Supporting Information for

“Progress towards machine learning reaction rate constants”

Evan Komp,[1] Nida Janulaitis[1] and Stéphanie Valleau,[1]

[1]*Department of Chemical Engineering, University of Washington, Seattle, Washington 98115, United States*

Supervised ML regression algorithms

We briefly describe neural networks (NNs), gaussian process regression (GPR), and random forests (RF) as these have been used frequently in the context of kinetics. For a more detailed description we refer the reader to e.g. Ref [1]

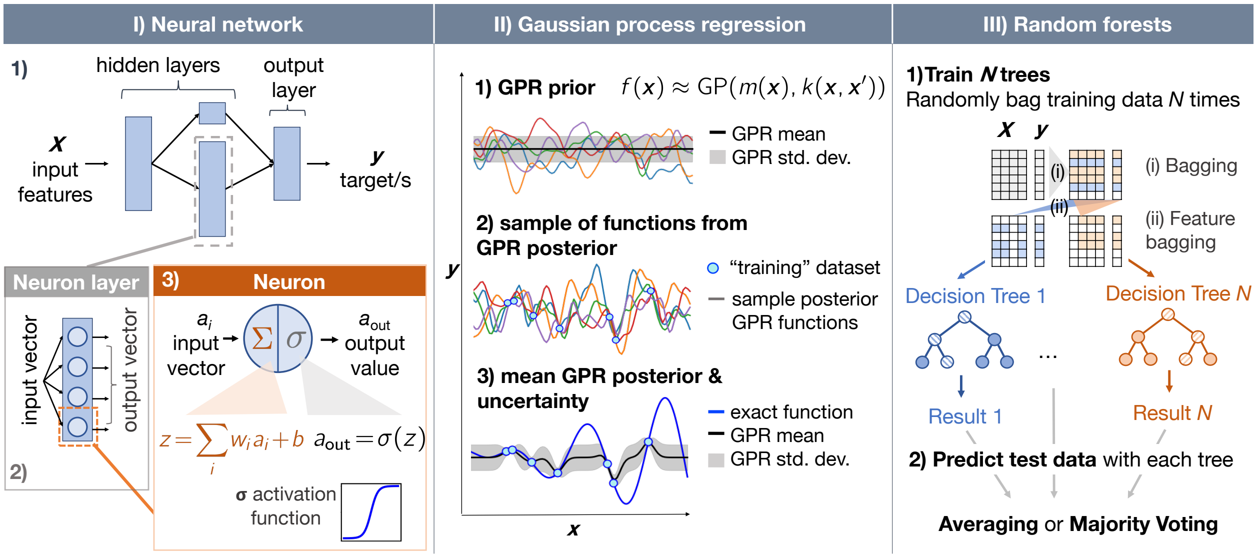

Figure 1: Panel I) Feedforward neural networks. *Subpanel 1)* Neural networks consist of layers of interconnected neurons which compute a target value, ***y***, from some input features ***X***. The feature-target relationship is established by the trained neuron weights and bias values. *Subpanel 2)* Neural network layer – a set of one or more neurons. *Subpanel 3)* Neurons: the building blocks of a neural networks. A neuron creates a linear combination of input values, a_i, for set values of weights, w_i and bias b, and applies an activation function, σ, to the result of the linear combination to produce the output value, a_{out}. The neuron’s weights and bias are parameters learned during training. Panel II) Gaussian process regression. A gaussian process prior with zero mean and a kernel of choice is defined, top plot shows example prior functions for the prediction of the target y from x. Second plot from top shows sample functions from the posterior, constructed from the prior based on the training data. Third plot shows the mean of the trained GPR and prediction uncertainty. Panel III) Random Forest regression. Training data is bagged into N bags, each of which is used to train a decision tree. During testing, the outputs from the trained decision trees are averaged or majority voting is used to provide the output target value.

Neural networks

Neural networks consist of layers of interconnected “neurons” (Figure 1, panel I:1), which compute an output target from input features. When more than one hidden layer is present, NN are referred to as deep neural networks (DNNs). For feedforward neural networks, the information is only passed forward. In each layer (Figure 1, panel I:2), neurons transform input values to output values by applying an “activation function” to the weighted and biased sum of inputs (Figure 1, panel I:3). These values are passed on to the next layer until the last layer, where the target output value/s, $\boldsymbol{y}$, are generated. The optimal values of neuron weights and bias are unknown. These parameters are found by minimizing a so-called "loss" function iteratively over "epochs" when training the NN. Epochs indicate one iteration over the entire training dataset. The loss function represents the error of the model with respect to the exact values. Several types of optimizers have been developed,[2] these include gradient and stochastic descent optimizers, where the change in loss is used to find optimal parameters,[3] as well as the ADAM optimizer.[4] NNs are modular in complexity and can capture highly nonlinear relationships.

Gaussian process regression

Gaussian process regression (GPR) is a non-parametric bayesian supervised machine learning model.[1] In GPRs, given the input training data, $\boldsymbol{X}$, one infers a posterior distribution over functions to make predictions, $\boldsymbol{y}^*$, for new inputs, $\boldsymbol{X}^*$, e.g. test data. The prior for the regression function is defined as a gaussian process; it depends on a mean function and kernel or covariance function (Figure 1, panel II:1). With the prior one can define the joint distribution of the training and test outputs. The posterior distribution is then obtained by conditioning the joint gaussian prior distribution on the observations, $\boldsymbol{X}, \boldsymbol{y}$ (Figure 1, panels II:2-3). GPs ability to predict depends on the choice of kernel and estimation of optimal kernel parameters. GPs require the inversion of the kernel matrix of size $N \times N$ with N the number of training points. This typically limits the use of GPRs to small systems due to the cost of the inversion $O(N^3)$. One advantage of GPs is that a measure of uncertainty is provided for each prediction.

Random forests

Random forests consist of a set of decision trees, each of which is trained to minimize loss by drawing decision boundaries (Figure 1, panel III). The target or output of the random forest is obtained by averaging over the decision trees or via majority voting. During training, input features are passed from a root node to inner nodes or decision nodes down to leaf nodes. At each decision node, the input data is split in two based on a decision. Splits are chosen as those which minimize the residual sum of squares of the predicted target with respect to the true target.

The trained decision tree then takes previously unseen features and decides on an output based on its learned structure and splits. Random forests are less likely to overfit than single decision trees, as each tree sees a different subset of the training data examples and features. In the context of compound classification a good description of random forests can be found here.[5]

References

1. Murphy, K. P. *Machine learning: a probabilistic perspective*. (MIT press, 2012).
2. Goodfellow, I., Bengio, Y., Courville, A. & Bengio, Y. *Deep learning*. vol. 1 (MIT press Cambridge, 2016).
3. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv:1609.04747* 1–14 (2016).
4. Kingma, D. P. & Ba, J. L. Adam: A method for stochastic optimization. *arXiv:1412.6980* 1–15 (2014).
5. Svetnik, V. *et al.* Random Forest: A Classification and Regression Tool for Compound Classification and QSAR Modeling. *J. Chem. Inf. Comput. Sci.* **43**, 1947–1958 (2003).