Main results optimizing 10d regression functions fitting a student-t distribution

Performance of "Meta" and "ACT" models on 10,000 freshly sampled regression functions using a student-t distribution to fit the data points and derive the regression parameters.

All models consist of an RNN network (LSTM cells) with 2 layers and a hidden state size of 40 units.
Models were trained for 50-100 epochs on 10,000 newly sampled functions per epoch (using a batch size of 125). The models were evaluated each 5th epoch on a fixed set of 10,000 validation functions. We picked the best model for the final test run. Each function was optimized for 100 time steps (referred to as horizon T).
The ACT models (see below) were trained for a fixed (T=100) and a stochastic horizon (E[T]=18 and E[T]=32 optimization steps).

More specific the following models were evaluated:
(1) **metaV1:** baseline model from L2L paper (lr=5e-6)
(2) **metaV3.2:** baseline model that uses "loss weights" equal to 1/T (where T again denotes the horizon). The metaV1 model uses loss-weights equal to 1 (lr=5e-6).
(3) **metaV3.1:** baseline model that uses loss-weights from a truncated geometric distribtuion p(t|100) with shape parameter (denoted "nu") equal to {0,1, 0.3, 0.6, 0.9} (lr=5e-6).
(4) **actV2:** the **extended** baseline model that in addition to the delta parameter values of the optimizees (functions to be optimized) generates the loss-weights which are interpreted in the model-context as probabilities and referred to as *qt-values* (denoted q(t | x, T)), the probability of performing t time steps i.e to stop after t steps.
The actV2 models were trained
a) with a fixed horizon T=100 and with four different truncated geometric distributions (as priors) using again the shape parameters {0,1, 0.3, 0.6, 0.9};
b) with a stochastic horizon E[T] equall to 18 and 32 optimization steps. For each mini-batch we sampled the horizon T from a (non-parametric) distribution p(T) with shape parameter equal to 0.92 and 0.95 respectively.
The learning rate for the actV2 was set to 5e-5.

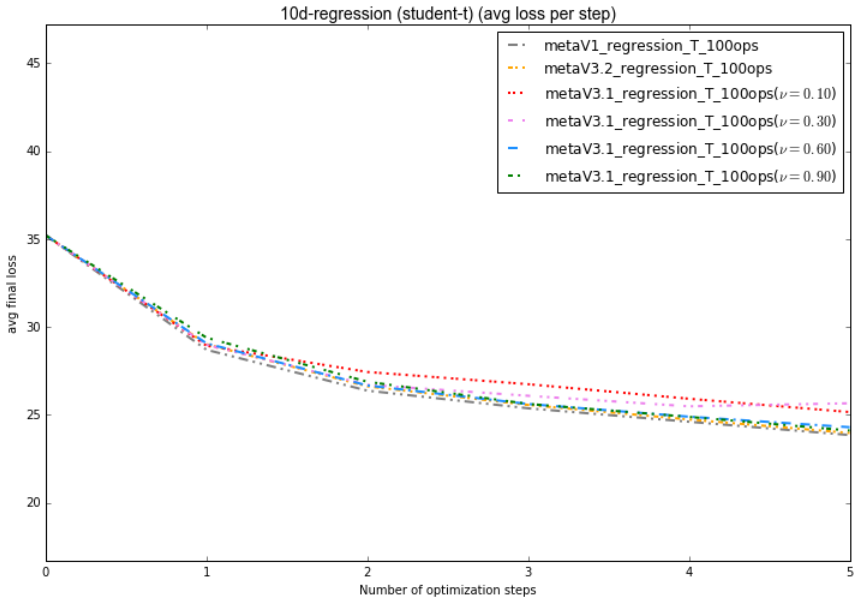
Main results

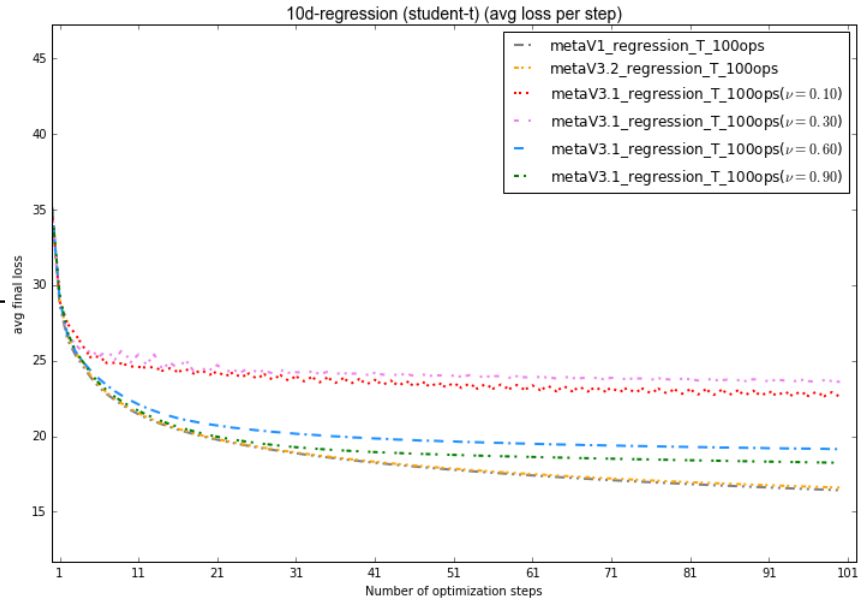
(1) **Baseline model with fixed *loss-weights*:**
- Using fixed "loss-weights" that follow a geometric distribution do not increase the performance of the baseline model (meta);
- Using *myopic* shape parameters for the geometric distribtuion (e.g. 0.1 or 0.3) detoriates the overall performance after 5-6 time steps. The effect is stronger compared to the earlier regression experiments where we fitted normal distributions;
- *Farsighted* shape parameter models *suffer* less from convergence problem but do not achieve the same performance as baseline model;
- Equally sized loss-weights per time step (e.g. 1/T) result in the same performance as the baseline model where all loss-weights have size 1;

(2) **ACTV2 models:**
- inferior performance for *myopic* priors in later time steps (converge to significant higher loss value). *Farsighted* priors converge to roughtly the same loss value as baseline models (page 7);
- inferior performance for fasighted priors in first time steps. This effect is *nearly* absent when using a stochastic horizon T during training (page 10);
- approximated posterior distributions q(t | x, T) are useful for determining a *halting step* (e.g. based on a threshold value);
- myopic priors seem to produce a qt distribution that is stronger input dependent. A stochastic training horizon helps amplifying this effect;

(3) When feeding "extra information" into the LSTM optimizer (gradients and parameters of the optimizees) the baseline model performs better in the first time steps (but does not converge to an overall lower loss value).

Compare performance meta optimizer with same model using **fixed weights** of truncated geometric distribution (instead of w_t=1)



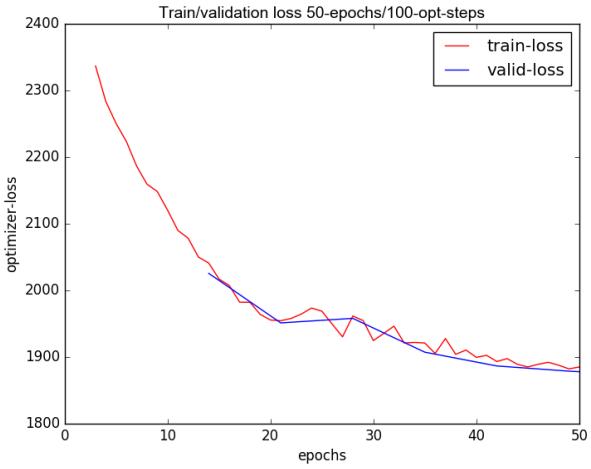


Observations:

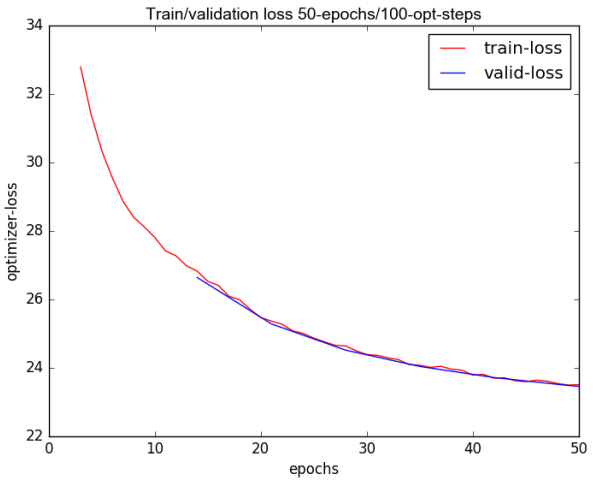
(1) Using fixed weighted losses from truncated geometric distribution does not increase overall performance of the model;
(2) MetaV1 and MetaV3.2 (loss-weights=1/T) perform roughly the same;
(3) "Aggressive" prior with shape-param=0.1 or 0.3 perform significantly worse than baseline model after 5/6 time steps;
(4) Compared to 10d regression problem (fitting normal distribution) the model performance diverges significantly after 5/6 time steps (fitting regression parameters with a student-t distribution seems to be a harder problem).

Loss during training for different meta models

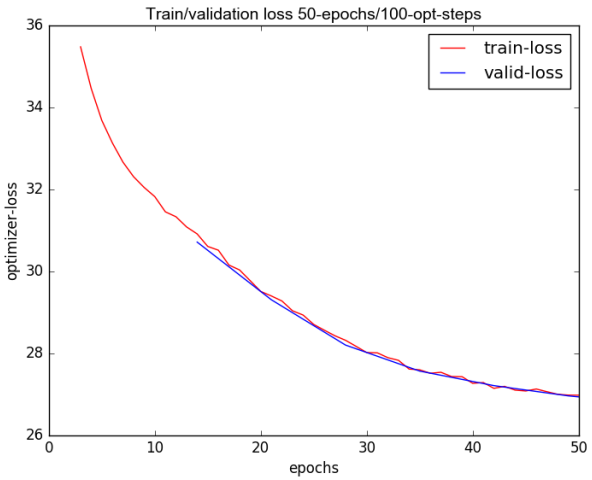
metaV1



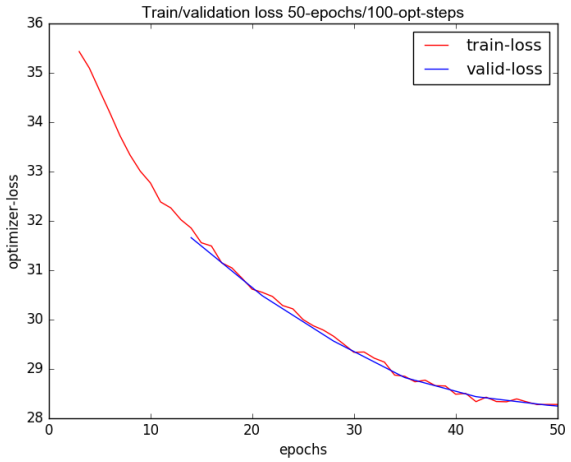
metaV3.1 (nu=0.9)



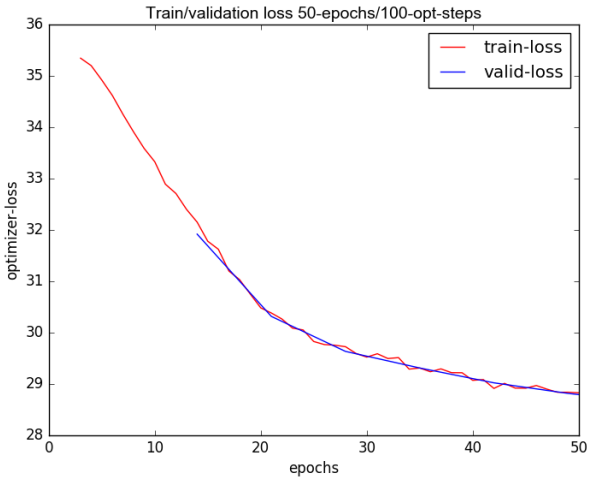
metaV3.1 (nu=0.6)



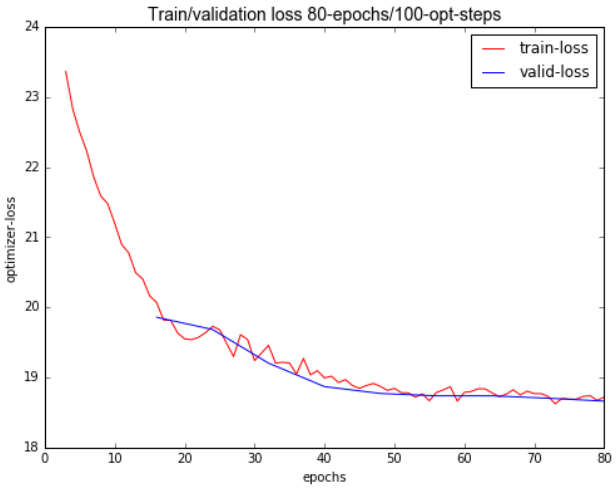
metaV3.1 (nu=0.3)



metaV3.1 (nu=0.1)



metaV3.2 (loss-weiths=1/T)



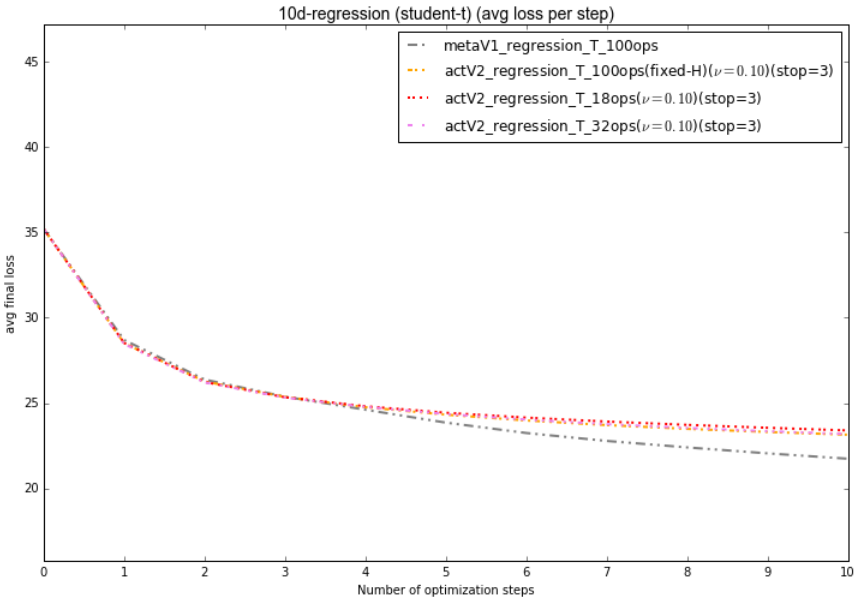
Compare performances ACTV2 model for shape pararmeter equal to **0.1** (myopic)

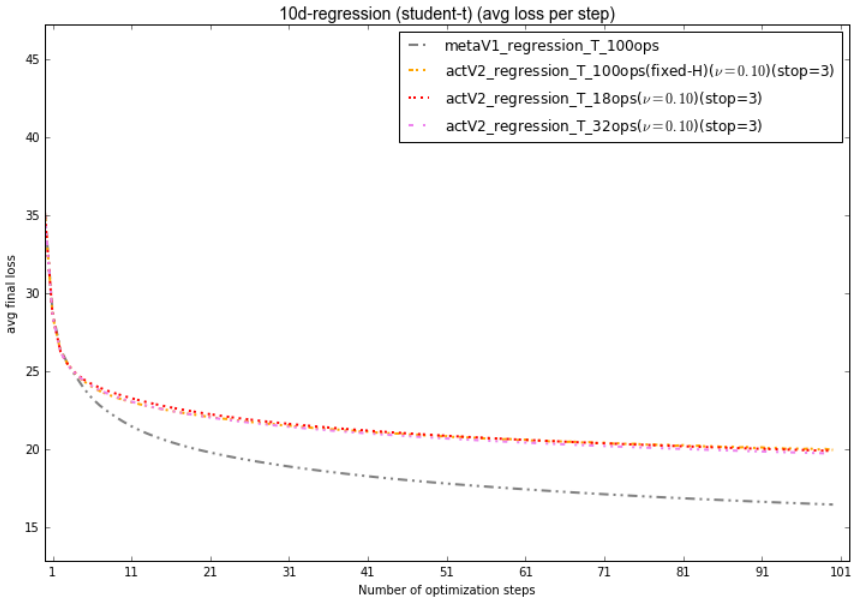
Comparing the performance of the baseline model (metaV1) with the actV2 model when trained with

a) fixed horizon T=100 and nu=**0.1**

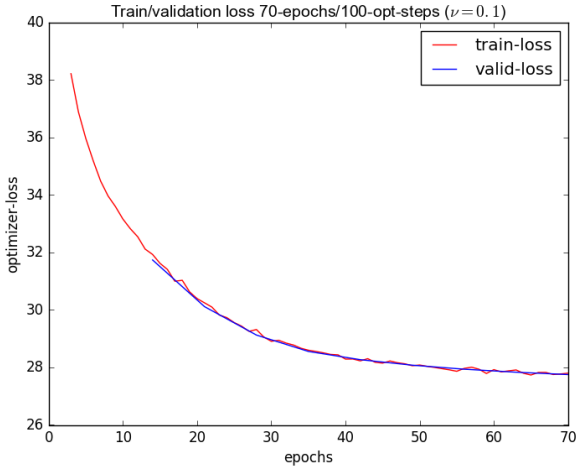
b) stochastic horizon E[T] = 18 and nu 0.1

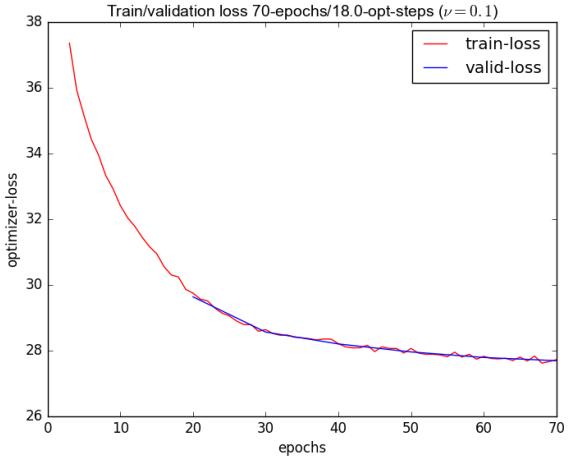
c) stochastic horizon E[T] = 32 and nu 0.1

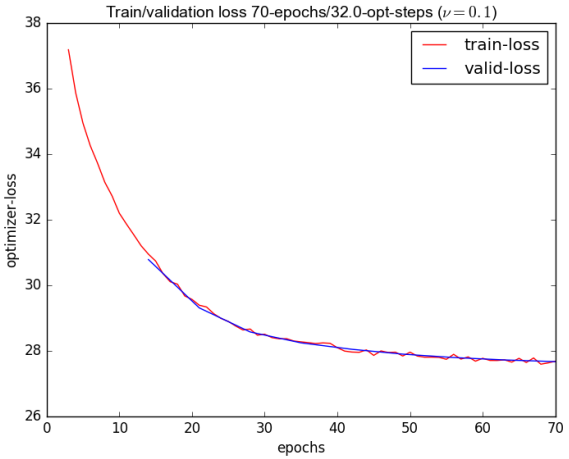




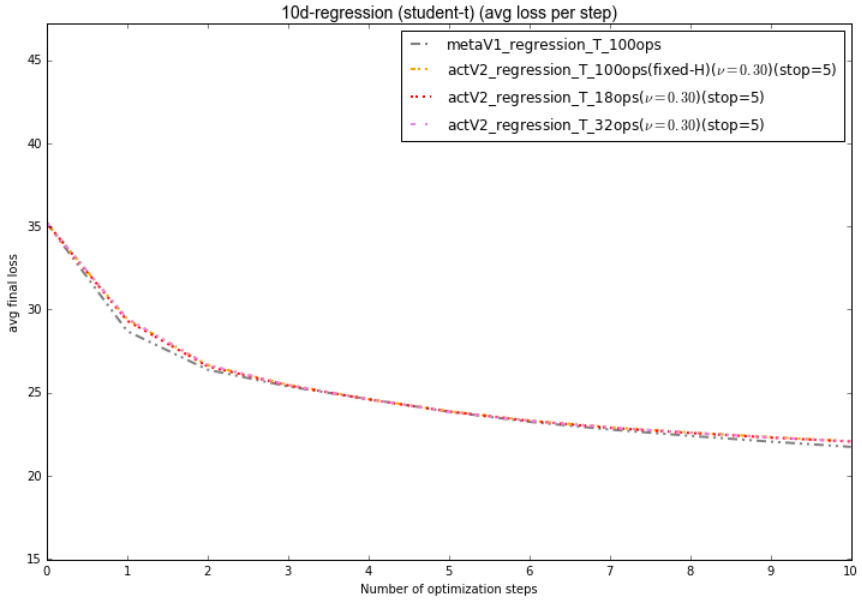
Learning curves ACTV2 models

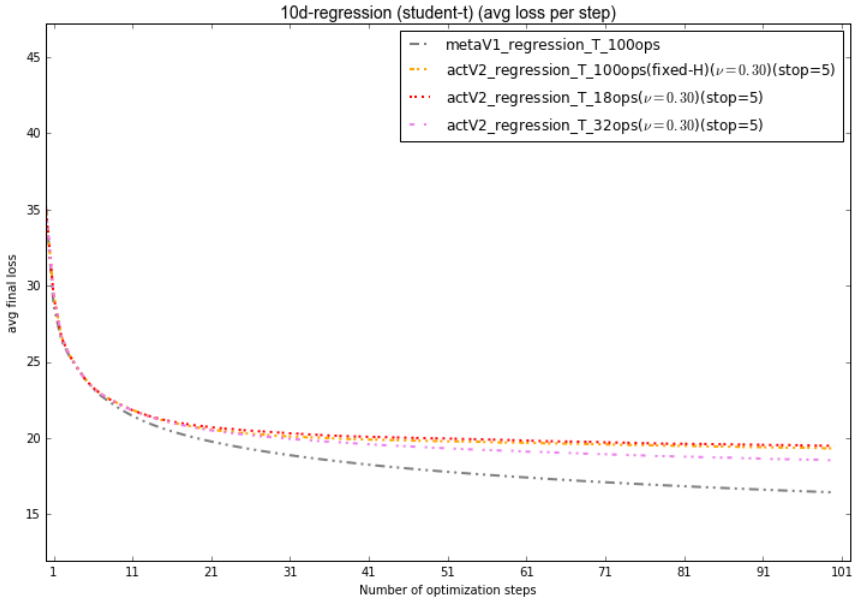




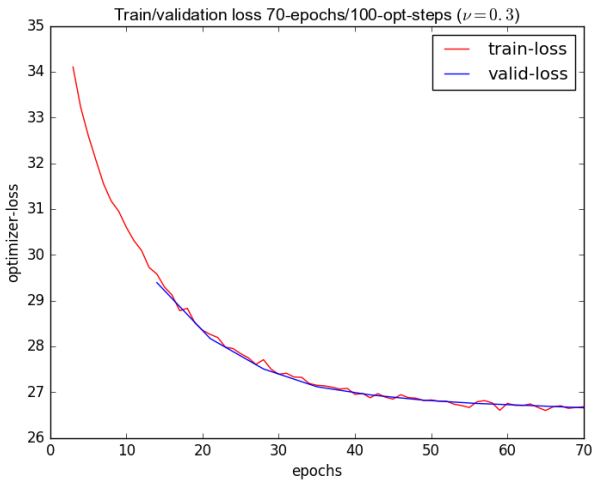


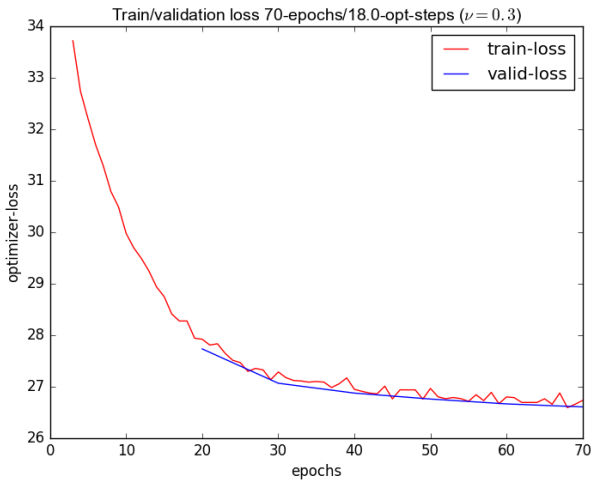
Compare performances ACTV2 model for shape pararmeter equal to **0.3** (myopic)

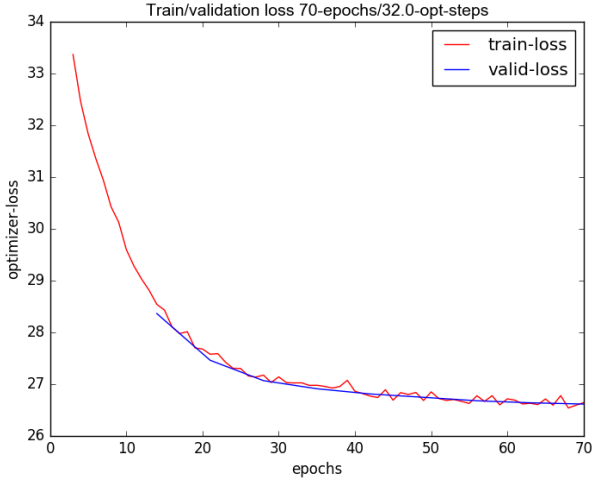




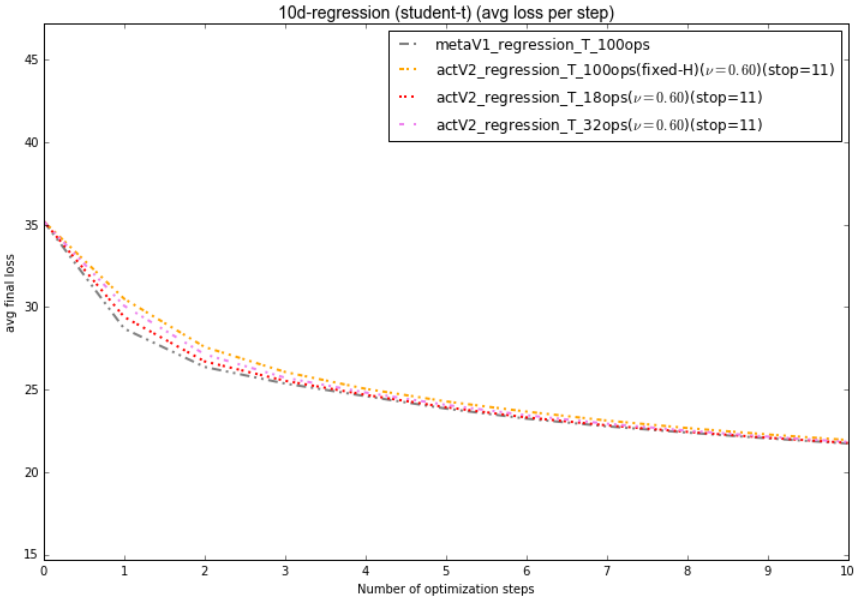
Learning curves ACTV2 models

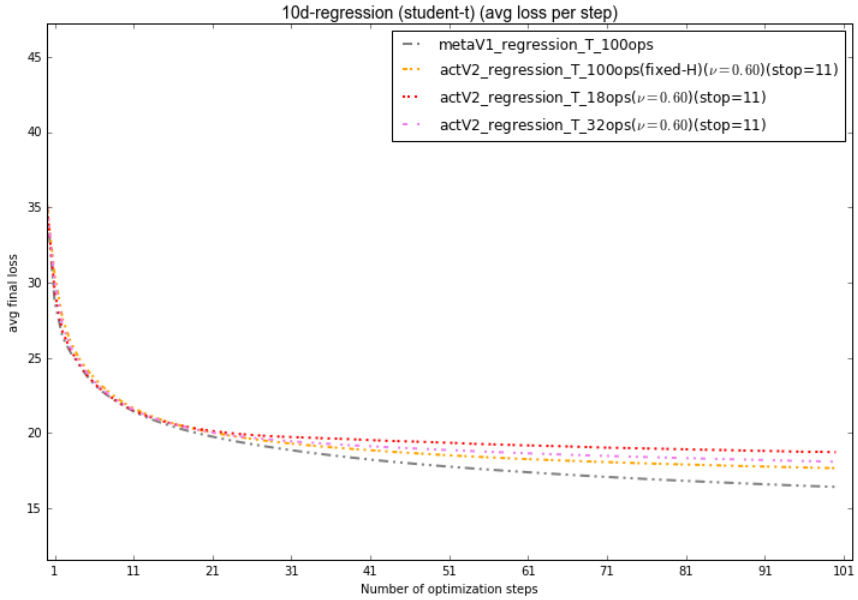




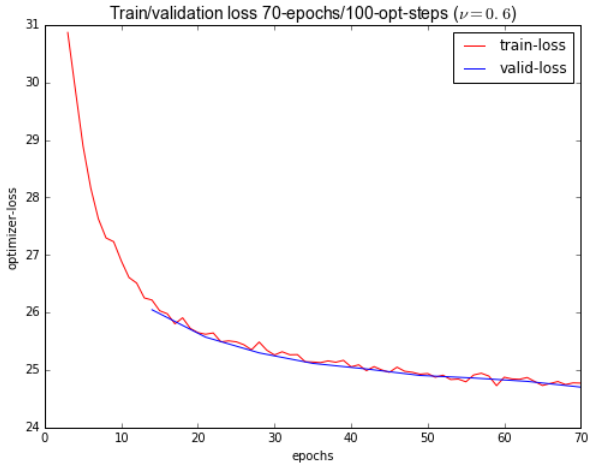


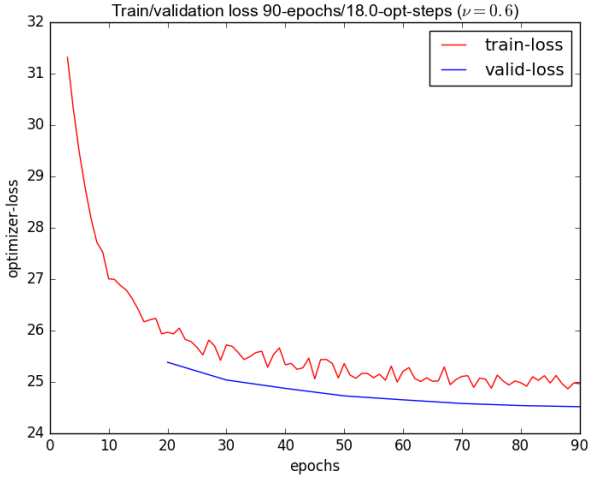
Compare performances ACTV2 model for shape pararmeter equal to **0.6**

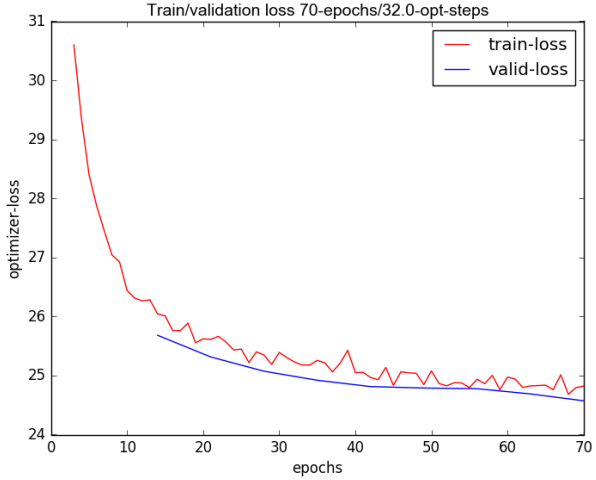




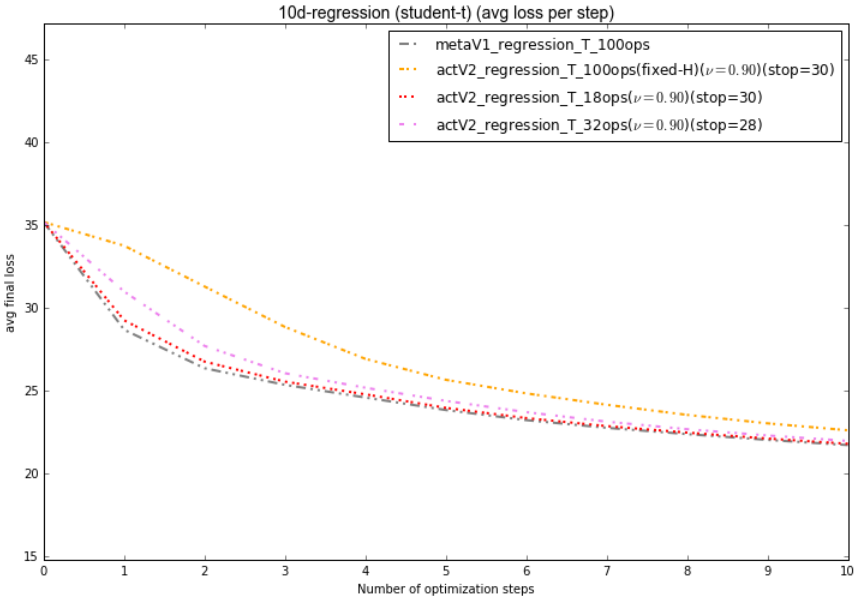
Learning curves ACTV2 models

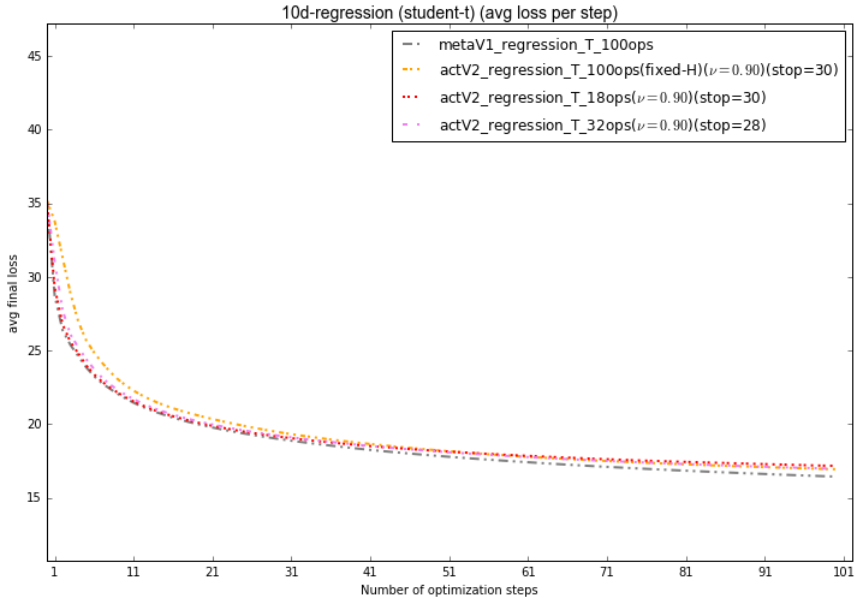




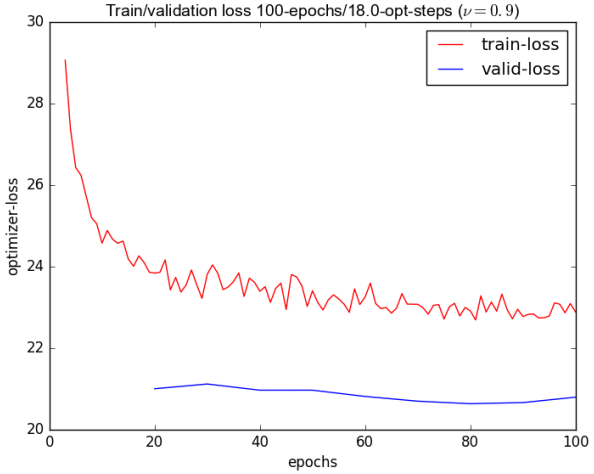


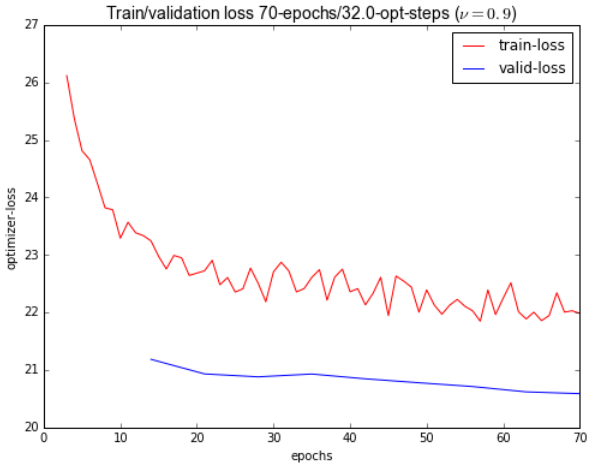
Compare performances ACTV2 model for shape pararmeter equal to **0.9**



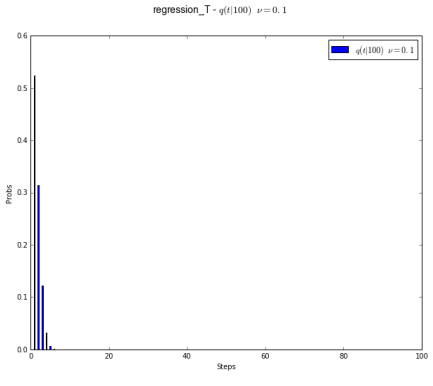


Learning curves ACTV2 models (best curves I could produce with different learning rates)

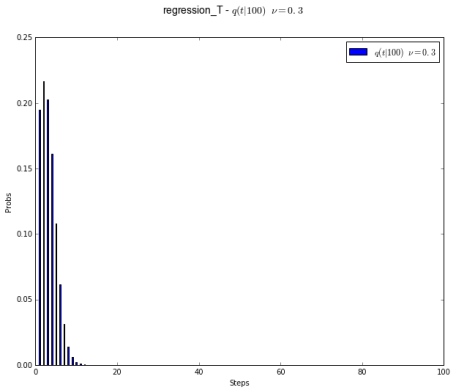




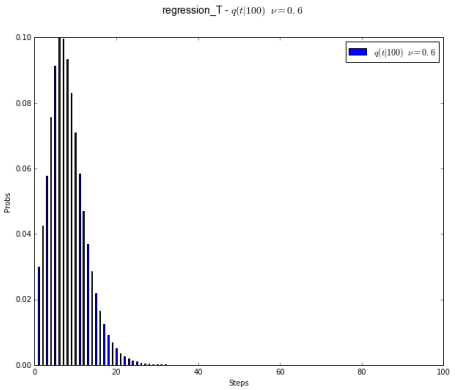
Approximated q(t|x, H) for ACTV2 models



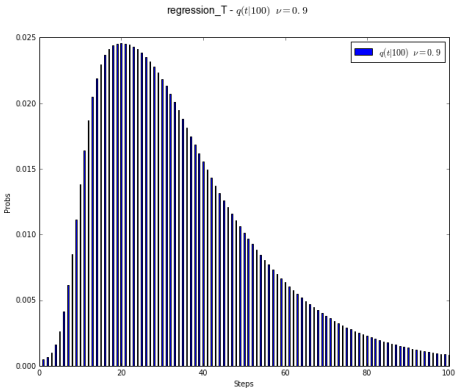
Max mode stats(N=10000): Range(1, 1) / mean=1.0 / median=1.0 / stddev=0.0



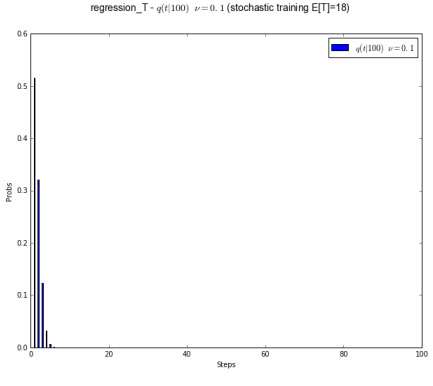
Max mode stats(N=10000): Range(1, 7) / mean=2.3 / median=2.0 / stddev=1.3



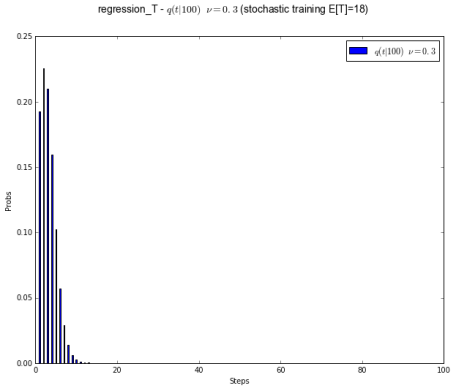
Max mode stats(N=10000): Range(1, 26) / mean=6.8 / median=7.0 / stddev=2.5



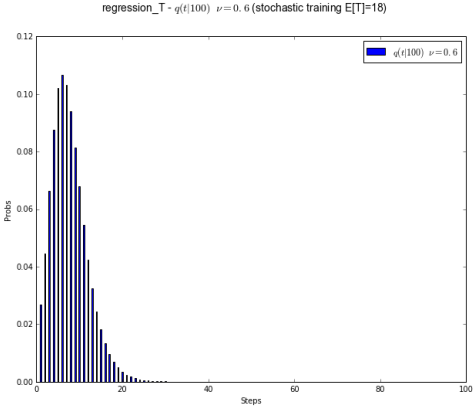
Max mode stats(N=10000): Range(6, 100) / mean=28.6 / median=24.0 / stddev=16.2



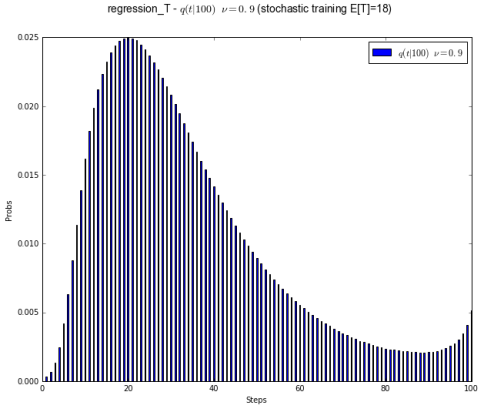
Max mode stats(N=10000): Range(1, 4) / mean=1.1 / median=1.0 / stddev=0.3



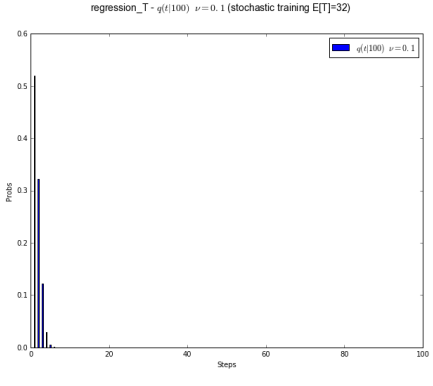
Max mode stats(N=10000): Range(1, 34) / mean=2.2 / median=2.0 / stddev=1.2



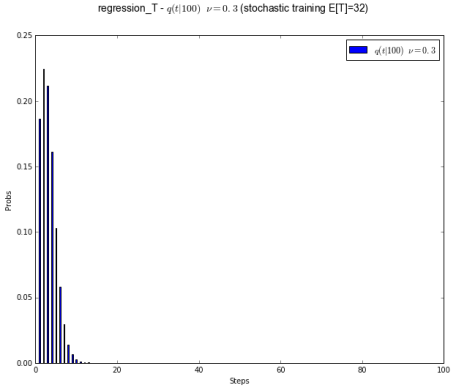
Max mode stats(N=10000): Range(2, 64) / mean=6.5 / median=6.0 / stddev=2.4



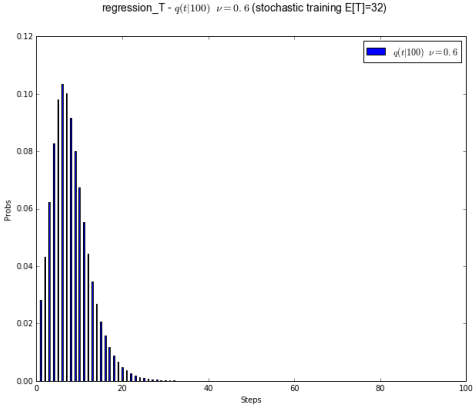
Max mode stats(N=10000): Range(7, 100) / mean=29.6 / median=23.0 / stddev=19.9



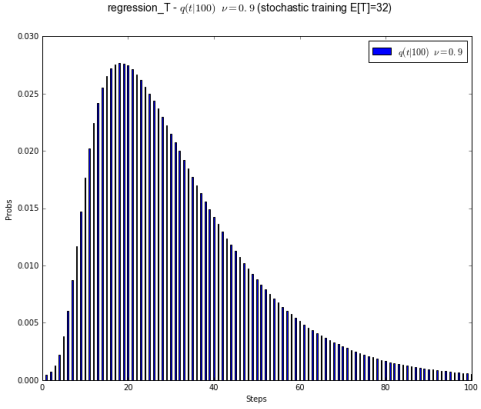
Max mode stats(N=10000): Range(1, 4) / mean=1.0 / median=1.0 / stddev=0.2



Max mode stats(N=10000): Range(1, 10) / mean=2.2 / median=2.0 / stddev=1.2



Max mode stats(N=10000): Range(2, 25) / mean=6.6 / median=6.0 / stddev=2.3

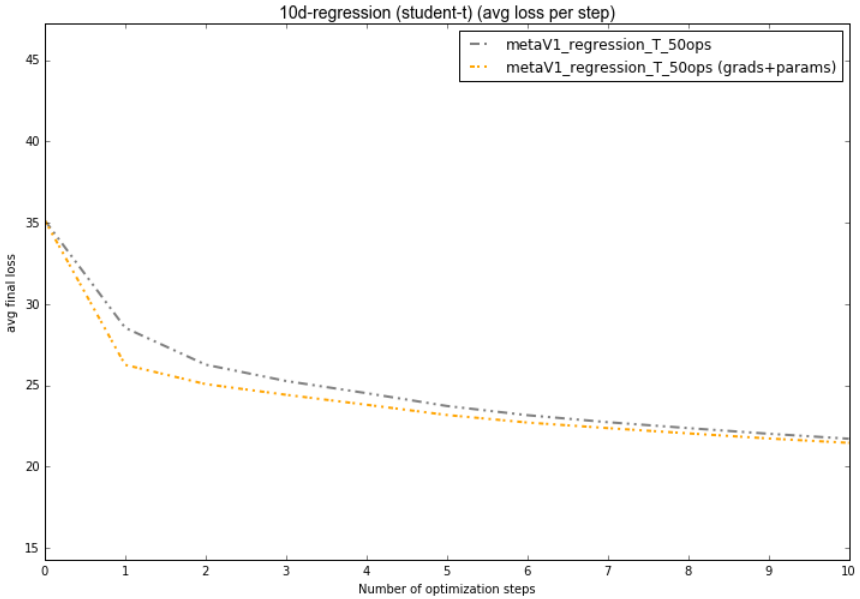


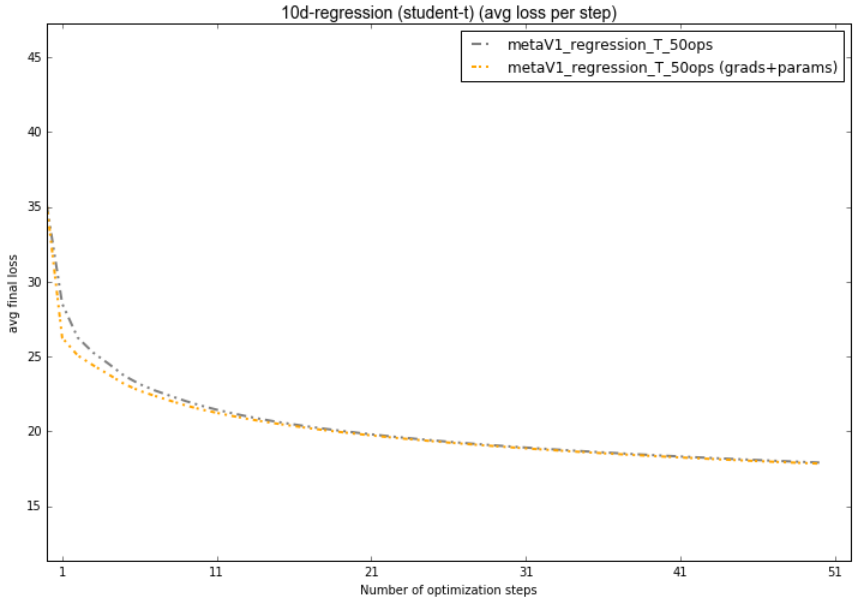
Max mode stats(N=10000): Range(6, 100) / mean=25.9 / median=21.0 / stddev=14.7

Performance meta model with additional input (gradients & parameters of optimizee)

In this experiment the optimizer (meta model) takes as input the gradients and the parameters of the optimizee as separate dimensions. Before feeding the information as input to the LSTM, the two dimensions are combined by means of a linear transformation.

It seems that this additional information helps the model to optimize the functions especially in the first step.





Comparing performace actV2 model with fixed and stochastic time horizon

Comparing the performance of 5 actV2 models with the baseline model on 10,000 test functions. All actV2 models were trained with a truncated geometric distribtution using a shape parameter of nu=0.9. Three of the 5 models were trained with a fixed time horizon {18, 32, 100}. The other 2 models were trained with a stochastic time horizon E[T]=18 respectively 32.

Take away, the stochastic training regime seems to have a siginificant performance effect in the first 20 time steps compared to a fixed time horizon. **Note**, still the baseline model (metaV1) still performance superior to all actV2 models (using loss-weights that equal 1 in all time steps).

Note all figures show the same model test-performances. Figure 1 shows the performance of the models over 100 time steps whereas the other two figures zoom into the first 20 and 10 time steps respectively.

Figure 1

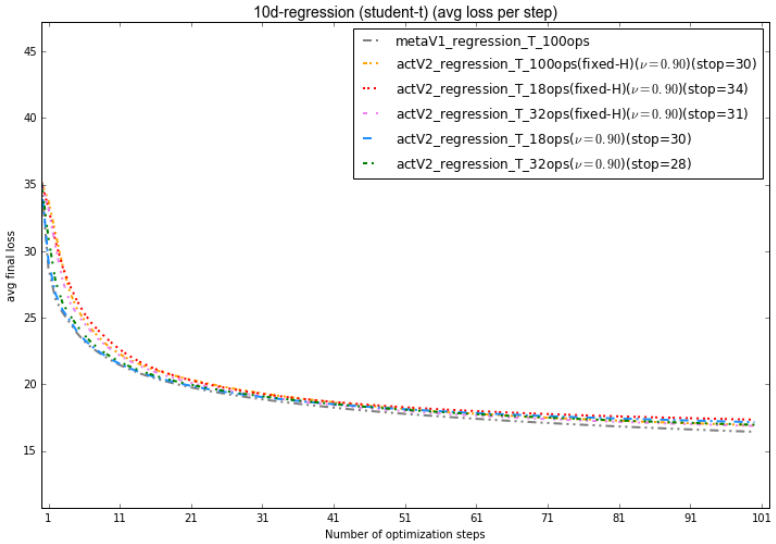


Figure 2

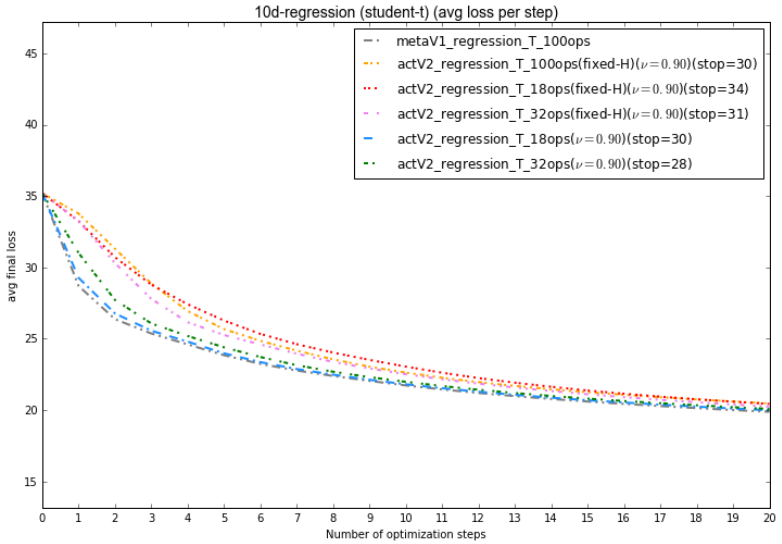
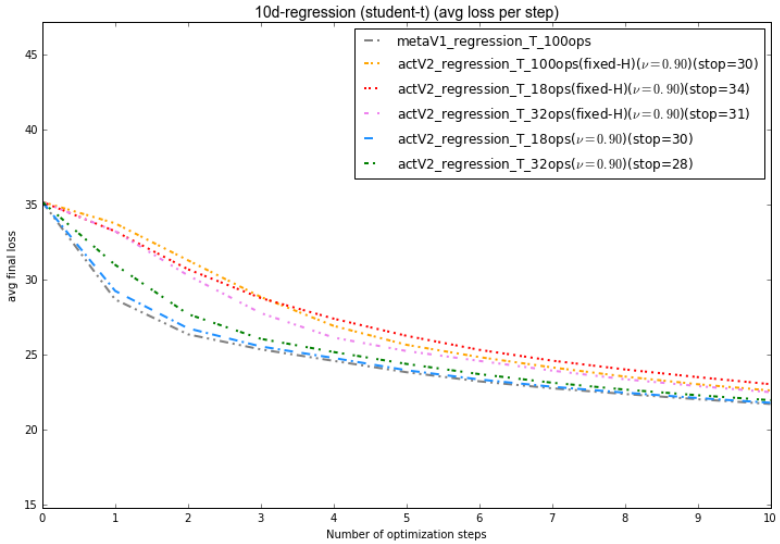


Figure 3



actV2 - distribution of *mode-step* and *stopping-step* derived from approximated q(t | x, T)

Take away: more myopic priors seem to produce a qt distribution that is more input dependent

For each of the ACTV2 models trained the following plots are shown:

(1) Figure in which the mode-step for a function is plotted against the distance between the initial (step 0) negative log-likelihood (NLL) value and the minimum NLL value for that function.

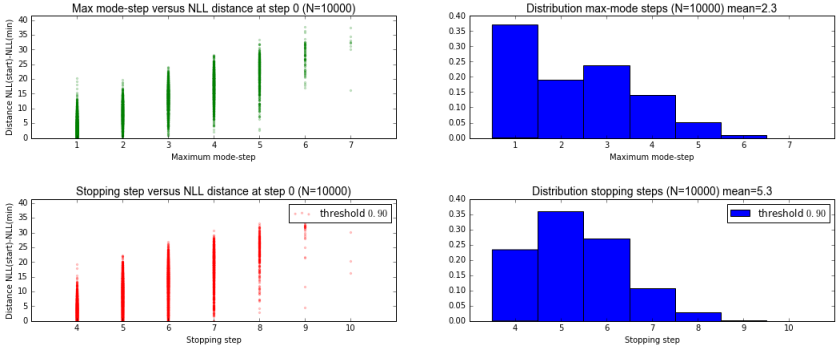
(2) Distribution of "mode steps"

(3) Scatter plot: for each function the *stopping step* was determined based on the q(t|100) cumulative distribution using a threshold of 0.9 (as indicated in the legend). The stopping step on the x-axis was plotted against the distance between the initial NLL and the minimum NLL value for a particular function to be optimized.

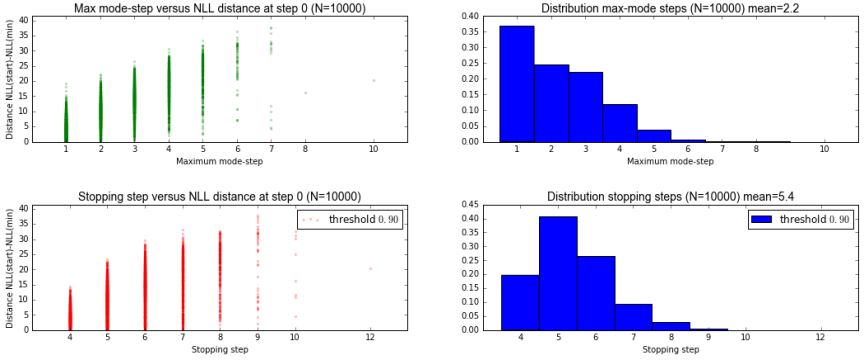
(4) Distribution of *stopping steps*

shape parameter = 0.3

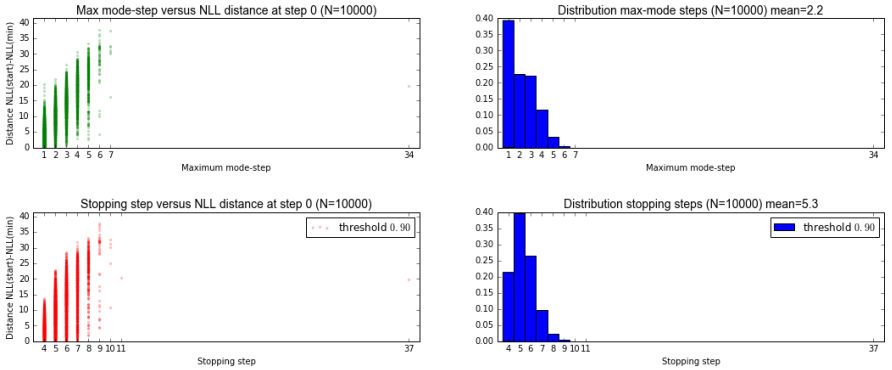
regression_T ($\nu = 0.3$) - 10000 functions



regression_T ($\nu = 0.3$) - 10000 functions (stochastic training E[T]=32)

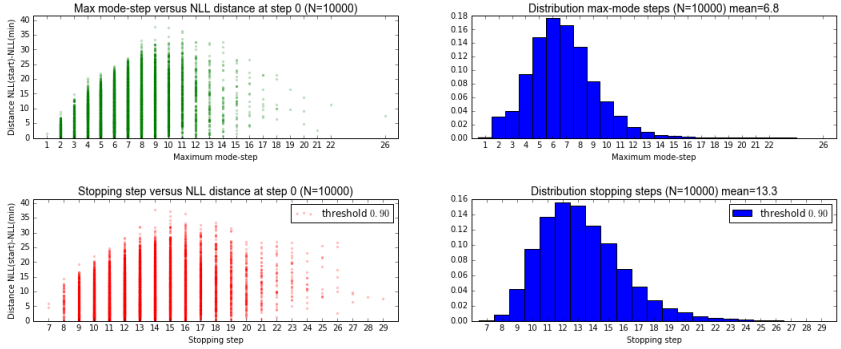


regression_T ($\nu = 0.3$) - 10000 functions (stochastic training E[T]=18)

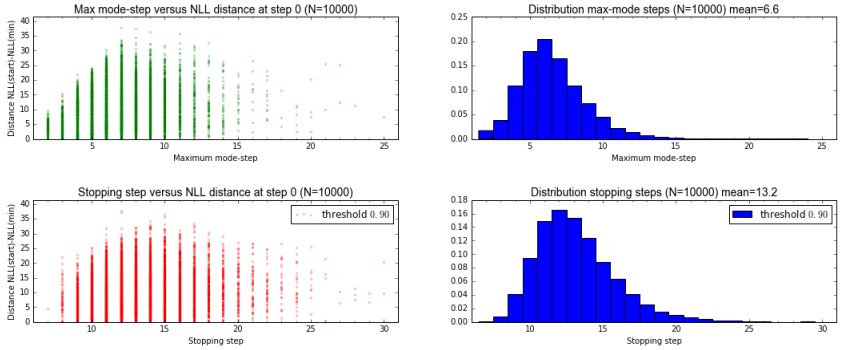


shape parameter = 0.6

regression_T ($\nu = 0.6$) - 10000 functions



regression_T ($\nu = 0.6$) - 10000 functions (stochastic training E[T]=32)



regression_T ($\nu = 0.6$) - 10000 functions (stochastic training E[T]=18)