

СТАРТАПУ ИИ:

Прогнозирование временных рядов на нейронных сетях. Потребление электричества (TensorFlow)

Обсуждаем: <https://t.me/devdvAI>, <https://t.me/devdvStartup>

Репозиторий: <https://github.com/akumidv/startup-khv-ai-study>

ПРОЦЕДУРА СОЗДАНИЯ МОДЕЛИ

ПОДГОТОВКА ДАННЫХ

СОЗДАНИЕ СТРУКТУРЫ МОДЕЛИ

ПОДБОР ПАРАМЕТРОВ ОБУЧЕНИЯ

ДАННЫЕ

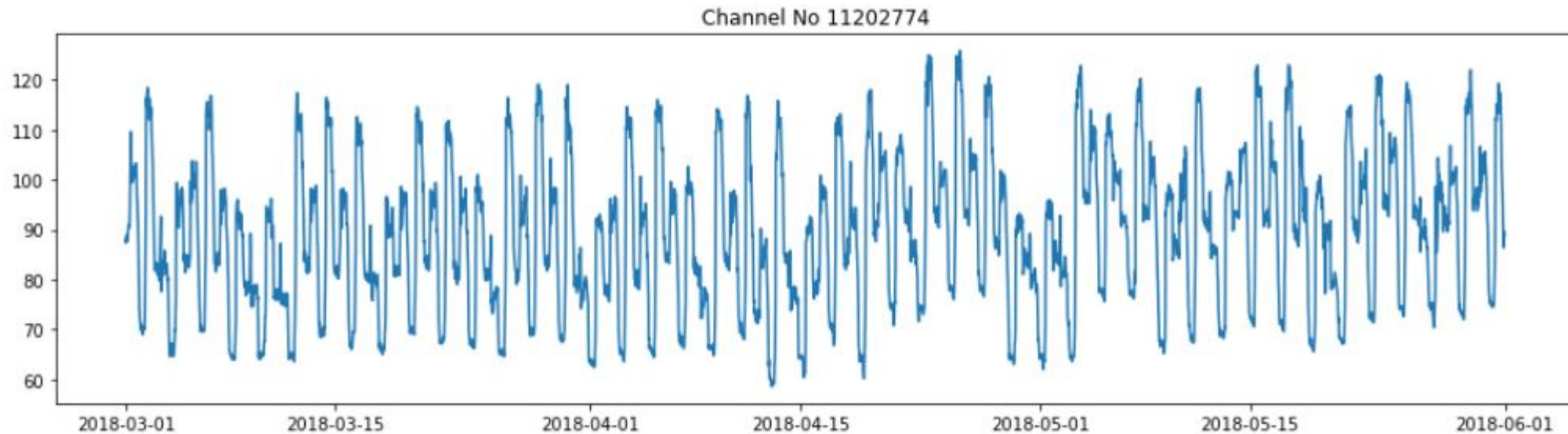
TIME SERIES – ВРЕМЕННЫЕ РЯДЫ

- Значения описывающие процесс
- Измерения в последовательные моменты времени
- Как правило равные интервалы

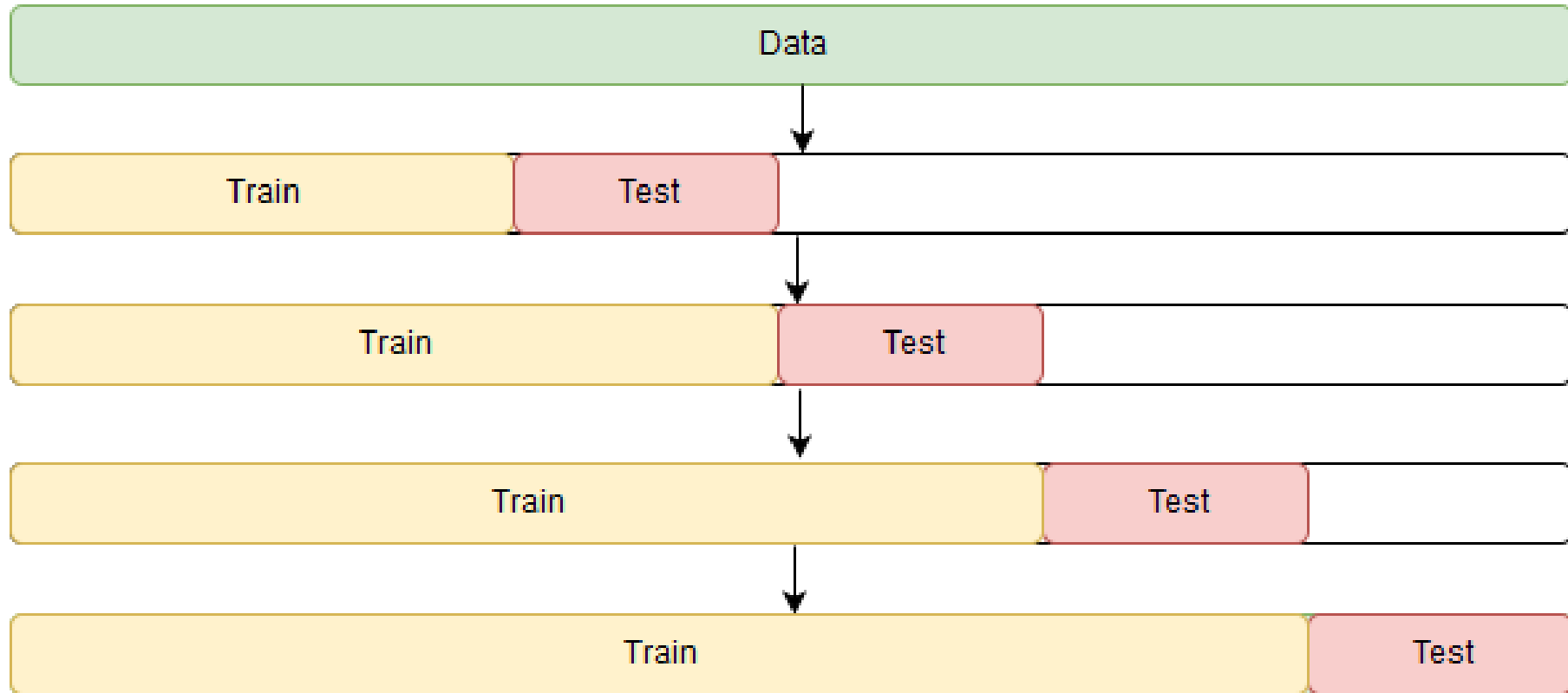
ПРИМЕРЫ?

СТРУКТУРА ДАННЫХ ВРЕМЕННОГО РЯДА

	pwr
2018-03-01 00:00:00	87.8
2018-03-01 00:30:00	87.4
2018-03-01 01:00:00	88.4
2018-03-01 01:30:00	88.8
2018-03-01 02:00:00	88.0



КАК ОБУЧАТЬ И ПРОВЕРЯТЬ ВРЕМЕННОЙ РЯД?



МОДЕЛЬ ПРОГНОЗИРОВАНИЯ

ЗАДАЧА ПРОГНОЗИРОВАНИЯ

ЗАВТРА КАК ВЧЕРА

СКОЛЬЗЯЩИЕ СРЕДНИИ: КОЛ-ВО ИНТЕРВАЛОВ, ЭКСПОТЕНЦИАЛЬНЫЕ...

...

НЕЙРОННЫЕ СЕТИ

ПРОГНОЗИРОВАНИЕ ПО СКОЛЬЗЯЩЕЙ СРЕДНЕЙ



ОСНОВНЫЕ ЗАДАЧИ ДЛЯ НЕЙРОННЫХ СЕТЕЙ

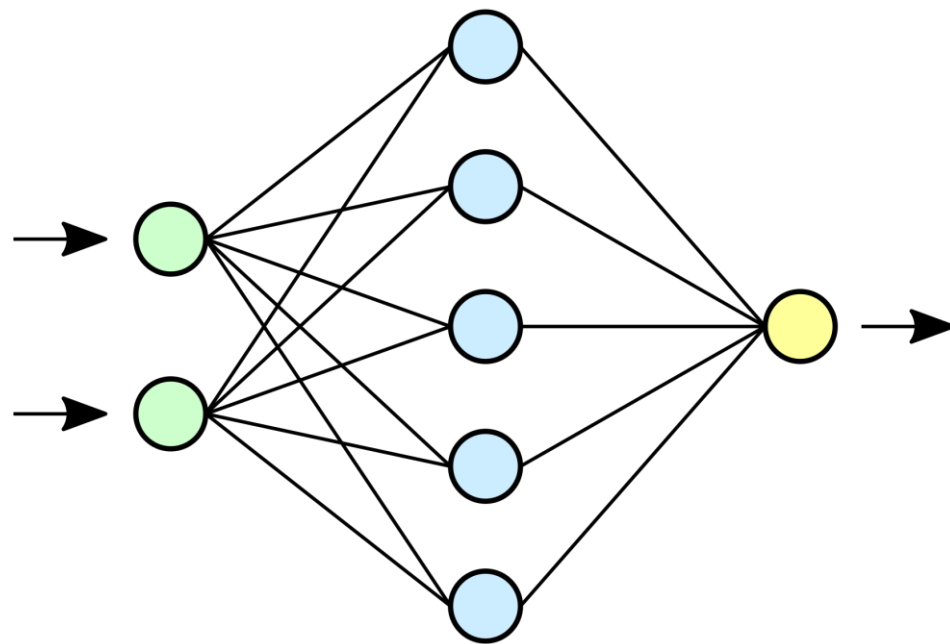
КЛАССИФИКАЦИЯ

РАСПОЗНАВАНИЕ

ПРЕДСКАЗАНИЕ

ЧТО ТАКОЕ НЕЙРОННАЯ СЕТЬ?

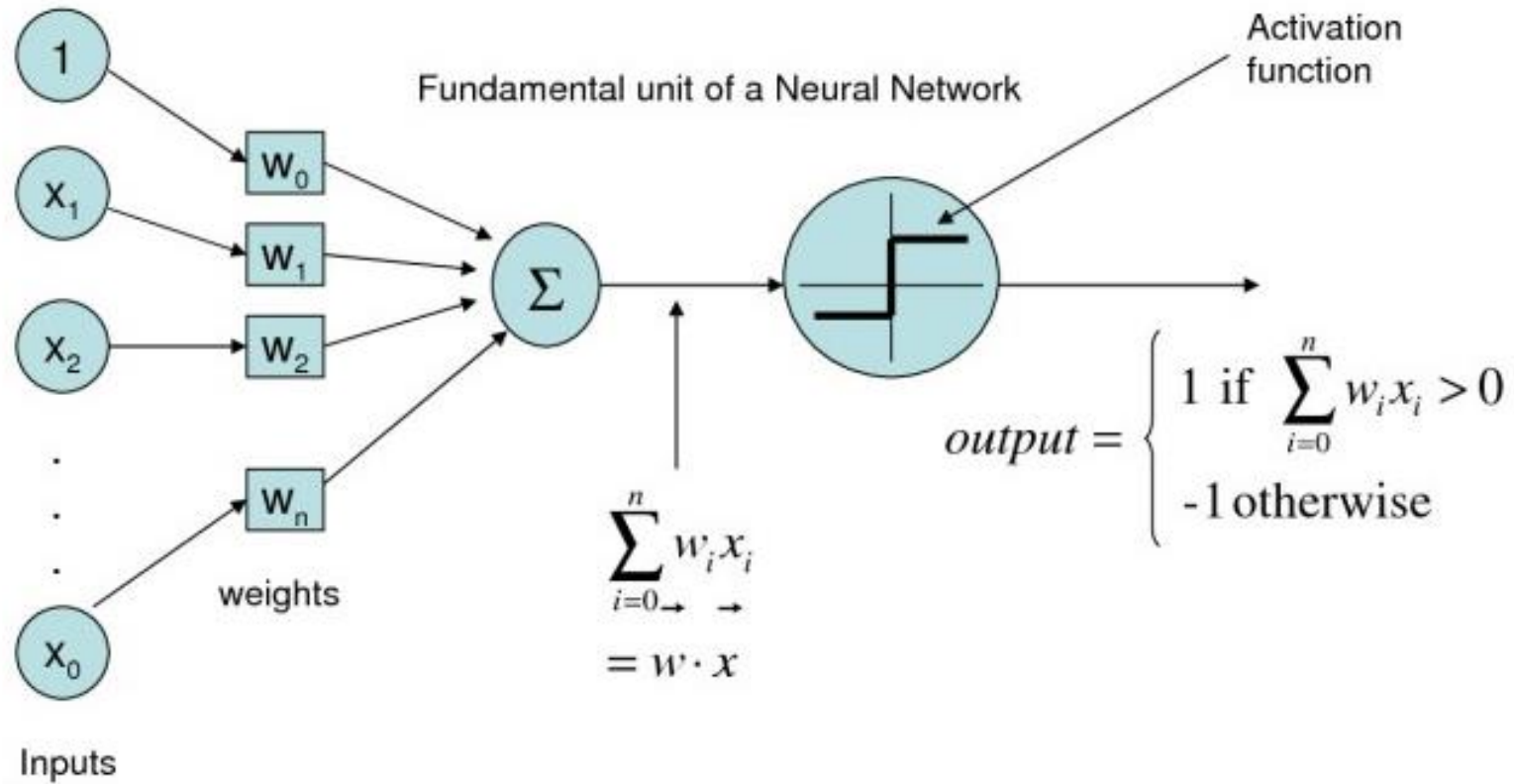
МАТЕМАТИЧЕСКАЯ МОДЕЛЬ



НЕЙРОННАЯ СЕТЬ НЕ ПРОГРАММИРУЕТСЯ,
А ОБУЧАЕТСЯ

ОСНОВА НЕЙРОН И СВЯЗИ МЕЖДУ НИМИ

The Perceptron



TENSORFLOW И KERAS



TensorFlow

МНОГОСЛОЙНЫЕ СЕТИ



Keras

УПРОЩЕНИЕ СОЗДАНИЯ И ОБУЧЕНИЯ
СЕТЕЙ (ВСТРОЕН В TENSORFLOW)

БАЗОВЕ СЛОИ KERAS

DENSE Полносвязанный слой `output=activation(dot(input,kernel)+bias)`

ACTIVATION Функция активации применяемая к выводам предыдущего слоя

DROPOUT Случайное обнуление выводов

FLATTEN Конвертации входящих данных в меньшую размерность

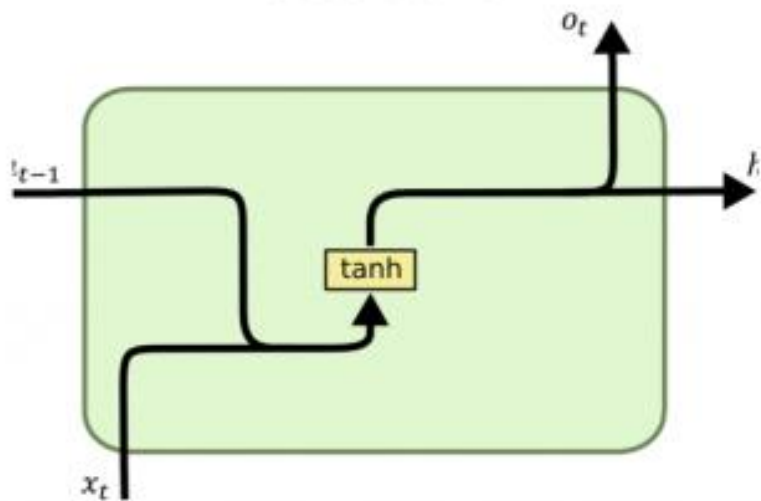
INPUT Входной слой

RESHAPE Изменение под размерность

РЕКУРРЕНТНЫЕ НЕЙРОННЫЕ СЕТИ

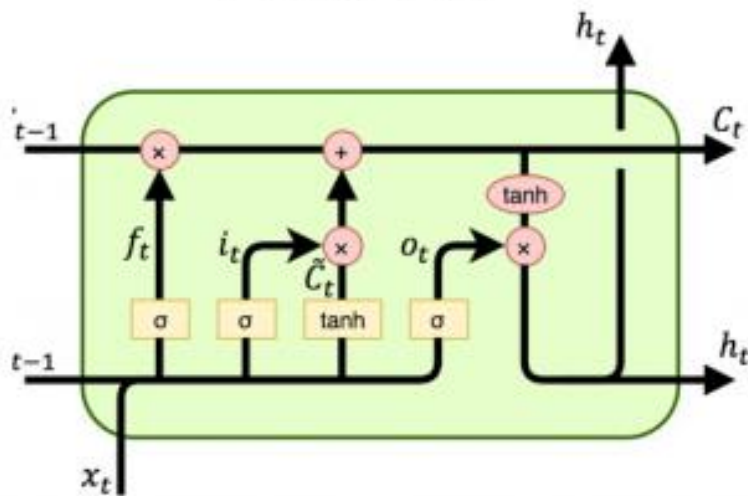
ДЛЯ ВРЕМЕННЫХ РЯДОВ И NLP

RNN



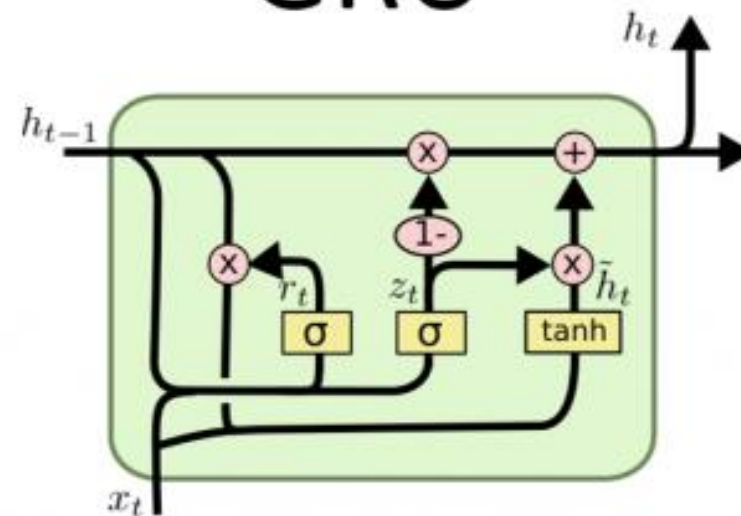
Вывод подается обратно в качестве входящих данных

LSTM



RNN с хранилищем для информации о предыдущих обучениях

GRU



Упрощение LSTM с забыванием

ОБУЧЕНИЕ

ОСНОВНЫЕ ПОНЯТИЯ

ЭПОХИ (EPOCHS)

БАТЧ (BATCH SIZE)

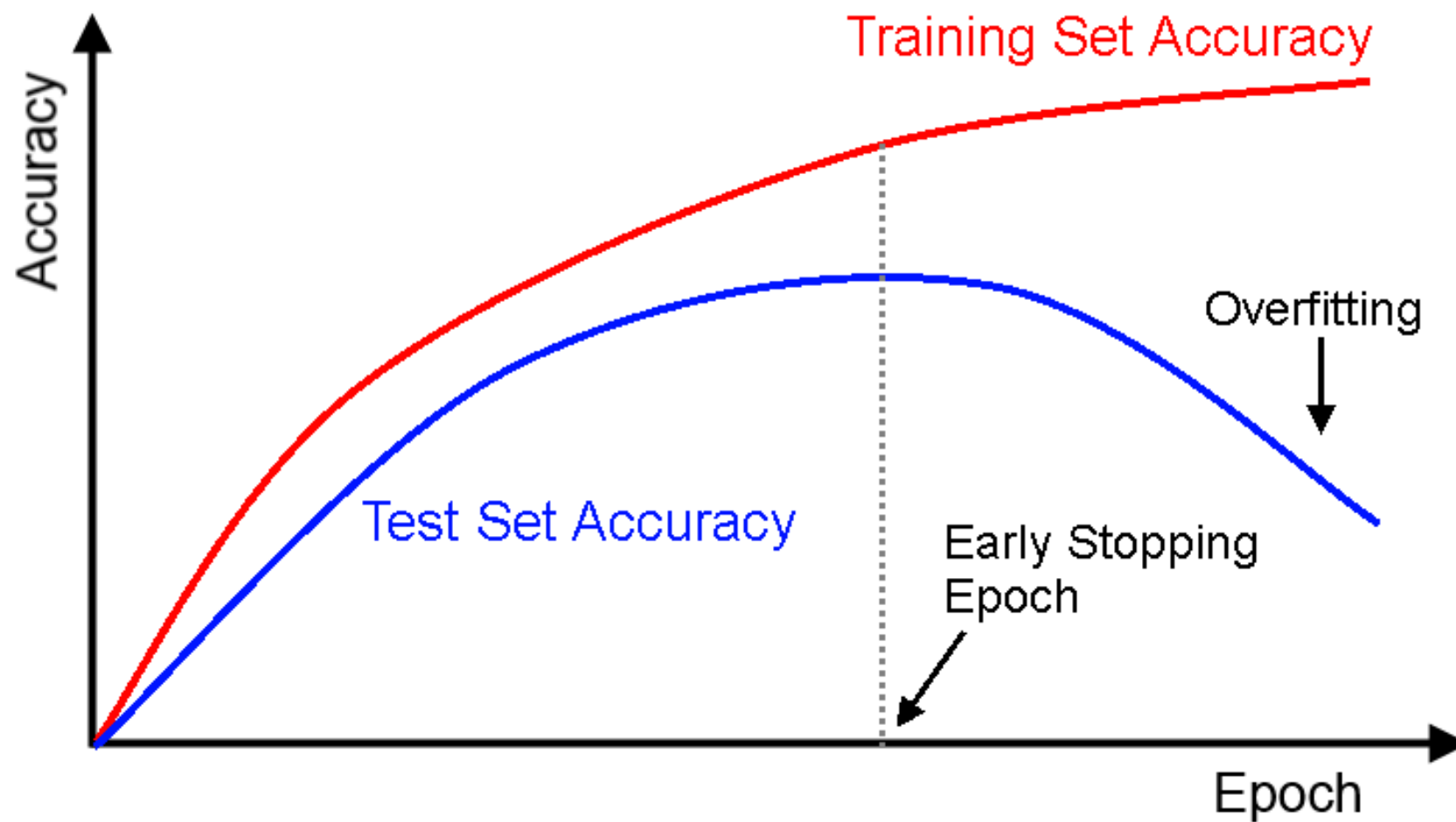
ПОТЕРИ ОБУЧЕНИЯ (TRAINING LOSS)

ВАЛИДАЦИОННАЯ/ТЕСТОВАЯ ПОТЕРИ (VALIDATION/TEST LOSS)

МЕТОД ПРОРЕЖИВАНИЯ/ИСКЛЮЧЕНИЯ (DROPOUT)

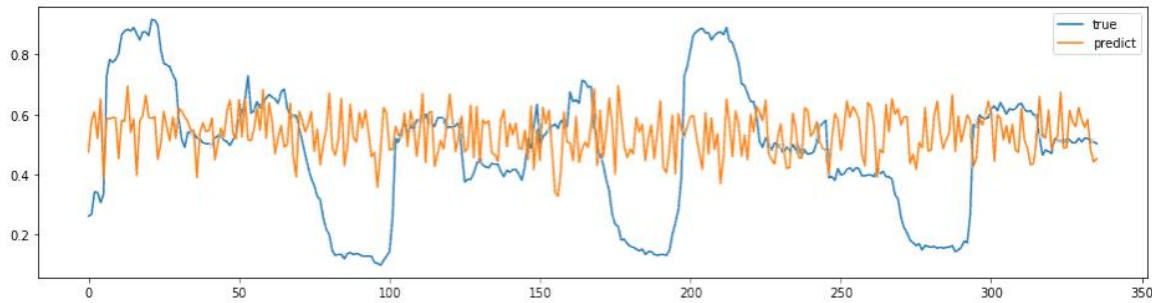
КОЭФФИЦИЕНТ ОБУЧЕНИЯ (LEARNING RATE)

ЭПОХА - EPOCH

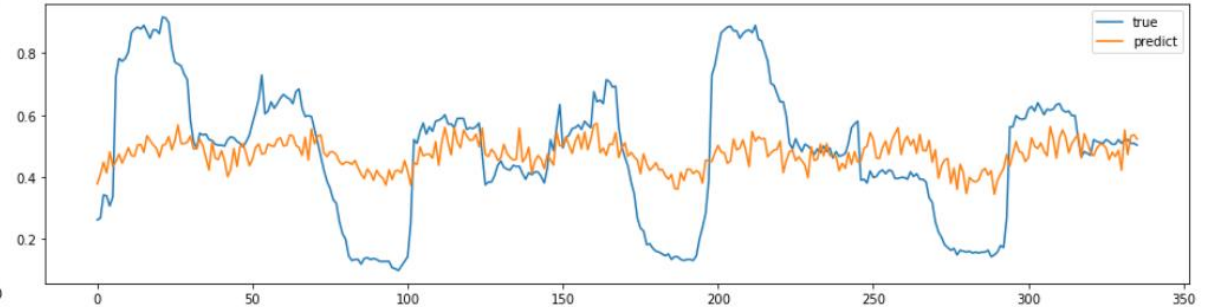


КАЧЕСТВО ПРОГНОЗИРОВАНИЯ ПО ЭПОХАМ

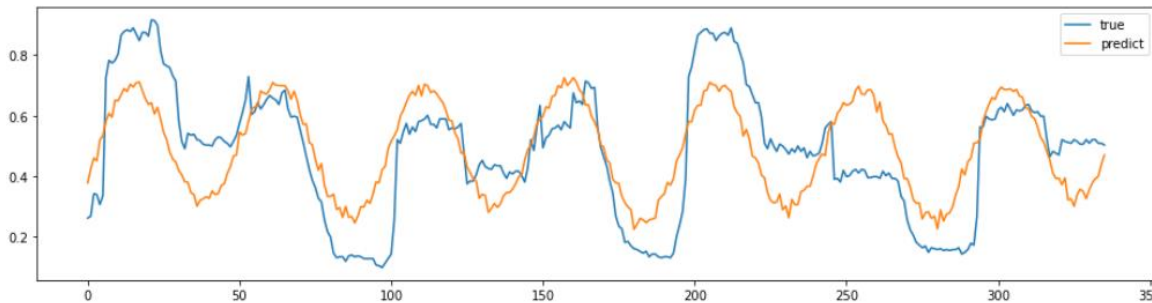
1 ЭПОХА, LOSS: 0.1518



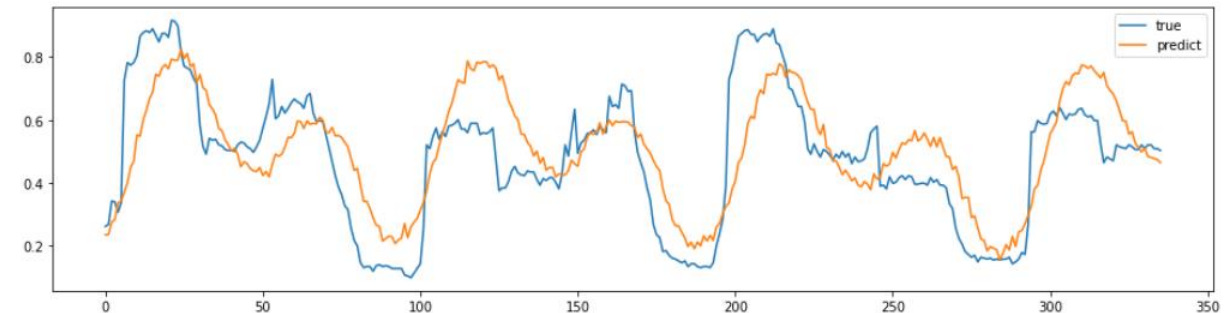
5 ЭПОХ, LOSS: 0.0517 VLOS: 0.0473



10 ЭПОХ, LOSS: 0.0310

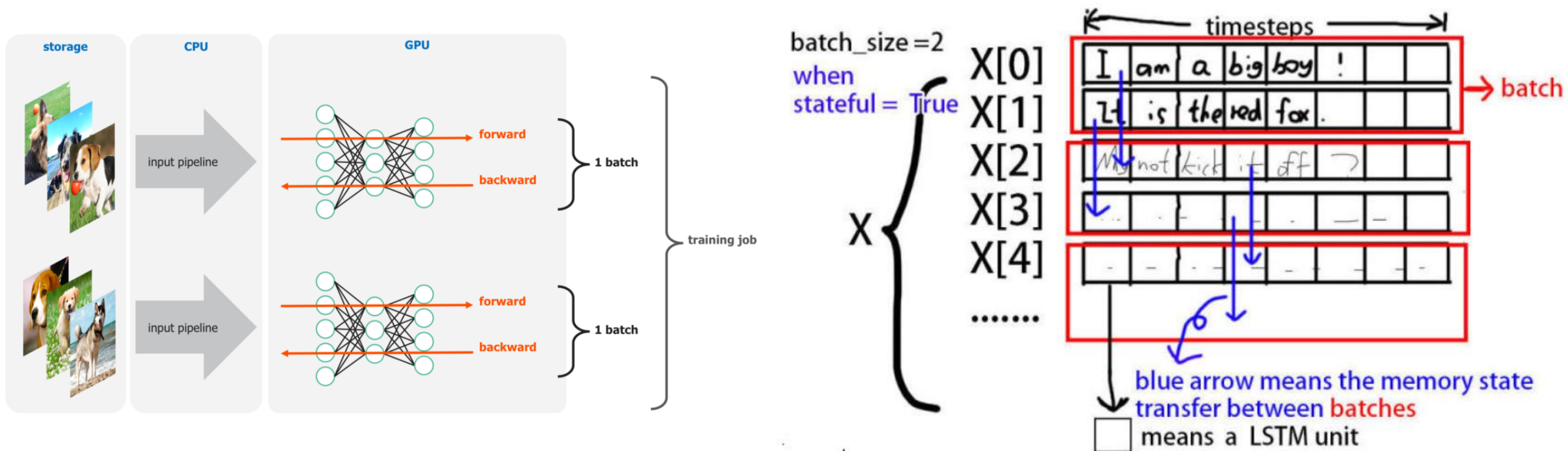


20 ЭПОХ, LOSS: 0.0207



BATCH SIZE

СКОРОСТЬ ОБУЧЕНИЯ И ТОЧНОСТЬ



BATCH SIZE НА 10-ТИ ЭПОХАХ

8

```
Epoch 00009: val_loss did not improve from 0.01524
Epoch 10/10
320/320 [=====] - 9s 29ms/step - loss: 0.0193 - val_loss: 0.0161

Epoch 00010: val_loss did not improve from 0.01524
CPU times: user 1min 35s, sys: 2.41 s, total: 1min 38s
Wall time: 1min 34s
```

32

```
Epoch 00009: val_loss improved from 0.03010 to 0.02993, saving model to /content/drive/M
Epoch 10/10
80/80 [=====] - 3s 39ms/step - loss: 0.0278 - val_loss: 0.0254

Epoch 00010: val_loss improved from 0.02993 to 0.02539, saving model to /content/drive/M
CPU times: user 30.4 s, sys: 2.18 s, total: 32.6 s
Wall time: 32.6 s
```

64

```
Epoch 00009: val_loss improved from 0.02909 to 0.02823, saving model to /content/drive/I
Epoch 10/10
40/40 [=====] - 2s 58ms/step - loss: 0.0320 - val_loss: 0.0275

Epoch 00010: val_loss improved from 0.02823 to 0.02750, saving model to /content/drive/I
CPU times: user 19.7 s, sys: 2.83 s, total: 22.1 s
Wall time: 24.4 s
```

256

```
Epoch 00009: val_loss improved from 0.04660 to 0.04547, saving model to /content/drive/Mj
Epoch 10/10
10/10 [=====] - 2s 163ms/step - loss: 0.0454 - val_loss: 0.0426

Epoch 00010: val_loss improved from 0.04547 to 0.04264, saving model to /content/drive/Mj
CPU times: user 0.03 s, sys: 3.44 s, total: 13.4 s
Wall time: 17.7 s
```

СВЯЗЬ BATCH С КАЧЕСТВОМ ОБУЧЕНИЯ

LOSS vs. LEARNING RATE FOR DIFFERENT BATCH SIZES

BATCH SIZE:

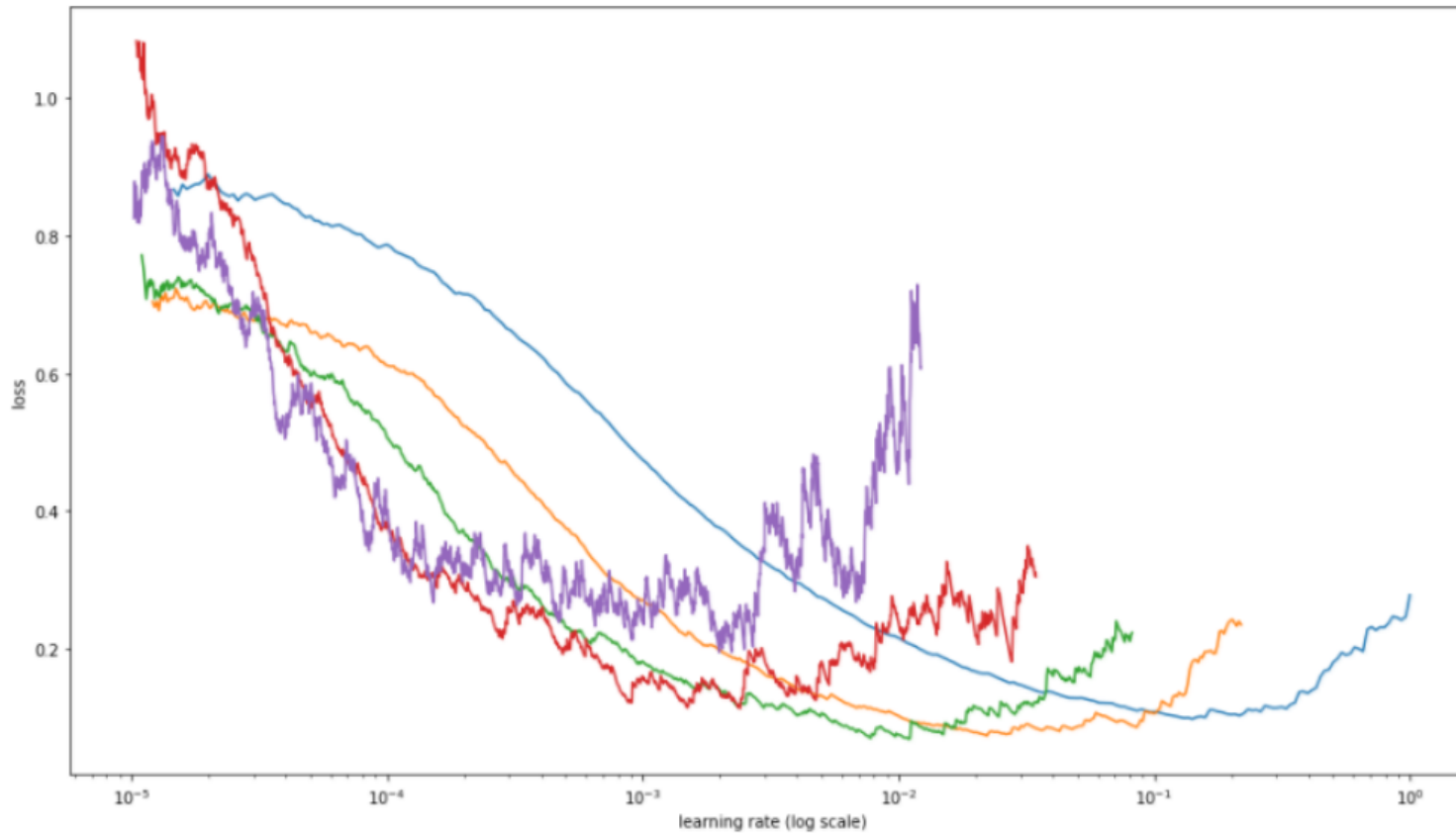
BS = 64

BS = 32

BS = 16

BS = 8

BS = 4



ПОТЕРИ (LOSS)

```
40/40 [=====] - ETA: 0s - loss: 0.0310 ████████████████████  
Epoch 00036: val_loss did not improve from 0.01216  
40/40 [=====] - 224s 6s/step - loss: 0.0310 - val_loss: 0.0298  
Epoch 37/40  
40/40 [=====] - ETA: 0s - loss: 0.0288 ████████████████████  
Epoch 00037: val_loss did not improve from 0.01216  
40/40 [=====] - 217s 5s/step - loss: 0.0288 - val_loss: 0.0242  
Epoch 38/40  
40/40 [=====] - ETA: 0s - loss: 0.0204 ████████████████████  
Epoch 00038: val_loss did not improve from 0.01216  
40/40 [=====] - 222s 6s/step - loss: 0.0204 - val_loss: 0.0139  
Epoch 39/40
```

ПЕРЕОБУЧЕННАЯ

Validation Loss

Training Loss

НЕДООБУЧЕННАЯ

Validation Loss

Training Loss

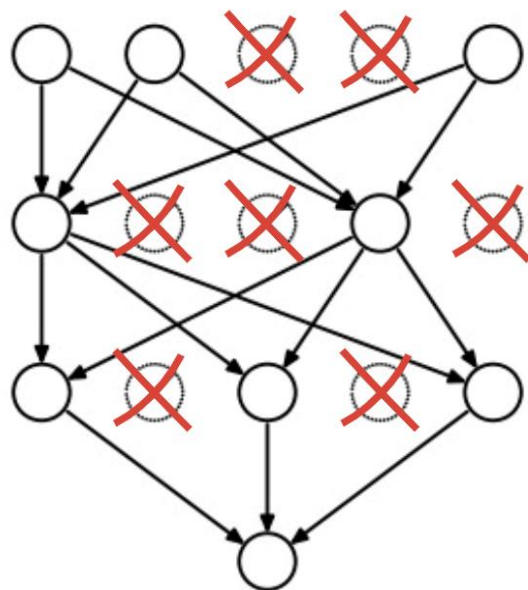
ЦЕЛЬ VAL_LOSS => 0

Validation Loss

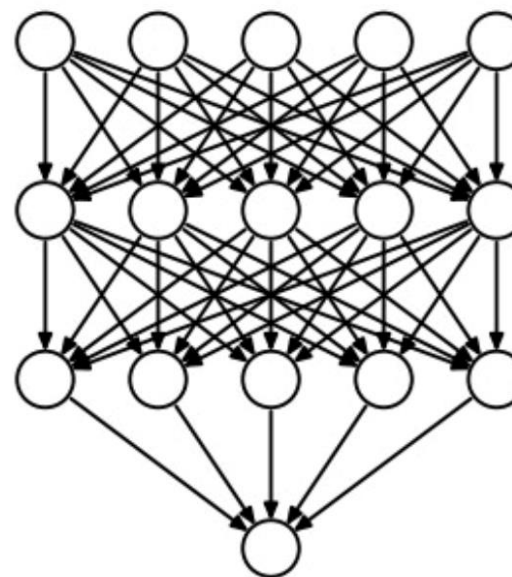
Training Loss

DROPOUT

ИСКЛЮЧАЕМ С ВЕРОЯТНОСТЬЮ p (ВЕРНЕТ 0)
ОБУЧАЕМ АНСАМБЛЬ, УСРЕДНЯЕМ \Rightarrow
УМЕНЬШАЕМ ПЕРЕОБУЧЕНИЕ



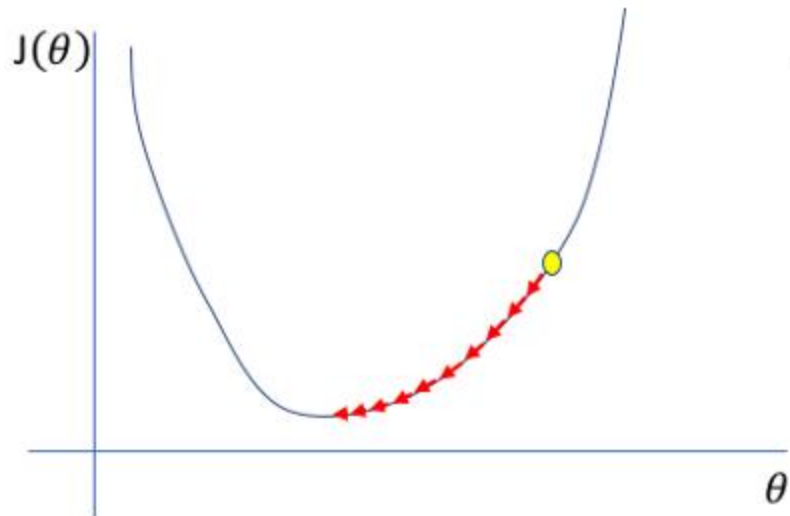
TRAINING
rate=0.4



EVALUATION
rate=0

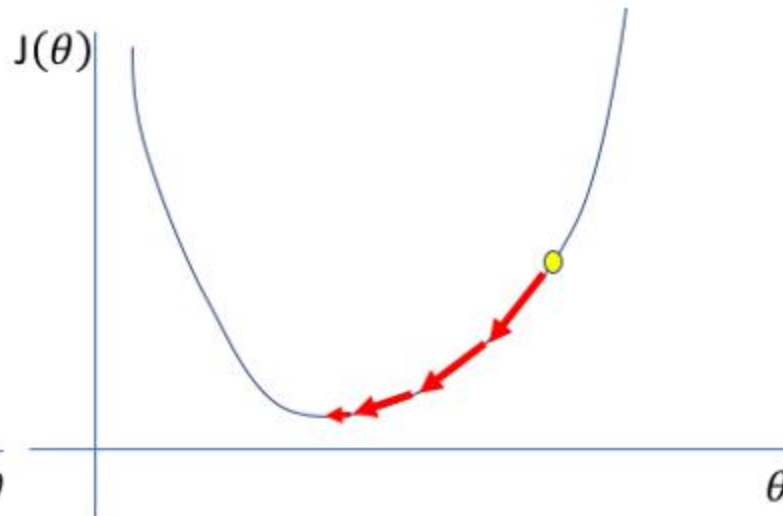
КОЭФФИЦИЕНТ ОБУЧЕНИЯ

Too low



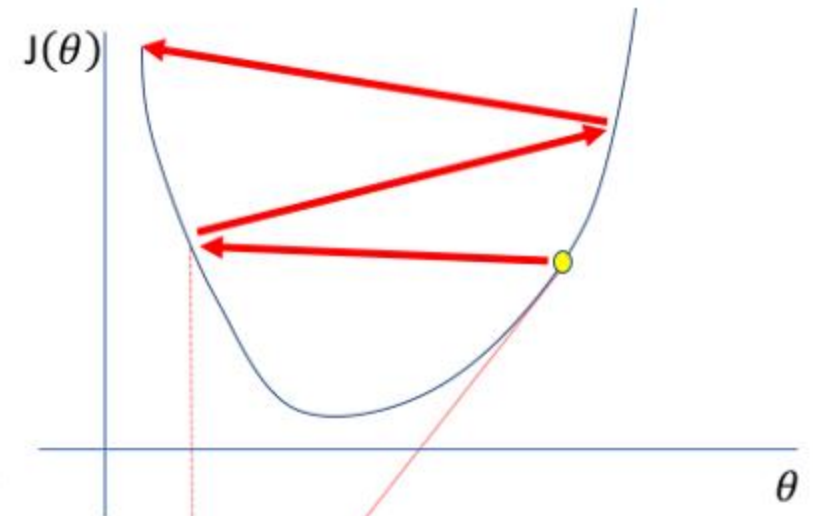
A small learning rate requires many updates before reaching the minimum point

Just right



The optimal learning rate swiftly reaches the minimum point

Too high

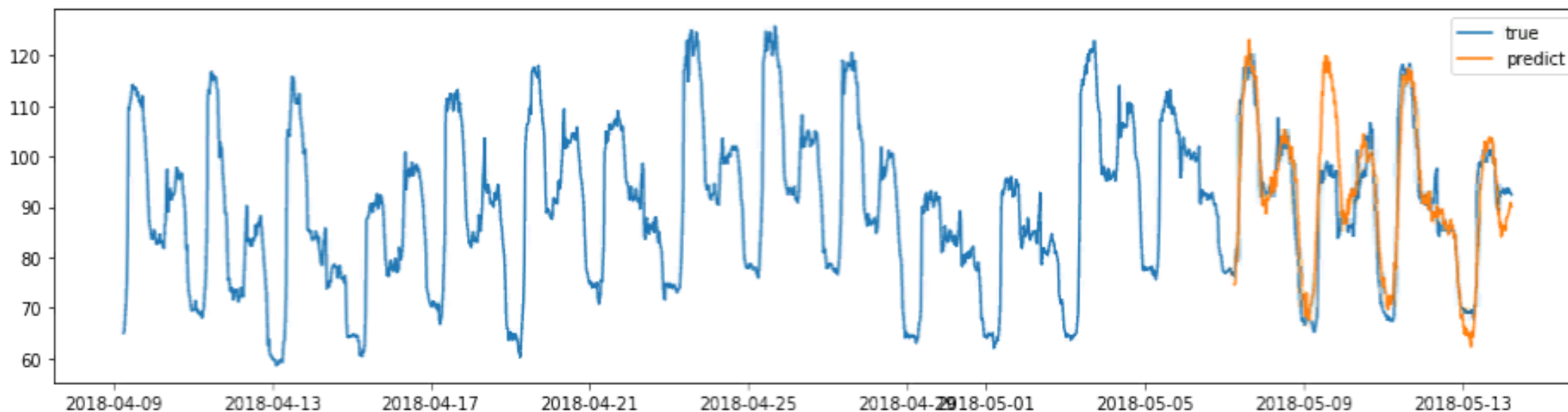
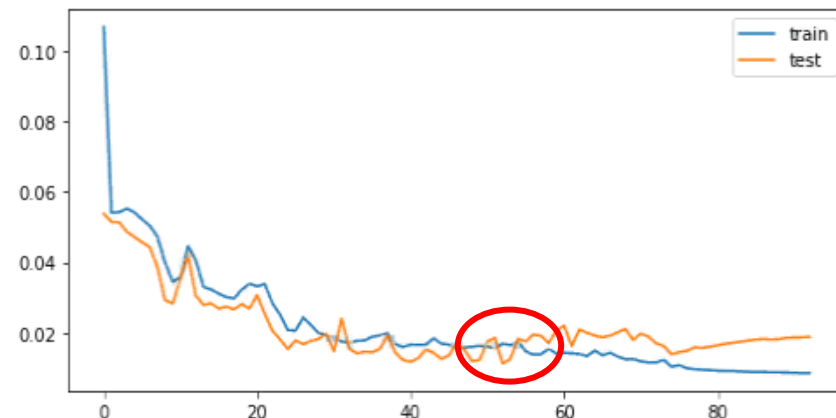
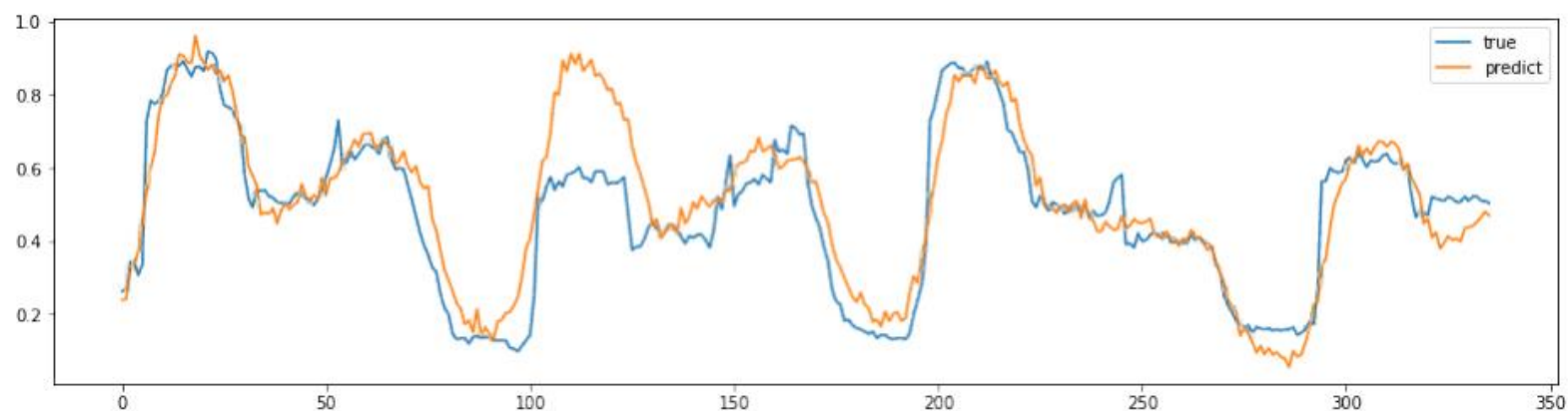


Too large of a learning rate causes drastic updates which lead to divergent behaviors

КАЧЕСТВО ПРОГНОЗИРОВАНИЯ

53-я ЭПОХА ИЗ 93, LOSS: 0.01132

ОШИБКА ПО ЭПОХАМ:



КАЧЕСТВО ПРОГНОЗИРОВАНИЯ

КОГДА ОСТАНОВИТЬ ОБУЧЕНИЕ?

КАК ПОДОБРАТЬ ПАРАМЕТРЫ ОБУЧЕНИЯ СЕТИ?

ВАРИАНТЫ ПОИСКА

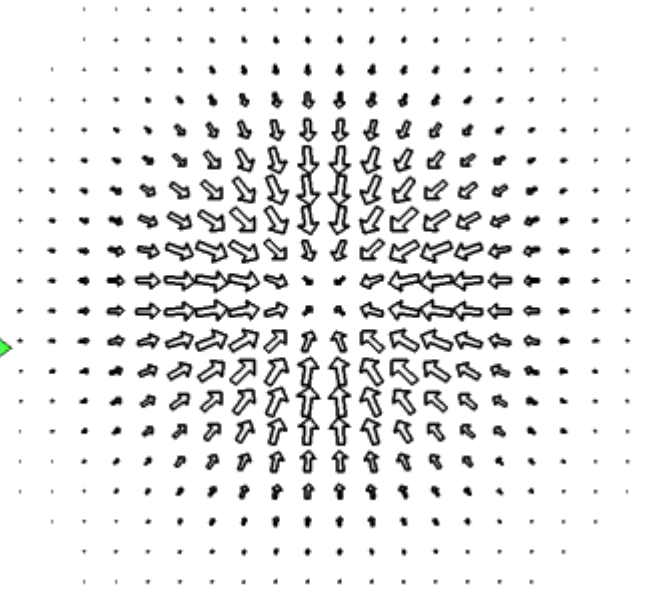
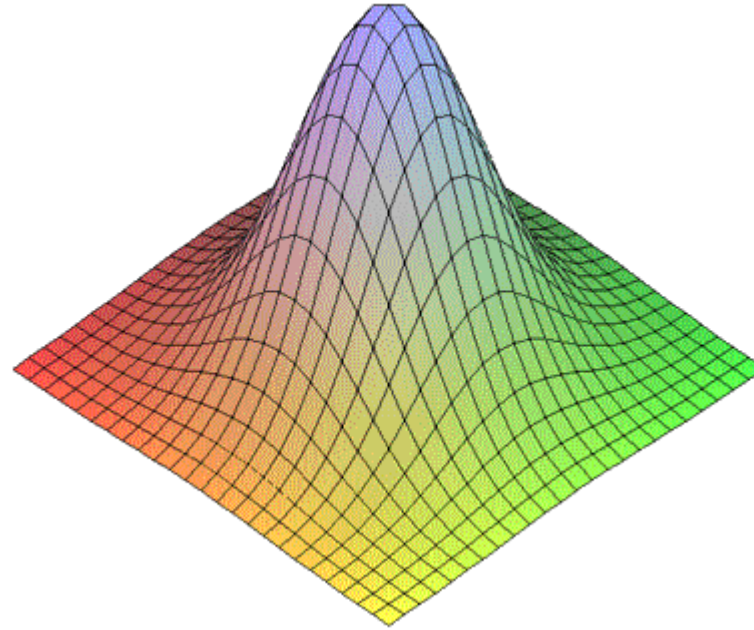
ИДТИ ПОСЛЕДОВАТЕЛЬНО

ИСКАТЬ СЛУЧАЙНО

ИСКАТЬ ПО ГРАДИЕНТУ

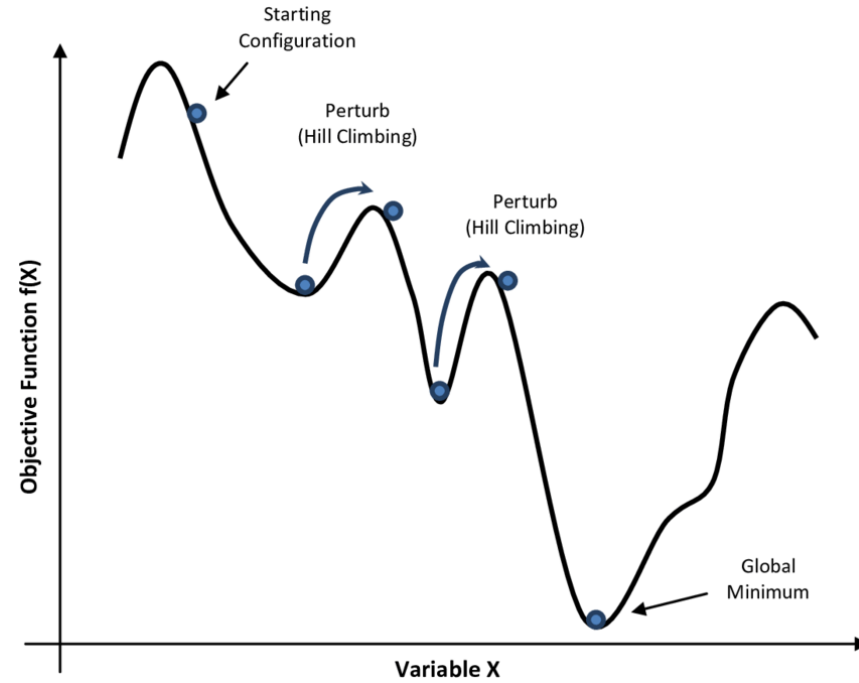
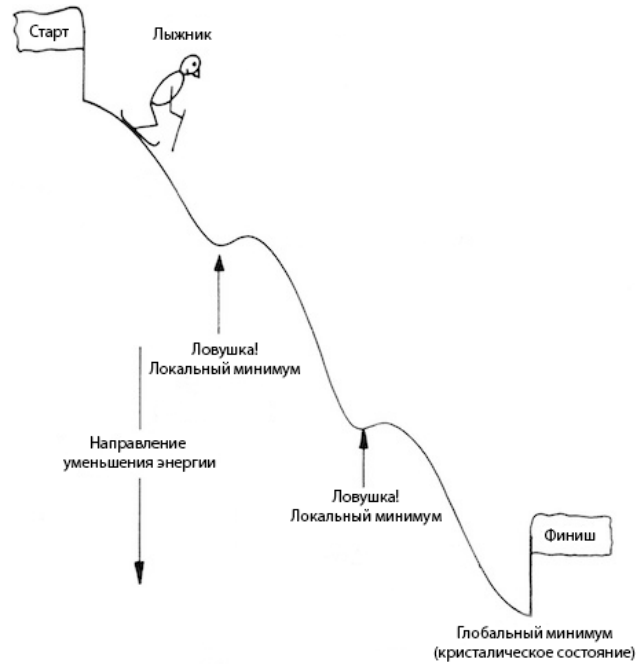
МЕТОД ОТЖИГА

...МНОЖЕСТВО ДРУГИХ

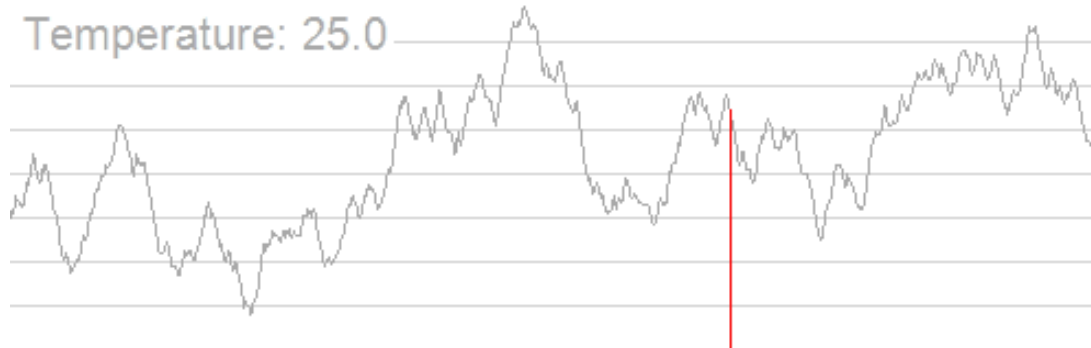
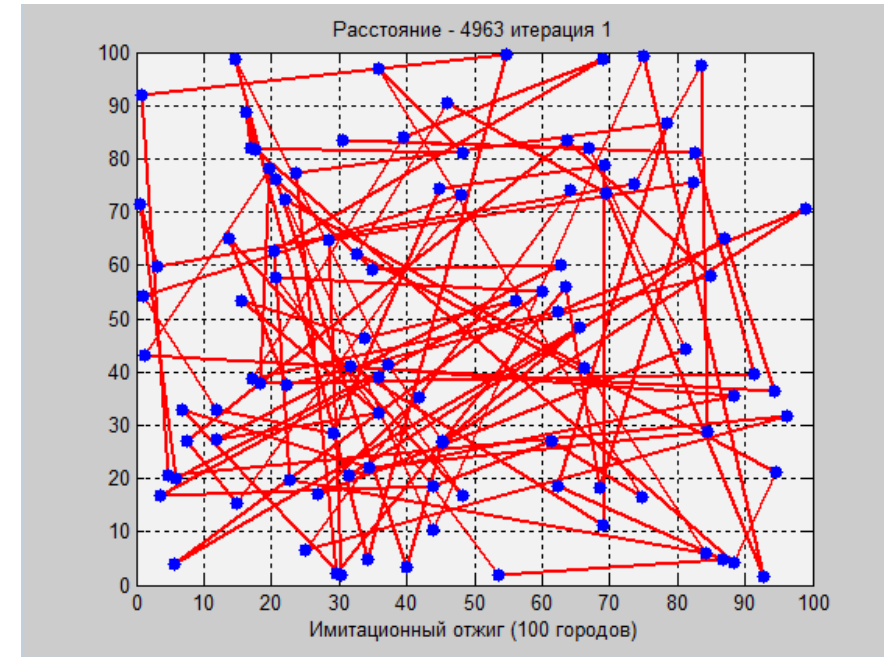


ГРАДИЕНТЫЙ БУРСИНГ

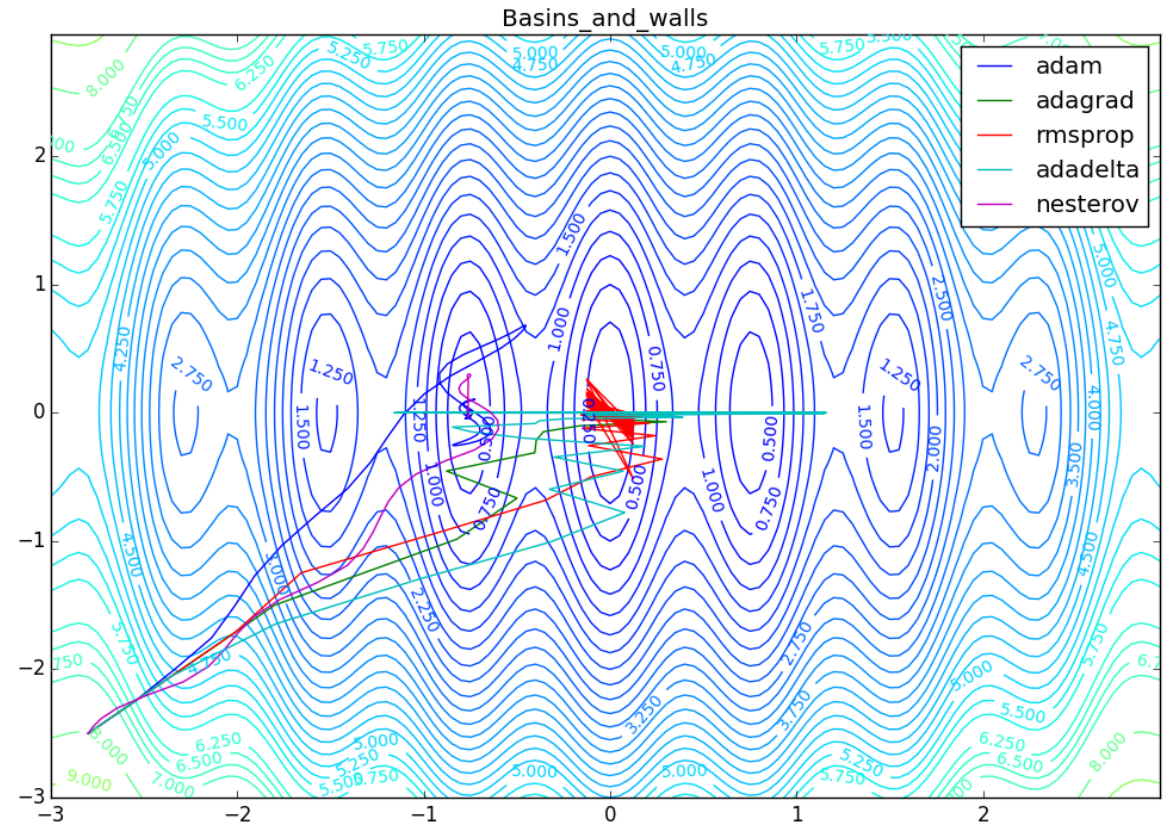
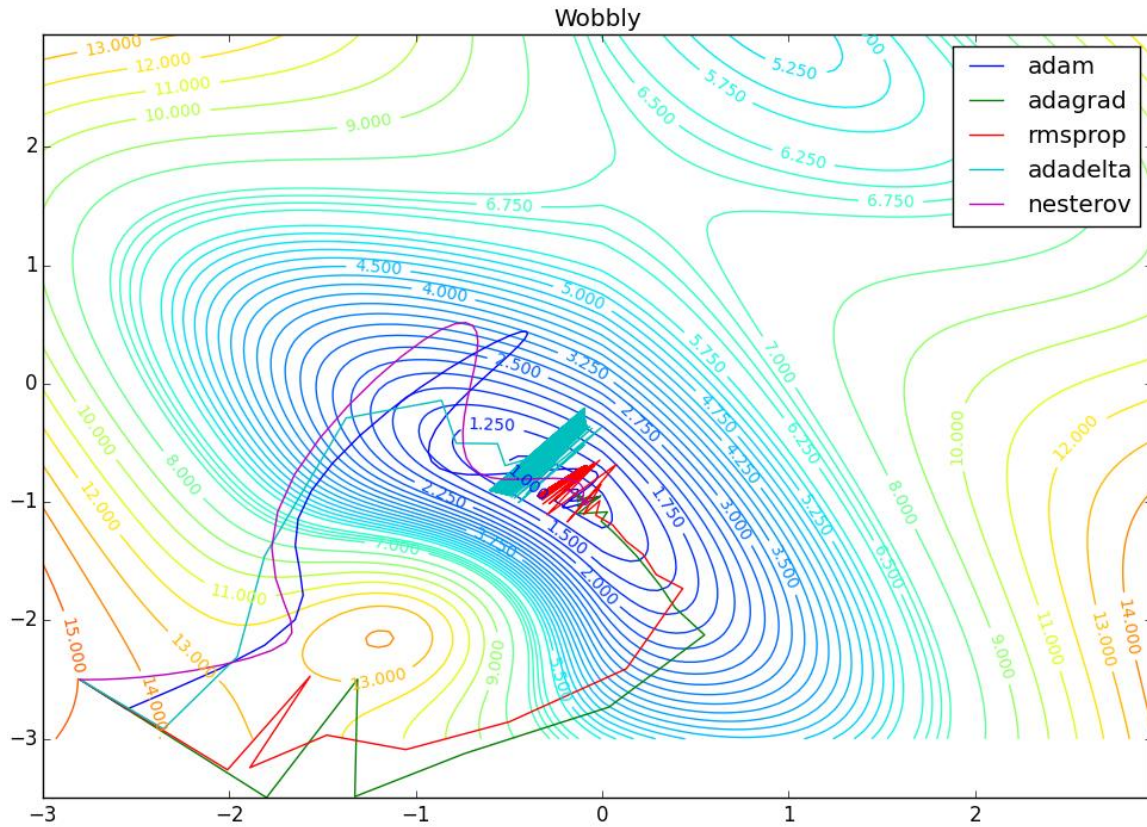
МЕТОД ОТЖИГА



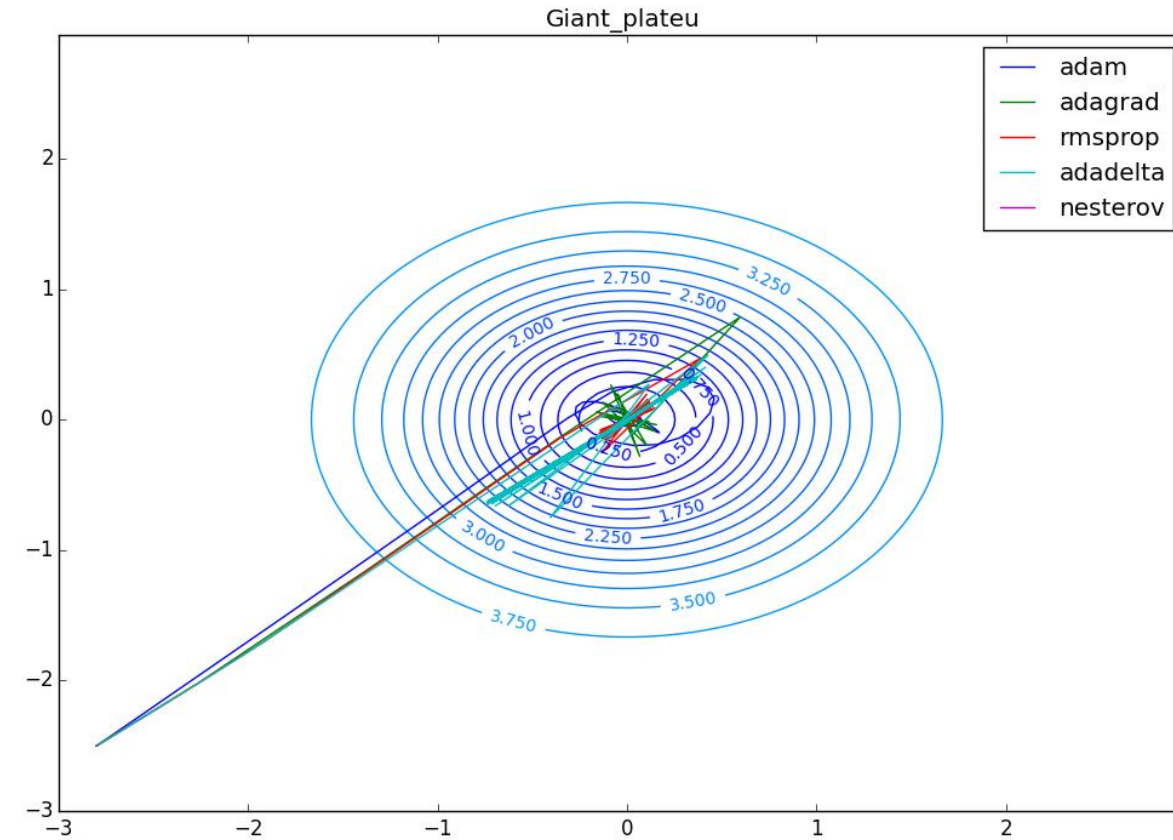
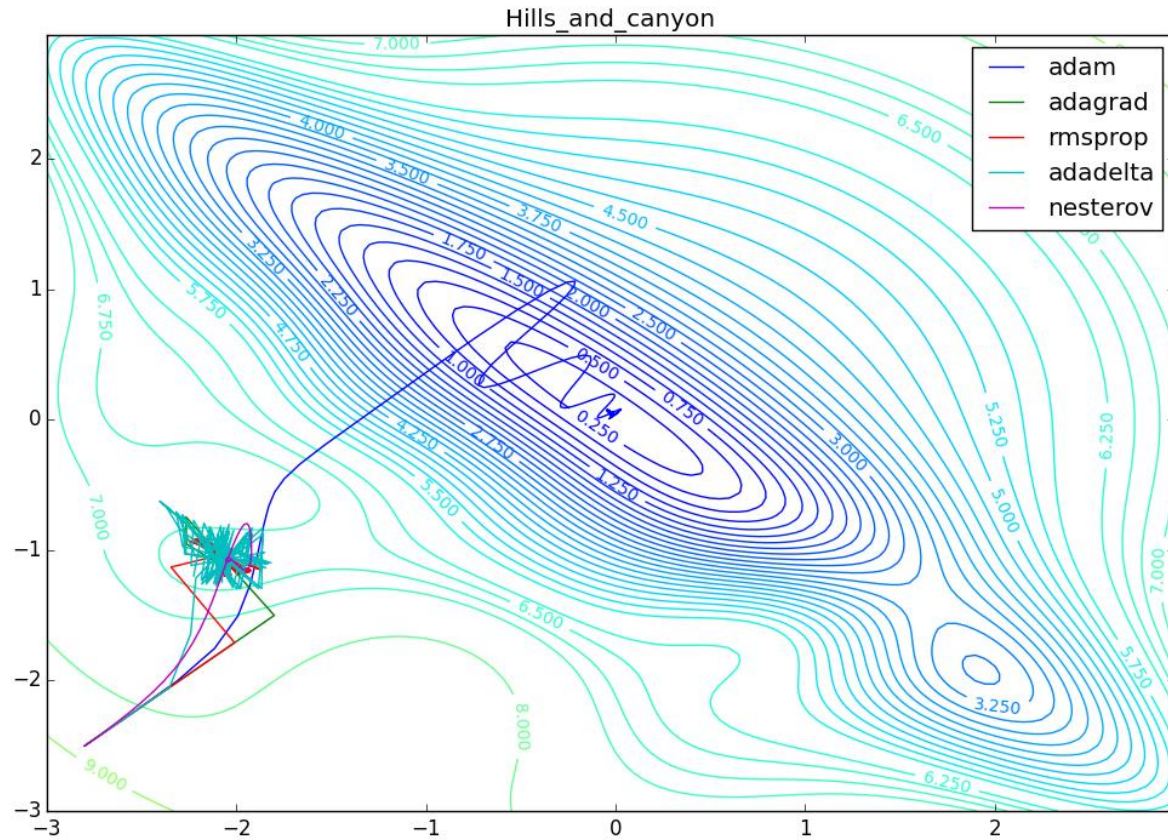
ЗАДАЧА КОММИВОЯЖЕРА



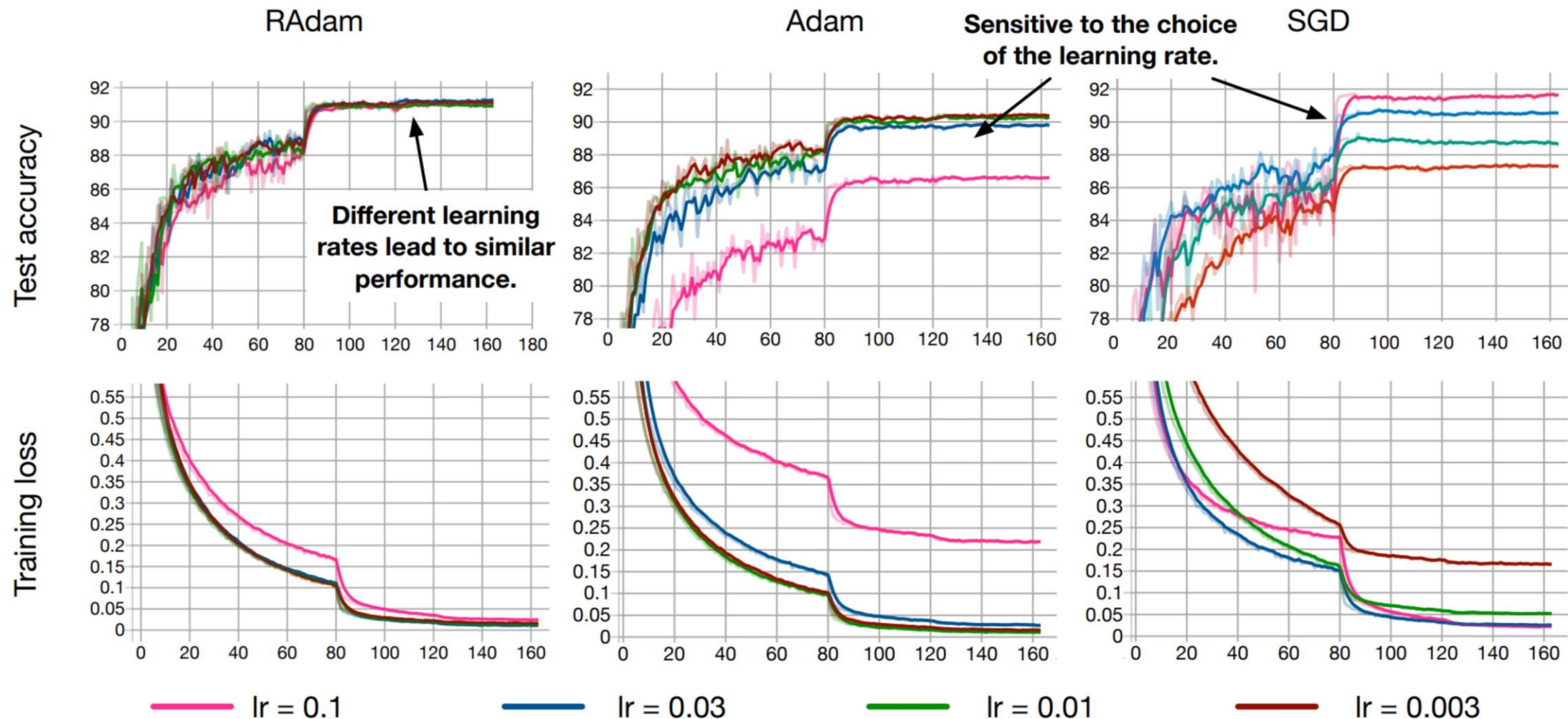
СРАВНЕНИЕ ФУНКЦИЙ ОПТИМИЗАЦИИ



СРАВНЕНИЕ ФУНКЦИЙ ОПТИМИЗАЦИИ

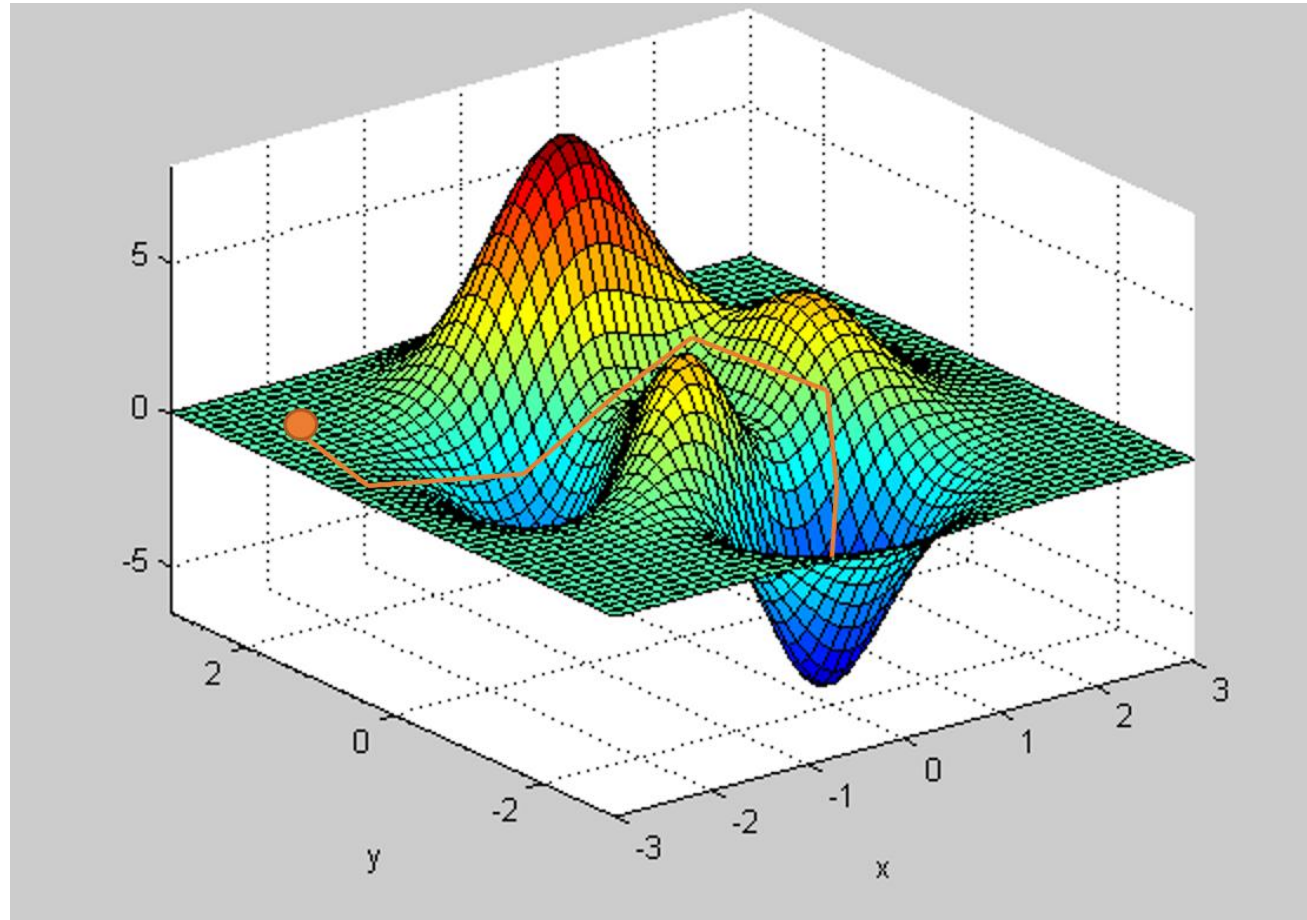


LR И АЛГОРИТМ ОПТИМИЗАЦИИ



КОГДА МЕНЯЕМ МНОЖЕСТВО ПАРАМЕТРОВ

МНОГОМЕРНОЕ ПРОСТРАНСТВО



ОПТИМИЗАЦИЯ HYPEROPT

```
space = {
    'units': hp.choice('units', [64, 128, 256, 512]),
    'dropout': hp.quniform('dropout', 0, .5, .01),
    'stack': hp.choice('stack', [True, False]),
    'patience': hp.choice('patience', [20, 40]),
}
```

(11/4) = 330 СОЧЕТАНИЙ

```
params: {'dropout': 0.31, 'patience': 20, 'stack': True, 'units': 128} | metric: 0.016431448981165886
```

```
=====
found better metric:
0.011289630085229874
```

```
params: {'dropout': 0.21, 'patience': 40, 'stack': True, 'units': 512} | metric: 0.011289630085229874
```

```
=====
```

```
params: {'dropout': 0.16, 'patience': 40, 'stack': False, 'units': 512} | metric: 0.01203224714845419
```

```
=====
```

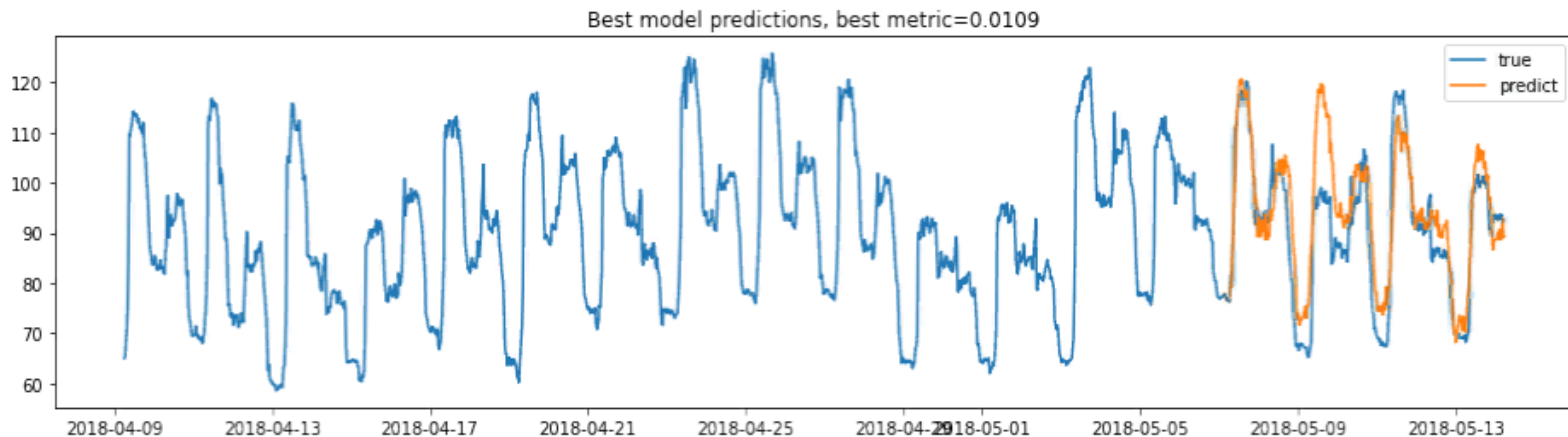
```
params: {'dropout': 0.48, 'patience': 40, 'stack': True, 'units': 64} | metric: 0.014453343115746975
```

```
=====
```

```
params: {'dropout': 0.34, 'patience': 40, 'stack': False, 'units': 256} | metric: 0.012039628811180592
```

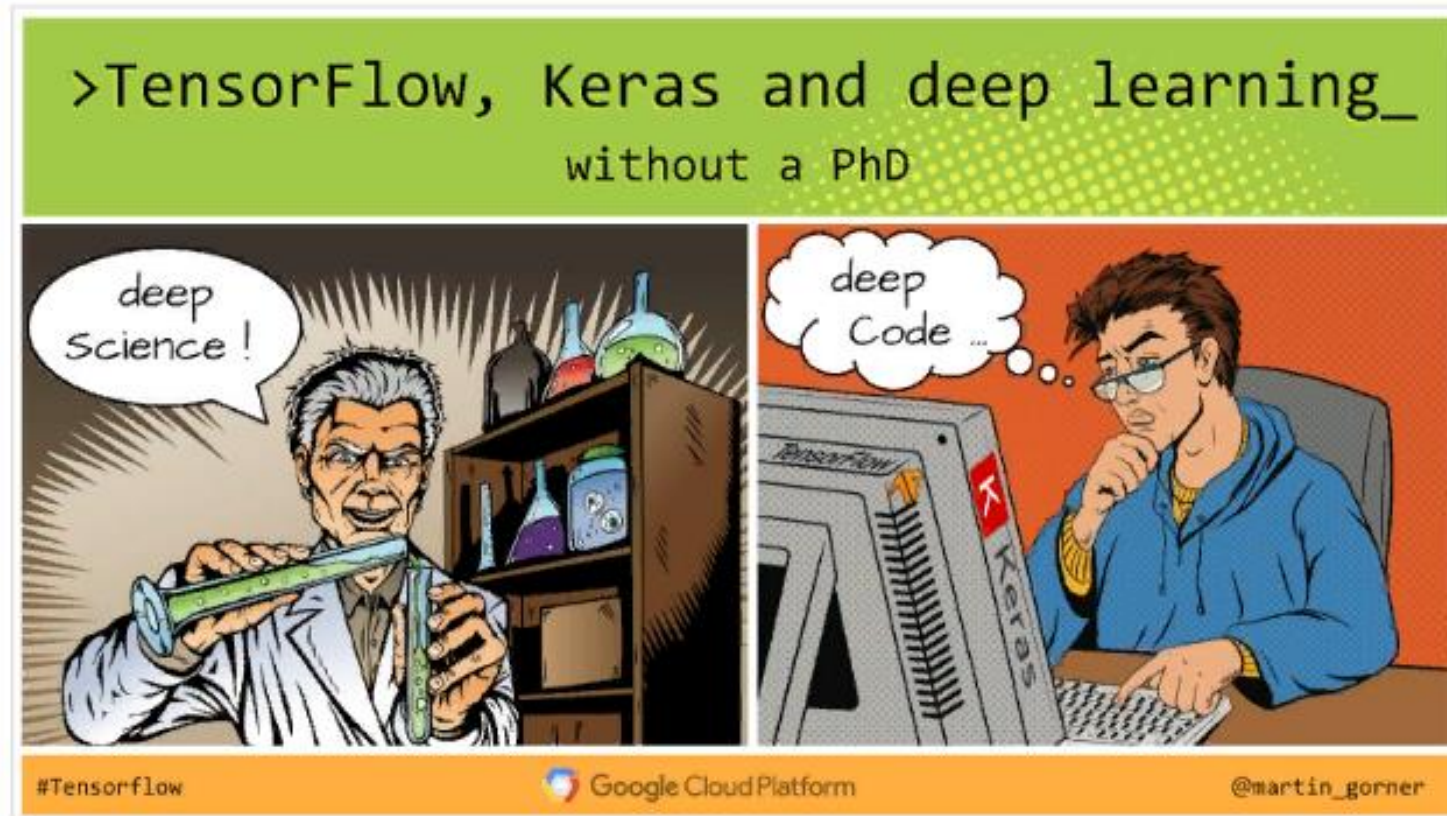
```
=====
found better metric:
0.010894561186432838
```

```
params: {'dropout': 0.2, 'patience': 20, 'stack': False, 'units': 512} | metric: 0.010894561186432838
```



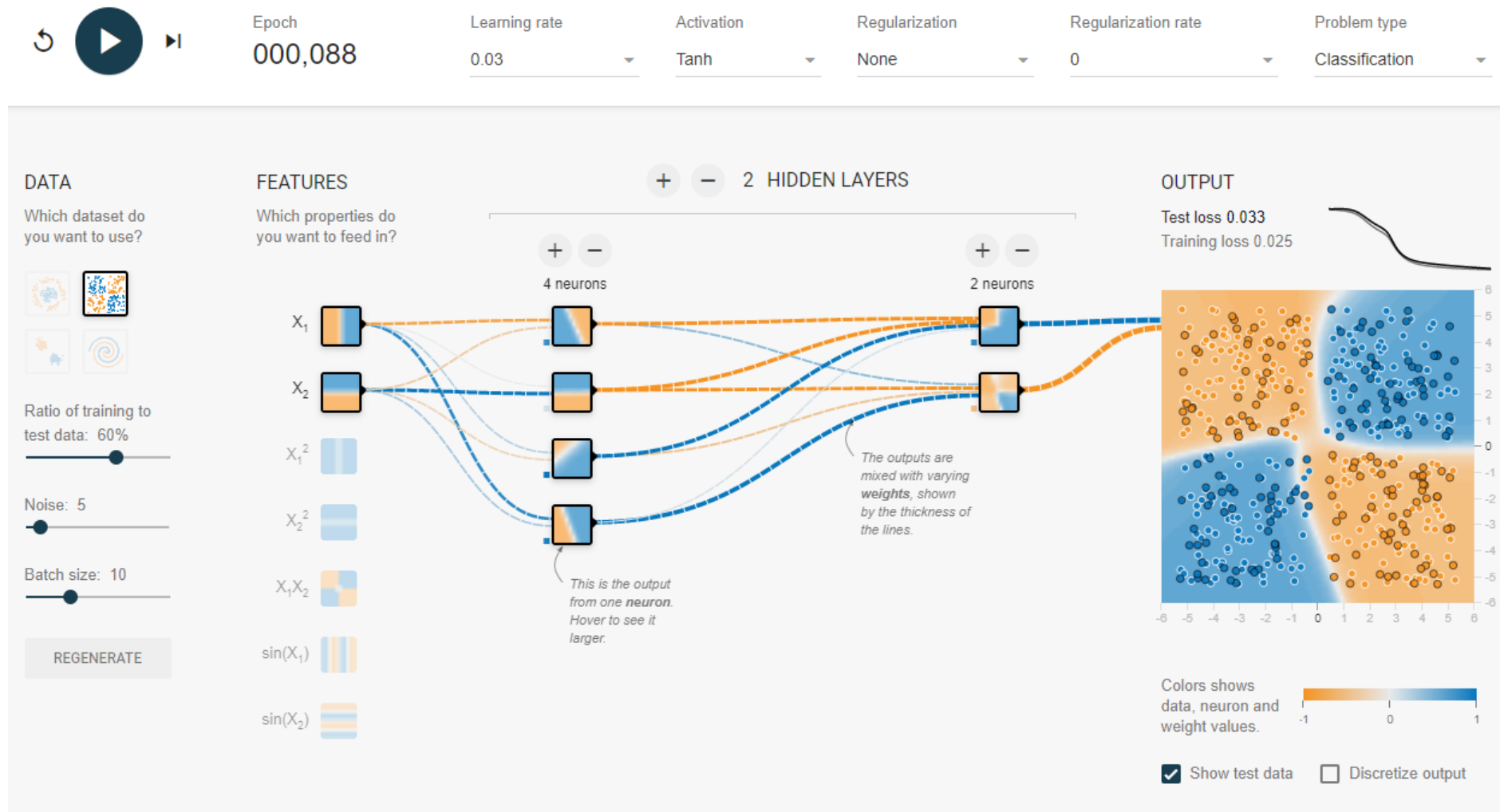
ПОДРОБНЕЕ ПРО ПАРАМЕТРЫ И СЛОИ

[HTTPS://CODELABS.DEVELOPERS.GOOGLE.COM/CODELABS/CLOUD-TENSORFLOW-MNIST?HL=EN](https://code-labs.developers.google.com/code-labs/cloud-tensorflow-mnist?hl=en)



ПОИГРАТЬСЯ БЕЗ КОДА С TENSORFLOW

[HTTPS://PLAYGROUND.TENSORFLOW.ORG](https://playground.tensorflow.org)



ДОРОЖНАЯ КАРТА МЕРОПРИЯТИЙ

СКРАПИНГ



ТЕКСТ



- Векторная оценка схожести предложений
- Классификация текста fasstext
- Предсказание текста, нейронные сети

ВРЕМЕННЫЕ РЯДЫ



- Прогнозирование на нейронных сетях
- AutoML (hyperopt)

ИЗОБРАЖЕНИЯ



- Классификация на нейронных сетях
- Детекция CV и нейронные сети

UPWORK ИЛИ СТАРТАП

КОНТАКТЫ

ОБСУЖДАЕМ

<https://t.me/devdvAI>

<https://t.me/devdvStartup>



РЕПОЗИТОРИЙ

<https://github.com/akumidv/startup-khv-ai-study>

АНДРЕЙ КУМИНОВ

+7 914 770 5846

<https://facebook.com/akuminov>

<https://vk.com/akumidv>