



Hyperparameters are configuration settings that are external to the machine learning model itself and need to be manually set before the training process begins. Setting/tuning these hyperparameters is extremely difficult for several reasons. There are large number of hyperparameters, and each hyperparameter can take on multiple values. One hyperparameter can affect another, thus understanding the complex dependencies requires a lot of experimentation which many can't afford as tuning them is time-consuming and expensive. Worst of all there is no universal solution, as one set of hyperparameters won't work for a different problem/model. So we provide a comprehensive approach to this problem. To know more about hyperparameters refer to this link :(<https://www.javatpoint.com/hyperparameters-in-machine-learning>).

Tuning Strategy:

Start with a problem statement and query the necessary data and make it ready (cleaning) for the model training.

Strategy to tune hyperparameters:

1. Identify an appropriately-scoped goal for the next round of experiments.
2. Design and run a set of experiments that makes progress towards this goal.
3. Learn what we can from the results.
4. Consider whether to launch the new best configuration.

Tuning Steps:

1. Start with a model configuration

Model architectures typically have various hyperparameters that determine the model's size and other details (e.g. number of layers, layer width, type of activation function). Start with small models and grow eventually. Try to choose standard model configuration, so one doesn't have to spend a lot of time tuning model hyper-parameters. Choose a starting optimizer, epoch, and performance metrics.

2. Choosing Batch Size

Viz 1: Draw for yourself

By keeping all hyperparameters the same, increasing batch size increases validation accuracy.

Viz 2: Draw for yourself

But by using the best hyperparameters for individual batch sizes, we can see that there is no effect of increasing batch size on performance.

Any batch size works but configurations of other hyperparameters like an epoch, optimizer, and regularizer change with batch size. So chose a single batch size for tuning all other hyperparameters.

Viz 3: Animation of steps needed vs batch size

In theory, the steps needed are halved for each doubling batch size. But in practice, it gets linear after a particular batch size. So, choose the largest available batch size for faster training and experiments.

Page 3

3. Finding Best Optimizer

Viz1: Best learning rate differs for different optimizers.

Viz 2: Tuning Adam Optimizer with sliders

Viz 3: Comparing convergence with best individual hyperparameters.

Page 4

4. Other Hyperparameters

Viz: Sun dial plot for other options

Page 5

Plot covering everything (A3)