First Steps with TensorFlow: Programming Exercises

Tools used:

* NumPy
* Pandas
* Matplotlib

The system recalculates the model's loss value and adjusts the model's weights and bias after each **iteration**. Each iteration is the span in which the system processes one batch. For example, if the **batch size** is 6, then the system recalculates the model's loss value and adjusts the model's weights and bias after processing every 6 examples.

One **epoch** spans sufficient iterations to process every example in the dataset. For example, if the batch size is 12, then each epoch lasts one iteration. However, if the batch size is 6, then each epoch consumes two iterations.

It is tempting to simply set the batch size to the number of examples in the dataset (12, in this case). However, the model might actually train faster on smaller batches. Conversely, very small batches might not contain enough information to help the model converge.

My understanding:

Num of iterations in an epoch = Data set size / Batch size

Summary of hyperparameter tuning

Most machine learning problems require a lot of hyperparameter tuning. Unfortunately, we can't provide concrete tuning rules for every model. Lowering the learning rate can help one model converge efficiently but make another model converge much too slowly. You must experiment to find the best set of hyperparameters for your dataset. That said, here are a few rules of thumb:

* Training loss should steadily decrease, steeply at first, and then more slowly until the slope of the curve reaches or approaches zero.
* If the training loss does not converge, train for more epochs.
* If the training loss decreases too slowly, increase the learning rate. Note that setting the training loss too high may also prevent training loss from converging.
* If the training loss varies wildly (that is, the training loss jumps around), decrease the learning rate.
* Lowering the learning rate while increasing the number of epochs or the batch size is often a good combination.
* Setting the batch size to a *very* small batch number can also cause instability. First, try large batch size values. Then, decrease the batch size until you see degradation.
* For real-world datasets consisting of a very large number of examples, the entire dataset might not fit into memory. In such cases, you'll need to reduce the batch size to enable a batch to fit into memory.

Remember: the ideal combination of hyperparameters is data dependent, so you must always experiment and verify.