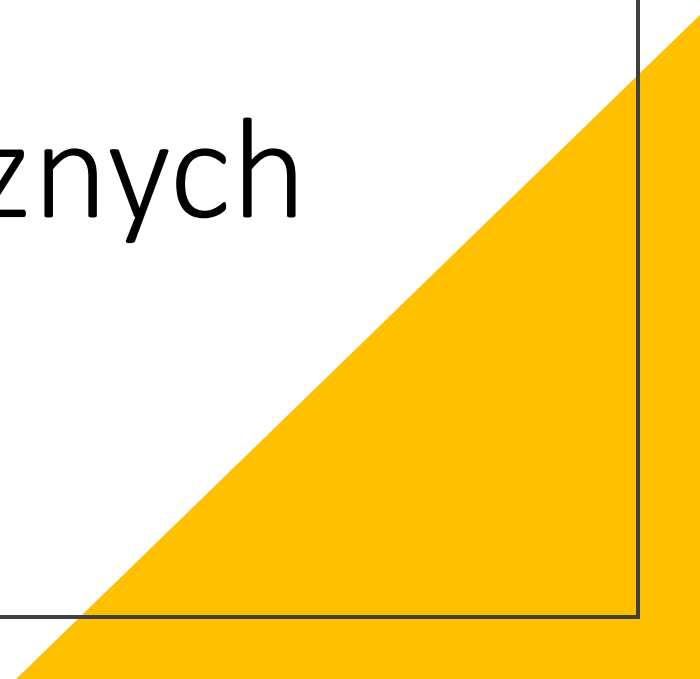


# Klasyfikacja enzymów na podstawie sekwencji i właściwości fizykochemicznych

Szymon Szrajer



Grupa (klasa)	Katalizator reakcji
<b>EC 1</b> <i>Oksydoreduktazy</i>	przenoszą ładunki (elektrony i jony $\text{H}_3\text{O}^+$ - protony) z cząsteczki substratu na cząsteczkę akceptora
<b>EC 2</b> <i>Transferazy</i>	przenoszą daną grupę funkcyjną (tiolową, aminową, itp.) z cząsteczki jednej substancji na cząsteczkę innej substancji
<b>EC 3</b> <i>Hydrolazy</i>	katalizują rozpad substratu pod wpływem wody (hydroliza); do grupy tej należy wiele enzymów trawiennych
<b>EC 4</b> <i>Liazy</i>	katalizują rozpad substratu bez hydrolizy
<b>EC 5</b> <i>Izomerazy</i>	zmieniają wzajemne położenie grup chemicznych bez rozkładu szkieletu związku
<b>EC 6</b> <i>Ligazy</i>	katalizują syntezę różnych cząsteczek; powstają wiązania chemiczne
<b>EC 7</b> <i>Translokazy</i>	katalizują ruch jonów i cząsteczek przez błony lub ich rozdział wewnątrz błon

# Dataset 1

	Enzyme classification [EC]	Enzyme classification [EC]		1	×	Remove	
AND	Taxonomy [OC]	Taxonomy [OC]		Vertebrata (vertebrates) [7742]		×	Remove
AND	Sequence length	From	To	600	1000		Remove

1\_2000.fa

2\_2000.fa

3\_2000.fa

4\_2000.fa

5\_2000.fa

6\_2000.fa


7\_2000.fa


non-enzyme\_14000.fa


# Dataset 2


- [EC 6.1](#) includes ligases used to form carbon-oxygen bonds
- [EC 6.2](#) includes ligases used to form carbon-sulfur bonds
- [EC 6.3](#) includes ligases used to form carbon-nitrogen bonds (including [argininosuccinate synthetase](#))
- [EC 6.4](#) includes ligases used to form carbon-carbon bonds
- [EC 6.5](#) includes ligases used to form [phosphoric ester](#) bonds


	Enzyme classification [EC]	6.5	Remove
AND	Taxonomy [OC]	7742	Remove
AND	Sequence length	From 600 To 1000	Remove


 non-enzyme\_5000.fa

 1\_6\_1000.fa

 2\_6\_1000.fa

 3\_6\_1000.fa

 4\_6\_1000.fa

 5\_6\_1000.fa

# Dane - Sekwencja

```
>tr|A0A671G7H5|A0A671G7H5_RHIFE tr|A0A671G7H5|A0A671G7H5_RHIFE Amine oxidase OS=Rhinolophus ferrumequinum OX=59479 GN=A0C1 PE=3 SV=1
MEQRWQLHGSPAAPGRRGGASEEAASVGKPRGHGTWQSQLGGNPCVPIKQLTSPPLHSRAMGRETLALGWAVAATLMLQALAMAEHSPGTPHASKASVFADLSAHELKAVRSFLWSRKELRLQSSRALITKNSVFLI
LNHALQEATKPLHQFFLATTGFSFHNCHLQCLTFTDVAPRGLASGERRSWFILQRYVEGYFLHPTGLELLLDHSSTNTQDWTVEQVWYNGKFYRSPEELARKYKKGEVDVVVLEDPLPKGKGAENMKDPPLFSSYK
SQTKEYIDVGWGLGTVTHELAPGIDCPNTATFLDALHYDITDDPIHYPRALCVFEMPMQVPLRRHFNSNFSGGFNFYAGLQGQVLVLRTTSTVYNYDYIWDIFIFYPNGVMETKVHATGYVHATFYTPEGLRYGTRLHT
PKKNAWGHQRSYRLQIHSMADQVLPPSLQEERAVTWARYPLAVTKYRESELYSSSIYNQNDPWPVVFEEFLRNNEYIEDEDLVAWVTVGFLHIPHSEDIPNTATPGNSVGFLLRPFNFPPEDPSLASRDTVIVW
```

Sekwencja zakodowana jest w postaci punktów izoelektrycznych odpowiadających poszczególnym aminokwasom, gdzie  $pI = 7 \rightarrow 0$

QGHEAA = [-1.35, -1.03, 0.59, -3.78, -0.98, -0.98]

# Dane fizykochemiczne

```
'Weight', 'Aromaticity', 'Instability', \
'Helix', 'Turn', 'Sheet', 'Extinction', \
'Charge10', 'Charge7', 'Charge4', \
'Isoelectric', 'GRAVY', 'Flexibility', \
'AverageWeight', 'Tiny', 'Small', \
'Aliphatic', 'Aromatic', 'NonPolar', \
'Polar', 'Charged', 'Basic', 'Acidic', \
'Ala', 'Arg', 'Asn', 'Asp', 'Cys', \
```

Weight	Aromaticity	Instability	Helix	Turn	Sheet
57787.31	0.05	46.83	0.21	0.31	0.26
79676.74	0.10	46.60	0.30	0.19	0.27
84141.27	0.09	37.82	0.25	0.26	0.19
75589.89	0.11	42.20	0.35	0.23	0.27
46106.36	0.09	36.52	0.34	0.24	0.27

wyjściowe statystyki

Weight	Aromaticity	Instability	Helix	Turn	Sheet
-0.479995	-1.007024	-0.159426	-1.310321	1.608531	-0.128542
-0.176982	0.903379	-0.180337	0.259898	-1.133641	0.116703
-0.115180	0.521298	-0.978589	-0.612446	0.465959	-1.845259
-0.233556	1.285459	-0.580372	1.132242	-0.219584	0.116703
-0.641693	0.521298	-1.096782	0.957774	0.008930	0.116703

przeskalowane statystyki

# Dane fizykochemiczne

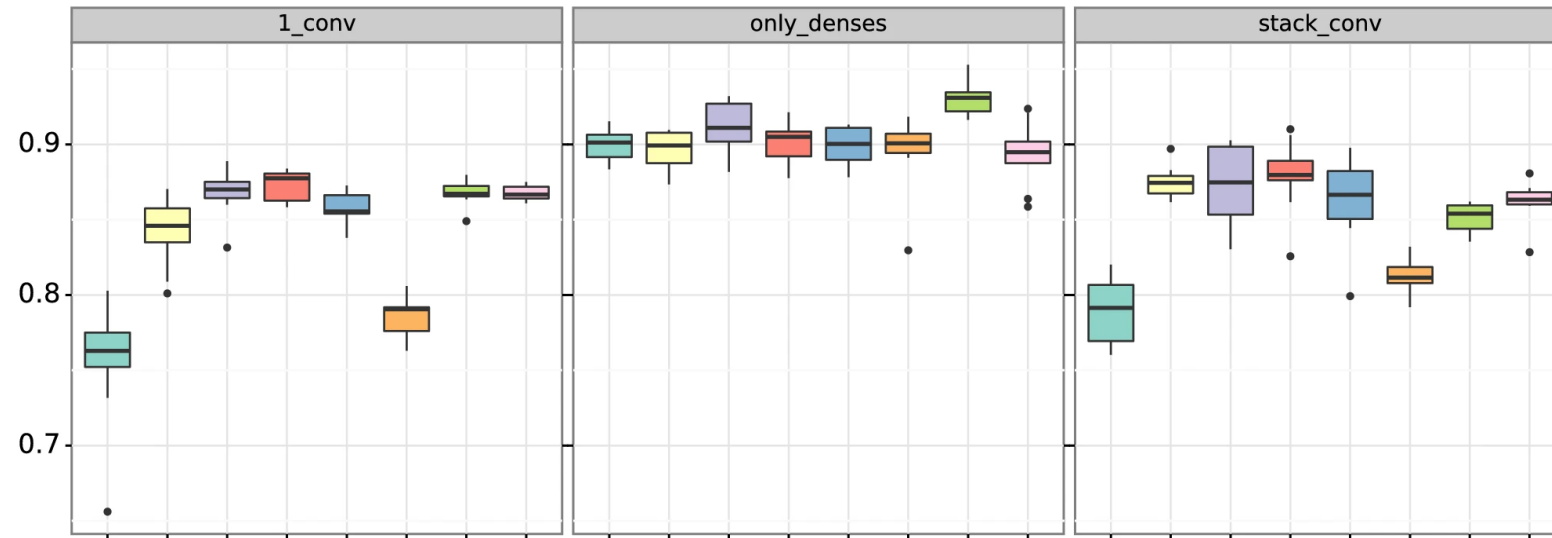
Amino acid composition		Microorganisms				Significance
		1	2	3	4	
Ala (A)	Thermophiles	8.7	11.6	12.5	13.9	*
	Mesophiles	4.6	5.8	10.1	6.5	
	Hyperthermophiles	10.2	10.0	11.2	9.5	
	Psychrophiles	14.1	10.8	10.8	7.9	
Arg (R)	Thermophiles	2.9	3.9	5.3	4.6	*
	Mesophiles	1.7	2.7	1.1	1.8	
	Hyperthermophiles	4.5	1.1	4.0	4.1	
	Psychrophiles	2.5	1.1	1.6	3.4	
Asn (N)	Thermophiles	6.1	7.8	4.1	3.9	*
	Mesophiles	10.2	8.7	5.3	10.5	
	Hyperthermophiles	4.5	5.3	4.0	4.9	
	Psychrophiles	6.9	5.7	5.8	6.6	
Asp (D)	Thermophiles	5.3	2.5	4.3	4.4	**
	Mesophiles	6.5	8.3	6.9	6.5	
	Hyperthermophiles	5.9	7.7	5.2	6.1	
	Psychrophiles	6.9	6.4	7.1	7.2	
Cys (C)	Thermophiles	0.5	0.0	1.4	1.2	*
	Mesophiles	0.4	0.0	0.0	0.2	
	Hyperthermophiles	0.5	0.0	1.2	0.5	
	Psychrophiles	1.6	0.9	1.3	1.9	

# Architektura sieci 1 – brak konwolucji

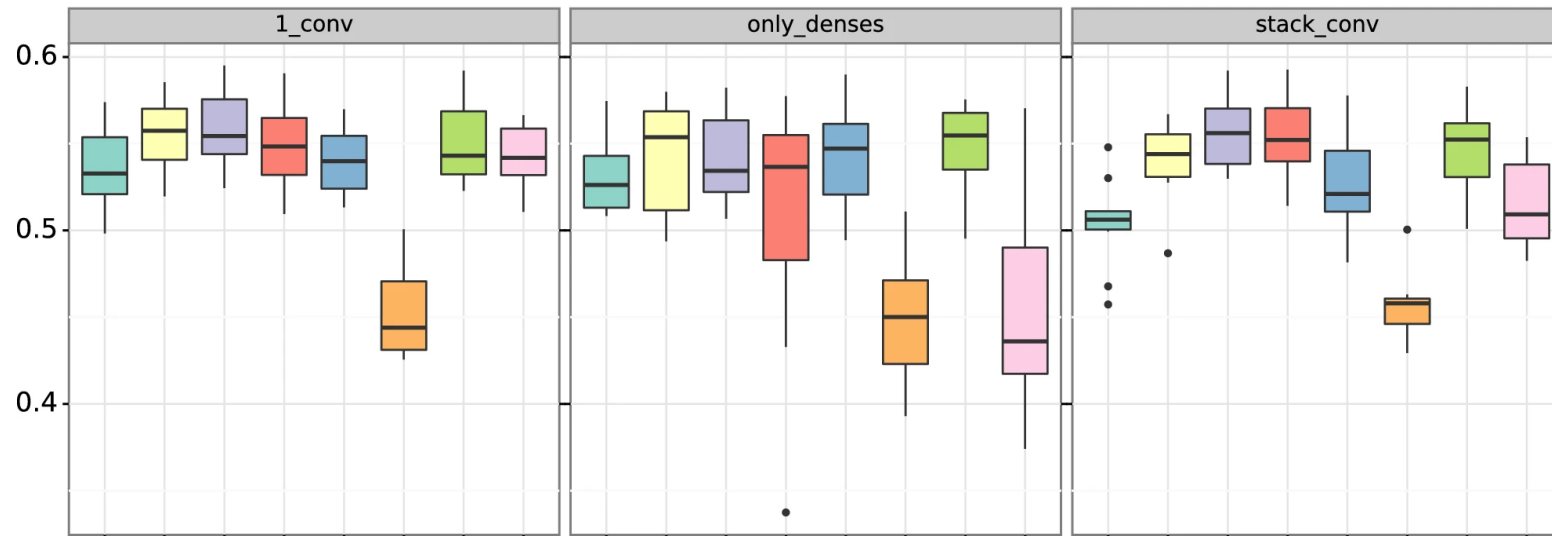
```
def noConv(input_size, hidden_size, num_classes):  
    inputs = Input(shape=(input_size,))  
    fc1 = Dense(hidden_size, activation='relu')(inputs)  
    dropout1 = Dropout(0.5)(fc1)  
    fc2 = Dense(hidden_size, activation='relu')(dropout1)  
    dropout2 = Dropout(0.25)(fc2)  
    fc3 = Dense(num_classes, activation='sigmoid')(dropout2)  
    model = Model(inputs, fc3)  
    return model
```



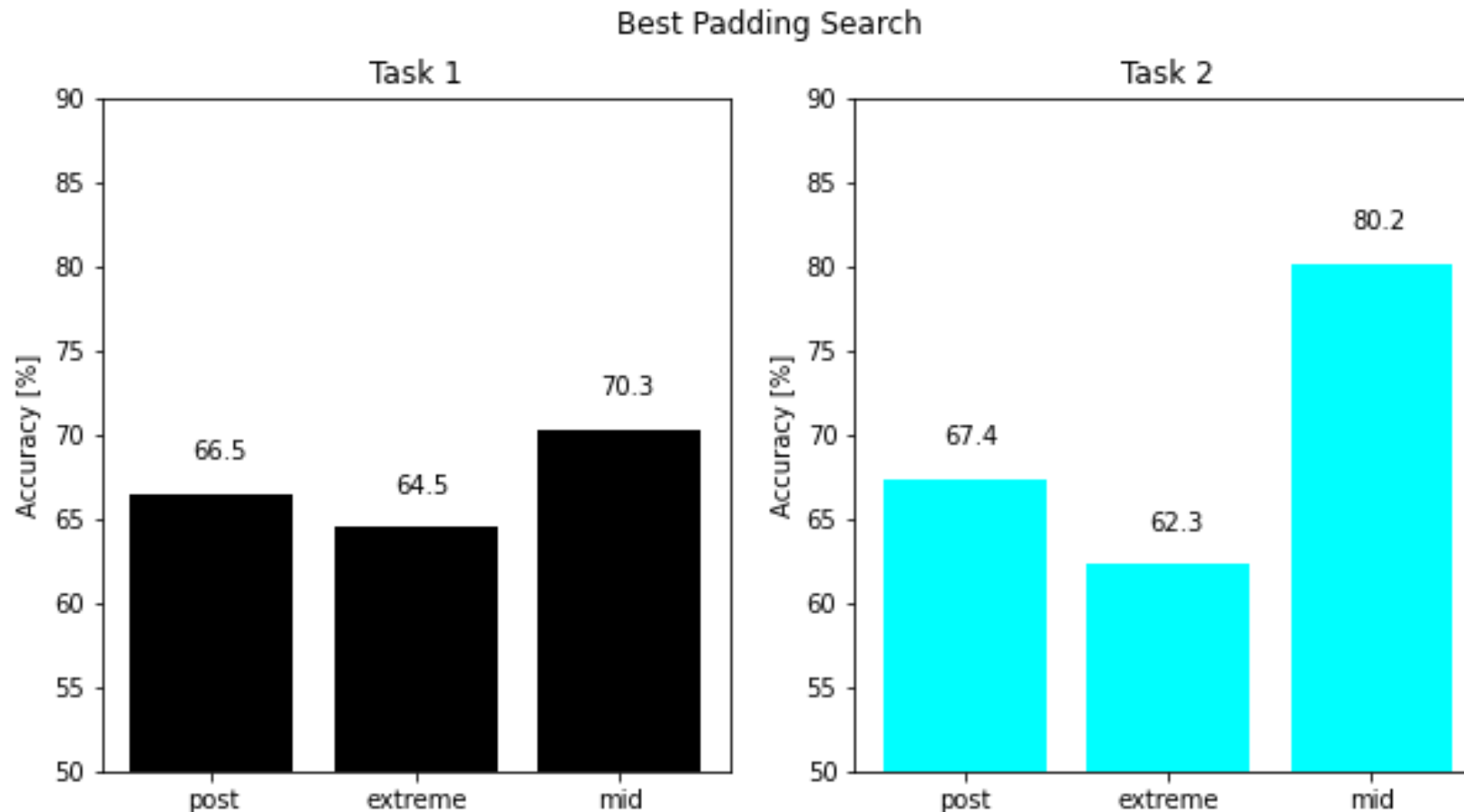
Task 1 - F1-score on test(10 holdouts)



Task 2 - F1-score on test(10 holdouts)



# Wybór najlepszego paddingu



Task1 – enzym lub nie enzym

Task2 – enzym z jednej z 7 klas

# Architektura 2 – jedna warstwa konwolucyjna

```
def oneCNN(input_size, num_classes):  
  
    inputs = Input(shape=(input_size, 1))  
    conv_layer1 = Conv1D(32, 3, activation='relu', \  
                        input_shape=(input_size, 1))(inputs)  
    pooling_layer1 = MaxPooling1D(pool_size=2)(conv_layer1)  
    dropout1 = Dropout(0.5)(pooling_layer1)  
    flatten = Flatten()(conv_layer1)  
    fc1 = Dense(16, activation='relu')(flatten)  
    fc2 = Dense(8, activation='relu')(fc1)  
    fc3 = Dense(num_classes, activation='sigmoid')(fc2)  
  
    model = Model(inputs, fc3)  
    return model
```

# Architektura 3 – pięć warstw konwolucyjnych

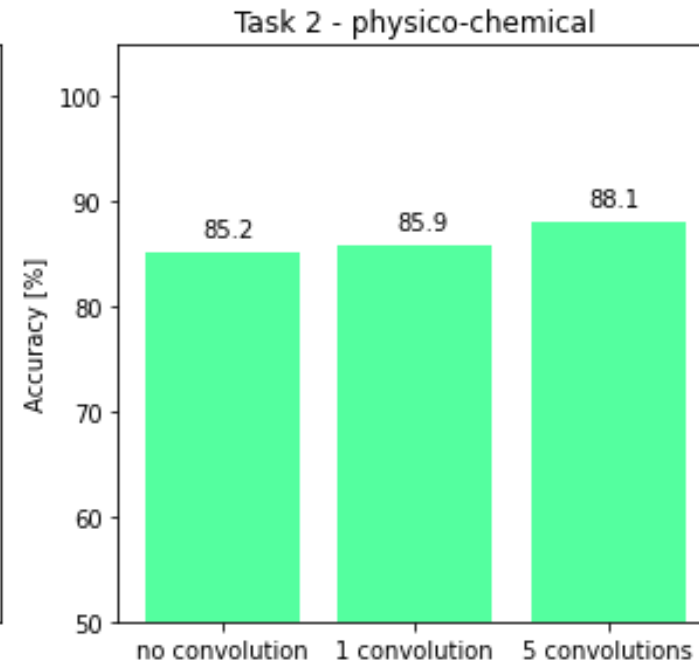
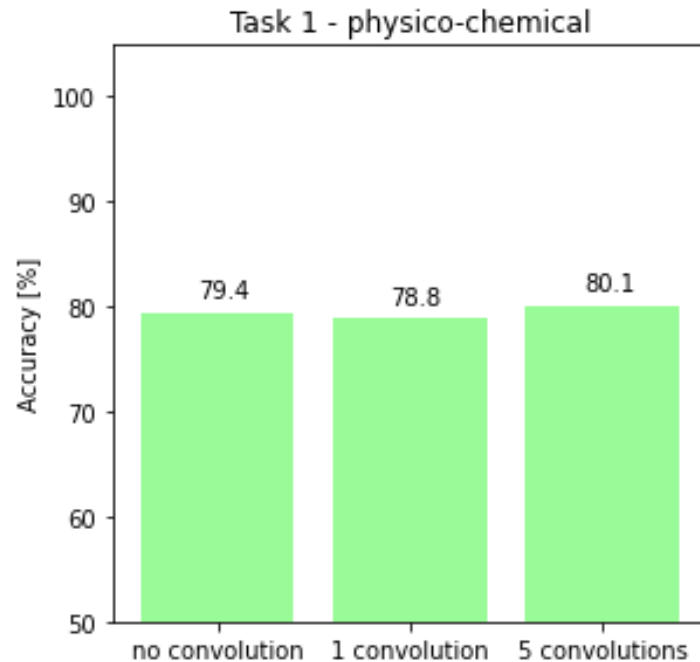
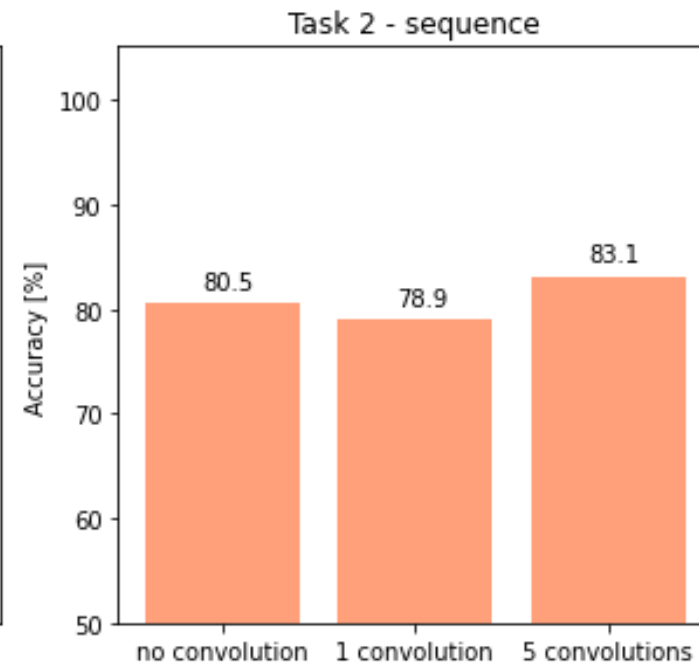
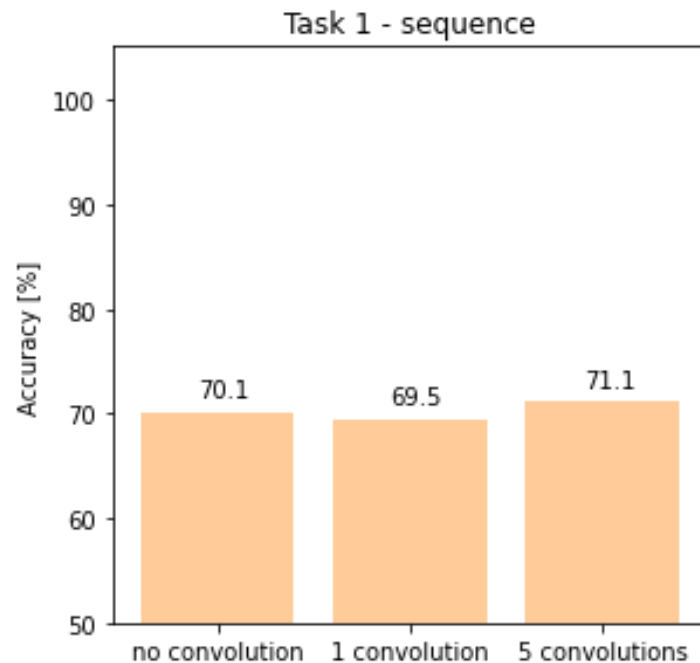
```
def stackedCNN(input_size, num_classes):  
    inputs = Input(shape=(input_size, 1))  
    conv_layer1 = Conv1D(32, 2, activation='relu', \  
                        input_shape=(input_size, 1))(inputs)  
    pooling_layer1 = MaxPooling1D(pool_size=2)(conv_layer1)  
    conv_layer2 = Conv1D(256, 2, activation='relu')(pooling_layer1)  
    dropout1 = Dropout(0.5)(conv_layer2)  
    conv_layer3 = Conv1D(128, 2, activation='relu')(dropout1)  
    pooling_layer2 = MaxPooling1D(pool_size=2)(conv_layer3)  
    conv_layer4 = Conv1D(64, 2, activation='relu')(pooling_layer2)  
    dropout2 = Dropout(0.25)(conv_layer4)  
    conv_layer5 = Conv1D(32, 2, activation='relu')(dropout2)  
    flatten = Flatten()(conv_layer5)  
    fc1 = Dense(16, activation='relu')(flatten)  
    fc2 = Dense(8, activation='relu')(fc1)  
    fc3 = Dense(num_classes, activation='sigmoid')(fc2)  
  
    model = Model(inputs, fc3)  
    return model
```

# Hiperparametry

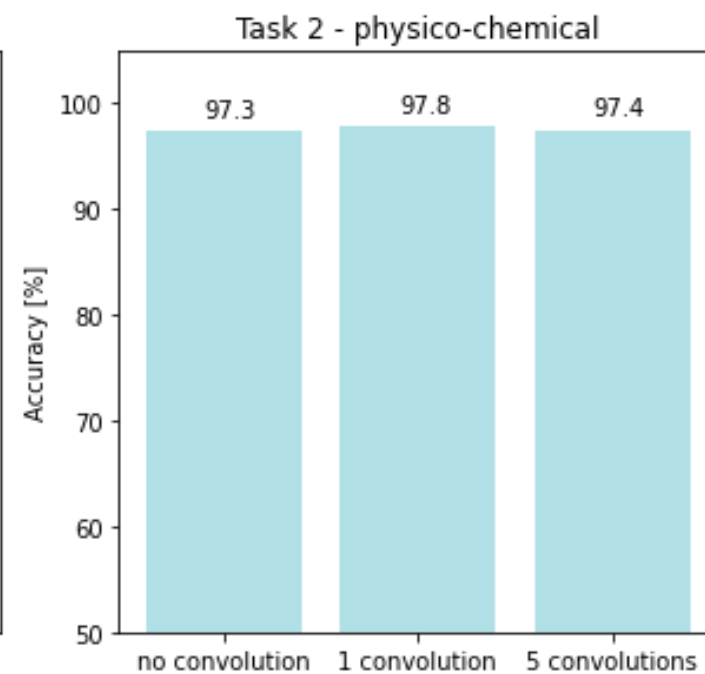
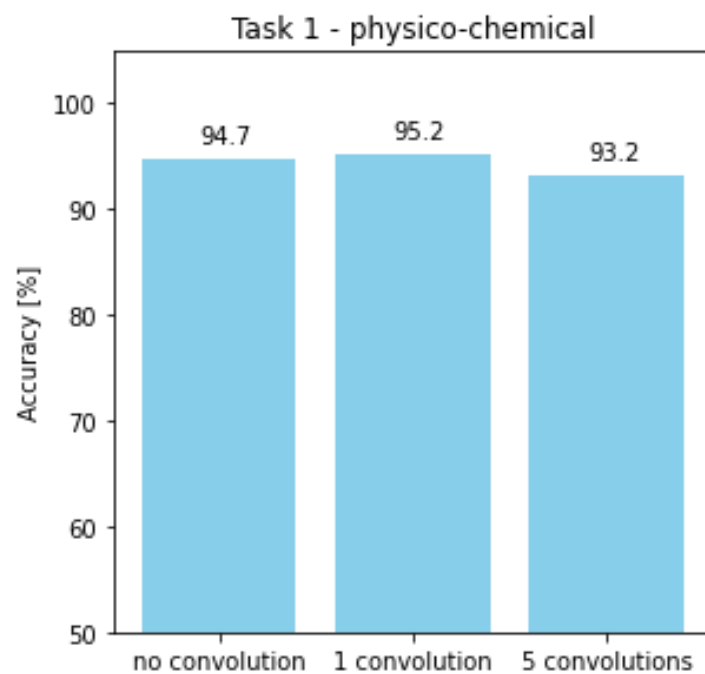
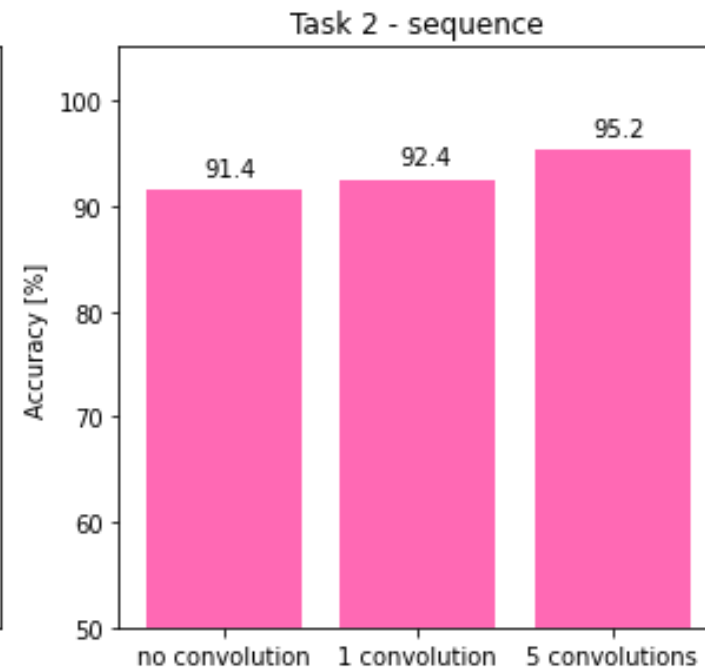
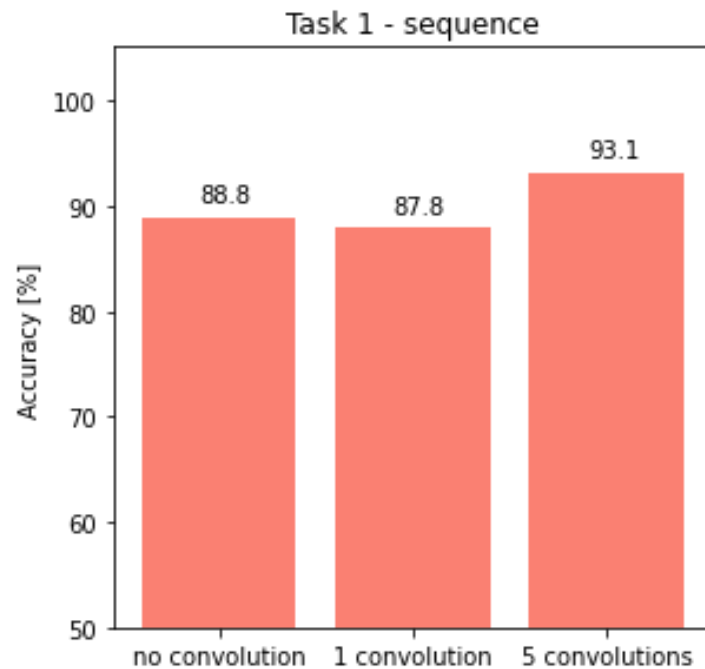
```
batch_size = 64  
epochs = 45
```

```
tf.keras.optimizers.Adam(  
    learning_rate=0.001,  
    beta_1=0.9,  
    beta_2=0.999,  
    epsilon=1e-07,  
    amsgrad=False,  
    weight_decay=None,  
    clipnorm=None,  
    clipvalue=None,  
    global_clipnorm=None,  
    use_ema=False,  
    ema_momentum=0.99,  
    ema_overwrite_frequency=None,  
    jit_compile=True,  
    name='Adam',  
    **kwargs  
)
```

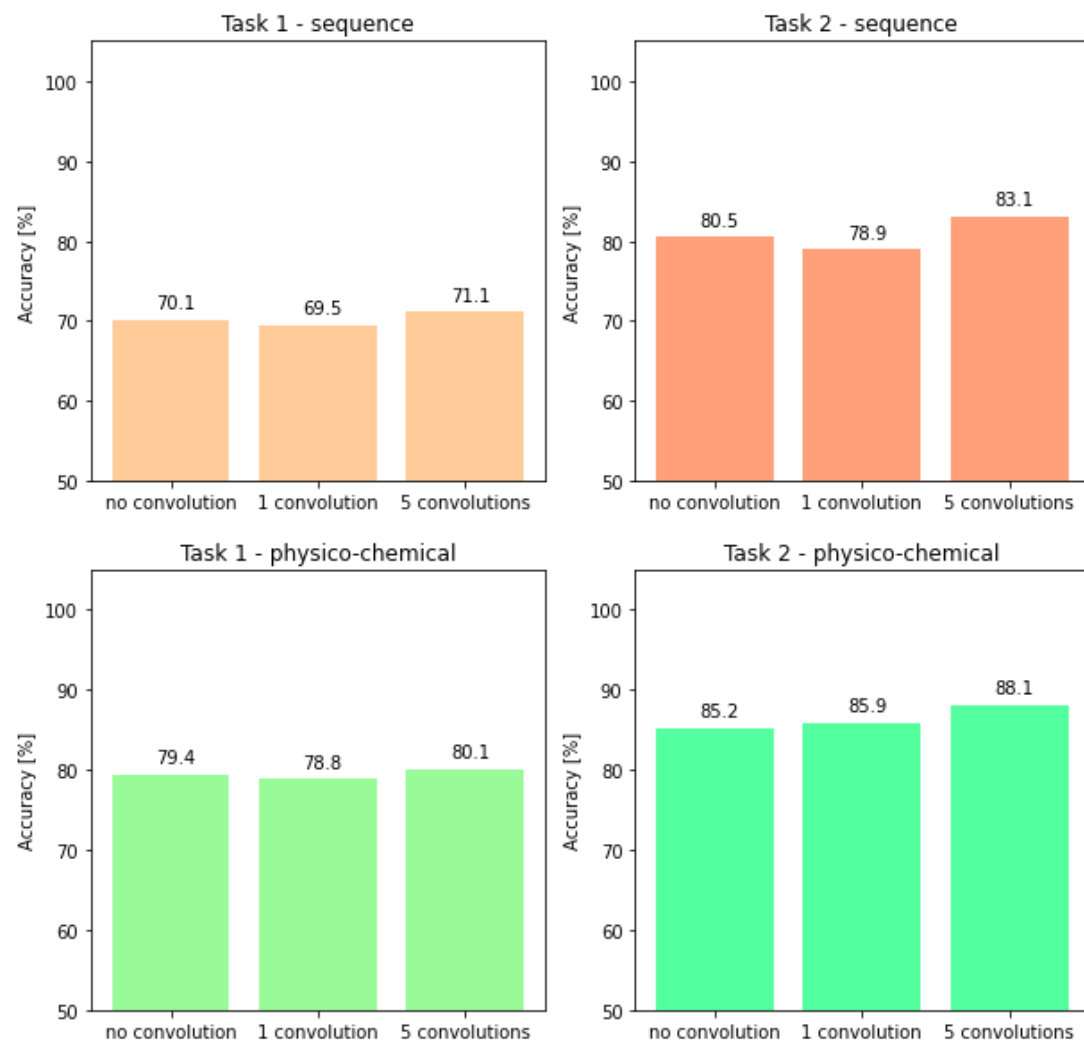
# Dataset 1 – wyniki



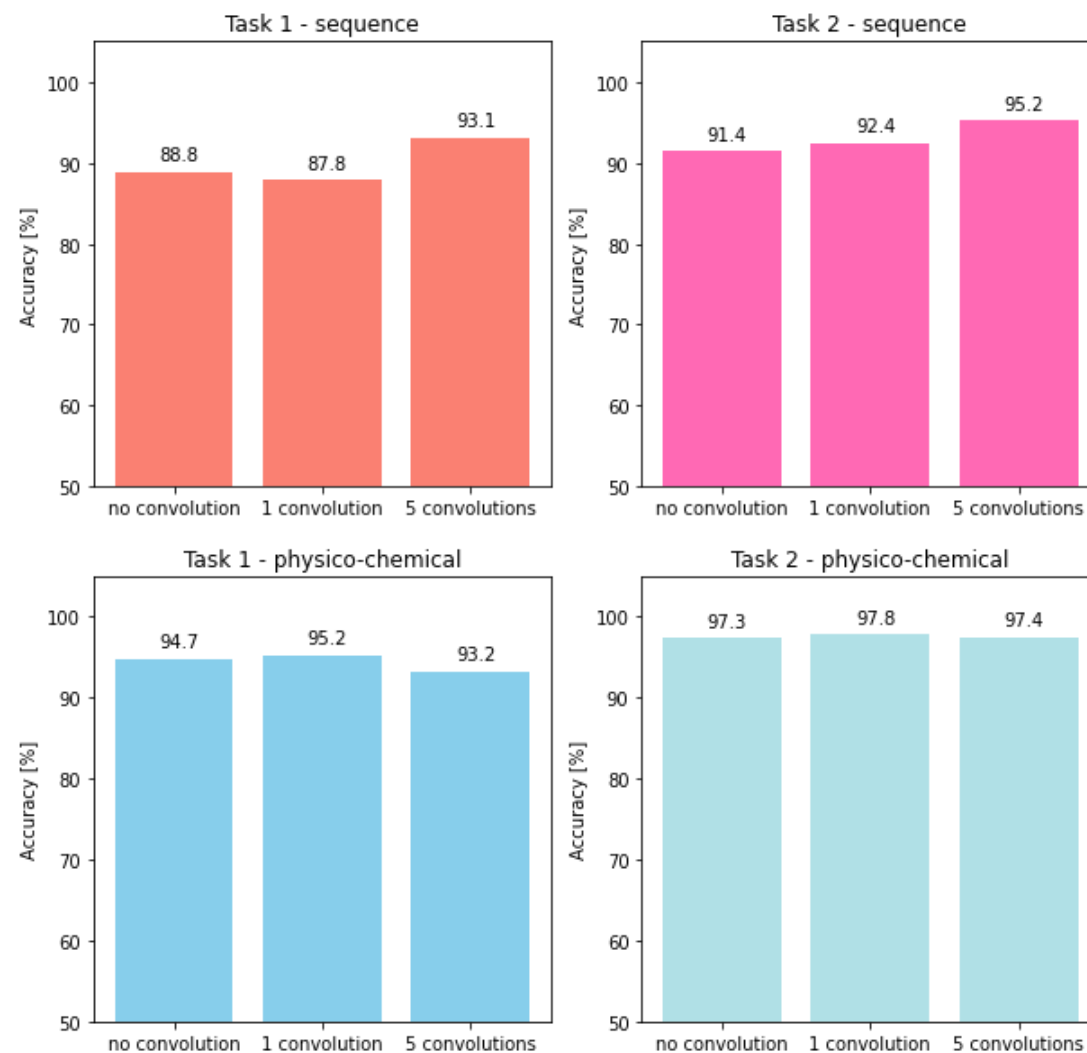
# Dataset 2 – wyniki



# Dataset 1



# Dataset 2

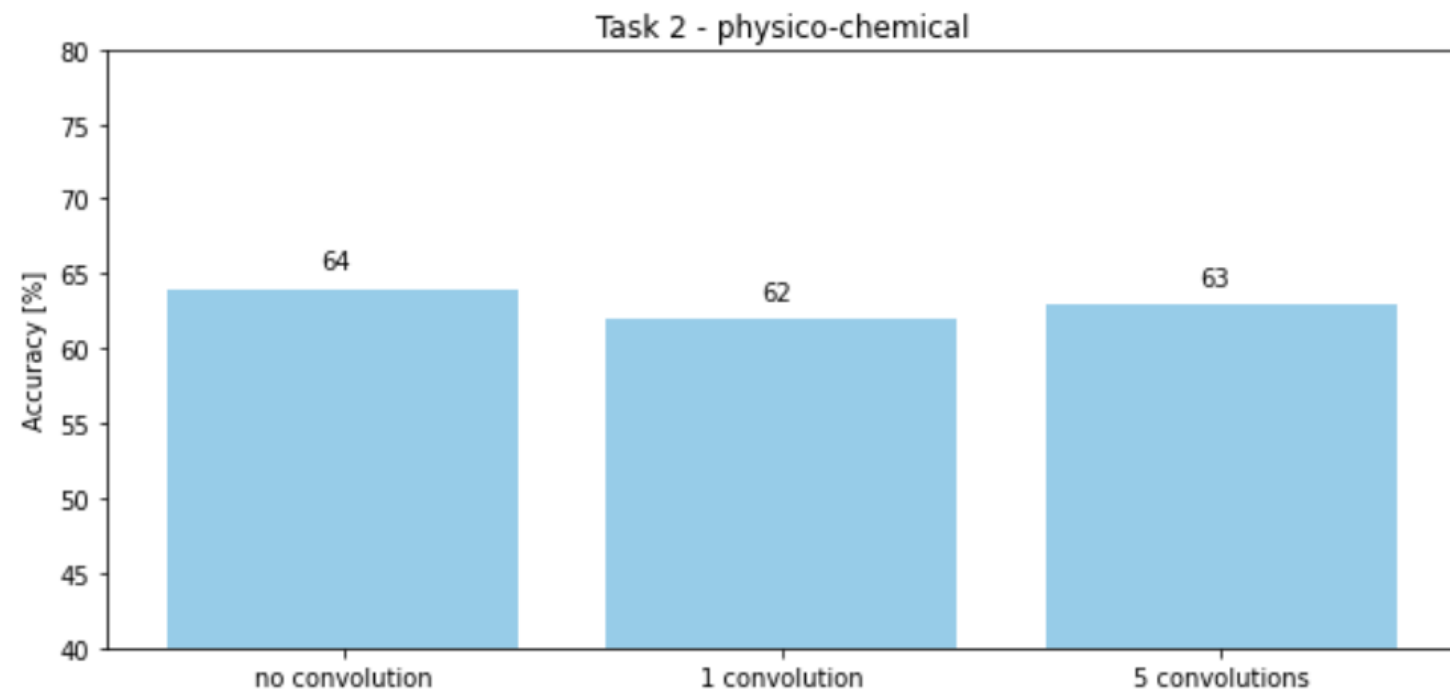
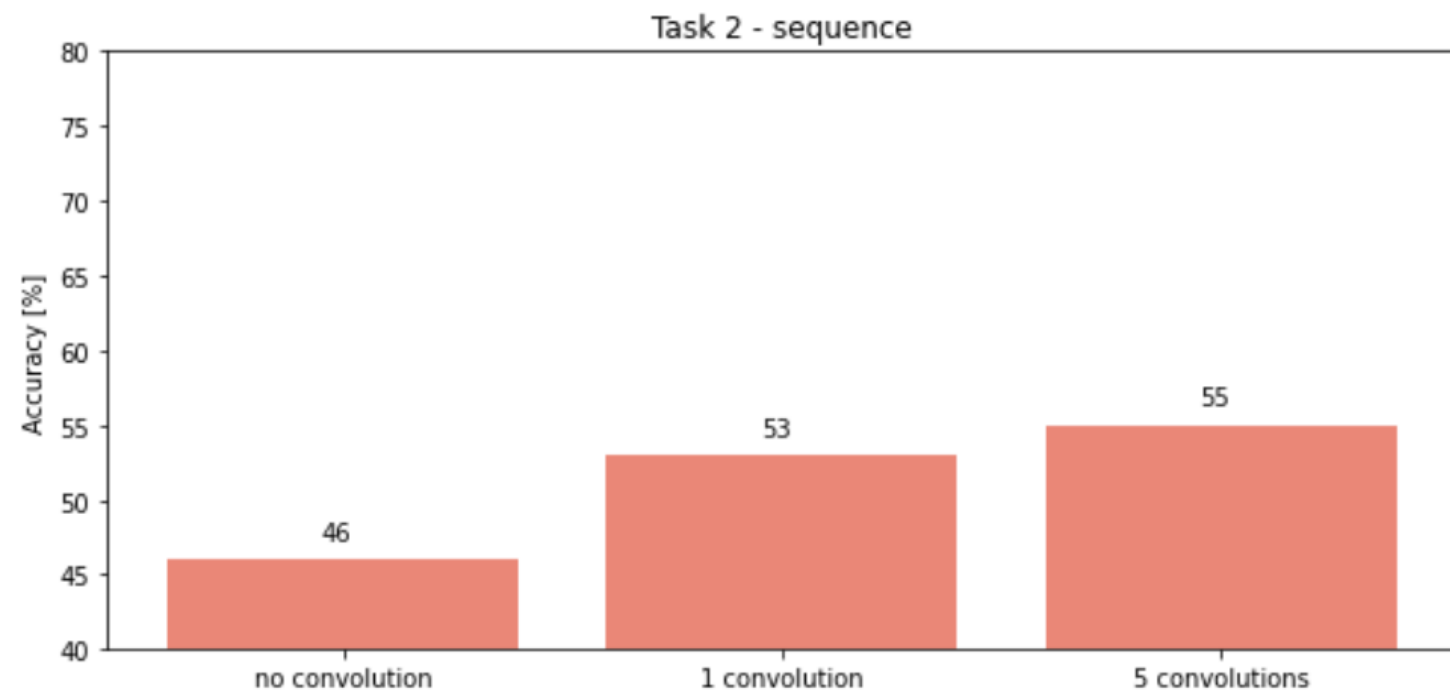




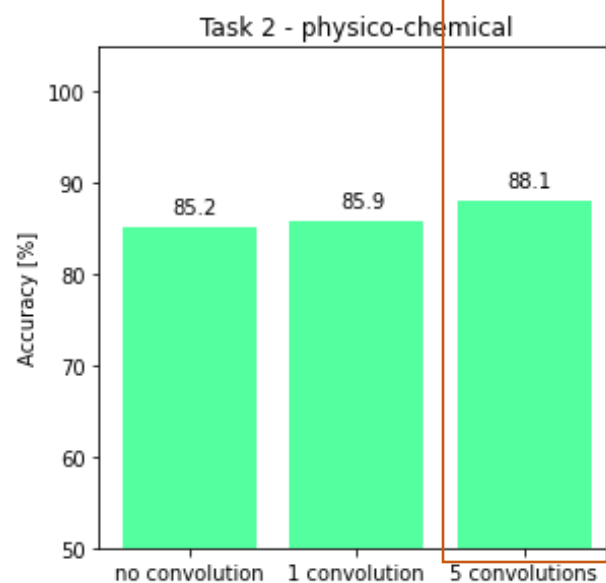
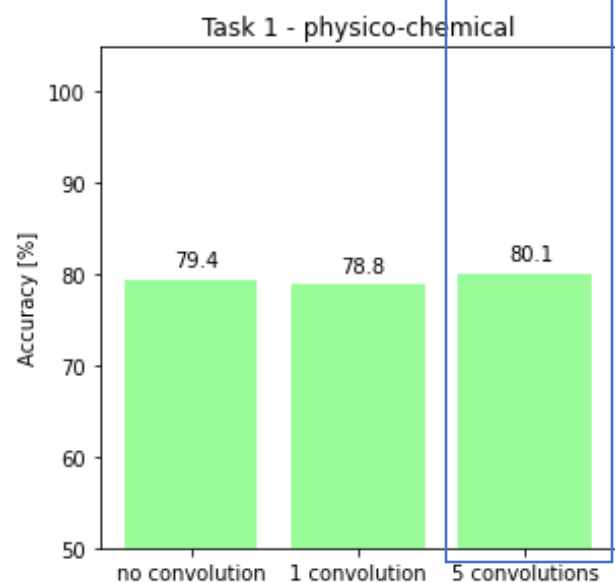
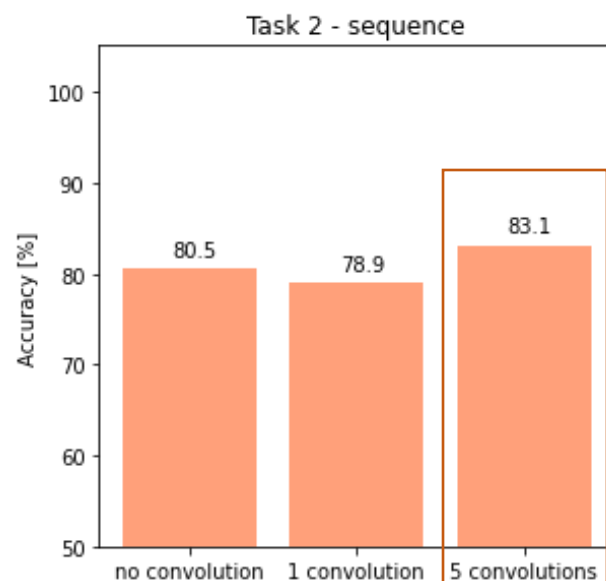
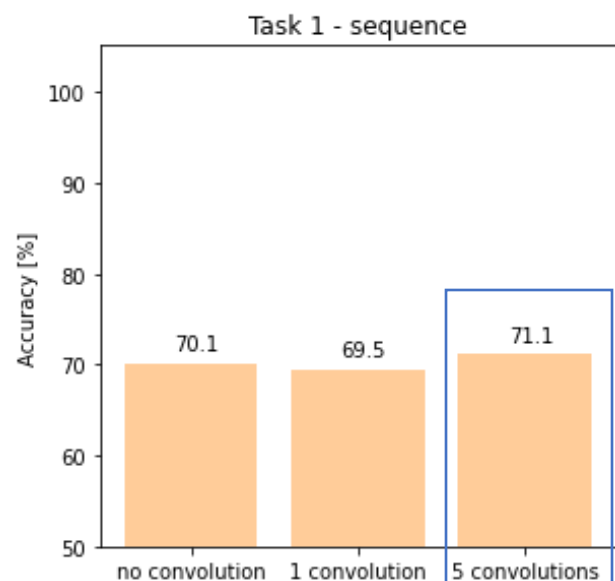
# Dataset 3 – lokalizacja komórkowa

- 290 sekwencji białek jądra komórkowego
- 290 sekwencji białek zewnątrzkomórkowych
- 290 sekwencji białek błony komórkowej

# Dataset 3 – wyniki



# Dataset 1 – kombinacja predykcji sieci



73.8%

89.9%

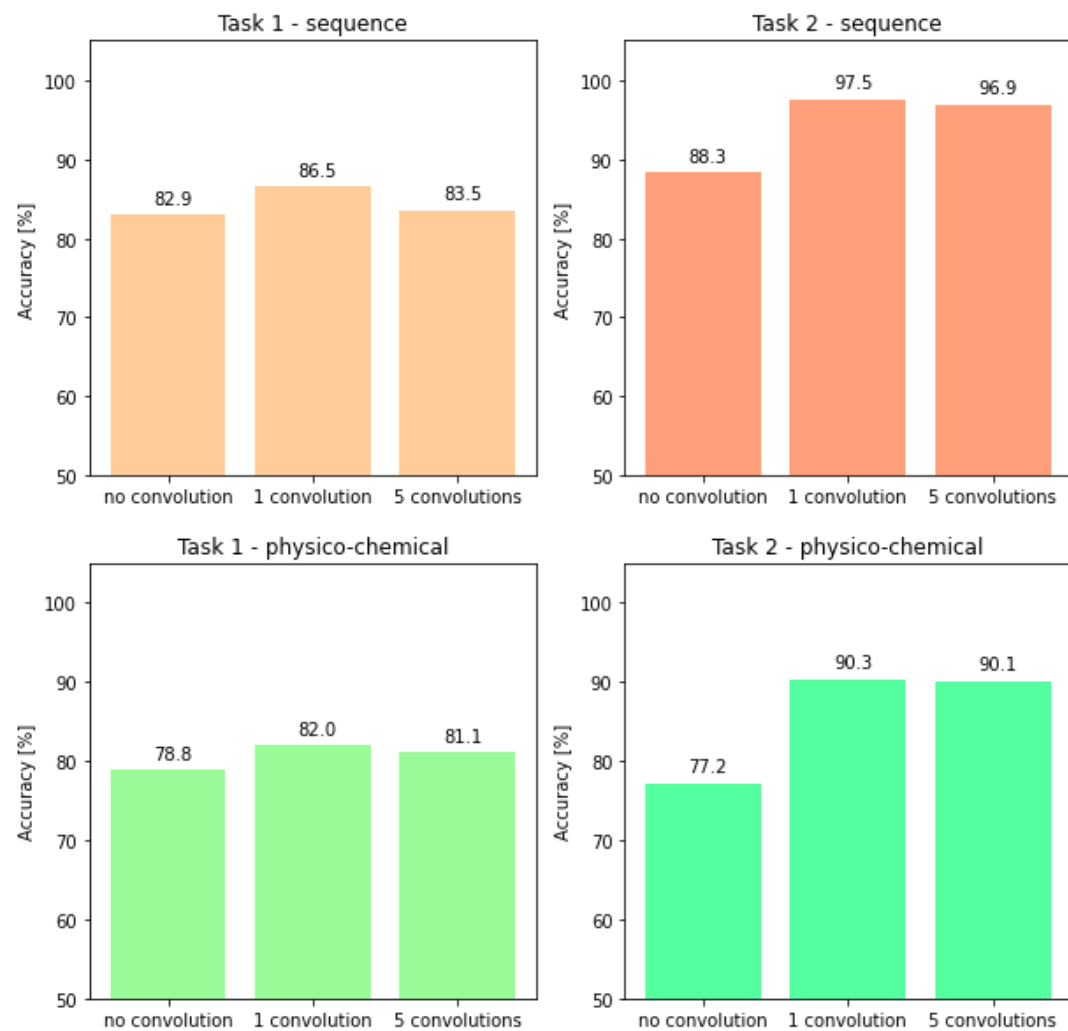
# Dyskusja - parametry task1

Dataset	Architektura	Wymiar danych	Parametry	Accuracy (test)
1	no_conv	28000 x 1000	68000	70.0%
1	5_conv	28000 x 1000	229000	71%
1	no_conv	28000 x 28	6000	79.4%
1	5_conv	28000 x 28	104000	80.1%
2	no_conv	10000 x 1000	68000	88.8%
2	5_conv	10000 x 1000	229000	93.1%
2	no_conv	10000 x 28	6000	94.7%
2	5_conv	10000 x 28	104000	93.2%

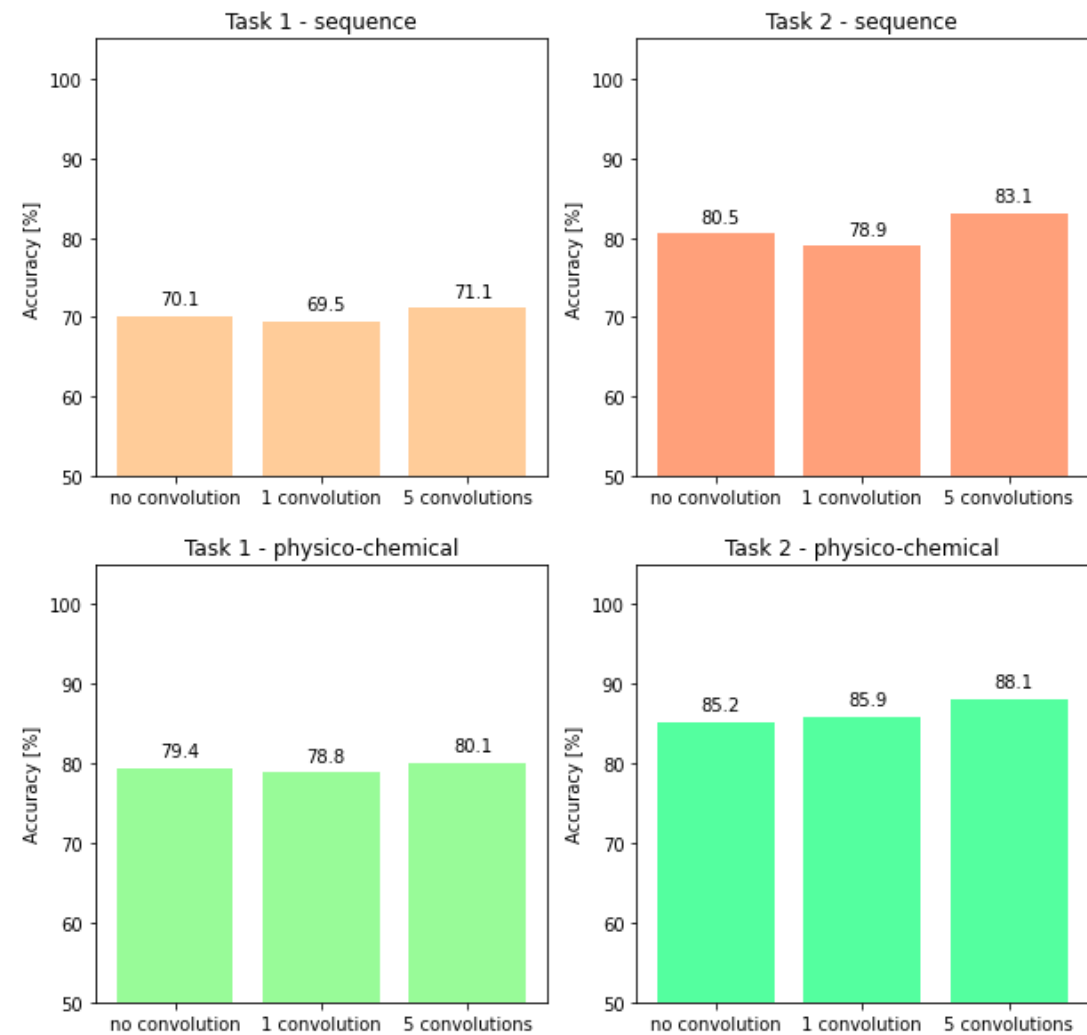
# Dyskusja - parametry task1

Dataset	Architektura	Wymiar danych	Parametry	Accuracy (test)
1	no_conv	28000 x 1000	68000	70.0%
1	5_conv	28000 x 1000	229000	71%
1	no_conv	28000 x 28	6000	79.4%
1	5_conv	28000 x 28	104000	80.1%
2	no_conv	10000 x 1000	68000	88.8%
2	5_conv	10000 x 1000	229000	93.1%
2	no_conv	10000 x 28	6000	94.7%
2	5_conv	10000 x 28	104000	93.2%

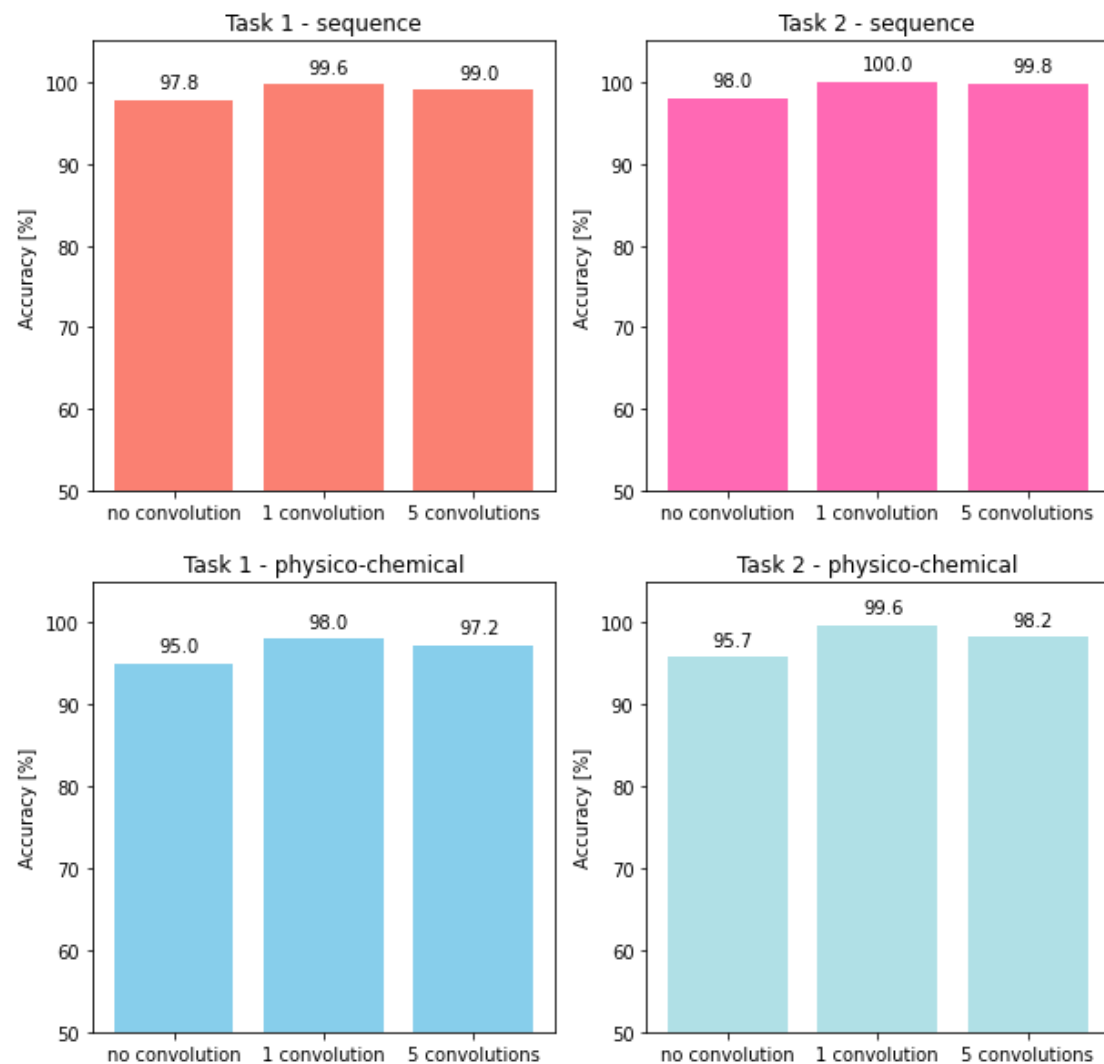
## zestaw treningowy



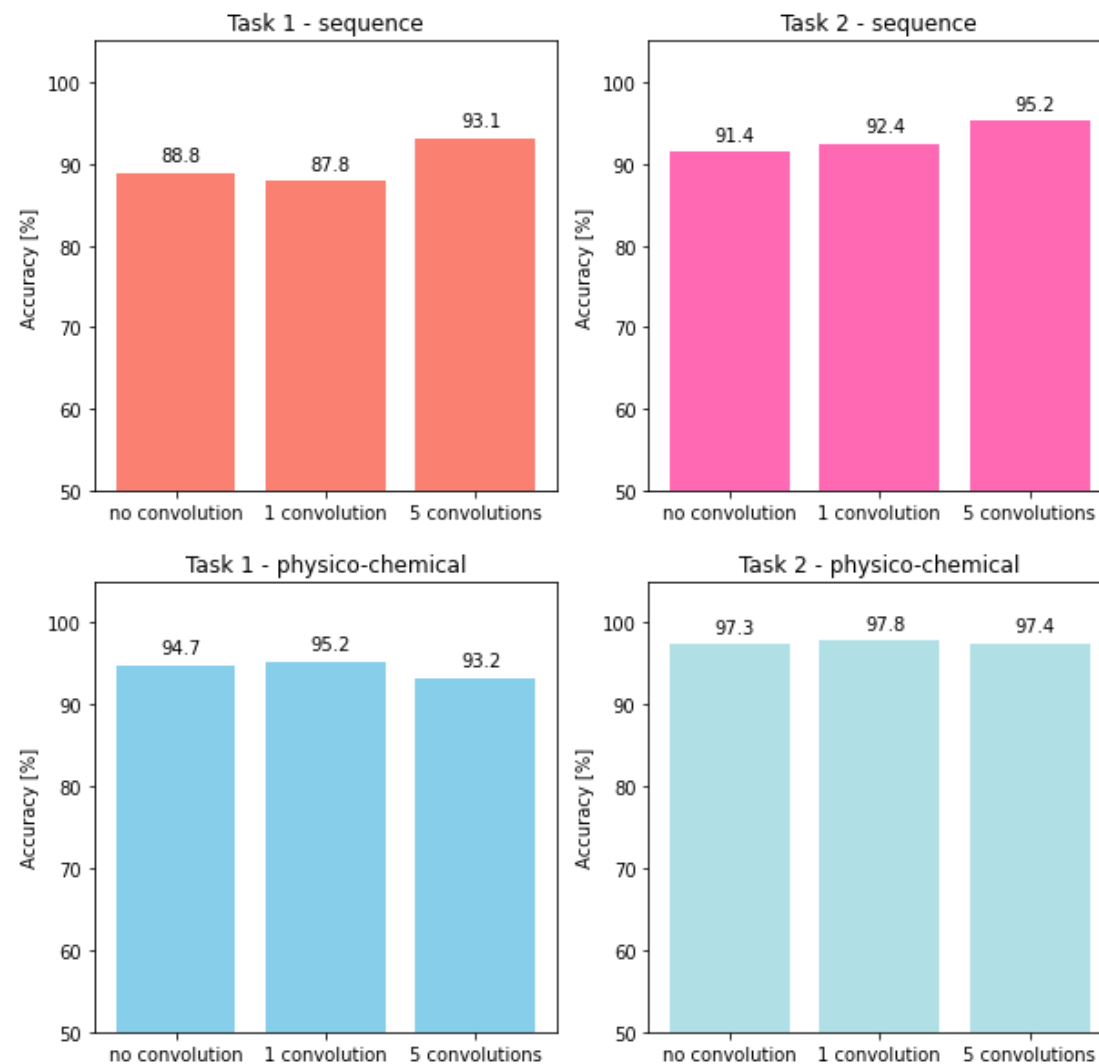
## zestaw testowy



## zestaw treningowy



## zestaw testowy



# Źródła

- [https://pl.wikipedia.org/wiki/Numer\\_EC](https://pl.wikipedia.org/wiki/Numer_EC)
- <https://en.wikipedia.org/wiki/Ligase>
- Raj et al., J Proteomics Bioinform 2017, 10:12 DOI: 10.4172/jpb.1000459
- <https://www.nature.com/articles/s41598-020-71450-8/figures/2>
- <https://www.sciencedirect.com/science/article/abs/pii/S0022519318305654>