



Национальный исследовательский  
Нижегородский государственный университет им. Н.И. Лобачевского  
Институт информационных технологий, математики и механики

Образовательный курс  
«Введение в глубокое обучение с использованием  
Intel® Neon Framework»

# Применение Intel® Neon Framework для решения задач компьютерного зрения

*При поддержке компании Intel*


Кустикова Валентина, Жильцов Максим,  
каф. МОСТ ИИТММ,  
ННГУ им. Н.И. Лобачевского

# Содержание

---

- ❑ Постановка практической задачи
- ❑ Последовательная модификация архитектур глубоких моделей, обучение и оценка качества работы этих моделей





# ПОСТАНОВКА ПРАКТИЧЕСКОЙ ЗАДАЧИ



# Задача предсказания пола человека по фотографии

- IMDB-WIKI dataset

[\[https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki\]](https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki)

IMDb



460,723 images

Wikipedia



62,328 images

- Разметка:

- окаймляющий прямоугольник для лица (задача детектирования)
- возраст изображенного человека (задача классификации)
- **пол изображенного человека (задача классификации)**
- ...

# Предварительная обработка данных

---

- Приведение изображений к одному размеру
  - Для больших изображений вырез из центра
  - Для маленьких изображений дополнение средним значением пикселей
- Вычитание среднего значения пикселей по набору из каналов изображения, деление на среднеквадратичное отклонение
- Удаление некорректных элементов из набора данных
  - Не у всех элементов набора данных присутствуют корректные метки пола



# Тренировочная и тестовая выборка

---

- ❑ Часть Wikipedia набора данных IMDB-WIKI
- ❑ Общее количество изображений – ~60 000
- ❑ Разрешение – 128×128
- ❑ Тренировочная и тестовая выборка – 2:1



# Критерий качества работы модели

- Точность классификации:

$$\frac{tp + tn}{tp + tn + fp + fn}$$

- $fp = \text{false positives}$ ,  $tp = \text{true positives}$
- $fn = \text{false negatives}$ ,  $fp = \text{false positives}$





# **МНОГОСЛОЙНЫЕ ПОЛНОСВЯЗНЫЕ НЕЙРОННЫЕ СЕТИ**





# Двухслойная сеть (FCNN-1)



# Двухслойная сеть (FCNN-1)

```
layers = [ DataTransform(transform=Normalizer(divisor=128.0)),  
          Affine(nout=16, init=Gaussian(scale=0.1),  
                activation=Tanh()),  
          Affine(nout=2, init=Gaussian(scale=0.1),  
                activation=Logistic(shortcut=True))] ]
```

```
model = Model(layers=layers)
```

```
cost = GeneralizedCost(costfunc=CrossEntropyBinary())
```

```
optimizer = GradientDescentMomentum(0.01, momentum_coef=0.9,  
                                     stochastic_round=args.rounding, wdecay=0.0005)
```

```
callbacks = Callbacks(model, eval_set=val_set,  
                      **args.callback_args)
```

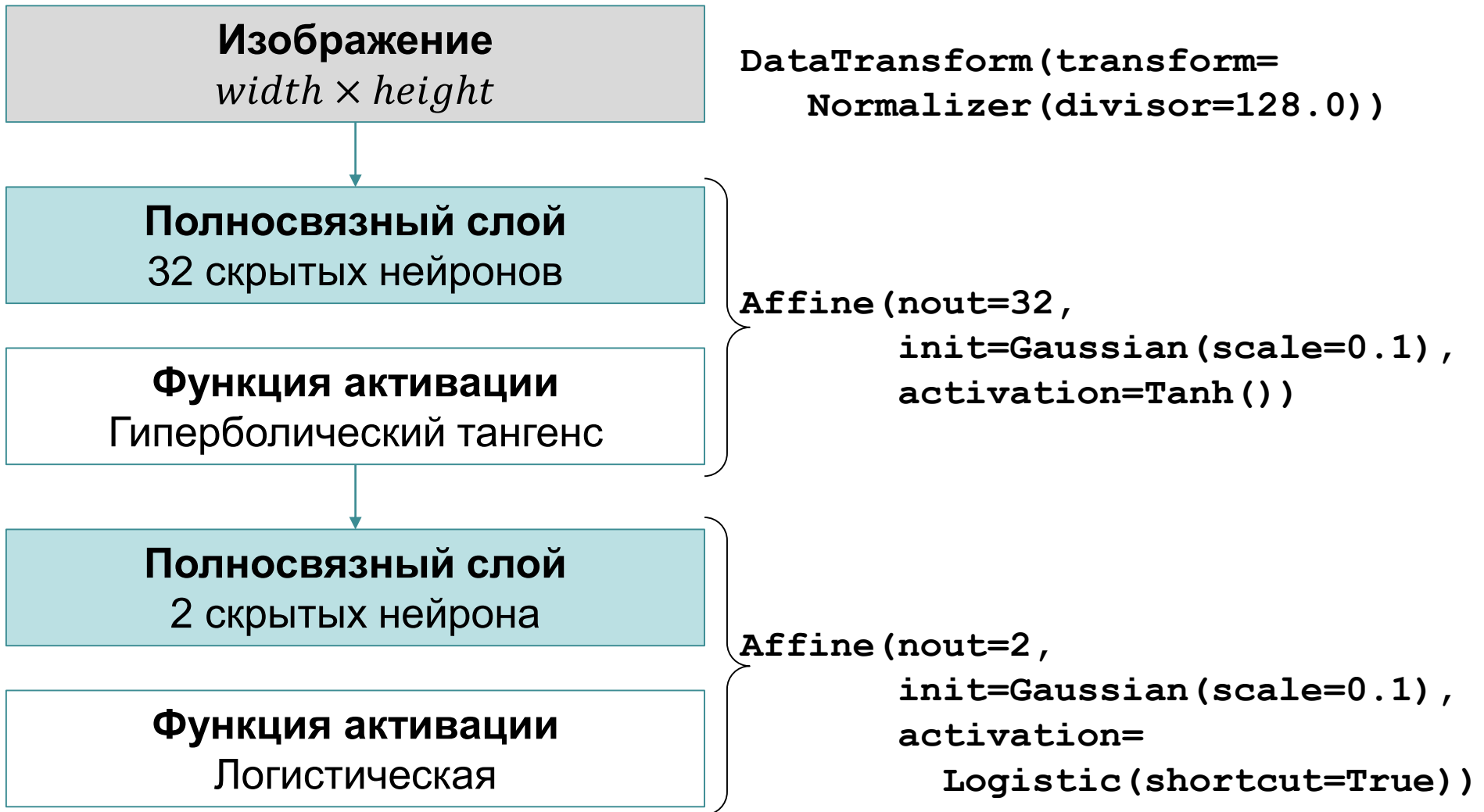
```
model.fit(train_set, optimizer=optimizer,  
          num_epochs=args.epochs, cost=cost, callbacks=callbacks)
```

# Результаты экспериментов

Сеть	Параметры обучения	Точность, %
FCNN-1	NAG-SGD, learning rate=0.01, weight decay=0.0005, batch size=100, epochs=10, momentum=0.9	<b>69.9</b>



# Двухслойная сеть (FCNN-2)

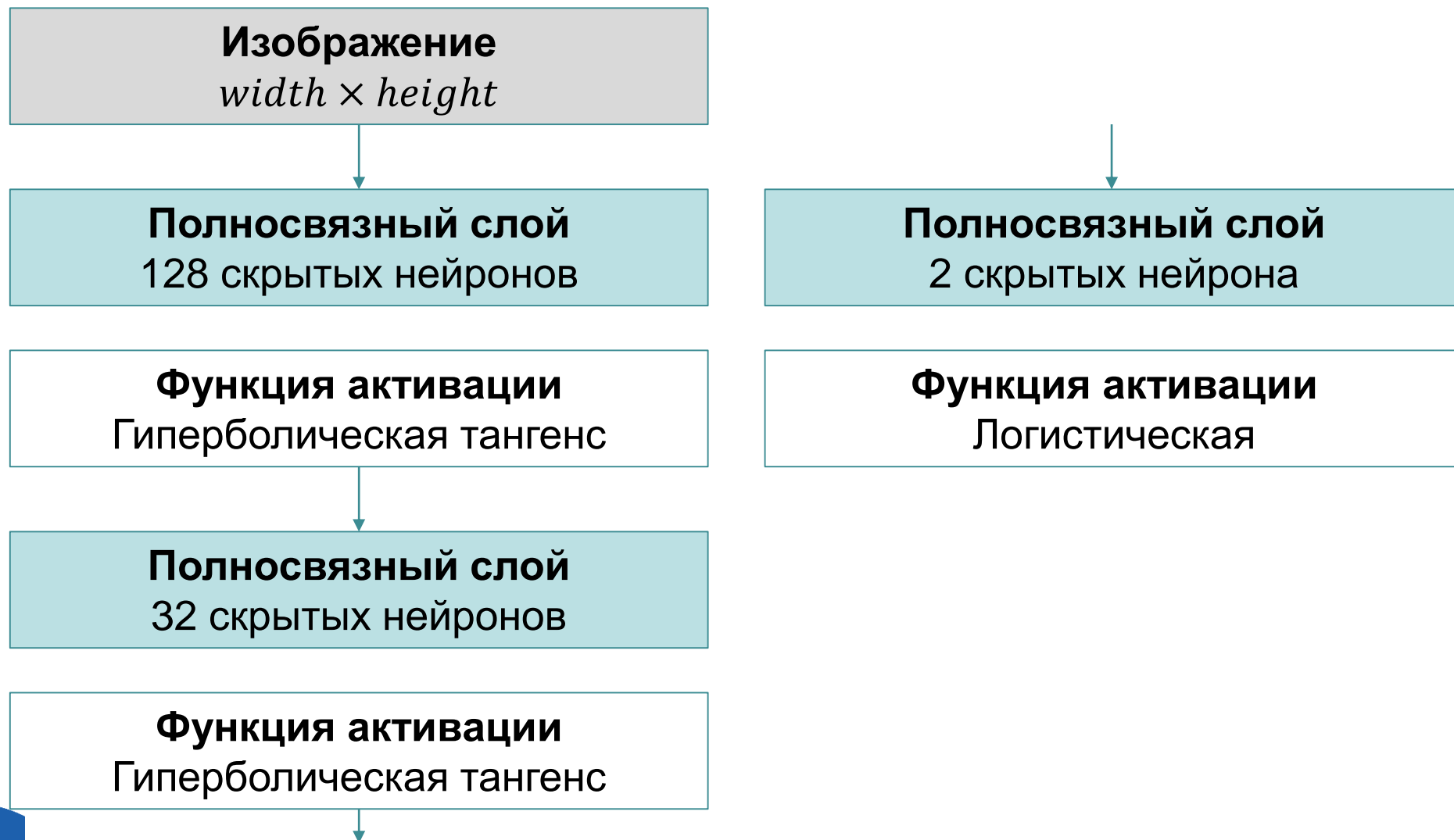


# Результаты экспериментов

Сеть	Параметры обучения	Точность, %
FCNN-1	NAG-SGD, learning rate=0.01, weight decay=0.0005, batch size=100, epochs=10, momentum=0.9	<b>69.9</b>
FCNN-2		68.1



# Трехслойная сеть (FCNN-3)



# Результаты экспериментов

Сеть	Параметры обучения	Точность, %
FCNN-1	NAG-SGD, learning rate=0.01, weight decay=0.0005, batch size=100, epochs=10, momentum=0.9	<b>69.9</b>
FCNN-2		68.1
FCNN-3		66.5



# Трехслойная сеть (FCNN-4)





# Результаты экспериментов

Сеть	Параметры обучения	Точность, %
FCNN-1	NAG-SGD, learning rate=0.01, weight decay=0.0005, batch size=100, epochs=10, momentum=0.9	69.9
FCNN-2		68.1
FCNN-3		66.5
FCNN-4		<b>75.8</b>

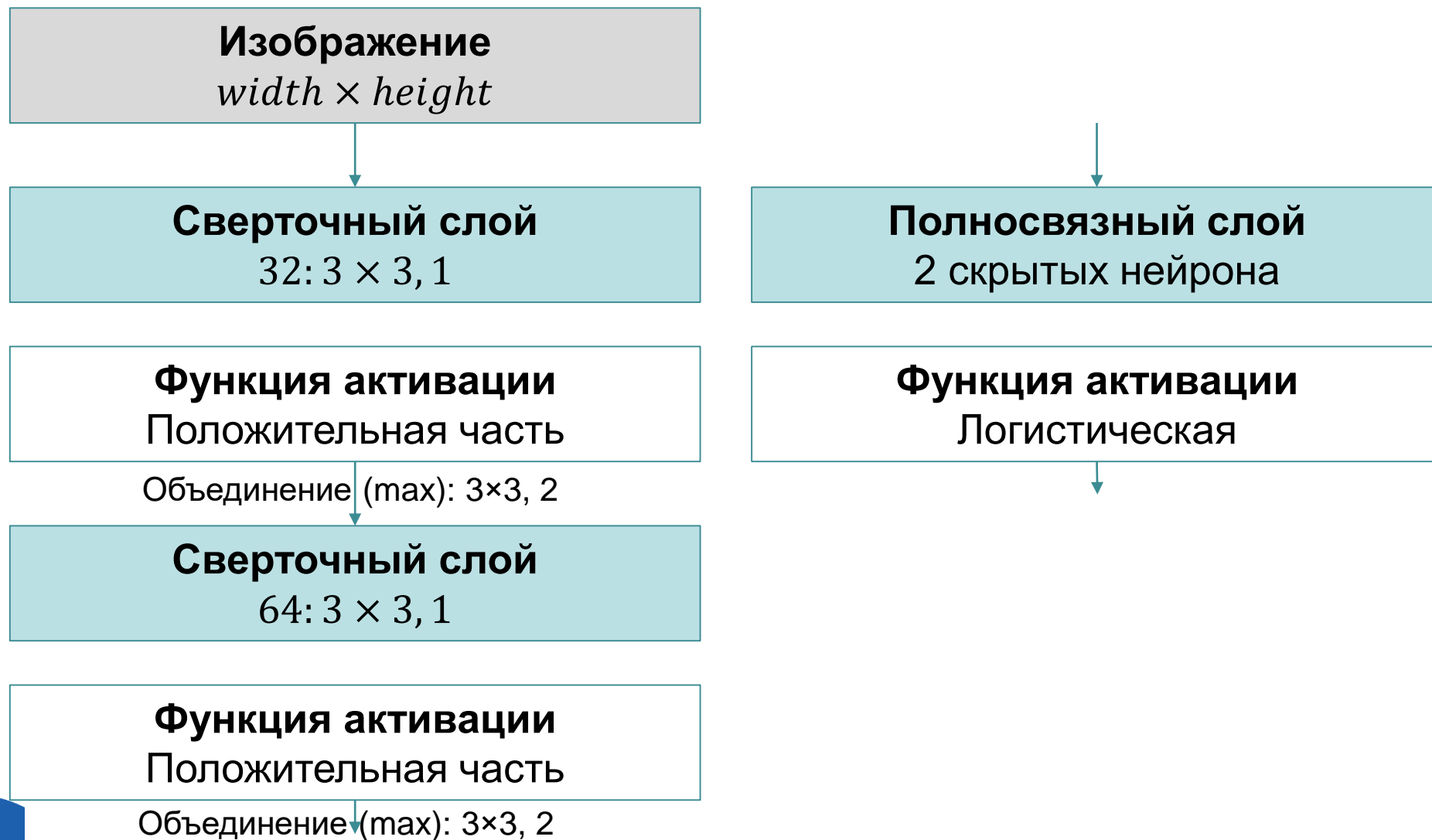


---

# СВЕРТОЧНЫЕ НЕЙРОННЫЕ СЕТИ



# Сверточная сеть (CNN-1)



# Сверточная сеть (CNN-1)

Изображение  
*width × height*

Сверточный слой  
32: 3 × 3, 1

Функция активации  
Положительная часть  
Объединение (max): 3×3, 2

```
DataTransform(transform=  
    Normalizer(divisor=128.0))
```

```
Conv(fshape=(3, 3, 32), padding=2,  
    strides=1, dilation=2,  
    init=Kaiming(),  
    activation=Rectlin()),  
Pooling(fshape=(3, 3), padding=1,  
    strides=2, op='max')
```

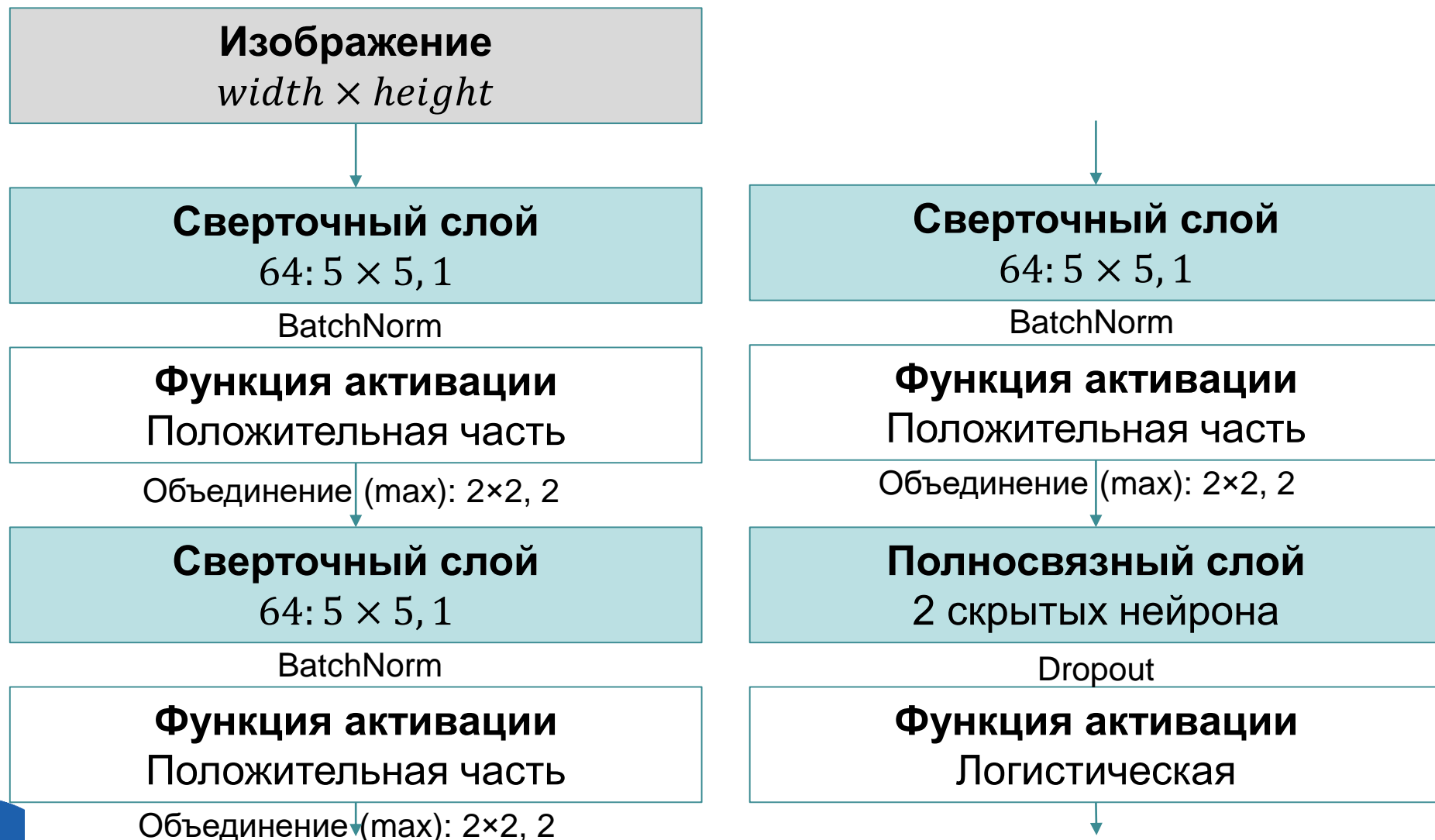


# Результаты экспериментов

Сеть	Параметры обучения	Точность, %
FCNN-1	NAG-SGD, learning rate=0.01, weight decay=0.0005, batch size=100, epochs=10, momentum=0.9	69.9
FCNN-2		68.1
FCNN-3		66.5
FCNN-4		75.8
CNN-1		<b>77.4</b>



# Сверточная сеть (CNN-2)



# Сверточная сеть (CNN-2)

## Сверточный слой

64:  $5 \times 5, 1$

BatchNorm

## Функция активации

Положительная часть

Объединение (max):  $2 \times 2, 2$

```
Conv(fshape=(5, 5, 64), padding=2,  
     strides=1, dilation=1,  
     init=Xavier(), bias=Constant(0)),  
BatchNorm(),  
Activation(Rectlin()),  
Pooling(fshape=(2, 2), padding=0,  
        strides=2, op='max')
```



# Результаты экспериментов

Сеть	Параметры обучения	Точность, %
FCNN-1	NAG-SGD, learning rate=0.01, weight decay=0.0005, batch size=100, epochs=10, momentum=0.9	69.9
FCNN-2		68.1
FCNN-3		66.5
FCNN-4		75.8
CNN-1		77.4
CNN-2		<b>79.0</b>





# А что дальше?

---

- ❑ Модификация архитектур глубоких моделей
- ❑ Модификация и перебор варьируемых параметров
  
- ❑ Анализ данных
  - Удаление данных, создающих «шум» в общем наборе тренировочных данных
  - Коррекция разметки
  - Изменение стратегии предварительной обработки данных
  
- ❑ Обзор существующих решений аналогичных задач
  - Использование «переноса обучения» (transfer learning)
  - Модификация широко известных моделей



# Заключение

---

- Общая схема решения задачи:
  - Сформулировать задачу (вход, выход)
  - Подготовить данные (разметка, предварительная обработка)
  - Определить критерий качества работы алгоритма
  - Посмотреть существующие методы решения поставленной задачи или близкой к поставленной
  - По возможности отталкиваться от существующих глубоких моделей
  - Разработка описаний моделей и проведение массовых экспериментов



# Основная литература

---

- ❑ Хайкин С. Нейронные сети. Полный курс. – М.: Издательский дом «Вильямс». – 2006. – 1104 с.
- ❑ Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика. – 2002. – 344 с.
- ❑ Goodfellow I., Bengio Y., Courville A. Deep Learning. – MIT Press. – 2016. – [<http://www.deeplearningbook.org>].



# Авторский коллектив

---

- ❑ **Кустикова Валентина Дмитриевна**  
к.т.н., ст.преп. каф. МОСТ ИИТММ,  
ННГУ им. Н.И. Лобачевского  
[valentina.kustikova@itmm.unn.ru](mailto:valentina.kustikova@itmm.unn.ru)
- ❑ **Золотых Николай Юрьевич**  
д.ф.-м.н., проф. каф. АГДМ ИИТММ,  
ННГУ им. Н.И. Лобачевского  
[Nikolai.Zolotykh@gmail.com](mailto:Nikolai.Zolotykh@gmail.com)
- ❑ **Жильцов Максим Сергеевич**  
магистрант каф. МОСТ ИИТММ,  
ННГУ им. Н.И. Лобачевского  
[zhiltsov.max35@gmail.com](mailto:zhiltsov.max35@gmail.com)

