

# Упрощение входных данных как способ увеличения эффективности нейронных сетей

Григорьев А.Д., Коробов Н.С., Куцевол П.Н., Лукоянов А.С.  
и Жариков И.

Московский физико-технический институт

*Курс: Машинное обучение*  
(практика, В. В. Стрижов)/весна 2019

## Задача

Для данной выборки растровых изображений рукописных цифр построить модель, оптимальным образом классифицирующую изображенный символ.

## Проблема

При достаточно высоком качестве классификации, существующие решения являются относительно ресурсоемкими. Долгое время обучения и предсказания, большие объемы занимаемой памяти критичны для мобильных устройств.

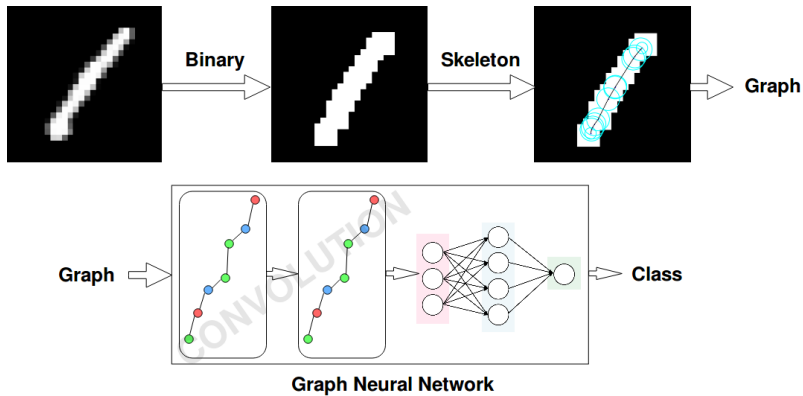
## Решения сверточными нейронными сетями

- 1 Yanai K., Tanno R., Okamoto K. Efficient mobile implementation of a cnn-based object recognition system //Proceedings of the 24th ACM international conference on Multimedia. – ACM, 2016. – С. 362-366.
- 2 Wan L. et al. Regularization of neural networks using dropconnect //International conference on machine learning. – 2013. – С. 1058-1066.

## Решения графовыми нейронными сетями

- 1 Battaglia P. W. et al. Relational inductive biases, deep learning, and graph networks //arXiv preprint arXiv:1806.01261. – 2018.
- 2 Fey M. et al. SplineCNN: Fast geometric deep learning with continuous B-spline kernels //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. – 2018. – С. 869-877.

# Предложенное решение



# Формальная постановка задачи

Дана выборка из пар: изображение  $I \in \mathbb{R}^{n \times m \times k}$ , класс изображения  $y$ .

$$D = (I_1, y_1), \dots, (I_n, y_n) \quad (1)$$

Пусть существует функция бинаризации изображения

$$B(I) : \mathbb{R}^{n \times m \times k} \rightarrow \mathbb{R}^{n \times m \times 1} \quad (2)$$

Пусть существует функция скелетонизации бинарного изображения

$$f(I_B) : \mathbb{R}^{n \times m \times 1} \rightarrow \{\mathbb{R}^p, \mathbb{R}^p, \dots, d \text{ раз.}, \mathbb{R}^p\} \quad (3)$$

А также пусть существует функция  $g$  отображающая скелет в граф  $G(E, V)$ , где каждой вершине  $v_i$  сопоставляется вектор признаков  $h_i \in \mathbb{R}^f$  и вектор координат  $x_i \in \mathbb{R}^d$ .

Тогда выборку изображений с ответами  $D$  можно отобразить в выборку графовых представлений картинок с ответами:

$$D_G = (g(f(B(I_1))), y_1), \dots, (g(f(B(I_n))), y_n) \quad (4)$$

Тогда задача будет выглядеть следующим образом:

$$\hat{w} = \arg \min_{\mathbf{w}} L(D_G, \mathbf{w}|i), \quad (5)$$

где  $L$  - функция потерь Cross Entropy Loss

$$L(D_G, \mathbf{w}|i) = - \sum_{j=1}^n y_j \log s(i(G_j(E, V)))_j \quad (6)$$

$$s(z)_j = \frac{\exp z_j}{\sum_k^C \exp z_k}, \quad (7)$$

где  $s$  - Softmax,  $i$  - нейронная сеть,  $C$  - количество классов.

## Цель эксперимента

Провести сравнение существующих подходов к классификации изображений с предложенным с точки зрения скорости обучения, скорости работы, требуемой для обучения памяти и точности классификации.

## Базы данных

- 1 MNIST Skeleton – база данных скелетных представлений картинок MNIST.
- 2 MNIST Superpixels 75 – база данных графовых представлений над super pixel, полученных из базы данных MNIST.

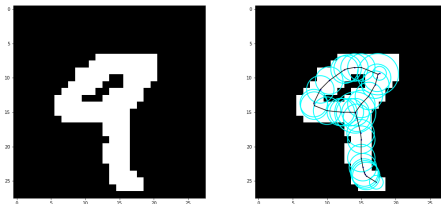


Рис.: Скелетное представление цифры 9

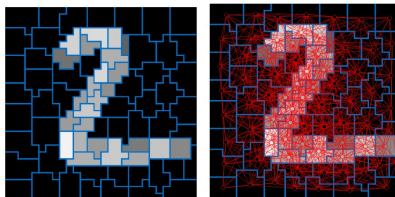


Рис.: Super pixels 75 и графовое представление над ними



# Message Passing Neural Network

Входные данные – графовые структуры, каждой из вершин сопоставляется вектор признаков  $h$ . Для каждой из вершин  $T$  раз происходит обмен информации с ее соседями с помощью функции передачи сообщения  $M$  с обновлением вектора признаков в каждой вершине с помощью функции  $U$ . Затем следует фаза вычитки информации из графа по всем вершинам –  $R$ .

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}) \quad (8)$$

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \quad (9)$$

$$\hat{y} = R(h_v^T | v \in G) \quad (10)$$

Пусть  $(G, l)$  – граф с заданной раскраской.

Для каждого слоя  $t \geq 0$   $k$ -GNN вычисляется вектор признаков  $f_k^{(t)}(s) \forall s \in [V(G)]^k$ , где  $[V(G)]^k$  – множество всех подмножеств  $V(G)$  мощности  $k$ .  $\sigma$  – функция активации.

$$f_k^{(t)}(s) = \sigma(f_k^{(t-1)}(s) \cdot W_1^{(t)} + \sum_{w \in N(v)} f_k^{(t-1)}(s) \cdot W_2^{(t)}) \quad (11)$$

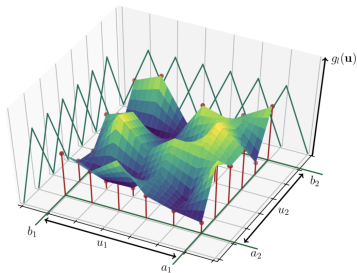
Входные данные – графовые структуры, каждой из вершин которых сопоставлена точка в  $d$ -мерном пространстве и вектор признаков. Для каждой вершины по всем ее соседям вычисляется карта весов в обобщенном пространстве координат:

$$w_{\mu, \Sigma}(u) = \exp\left(-\frac{1}{2}(u - \mu)^T \Sigma^{-1}(u - \mu)\right), \quad (12)$$

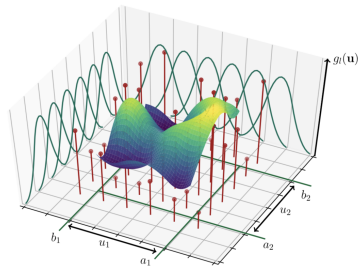
где  $\Sigma$  и  $\mu$  - обучаемые параметры. Тогда операция свертки выглядит как:

$$(f \star g) = \sum_{j=1}^J \sum_{k=1}^n g_j w_{\mu_k, \Sigma_k}(u(j, k)) f_k \quad (13)$$

Входные данные – графовые структуры, каждой из вершин которых сопоставлена точка в  $d$ -мерном пространстве и вектор признаков. Обучаемыми параметрами являются коэффициенты с которыми суммируется заранее выбранная базисная функция, определенная на пространстве координат помещенная, в узлы равномерной сетки на выбранном интервале.



(a) Linear B-spline basis functions



(b) Quadratic B-spline basis functions

# Результаты эксперимента

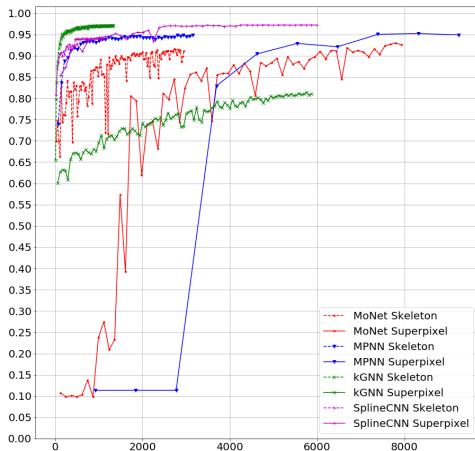


Рис.: Зависимость *асси́ра* от времени обучения в секундах

**Таблица:** Использование GPU в процессе обучения *в Гб*

	MPNN	$k$ -GNN	MoNet	SplineCNN
Skeleton	<b>1.50</b>	<b>0.34</b>	<b>0.32</b>	<b>0.73</b>
Supapixel	10.76	0.85	1.54	1.04

**Таблица:** Время предсказания на тесте с batch size 1 *в секундах*

	MPNN	$k$ -GNN	MoNet	SplineCNN
Skeleton	<b>147.2</b>	47.48	<b>56.2</b>	38.8
Supapixel	211.3	<b>26.23</b>	140.79	<b>17.9</b>

- Предложен метод увеличения эффективности нейронных сетей в задаче распознавания символов, основанный на упрощении входных данных.
- Показана эффективность предложенного метода в терминах используемой во время обучения памяти при отсутствии существенного уменьшения качества по сравнению со стандартным подходом.
- Проведено сравнение времени обучения и предсказания нейросетей, работающих со скелетами и суперпикселями соответственно.