Installation troubles:

* Make sure you have installed python 64 bit , make sure python is added to path, there should be a check box in the installer
* There could be a versioning problem, e.g. in Nov 2020 Python 3.9 version, pip could not install tensorflow, so revert to 3.8.5.  
  <https://stackoverflow.com/questions/48720833/could-not-find-a-version-that-satisfies-the-requirement-tensorflow>

Is There a Standard Heuristic for Model Tuning?

This is a commonly asked question. The short answer is that the effects of different hyperparameters are data dependent. So there are no hard-and-fast rules; you'll need to test on your data.

That said, here are a few rules of thumb that may help guide you:

* Training error should steadily decrease, steeply at first, and should eventually plateau as training converges.
  + If the training has not converged, try running it for longer.
  + If the training error decreases too slowly, increasing the learning rate may help it decrease faster.
* But sometimes the exact opposite may happen if the learning rate is too high.
  + If the training error varies wildly, try decreasing the learning rate.
  + Lower learning rate plus larger number of steps or larger batch size is often a good combination.
* Very small batch sizes can also cause instability. First try larger values like 100 or 1000, and decrease until you see degradation.

Again, never go strictly by these rules of thumb, because the effects are data dependent. Always experiment and verify.

Common hyperparameters :

* steps, which is the total number of training iterations. One step calculates the loss from one batch and uses that value to modify the model's weights once.
* batch size, which is the number of examples (chosen at random) for a single step. For example, the batch size for SGD is 1.