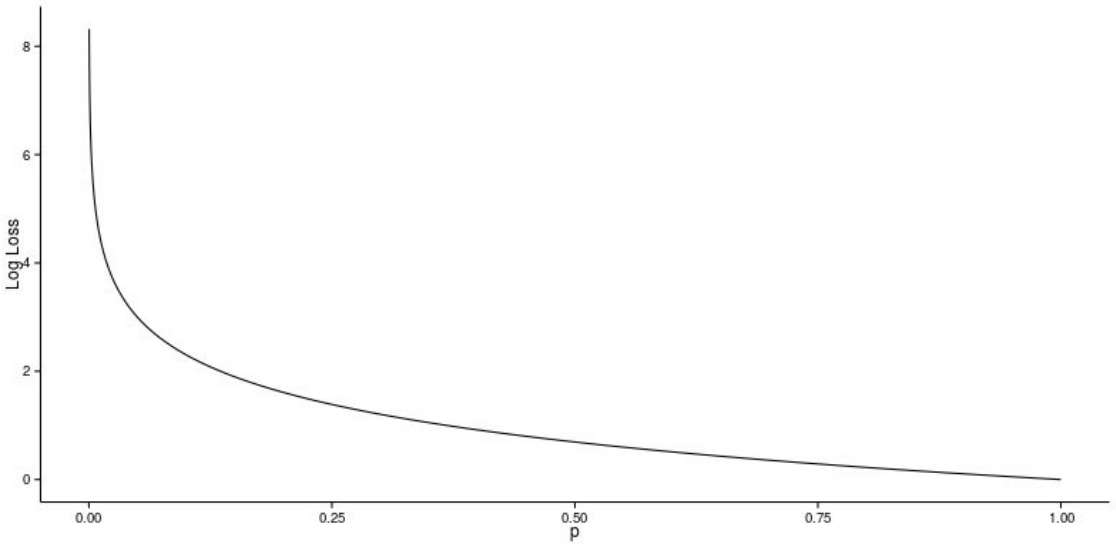


- ReLU: $\max(0, x)$
- Sigmoid: $\frac{1}{1+e^{-x}}$
- Softplus: $\ln(1+e^{x})$

Перекрестная энтропия (log loss)

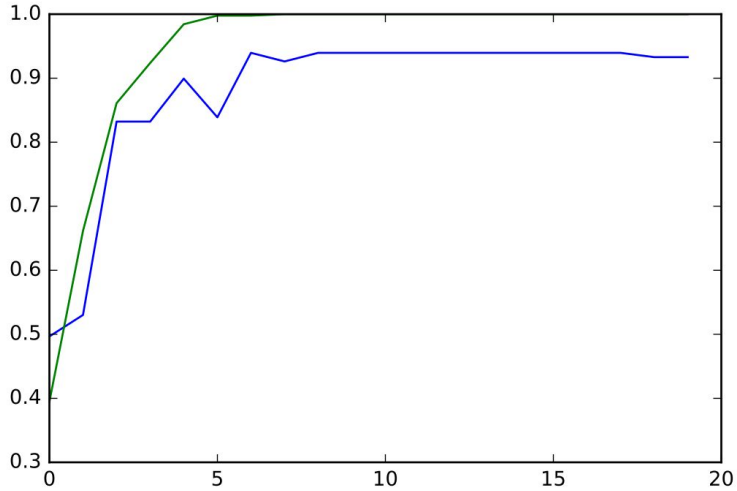
$$-\frac{1}{N}\sum_{i=1}^{N}\left(y_i \log p_i + (1-y_i)\log(1-p_i)\right)$$

Считаем сумму по правильному классу и всем остальным по отдельности



Статистика

- Одна эпоха (~500 клеток) работает за 2 секунды на GeForce GTX 860m
- До достижения 100% на тренировочной выборке достаточно 15-20 эпох
- Точность классификации около 90%

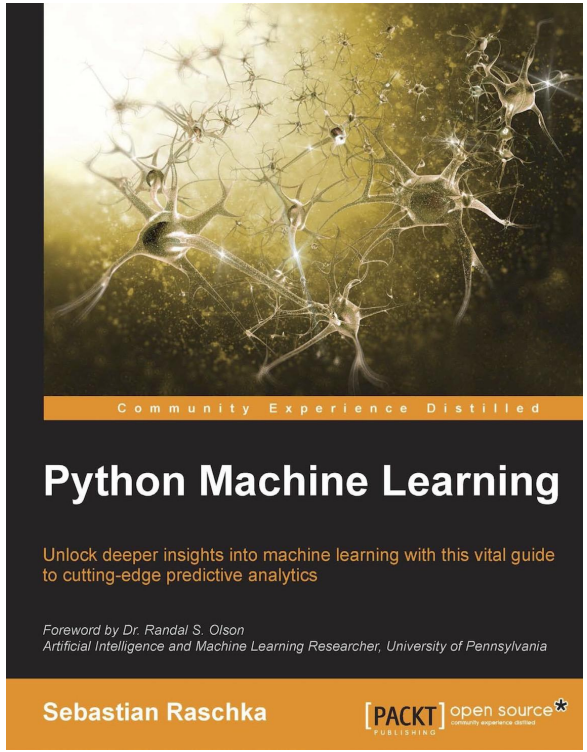


Загадки

- Более сложные и глубокие сети (в том числе и заточенные на схожие задачи) дают худший результат (вероятно, я не умею их готовить)
- Оптимизаторы время от времени сразу падают в очень плохой локальный минимум
- Регуляризация с помощью dropout слоев никак не помогает бороться с переобучением

Что я узнал

- Многое про машинное обучение вообще и про нейронные сети в частности
- Различные библиотеки под Python, научился с ними работать
- Кое-что об обработке изображений
- Как проводить вычисления на GPU (CUDA)

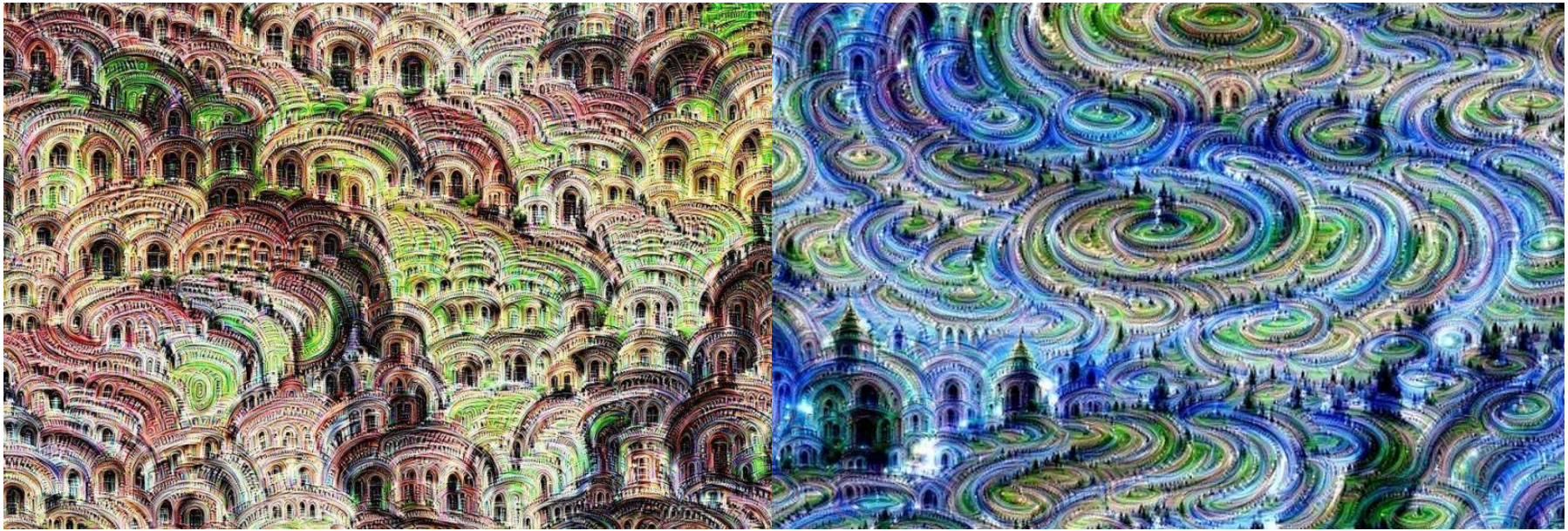


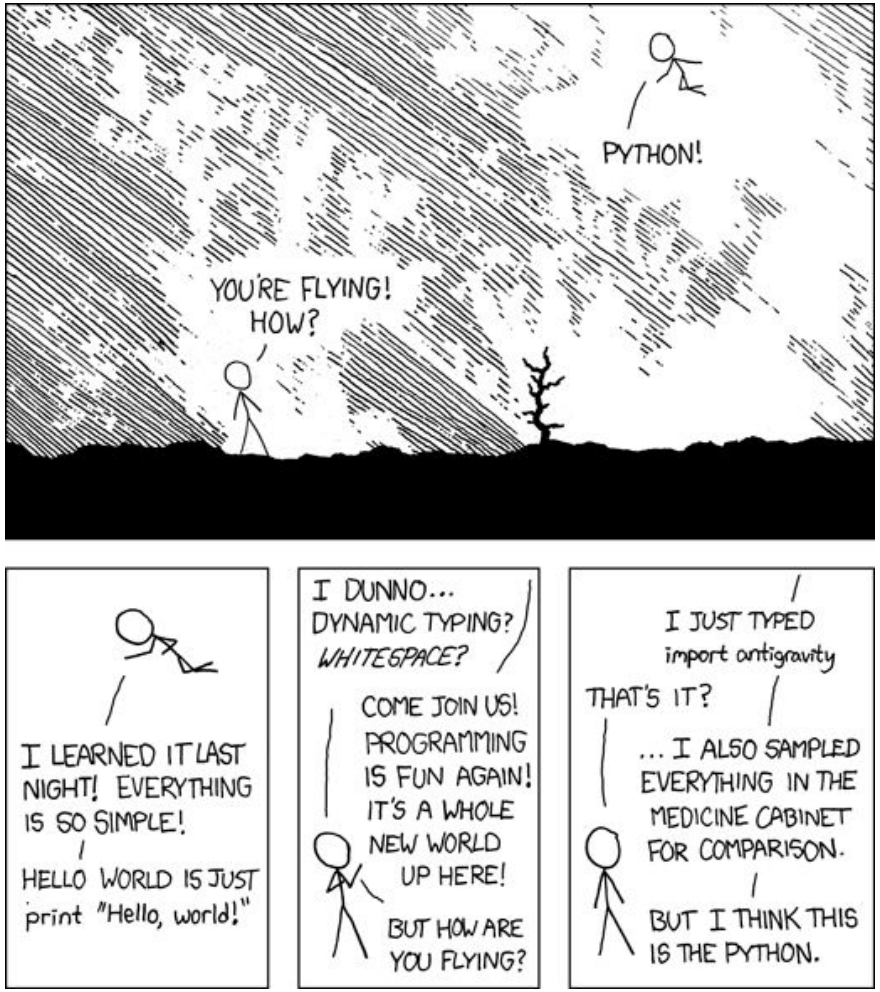
Что еще можно сделать

- Использовать информацию о том, на какой картинке находится клетка
- Качественнее вырезать клетки. Вероятно, стоит отказаться от блюра и заменить его более умным последовательным фильтром
- Hyperparameter optimization (Grid search with cross-validation)

Спасибо за внимание

https://github.com/Qumeric/paclitaxel-classify




PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.