

# Сверточные импульсные нейронные сети для классификации изображений

Михаил Киселев

Старший исследователь



# Мотивация

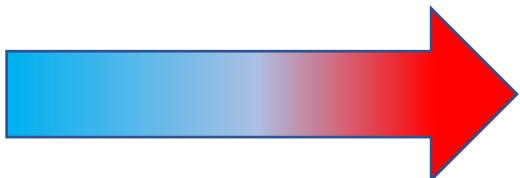
- Сверточная сеть – необходимый элемент всех нейронных сетей, показывающих высокую точность классификации изображений.
- В режиме работы (inference) прямой перенос идеи сверточных сетей в импульсный домен не вызывает проблем.
- В режиме обучения такой перенос проблематичен, так как нарушает принцип локальности – если каждая позиция сверточной матрицы реализуется отдельным нейроном, это означает, что один акт изменения элемента этой матрицы соответствует одновременному однаковому изменению синаптического веса сразу у многих нейронов.



Решению данной проблемы и посвящено настоящее исследование.

# Идея нашего метода вытекает из 2 основных постулатов

1. Недопустимы нелокальные операции над синаптическими весами (no weight sharing).
2. Свертки зависят от области применения, а не от конкретной задачи – для разных задач из одной области могут быть применены одни и те же свертки.



Сверточные матрицы могут быть получены на основе большой репрезентативной выборки изображений из области предполагаемого применения сети и зафиксированы в виде непластичных слоев ИмНС. Обучение решению конкретных задач классификации будет при этом производится вышележащими слоями на основе архитектуры CoLaNET.

Алгоритм получения сверточных  
матриц для изображений из  
данной прикладной области

### **Входные данные:**

- Большой набор изображений (матриц яркостей пикселей)  $\mathbf{G}$
- Размер сверточных матриц  $K$
- Число сверточных матриц  $N_C$
- Сдвиг свертки  $s$

### **Гиперпараметры алгоритма:**

- Пороговая яркость пикселя  $B$
- Нижняя граница значения элементов сверточной матрицы  $w_{min} (<0)$
- Верхняя граница значения элементов сверточной матрицы  $w_{max}$
- Скорость обучения  $l$

### **Результат алгоритма:**

Набор сверточных матриц  $\mathbf{w} = w_{aij}$ , где  $0 \leq a < N_C, 0 \leq i, j < K$

Алгоритм состоит из итераций. Каждая итерация соответствует одному случайно взятому изображению из  $\mathbf{G}$ . Возможно несколько эпох.

Кроме  $\mathbf{w}$  в алгоритме участвует вспомогательный тензор  $\mathbf{W}$ , той же размерности, что  $\mathbf{w}$ .

Элементы  $\mathbf{W}$  и  $\mathbf{w}$  всегда связаны соотношением

$$w(W) = w_{min} + \frac{(w_{max} - w_{min})\max(W, 0)}{w_{max} - w_{min} + \max(W, 0)}$$

и передаются из итерации в итерацию.

В начале алгоритма все элементы  $\mathbf{w}$  и  $\mathbf{W}$  равны 0 и  $-w_{min}(w_{max} - w_{min})/w_{max}$ , соответственно.

# Формальная запись одной итерации алгоритма

```
c ← the tensor of convolutions of g with w with the stride s.
E ← Ø
while max(c) > 0 & |E| < NC
    <a, p, q> ← indices of random element of c such that capq = max(c)
    if a ∉ E
        E ← E ∪ {a}
        gw ← K×K tile of g with upper left corner coordinates (ps, qs)
        if max(c) < 1
            nb ← the number of pixels in gw brighter than B.
            if 0 < nb < K2
                for all 0 ≤ i, j < K
                    Waij ←  $\begin{cases} W_{aij} + l & \text{if } g_{ij}^w > B \\ W_{aij} - \frac{ln_b}{K^2 - n_b} & \text{o/w} \end{cases}$ 
                    waij ← w(Waij)
                end for
            end if
        end if
    else
        p ← 0, nb ← 0, k ← 0
        r ← set of all values in gw in decreasing order
        while p < 1
            for all 0 ≤ i, j < K
                if gijw = rk
                    p ← p + rk waij
                    Waij ← Waij + l
                    waij ← w(Waij)
                    nb ← nb + 1
                end if
            end for
            k ← k + 1
        end while
        for all 0 ≤ i, j < K
            if gijw ≤ rk
                Waij ← Waij -  $\frac{ln_b}{K^2 - n_b}$ 
                waij ← w(Waij)
            end if
        end for
    end if
    cbrt ← 0 for all <b, r, t> for which r = p & t = q
end while
```

# Неформальное описание одной итерации алгоритма

- 1. Считаем свертки нового изображения с существующим тензором сверток**
- 2. Для каждой свертки находим самое большое значение, но так, чтобы в этой же позиции другие свертки не имели больших значений.**
- 3. Если это значение мало (< 1), увеличиваем те элементы сверточной матрицы, которые соответствуют ярким пикселям, остальные элементы соответственно уменьшаем.**
- 4. Если это значение велико, находим самые яркие пиксели пока их сумма, взвешенная соответствующими сверточными коэффициентами не становится большей 1. Все соответствующие им элементы сверточной матрицы увеличиваем, остальные соответственно уменьшаем.**

# Портирование сверточных матриц в сверточную импульсную нейронную сеть

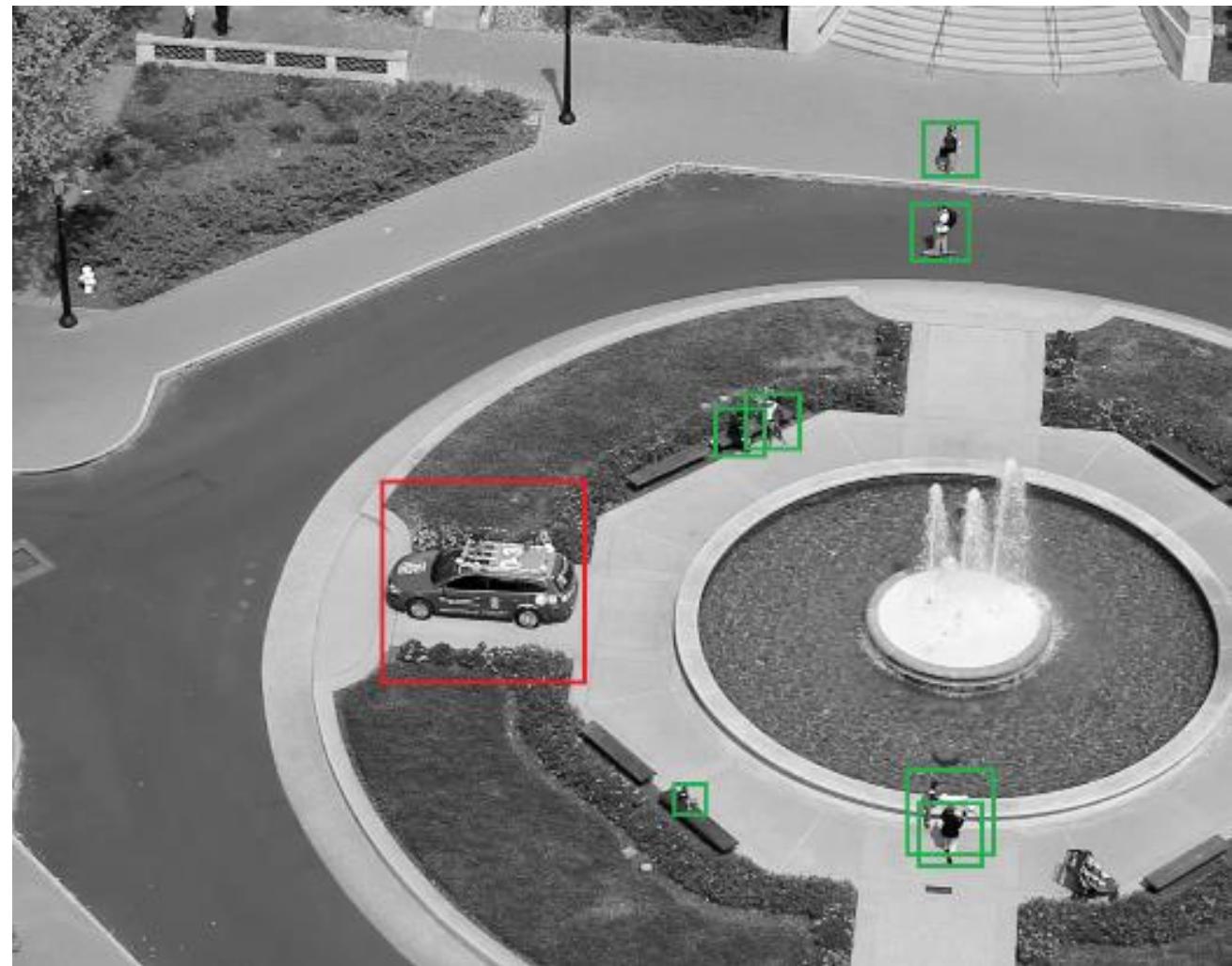
- Синаптические веса нейронов сверточного слоя пропорциональны элементам  $w$ . Коэффициент пропорциональности устанавливается из расчета заданной средней частоты срабатывания нейронов сверточного слоя.
- Так как для случая частотного кодирования реализовать max pooling сложно, используется mean pooling (точнее, sum pooling). Это соответствует нейрону, у которого все синаптические веса чуть больше порогового потенциала.
- Для получения точного sum pooling изменение мембранныго потенциала нейрона при срабатывании чуть изменено по сравнению со стандартной моделью LIF:

$$if \ u \geq u_{THR}, u \leftarrow u - u_{THR}, firing$$

Применение сверточной  
импульсной нейронной сети для  
классификации объектов в  
датасете Neovision2

# Этапы предобработки изображений в датасете Neovision2

## Выделение объектов



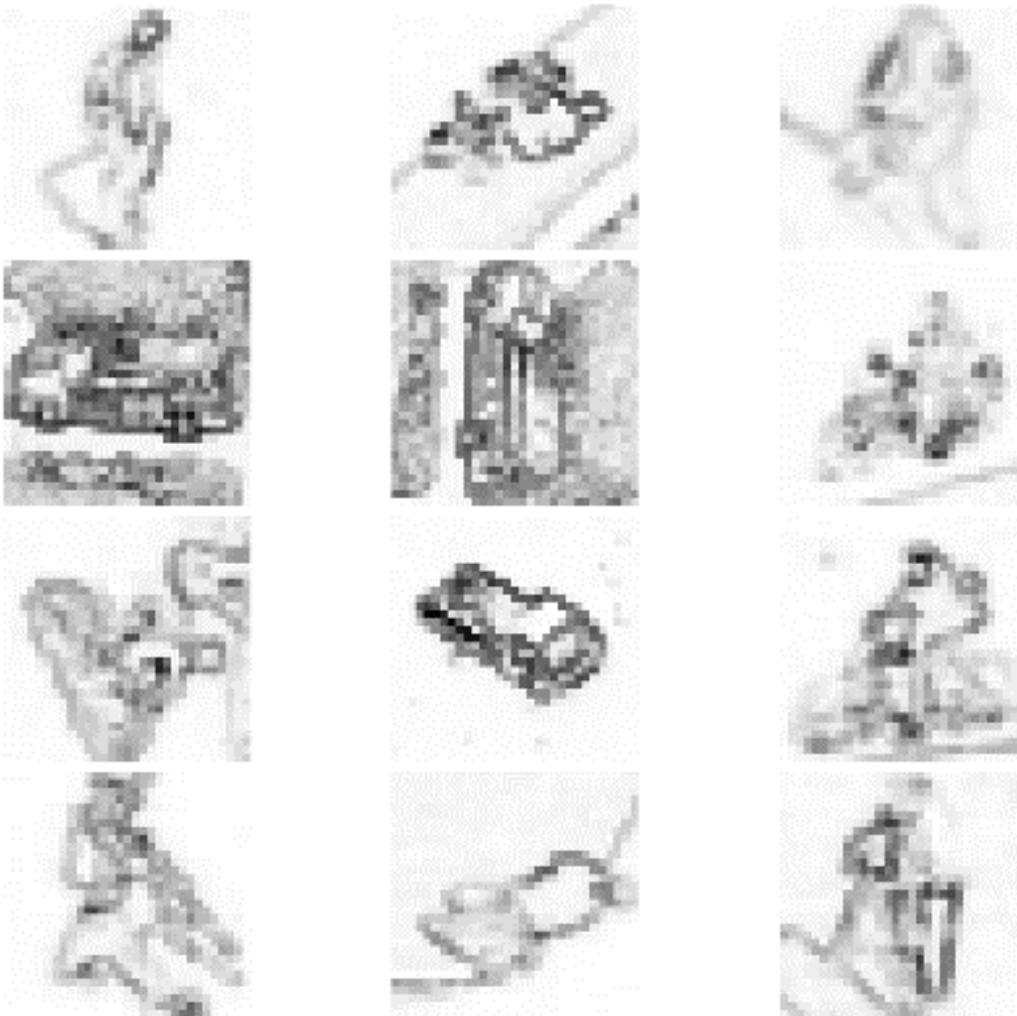
Этапы предобработки  
изображений в  
датасете Neovision2

Построение разностной  
термокарты

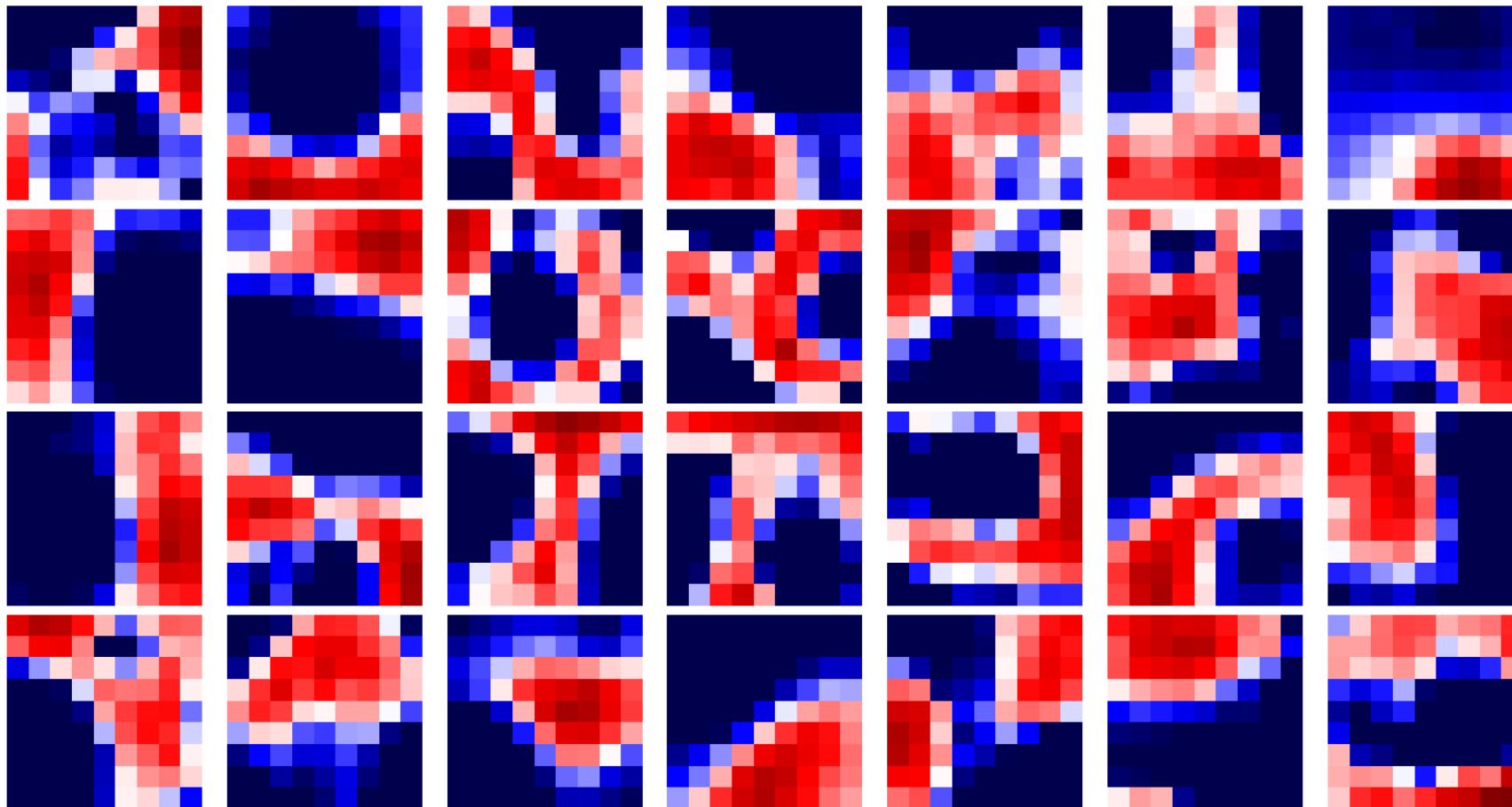


Этапы предобработки изображений в датасете Neovision2

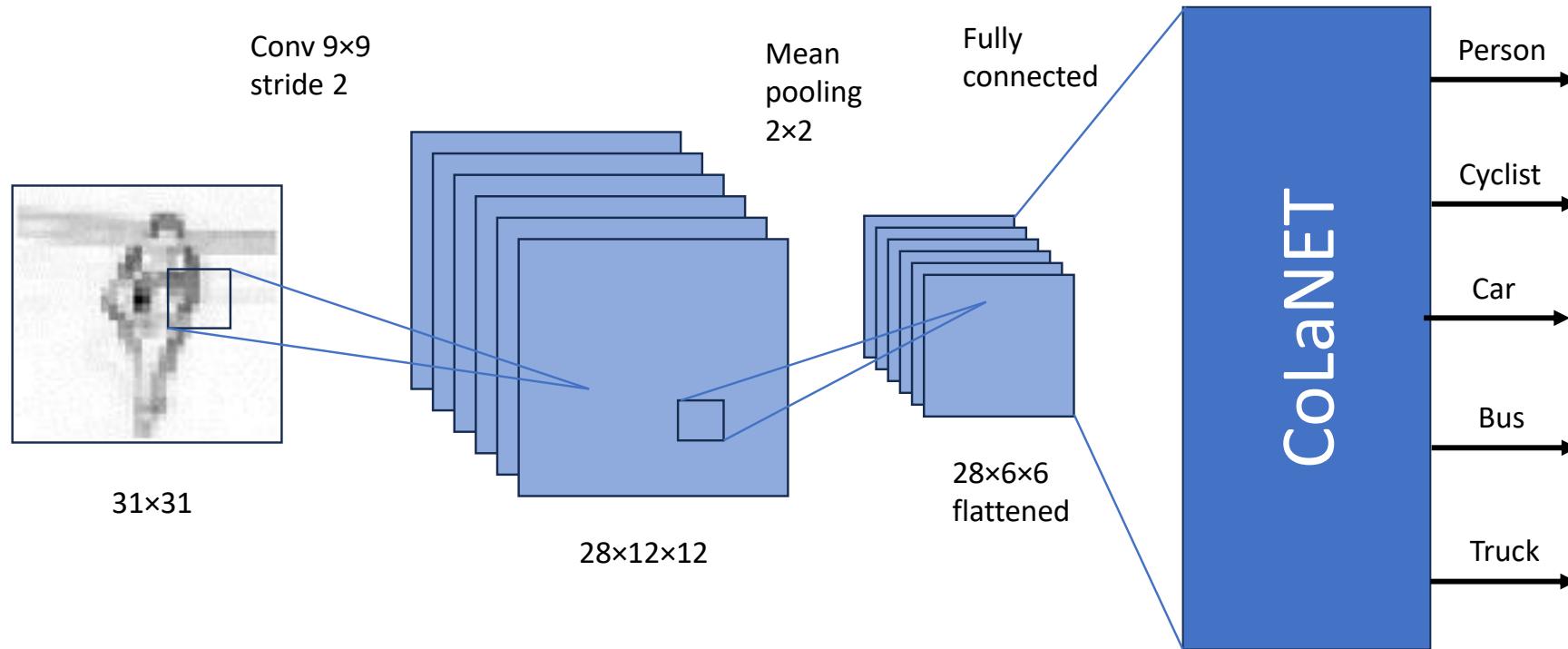
Создание массива распознанных объектов



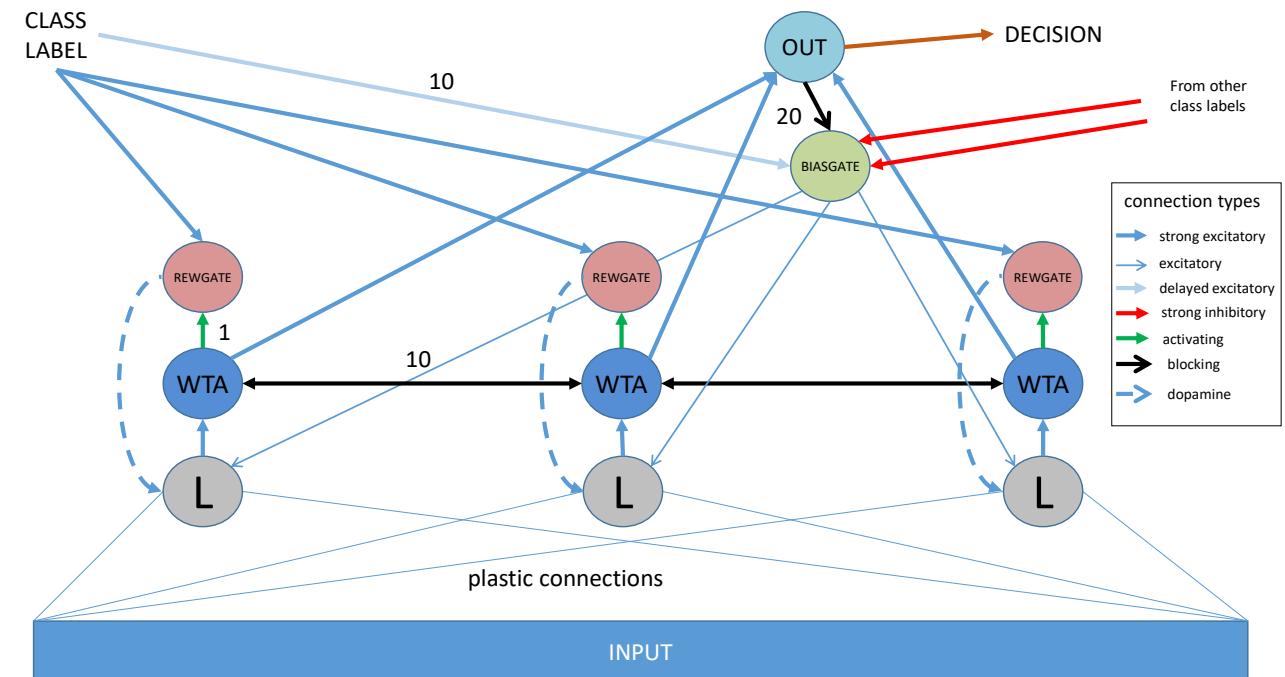
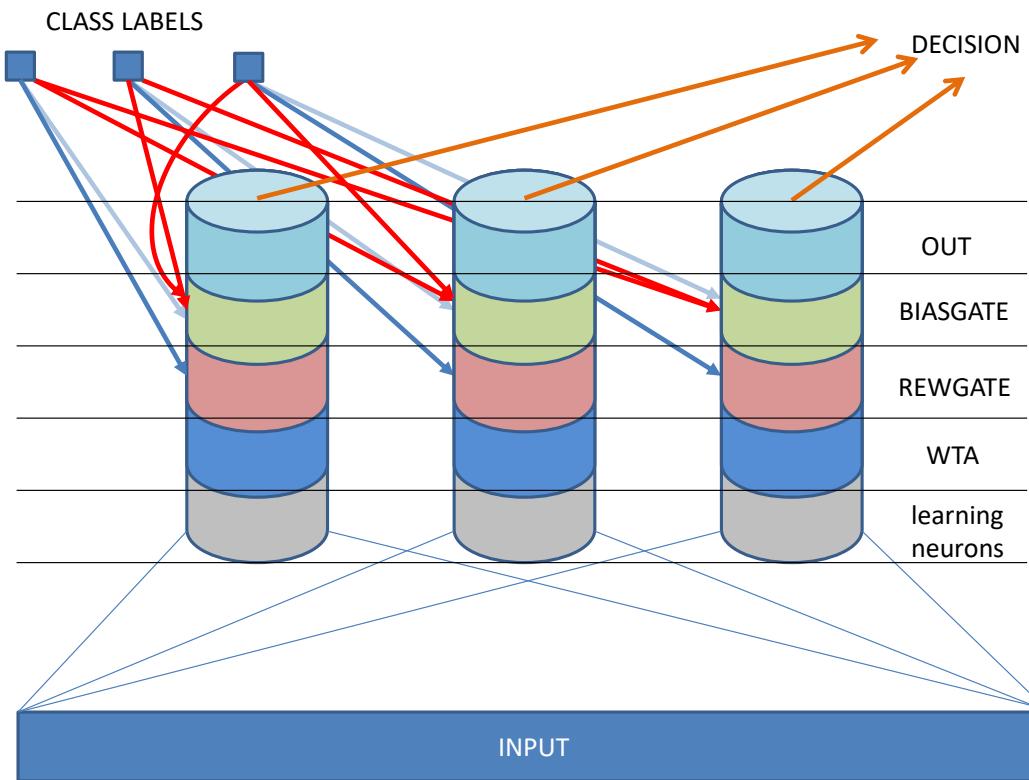
# Полученные сверточные матрицы



# Структура классифицирующей сверточной импульсной нейросети



# Схематичная структура сети CoLaNET



# Сравнение результатов

# С чем сравнивали

**CNN1.** Классическая сверточная сеть, дающая точность >99% на MNIST

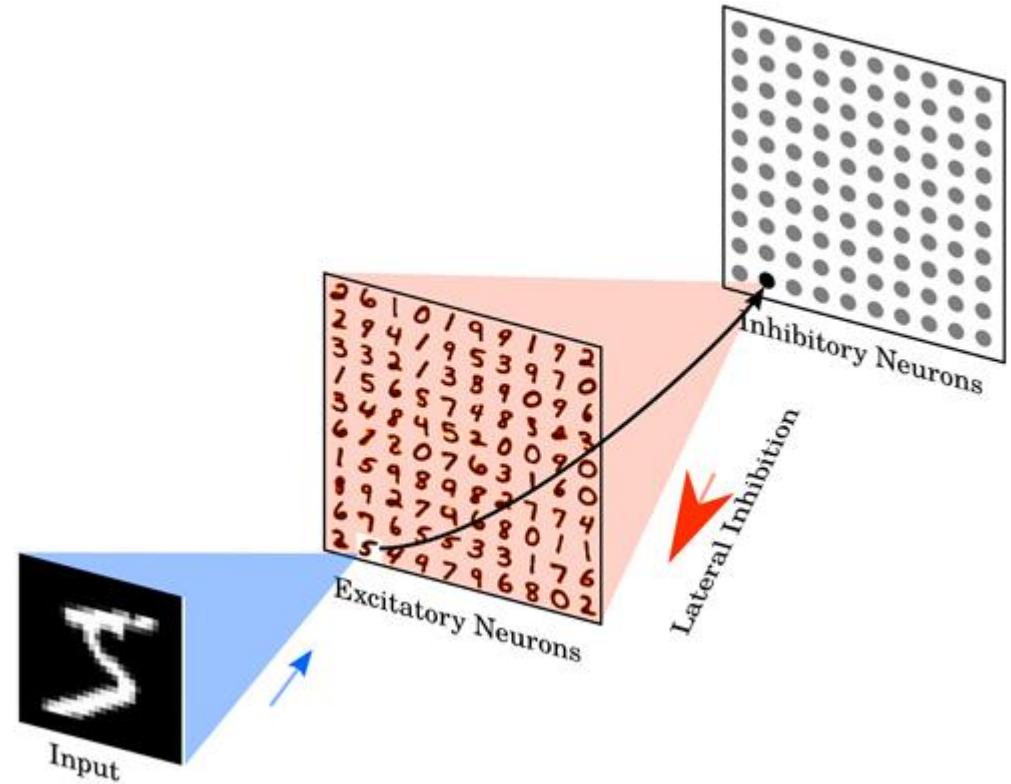
1. convolutional  $32 \times 4 \times 4$ , stride 1, activation - ReLu
2. max pooling  $2 \times 2$
3. convolutional  $64 \times 3 \times 3$ , stride 1, activation - ReLu
4. max pooling  $2 \times 2$
5. flatten, dropout 0.5
6. fully connected, activation – softmax, 5 target classes

**CNN2.** Сверточная сеть, наиболее близкая по структуре к нашей

1. convolutional  $28 \times 9 \times 9$ , stride 2, activation - ReLu
2. max pooling  $2 \times 2$
3. flatten, dropout 0.5
4. fully connected, activation – softmax, 5 target classes

**ULSNN.** Импульсная сеть с обучающимся без учителя слоем

P. Diehl , M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity", Frontiers in Computational Neuroscience, vol. 9, 1015, <https://www.frontiersin.org/journals/computational-neuroscience/articles/10.3389/fncom.2015.00099>



# Результаты сравнения

<b>Сеть</b>	<b>Точность, %</b>	
	<i>средняя</i>	<i>стандартное отклонение</i>
CNN1	94.35	0.32
CNN2	92.34	0.42
ULSNN	78.30	0.28
CSNN	91.58	1.1

# Аналитическое исследование динамики обучения CoLaNET

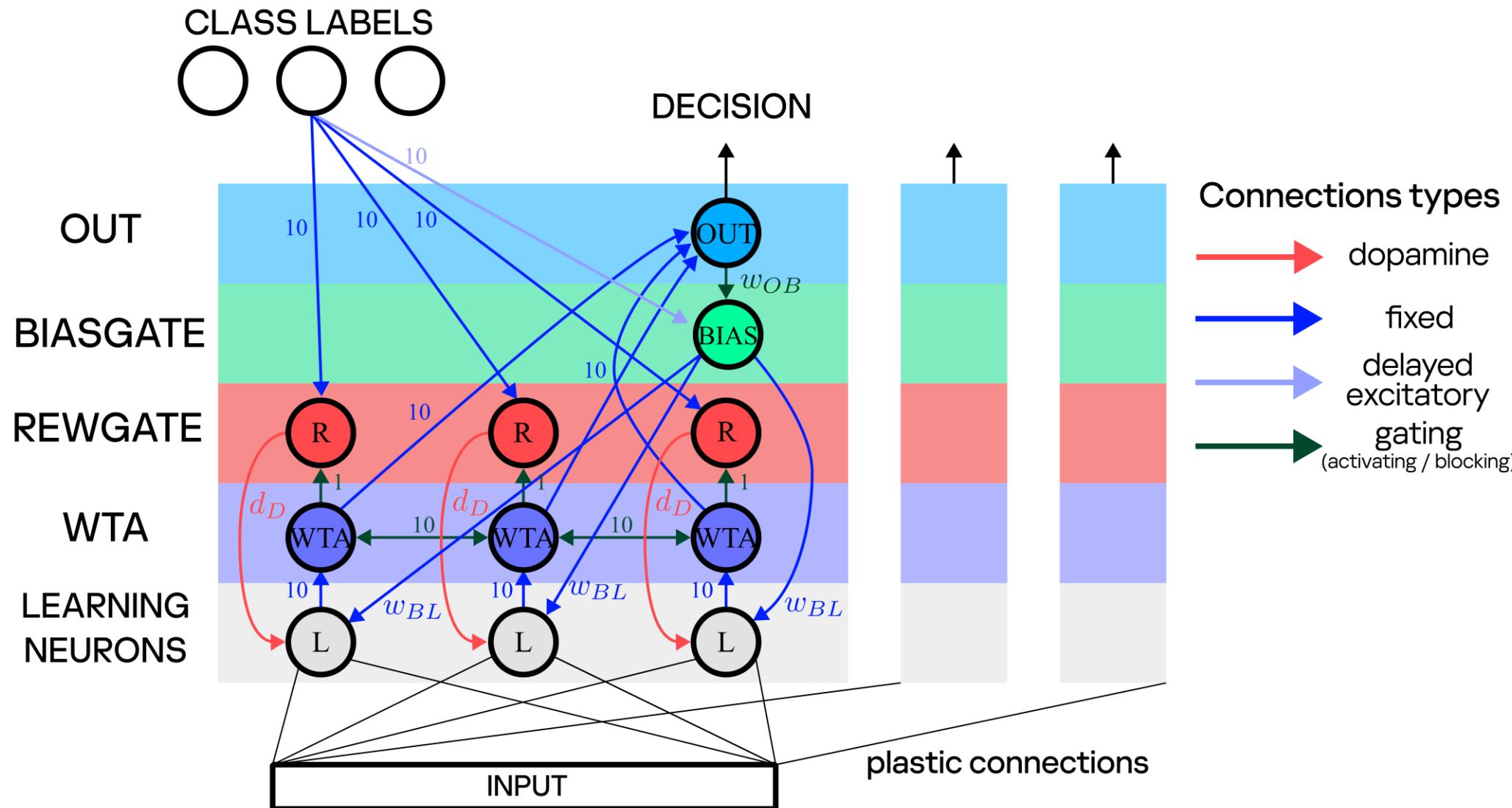
Олег Горюнов, Владимир Клиньшов  
НИУ ВШЭ Нижний Новгород

Михаил Киселев  
Kaspersky

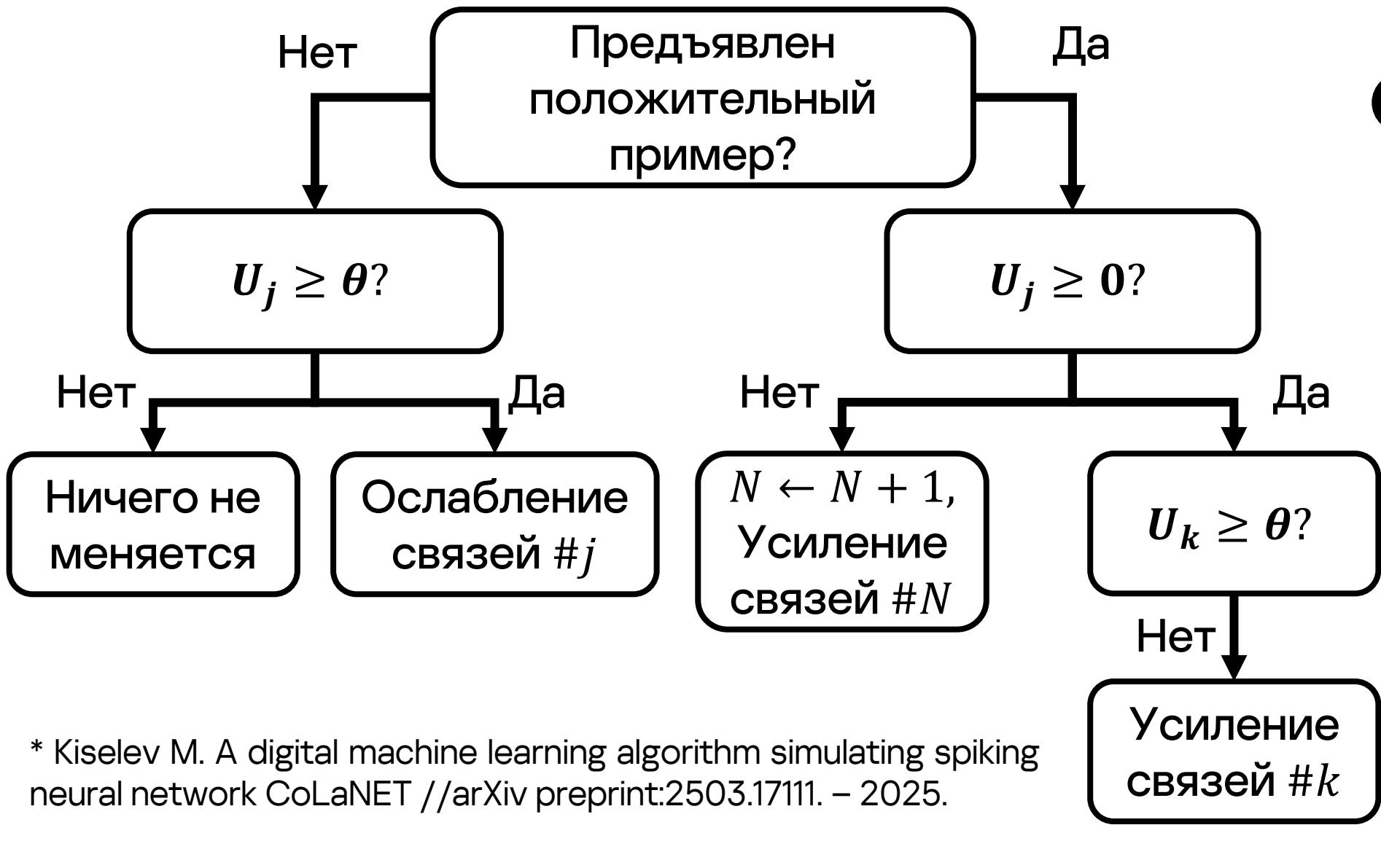


# CoLaNET

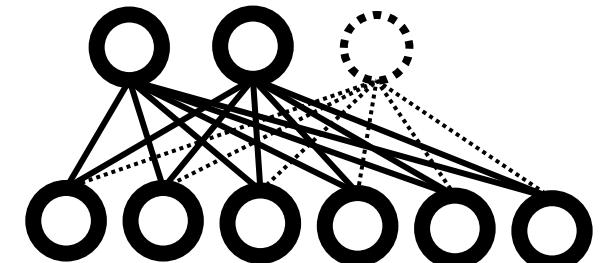
\* Kiselev M. CoLaNET - A Spiking Neural Network with Columnar Layered Architecture for Classification // arXiv preprint: 2409.01230. – 2025.



# Simplified Digital Algorithm



## Classifying Neurons



$$U_j = \sum_i w_{ji} n_i$$

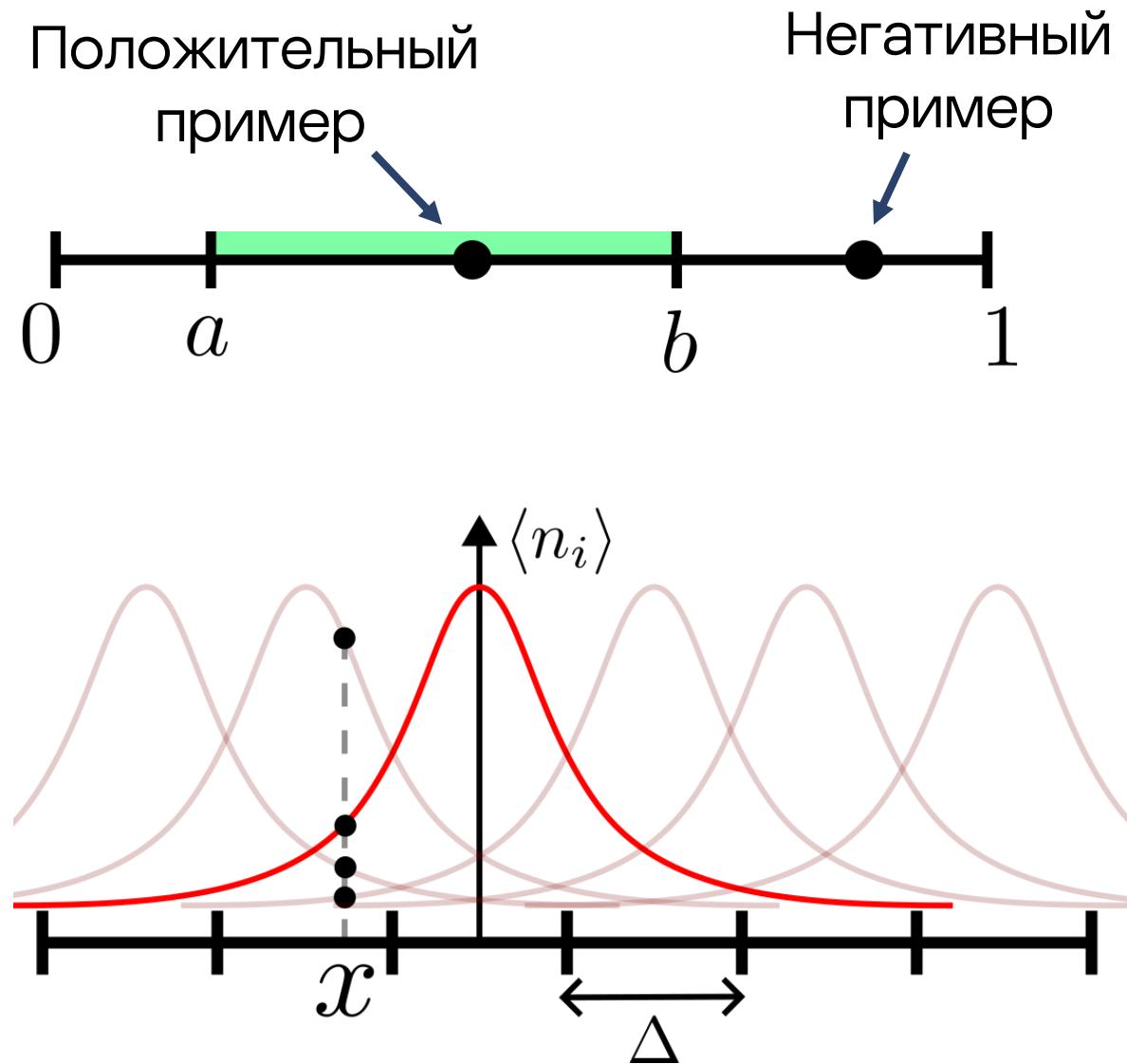
\* Kiselev M. A digital machine learning algorithm simulating spiking neural network CoLaNET //arXiv preprint:2503.17111. – 2025.

# Задача бинарной классификации

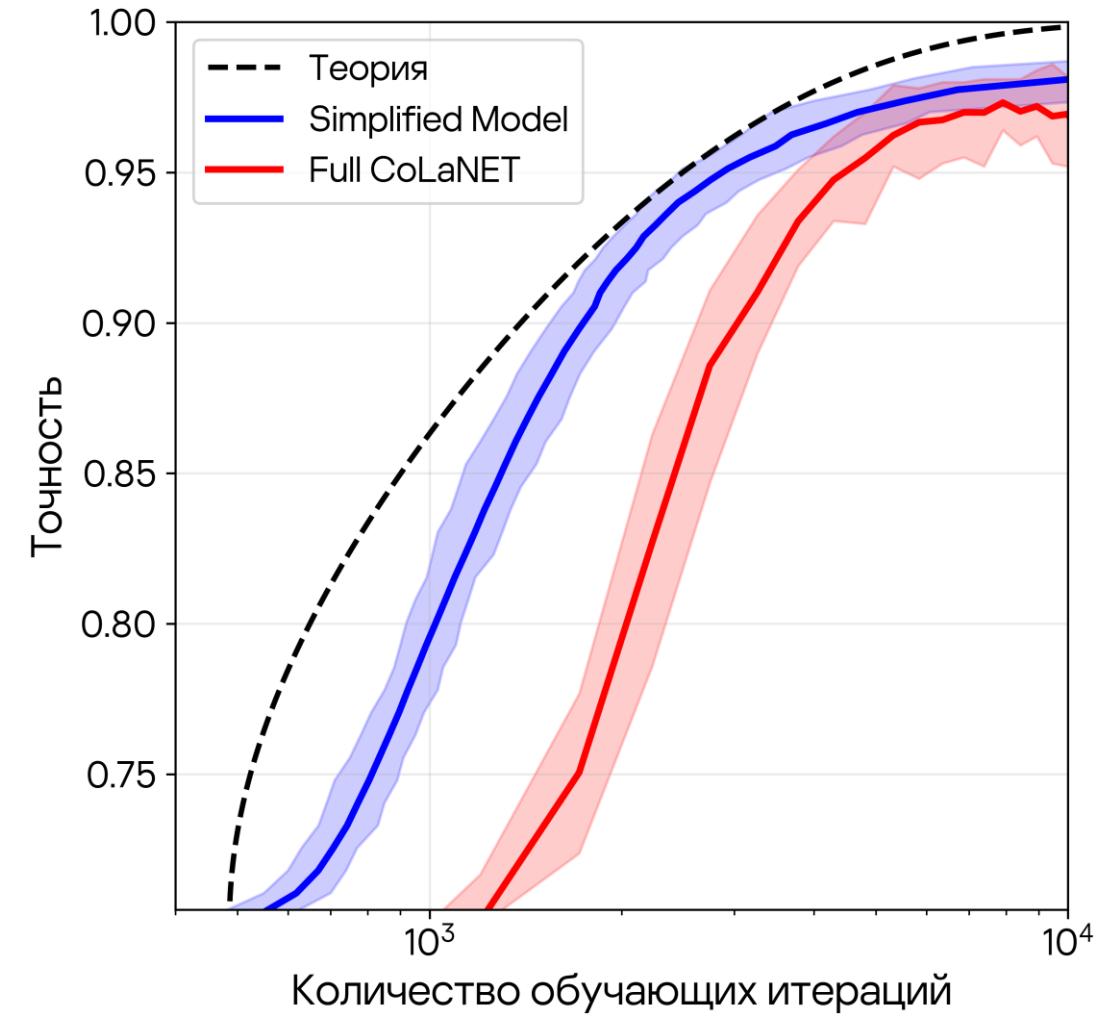
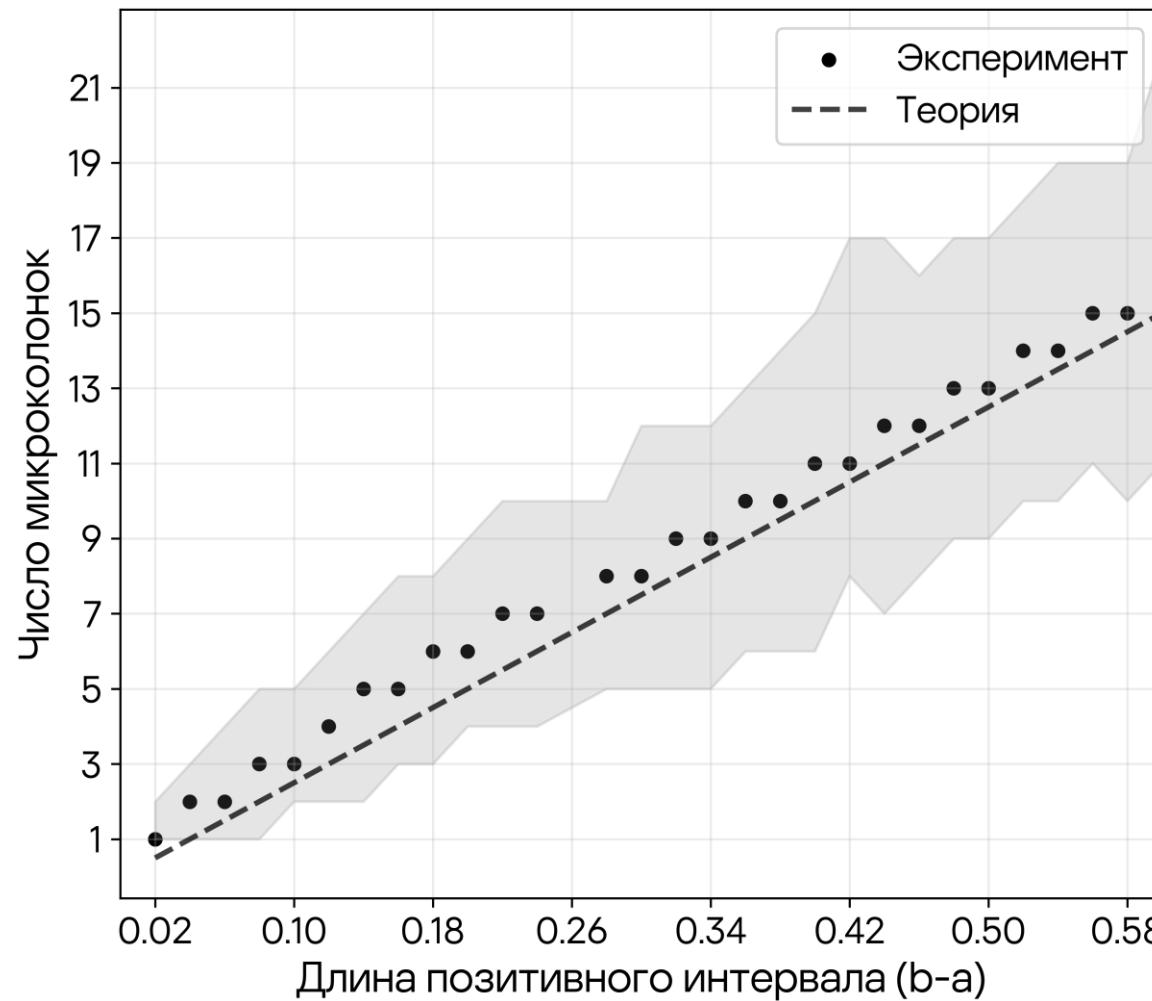
Кодирование входной активности  
в виде **Гауссовых рецептивных  
полей**:

$$f_i(x) = f_{max} \exp\left(-\frac{(x - m_i)^2}{2\sigma^2}\right)$$

**Фактическое число** импульсов  
определяется **по распределению**  
**Пуассона** с соответствующим  
средним.



# Теоретические результаты



# Зачем это нужно?

1. Понимание работы CoLaNET
2. Оптимизация гиперпараметров

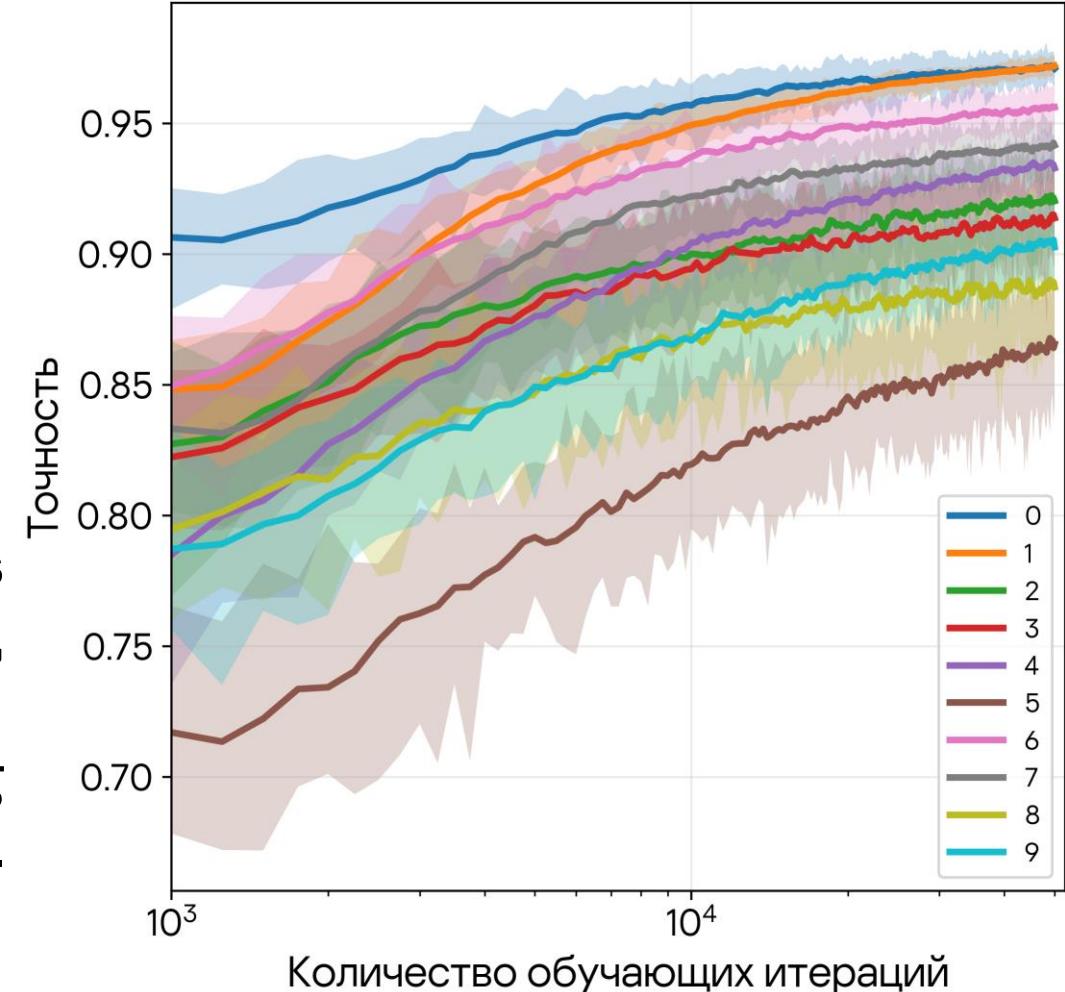
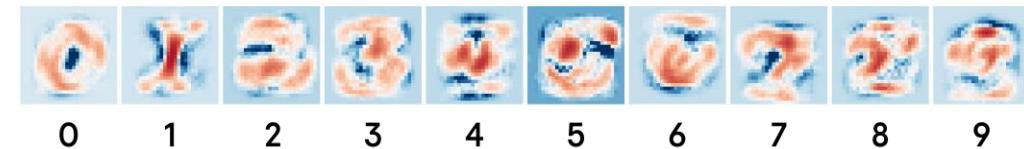
## Перспективы

1. Более сложные задачи
2. Более сложные архитектуры

## Публикации

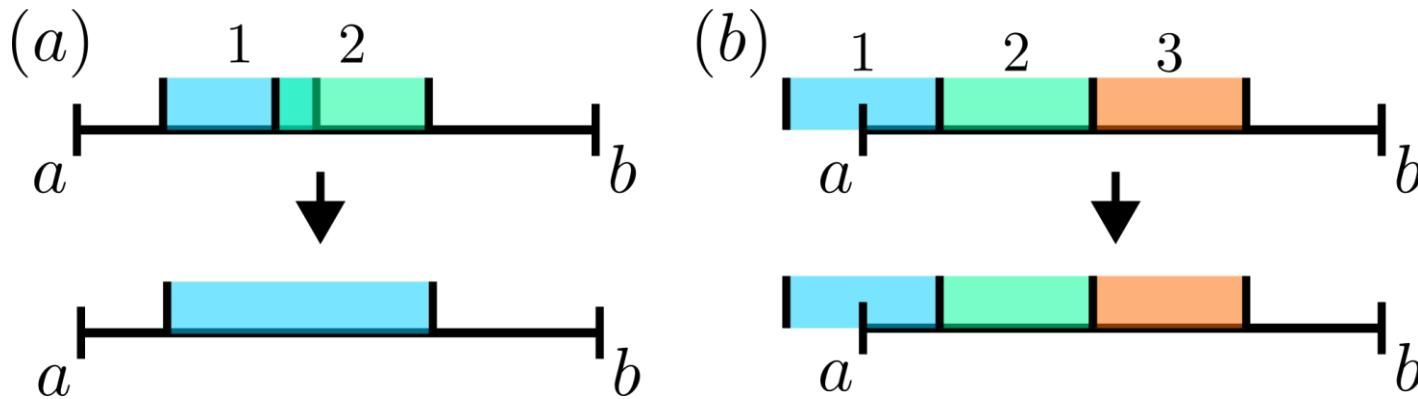
1. Goryunov et al. "Modeling the Training Dynamics of CoLaNET," *IEEE Proceedings of DCNA*, pp. 39-41, 2025.
2. Goryunov et al. "Understanding the training dynamics of CoLaNET by its simplified model". *Chaos, Solitons and Fractals* (submitted)

## MNIST



Спасибо за внимание!

# Теоретические результаты



Область активности для одного примера  $l$ :

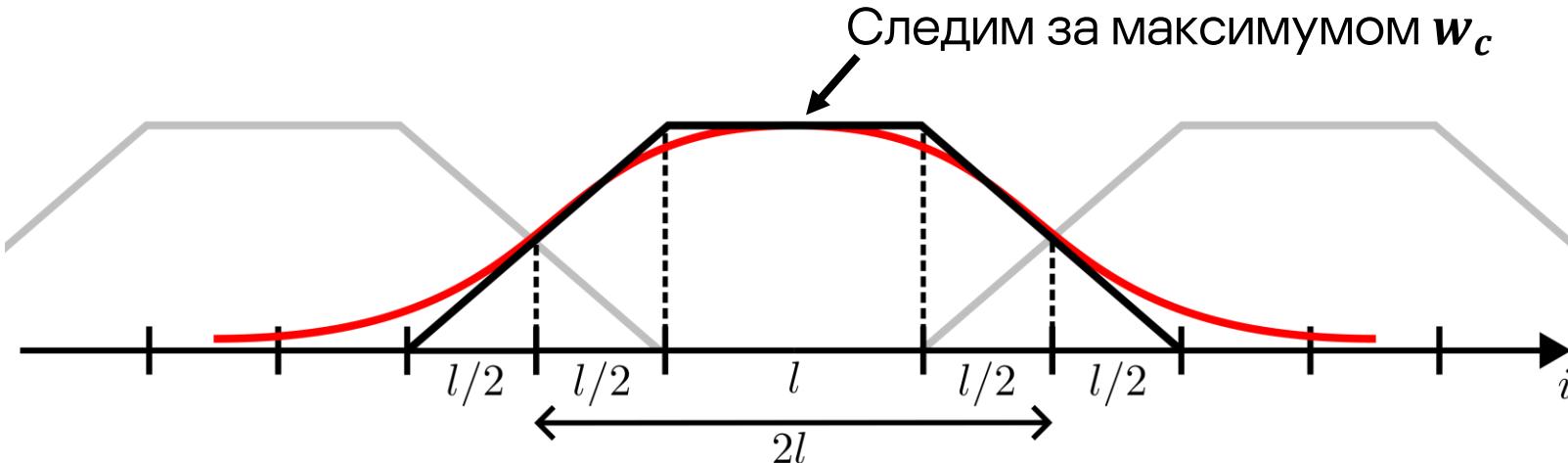
$$l = 2\sigma \sqrt{2 \ln f_{max} T_p}$$

Наиболее вероятное число микроколонок  $N$ :

$$N \approx \frac{b - a}{2l}$$

# Теоретические результаты

Апроксимируем вес в виде Гауссовой кривой



Тогда мембранный потенциал  $j$ -го нейрона:

$$U_j = T_p f_{max} n w_c \sqrt{\frac{2\pi\sigma^2\chi^2}{\sigma^2 + \chi^2}} \exp\left(-\frac{(x_k - x_j)^2}{2(\sigma^2 + \chi^2)}\right)$$

Откуда можно найти вес  $w_{t_0}$ , когда нейрон начинает реагировать:

$$w_{t_0} = \frac{\theta}{T_p f_{max} n} \sqrt{\frac{2\pi\sigma^2\chi^2}{\sigma^2 + \chi^2}} \rightarrow W_{t0} = \frac{(w_{max} - w_{min})(w_{t0} - w_{min})}{(w_{max} - w_{t0})}$$

Пока нейрон не реагирует ни на один пример, ресурс в центре изменяется по следующему ДУ:

$$\frac{dW_c}{dt} \approx \frac{dl}{b-a} \rightarrow t_0 = \frac{w_{t_0}}{d} \frac{(b-a)}{l} \frac{(w_{max} - w_{min})^2}{w_{max}(w_{max} - w_{t_0})}$$

Далее появляется область, на примеры из которой нейрон самостоятельно реагирует:

$$l_f(w_t) = 2 \sqrt{2(\sigma^2 + \chi^2) \ln \frac{w_t}{w_{t_0}}}$$

$$\frac{dW_c}{dt} \approx \frac{dl}{b-a} \left( 1 - \frac{l_f}{2l} \right)$$

Линеаризуя последнее уравнение, можно получить:

$$W(t) = W_1 + (W_{t_0} - W_1) \exp\left(-\frac{t - t_0}{t_1}\right)$$

где

$$t_1 = \frac{w_1}{d} \frac{(b - a)}{l} \frac{l^2}{\sigma^2 + \chi^2} \frac{(w_{max} - w_{min})^2}{(w_{max} - w_1)^2}$$

$$W_1 = \frac{(w_{max} - w_{min})(w_1 - w_{min})}{(w_{max} - w_1)}$$

Точность классификации можно найти по формуле:

$$Accuracy = 1 - (b - a) \left(1 - \frac{l_f}{2l}\right)$$