

Нижегородский Государственный Технический Университет  
им. Р.Е. Алексеева

Выпускная квалификационная работа

**Тема: Программная система диагностики  
состояния механизма по вибрационному сигналу**

Выполнил: студент группы 16-В-2

Гусев Никита Алексеевич

Научный руководитель: к.т.н., доцент

Гай Василий Евгеньевич

Нижний Новгород  
2020

# Цель и задачи работы

**Цель работы:** разработка системы диагностики состояния механизма по виброционному сигналу

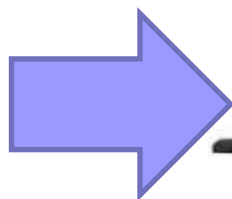
**Задачи работы:**

- применение модели одномерной свёрточной сети для вибродиагностики состояния механизмов;
- выполнение вычислительного эксперимента с целью проверки работоспособности разработанного алгоритма
- разработка контрольно-проверочного стенда для сопряжения с нейронной сетью

# Актуальность работы



Экспертный подход



Современный подход



# **Объект и предмет исследования**

**Объект исследования:** модель классификации состояний на основе одномерной свёрточной нейронной сети

**Предмет исследования:** применимость модели для разработки системы диагностики состояния механизма по виброционному сигналу

# Структура разрабатываемой системы

**Система включает в себя:**

- подсистему обучения нейронной сети;
- подсистему оценки состояния механизма посредством обученной нейронной сети

# Выбор средств разработки

Выбор языка программирования:

- **Python: TensorFlow, Numpy, Keras, Scikit-learn**

Выбор среды разработки для языка Python:

- **Spyder** задействован для реализации подсистемы оценки состояния механизма, оснащённой графической оболочкой для работы пользователя
- **Google Colaboratory** использовался для разработки подсистемы обучения нейронной сети и самого тестирования нейронной сети в ходе эксперимента

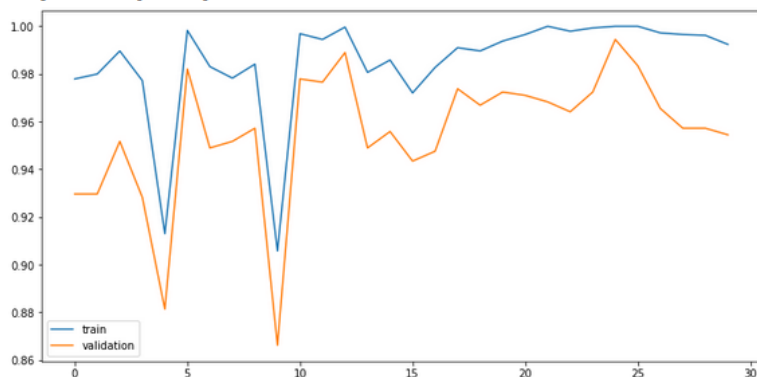
# Особенности Google Colaboratory

```
[ ] working_dir = Path('/content/drive/My Drive/')
DATA_PATH = Path("/content/drive/My Drive/Data")
save_model_path = working_dir / 'Model'
DE_path = DATA_PATH / '12k_DE'
```

```
for path in [DATA_PATH, save_model_path]:
    if not path.exists():
        path.mkdir(parents=True)
```

```
[ ] ##### HYPERPARAMETERS #####
bs = 64
lr = 0.001
wd = 1e-5
betas=(0.99, 0.999)
device = torch.device("cuda") if torch.cuda.is_available()
random_seed = 42
```

```
[ ] <matplotlib.legend.Legend at 0x7f5554f05908>
```



```
[ ] torch.save(model.state_dict(), save_model_path / 'model.pth')
```

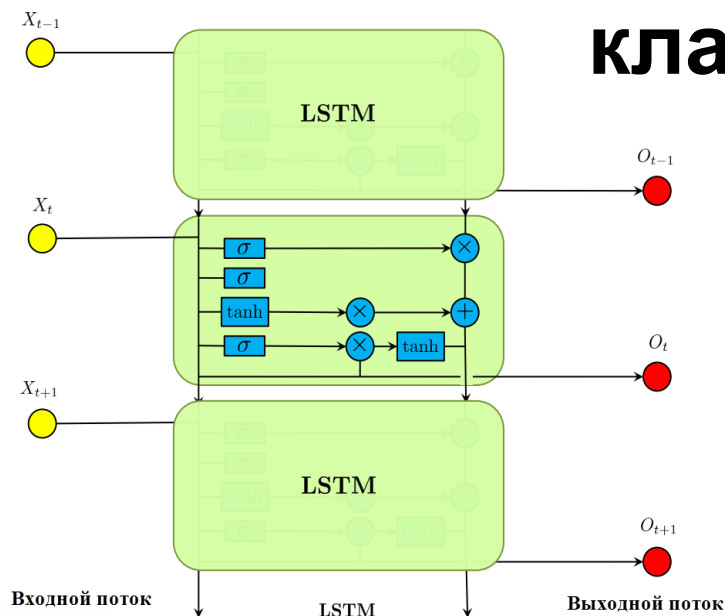
```
[ ] # Checking the saved model, if it works or not.
model2 = nn_model.CNN_1D_2L(len(features))
```

```
[ ] model2.load_state_dict(torch.load(save_model_path / 'model.pth'))
model2.eval()
```

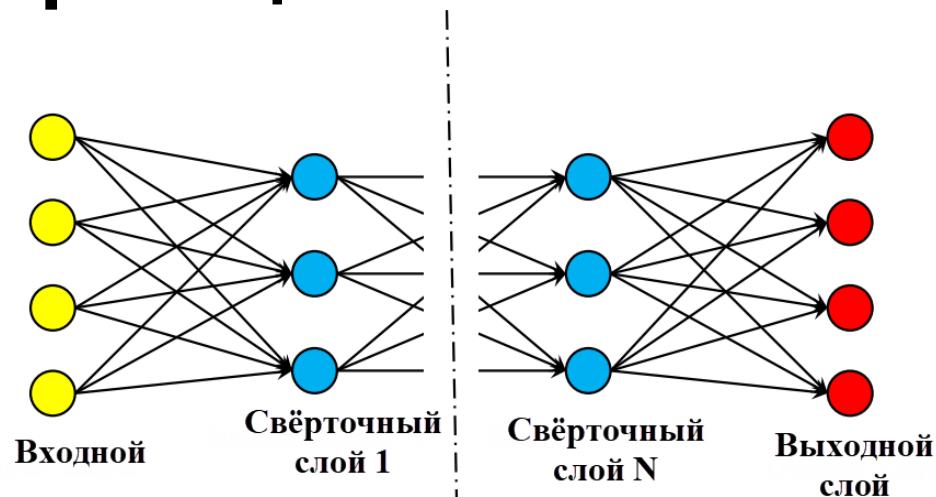
```
## Train
epochs = 30
model, metrics = fit(epochs, model, loss_func, opt, train_dl, val_dl)
```

EPOCH	Train Loss	Val Loss	Train Acc	Val Acc
0	0.11307	0.65193	0.97791	0.92966
1	0.12938	0.57708	0.97998	0.92966
2	0.03989	0.32134	0.98964	0.95172
3	0.11520	0.68035	0.97722	0.92828
4	0.78238	1.55256	0.91301	0.88138
5	0.00452	0.11253	0.99827	0.98207
6	0.08034	0.44734	0.98309	0.94897
7	0.13069	0.48065	0.97825	0.95172
8	0.09219	0.42594	0.98412	0.95724
9	0.80148	1.74862	0.90576	0.86621
10	0.00992	0.17134	0.99689	0.97793
11	0.02524	0.25879	0.99448	0.97655
12	0.00185	0.09400	0.99965	0.98897
13	0.08145	0.45166	0.98067	0.94897
14	0.08229	0.42859	0.98585	0.95586
15	0.19932	0.69337	0.97204	0.94345
16	0.19849	0.65429	0.98274	0.94759
17	0.11621	0.36588	0.99103	0.97379
18	0.12844	0.41497	0.98964	0.96690
19	0.07164	0.30259	0.99379	0.97241
20	0.02001	0.22377	0.99655	0.97103
21	0.00117	0.20849	1.00000	0.96828
22	0.00490	0.29586	0.99793	0.96414
23	0.00188	0.23267	0.99931	0.97241
24	0.00000	0.03840	1.00000	0.99448
25	0.00047	0.10282	1.00000	0.98345
26	0.00380	0.18238	0.99724	0.96552
27	0.00898	0.26972	0.99655	0.95724
28	0.00811	0.24056	0.99620	0.95724
29	0.02737	0.33180	0.99241	0.95448

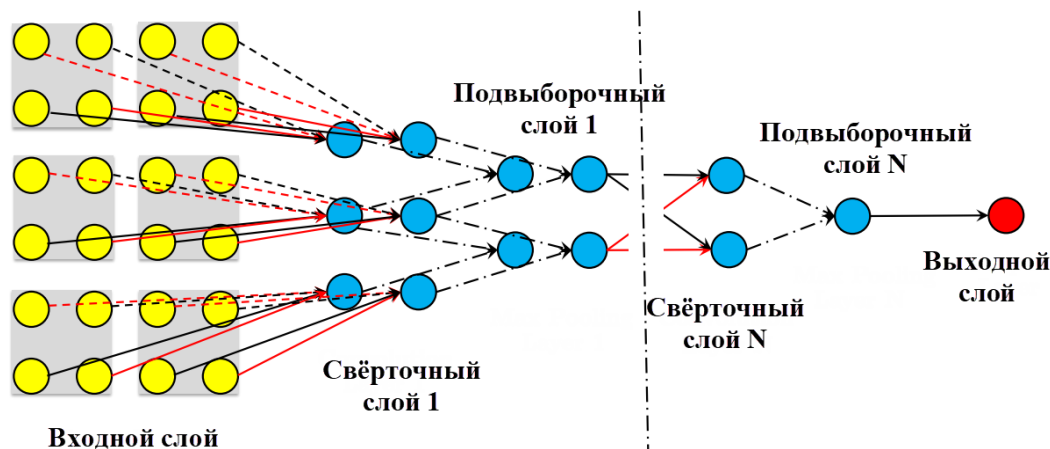
# Обзор нейросетевых методов классификации



Рекуррентные нейронные сети (RNN)



Авто-кодировщики



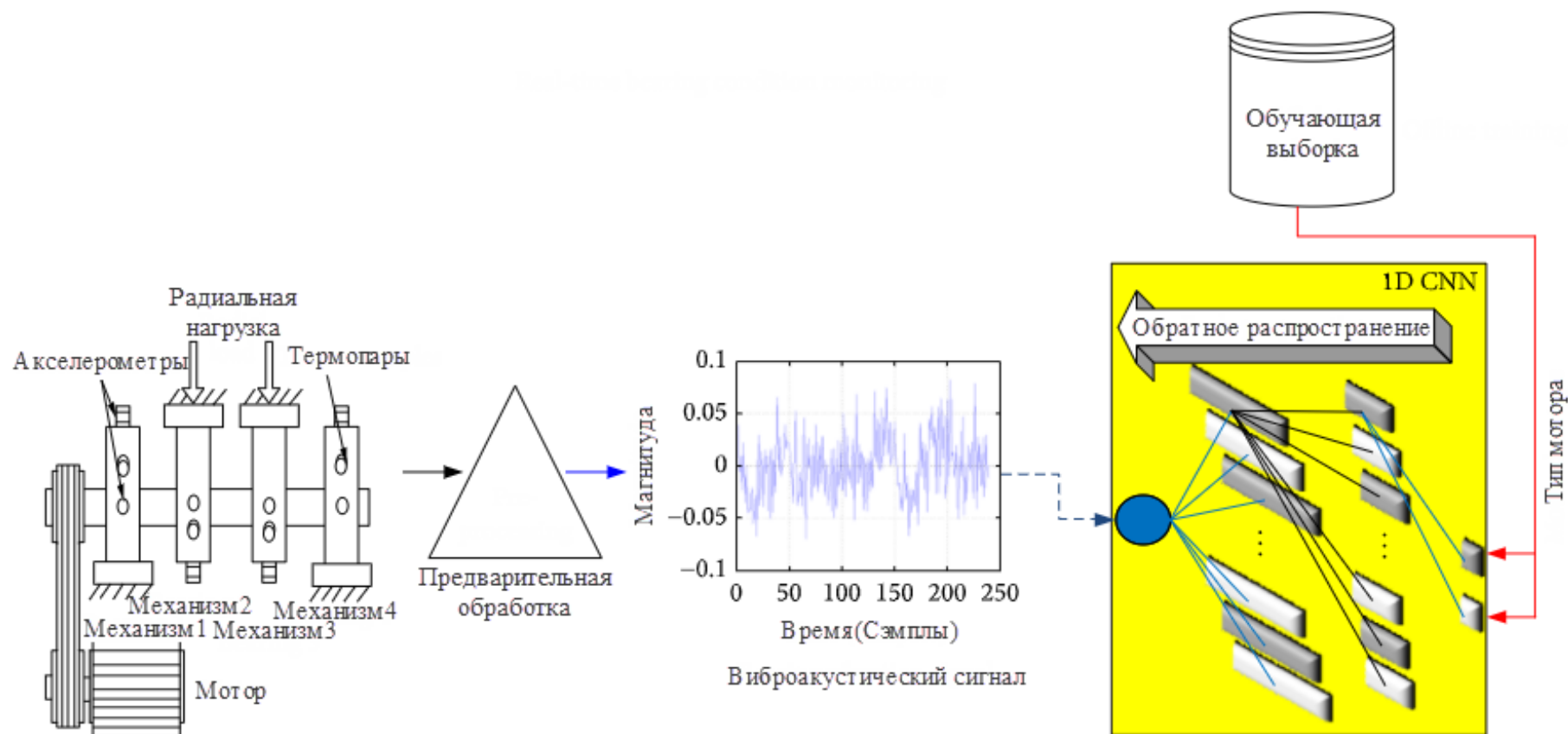
Сверточные нейронные сети (CNN)



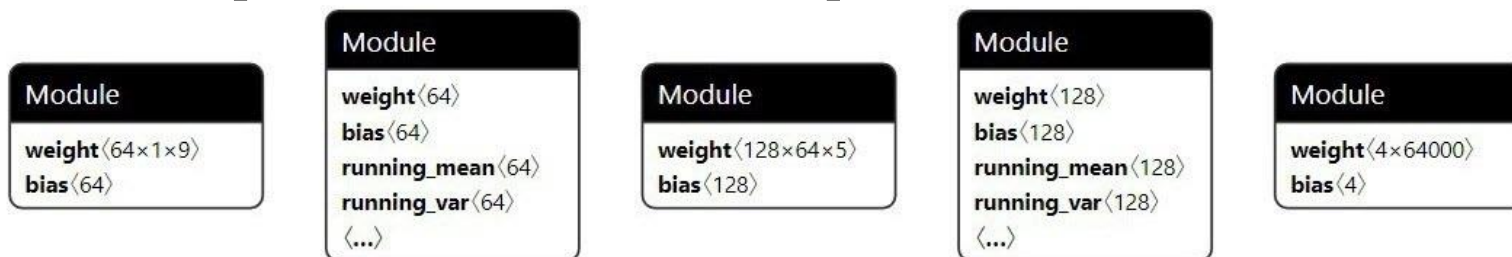
# Обоснование выбора CNN

- Обученная сеть максимально приспособлена для решения задачи выявления признаков и классификации состояний многомерных объектов
- Данный тип нейронной сети с высокой точностью выявляет периодические закономерности в ходе анализа акустических сигналов, что позволит внедрить в систему модуль, осуществляющий предиктивный анализ сигналов с целью обеспечения возможности ещё на ранней стадии предупреждать возможные неисправности;
- CNN нейронные сети обучаются на необработанных данных, что обеспечивает высокую стабильность и точность классификации.

# Схема работы нейронной сети



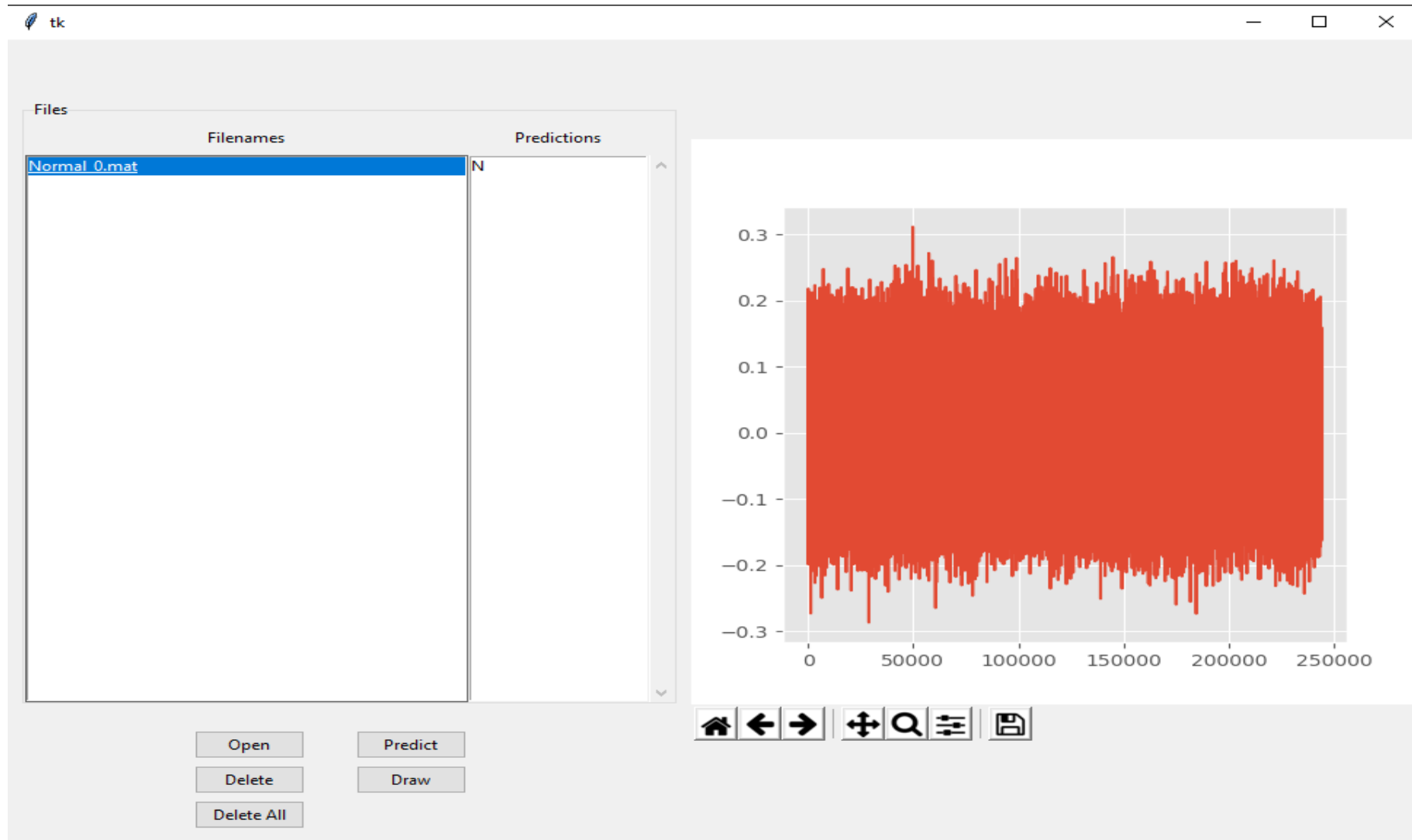
# Структурная модель разрабатываемой одномерной свёрточной нейронной сети



```
model2.load_state_dict(torch.load(save_model_path / 'model.pth'))  
model2.eval()
```

```
CNN_1D_2L(  
  (layer1): Sequential(  
    (0): Conv1d(1, 64, kernel_size=(9,), stride=(1,), padding=(4,))  
    (1): BatchNorm1d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (2): ReLU()  
    (3): Dropout(p=0.5, inplace=False)  
    (4): MaxPool1d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
  )  
  (layer2): Sequential(  
    (0): Conv1d(64, 128, kernel_size=(5,), stride=(1,), padding=(2,))  
    (1): BatchNorm1d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)  
    (2): ReLU()  
    (3): Dropout(p=0.5, inplace=False)  
    (4): AvgPool1d(kernel_size=(2,), stride=(2,), padding=(0,))  
  )  
  (linear1): Linear(in_features=64000, out_features=4, bias=True)  
)
```

# Программная реализация



# Программная реализация

Filenames	Predictions
B021_1.mat	F
B007_2.mat	F
B007_0.mat	F
IR021_3.mat	F
Normal_1.mat	N
Normal_1.mat	N
Normal_3.mat	N
Normal_0.mat	N

# Тестирование нейронной сети

**Точность работы нейронной сети:**

$$Acc = \frac{N - E}{N} * 100\%$$

где  $N$  – мощность выборки, а  $E$  – количество ошибок классификации

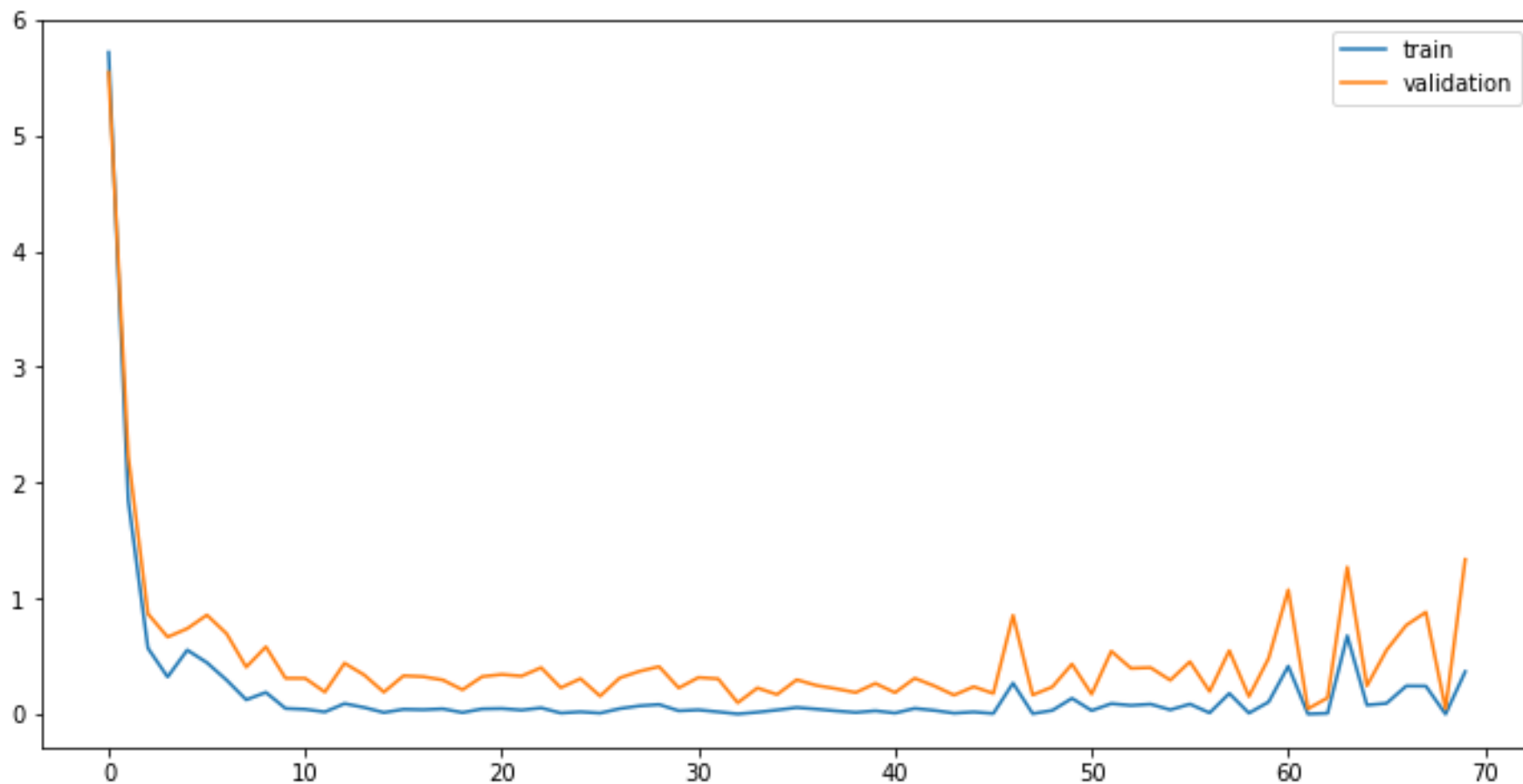
**Количество потерь:**

$$E_p = CKO[t_i^p, (y_1^L, \dots, y_{N_L}^L)] = \sum_{i=1}^{N_L} (y_i^L - t_i^p)^2$$

где  $N_L$  – мощность эталонной базы (количество классов в обучающей выборке),  $p$  – номер эталона,  $t_i^p$  и  $(y_1^L, \dots, y_{N_L}^L)$  – вектор  $p$ -го эталона и выходной вектор соответственно

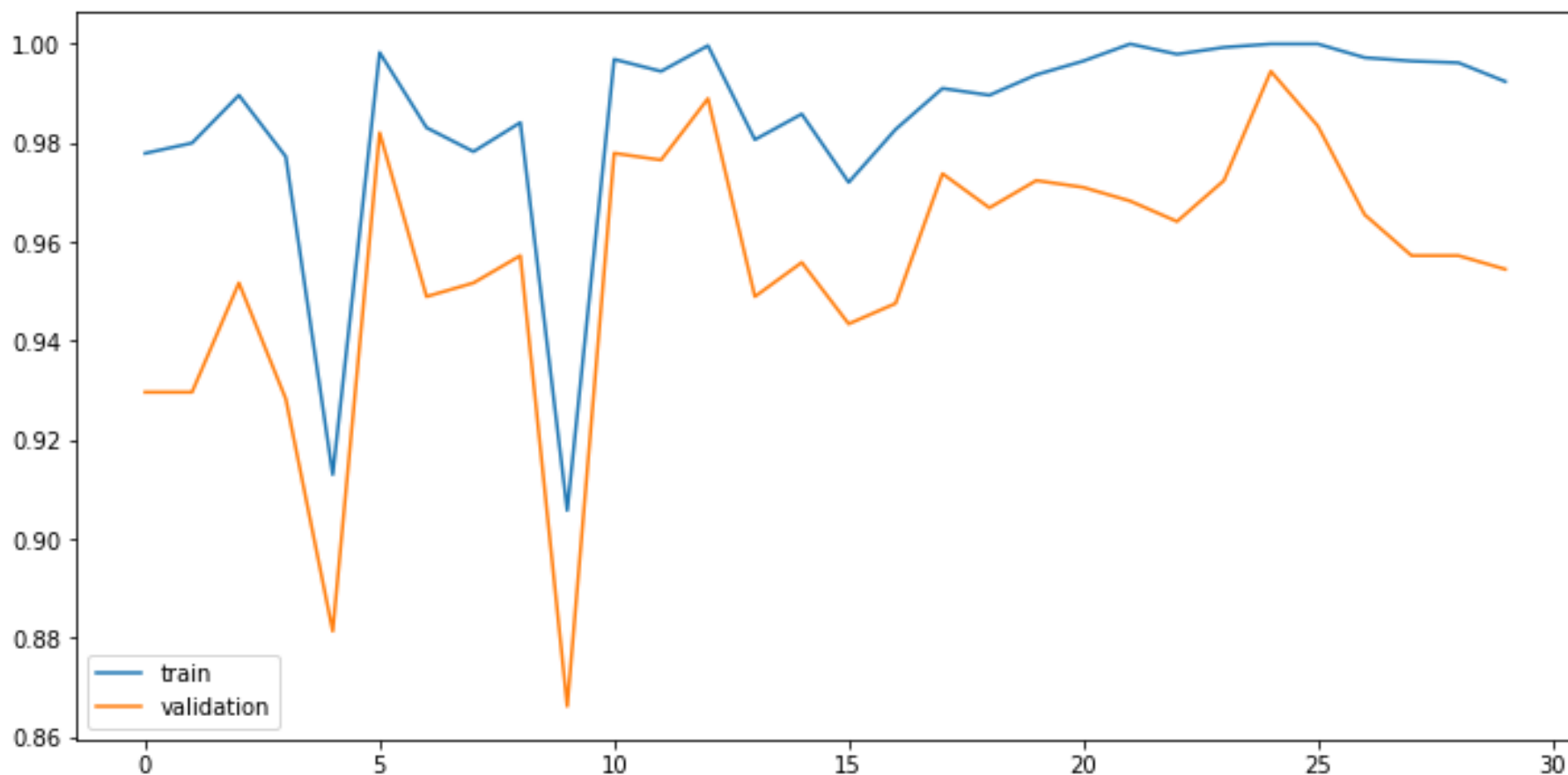
# Результаты тестирования

## График потерь



# Результаты тестирования

## График точности





# Стенд для диагностирования



# Выводы по результатам исследований

- Была спроектирована и программно реализована система для обнаружения дефекта механизма по виброакустическому сигналу.
- Программная система предоставляет высокоэффективную и многофункциональную разработку алгоритма для определения неисправностей различных механизмов по виброакустическому сигналу.
- В ходе тестирования была доказана эффективная и функциональная работоспособность, возможность применения данной системы для сформулированной задачи.
- Разработанная система имеет высокую практическую ценность, так как решает актуальную и общую задачу классификации виброакустических сигналов.
- Данная система применима с различным набором данных и легко интегрируется для определения неисправностей у множества механизмов.

# Перспективы дальнейшего развития

- Интегрирование системы в узкую область анализа, то есть разработка интерфейса для определения дефектов подшипников качения с применением мультиклассификации, а именно обнаружения неисправностей внутри и снаружи детали.
- Анализ более эвристических подходов к построению нейронной сети и выбор наиболее подходящего варианта для заданной области.
- Изменение исходного кода с целью оптимизации алгоритма и улучшения производительности системы в целом.

# Публикации по теме выпускной квалификационной работы

- Гусев, Н.А. Программная система диагностики состояния механизма по вибрационному сигналу / Н.А. Гусев // «Информационные системы и технологии (ИСТ-2020)»: материалы XXVI международной научно-технической конференции, Н. Новгород, 24 апреля 2020 г. – Н.Новгород: НГТУ. – 2020.



**Спасибо за внимание!**