**Parametric Linear Regression using deep learning with concrete dropout for uncertainty estimation**

In order to reliably estimate the uncertainty of the prediction in a neural net we set up a little toy example of a parametric model for linear regression. In this case the least-square solution can be derived analytically.

With a fixed vector (support vector) calculated different point clouds y

10000 random samples of and in [-5; 5] und in [0; 0.5] is used to train a neural net for estimating the parameters and given a point cloud .

The neural net suggests a parameter set and , which are used to calculate a model output with

The loss function is the MSE loss

A neural net with 3 hidden layers (1024 nodes) was used to perform the regression.

According to … Dropout used in training and prediction phase can be used to draw samples form the predictive distribution and hence determine the model’s uncertainty for the prediction. The extension to concrete dropout allows the adaptation of the dropout rate per layer during the training. Therefore, no grid search for this hyperparameter needs to be done. However, for a first test we used a fixed dropout rate of 0.2 in the hidden layers and 1e-5 in the output layer.

For the linear regression task, the least square solution can be derived analytically. Defining

Then the least-square estimate for the parameters is derived by

With covariance matrix

By deriving the estimation of and we can regain a sample distribution of size M using the multivariate distribution in python:

*lsqm = numpy.random.multivariate\_normal(mean= , cov=, size=M)*

This finally allows for the comparision of M random dropout sample predictions of the neural network with the parameters defined by the least-squares procedure.

Figures 1 and 2 show the results of the least-square procedure (Figure 1) and the neural net (Figure 2) for 10 different examples of a testset that was not used for training of the NN. The scatter plots show the data, the skyblue line the mean and red the different variations of 50 samples.

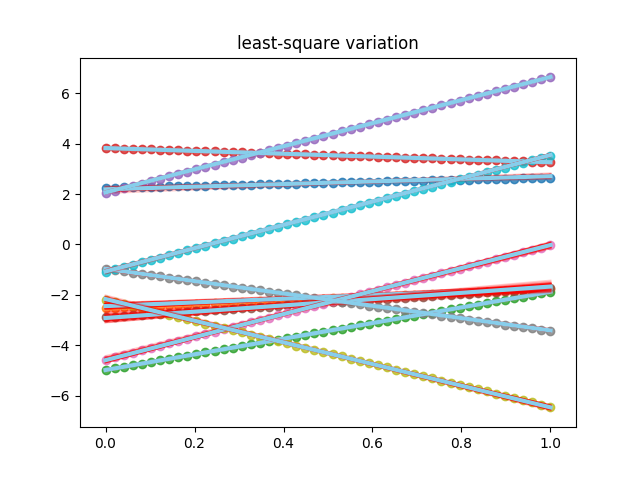


Figure Predicted mean and variation for 50 samples on 10 testlines following the analytical least-square procedure

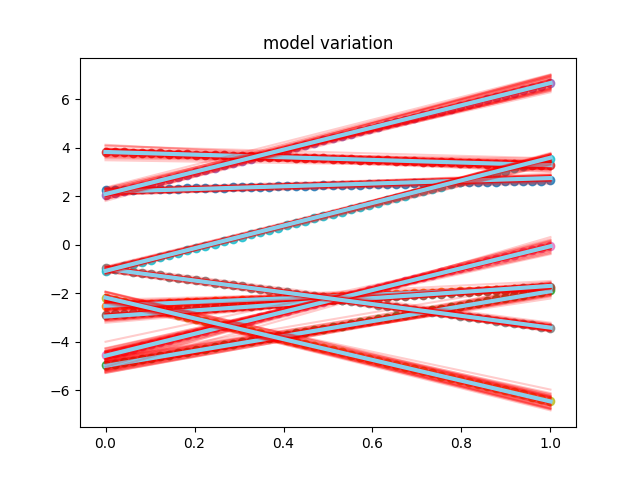


Figure Predicted mean and variation for 50 dropout samples of 10 testlines of the trained neural network with a dropout rate of 0.2

These figures indicate that both, the neural net and the least-squares approach are able to precisely predict the parameters b0 and b1. However, the variation shows some differences. This is further verified by Table 1, where the actual numbers for the 10 test scenarios are compared.

Figure Ground truth, least-square estimate and model estimate for the parameters b0 and b1 for 10 different test cases.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | b0 LSQM | b0 | b0 Model | s | b1 LSQM | b1 | b1 Model |
| 1 | 2.1879 +- 0.0451 | 2.2452 | 2.1775 +- 0.0685 | 0.1546 | 0.5209 +- 0.0689 | 0.4023 | 0.5658 +- 0.0812 |
| 2 | -2.4444 +- 0.0671 | -2.5189 | -2.5208 +- 0.0992 | 0.2418 | 0.7256 +- 0.1161 | 0.7378 | 0.7607 +- 0.1122 |
| 3 | -4.9812 +- 0.0157 | -4.9809 | -4.9985 +- 0.1728 | 0.0588 | 3.1047 +- 0.0276 | 3.0993 | 3.0758 +- 0.2032 |
| 4 | 3.8241+- 0.0041 | 3.8275 | 3.8181 +- 0.1275 | 0.0179 | -0.5625 +- 0.0073 | -0.5648 | -0.5438 +- 0.1797 |
| 5 | 2.0642 +- 0.0187 | 2.0403 | 2.0832 +- 0.1200 | 0.0734 | 4.5811 +- 0.0292 | 4.6017 | 4.5776 +- 0.2329 |
| 6 | -2.9175 +- 0.1006 | -2.8877 | -2.9656 +- 0.1100 | 0.3918 | 1.2476 +- 0.1863 | 1.1499 | 1.2646 +- 0.1244 |
| 7 | -4.5913 +- 0.0577 | -4.5662 | -4.5808 +- 0.2072 | 0.2091 | 4.5698 +- 0.0990 | 4.5262 | 4.5143 +- 0.3035 |
| 8 | -0.9649 +- 0.0260 | -0.9644 | -0.9952 +- 0.0548 | 0.1005 | -2.4458 +- 0.0423 | -2.4668 | -2.4277 +- 0.0748 |
| 9 | -2.1657 +- 0.0500 | -2.1874 | -2.1844 +- 0.1150 | 0.1757 | -4.2977 +- 0.0903 | -4.2514 | -4.2817 +- 0.2007 |
| 10 | -1.0706 +- 0.0295 | -1.0854 | -1.0831 +- 0.0793 | 0.1402 | 4.6210 +- 0.0496 | 4.6059 | 4.6663 +- 0.1608 |

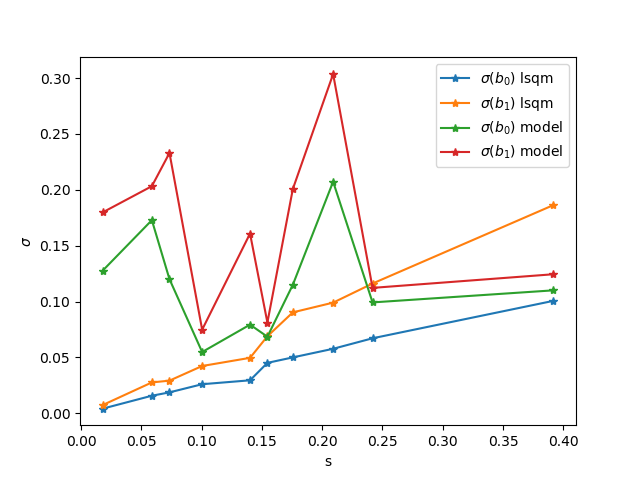
Both methods yield good estimates of the actual parameters b0 and b1. However, the standard deviation of b0 and b1 for the least-square solution depends on the noise level s present in the data, as further strengthened by Figure 4. While the standard deviation of the least-square estimates

Figure Standard deviation of the parameters b0 and b1 for the neural net and the lsqm in dependence on the noise level present in the data.

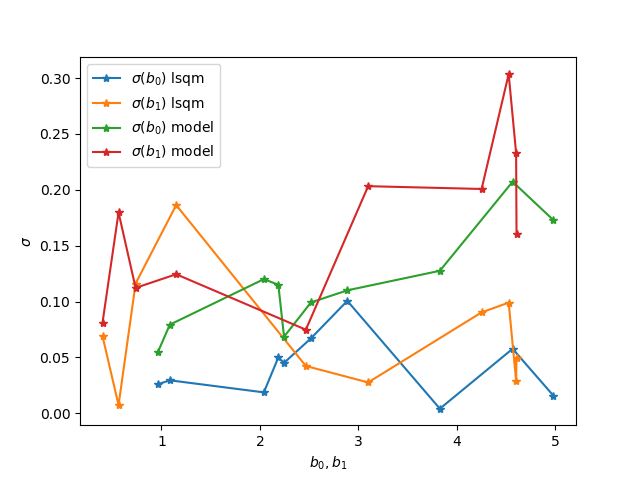
(orange and blue line) increase with increasing data noise s, the model standard deviation of the different dropout samples shows an arbitrary relation to the noise present in the data. It seems like the standard deviation of the neural net, depends more on the actual value of the estimations b0 and b1. The higher the parameter estimates, the higher the standard deviation in the dropout samples (see Figure 5).

Figure Standard deviation of the parameters b0 and b1 for the neural net and the lsqm in dependence on the values of b0 and b1 respectively.

In order to prevent the grid search for the optimal dropout rate, Gal et. al. extended their model to Concrete dropout. Within this procedure the dropout rate is adjusted while training the neural net, by deriving it’s derivative in the loss function. The dropout rate of each layer hereby influences the deviation from the model output and the ground truth as well as an additional dropout regularization term, that was added to the loss function and is basically the entropy.

With an initial dropout rate of 0.3 in every layer, the neural net was trained on the 10000 training samples for 10000 epochs. The final dropout rates are listed below:

* Input -> Hidden\_1: 6.1e-02
* Hidden\_1 -> Hidden\_2: 4.3e-03
* Hidden\_2 -> Hidden\_3: 4.5 e-02
* Hidden\_3 -> b0: 3.9e-03
* Hidden\_3 -> b1: 3.7e-03

The network tries to minimize the dropout rates in every layer.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | b0 LSQM | b0 | b0 Model | s | b1 LSQM | b1 | b1 Model |
| 1 | 2.1879 +- 0.0451 | 2.2452 | 2.1331 +- 0.0394 | 0.1546 | 0.5209 +- 0.0689 | 0.4023 | 0.6792 +- 0.0714 |
| 2 | -2.4444 +- 0.0671 | -2.5189 | -2.4304 +- 0.0337 | 0.2418 | 0.7256 +- 0.1161 | 0.7378 | 0.7113 +- 0.0578 |
| 3 | -4.9812 +- 0.0157 | -4.9809 | -4.9464 +- 0.0731 | 0.0588 | 3.1047 +- 0.0276 | 3.0993 | 3.0220 +- 0.1148 |
| 4 | 3.8241+- 0.0041 | 3.8275 | 3.7527 +- 0.0486 | 0.0179 | -0.5625 +- 0.0073 | -0.5648 | -0.3657 +- 0.0822 |
| 5 | 2.0642 +- 0.0187 | 2.0403 | 2.0541 +- 0.0653 | 0.0734 | 4.5811 +- 0.0292 | 4.6017 | 4.6883 +- 0.0832 |
| 6 | -2.9175 +- 0.1006 | -2.8877 | -2.9396 +- 0.0541 | 0.3918 | 1.2476 +- 0.1863 | 1.1499 | 1.2329 +- 0.1069 |
| 7 | -4.5913 +- 0.0577 | -4.5662 | -4.5429 +- 0.0665 | 0.2091 | 4.5698 +- 0.0990 | 4.5262 | 4.4126 +- 0.0882 |
| 8 | -0.9649 +- 0.0260 | -0.9644 | -0.9259 +- 0.0373 | 0.1005 | -2.4458 +- 0.0423 | -2.4668 | -2.4898 +- 0.0683 |
| 9 | -2.1657 +- 0.0500 | -2.1874 | -2.1280 +- 0.0576 | 0.1757 | -4.2977 +- 0.0903 | -4.2514 | -4.3796 +- 0.1055 |
| 10 | -1.0706 +- 0.0295 | -1.0854 | -1.0234 +- 0.0478 | 0.1402 | 4.6210 +- 0.0496 | 4.6059 | 4.6183 +- 0.0921 |

