The Detection Threshold of Multiple-Line Systems

Brendan Watters

Supervised by Dr. Marc Sarzi

20 January, 2025

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Armagh Observatory & Planetarium
Exploring the Cosmos since 1790

Abstract

We train a number of artificial neural networks to differentiate between pure noise and synthetic spectra of various signal to noise ratios, generated based on the spectra of four different types of extended astrophysical sources: low-ionisation nuclear emission regions (LINERs), planetary nebulae, Seyfert galaxies and star forming regions. Each neural network learns to differentiate between noise and one of the chosen classes of spectra.

We also compare the accuracy of fitting separate spectral lines independently and fitting spectral lines from the same source at the same time, in such a way that the best fit for all lines together is found, rather than the best fit for each line individually, i.e. rather than simply finding the best fit for the Hα line and then best fit for each of the [O III] lines etc., the fit that gave the best fit but same standard deviations and velocities for all of the lines together was found. It was found that fitting all of the lines together produced more accurate results than treating each line separately.

Acknowledgments

I would like to thank Dr. Marc Sarzi for his continued supervision and support throughout the course of this project. The code I used in this project was based on work done by Anna Cestaro, an undergraduate summer intern at Armagh Observatory. I would like to thank Saskia Schlagenhauf and Arjun Chawla for their guidance, Andrew Marshall-Lee for his advice on machine learning and everyone at Armagh Observatory and Planetarium for their general help and guidance during my time there. Thank you also to the Aontas na Mac Léinn, Coláiste na Trionóide/Trinity College Dublin Students' Union Refresh IT Repairs for their help repairing my laptop, and The Laptop Shop for repairing my laptop the second time it broke throughout the course of this project. Thank you to my tutor Prof. Stefan Hutzler for helping to organise other arrangements for me while my laptop was broken.

Contents

1 Introduction

Spectra are used in astronomy to determine various details about astronomical objects such as what elements are present and in what abundances, or the temperature and density of the source. Both the shape of the continuum spectrum as well as the varieties of emission and absorption lines present are useful to gather information about the system. This project is concerned exclusively with emission lines (although I do believe it could be extended to absorption lines in the future with minimal alterations to the code).

Not all spectral lines are created equal, some can have signal to noise ratios (S/Ns) which are extremely high and stand out obviously from the background continuum and noise, while others can have S/Ns as low as or lower than 1. However all lines are interesting and can be used to determine more about a source, so it is beneficial to recover as much information about as many lines as possible, which is the focus of part of this project.

This information includes the fluxes of the lines and the ratio between the fluxes of different lines and redshifts/positions of the lines as well as the profile of the line, and is associated parameters, e.g. if the line is Gaussian in profile then its standard deviation should be measured. Each piece of information about a spectral line can be used to determine something about the source being observed, for example, the widths of emission lines can be used to determine the type of source being observed, for instance Seyfert 1 galaxies (a type of active galaxy) have broad permitted lines but thinner forbidden lines, while Seyfert 2 galaxies have permitted and forbidden lines which are approximately the same width (thinner).[1]

The fluxes of the emission lines can be used to estimate the star formation rate, the amount of ionised hydrogen, H II, ionised by young massive O-type stars in a region can be measured, allowing the overall rate of star formation be measured from knowledge of stellar lifetimes and the initial mass function.[2] Whereas the positions and redshifts of emission lines in galaxies can be used to determine the mass of the galaxy through the measurement of the rotation of the galaxy by tracking the speed of rotation at different

distances from the galactic centre, constructing a rotation curve from that and finding a mass from that.[3]

Clearly, if these parameters of lines can be measured more accurately then more accurate measurements of these values and others can be made, which is one part of what this project aims to do.

The other part of this project is focused on determining when a line truly has been detected and when it is simply noise that has been detected and been erroneously labelled and fit as a true spectrum of emission lines. This is a difficult task to do by eye or conventional programming methods, as such I have used machine learning, specifically artificial neural networks, which I have trained to differentiate between a true spectrum and noise. This again could be used for analysis of spectra with lines at low S/N as it could determine if it is a real signal that has been detected or simply noise.

2 Theory and Background

2.1 Astrophysical Spectra

The spectrum of the radiation emitted by an astrophysical source depends on a number of factors; the types of elements present, their abundances, their ionisation states, and the density and temperature (which itself influences the ionisation state of the elements present) of the source.[4]

The spectra of nebulae and active galactic nuclei (AGNs) are dominated by emission lines. Some of the more common lines found in the optical spectra of nebulae are the forbidden lines emitted by [O III] at $\lambda = 4959$ Å and $\lambda = 5007$ Å and by [N II] at $\lambda = 6548$ Å and $\lambda = 6583$ Å. The permitted Balmer lines of hydrogen are also very common in the optical spectra. There are also spectral lines in the infrared and ultraviolet regions that are particularly strong (such as [S III] with $\lambda = 9096$ Å and $\lambda = 9523$ Å and the [O II] at $\lambda = 3726$ Å and $\lambda = 3729$ Å respectively), however this project deals only with those in the optical.[4][5]

There is also continuum emission in these regions from different processes, however I will not go into details as it is not important to this project.

The emission lines in the majority of nebulae are due to the ultraviolet light of nearby hot, massive stars exciting and ionising the hydrogen atoms (the most common type of atom in this environment and most others) in the nebula. This can be achieved by a photon with an energy of at least 13.6 eV (wavelengths less than 911.6 Å). Any excess energy from the photon is transferred to the kinetic energy of the newly emitted photo-electron. This energy is transferred throughout the nebula through collisions involving both electrons and ions.[4][5]

These aforementioned collisions excite electrons in the (non-hydrogen) ions to higher levels. The probabilities of downward radiative transitions from these excited levels are very low and as such, in higher density environments such as on Earth they do not occur as the ion is usually collisionally de-excited before a radiative transition can occur, and as such they are referred to as 'forbidden' transitions. However due to the lower densities of nebulae collisional de-excitation is less likely to occur than these transitions and so the nebula emits a forbidden line spectrum.[4]

The relative rates at which electrons undergo forbidden radiative transitions from the same excited state (and hence the relative intensities of the associated emission lines) are dictated by the ratio of the probabilities of said transitions. An example is the aforementioned [O III] lines at 4959 Å and 5007 Å which arise from the transition between the $^1D_2$ level and the $^3P_1$ and $^3P_2$ levels (the $^1D_2$ to $^3P_0$ transition does occur but it has a much lower transition probability and so can be ignored here). The ratio of these probabilities is approximately 1:3 and so is the observed ratio of intensities of the emission lines.[4] This of course applies to other species and lines as well.

There is reddening to consider also. Reddening is the name given to the phenomenon where shorter wavelength light is scattered more by the interstellar medium, making a source appear 'redder'. The intensity of light, I_λ, after passing through an interstellar medium with optical depth τ_λ is

$$I_\lambda = I_{\lambda 0} e^{-\tau_\lambda} \tag{2.1}$$

where $I_{\lambda 0}$ is the intensity that would be observed if there was no interstellar medium. The optical depth can be given by

$$\tau_\lambda = C f(\lambda) \tag{2.2}$$

where C is a constant for each star and $f(\lambda)$ is a function that is approximately the same for each star in the galaxy. The ratio of observed intensities from two different wavelengths is therefore given by

$$\begin{aligned} \frac{I_{\lambda_1}}{I_{\lambda_2}} &= \frac{I_{\lambda_1 0}}{I_{\lambda_2 0}} e^{-\left(\tau_{\lambda_1} - \tau_{\lambda_2}\right)} \\ &= \frac{I_{\lambda_1 0}}{I_{\lambda_2 0}} e^{-C(f(\lambda_1) - f(\lambda_2))} \end{aligned} \tag{2.3}$$

where λ_1 and λ_2 are the two wavelengths.[4]

However the optical depth of dust is approximately inversely proportional to the wavelength so in the case where the wavelengths similar the ratio of optical depths is approximately 1 and so reddening can be ignored.[6]

Some of the ions do of course recombine with electrons, however this happens at excited levels, when then decay to the ground states through radiative de-excitation. For example this produces the aforementioned H I Balmer series lines as well as the H I Paschen series (it also occurs for other species however the H I lines are the strongest of these lines).[4]

For recombination lines, such as the hydrogen Balmer series, the probabilities of transitions from the excited reionisation levels, and hence intensities of the lines, are dependent on the temperature and density of the gas, however this is a weak dependence. For example at a some n_e and T = 10^4 K the ratio between the Hα and Hβ intensities is 2.87, while at the same electron density and T = 2×10^4 it is 2.67.[4][5]

The ratio of the intensities of the α, β and γ Balmer lines, I(Hα):I(Hβ):I(Hγ), is known as the Balmer decrement. However due to reddening from dust both in the nebula and in the intervening medium the observed Balmer decrement can be different from the Balmer decrement at the source.[6][5]

2.2 Artificial Neural Networks

An artificial neural network is a computational system that mimics the workings of a biological brain. It uses a number of interconnected nodes (here called neurons) separated into a number of layers – at least two, one for input and one for output – which take in some input, perform some simple operation on it, and produce some output. [7]

The perceptron, among the first neural networks, was developed by Rosenblatt from the 1950s onwards[8], in it the neurons are divided into layers, starting with an input layer containing n nodes (with $n \geq 2$) which each take a single number as an input. The nodes in the next layer then take as input the weighted sum of all inputs from the first layer, it then subtracts a fixed threshold value from the input and applies a transfer function to produce the 'activation' of each neuron y_j, this can be expressed as

$$y_j = f\left(\sum_i w_{ij}x_i - \theta_j\right) \tag{2.4}$$

where x_i is the input activation on the i^{th} neuron on the input layer, w_{ij} is the weighting of the connection between the nodes, θ_j is the threshold value for the neuron, and f is the transfer function. This is then repeated for whatever number of 'hidden layers' – the layers of neurons between the input and output layers – are in the network (which can be 0) and finally the output layer which contains at least one neuron which gives some output. The values of w_{ij} and θ_j are initially unknown and must be adjusted throughout the training process to give the desired output.[7] Figure 2.1 shows an illustration of a network of this style.

The choice of the number of hidden layers as well as the number of neurons in each of the hidden layers is more open, however there has been work that suggests the optimal

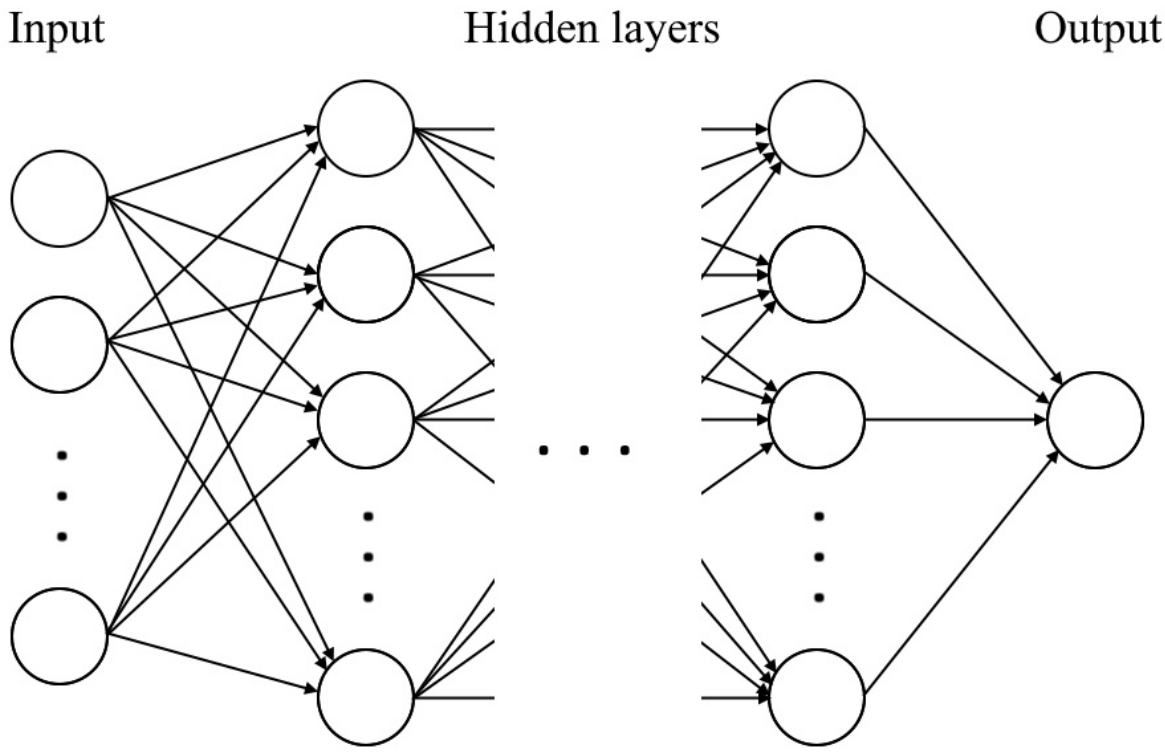


Figure 2.1: The design of a perceptron-based neural network which I used in this project. It takes n input values which pass through some number m of hidden layers each containing n_i neurons before giving one output. Diagram based on one found in [7]

.

number of hidden neurons, n, to learn the relation between N samples and give m outputs in a network with two hidden layers is given by

$$n = 2\sqrt{(m+2)N} \tag{2.5}$$

and the number of neurons in the first and second hidden layers (n_1 and n_2 respectively) is given by

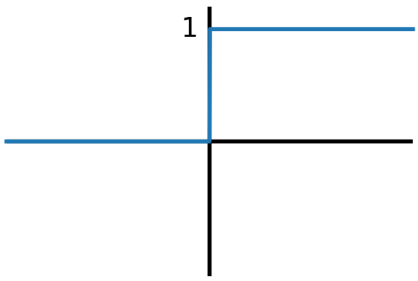
$$n_1 = \sqrt{(m+2)N} + 2\sqrt{\frac{N}{m+2}} \tag{2.6}$$

and

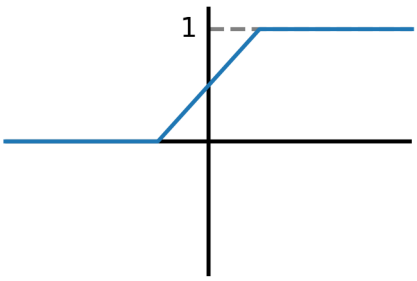
$$n_2 = m\sqrt{\frac{N}{m+2}} \tag{2.7}$$

respectively (of course n, n_1 and n_2 must also be rounded so that they are integers).[9]

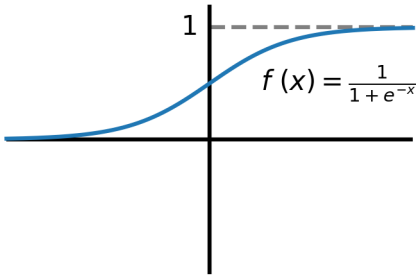
As described above, when a neuron receives input signals it calculates the weighted sum of the inputs, subtracts the threshold value and passes it through a transfer function, or activation function. One of the most significant differences between neurons in these networks is their transfer function. The transfer functions typically have some maximum and minimum values output values, however the shapes of the functions can vary widely. Some examples are shown in figure 2.2.



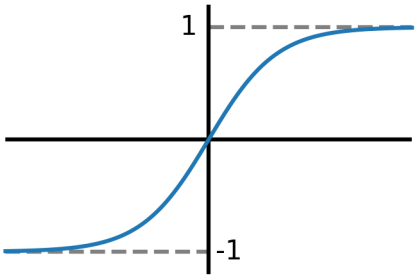
(a) Threshold function.



(b) Hard limit function.



(c) Sigmoid function.



(d) Tanh function.

Figure 2.2: Some of the possible transfer functions for an artificial neural network.

The most simple transfer function is a binary threshold function, (figure 2.2a) if the input is greater than or equal to some threshold, the output is 1, otherwise it is 0. The hard-limit function (figure 2.2b) is similar to the threshold function, if the input is greater than or equal to some value the output is 1, and if it is less than or equal to some other value the output is 0. Between these values the output is a linear function of the input.[7]

The sigmoid, $f(x) = 1/(1 + e^{-x})$, and hyperbolic tangent functions (figures 2.2c and 2.2d) have the distinction of being defined and differentiable at all points on the interval $(-\infty, \infty)$, they are also monotonically increasing (their derivatives are always ≥ 0) and have upper and lower asymptotes. These aspects are important as they are necessary for the use of back-propagation when training the network.[7]

Back-propagation is one way in which the weights between connections and the threshold values (w_{ij} and θ_j from equation 2.4) are adjusted during training to give a combination that means the network gives the desired output. Back-propagation networks are very similar to Perceptron networks and in fact once they are trained they behave in the same manner, however they must have at least one hidden layer in their structure and the training of the networks is different.[7]

The way that the back-propagation network is trained is using supervised learning, in supervised learning a model is given some inputs with some expected associated outputs. The model initially has arbitrary w_{ij} and θ_j so the initial output could be any of the possible outputs of the model. For one input, i, the square error of the output value, E_i is given by

$$E_i = \frac{1}{2} \sum_j (t_j - y_j)^2 \tag{2.8}$$

where t_j is the target output and y_j is the actual output of neuron j. The sum of all inputs, $E = \sum_i E_i$, is then found. The aim now is to minimise E by adjusting the weights, w_{ij}. The new weights, $w_{ij,new}$ are given by

$$w_{ij,new} = w_{ij,old} + \alpha \text{Delta}(w_{ij,old}) - y_j \tag{2.9}$$

where α is a constant used to control by how much the Delta function affects the weights. Delta is defined as

$$\text{Delta}(w_{ij}) \propto -\frac{\partial E}{\partial w_{ij}} \ . \tag{2.10}$$

By adjusting the weights of all connections in the network proportionally to the derivative of the error they can be adjusted over many iterations to reach a minimum value of E and optimise the network. The value of $\frac{\partial E}{\partial w_{ij}}$ is of course found using the chain rule,

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial w_{ij}} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial w_{j,TOT}}\frac{\partial w_{j,TOT}}{\partial w_{ij}} \tag{2.11}$$

where $w_{j,TOT} = \sum_i w_{ij}$. So

$$\frac{\partial w_{j,TOT}}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}}\left(\sum_k w_{kj}x_k - \theta_j\right) = \frac{\partial}{\partial w_{ij}} w_{ij}x_i = x_i \tag{2.12}$$

where k is a dummy variable to sum over instead of i as in equation 2.4 as i now has to be fixed. As in equation 2.4, $y_j = f\left(\sum_i w_{ij}x_i - \theta_j\right)$, so

$$\frac{\partial y_j}{\partial w_{j,TOT}} = \frac{f(w_{j,TOT})}{\partial w_{j,TOT}} \tag{2.13}$$

where we do not write x_{TOT} or θ_j as they are considered constants here. This is why it is important that the transfer function, f be differentiable for back-propagation to be implemented.[7][10]

This is carried out some number of times until the error reaches and begins to fluctuate around a minimum at which point the network can be considered to have been trained and is now ready to use.

3 Method

A link to a GitHub repository containing the code written throughout the course of this project can be found in Appendix H.

3.1 Generation and Fitting of Spectra

3.1.1 Initial Peak Generation and Fitting Code

The code used to generate the spectra was based on code produced by Anna Cestaro, a former undergraduate summer intern at Armagh Observatory who developed code which would generate data made up of Gaussian curves and random noise before attempting to fit Gaussian functions to this data. These Gaussian curves represented spectral lines which had been Doppler broadened. The program would repeat this a set number of times, `Nsim`, for different randomly-determined amplitude-to-noise ratios (A/Ns) and compare the standard deviation of the input data σ_{in} with the standard deviation of the fitted curve σ_{out} by plotting the relative differences of these values $\left(\left(\sigma_{out} - \sigma_{in}\right)/\sigma_{in}\right)$ against the outputted A/N values.

The number of Gaussian curves and their parameters are determined by a text file that is read by the program which then generates the peaks accordingly. The parameters that the code takes in are the position of the peak (u), the standard deviation of the peak (σ) and the amplitude of the peak (A). It then fits the peaks using the python module `lmfit`[11], keeping the same separation between the peaks (Δu) as there was in the input data, and ensuring that σ is the same for all peaks. An example of one of the spectra generated and fit by this code is show in figure 3.1

3.1.2 Modifications Made to the Code

The initial program took an input from a text file of the form shown in table 3.1. Importantly the initial program generated peaks with amplitudes proportional to the amplitude of the first peak, which itself was determined by the randomly generated A/N, and ensured that the spacing between the fitted peaks was the same as the spacing between the peaks that were input, and that all peaks had the same σ.

The first modification that was made was to allow the input of a minimum (A_l) and maximum (A_u) bound for the amplitude, the program would then choose a random amplitude between these two values as the amplitude for the peak. The ability to specify

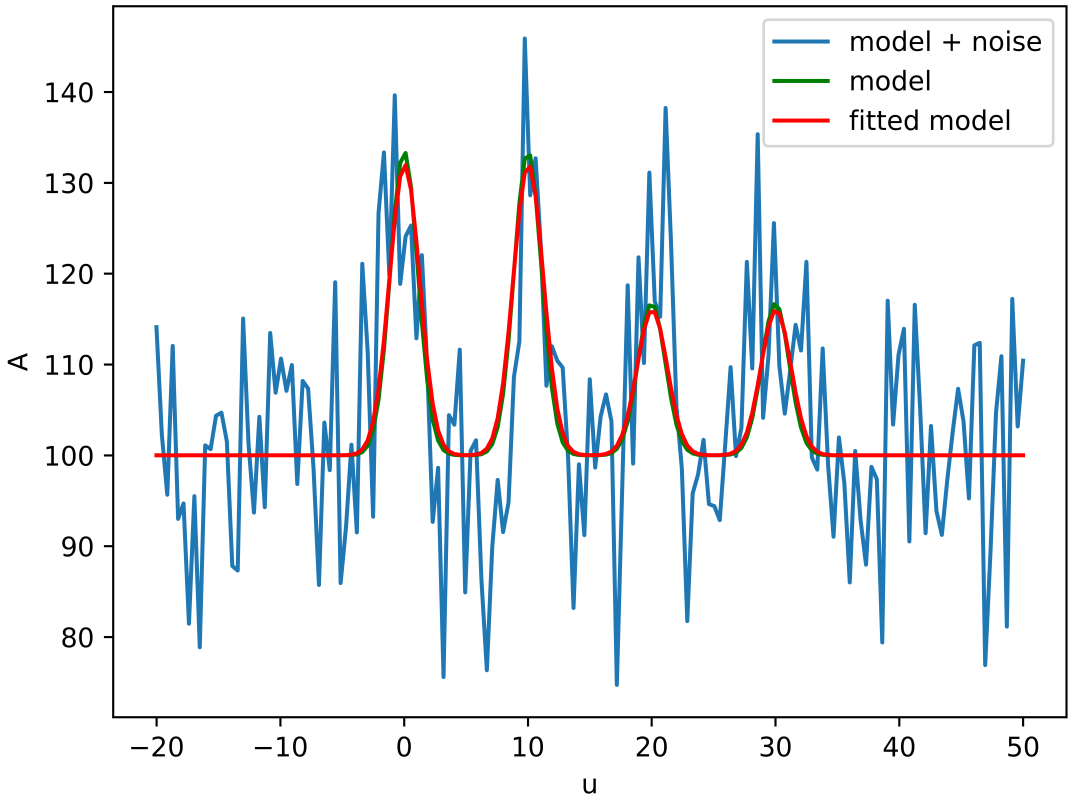


Figure 3.1: A spectrum generated and fit to by the initial code. The spectrum is made up of Gaussian peaks with noise added to them. The units on both axes are arbitrary and represent the amplitude of the signal (A) against position (u).

which line a particular line's amplitude would be relative to, i.e. if it was from the same species as another line, was also added. For example, in table 3.2 in the first column `l` denotes that the amplitude of that peak is independent of any other peak, while `di` means that that the peak's amplitude is given relative to the amplitude of peak `i` (indexing from 0), i.e. in this example the second line, with $u = 10$ will always have an amplitude one third that of the line at index 2 (the third line) whose own amplitude can be between 0 and 0.5 times the A/N.

The ability to give the lines a velocity relative to the observer (and hence a red or blueshift) was added and the program was changed to plotting lines at their actual wavelengths and with realistic ratios between lines arising from the same species. If a line was marked as arising from the same species as another its velocity and line profile were fixed to be the same as those of the main line of that species (marked with `l` in the

Table 3.1: The form of the input file taken by the initial program. The top line was not included in the file and is simply there to show what each value corresponds to.

u	σ	A
0	1	1.0
1	1	0.5
⋮	⋮	⋮

Table 3.2: The form of the input file taken by the first modification to the initial program. Again, the top line was not included in the file and is simply there to show what each value corresponds to.

free or dependant	u	σ	A_l	A_u
d3	0	1	0.0	0.333
d2	10	1	0.333	0.333
l	20	1	0.0	0.5
l	30	1	0.0	1.0

file).

Through a number of iterations of input file designs the use of two separate files, one for inputting the data used to plot the spectrum, and the other for the data used to try to fit the model to the spectrum, was settled on. The structure of the input file used for plotting is shown in table 3.3. Here, `wl` corresponds to the wavelength of the line measured in angstrom, `A_in_l` and `A_in_u` are the upper and lower bounds on the relative amplitudes of the line respectively (while they were always given as the same value throughout the latter part of the work done on this project, the option for them to be different was kept for future use), `v_in` is the velocity of the source relative to the observer in km/s, `sig_in` is the standard deviation of the Gaussian peak in units of km/s, and `free` is the same as the first column in table 3.2.

The file used for fitting the data was laid out in a very similar manner to that used for plotting and is shown in table 3.4. Any columns with the same titles as table 3.3 denote the same information `A_l` and `A_u` are the upper and lower limits for the line's amplitude, any line marked with `l` in `free` is given an infinite upper limit to its amplitude and the other line amplitudes are relative to the lines that they are dependent on, i.e. if the [O III] line at 5006.77 Å is found to have an A/N of 10 then the [O III] line at 4958.83 Å has to have an A/N of 3.35. The columns `v_guess` and `sig_guess` are the initial guesses

Table 3.3: The form of the input file taken by the final iteration of the program. Here, the top line was included in the file. All values of `wl`, `A_in_l` and `A_in_u` were taken from [12].

index	name	wl	A_in_l	A_in_u	v_in	sig_in	free
0	Hb	4861.32	0.349	0.349	100	100	d4
1	OIIIa	5006.77	0.714	0.714	100	100	l
2	OIIIb	4958.83	0.335	0.335	100	100	d1
3	NIIb	6547.96	0.340	0.340	100	100	d5
4	Ha	6562.80	1.0	1.0	100	100	l
5	NIIa	6583.34	0.629	0.629	100	100	l
6	SIIb	6716.31	1.0	1.0	100	100	d7
7	SIIa	6730.68	0.8	0.8	100	100	l

Table 3.4: The form of the fitting file taken by the final iteration of the program. Here, the top line was included in the file. All values of `wl`, `A_l` and `A_u` were taken from [12].

index	name	wl	A_l	A_u	v_guess	sig_guess	free	tied
0	Hb	4861.32	0.349	0.349	100	100	d4	t4
1	OIIIa	5006.77	0.0	inf	100	100	l	t4
2	OIIIb	4958.83	0.335	0.335	100	100	d1	t4
3	NIIb	6547.96	0.340	0.340	100	100	d5	t4
4	Ha	6562.80	0.0	inf	100	100	l	f
5	NIIa	6583.34	0.0	inf	100	100	l	t4
6	SIIb	6716.31	1.0	1.0	100	100	d7	v4
7	SIIa	6730.68	0.0	inf	100	100	l	v4

at the velocity and standard deviations of the line respectively. The final column, `tied` can have values of `f`, `vi` or `ti`. `f` means that the line is completely independent of the other lines, `vi` means that the velocity of this line should be the same as that of the line at index `i`, and `ti` means that the velocity and profile of this line should be the same as that of the line at index `i`. This is not an unreasonable restriction to place, as if the spectrum is from a resolved source it is likely that all of the gas in that resolved region is at the same approximate temperature and is moving at the same speed relative to the observer.

An example of one of the spectra generated and fit by the the program is shown in figure 3.2.

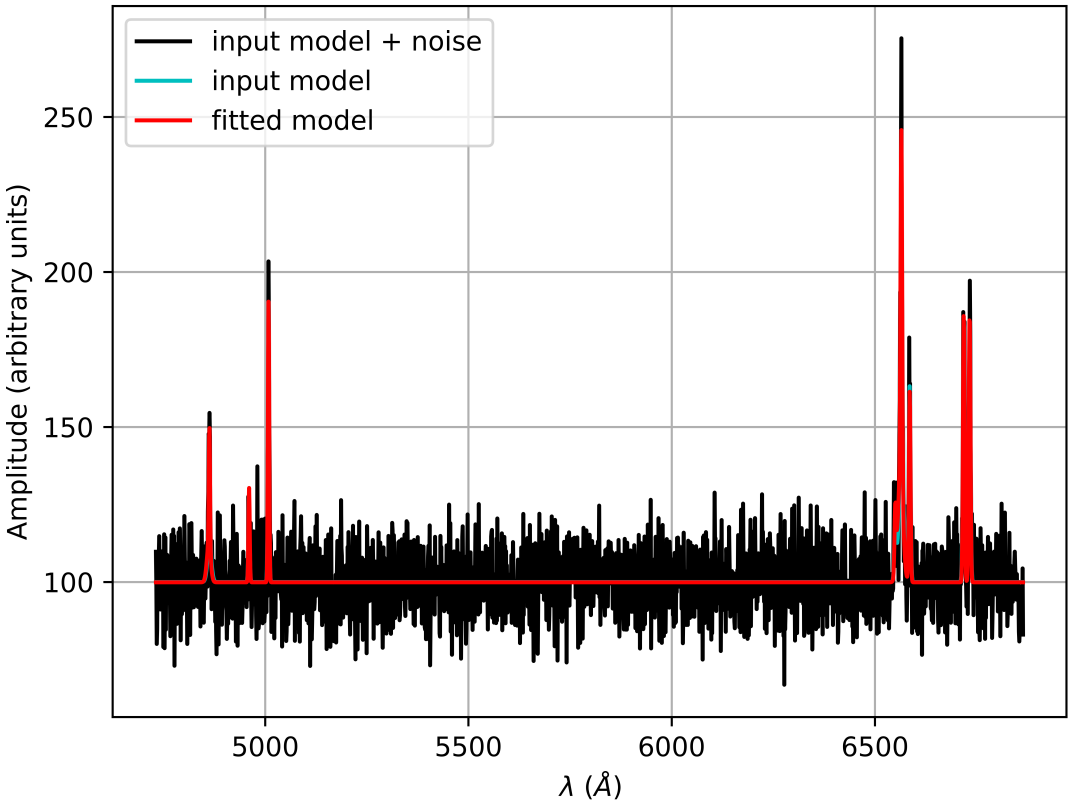


Figure 3.2: A spectrum generated and fit to by the final code.

3.2 False Positive Detection

Using a template based off of code to train an artificial neural network made by Andrew Marshall-Lee, one of the PhD students at Armagh Observatory, a program was made to train and subsequently test neural networks which would be able to detect false positives in fitted spectral data.

A number of different spectra of different A/N (the A/N of the Hα line was randomly chosen from a uniform distribution between 0 and 10 and all other lines were scaled off of that) were generated. Pure noise was generated an equal number of times. A model spectrum was fit to both the spectra and the noise in the way described above. In each case the amplitude to noise ratios of the fitted peaks of each line were taken and stored. The list of lines and wavelengths used was the same as those listed above in the previous section. The lines were also all redshifted by 100 km/s and had standard deviations equivalent to 100 km/s.

Four different sets of ratios of line amplitudes given to me by my supervisor were used which were indicative of: star forming regions, planetary nebulae, LINERs (low-ionisation nuclear emission-line regions, regions in galactic nuclei whose spectra are dominated by forbidden lines from atoms in unionised or low ionisation states[13]) and the active galactic nuclei of Seyfert galaxies. The exact ratios used in each case can be found in table A.1 in Appendix A.

These were then used with TensorFlow[14] a Python library used for machine learning to train neural networks. The amplitudes of the actual spectra were fed in and given an expected output of 1 (detection of a real line) and the amplitudes of the fitted noise were given an expected output of 0 (no detection of a line). The networks were allowed to train for some number of epochs and were then tested. An epoch is simply an iteration over the entire dataset split into a number of "batches". Networks were trained on one set of line ratios separately, and on a combination of all line ratios. Those that were trained separately were trained on a total of 20,000 inputs, 10,000 real synthetic spectra and 10,000 sets of noise, this meant that all of the networks had $n_1 = 408$ nodes in the first hidden layer and $n_2 = 82$ in the second hidden layer according to equations 2.6 and 2.7 respectively.

The networks themselves had four layers, two hidden layers which were chosen based on the recommendation of Andrew, as well as the input and output layers. The input layer had 8 neurons, one for the amplitude of each of the lines and the output layer had one neuron which would report a value between 0 (definitely pure noise) and 1 (definitely a real spectrum of the eight lines). The hidden layers had differing numbers of nodes depending on the number of input samples as given by equations 2.5, 2.6 and 2.7. The output layer had a sigmoid activation function while the others had tanh activation functions.

After the networks were trained they were then tested using unseen data which was generated in the same manner as the data it was trained on in order to see how effective they were at detecting false positives, each network was tested using 56,000 inputs, half of which were real spectra with the other half being noise. Each spectrum was classified

as a true positive, a true negative, a false positive or a false negative based on the activation of the final neuron in the network. If the final activation was above a certain cut-off, C, that spectrum was classified as a real one, and below that it was classified as a false detection. A number of cut-offs were tested to find the optimal value for each neural network which would minimise the total number of incorrect classifications for that particular network.

3.3 Recovery of Line Parameters

In order to determine at which low values of A/N the features of the lines could be recovered accurately a number of different spectra of varying A/N were again generated, using the parameters given in tables B.1 and B.2 in Appendix B.

From these fits the relative difference in the input values of the flux, F, velocity, v, and standard deviation of the Gaussian, σ, were calculated i.e. for the flux, $(F_{out} - F_{in})/F_{in}$ was calculated where F_{out} and F_{in} are the flux of the fitted line and the flux of the input line respectively. Corresponding values were also found for v and σ.

The flux here is simply the area under the emission line and above the background level and in the case of a Gaussian line of the form $f(x) = Ae^{-x^2/2\sigma^2}$ as in this project is given by

$$F = A\sigma\sqrt{2\pi} \tag{3.1}$$

where A is the amplitude of the line (see Appendix C for the derivation of this).

The relative differences in these values for each fitted spectrum were placed into a grid with the A/N of the Hα line (the brightest line in the initial spectrum) on the x-axis and the A/N of the line of interest on the y-axis. These were then plotted as a scatter plot on the same grid, with the size of each point indicative of the number of values in each grid box. Two of these plots were made for each combination of line and value, one where the colour of each point indicated the median of the values in the grid box, the other using colour to indicate the standard deviation of the values in the grid box.

As previously mentioned the ratios of the amplitudes of certain lines were fixed by the fitting file, also the standard deviations of the Gaussian profiles as well as the redshift velocities of the lines were fixed to all be the same. This part was repeated without that functionality to see how the fixing of these parameters affected the recovery of the line parameters.

An interactive tool was also made, allowing any point on the scatter plot to be clicked to display each of the generated spectra as points on a graph of the relative difference against the A/N of the lines under investigation, with the particular spectra represented by that point highlighted. Each of these points could also be clicked to investigate that particular spectrum.

4 Results

4.1 False Positive Detection

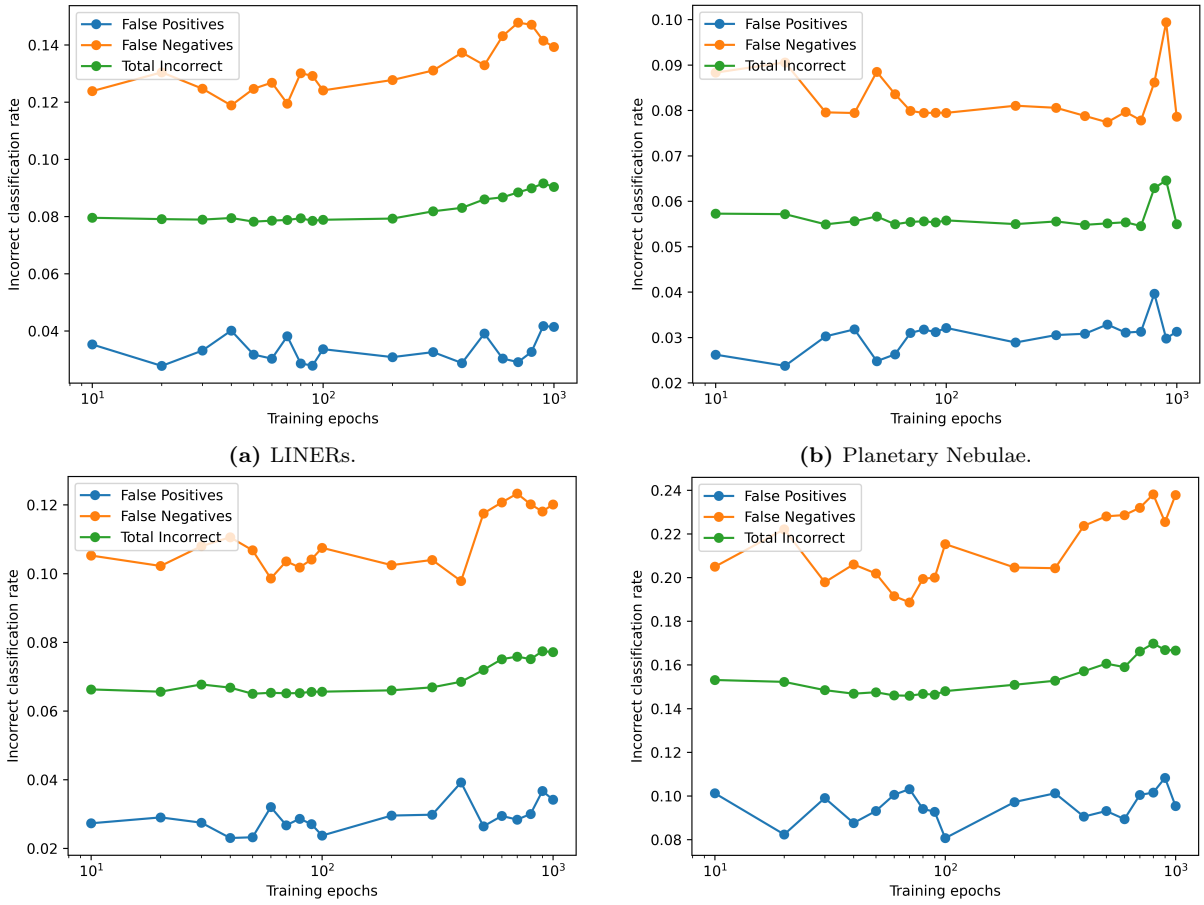
A number of neural networks were generated and tested for each of the sets of line ratios. The performance of a network was judged based on the percentage of false positives and negatives (FP and FN respectively) it incorrectly classified. Initially 500 or 1000 epochs were used in the training however it was noticed that increasing the number of training epochs from 500 to 1000 did not substantially increase the accuracy of the networks, in fact it often-times decreased the overall accuracy of the network, see table 4.1 for an illustration of this. Increasing the number of training epochs did increase the value of the output activation cut-off, C, above which an input would be considered to be a spectrum and the incorrect classification would be minimised.

Upon noticing this it was decided to train networks over a range of numbers of epochs, in steps of 10 over the range 10 to 100 inclusive and then in steps of 100 up to 1000. The rate of incorrect classifications against the number of training epochs for the different neural networks is show in figure 4.1.

These graphs show that for each type of data a fairly consistent proportion of false

Table 4.1: The initial neural networks, the percentage of inputs which were incorrectly classified the opposite type of input (false positives (FP) and false negatives (FN)) and the number of training epochs used to train the network are shown. C is the cut-off above which an input is classified as a real spectrum and which would give a minimal incorrect classification rate.

Network	FP (%)	FN (%)	C	Epochs
LINER_ANN_0	3.293	14.193	0.649	500
planetary_nebula_ANN_0	3.300	8.246	0.589	500
seyfert_ANN_0	2.839	11.329	0.732	500
star_forming_ANN_0	9.443	22.939	0.580	500
LINER_ANN_1	3.779	14.225	0.920	1000
planetary_nebula_ANN_1	3.496	9.350	0.736	1000
seyfert_ANN_1	3.246	12.154	0.895	1000
star_forming_ANN_1	10.454	23.357	0.711	1000



(a) LINERs.

(b) Planetary Nebulae.

(c) Seyfert galaxies.

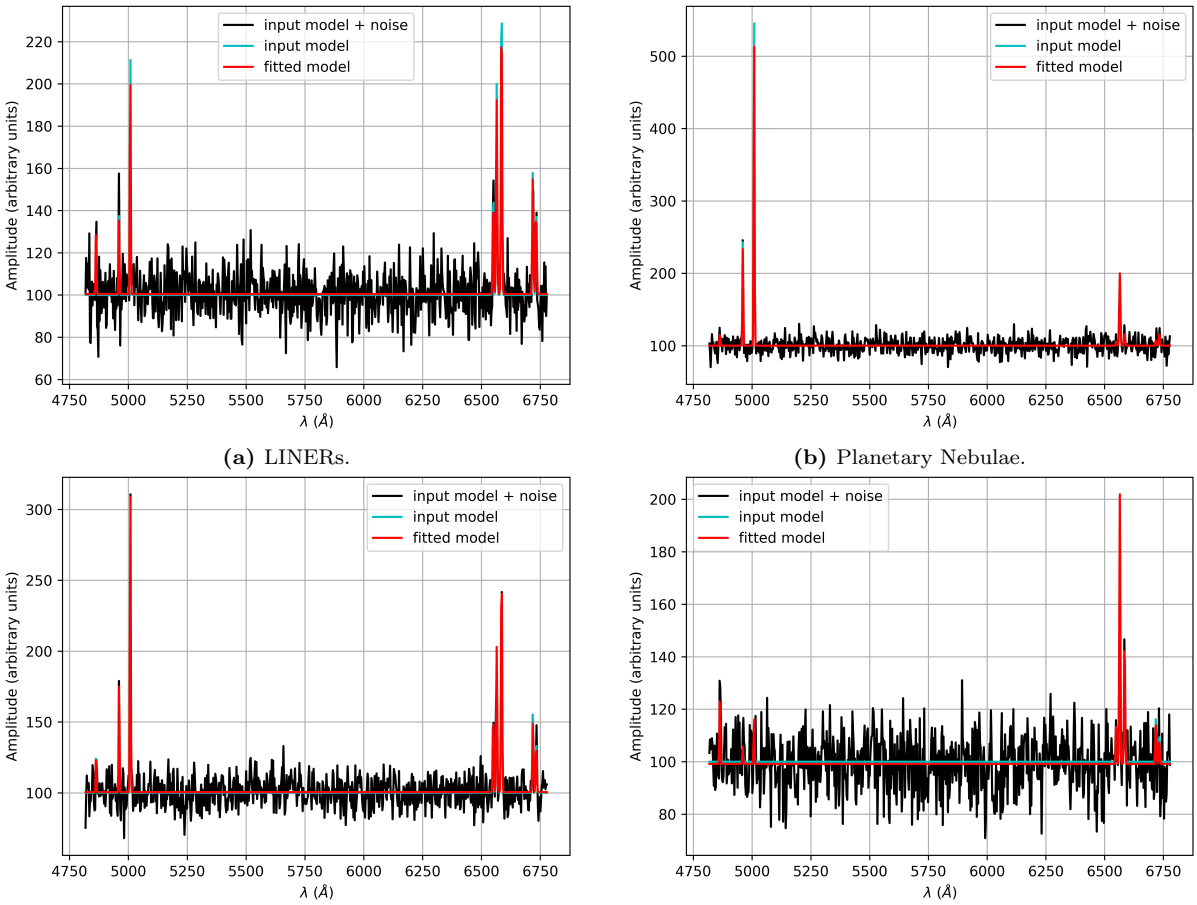
(d) Star forming regions.

Figure 4.1: The proportion of incorrect classifications by the neural networks, the labels underneath each show the types of data they were trained and tested on.

positives are reported, with some oscillation. With regards to the number of false negatives reported for all sets of data but the planetary nebulae, figure 4.1b, there appears to be a general increase in the number of false negatives as the number of training epochs increases. For the planetary nebula data there is a relatively consistent rate of false negatives reported except for a spike at 800 and 900 epochs, however at 1000 epochs it had returned to approximately the same rate as previous, unfortunately, due to the 'black-box' nature of neural networks the reason for this can't be effectively determined. The numerical values of the performance of these networks along with the values of the cut-off that gave the best performance are shown in table D.2 in Appendix D.

The small oscillations can likely be explained as the model oscillating around the minimum of its error function, and at the specific arbitrary point the training was stopped it was on one side or the other of the error function. The increase in incorrect classifications as the number of epochs increases may be an example of 'overfitting'. Overfitting can occur when a network is too complex, or when it is tried too hard to find the best fit to the data, and instead of finding a general rule it instead memorises the specific features or "peculiarities of the training data"[15].[16] This could be compared to a child learning to read who, rather than learning what sound each letter makes and what the words mean, has instead memorised each page of a book that has been read to them many times over and so can 'read' that book but no others, or a dog that has learned 'go to bed' means to go to their own bed, but if one were to say to it 'go to John's bed' it would not understand what was meant even if it knew that John was its owner. In each case they have learned how to respond to specific stimuli but do not understand the meaning behind said stimuli.

This does raise the question of if perhaps all of these networks have been overfitted. All of the data, both the training and testing data, was generated in the same manner, so if there are any peculiarities of the training dataset it is quite possible that they also exist in the testing data. The solution to this would be to test with actual recorded data, or failing that, data that is generated in a different manner – perhaps simply by repeating this process using a different library or even language altogether, or by using



(a) LINERs.

(b) Planetary Nebulae.

(c) Seyfert galaxies.

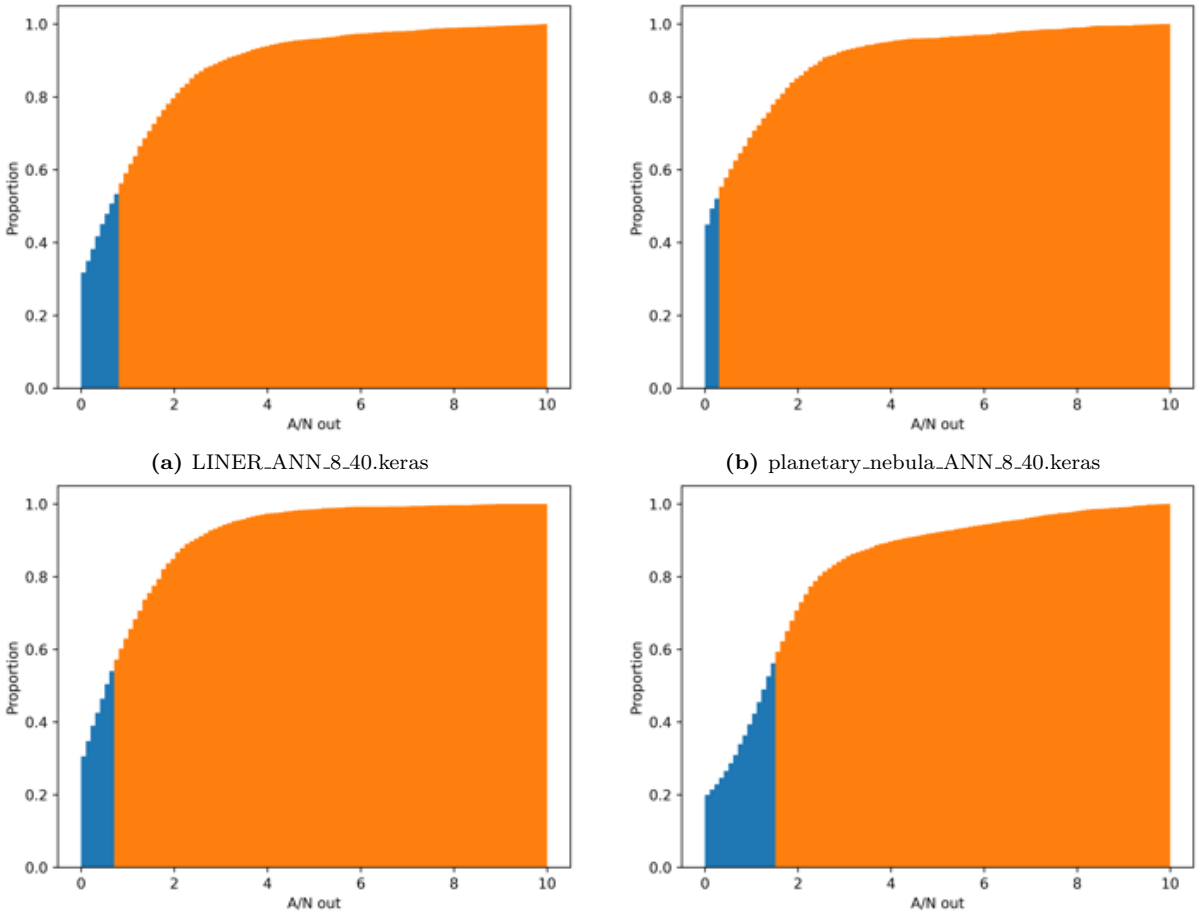
(d) Star forming regions.

Figure 4.2: The spectra of the different types of regions used in the training of neural networks. In each spectrum the A/N of the Hα line is 10 and the other lines are scaled according to the values in table A.1.

randomly generated numbers rather than actually generating and fitting spectra.

The networks also become very accurate after a small numbers of training epochs. This is likely because of the training method, 20,000 datasets are fed in, 20% are put aside for validation checking, leaving 16,000 datasets to train against, in batches of 64. This means that there are 250 training sessions per epoch, so it is not necessarily surprising that the networks are so accurate after such a small number of training epochs.

The lack of consistency in rates of incorrect classification across the different sets of data, most notably the high proportion of incorrect classifications of star forming region data, is likely due to the relative strengths of the different lines in each region, see figure 4.2 for an example of each spectrum. See also table A.1 for the relative amplitudes of



(a) LINER_ANN_8_40.keras

(b) planetary_nebula_ANN_8_40.keras

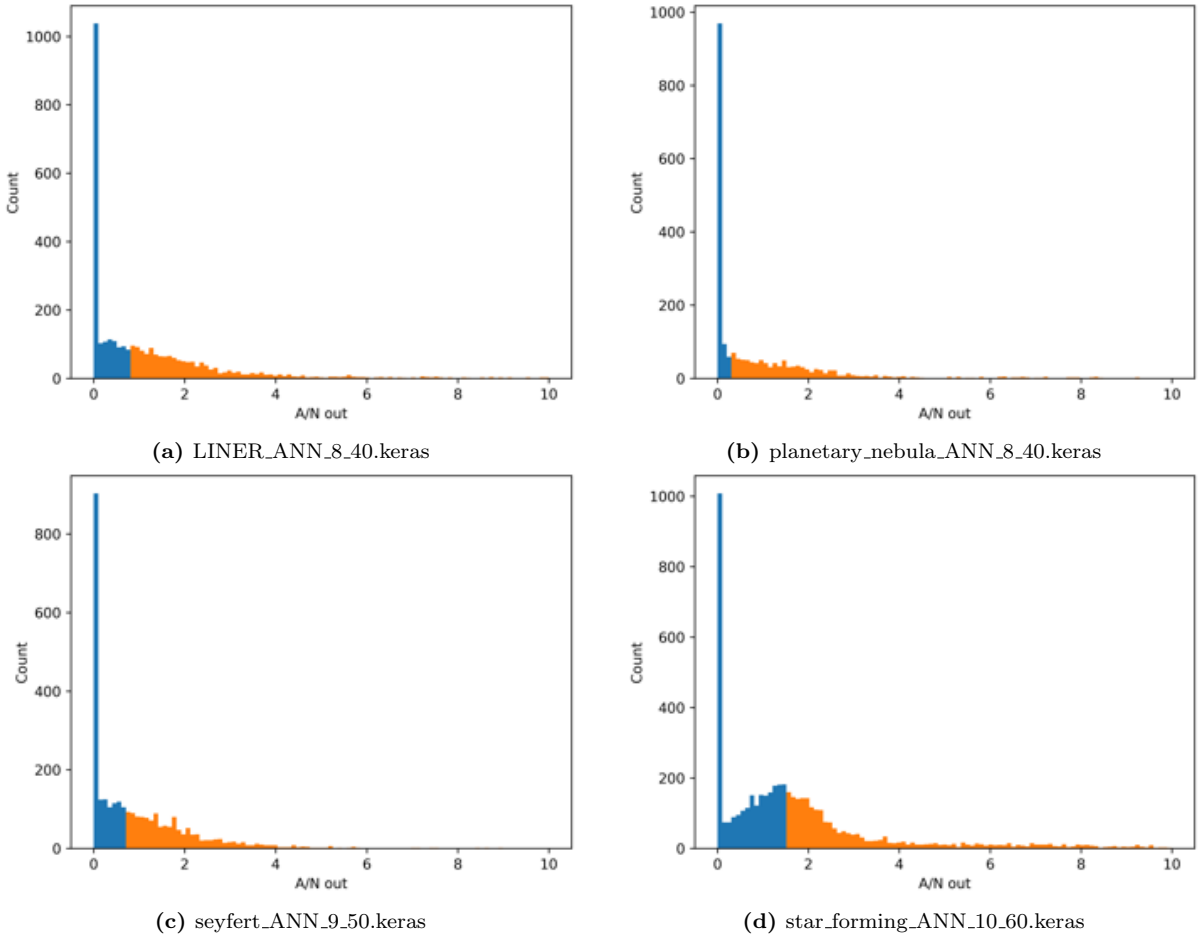
(c) seyfert_ANN_9_50.keras

(d) star_forming_ANN_10_60.keras

Figure 4.3: The A/N of the Hα line in spectra which were falsely reported by selected neural networks as being noise. The bars in blue are the half with lower A/N and the bars in orange are the half with higher A/N.

each of the lines in each type of data.

When generating the data, the Hα line was given an A/N of between 0 and 10, which means that for the planetary nebula data for example, when the Hα line has an A/N of 1 the two [O III] lines have A/Ns of approximately 4.8 and 1.5, however for the star forming regions, the region with the worst performing associated neural network when the Hα line has an A/N of 1 the strongest line other than Hα, which is [N II] line at $\lambda = 6547.96$ Å, has an A/N of only approximately 0.4, which is unlikely to be detected. As such the networks can correctly identify false positives at lower Hα A/N for planetary nebulae than for star forming regions. For example, in all of the spectra in figure 4.2 the



(a) LINER_ANN_8_40.keras

(b) planetary_nebula_ANN_8_40.keras

(c) seyfert_ANN_9_50.keras

(d) star_forming_ANN_10_60.keras

Figure 4.4: The A/N of the Hα line in spectra which were falsely reported by selected neural networks as being noise. The bars in blue are the half with lower A/N and the bars in orange are the half with higher A/N. This shows the same data as figure 4.3 in a non-cumulative manner.

A/N of the Hα line is 10, the highest value that was generated, however in the spectrum from the star forming region, the other lines, apart from the aforementioned [N II] line and possibly the Hβ line, are seen to be at amplitudes on par with the noise.

There is also an at first alarmingly high proportion of false negatives reported in all datasets, with the rate of false negatives consistently being close to or above 10%, however as shown in figure 4.3 the majority of these are at very low A/N for all of the networks. For all networks 50% of the false negatives had a Hα line whose A/N was ≤ 2, and for the majority of the networks which were not trained on star forming region data that can be lowered to ≤ 1. It can also be seen in figure 4.4 that there are very few false negatives in which the A/N of the Hα line has an A/N ≥ 3. The cumulative

and non-cumulative histograms for each of the other neural networks can be found in Appendices E and F respectively.

The fact that these false negatives are overwhelmingly due to spectra in the lower range of A/N shows that these neural networks do work at detecting spectra, and their low rate of false positive reports suggests that they may be better than simple visual inspection or other simpler plotting methods, especially in the case of large datasets.

4.2 Recovery of Line Parameters

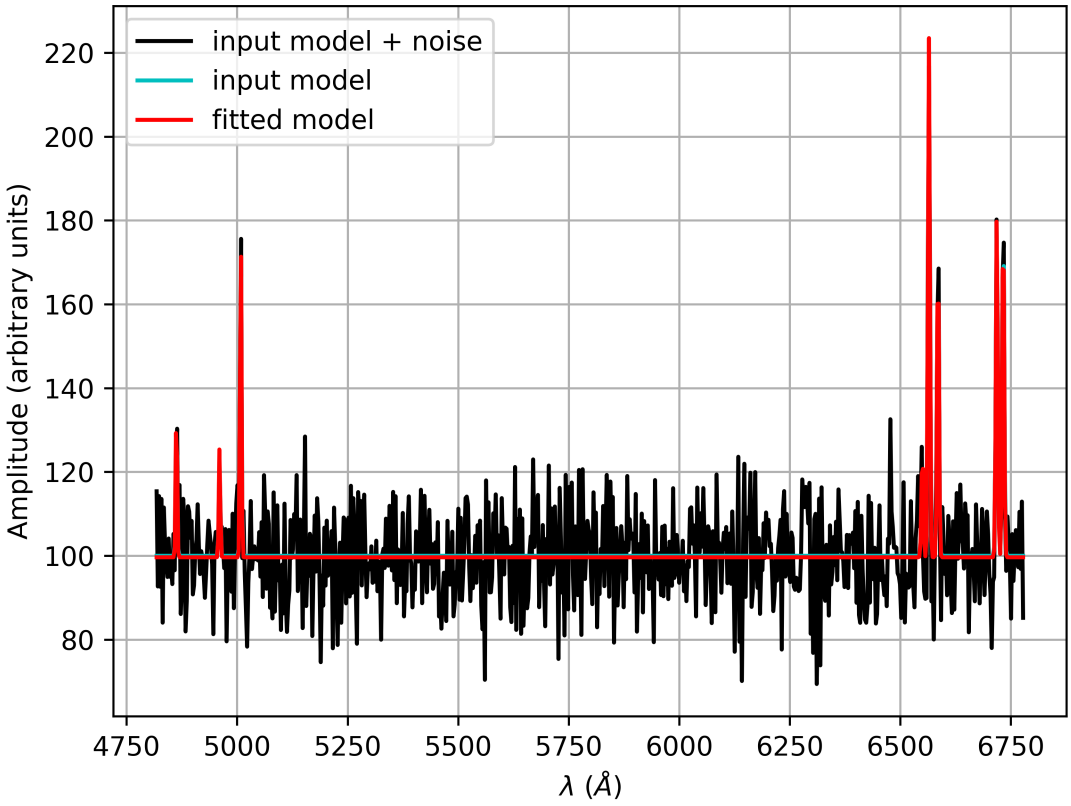


Figure 4.5: An example of one of the spectra used in this section. In this spectrum the input Hα line has an A/N of 10.

A total of 12000 simulated spectra were generated and fit, 6000 for the lines of fixed ratios, and the same velocities and standard deviations, and 6000 for the lines without said fixed ratios.

Dealing first with the lines with fixed ratios, as mentioned the spectra were plotted

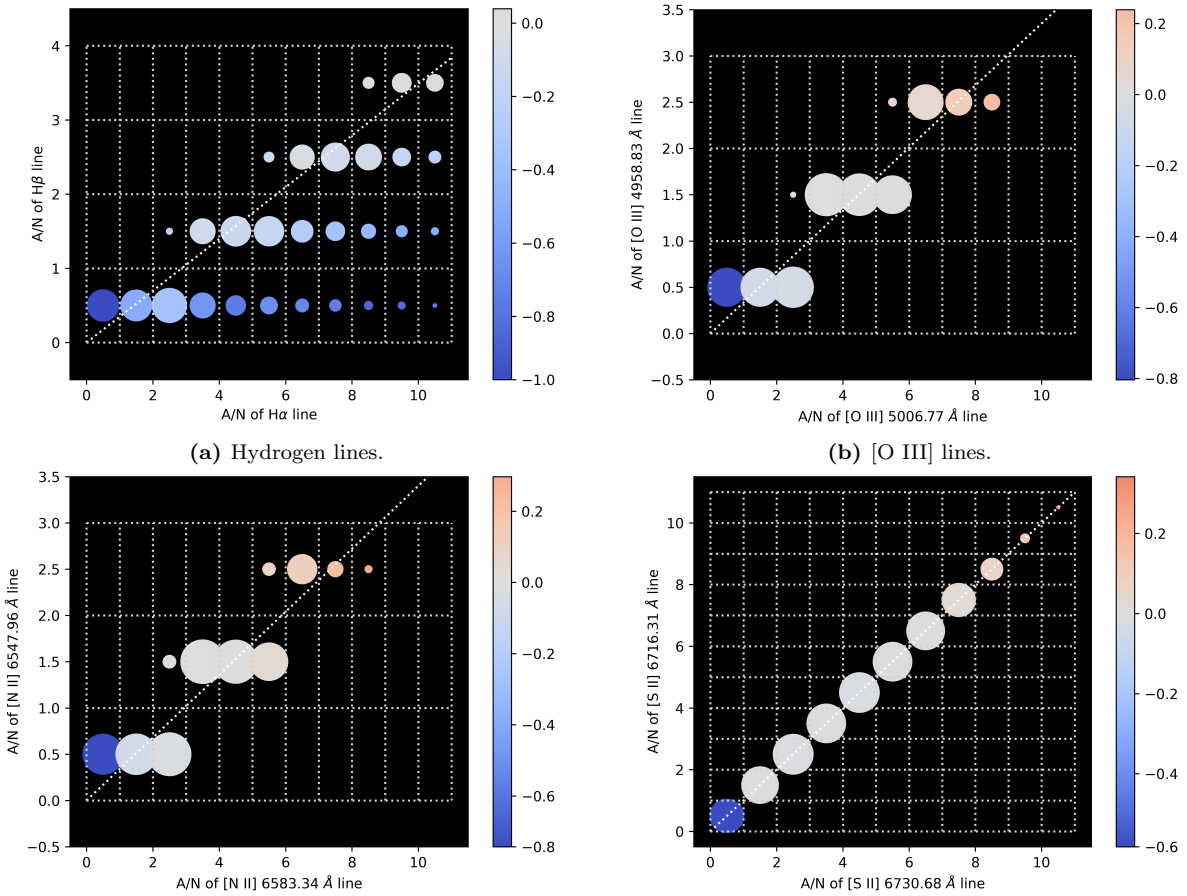
with the line ratios given in tables B.1 and B.2. Figure 4.5 shows an example of one of these spectra.

The scatter plots in which the size of the point represented the number of spectra in each grid area were generated with the A/N of the "main" line for each species on the x-axis and the the A/N of the other line on the y-axis. These scatter plots were chosen over other potential plots such as simply a grid with a colour map on it as it was felt the number of points in each grid space was necessary to know, as for example a grid space representing only two spectra which overall had a median $(F_{out} - F_{in})/F_{in}$ of 2 should be treated and represented differently than a grid space representing two hundred spectra with the same median $(F_{out} - F_{in})/F_{in}$ value. The scatter plots representing the medians of the relative differences in the fluxes are shown in figure 4.6, while the standard deviations of the same are shown in figure 4.7. The plots for the other parameters, the velocities and standard deviations, can be found in Appendix G.

As can be seen from the plots, for the fluxes of all of the species apart from hydrogen when the A/N of the 'main' line (the one represented on the x-axis) is > 1 the median difference between input and output tends to be around 0. For hydrogen this occurs around A/N > 2 for Hα but only when the A/N of Hβ is greater than 1. In all species the medians are the furthest departed from 0 and the standard deviations are highest for lower values of A/N. All of this is also true for the other parameters: the velocity of the lines and their profiles.

For the [O III] and [N II] lines as the medians approach 0 the standard deviations become lower as well, however for the hydrogen and [S II] lines this is not the case, here the corresponding standard deviations are noticeably higher and this is due to a number of large outliers increasing the variance in the data.

Note that for all species but hydrogen the points only appear in boxes intersected by the dotted line (while it is difficult to see this for some of the points, the dotted line does at least pass through the corner of these boxes), while for hydrogen it is in those boxes and below them, this is because while all species had a fixed ratio at which they generated, the non-hydrogen species had a fixed ratio at which they were fit, while



(a) Hydrogen lines.

(b) [O III] lines.

(c) [N II] lines.

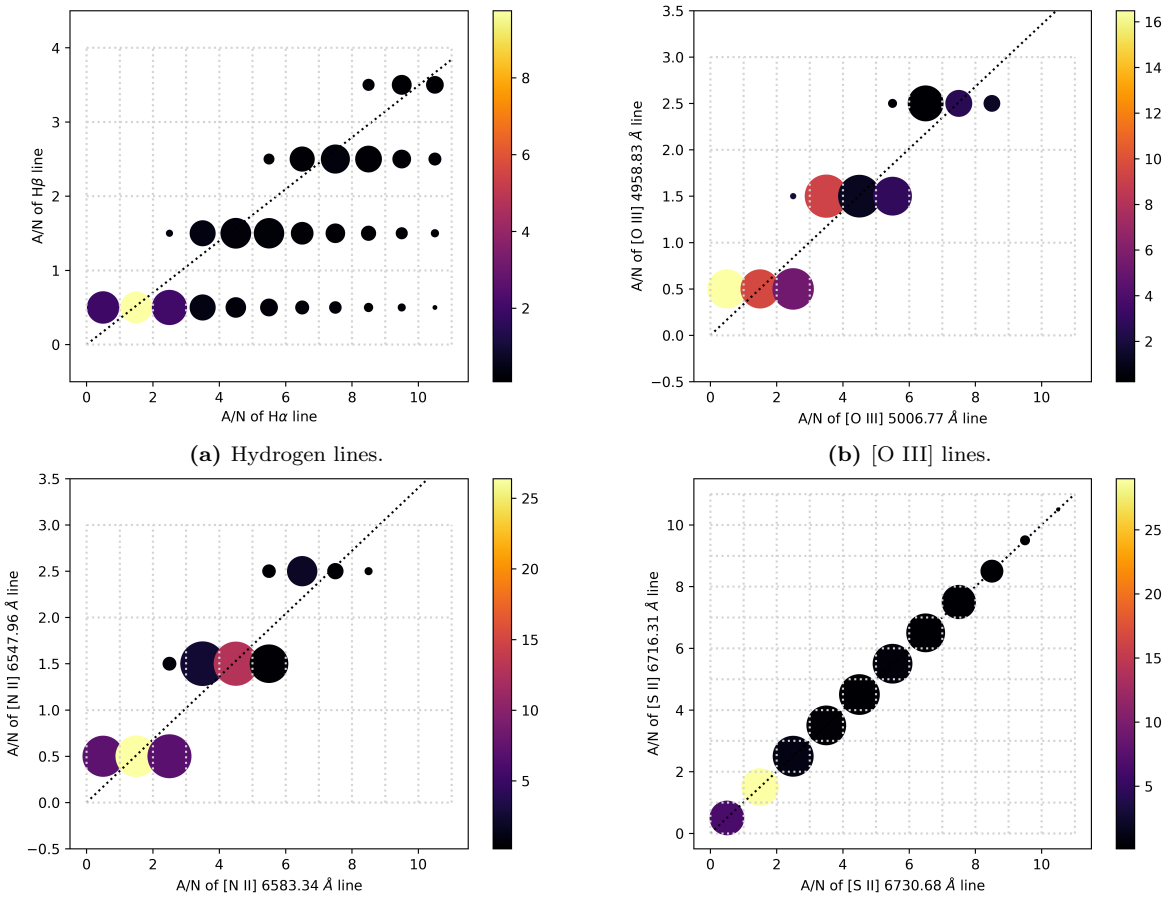
(d) [S II] lines.

Figure 4.6: The medians in the relative difference in the input and output fluxes for lines of different species against the A/N of the lines. The size of a point represents the number of points in that that grid square. The dotted line through each figure shows the expected ratio of the lines.

hydrogen only had an upper limit to the ratio. This was to account for the fact that due to reddening and the large λ difference between the two hydrogen lines the observed ratio of the lines cannot be exactly predicted, while for other species, because there is such a small difference in the wavelengths of their observed lines there is approximately the same amount of reddening for each line so the effects of this can be ignored.

When this is compared to the results of the simulations which had each line fit independently some clear differences can be seen. Again, the plots for the fluxes can be seen here in figures 4.8 and 4.9, while the plots for the velocity and line profile can be found in Appendix G.

As can be seen in these plots fitting each line independently most noticeably leads



(a) Hydrogen lines.

(b) [O III] lines.

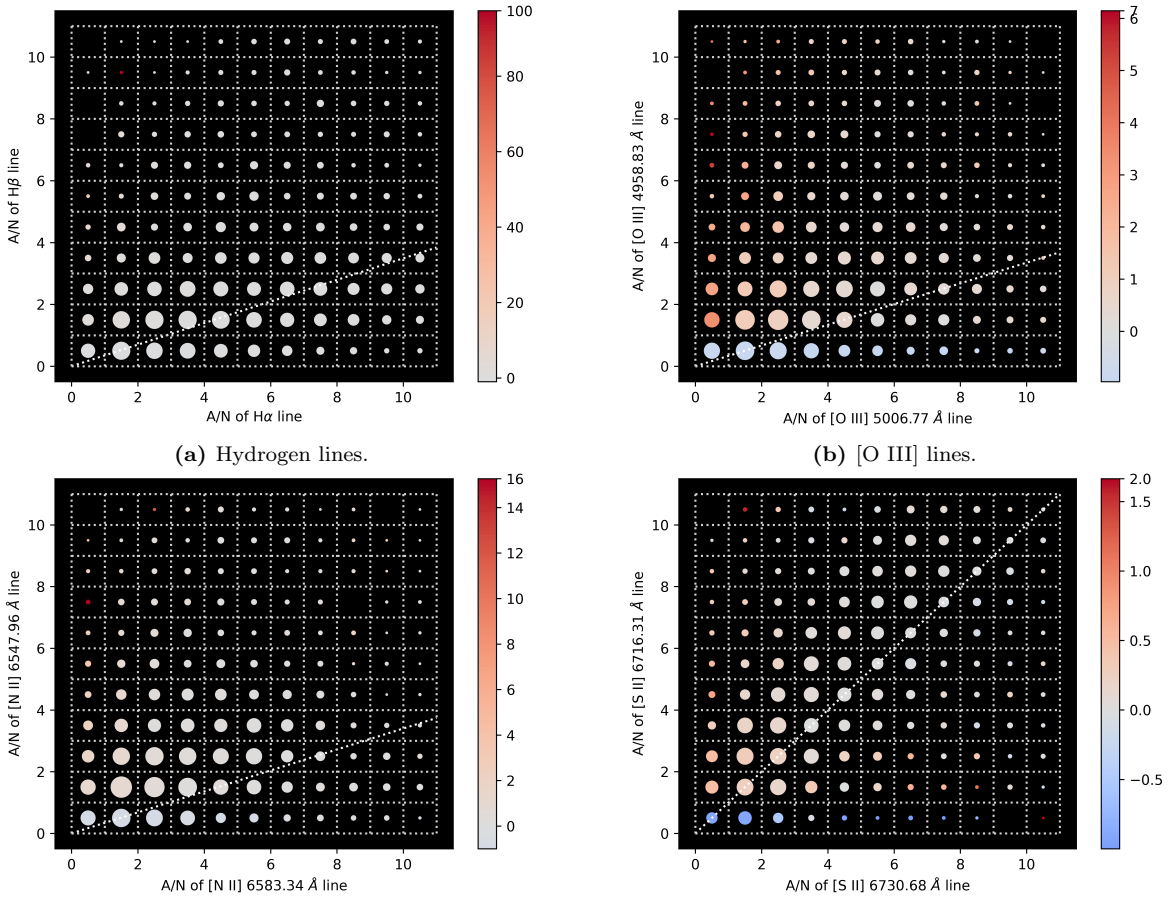
(c) [N II] lines.

(d) [S II] lines.

Figure 4.7: The standard deviations in the relative difference in the input and output fluxes for lines of different species against the A/N of the lines. The size of a point represents the number of points in that that grid square. The dotted line through each figure shows the expected ratio of the lines.

to lines of the same species being fit outside of the expected ratios, however due to the noise present in the spectrum and the fact that the lines are not being confined to certain ratios this is not surprising. What can also be seen and again is not surprising is the spread of the A/N values, there are significantly more grid squares which are populated with data from some lines, however the number of lines each represents is small as shown by the small size of each point.

These plots also show that while the medians do not diverge significantly further from 0 except for some outliers caused by only a few lines, the standard deviations of the data from the median is higher in general. Again, at higher A/N, and closer to the actual ratio of the lines the medians and standard deviations do become closer to 0. Similarly,



(a) Hydrogen lines.

(b) [O III] lines.

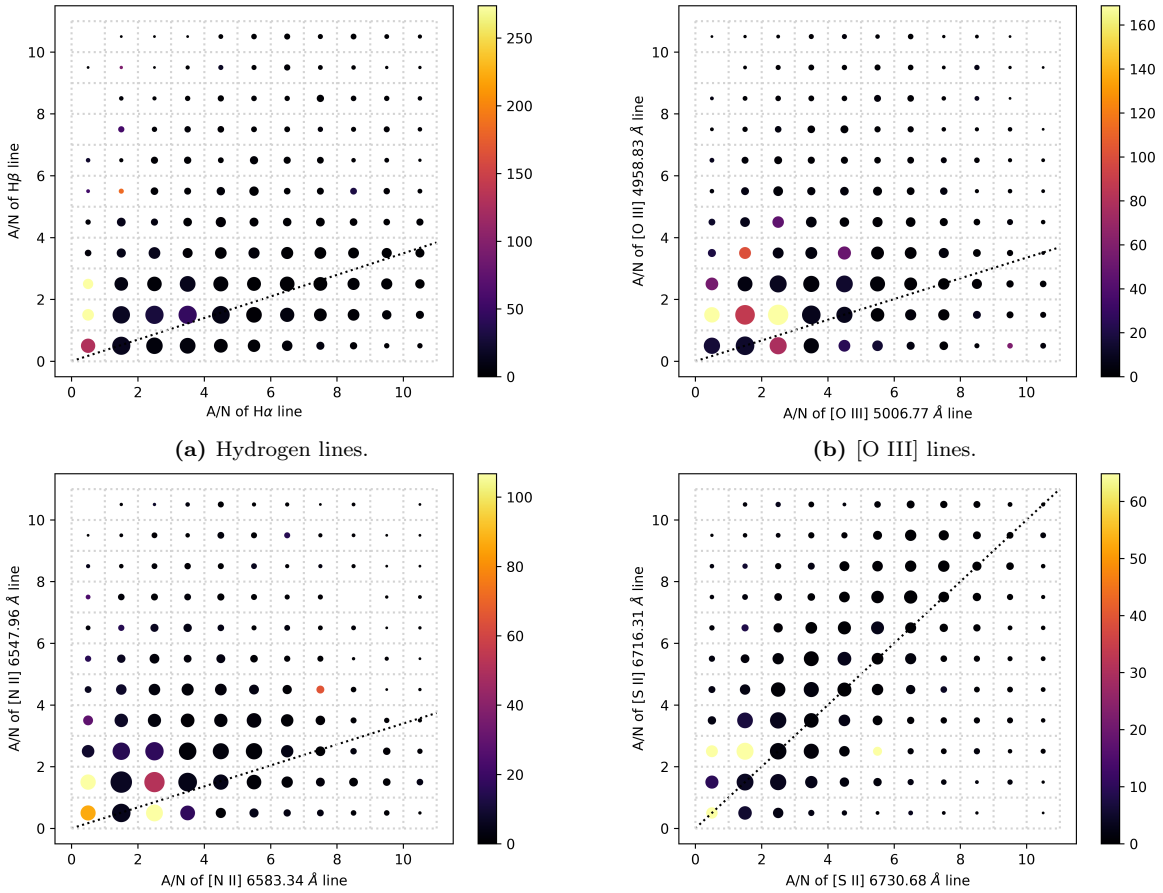
(c) [N II] lines.

(d) [S II] lines.

Figure 4.8: The medians in the relative difference in the input and output fluxes for lines of different species against the A/N of the lines for the lines which were fit independently. The size of a point represents the number of points in that that grid square. The dotted line through each figure shows the expected ratio of the lines.

this is all reflected in the plots of the velocity and standard deviation data which can be found in Appendix G also.

When the fitting was carried out taking into account the ratios in the amplitudes of the lines and fixing their velocities and standard deviations to be the same the parameters of the lines were recovered successfully and more consistently at lower A/N values than they were when that was not taken into account. This is as was expected, as the use of eight spectral lines allows the effects of noise to be reduced. When the ratios and other parameters are not fixed, if there is particularly strong noise in the area of the line it would likely detrimentally affect the recovered parameters, however if the program is



(a) Hydrogen lines.

(b) [O III] lines.

(c) [N II] lines.

(d) [S II] lines.

Figure 4.9: The standard deviations in the relative difference in the input and output fluxes for lines of different species against the A/N of the lines for the lines which were fit independently. The size of a point represents the number of points in that that grid square. The dotted line through each figure shows the expected ratio of the lines.

designed to find the best fit for two or more spectral lines together the effects of noise around one line on the overall fit can significantly reduced, and as a result that, while it may not be completely correct, can be much closer to correct than the result obtained when simply fitting to one line.

The interactive tool was also developed, out of a want to see the spectra of the outliers. Its use is demonstrated in figure 4.10. It would initially produce one of the scatter plots show in this section, then when one point was clicked on it would produced a plot of the relative difference in the input and output values of the parameter against the A/N of the line for both lines under investigation. Then within these plots each point can be

clicked to view the spectrum associated with that point.

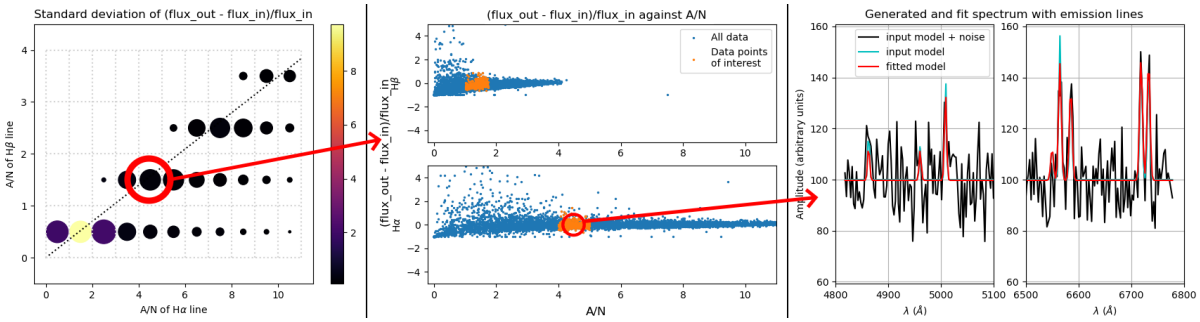


Figure 4.10: The interactive tool, first the point corresponding to a Hα A/N between 4 and 5 and a Hβ A/N between 1 and 2 on the left is clicked, producing the middle plot. The middle plot shows the raw data behind the scatter plots and highlights the points corresponding to the area clicked previously. Then one of these points is clicked and the right plot is produced showing that particular spectrum. The spectrum is split in two simply to allow for easier readability as there is a large gap in the middle of all spectra used in this project which had no features present.

5 Conclusions

Overall this was a success, the neural networks successfully differentiated spectra from noise over 80% of the time in total, and it was shown that the line parameters were recovered accurately at lower A/N when lines were fit with attention paid to the other lines from the same species than when they were fit independently of each other.

A number of neural networks were developed which could correctly distinguish between the synthetic spectra they were trained on and noise for over 80% of the data, and over 90% of the data not based off of star-forming regions, and with datasets which extend to higher amplitude to noise ratios they likely would perform even better as the spectral lines would be more distinct. There is the possibility of overfitting in these networks however without access to real spectra or other synthetic spectra which were generated in a completely different manner this is next to impossible to test. That being said, the next step with any of these networks would be to test them with real spectral data and see how they truly perform in the 'real world' as it were.

Other future work could include training similar networks to distinguish between the four types of spectra, perhaps by repeating the process, except with four outputs to each

network, each signifying one type of spectrum. This could of course be extended to other types of spectra as well, and other spectral lines could be included also.

Currently the data, be it a spectrum or noise, must be fit before it can be tested by a neural network. While this is possible – if even easy – to automate, one could also simply train a network on an entire spectrum. This is something which was considered in the course of this project, however the size of the spectrum i.e. the number of nodes on the input layer, 719 data points in this project (although for other spectra it could even be in the thousands of data points), as compared to the size of the data actually used here, 8 numbers representing the amplitudes of the lines, caused the neural network to take a prohibitively long time to compute so this idea was not further investigated. In addition there is the issue of a loss of generality as the spectra of different instruments are not necessarily going to have the same size of data, however this does not have to be an issue as separate networks could be trained for each instrument.

The recovery of the line parameters was also a success. When the fitting was carried out taking into account the ratios in the amplitudes of the lines and assuming the spectrum came from the same resolved source the parameters of the lines were recovered successfully and more consistently at lower A/N values than they were when that was not taken into account. This was only tested on synthetic spectra also, and while of course finding the relative difference in the input and output values is not possible with real spectra, this method of fitting could perhaps be tested against established methods of fitting real spectra to see how it performs on real data.

Every spectrum in this project had an underlying flat continuum of 100, in principle the continuum should not matter to the fitting process as, as long as it is known, it can be accounted for in the fitting. However if this work is to be used in future then different continua should be tested to verify this.

References

[1] I. Robson, *Active galactic nuclei*. John Wiley and Sons, 1996.

[2] B. T. Draine, *Physics of the Interstellar and Intergalactic Medium*. Princeton University Press, 2011.

[3] S. Courteau *et al.*, "Galaxy masses," *Reviews of Modern Physics*, vol. 86, no. 1, pp. 47–119, Jan. 2014.

[4] D. E. Osterbrock and G. J. Ferland, *Astrophysics of Gaseous Nebulae and Active Galactic Nuclei*. University Science Books, 2006.

[5] D. Emerson, *Interpreting Astronomical Spectra*. John Wiley and Sons, 1996.

[6] A. K. Pradhan and S. N. Nahar, *Atomic Astrophysics and Spectroscopy*. Cambridge University Press, 2011.

[7] A. J. Maren, C. T. Harston, and R. M. Pap, *Handbook of Neural Computing Applications*. Academic Press, Inc., 01 1990.

[8] F. Rosenblatt, "Perceptron simulation experiments," *Proceedings of the IRE*, vol. 48, no. 3, pp. 301–309, 1960.

[9] G.-B. Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," *IEEE transactions on neural networks*, vol. 14, no. 2, pp. 274–281, 2003.

[10] P. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[11] M. Newville *et al.*, *lmfit/lmfit-py: 1.3.2*. Zenodo, Jul. 2024. [Online]. Available: https://doi.org/10.5281/zenodo.12785036

[12] D. E. Osterbrock, *Astrophysics of Gaseous Nebulae and Active Galactic Nuclei*. Mill Valley, Calif: University Science Books, 1989.

[13] T. M. Heckman, "An Optical and Radio Survey of the Nuclei of Bright Galaxies - Activity in the Normal Galactic Nuclei," A&A, vol. 87, p. 152, Jul. 1980.

[14] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[15] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM Comput. Surv.*, vol. 27, no. 3, p. 326–327, Sep. 1995. [Online]. Available: https://doi.org/10.1145/212094.212114

[16] M. M. Bejani and M. Ghatee, "A systematic review on overfitting control in shallow and deep neural networks," *Artificial Intelligence Review*, vol. 54, no. 8, pp. 6391–6438, 2021.

A Line Ratios of Different Regions

The line amplitude ratios used when generating spectra for each of the different regions in section 3.2 False Positive Detection are shown in table A.1 below.

Table A.1: The line amplitude ratios in each of the four regions modelled, all values are given relative to the amplitude of the Hα line. The 'a' and 'b' labelling of the lines from the same species are simply my own labelling to be able to refer to each line easily.

Line	λ (Å)	LINER	Planetary Nebula	Seyfert	Star Forming
Hβ	4861.32	0.3674	0.1895	0.3150	0.2851
[O III]a	5006.77	1.1941	4.7768	2.2272	0.1602
[O III]b	4958.83	0.3914	1.4966	0.7780	0.0552
[N II]b	6547.96	0.4983	0.0597	0.5185	0.1300
Hα	6562.80	1.0000	1.0000	1.0000	1.0000
[N II]a	6583.34	1.4963	0.1734	1.5510	0.3819
[S II]b	6716.31	0.6109	0.0610	0.5786	0.1707
[S II]a	6730.68	0.4278	0.1404	0.3815	0.1088

B Plotting and Fitting Parameters for Recovery of Line Parameters

The files used to plot and then fit the spectra for section 3.3 are shown in tables B.1 B.2 and below.

Table B.1: The file used to plot the spectra for the investigation of the recovery of the line parameters at different A/N.

index	name	wl	A_in_l	A_in_u	v_in	sig_in	free
0	Hb	4861.32	0.349	0.349	100	100	d4
1	OIIIa	5006.77	0.714	0.714	100	100	l
2	OIIIb	4958.83	0.335	0.335	100	100	d1
3	NIIb	6547.96	0.340	0.340	100	100	d5
4	Ha	6562.80	1.0	1.0	100	100	l
5	NIIa	6583.34	0.629	0.629	100	100	l
6	SIIb	6716.31	1.0	1.0	100	100	d7
7	SIIa	6730.68	0.8	0.8	100	100	l

Table B.2: The file used to fit the spectra for the investigation of the recovery of the line parameters at different A/N.

index	name	wl	A_l	A_u	v_guess	sig_guess	free	tied
0	Hb	4861.32	0.0	0.349	100	100	d4	t4
1	OIIIa	5006.77	0.0	inf	100	100	l	t4
2	OIIIb	4958.83	0.335	0.335	100	100	d1	t4
3	NIIb	6547.96	0.340	0.340	100	100	d5	t4
4	Ha	6562.80	0.0	inf	100	100	l	f
5	NIIa	6583.34	0.0	inf	100	100	l	t4
6	SIIb	6716.31	1.0	1.0	100	100	d7	t4
7	SIIa	6730.68	0.0	inf	100	100	l	t4

C Gaussian Integral Derivation

The integral of a Gaussian function of the form $f(x) = Ae^{-x^2/2\sigma^2}$ is given by

$$F(x) = \int_{-\infty}^{\infty} dx A e^{-\frac{x^2}{2\sigma^2}} \ . \tag{C.1}$$

Taking a corresponding function of y, $g(y) = Ae^{-y^2/2\sigma^2}$, it is clear that the integral of this function $g(y)$ with respect to y is equivalent to the integral of $f(x)$ with respect to x. This gives us

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} dxdy f(x)g(y) = F(x)G(y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} dxdy A^2 e^{-\frac{x^2+y^2}{2\sigma^2}} \ . \tag{C.2}$$

Transforming to polar coordinates,

$$F(x)G(y) = A^2 \int_0^{2\pi} d\theta \int_0^{\infty} dr e^{-\frac{r^2}{2\sigma^2}} r = 2\pi A^2 \int_0^{\infty} dr e^{-\frac{r^2}{2\sigma^2}} r \tag{C.3}$$

where $r^2 = x^2 + y^2$. Then through u substitution with $u = -r^2/2\sigma^2$ and

$$du = -\frac{r}{\sigma^2} dr \implies -\sigma^2 du = r dr \tag{C.4}$$

giving

$$F(x)G(y) = -2\pi\sigma^2 A^2 \int_0^{-\infty} du e^u \tag{C.5}$$

which can be easily solved to give

$$F(x)G(y) = 2\pi\sigma^2 A^2 \tag{C.6}$$

clearly in one dimension, $F(x)G(y) \equiv F(x)^2$, meaning

$$F(x) = \int_{-\infty}^{\infty} dx A e^{-\frac{x^2}{2\sigma^2}} = A\sigma\sqrt{2\pi}\ . \tag{C.7}$$

D Neural Network Performance

The overall performance of each of the neural networks generated in this project is shown in the tables below. In each table FP and FN denote the percentage of false positives and negatives respectively that the neural network reports after being tested on the data, C is the cut-off value above which data is considered to represent a real spectrum, and Epochs is the number of epochs used in the training of each network.

Table D.1 shows again the initial neural networks and their performance.

Table D.1: The initial neural networks, this is simply a reproduction of table 4.1.

Network	FP (%)	FN (%)	C	Epochs
LINER_ANN_0	3.293	14.193	0.649	500
planetary_nebula_ANN_0	3.300	8.246	0.589	500
seyfert_ANN_0	2.839	11.329	0.732	500
star_forming_ANN_0	9.443	22.939	0.580	500
LINER_ANN_1	3.779	14.225	0.920	1000
planetary_nebula_ANN_1	3.496	9.350	0.736	1000
seyfert_ANN_1	3.246	12.154	0.895	1000
star_forming_ANN_1	10.454	23.357	0.711	1000

Table D.2 shows the other neural networks and their performance.

Table D.2: The other trained neural networks arranged by the data type they were trained on and for.

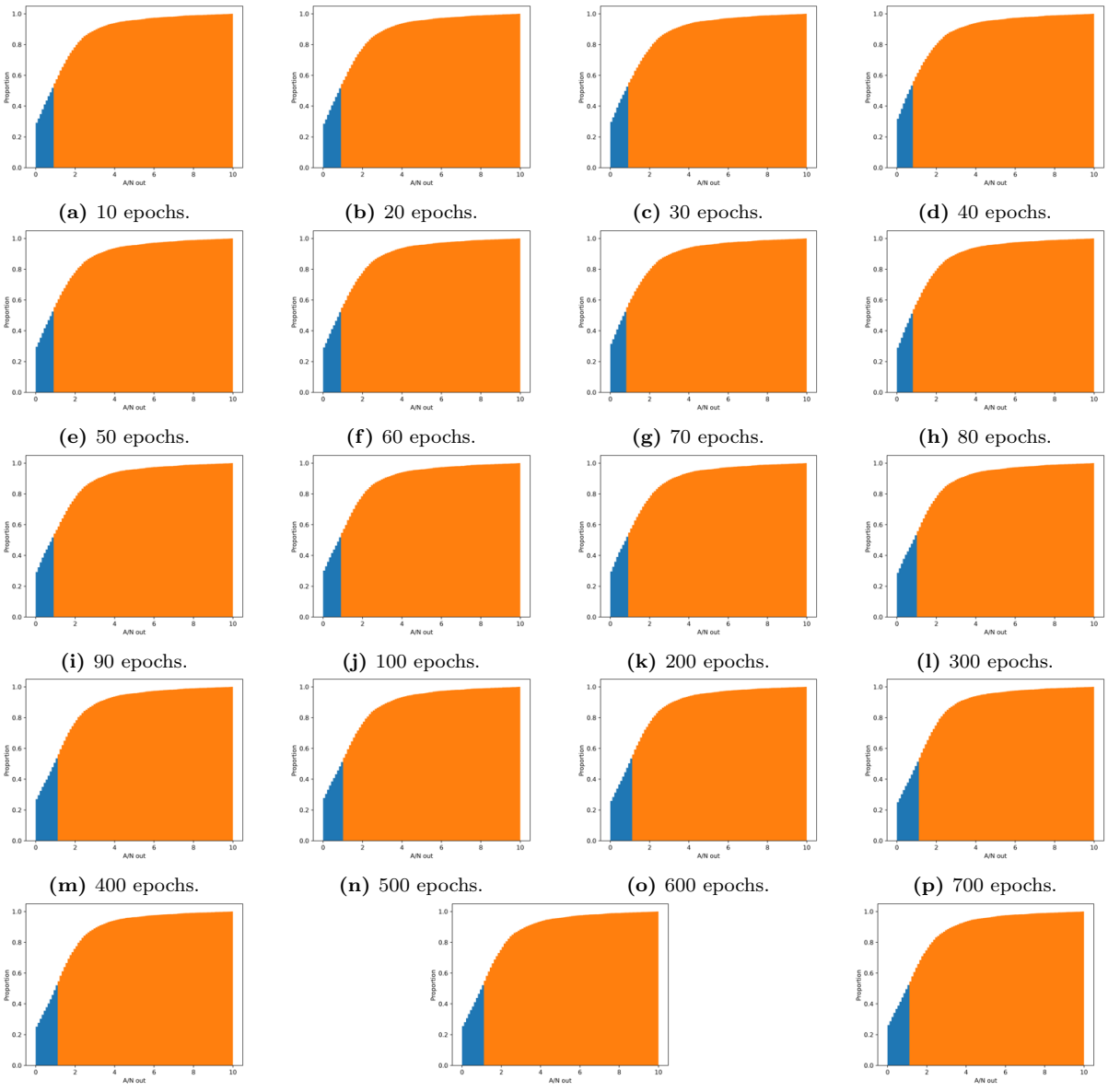
Network	FP (%)	FN (%)	C	Epochs
LINER_ANN_5_10	3.529	12.386	0.516	10
LINER_ANN_6_20	2.779	13.043	0.529	20
LINER_ANN_7_30	3.314	12.471	0.464	30

Network	FP (%)	FN (%)	C	Epochs
LINER_ANN_8_40	4.014	11.886	0.597	40
LINER_ANN_9_50	3.175	12.468	0.448	50
LINER_ANN_11_70	3.814	11.950	0.419	70
LINER_ANN_12_80	2.864	13.014	0.478	80
LINER_ANN_13_90	2.789	12.918	0.521	90
LINER_ANN_0_100	3.364	12.411	0.578	100
LINER_ANN_1_200	3.086	12.775	0.576	200
LINER_ANN_2_300	3.261	13.107	0.587	300
LINER_ANN_3_400	2.879	13.732	0.638	400
LINER_ANN_4_500	3.911	13.293	0.673	500
LINER_ANN_10_60	3.032	12.682	0.501	60
LINER_ANN_14_600	3.036	14.311	0.798	600
LINER_ANN_15_700	2.911	14.779	0.826	700
LINER_ANN_16_800	3.268	14.707	0.806	800
LINER_ANN_17_900	4.175	14.150	0.835	900
LINER_ANN_18_1000	4.143	13.929	0.866	1000
planetary_nebula_ANN_5_10	2.621	8.832	0.628	10
planetary_nebula_ANN_6_20	2.375	9.057	0.516	20
planetary_nebula_ANN_7_30	3.025	7.957	0.433	30
planetary_nebula_ANN_8_40	3.179	7.943	0.465	40
planetary_nebula_ANN_9_50	2.479	8.850	0.646	50
planetary_nebula_ANN_10_60	2.629	8.357	0.406	60
planetary_nebula_ANN_11_70	3.100	7.989	0.525	70
planetary_nebula_ANN_12_80	3.175	7.943	0.468	80
planetary_nebula_ANN_13_90	3.118	7.950	0.556	90
planetary_nebula_ANN_0_100	3.211	7.946	0.350	100
planetary_nebula_ANN_1_200	2.889	8.104	0.468	200
planetary_nebula_ANN_2_300	3.054	8.057	0.601	300
planetary_nebula_ANN_3_400	3.082	7.879	0.479	400
planetary_nebula_ANN_4_500	3.286	7.739	0.558	500
planetary_nebula_ANN_14_600	3.107	7.968	0.468	600
planetary_nebula_ANN_15_700	3.129	7.779	0.479	700
planetary_nebula_ANN_16_800	3.964	8.618	0.606	800
planetary_nebula_ANN_17_900	2.979	9.939	0.792	900
planetary_nebula_ANN_18_1000	3.129	7.861	0.512	1000
seyfert_ANN_5_10	2.729	10.525	0.764	10
seyfert_ANN_6_20	2.900	10.221	0.615	20
seyfert_ANN_7_30	2.746	10.796	0.603	30
seyfert_ANN_8_40	2.300	11.057	0.563	40
seyfert_ANN_9_50	2.321	10.679	0.526	50
seyfert_ANN_10_60	3.200	9.857	0.491	60
seyfert_ANN_11_70	2.671	10.357	0.562	70

Network	FP (%)	FN (%)	C	Epochs
seyfert_ANN_12_80	2.861	10.179	0.328	80
seyfert_ANN_13_90	2.704	10.411	0.477	90
seyfert_ANN_0_100	2.371	10.75	0.487	100
seyfert_ANN_1_200	2.954	10.246	0.500	200
seyfert_ANN_2_300	2.979	10.396	0.559	300
seyfert_ANN_3_400	3.918	9.782	0.451	400
seyfert_ANN_4_500	2.639	11.75	0.732	500
seyfert_ANN_14_600	2.943	12.071	0.818	600
seyfert_ANN_15_700	2.836	12.329	0.861	700
seyfert_ANN_16_800	3.000	12.018	0.819	800
seyfert_ANN_17_900	3.668	11.807	0.857	900
seyfert_ANN_18_1000	3.418	12.014	0.883	1000
star_forming_ANN_5_10	10.121	20.496	0.419	10
star_forming_ANN_6_20	8.236	22.211	0.554	20
star_forming_ANN_7_30	9.904	19.786	0.508	30
star_forming_ANN_8_40	8.761	20.604	0.498	40
star_forming_ANN_9_50	9.311	20.186	0.565	50
star_forming_ANN_10_60	10.054	19.15	0.452	60
star_forming_ANN_11_70	10.314	18.864	0.563	70
star_forming_ANN_12_80	9.407	19.939	0.497	80
star_forming_ANN_13_90	9.275	20.000	0.501	90
star_forming_ANN_0_100	8.068	21.532	0.560	100
star_forming_ANN_1_200	9.721	20.461	0.538	200
star_forming_ANN_2_300	10.121	20.429	0.476	300
star_forming_ANN_3_400	9.054	22.368	0.608	400
star_forming_ANN_4_500	9.311	22.804	0.586	500
star_forming_ANN_14_600	8.939	22.861	0.68	600
star_forming_ANN_15_700	10.046	23.196	0.605	700
star_forming_ANN_16_800	10.154	23.807	0.704	800
star_forming_ANN_17_900	10.829	22.543	0.630	900
star_forming_ANN_18_1000	9.539	23.782	0.719	1000

E Neural Network Cumulative Histograms

The cumulative histograms showing the distribution of the false negatives for each of the neural networks are shown in figures E.1, E.2, E.3 and E.4.



(a) 10 epochs. **(b)** 20 epochs. **(c)** 30 epochs. **(d)** 40 epochs.

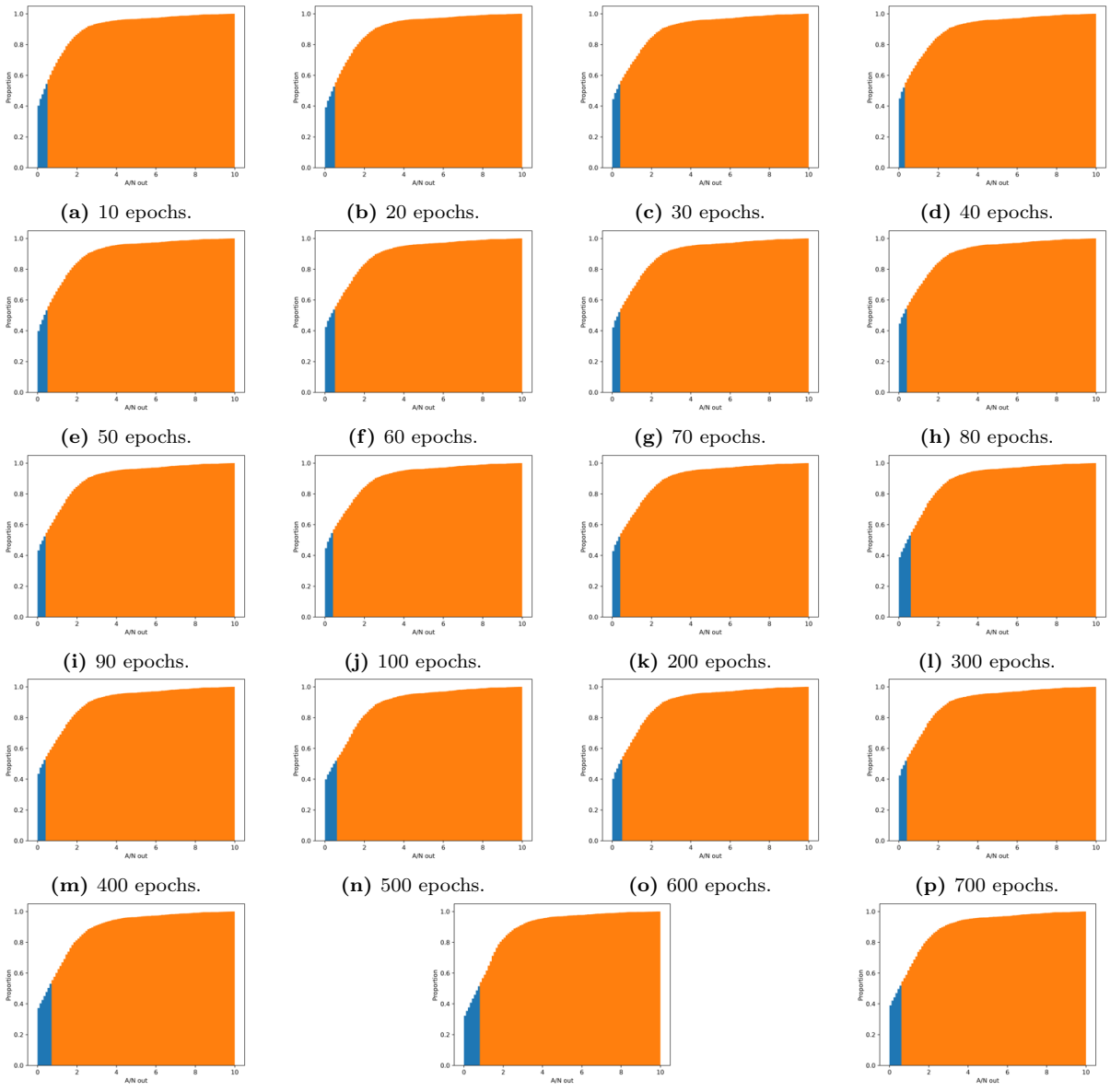
(e) 50 epochs. **(f)** 60 epochs. **(g)** 70 epochs. **(h)** 80 epochs.

(i) 90 epochs. **(j)** 100 epochs. **(k)** 200 epochs. **(l)** 300 epochs.

(m) 400 epochs. **(n)** 500 epochs. **(o)** 600 epochs. **(p)** 700 epochs.

(q) 800 epochs. **(r)** 900 epochs. **(s)** 1000 epochs.

Figure E.1: Histograms showing the cumulative distribution of false negatives in networks trained over a number of different epochs for use with LINER spectra. The bars in blue are the half with lower A/N and the bars in orange are the half with higher A/N. The y-axis shows the proportion of spectra from 0 to 1 and the x-axis shows the A/N of the Hα line.



(a) 10 epochs. **(b)** 20 epochs. **(c)** 30 epochs. **(d)** 40 epochs.

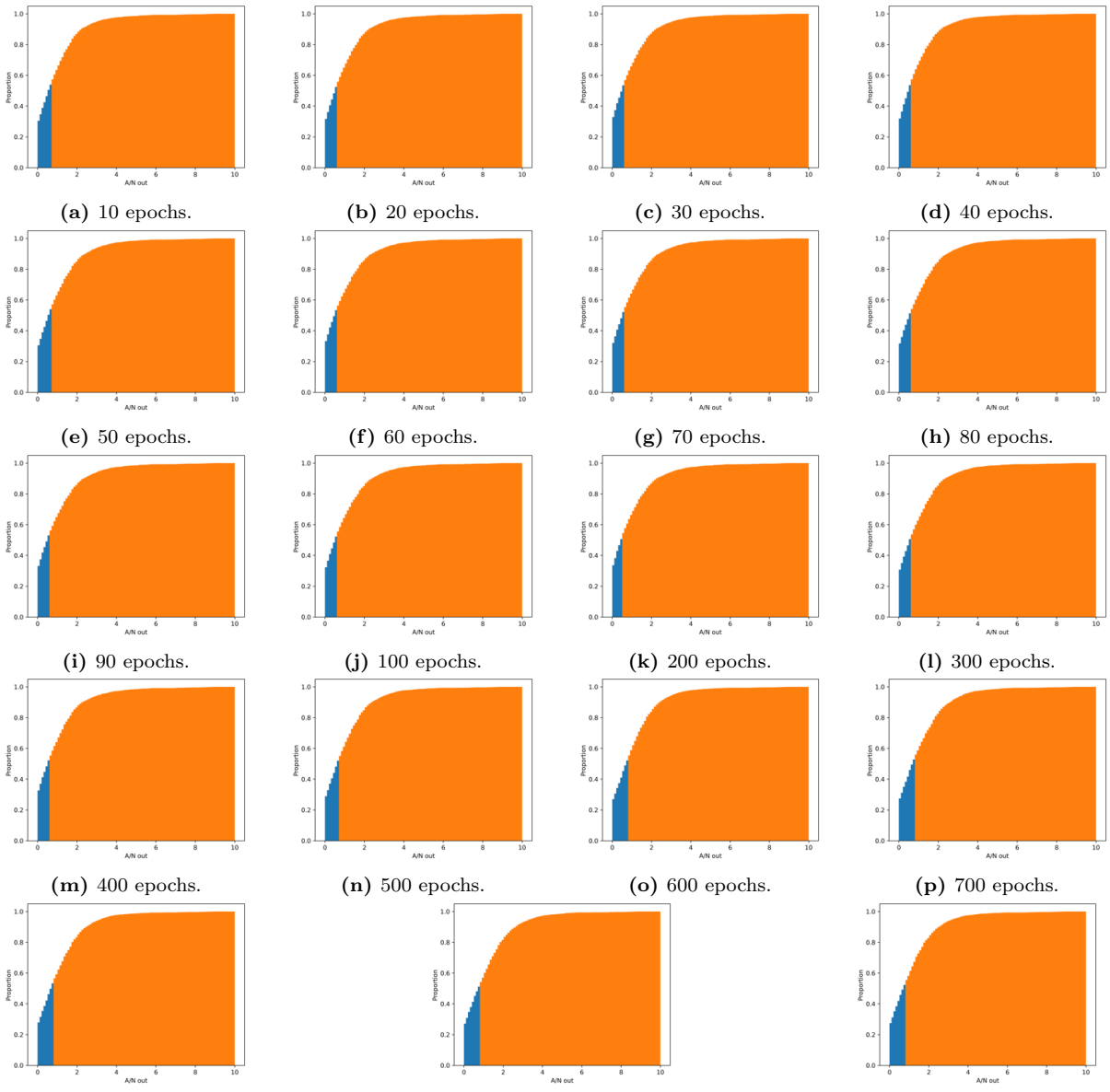
(e) 50 epochs. **(f)** 60 epochs. **(g)** 70 epochs. **(h)** 80 epochs.

(i) 90 epochs. **(j)** 100 epochs. **(k)** 200 epochs. **(l)** 300 epochs.

(m) 400 epochs. **(n)** 500 epochs. **(o)** 600 epochs. **(p)** 700 epochs.

(q) 800 epochs. **(r)** 900 epochs. **(s)** 1000 epochs.

Figure E.2: Histograms showing the cumulative distribution of false negatives in networks trained over a number of different epochs for use with planetary nebula spectra. The bars in blue are the half with lower A/N and the bars in orange are the half with higher A/N. The y-axis shows the proportion of spectra from 0 to 1 and the x-axis shows the A/N of the Hα line.



(a) 10 epochs. **(b)** 20 epochs. **(c)** 30 epochs. **(d)** 40 epochs.

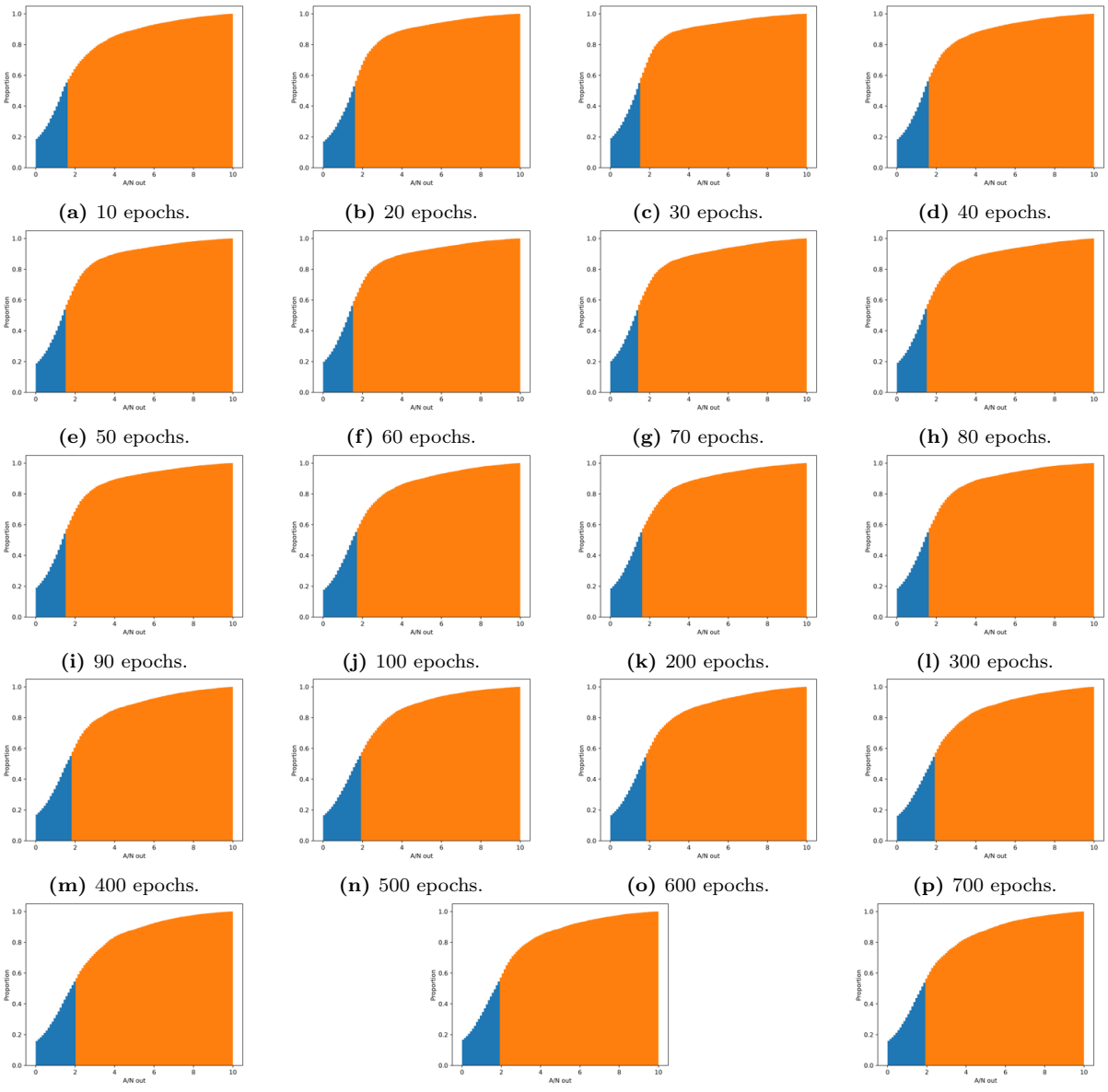
(e) 50 epochs. **(f)** 60 epochs. **(g)** 70 epochs. **(h)** 80 epochs.

(i) 90 epochs. **(j)** 100 epochs. **(k)** 200 epochs. **(l)** 300 epochs.

(m) 400 epochs. **(n)** 500 epochs. **(o)** 600 epochs. **(p)** 700 epochs.

(q) 800 epochs. **(r)** 900 epochs. **(s)** 1000 epochs.

Figure E.3: Histograms showing the cumulative distribution of false negatives in networks trained over a number of different epochs for use with Seyfert galaxy spectra. The bars in blue are the half with lower A/N and the bars in orange are the half with higher A/N. The y-axis shows the proportion of spectra from 0 to 1 and the x-axis shows the A/N of the Hα line.



(a) 10 epochs. **(b)** 20 epochs. **(c)** 30 epochs. **(d)** 40 epochs.

(e) 50 epochs. **(f)** 60 epochs. **(g)** 70 epochs. **(h)** 80 epochs.

(i) 90 epochs. **(j)** 100 epochs. **(k)** 200 epochs. **(l)** 300 epochs.

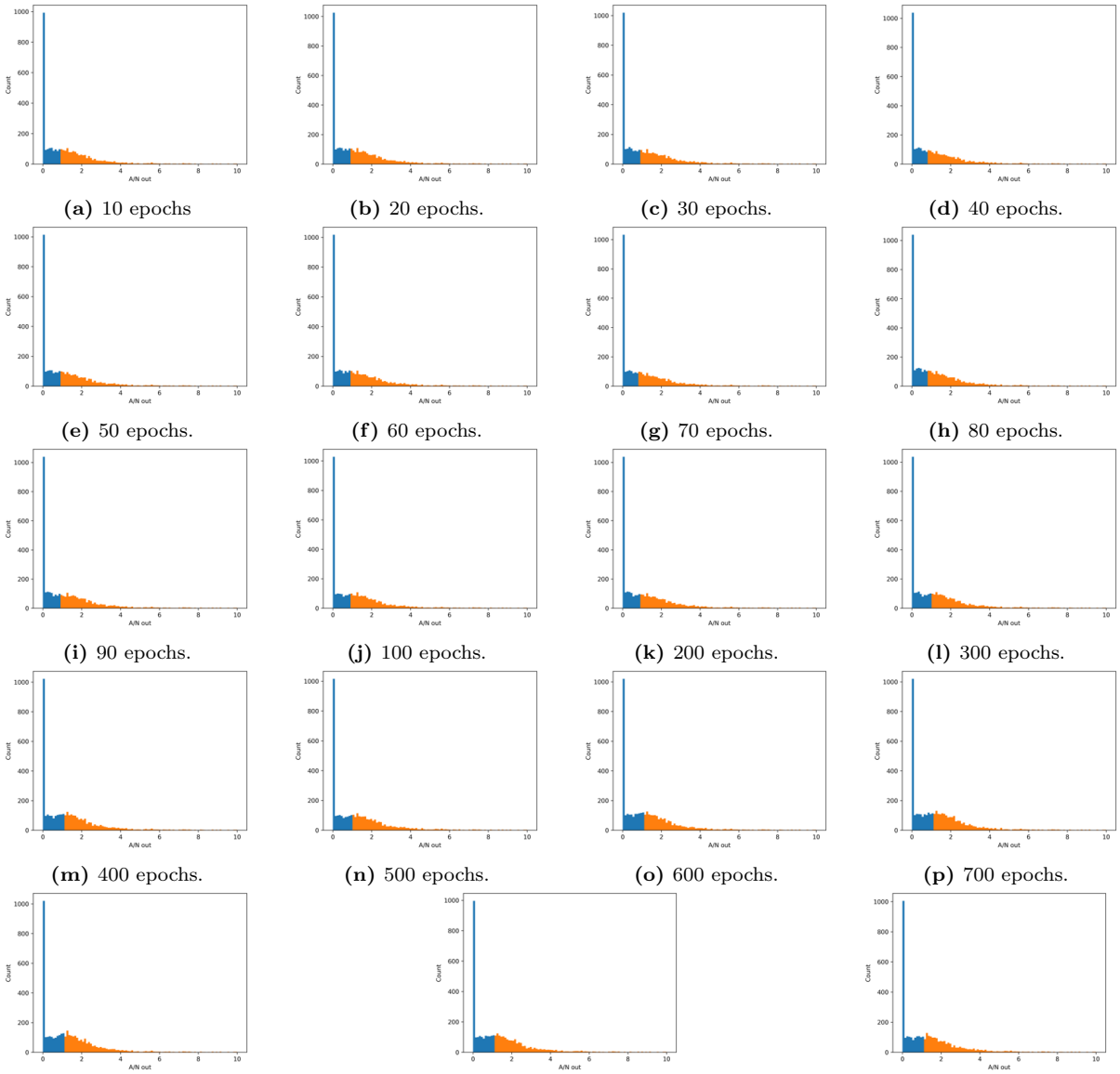
(m) 400 epochs. **(n)** 500 epochs. **(o)** 600 epochs. **(p)** 700 epochs.

(q) 800 epochs. **(r)** 900 epochs. **(s)** 1000 epochs.

Figure E.4: Histograms showing the cumulative distribution of false negatives in networks trained over a number of different epochs for use with star forming region spectra. The bars in blue are the half with lower A/N and the bars in orange are the half with higher A/N. The y-axis shows the proportion of spectra from 0 to 1 and the x-axis shows the A/N of the Hα line.

F Neural Network Histograms

The histograms showing the distribution of the false negatives for each of the neural networks are shown in figures F.1, F.2, F.3, F.4.



(a) 10 epochs **(b)** 20 epochs. **(c)** 30 epochs. **(d)** 40 epochs.

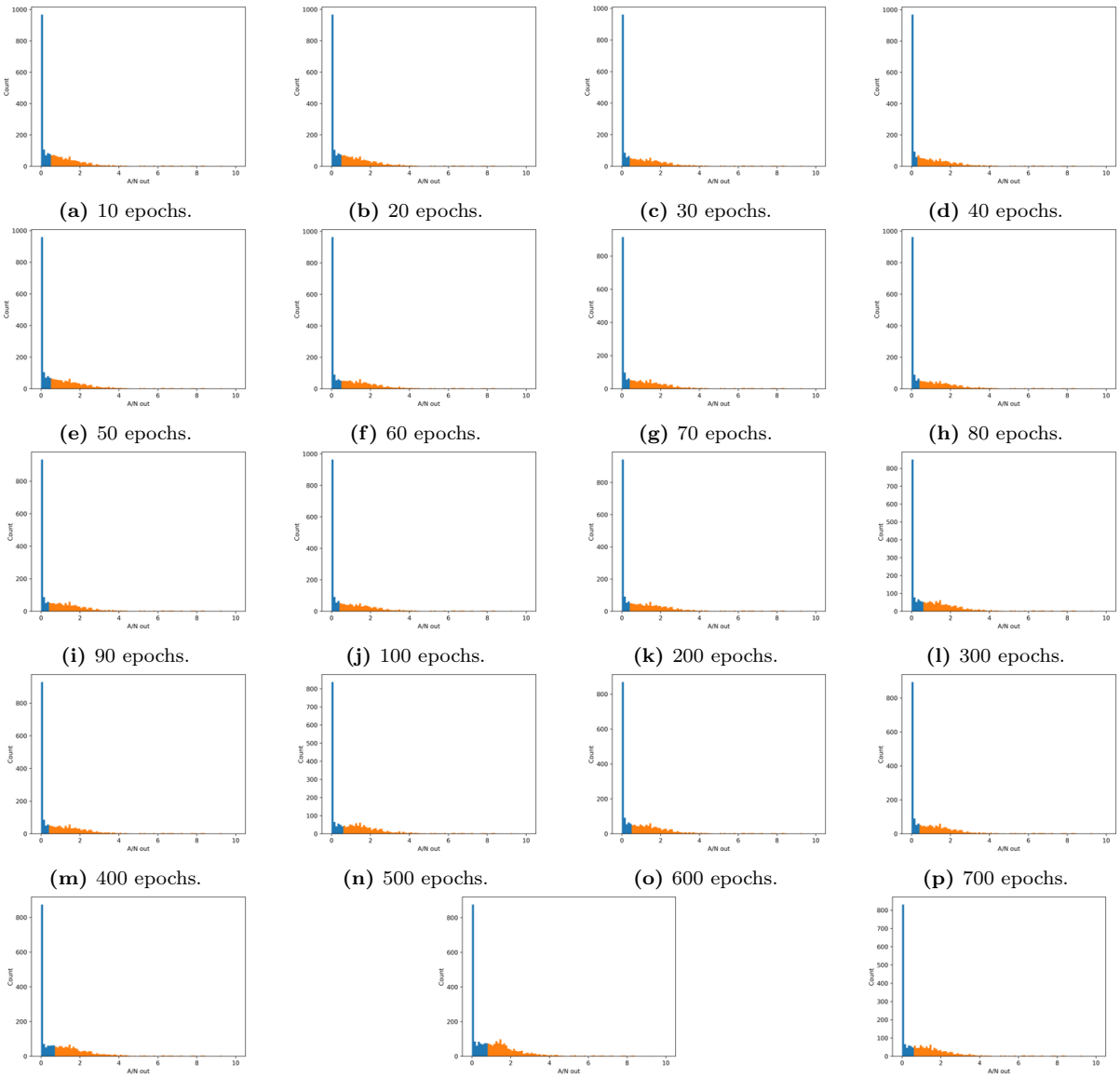
(e) 50 epochs. **(f)** 60 epochs. **(g)** 70 epochs. **(h)** 80 epochs.

(i) 90 epochs. **(j)** 100 epochs. **(k)** 200 epochs. **(l)** 300 epochs.

(m) 400 epochs. **(n)** 500 epochs. **(o)** 600 epochs. **(p)** 700 epochs.

(q) 800 epochs. **(r)** 900 epochs. **(s)** 1000 epochs.

Figure F.1: Histograms showing the distribution of false negatives in networks networks trained over a number of different epochs for use with LINER spectra. The bars in blue are the half with lower A/N and the bars in orange are the half with higher A/N. The y-axis shows the number of spectra and the x-axis shows the A/N of the Hα line.



(a) 10 epochs. **(b)** 20 epochs. **(c)** 30 epochs. **(d)** 40 epochs.

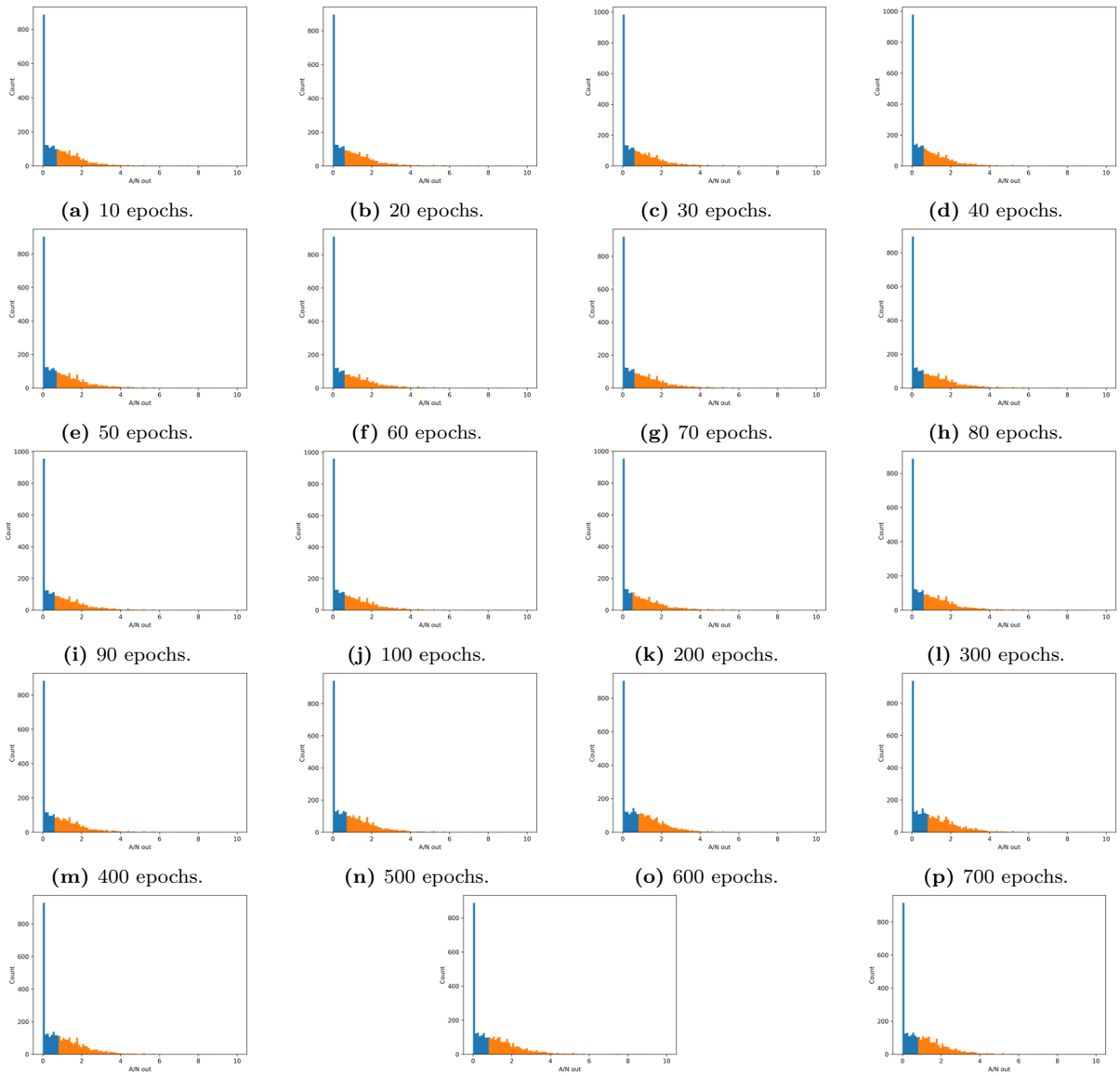
(e) 50 epochs. **(f)** 60 epochs. **(g)** 70 epochs. **(h)** 80 epochs.

(i) 90 epochs. **(j)** 100 epochs. **(k)** 200 epochs. **(l)** 300 epochs.

(m) 400 epochs. **(n)** 500 epochs. **(o)** 600 epochs. **(p)** 700 epochs.

(q) 800 epochs. **(r)** 900 epochs. **(s)** 1000 epochs.

Figure F.2: Histograms showing the distribution of false negatives in networks networks trained over a number of different epochs for use with planetary nebula spectra. The bars in blue are the half with lower A/N and the bars in orange are the half with higher A/N. The y-axis shows the number of spectra and the x-axis shows the A/N of the Hα line.



(a) 10 epochs. **(b)** 20 epochs. **(c)** 30 epochs. **(d)** 40 epochs.

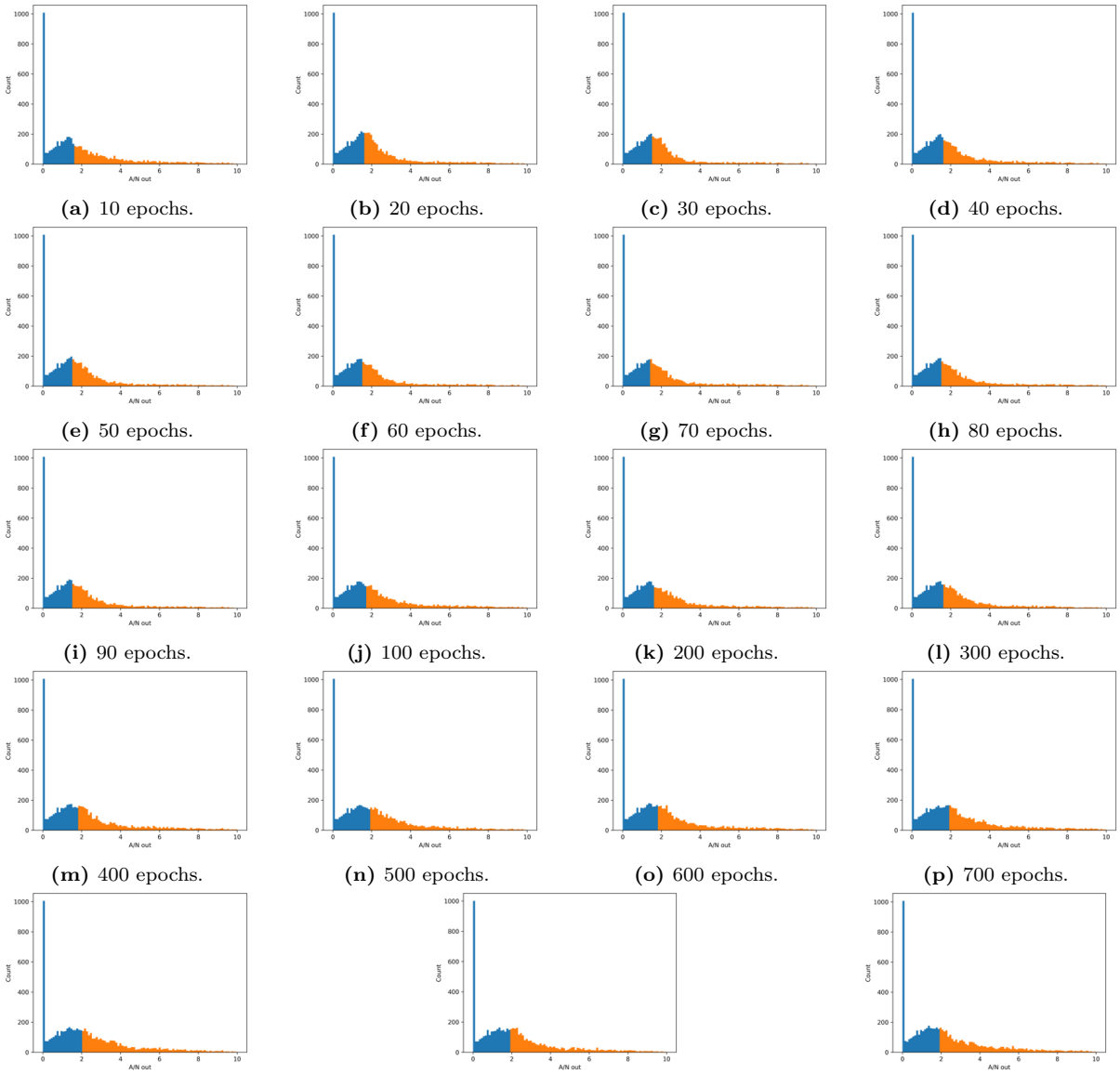
(e) 50 epochs. **(f)** 60 epochs. **(g)** 70 epochs. **(h)** 80 epochs.

(i) 90 epochs. **(j)** 100 epochs. **(k)** 200 epochs. **(l)** 300 epochs.

(m) 400 epochs. **(n)** 500 epochs. **(o)** 600 epochs. **(p)** 700 epochs.

(q) 800 epochs. **(r)** 900 epochs. **(s)** 1000 epochs.

Figure F.3: Histograms showing the distribution of false negatives in networks networks trained over a number of different epochs for use with Seyfert galaxy spectra. The bars in blue are the half with lower A/N and the bars in orange are the half with higher A/N. The y-axis shows the number of spectra and the x-axis shows the A/N of the Hα line.



(a) 10 epochs. **(b)** 20 epochs. **(c)** 30 epochs. **(d)** 40 epochs.

(e) 50 epochs. **(f)** 60 epochs. **(g)** 70 epochs. **(h)** 80 epochs.

(i) 90 epochs. **(j)** 100 epochs. **(k)** 200 epochs. **(l)** 300 epochs.

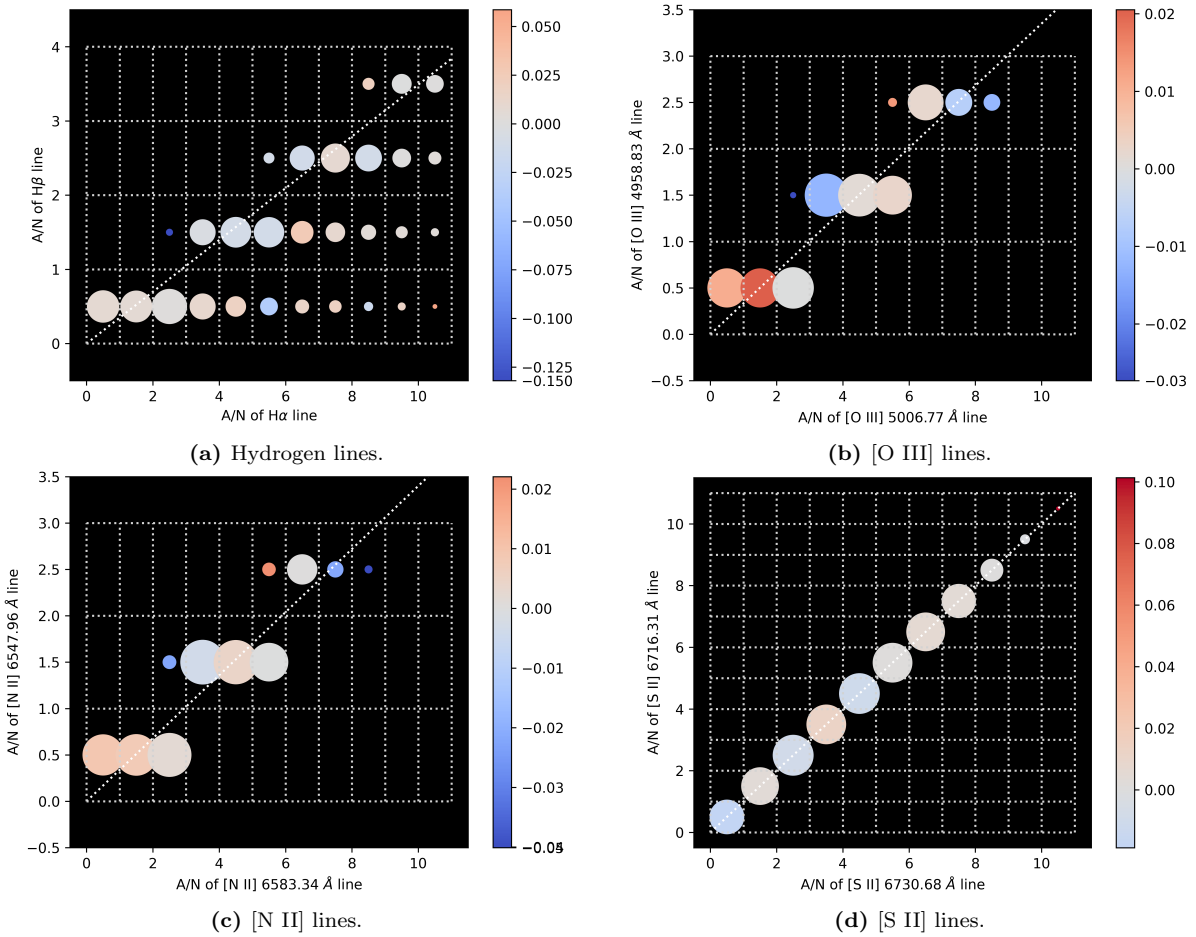
(m) 400 epochs. **(n)** 500 epochs. **(o)** 600 epochs. **(p)** 700 epochs.

(q) 800 epochs. **(r)** 900 epochs. **(s)** 1000 epochs.

Figure F.4: Histograms showing the distribution of false negatives in networks networks trained over a number of different epochs for use with star forming region spectra. The bars in blue are the half with lower A/N and the bars in orange are the half with higher A/N. The y-axis shows the number of spectra and the x-axis shows the A/N of the Hα line.

G Line Parameter Recovery Plots

The scatter plots representing the medians and standard deviations of the relative differences in the velocities are shown in figures G.1 and G.2 respectively, while the same for the standard deviations are shown in figures G.3 and G.4 respectively. The analogous plots for the simulations in which each line was fit separately are shown in figures G.5, G.6, G.7 and G.8



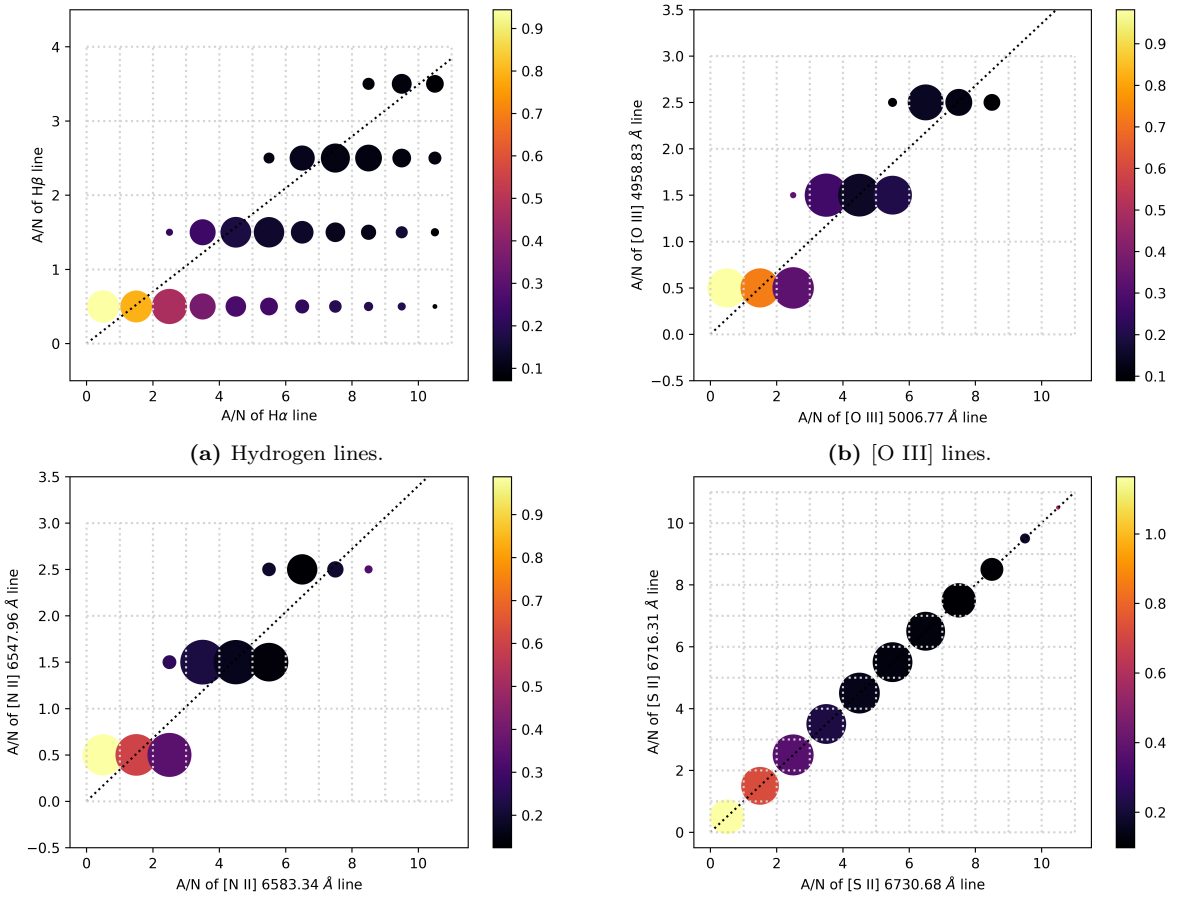
(a) Hydrogen lines.

(b) [O III] lines.

(c) [N II] lines.

(d) [S II] lines.

Figure G.1: The medians in the relative difference in the input and output velocities for lines of different species against the A/N of the lines. The size of a point represents the number of points in that that grid square. The dotted line through each figure shows the expected ratio of the lines.



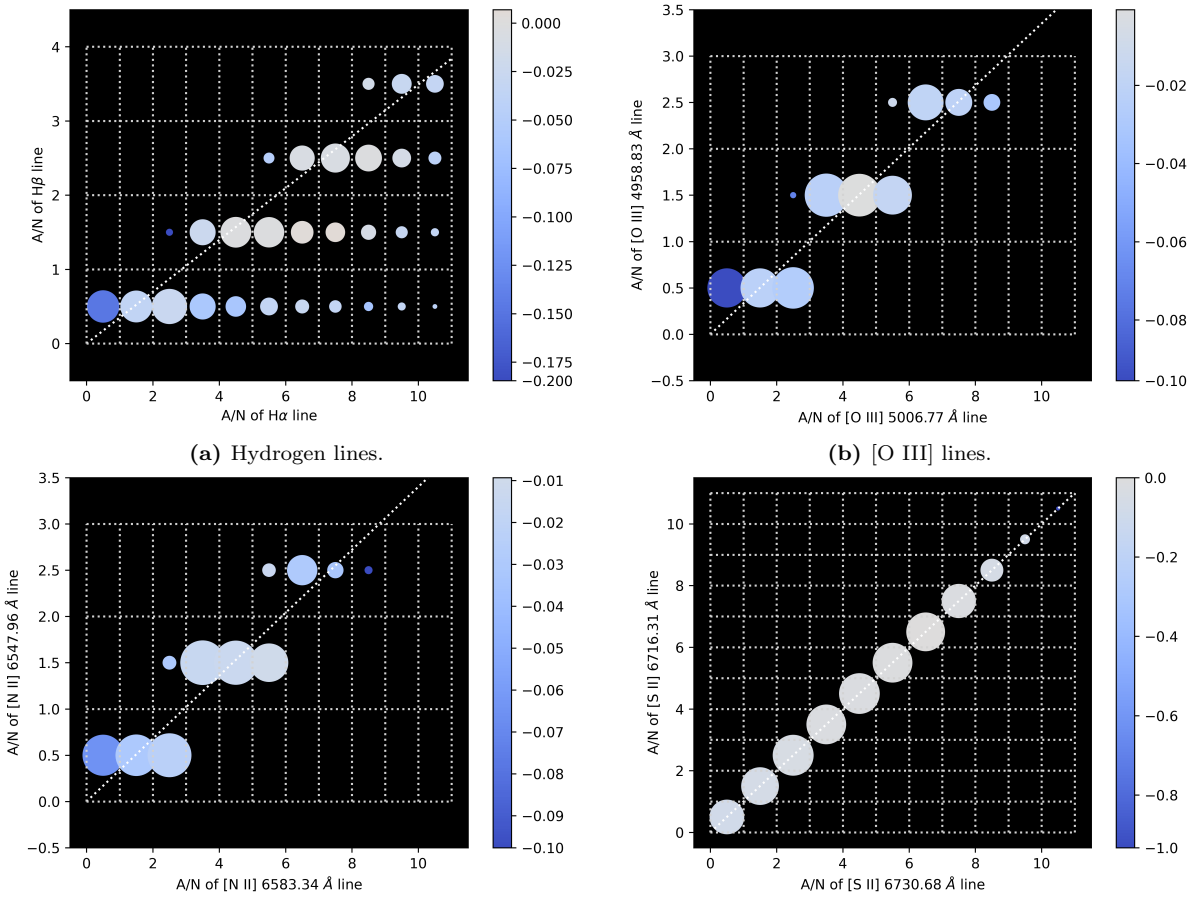
(a) Hydrogen lines.

(b) [O III] lines.

(c) [N II] lines.

(d) [S II] lines.

Figure G.2: The standard deviations in the relative difference in the input and output velocities for lines of different species against the A/N of the lines. The size of a point represents the number of points in that that grid square. The dotted line through each figure shows the expected ratio of the lines.



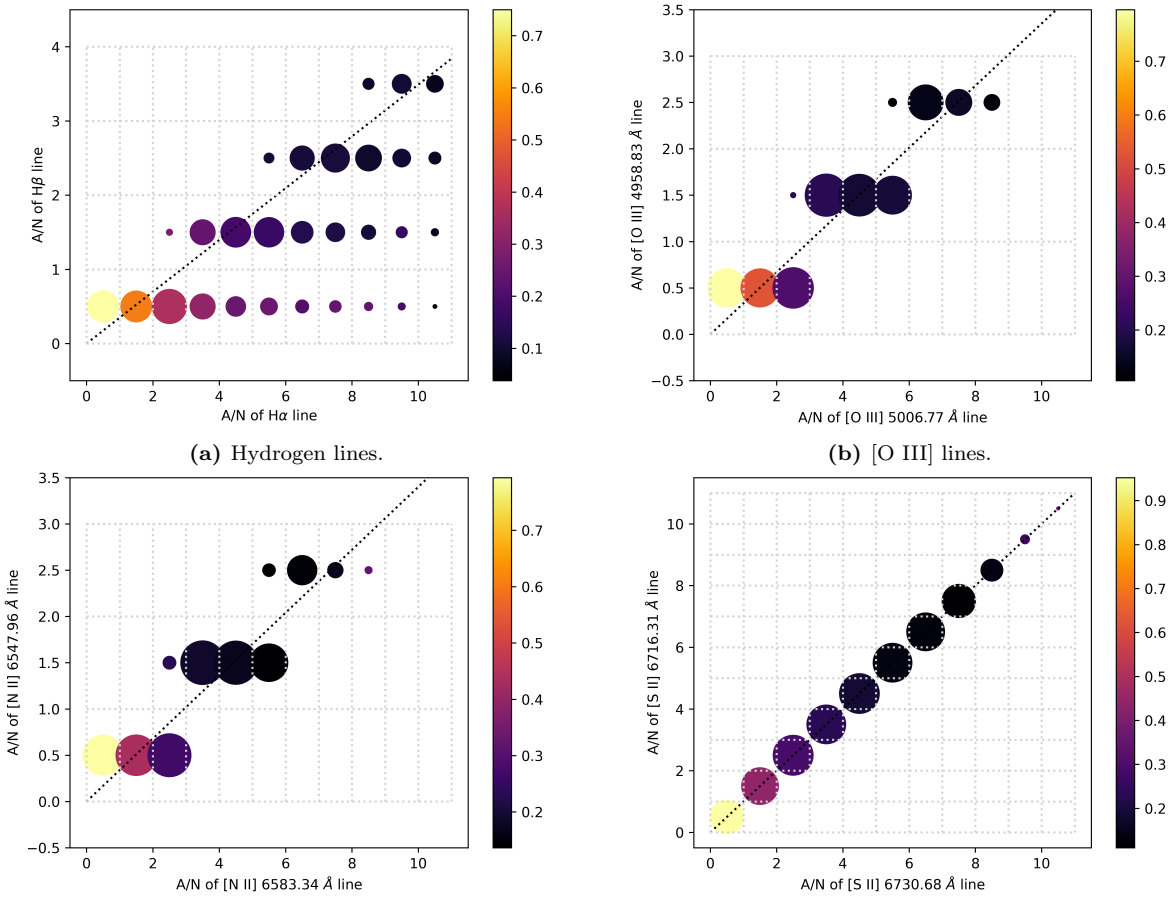
(a) Hydrogen lines.

(b) [O III] lines.

(c) [N II] lines.

(d) [S II] lines.

Figure G.3: The medians in the relative difference in the input and output standard deviations of the line profiles for lines of different species against the A/N of the lines. The size of a point represents the number of points in that that grid square. The dotted line through each figure shows the expected ratio of the lines.



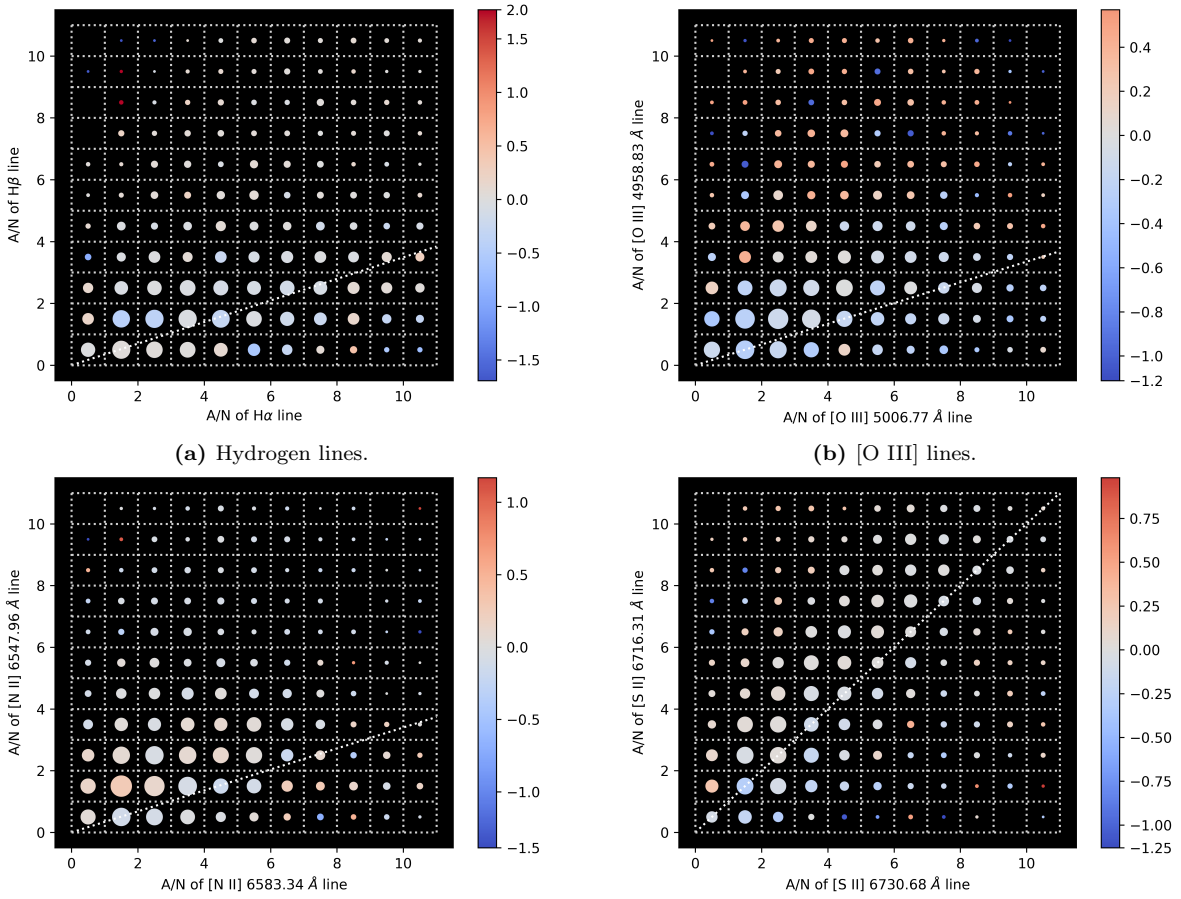
(a) Hydrogen lines.

(b) [O III] lines.

(c) [N II] lines.

(d) [S II] lines.

Figure G.4: The standard deviations in the relative difference in the input and output standard deviations of the line profiles for lines of different species against the A/N of the lines. The size of a point represents the number of points in that that grid square. The dotted line through each figure shows the expected ratio of the lines.



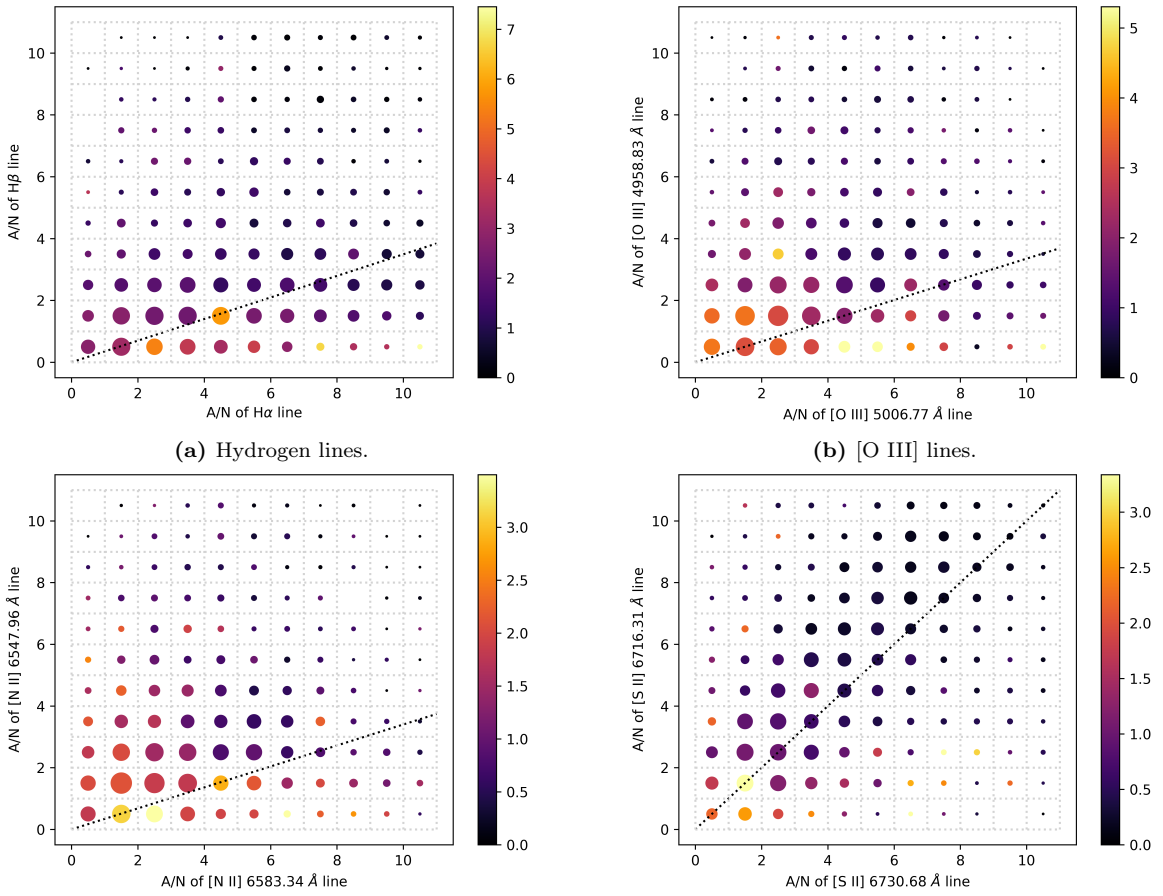
(a) Hydrogen lines.

(b) [O III] lines.

(c) [N II] lines.

(d) [S II] lines.

Figure G.5: The medians in the relative difference in the input and output velocities for lines of different species against the A/N of the lines for the lines which were fit independently. The size of a point represents the number of points in that that grid square. The dotted line through each figure shows the expected ratio of the lines.



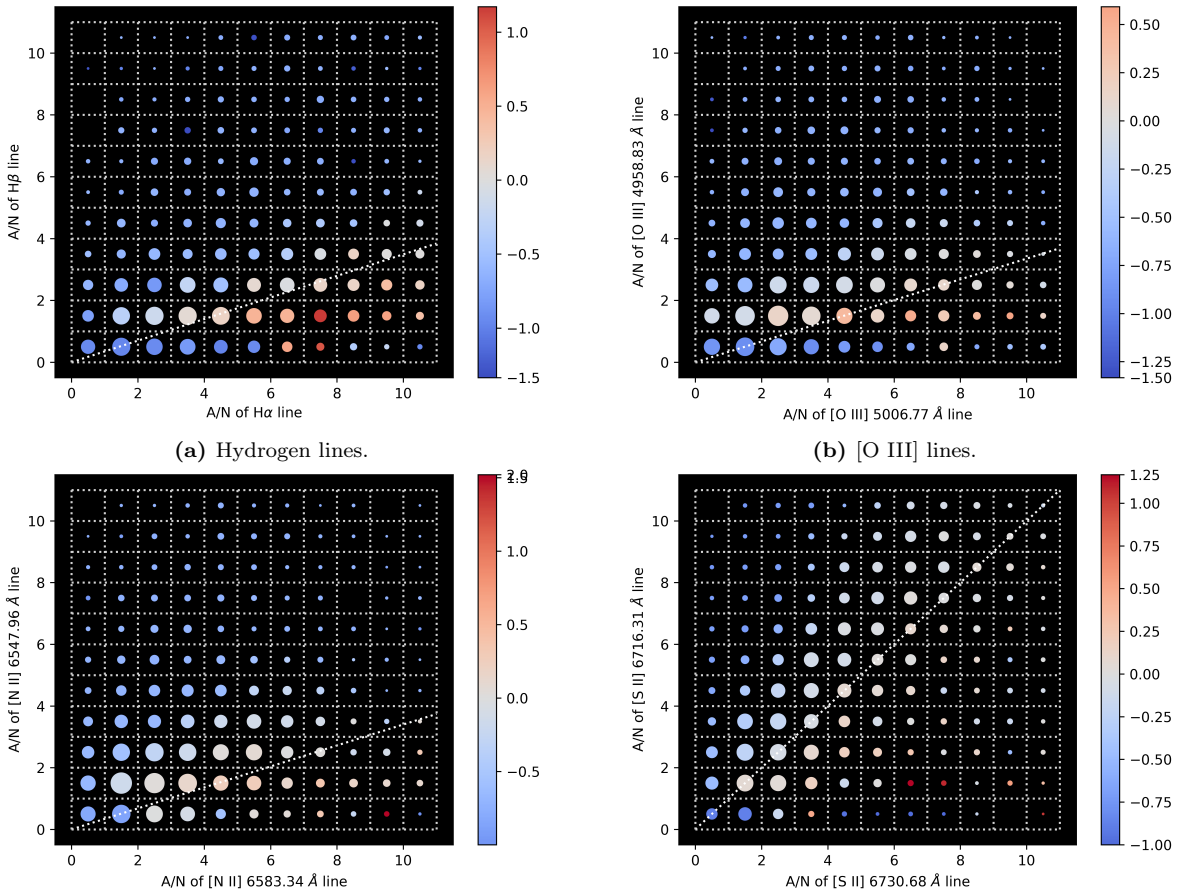
(a) Hydrogen lines.

(b) [O III] lines.

(c) [N II] lines.

(d) [S II] lines.

Figure G.6: The standard deviations in the relative difference in the input and output velocities for lines of different species against the A/N of the lines for the lines which were fit independently. The size of a point represents the number of points in that that grid square. The dotted line through each figure shows the expected ratio of the lines.



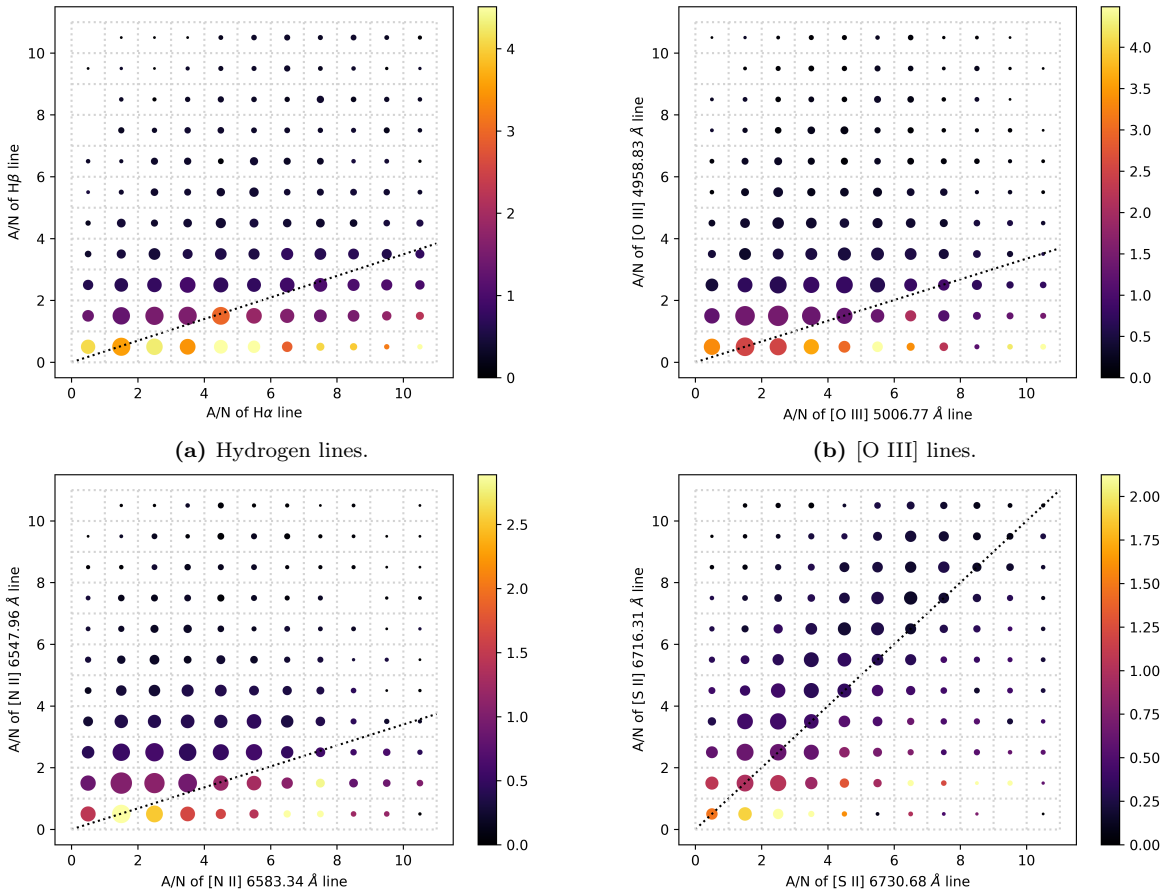
(a) Hydrogen lines.

(b) [O III] lines.

(c) [N II] lines.

(d) [S II] lines.

Figure G.7: The medians in the relative difference in the input and output standard deviations of the line profiles for lines of different species against the A/N of the lines for the lines which were fit independently. The size of a point represents the number of points in that that grid square. The dotted line through each figure shows the expected ratio of the lines.



(a) Hydrogen lines.

(b) [O III] lines.

(c) [N II] lines.

(d) [S II] lines.

Figure G.8: The standard deviations in the relative difference in the input and output standard deviations of the line profiles for lines of different species against the A/N of the lines for the lines which were fit independently. The size of a point represents the number of points in that that grid square. The dotted line through each figure shows the expected ratio of the lines.

H Code

The code I used in this project can be found here: github.com/ismisebrendan/capstone. The code is written in python and makes use of a number of libraries including NumPy, Matplotlib, lmfit[11] and TensorFlow[14] as well as number of the built-in python packages.