Addressing some of the challenges when optimizing long-to-train-models

Tobias Andreasen — SigOpt, tobias@sigopt.com

SIGOPT

Abstract

*SigOpt provides an extensive set of **advanced features**, which help you, the expert, **save time** while **increasing model performance** via **optimization**.*

*In order to do so, we have created a **framework** that allows you, the expert, to **customize** the optimization process in order to **target** your specific problems.*

Agenda

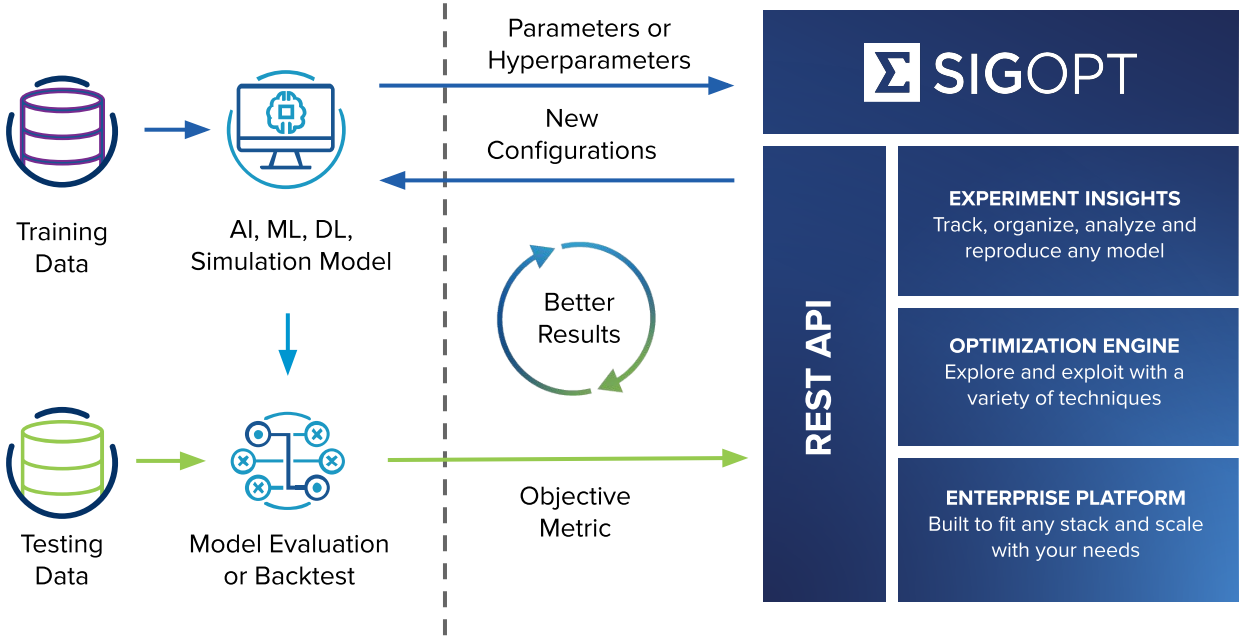
1. Framing up the problem
2. How to address these problems
    a. Addressing long training times
    b. Utilize compute resources
    c. Using previous results
3. Conclusions / Next steps



Accelerate and amplify the
impact of modelers everywhere

SigOpt: API-enabled parameter tuning

SigOpt delivers iterative, automated optimization



SigOpt. Confidential.

Framing up the problem

Framing up the problem



Framing up the problem



Framing up the problem

In this particular situation, we are making the following assumptions

"*We want to continue to re-optimize some model at some regular cadence, under the assumption that the underlying data is changed slightly before re-optimizing the model*"

Addressing long training times: Our model takes a long time to train

Utilize compute resources: We do have available compute resources

Using previous results: We know the best results from the previous optimizations

Addressing long training times

Addressing long training times

Our model takes a long time to train

Issue: every hyperparameter configuration takes a long time to evaluate

Addressing long training times

Our model takes a long time to train

Issue: every hyperparameter configuration takes a long time to evaluate

Solution: make the search as efficient as possible in order to evaluate as few samples as possible

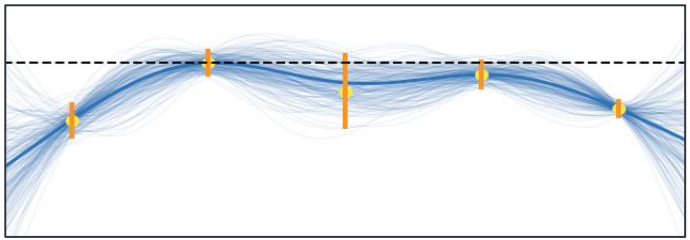
Addressing long training times

Our model takes a long time to train

Issue: every hyperparameter configuration takes a long time to evaluate

Solution: make the search as efficient as possible in order to evaluate as few samples as possible

Build a statistical model

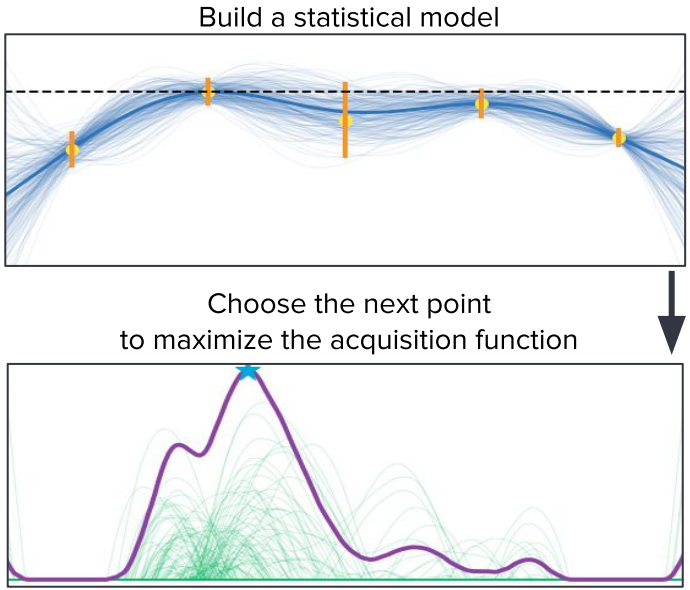


Addressing long training times

Our model takes a long time to train

Issue: every hyperparameter configuration takes a long time to evaluate

Solution: make the search as efficient as possible in order to evaluate as few samples as possible

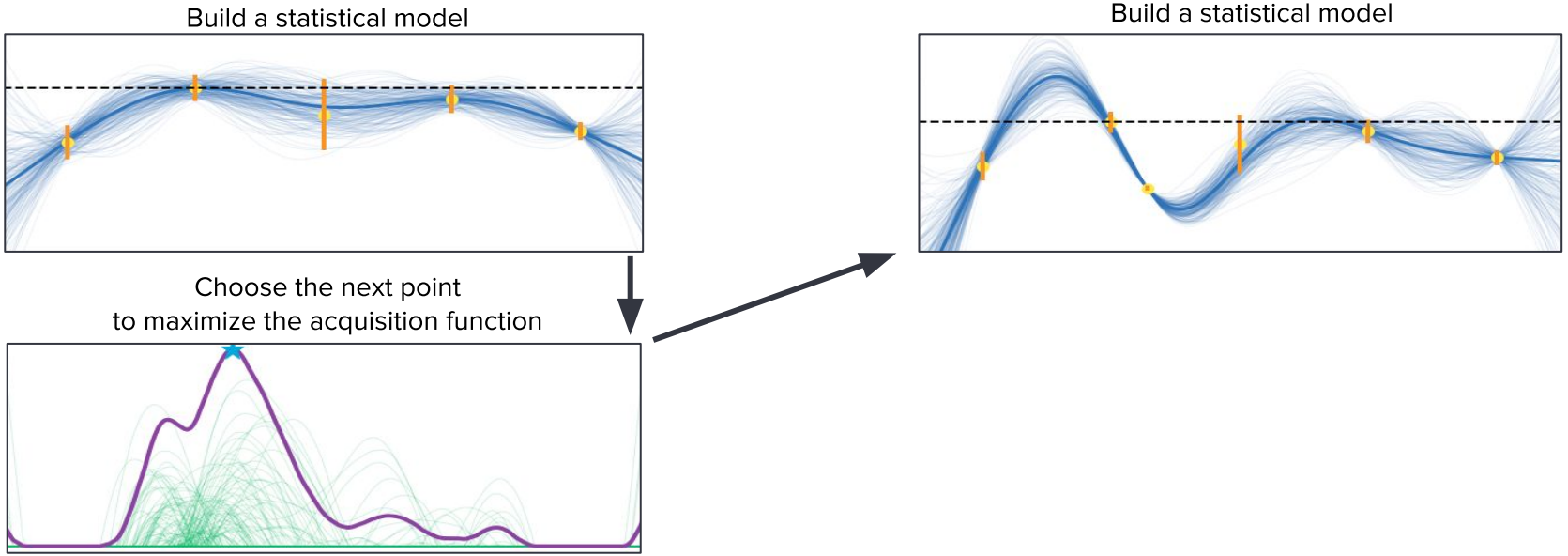


Addressing long training times

Our model takes a long time to train

Issue: every hyperparameter configuration takes a long time to evaluate

Solution: make the search as efficient as possible in order to evaluate as few samples as possible

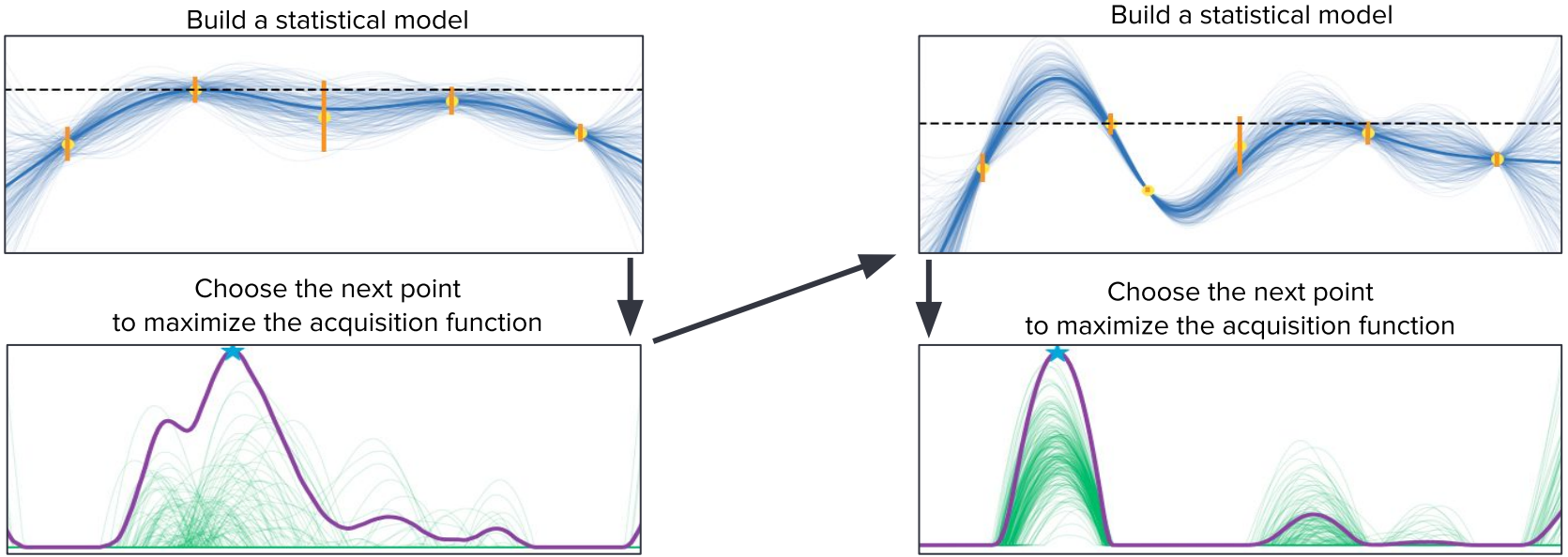


Addressing long training times

Our model takes a long time to train

Issue: every hyperparameter configuration takes a long time to evaluate

Solution: make the search as efficient as possible in order to evaluate as few samples as possible



Utilize compute resources

Utilize compute resources

We do have available compute resources

Issue: the delta in training times for different configurations of hyperparameters are very different

Utilize compute resources

We do have available compute resources

Issue: the delta in training times for different configurations of hyperparameters are very different

Solution: leverage asynchronous parallel optimization in order to accommodate the difference in training times

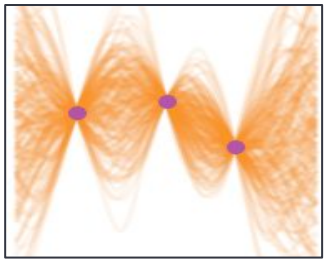
Utilize compute resources

We do have available compute resources

Issue: the delta in training times for different configurations of hyperparameters are very different

Solution: leverage asynchronous parallel optimization in order to accommodate the difference in training times

Statistical Model



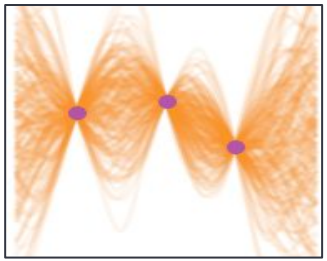
Utilize compute resources

We do have available compute resources

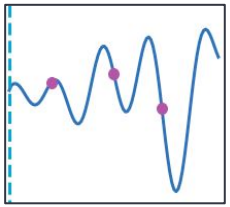
Issue: the delta in training times for different configurations of hyperparameters are very different

Solution: leverage asynchronous parallel optimization in order to accommodate the difference in training times

Statistical Model



Parallel bandwidth = 1



Next point(s) to evaluate:

Parallel bandwidth represent the # of available compute resources

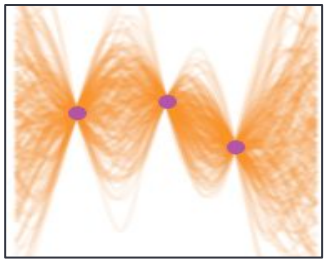
Utilize compute resources

We do have available compute resources

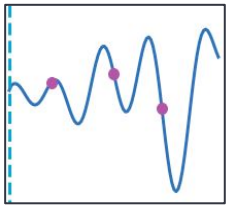
Issue: the delta in training times for different configurations of hyperparameters are very different

Solution: leverage asynchronous parallel optimization in order to accommodate the difference in training times

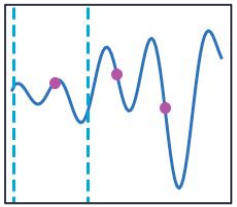
Statistical Model



Parallel bandwidth = 1



Parallel bandwidth = 2



Next point(s) to evaluate:

Parallel bandwidth represent the # of available compute resources

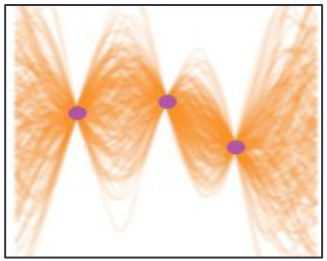
Utilize compute resources

We do have available compute resources

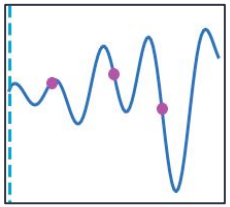
Issue: the delta in training times for different configurations of hyperparameters are very different

Solution: leverage asynchronous parallel optimization in order to accommodate the difference in training times

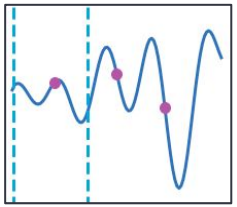
Statistical Model



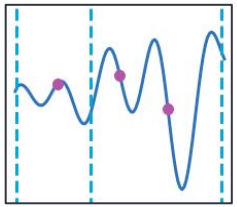
Parallel bandwidth = 1



Parallel bandwidth = 2



Parallel bandwidth = 3



Next point(s) to evaluate:

Parallel bandwidth represent the # of available compute resources

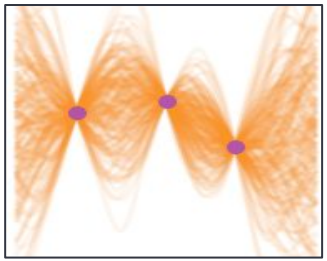
Utilize compute resources

We do have available compute resources

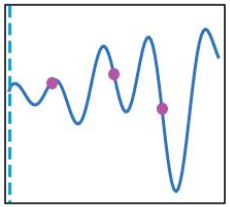
Issue: the delta in training times for different configurations of hyperparameters are very different

Solution: leverage asynchronous parallel optimization in order to accommodate the difference in training times

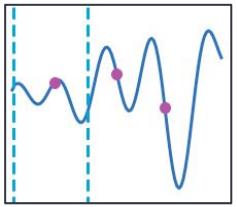
Statistical Model



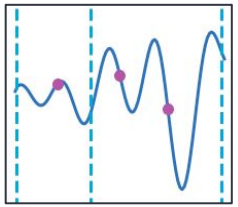
Parallel bandwidth = 1



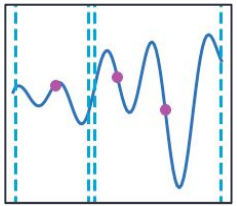
Parallel bandwidth = 2



Parallel bandwidth = 3



Parallel bandwidth = 4



Next point(s) to evaluate:

Parallel bandwidth represent the # of available compute resources

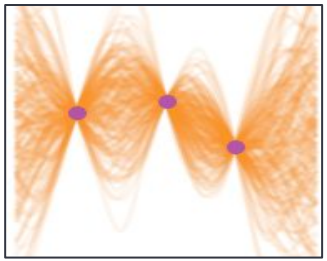
Utilize compute resources

We do have available compute resources

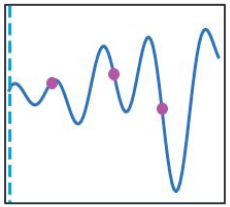
Issue: the delta in training times for different configurations of hyperparameters are very different

Solution: leverage asynchronous parallel optimization in order to accommodate the difference in training times

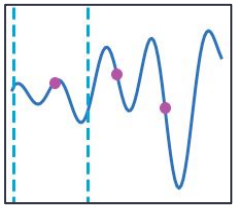
Statistical Model



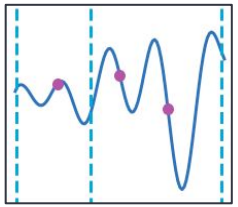
Parallel bandwidth = 1



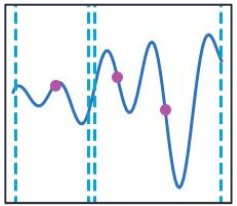
Parallel bandwidth = 2



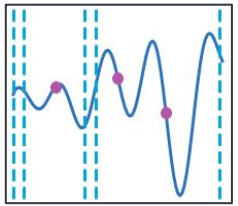
Parallel bandwidth = 3



Parallel bandwidth = 4



Parallel bandwidth = 5



Next point(s) to evaluate:

Parallel bandwidth represent the # of available compute resources

Using previous results

Using previous results

We know the best results from the previous optimizations

Issue: we don’t want to start the optimization process all over every time that we want to re-optimize a model

Using previous results

We know the best results from the previous optimizations

Issue: we don’t want to start the optimization process all over every time that we want to re-optimize a model

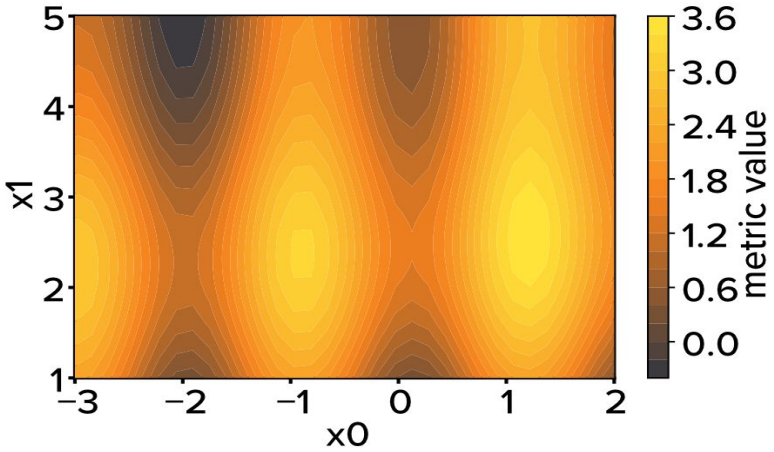
Solution: input prior information in the form of probability density functions

Using previous results

We know the best results from the previous optimizations

Issue: we don’t want to start the optimization process all over every time that we want to re-optimize a model

Solution: input prior information in the form of probability density functions

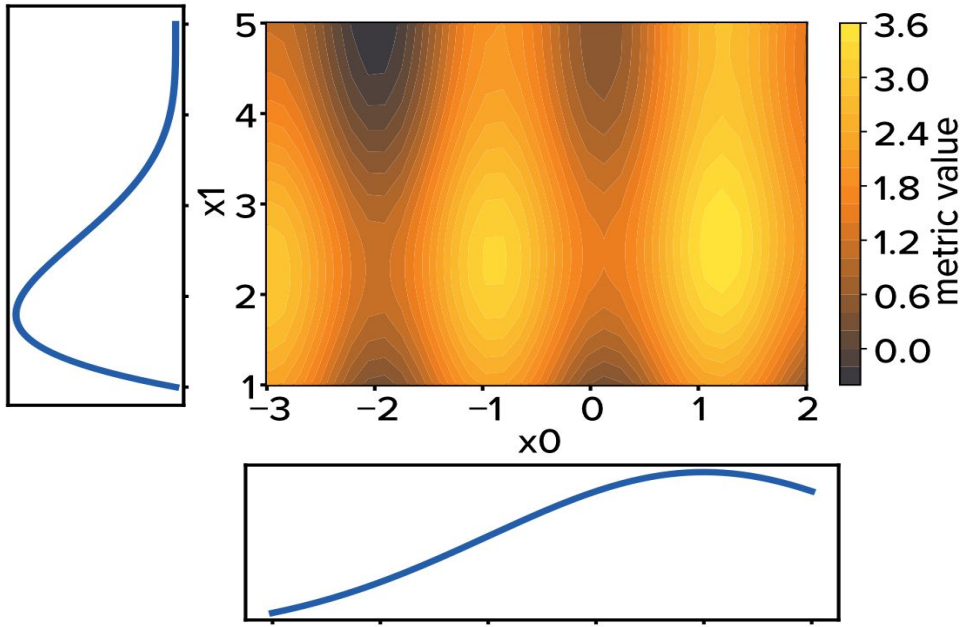


Using previous results

We know the best results from the previous optimizations

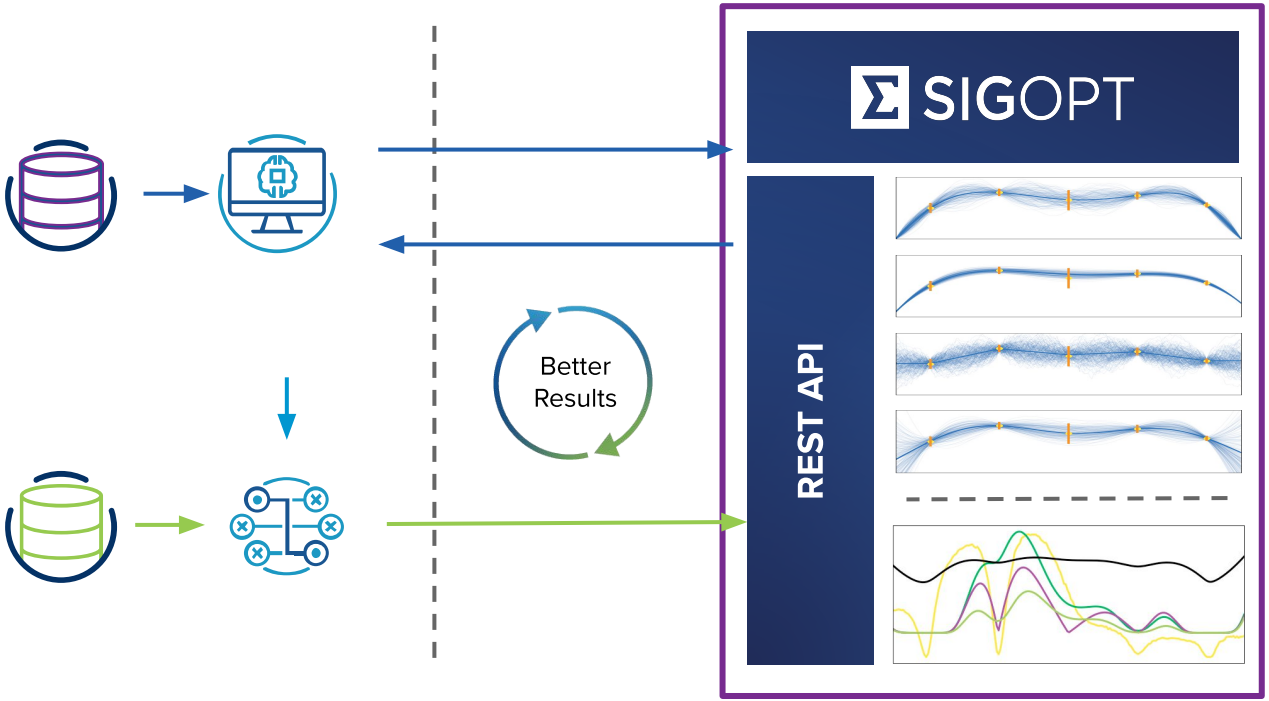
Issue: we don't want to start the optimization process all over every time that we want to re-optimize a model

Solution: input prior information in the form of probability density functions



Conclusion / next steps

The SigOpt API handles the complexity



Conclusion

Bayesian optimization allows you to tackle these problems:

Addressing long training times: Using efficient search in order to do as few samples as possible

Utilize compute resources: Use asynchronous parallel optimization in order utilise compute resources

Using previous results: Incode prior results in the form of probability density functions

Next steps

What are other ways for us to guide the optimization:

Add minimum requirements to metrics: imposing thresholds on our metrics

Adding guardrail metrics: constrain the search space in the form of guardrail metrics

Experiment transfer: link most recent experiments to previous experiments

Multitask optimization: use low fidelity versions of the model for efficient search

Thank you for listening in!

For questions and comments: tobias@sigopt.com

For more information visit: **https://sigopt.com/**

