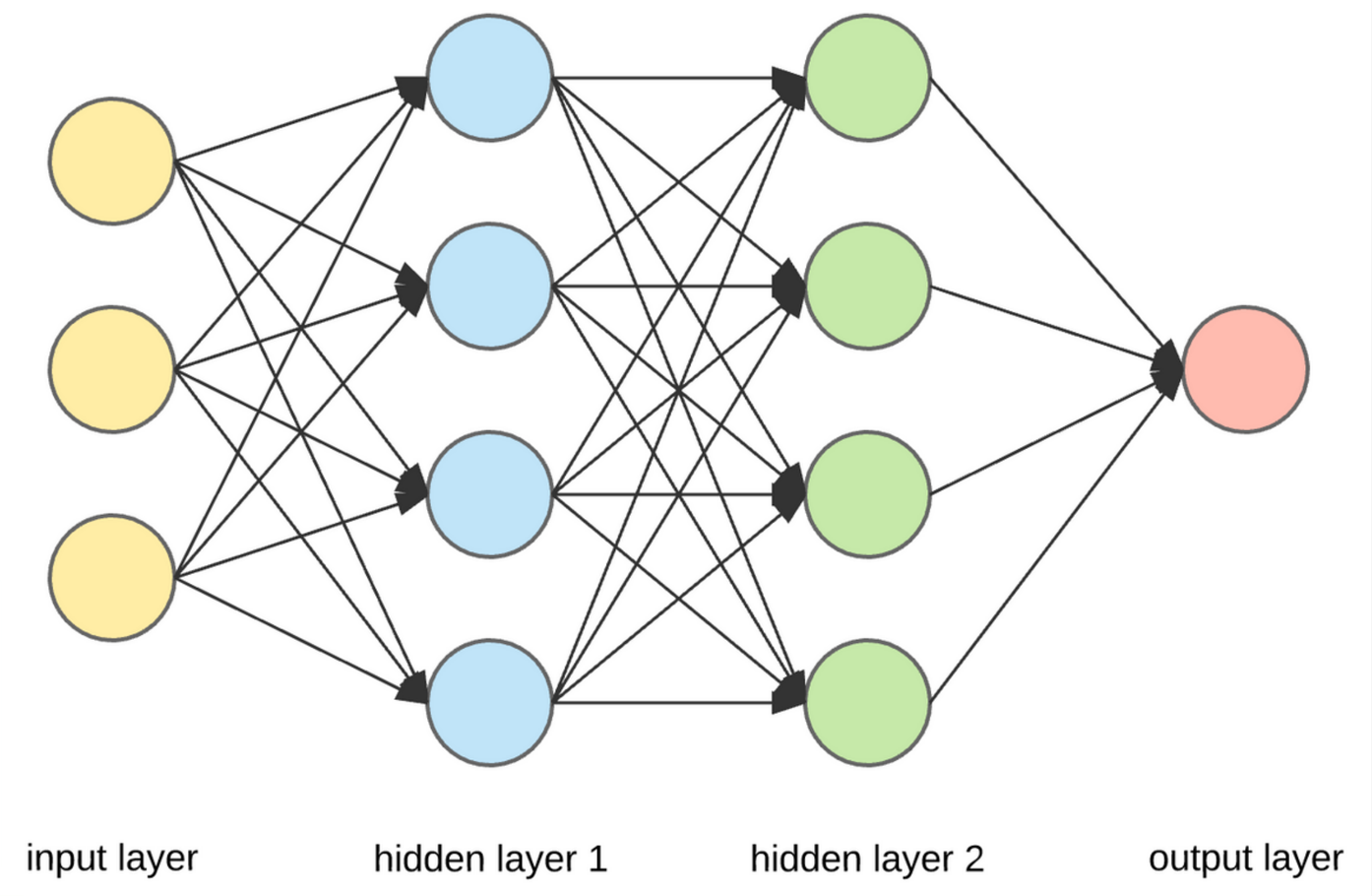
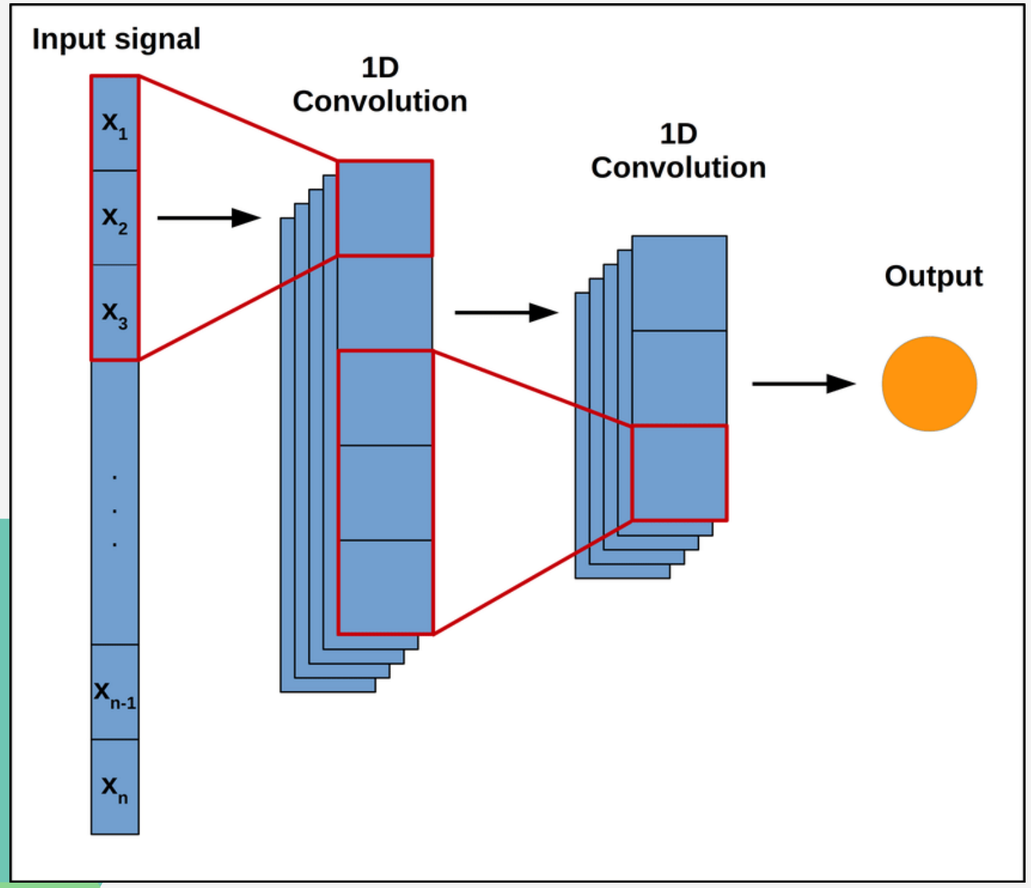
NN Classifier for fish microbiome

Marek Sztuka

Neural Network

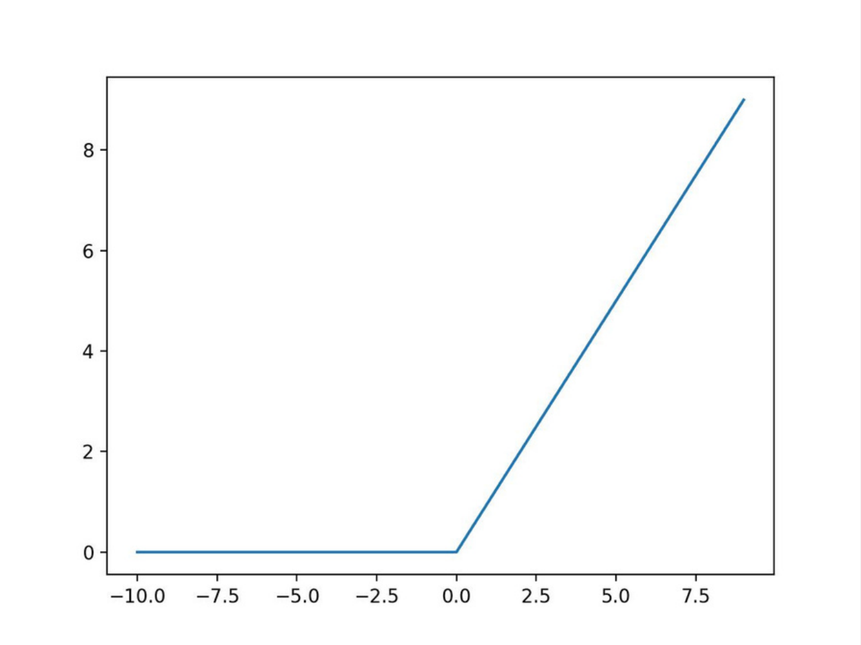


Convolution



Parameters

- Relu as activation for all layers
- Softmax for output layer
- Normalisation across rows
- Epsylion - 0.000001
- metrics - accuracy



Data

number	setup	pond numbers	water supplementation	feed suplementation
0	control	8,14,18,24,29	NO	NO
1	set 1	12,16,19,23,26	Em farma	NO
2	set 2	10,21,28,30,32	Em farma	EM
3	set 3	9,13,17,20,27	EM	NO
4	set 4	11,15,22,25,31	EM	EM

Bacteria samples

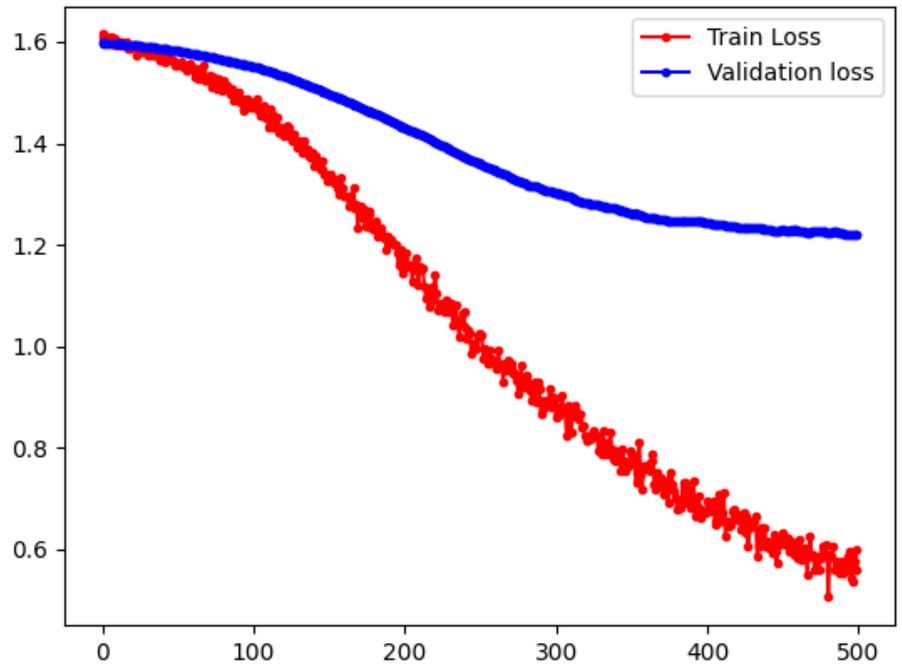
126 collumns with bacteria abduance

pond	d__Bacteria;p__Firmicutes;c__Bacilli;o__Erysipelotrichales;f__Erysipelotrichaceae
Gut_S10	2319.0
Gut_S10	3391.0
Gut_S10	1338.0
Gut_S10	1582.0
Gut_S10	1463.0

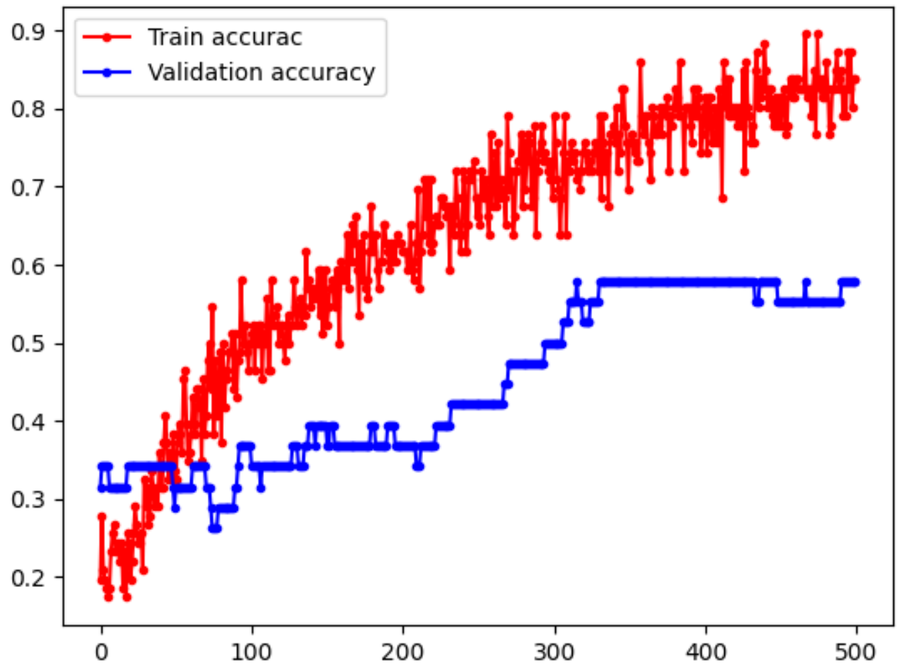
Dense

Best results:

No normalisation, no epsylion, 2 dropouts 0.25, learning rate - 0.0001



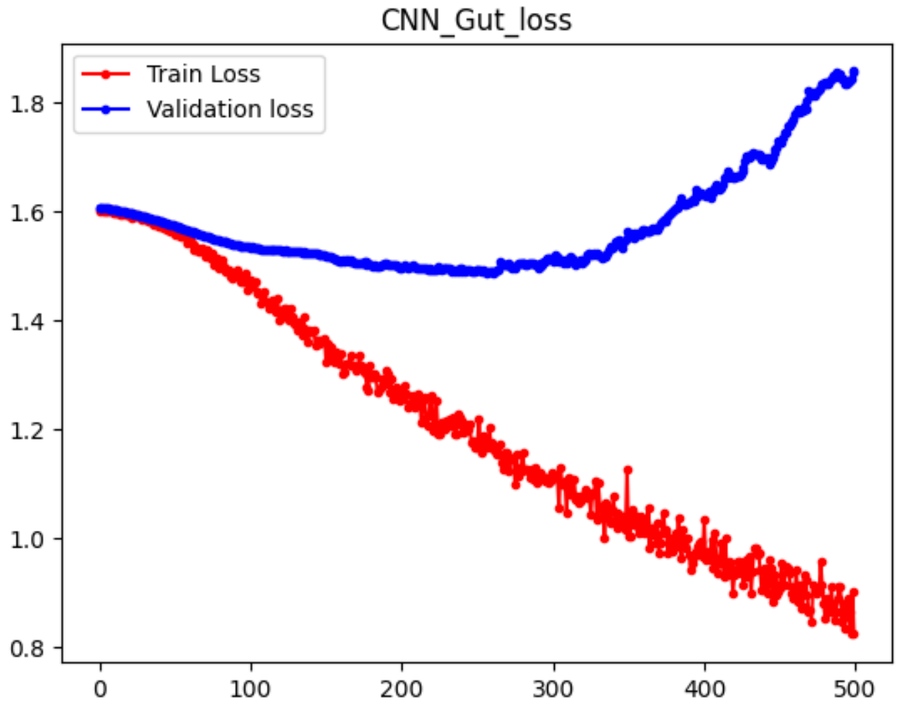

```
Layer (type)
=====================
dense_16 (Dense)

dropout_4 (Dropout)

dense_17 (Dense)

dropout_5 (Dropout)

dense_18 (Dense)

dense_19 (Dense)
```



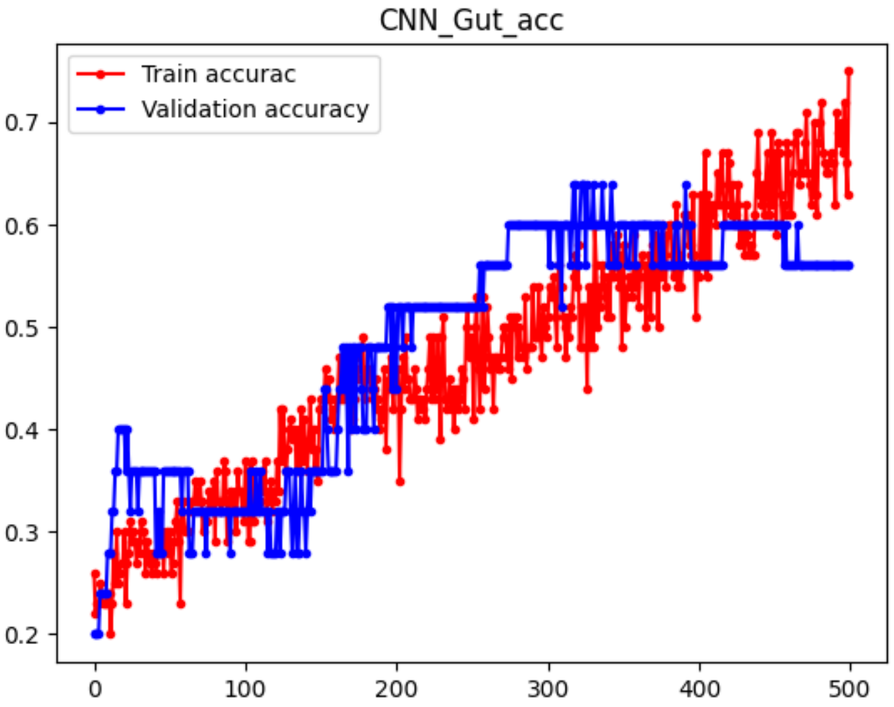
Convolution

Best results:

Normalisation, epsylion, dropout 0.25, learning rate - 0.0001



Layer (type)
conv1d_23 (Conv1D)
conv1d_24 (Conv1D)
flatten_8 (Flatten)
dense_31 (Dense)
dropout_8 (Dropout)
dense_32 (Dense)
dense_33 (Dense)



KEGG path samples

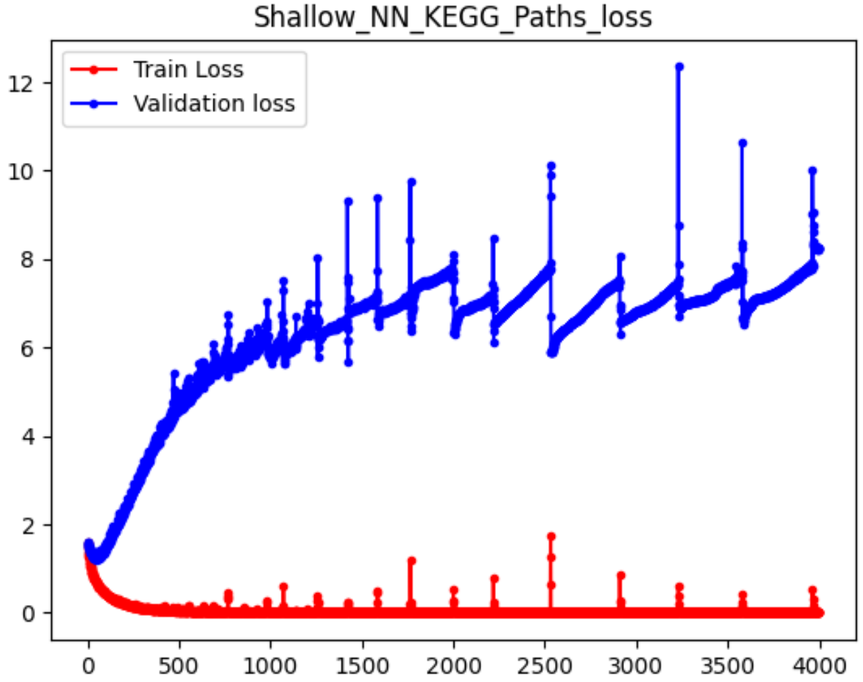
222 Columns with KEGG paths abduance

KEGG	ko05340	ko00564	ko00680	ko00562	ko03030	ko00561	ko00440	ko00250	ko00740
S32.73.Gut	14790.25	123736.1875	148050.1875	18689.080078	132018.515625	89007.296875	7374.740234	186244.96875	57747.929688
S32.72.Gut	8609.860352	68135.296875	100486.039062	12440.19043	65381.25	45692.148438	6406.540039	101287.507812	29035.929688
S32.75.Gut	7553.470215	83516.789062	126634.679688	11049.459961	93353.84375	50052.351562	2431.969971	125494.703125	39925.921875
S32.74.Gut	7883.540039	58946.179688	82612.476562	9036.230469	58041.859375	38942.230469	4435.240234	86281.023438	25553.609375

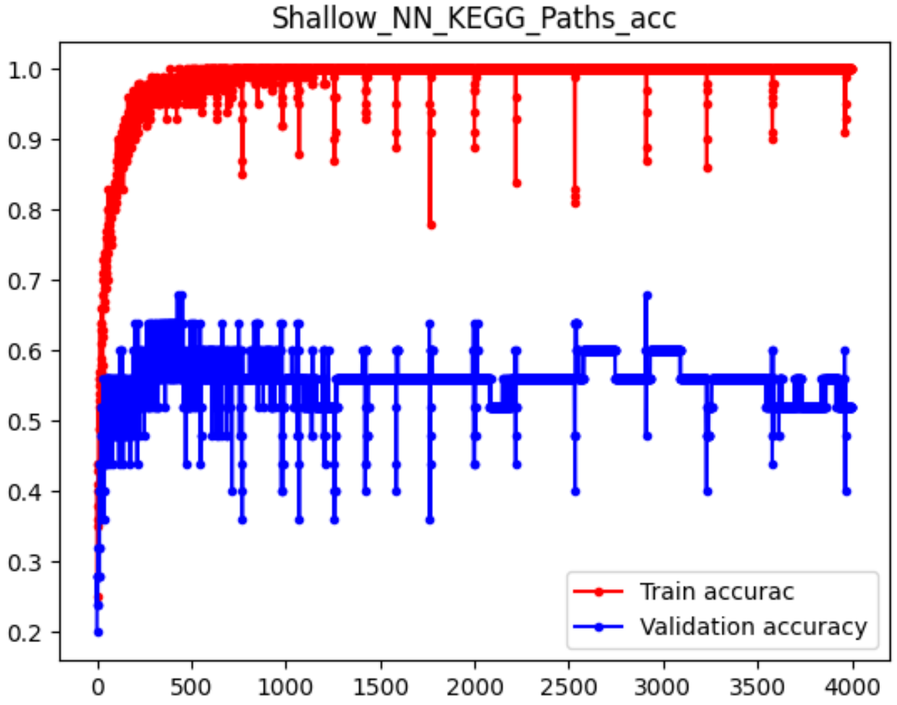
Shallow

Best results:

No Normalisation, no epsylion, dropout 0, learning rate - 0.0005



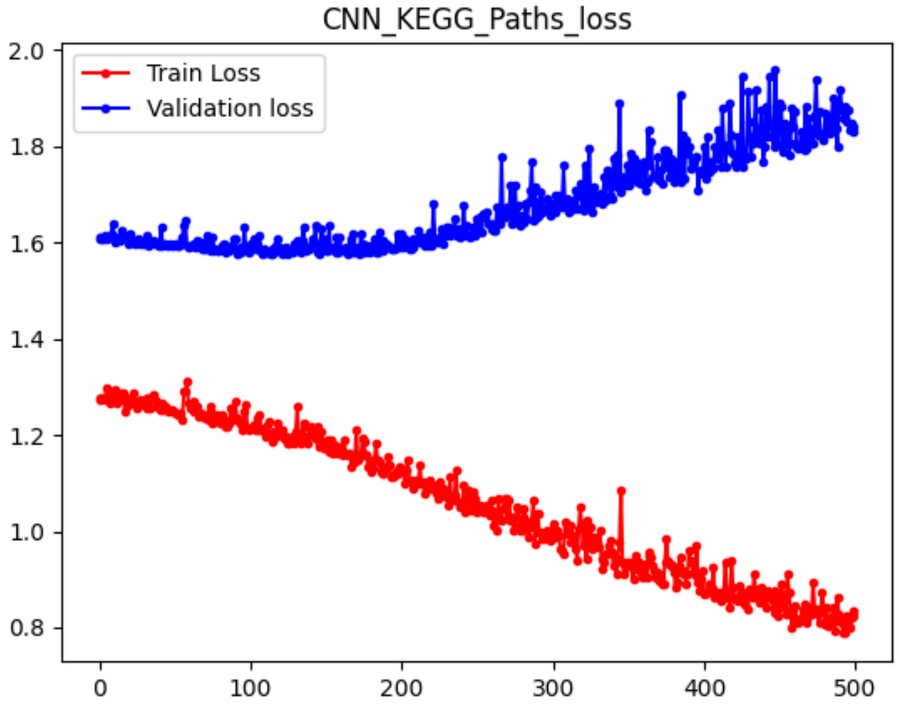

```
 Layer (type)
=====================
 dense_26 (Dense)

 dropout_6 (Dropout)

 dense_27 (Dense)

 dense_28 (Dense)
```



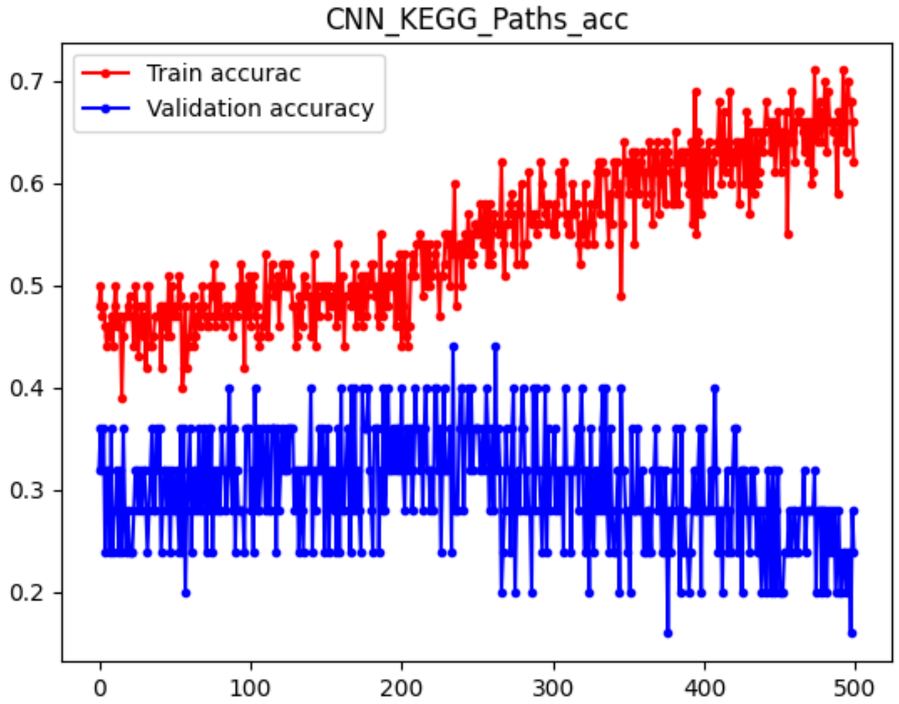
Convolution

Best results:

Normalisation, no epsylion, dropout 0, learning rate - 0.0001




```
Layer (type)
=====================
conv1d_17 (Conv1D)

dense_25 (Dense)

dense_26 (Dense)

flatten_8 (Flatten)

dense_27 (Dense)
```



Orthology Samples

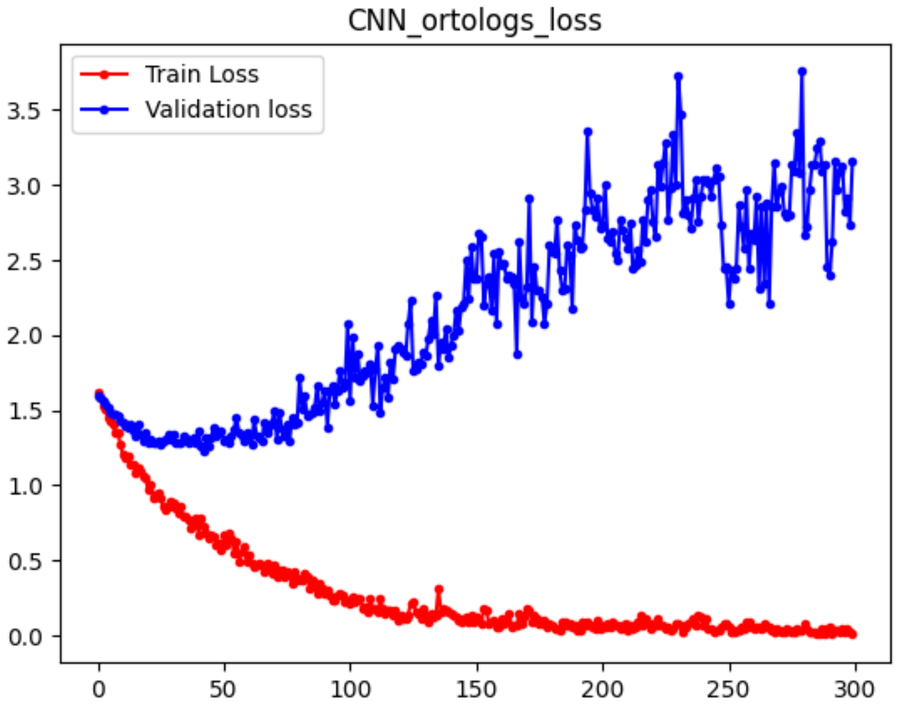
6298 columns with orthology abduance

pond	K00001	K00003	K00004	K00005	K00009	K00010	K00012	K00013	K00014
S08	121.000000	2010.839966	221.0	4104.500000	5246.000000	183.000000	7449.839844	6168.839844	6386.339844
S08	70.000000	3880.669922	64.0	3758.500000	6821.000000	1.000000	10281.669922	6992.669922	7375.169922
S08	155.000000	2125.669922	72.0	5275.000000	6638.750000	123.000000	9043.419922	7181.419922	7499.419922
S08	214.660004	3895.159912	57.0	5396.569824	8367.660156	2.000000	12078.820312	8575.820312	9077.820312
S09	2271.340088	765.340027	81.0	6644.120117	4271.640137	172.979996	5271.140137	5174.140137	5393.459961

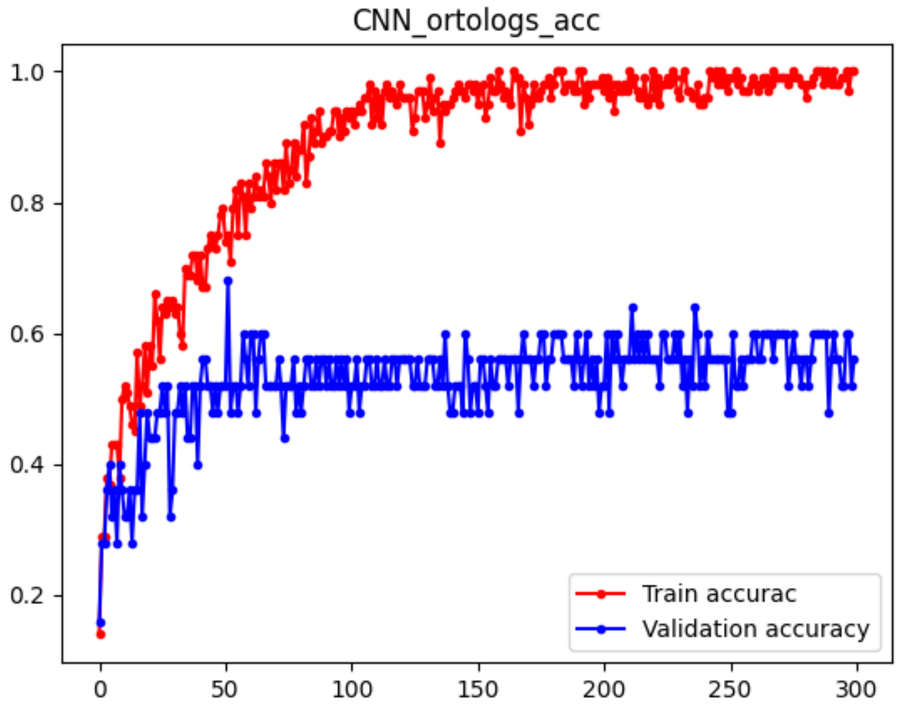
Convolution

Best results:

Normalisation, no epsylion, dropout 0.35, learning rate - 0.0001



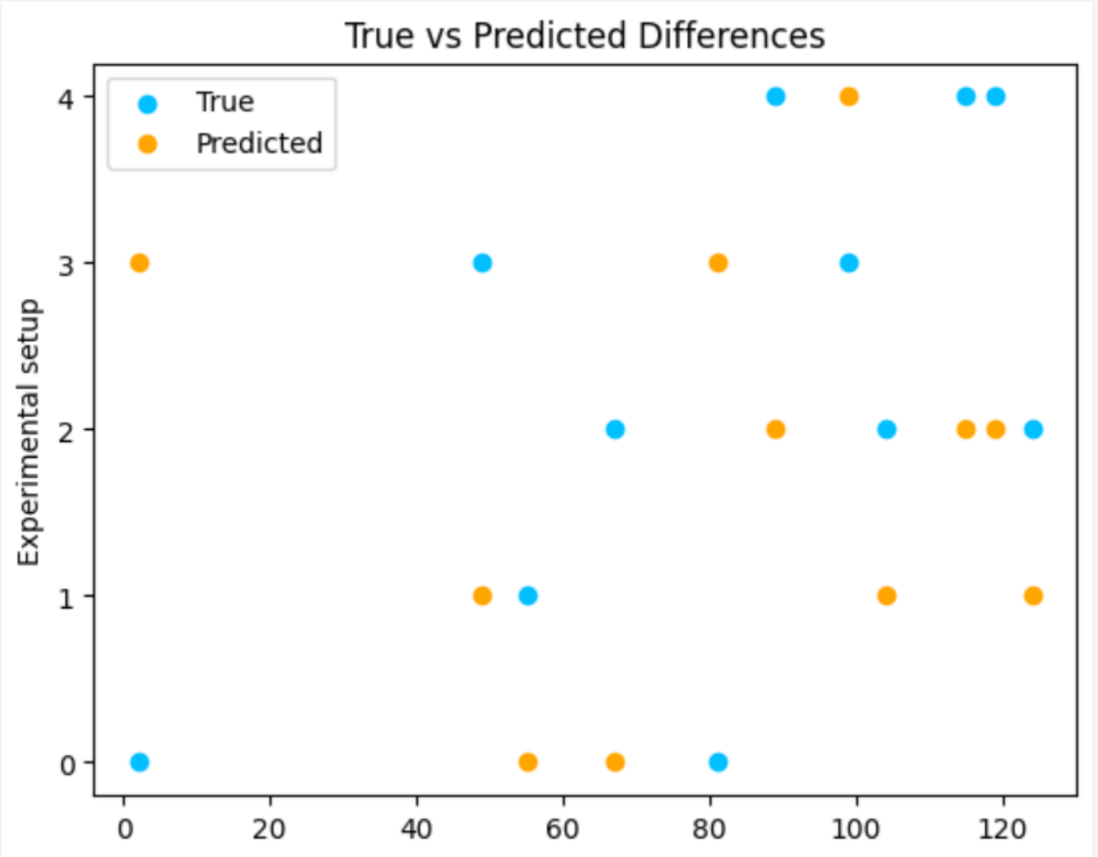

```
Layer (type)
=====================
conv1d_6 (Conv1D)

conv1d_7 (Conv1D)

dense_10 (Dense)

dense_11 (Dense)

flatten_3 (Flatten)

dense_12 (Dense)

dense_13 (Dense)
```



True vs Predicted

For CNN Ortology samples

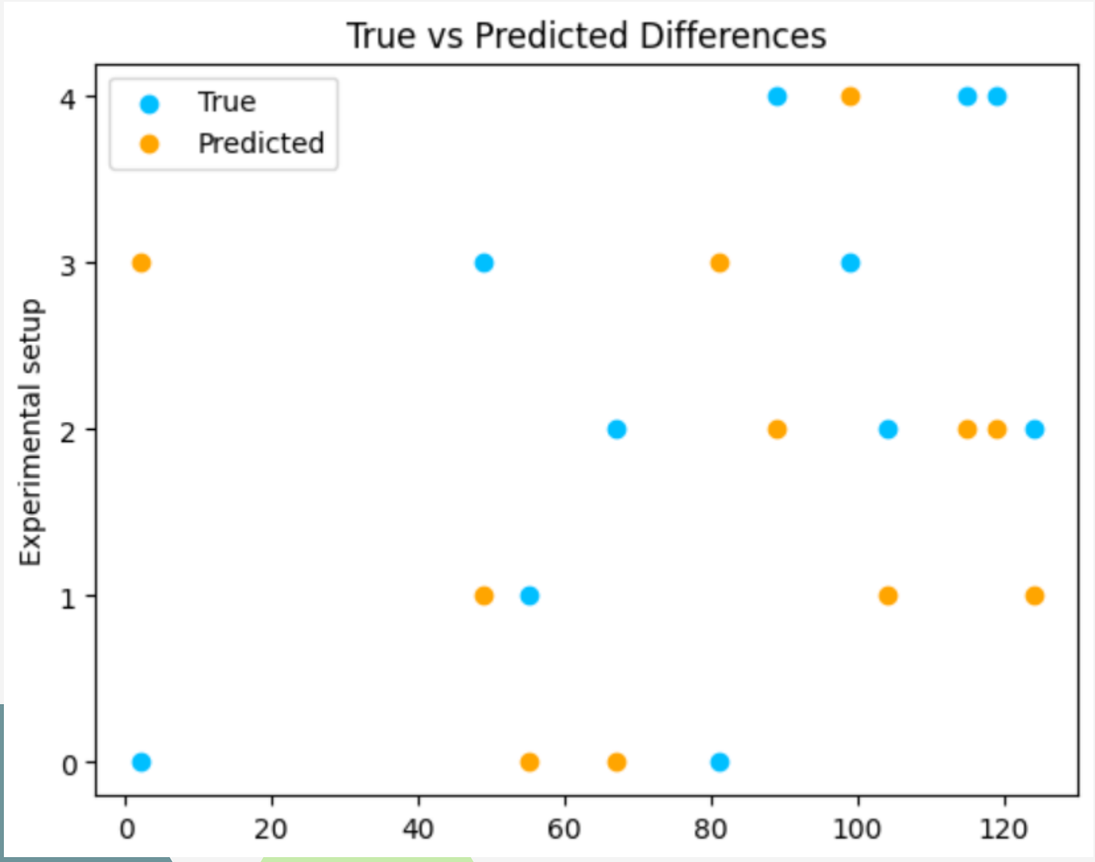
number	setup	pond numbers	water supplementation	feed suplementation
0	control	8,14,18,24,29	NO	NO
1	set 1	12,16,19,23,26	Em farma	NO
2	set 2	10,21,28,30,32	Em farma	EM
3	set 3	9,13,17,20,27	EM	NO
4	set 4	11,15,22,25,31	EM	EM

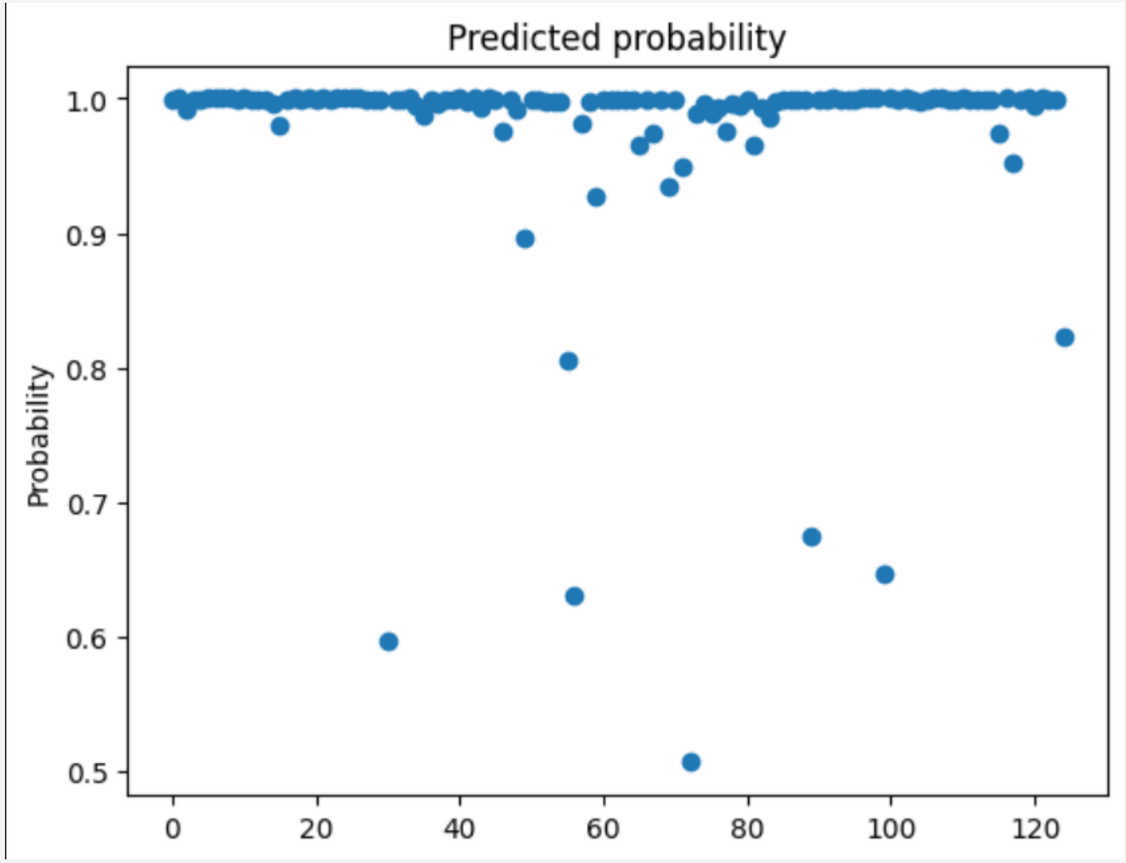


Prediction probability

For CNN Ortology samples

```
array([[9.99607027e-01, 3.17926606e-04, 1.48739928e-05, 4.54042129e-05,
        1.47383544e-05],
       [9.99982834e-01, 1.68249899e-05, 4.00250030e-07, 2.97555598e-08,
        1.47072754e-08],
```





Conclusions

- Normalisation worked better for convolution but it worsened the results for dense networks
- Vectorized class (y) input helped results
- Not much difference between using and not using epsylion
- General problem is lack of samples (only 125)
- Accuracy of 0.6 seemed to be a threshold
- Common problem was overfitting
- Best CNN was for orthology samples
- Is relu good activation function ?