ANNA: Ancillary Scripts
User Guide

Donald Lee-Brown
donald[at]ku.edu
Department of Physics & Astronomy
The University of Kansas

April 20, 2018

1 Overview

Instructions for running the scripts necessary to generate and post-process the training sets used by ANNA. The two Python scripts are `SPECGENv1.6.py` and `POSTPROCv1.10.py` - the former generates a set of (typically high-resolution) continuum-fit spectra using the standard Kurucz model atmosphere grid and a supplied linelist, while the latter applies various postprocessing effects, including velocity shifts and resolution degradation. Both are controlled by their own parameter files.

These scripts were written to generate ANNA training sets for Hydra spectra, and while they should work when generating other spectra, the code isn't as clean as the main ANNA code since they are mostly intended for in-house usage. Have fun!

For the quick-and-dirty usage rundown, skip to Section 3.

2 Required Software

Scripts have been updated to work with Python 3.5+ (I recommend the Anaconda Python installation), but at some point in the past they were compatible with 2.7+. I don't know if they still are.

Modules used (all should be bundled with Anaconda):

- subprocess
- random
- multiprocessing
- functools
- time

- pandas
- numpy
- array
- glob
- math
- scipy
- os

You will also want to run these on a Linux-based system (not sure if macOS will work, but Windows will definitely not because the scripts use Unix commands).

The Kurucz model atmosphere grid (directory with the AM*.dat and AP*.dat atmosphere files, as well as `mspawn` and `mspSep08.c`. Obviously you must also have a C compiler installed as well.

SPECTRUM 2.76+ (http://www.appstate.edu/ grayro/spectrum/spectrum.html) installed such that you can call it from the terminal (i.e. you can just type "spectrum" into the terminal and the program will run). Installing SPECTRUM requires the GCC compiler (already installed on most Linux distributions).

3 Typical Usage

Once all the software is installed, to run the scripts, first make an empty directory somewhere - this is where everything will be run. Then in the directory place the `SPECGENv1.6.py` and `POSTPROCv1.10.py` scripts, the `generate.param` and `postproc.param` parameter files, whatever linelist you are using (which should be in the format that SPECTRUM requires), the `stdatom.dat` file required by SPECTRUM, and a subdirectory titled "atm" that contains the AM*.dat and AP*.dat atmosphere files, as well as `mspawn` and `mspSep08.c`. Most of this stuff is hardcoded into the scripts, so make sure, for example, that the parameter file for `SPECGEN` is named "generate.param".

After setting the parameter file values (see Section 4), the first script to run is `SPECGENv1.6.py`. This will generate the specified number of continuum-fit spectra with varied temperature, gravity, microturbulent velocity, and metallicity, at a given resolution with specified wavelength coverage. Each spectrum will be dumped to a two-column text file (wavelength, flux), and all the selected parameters will be put in `atmparams.out`. Do NOT alter or move the files or lose the `atmparams.out` file or you will have to start over. `SPECGENv1.6.py` is parallelized and will hog the CPU, so plan on getting a few cups of coffee or something while the computer is busy.

The Kurucz atmospheres do not permit arbitrary combinations of atmospheric parameters. For example, the grid only extends to [Fe/H] = +0.5,

so requesting a model with [Fe/H] + 2.0 will result in an error. Illegal combinations of parameters will result in an "atmfault" line being added to the `atmparams.out` file produced by `SPECGEN`. It's a good idea to double check to make sure there aren't any of these.

After `SPECGENv1.6.py` has been run, the generated spectra can be postprocessed using `POSTPROCv1.10.py`. This script will read in the spectra and use SPECTRUM to apply things like rotational velocity broadening, etc. and will also downgrade the resolution of each spectrum to the target resolution to train ANNA. If more postprocessed spectra are requested than were originally created by `SPECGEN`, spectra will be reused (with different postprocessing parameters). `POSTPROC` produces a single two-column text file (wavelength, flux) for each requested spectrum; options in the parameter file include removing these after the script is done and packaging them all into a binary file with the specific format required by ANNA. Like `SPECGEN`, `POSTPROC` is parallelized. `POSTPROC` will also output a file, "atmparams.postprocess.out", which lists the postprocessing parameters used to generate each spectrum.

Generally, to train ANNA to parameterize Hydra spectra spanning a moderate range in atmospheric parameters (for example, 2000 K in surface temperature, gravities including giants and dwarfs, 0.5 dex in metallicity), around 10,000 high-resolution spectra should be generated with `SPECGEN`. The output of `POSTPROC` should then be set to contain around 200,000 spectra (in binary format). When training ANNA, it is recommended to use a separately-generated set of cross-validation spectra, so `SPECGEN` and `POSTPROC` can be run an additional time to generate a sample of $\sim$3000-10000 spectra (also in binary format).

Note that since `POSTPROC` can be run separately from `SPECGEN`, it is possible to generate a single library of high-resolution spectra using `SPECGEN`, then use those spectra as a base to generate several training sets using `POSTPROC`. This can save an appreciable amount of time. Just bear in mind that `SPECGEN` sets the ranges for the key atmospheric parameters of interest (e.g., temperature, metallicity), so if ANNA needs to be used to parameterize different spectra, then `SPECGEN` should be re-run. As another example of the utility of splitting `SPECGEN` and `POSTPROC`, testing of ANNA indicates that the network is only capable of accurately parameterizing stars with radial velocities within a 40 km/s range at a time. Thus, if the sample to analyze with ANNA consists of stars with radial velocities spanning 200 km/s, `POSTPROC` should be used several times to generate a few training sets with smaller radial velocity ranges that together span the entire 200 km/s. The same `SPECGEN` run can be used for each of these `POSTPROC` iterations.

Both `POSTPROC` and `SPECGEN` can be pretty users of disk space - expect to use a few GB of space for each run when training Hydra. It typically takes a few hours for a computer to generate and postprocess the few hundred thousand spectra required to train ANNA.

4 Parameter Files

Parameter values used for Hydra are given in parentheses next to each item.

SPECGEN Parameters

- `NUM_STARS: integer`
  Number of spectra to generate (Hydra: 10000 for training, 1000 for cross-validation).
- `TEMP_RANGE: float,float`
  Temperature range (in K) to randomly sample when generating spectra (e.g. 5000.0,6500.0).
- `GRAV_RANGE: float,float`
  Gravity range (log(g)) to randomly sample when generating spectra (e.g. 2.5,5.0).
- `MET_RANGE: float,float`
  $[Fe/H]$ range (dex) to randomly sample when generating spectra (e.g., -0.5,0.5).
- `VT_RANGE: float,float`
  Microturbulent velocity range (km/s) to randomly sample when generating spectra (e.g. 1.0,1.5).
- `WAVE_RANGE: float,float`
  Wavelength coverage of each spectrum, in Angstroms (Hydra: 6600.0, 6900.0).
- `NAT_RESOLU: float`
  Resolution (in Anstroms) of the generated spectra. This should be set so the output spectra are high resolution (relative to what is produced by the spectrograph) (Hydra: 0.01).
- `SLINELIST: string`
  Name of the linelist to use when generating spectra (Hydra: valdtweakv1.lst).

POSTPROC Parameters

- `NUMBER_OUTPUTS: integer`
  Number of spectra to generate (Hydra: 200000 for training, 5000 for cross-validation).
- `BINARY_OUTPUT: YES/NO`
  After running `POSTPROC`, package all postprocessed spectra into a single binary file with name "allspec" (Hydra: YES).
- `VERBOSE_OUTPUT: YES/NO`
  Retain individual postprocessed spectra after running. This is most useful when you are only interested in using the binary output to train ANNA (Hydra: NO).

- `NAT_RESOLU: float`
  Input resolution in Angstroms. Should be the same as `NAT_RESOLU` used for `SPECGEN` (Hydra: 0.01).

- `SMO_RESOLU: float`
  Smoothed resolution of the postprocessed spectra, in Angstroms. Spectra are smoothed assuming a Gaussian line spread function (Hydra: 0.615 is fine; 0.738733 has been used in the past and works as well).

- `PIXEL_SCALE: float`
  Pixel scale in Angstroms. Due to a quirk in SPECTRUM, this must be an integer multiple of `NAT_RESOLU`. The program will still run if its not an integer multiple, but things end up looking very strange. Thus, with Hydra, the pixel scale is actually 0.205 Angstroms/px, but because `NAT_RESOLU` is set to 0.01, the closest integer multiple to use for `PIXEL_SCALE` is 0.200. Note that ANNA interpolates everything onto a common wavelength grid so this is fine (Hydra: 0.200).

- `ROT_VELOCI: float,float`
  Range in rotational velocity (km/s) to sample when postprocessing (e.g. 0.0,15.0).

- `ROT_FUDGE: float`
  Fudge factor that adds a constant rotational velocity (km/s) to whatever was selected for a given spectrum. Useful when the line spread function of the spectrograph isn't able to be matched by the results of `SMO_RESOLU` (Hydra: 0.0).

- `RAD_VELOCI: float,float`
  Range in radial velocity (km/s) sample when postprocessing (e.g. -20.0,20.0).

- `WAVE_RANGE: float,float`
  Wavelength range (in Angstroms) to trim each postprocessed spectrum to. Should be smaller than `WAVE_RANGE` used for `SPECGEN` but if not, spectra will be padded with 1.0 flux (continuum level) (Hydra: 6625.0,6825.0 - this region is small but generally the continuum fitting is reliable and the lines are relatively well-modeled by SPECTRUM and the linelist).