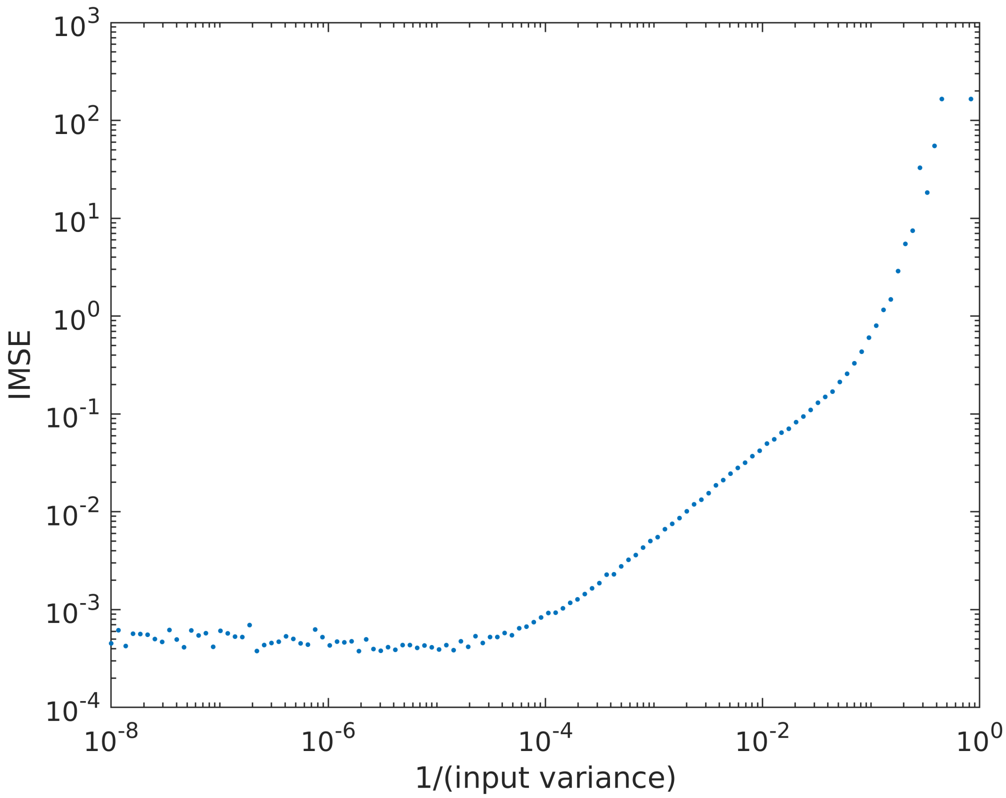
Curve fitting stability

The goal is to verify that our curve fitting is consistent when the total number of neurons increases, and that the fit recapitulates facts that are true by construction.

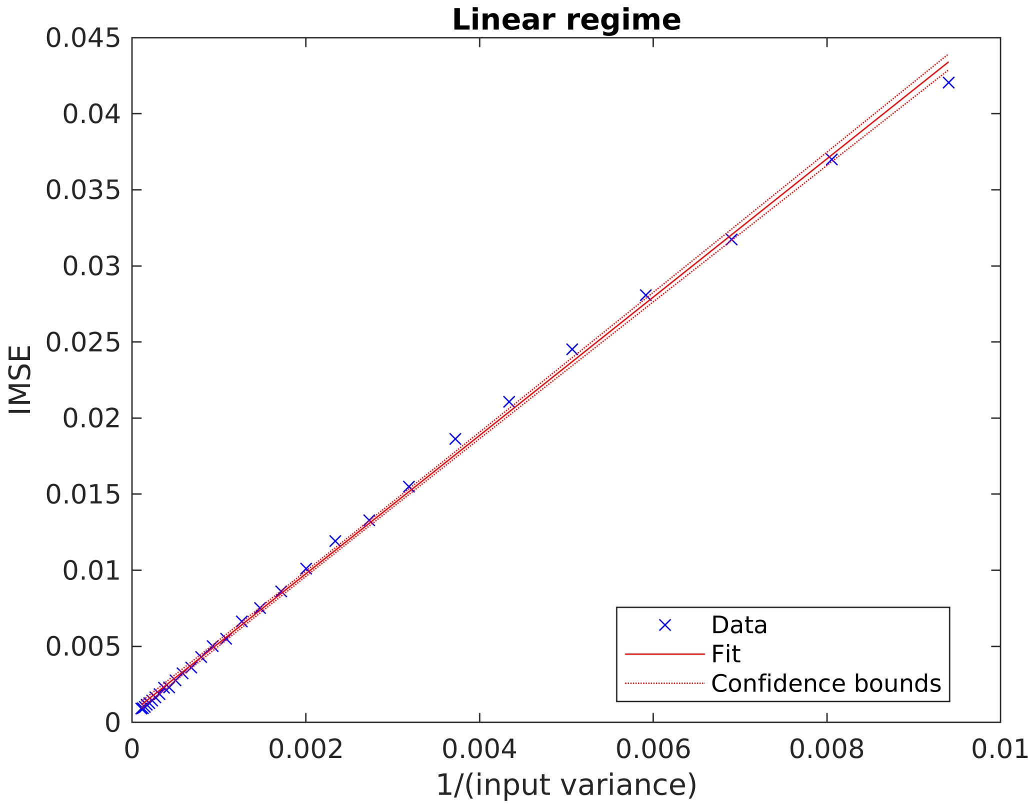
Validating the IMSE metric

The simulated datasets have one parameter that we will vary freely, the common input noise to all the neurons. In the limit of infinite neurons, the decoding accuracy will be exactly the accuracy of the common input noise. We can simulate that scenario by having one neuron that only has input noise but no output noise. The accuracy of the input noise should match the IMSE accuracy.



We can see that there is a linear correspondence between 1e-4 and 0.01 inverse input variance. These values correspond to:

```
low_IIV_bound = 1.0000e-04
high_IIV_bound = 0.0100
low_IMSE_bound = 8.6669e-04
high_IMSE_bound = 0.0459
```



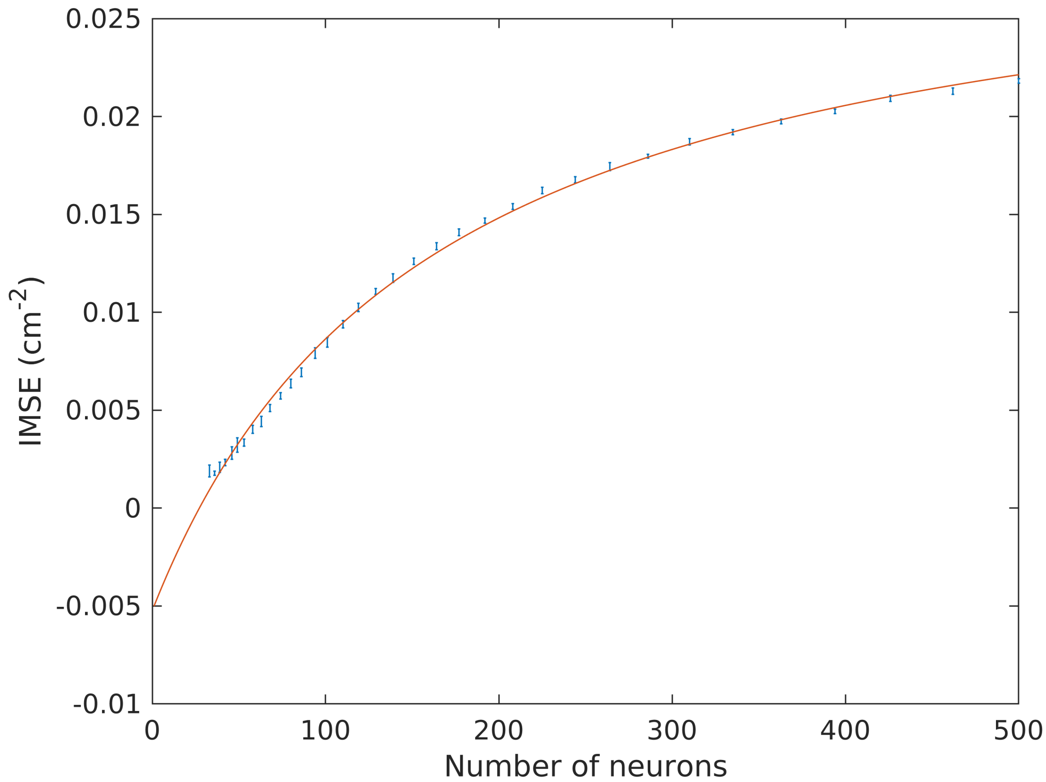

```
mdl =
Linear regression model:
    y ~ 1 + x1

Estimated Coefficients:
                   Estimate        SE         tStat       pValue
                   _________    __________    ______    __________

    (Intercept)    0.0006812    0.00011161    6.1032    1.3872e-06
    x1                 4.545      0.033534    135.53    5.3503e-41


Number of observations: 30, Error degrees of freedom: 28
Root Mean Squared Error: 0.000464
R-squared: 0.998,  Adjusted R-Squared: 0.998
F-statistic vs. constant model: 1.84e+04, p-value = 5.35e-41

to_imse_const = 4.5450
```

The relationship is nearly proportional, with a constant of ~4.5. We will only fit curves to IMSE values that are within the linear regime.

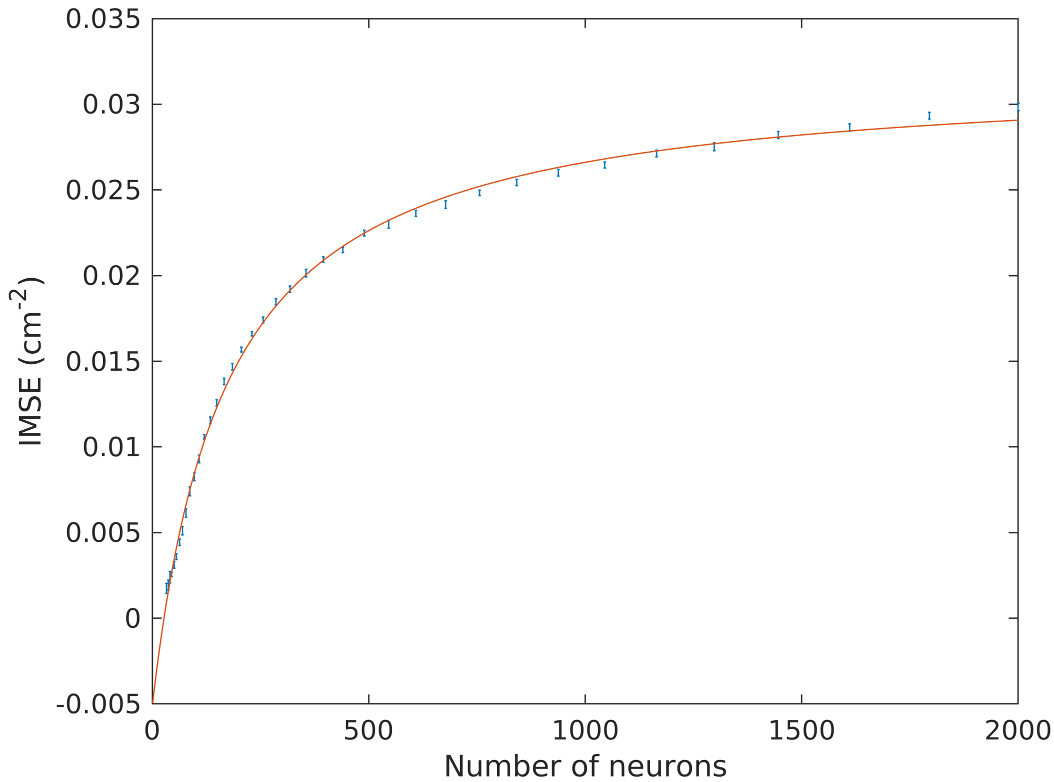
Fit stability with changing maximum ensemble size

We would now like to construct simulated data with input noise of a standard deviation of 10cm, and to be able to recapitulate this property from the decoding curves using 500 (low) and 2000 (high) neurons.

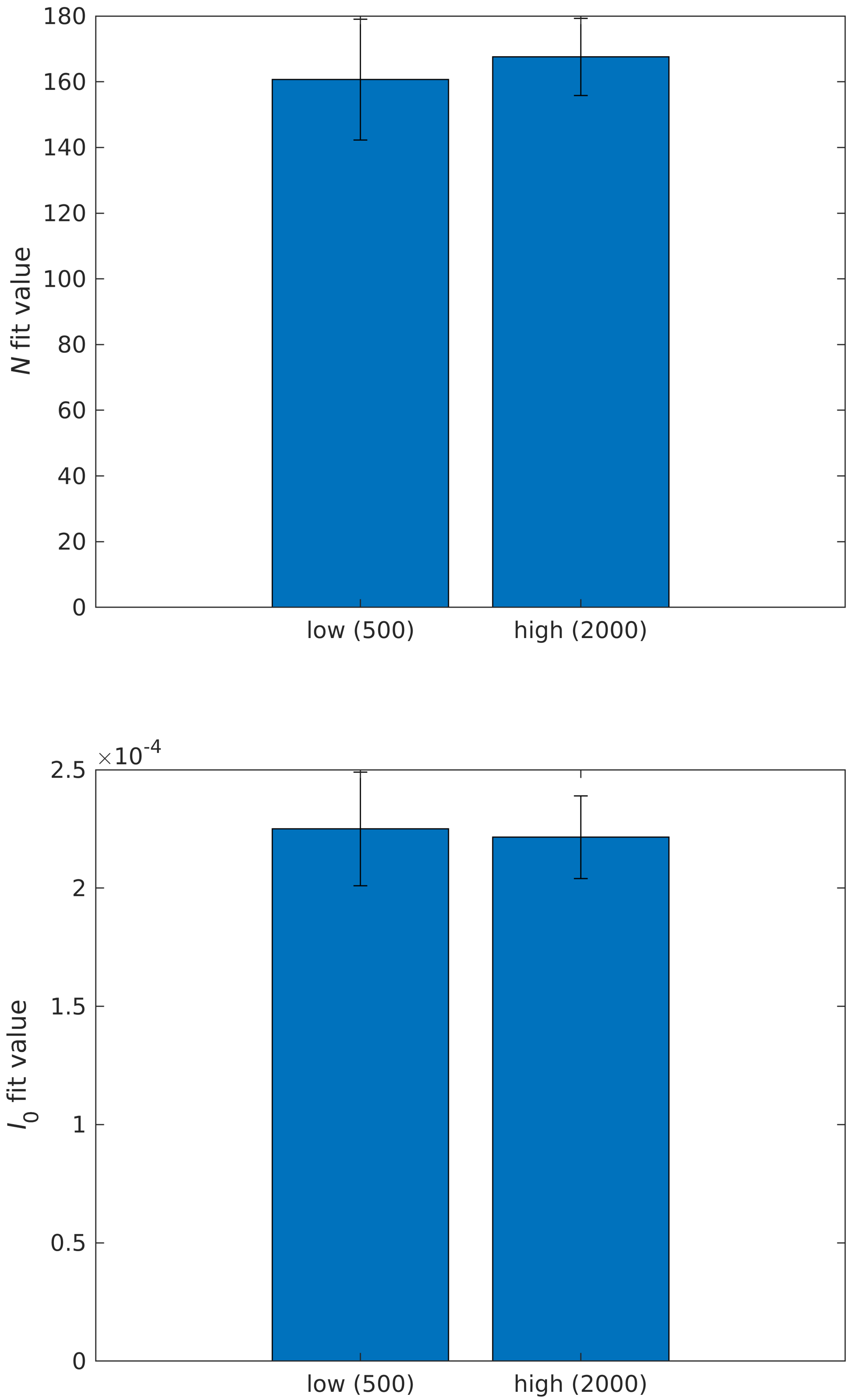
```
Warning: Removed some values.

Warning: Removed some values.
```



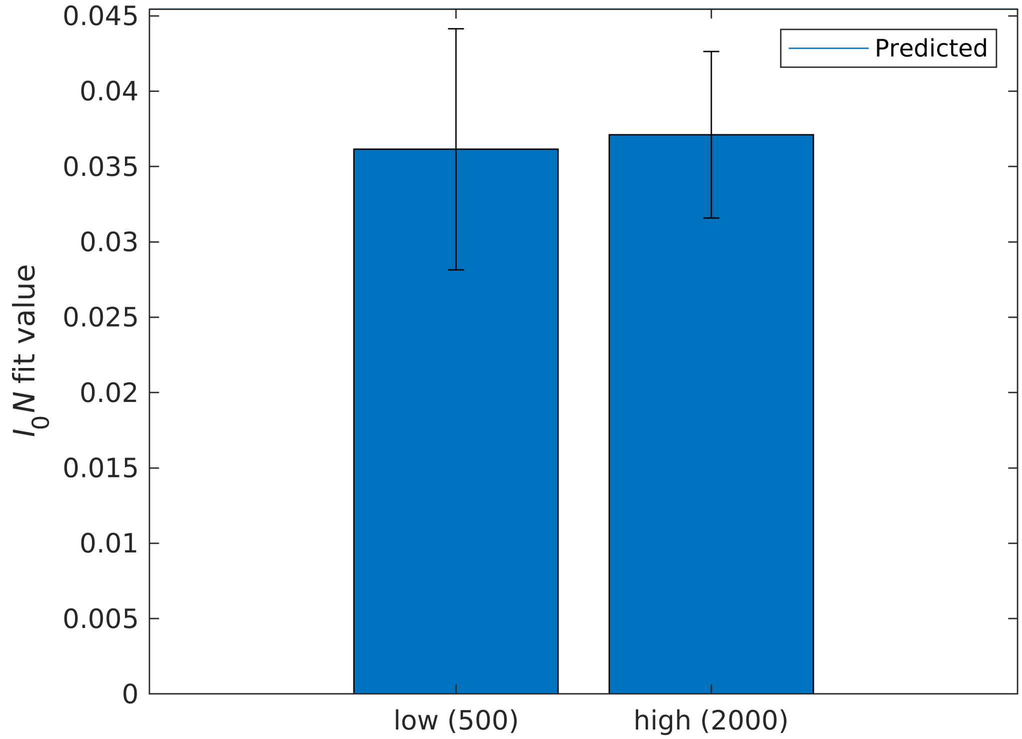

```
fr_low =
     General model:
     fr_low(n) = I_0*n / (1 + n/N) + c
     Coefficients (with 95% confidence bounds):
       I_0 =    0.000225  (0.000201, 0.000249)
       N =       160.7  (142.3, 179.1)
       c =   -0.005227  (-0.006025, -0.004428)
```




```
fr_high =
     General model:
     fr_high(n) = I_0*n / (1 + n/N) + c
     Coefficients (with 95% confidence bounds):
       I_0 =   0.0002215  (0.000204, 0.000239)
       N =       167.6  (155.8, 179.3)
       c =   -0.005174  (-0.005887, -0.004461)
```



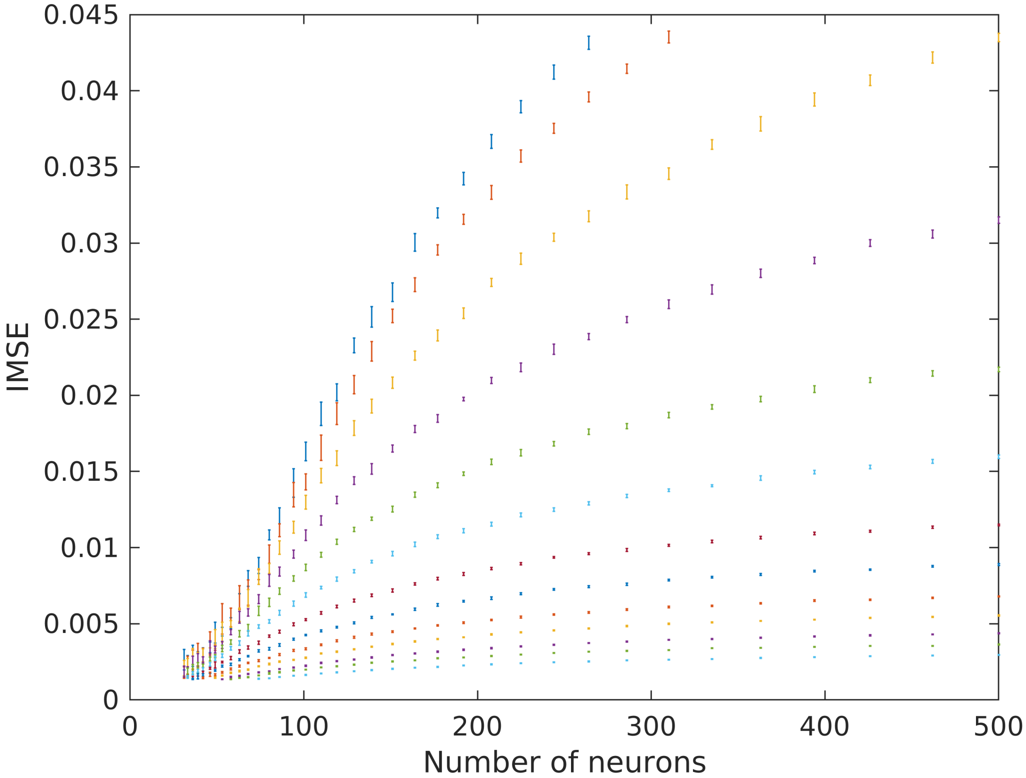
180
160
140
120
100
80
60
40
20
0
N fit value
low (500)
high (2000)
×10^-4
2.5
2
1.5
1
0.5
0
I0 fit value
low (500)
high (2000)

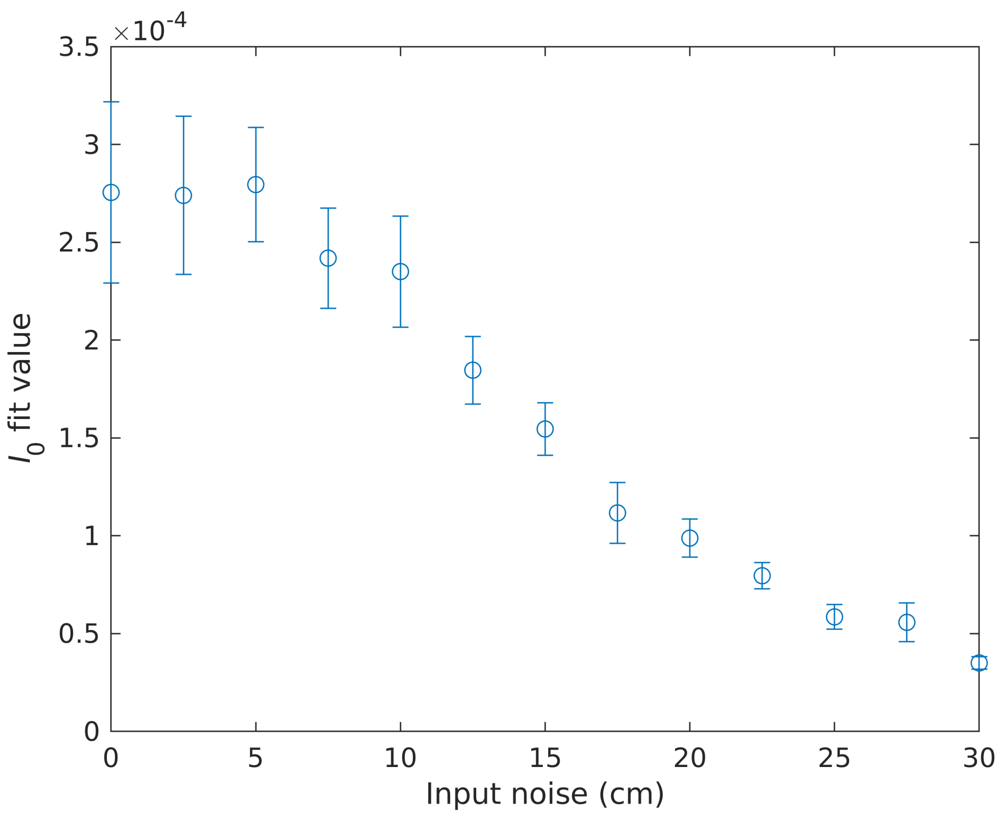


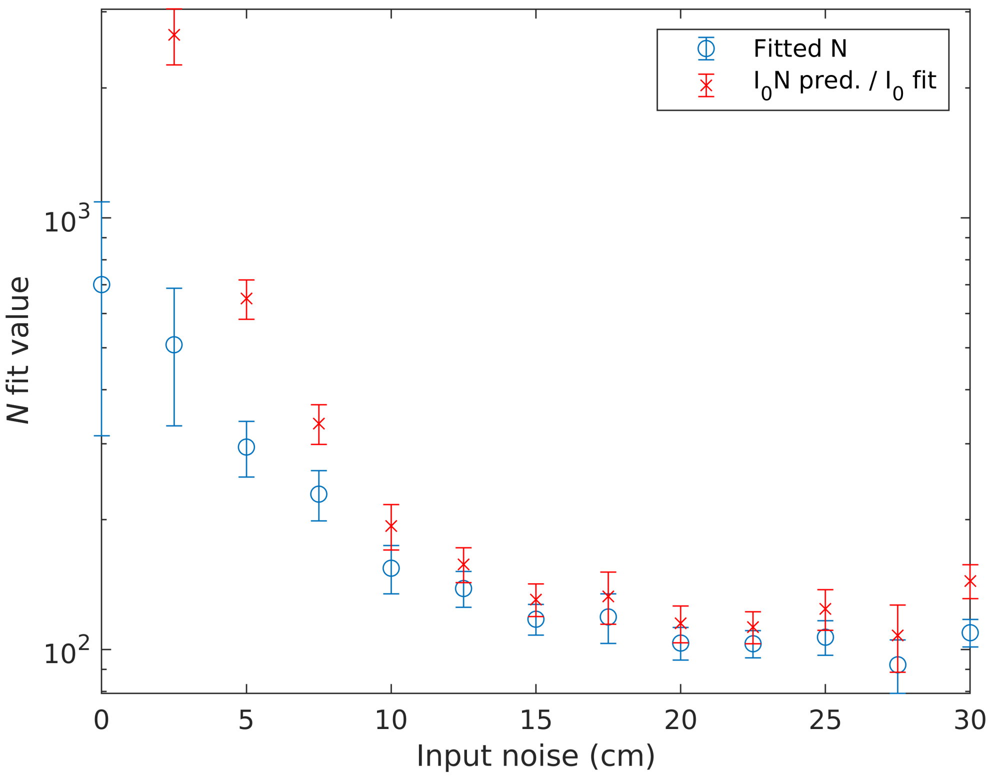
Varying the input noise

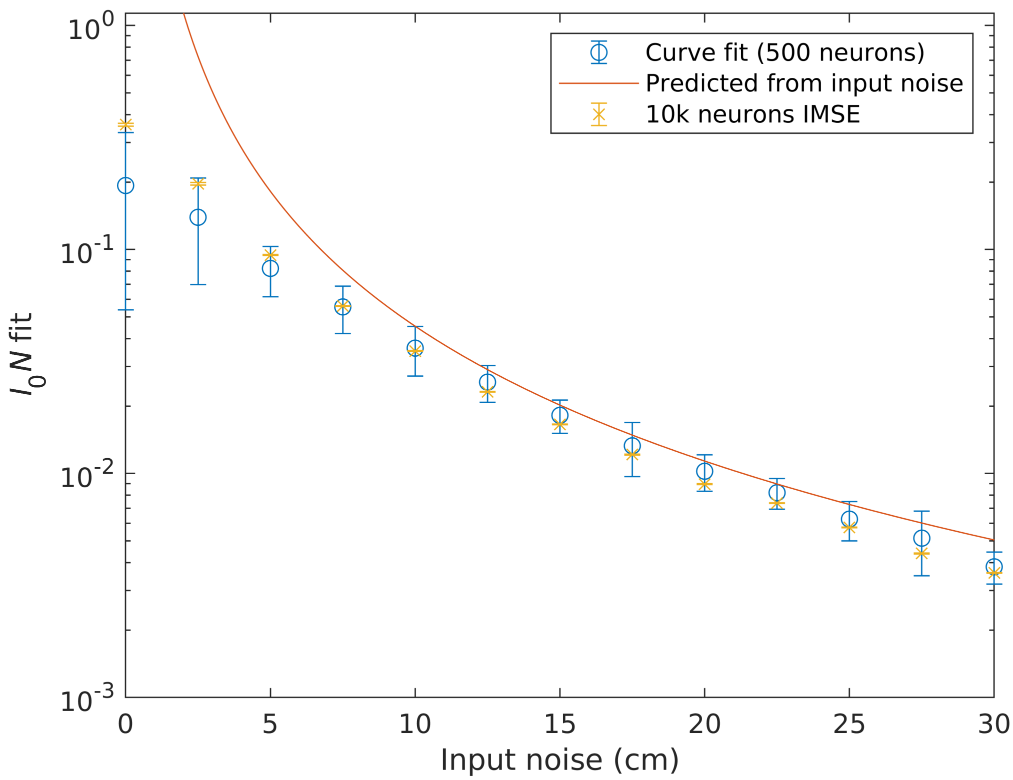
The next check is to see whether the curve fitting is able to characterize a range of input noise values.

```
Warning: Removed some values.
Warning: Removed some values.
Warning: Removed some values.
Warning: Removed some values.
Warning: Removed some values.
Warning: Removed some values.
Warning: Removed some values.
Warning: Removed some values.
Warning: Removed some values.
Warning: Removed some values.
Warning: Removed some values.
Warning: Removed some values.
Warning: Removed some values.
```


IMSE
Number of neurons


$\times 10^{-4}$
I_0 fit value
Input noise (cm)


Fitted N
I_0N pred. / I_0 fit
N fit value
10^3
10^2
0
5
10
15
20
25
30
Input noise (cm)


Curve fit (500 neurons)
Predicted from input noise
10k neurons IMSE
I_0N fit
10^0
10^{-1}
10^{-2}
10^{-3}
0
5
10
15
20
25
30
Input noise (cm)