`

Literature Review of Deep Learning Applications in Astronomical Data Analysis

Engler, S.T., University of Hawaii, Department of Information and Computer Science, April 2020

1. Introduction

Astronomy is undergoing a paradigm shift as the data collected using ground and space telescopes are now much larger than before. Consequently, deep learning (DL) is beginning to have an impact in astronomy and astrophysics by supporting the analysis of these large datasets. This literature review focuses specifically on DL as applied to astronomy and astrophysics. DL has two main applications in astronomical data processing. The first is image processing through convolutional neural networks (CNNs), which has several astronomical applications, such as galaxy age categorization and stellar spectral classification. The second category is time series analysis, which is typically applied in exoplanet identification using space telescope (e.g. Kepler and TESS) data, or ground-based observations using the measurements of radial velocity extracted from spectral analysis.

This document follows the example of a systematic literature review conducted by (Hordri et al. 2017), where the same methodology was applied to evaluate research as outlined in *Section 3*. A manual document search was conducted in three databases: Google Scholar, University Hawaii Manoa's "OneSearch", and Cornell University's "arXiv". Only 65 journal articles were returned with "deep learning" as the first keyword, the logical operator 'AND', followed by either "astronomy" or "astrophysics" as the second. From this list I chose eleven primary studies to represent different research subsets of astronomy. The primary studies include ten journal articles and one peer reviewed conference paper.

2. Fundamental concepts of Deep Learning

This literature review covers many techniques in the computer science and Astronomical disciplines. Before proceeding, we review some terms and concepts of deep learning and astronomy.

`

2.1 Artificial neural network

An artificial neural network (ANN) consists of layers of objects called neurons that act like switches, and are governed by various equations such as the sigmoid function (see Sec 2.2). Each layer is connected to a higher or lower layer, *j*, by weights, *w_j.* as seen in Figure 1. Given an input *x_j*, the weighted value, *z,* can be calculated as the sum of the input value times the weight. Taking into account bias, b, the weighted value of z is seen in Eq.(1) .

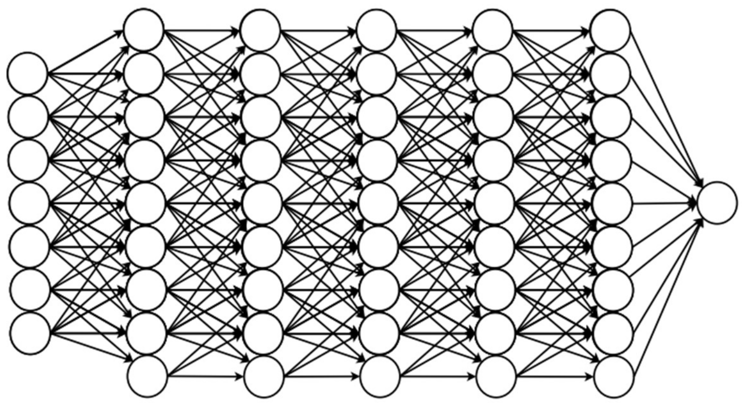

Figure 1:The Deep Artificial Neural Network (Sadowski 2016)

$$z = \sum_j w_j x_j + b \quad (1)$$

2.2 Activation Function

The sigmoid function seen in Eq. (2) is multiplied by the weight giving the node a differential value which turns the network into a non-linear solver.

$$\sigma(z) = \frac{1}{1 + e^{(-z)}} \qquad (2)$$

The rectified linear function, Eq.(3) is a simple non-linear function. Once values are passed in it will strip out the negative values, setting them to zero. It will leave the remaining positive values the same. What is important about this function is that the gradient is 1 for positive numbers, and zero for negative. This is an important property for training neural-nets as will be described in *Section 2*. The non-linear functions will create more complex relationships between nodes. Using non-linear activation functions increases the complexity of the structure with each layer of the network.

$$R(z) = \max(0, z) \qquad (3)$$

Each layer on the network receives weighted input, transforms it with mostly non-linear functions and then pass these values to the next layer. Layers dominantly use the same activation functions for all the nodes on the same layer. This continues until data going in the input layer makes its way to the output layer. Non-linear activation functions are important to deep learning because a chain of matrix multiplications can be represented by a single multiplication, allowing features for 100's of linear transformation be reproduced by a single layer.

2.3 Training Deep Architecture

The neural networks are trained on data that have solutions in order to provide feedback to the system. Normally the initial network weights are randomized as a starting point. The data can be anything from time-series data to images. For training, the dataset is broken up into different training and validation sets. The data will commonly have 20% of the dataset aside for validation, and another 20% for testing, and 60% for training. During the training cycle of forward/back propagation the weights are adjusted in the neural net such that the accuracy is improved not for single parameter but for all the parameters at the same time. To accomplish this the error is calculated within the parameter space. This creates an error surface on which one now must locate the lowest point on the surface to find the most accurate weights for all the parameters. Finding the minimum of this surface is most often accomplished through gradient descent.

2.4 Gradient Decent

Figure 3 is an example of an error surface. The surface represents the error within the solution space, and by taking the first derivative of the error function allows one to find the negative gradient that allows one to move in the direction of the least error on the surface. There are numerous methods to accomplish this, which will not be covered in this literature review. There are also numerous complications associated with finding the minimum in such a solution space, such as having the result getting trapped in a localized minimum instead of a global one. If the solution is in the local minimum it may not yield the best result. Local minimum can be in the shape of saddle points with sometimes dramatic descents/ascents and which can trap some algorithms. Often cross entropy error measure is used to evaluate the error in neural networks using logistic regression as it allows one to find a global minimum rather than a localized one. (Maclaurin, Duvenaud, and Adams 2015) Cross entropy, seen in Eq.(4), allows one to tune the weights through parameters and can be used to define a loss function in machine learning and optimization. The true probability p_n is the true label, and the given distribution q_n is given by the logistic function where the predicted value of the current model is when we set $q_{y=0} = 1 - \hat{y}$ and $q_{y=1} = \hat{y} = 1/(1 + e^{-w\,x_n})$ to give the average loss function. (Murphy 2012)

$$L(\boldsymbol{w}) = \frac{1}{N}\sum_{n=1}^{N} H(p_n, q_n) = \sum_i p_i \log q_i - \frac{1}{N}\sum_{n=1}^{N}[y_n \log \hat{y} + (1-y)\log(1-\hat{y})] \quad (4)$$

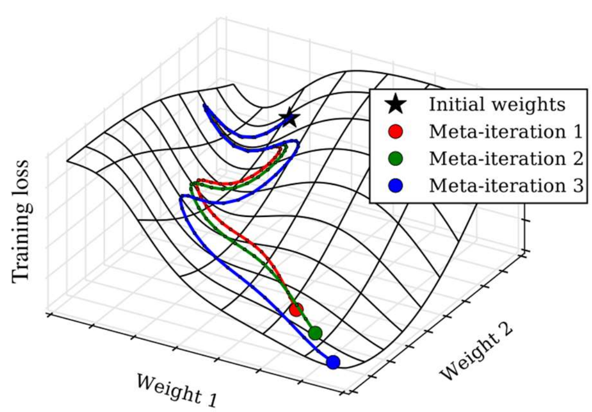

Figure 2: The error surface of a neural network. The solution is found at the lowest point on the surface The different paths represent different methods for finding the global minimum. The meta-

iteration is run over each stochastic gradient decent to find the lowest error for parameters 1 and 2. The training loss is given by the derivative of Eq (4). (Maclaurin, Duvenaud, and Adams 2015)

2.5 Backpropagation

Errors are calculated from the training data normally using the mean squared error (MSE), which is then passed backwards from the output to the input in the network. The ANN will then calculate again to minimize the error in the result by changing the weights of the neurons. This process cycles until the error is minimized. This is accomplished by using the chain rule on the MSE, the gradient of the error is calculated and allows the modification of the weights to provide a solution. *Fig. 3* shows the backpropagation algorithm across two layers of neurons. (Xu 2020)

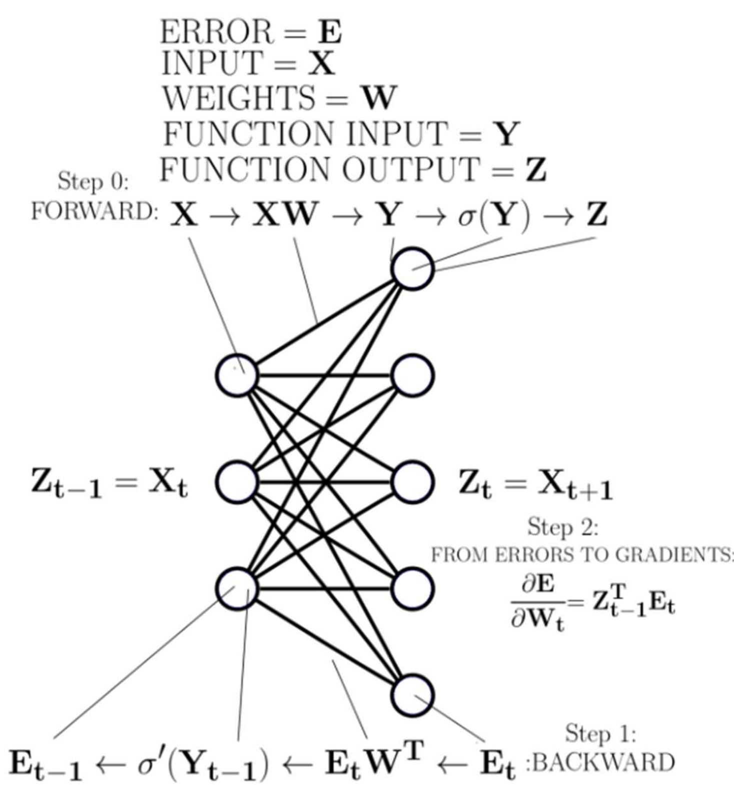

Figure 3: Backpropagation algorithm (Xu 2020)

Step 0: Forward propagation. Calculate the hidden nodes, then calculate the output node.

Step 1: Backward propagation of error to previous timestep

Step 2: Calculate the gradients from the errors with respect to the weights and update them.

\`

2.6 Convolutional Neural Network

Fukushima was the first person to design a ANN that was able to recognize hand-written numbers (Fukushima 1988). A CNN in its simplest form consists of a feature extractor and classification (Alom et al. 2018). A CNN takes advantage of the image inputs and allows the architecture to constrain it in an efficient way. When input is passed to the convolution layer, the output of that layer is an activation map that extracts features from the input data to pass further. The convolution leverages spatial information so that, on each layer, more and more spatial features are recognized. The convolution layer consists of feature maps going through kernels that are passed into activation functions (such as the sigmoid, SoftMax, or rectified linear), which create feature maps. These feature maps can then be combined with multiple input feature maps (Hála 2014) Convolutional networks consist of three different attributes: local repetitive fields, shared weights, and pooling (max pooling).

Local repetitive fields preserve spatial information by storing image pixels in a matrix. Then a single hidden neuron can be connected to a submatrix within the image matrix. Shared weights and bias within the hidden layer are used to learn position independent sub-features of the image.

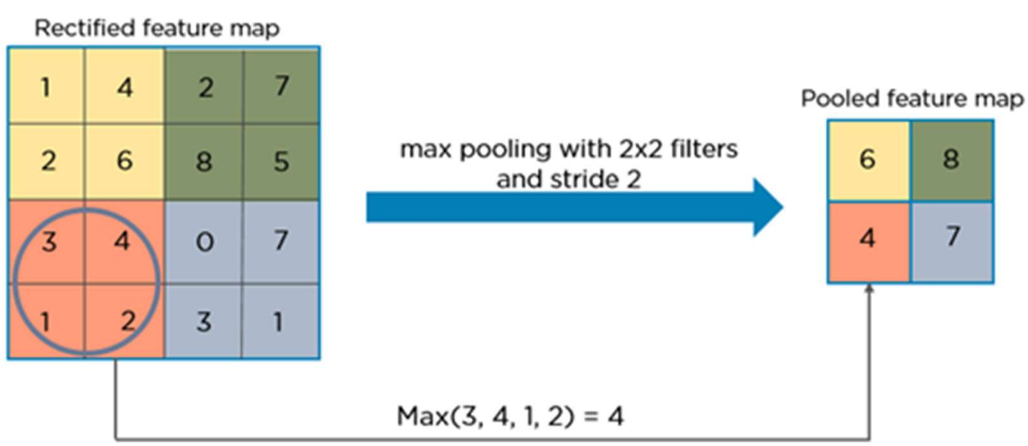

Figure 4: Example of max pooling where a 4x4 grid is broken into four pieces and subsequently the max value is taken from each of the four pieces to form a 2x2 grid. (Xu 2020)

Pooling layers are summaries of the output of a feature map. Commonly, the maximum value in a submatrix is combined with other max values in other sub-matrices to produce a new submatrix, as shown in Figure 4 This can also be done by taking the average of the submatrix to produce an averaging pooling layer. The pooling layer allows the network to process shape variations. This layer prevents the NN from having too many parameters to control (Alom et al. 2018), which allows the CNN to be less sensitive to the gradient problem (discussed in Sec. 2, below) and produces a highly accurate and efficient NN. [ibid] Pooling layers further reduce the number of parameters. [ibid] The max-pooling layer is a form of subsampling, and it down-samples the output from the convolution layer. Each output layer is down-sampled into half the size of its incoming dimension. The down sample is obtained by averaging the neighboring values or taking the maximum value.

Shared weights and bias score the features extracted from the convolution layer. This information is backpropagated through the CNN with the convolutional operation on the filters and their previous layer. (Baron et al. 2015) The classification layer is often a feed-forward NN. (Nair and Hinton 2010) From the output, a mean square loss is then calculated and backpropagated through the system. (Alom et al. 2018) The number of parameters of the network are significantly reduced through the CNN topology because of the sharing of parameters. (Yang et al. 2016)

A typical CNN for object detection in images is shown in *Figure 6*, and it is an important tool in several astronomical applications. The CNN consists of four layers. The image itself is the network input and is broken up into convolutional layers. (Gupta, Ankita, Gurunath Gurrala 2018) The convolutional layer is isolated from the previous layer to attach more weight to the localized structure. (Yann and Yoshua 1995) The first layer extracts basic features, which are then combined by higher layers to create new features. Ultimately, a feature map is created, which acts as a filter to extract similar features within the images. Once the convolutional layer extracts the feature map, it enters a pooling layer. The pooling layer extracts important features from the convolutional layer. This second set of feature maps is then passed through another convolutional layer to place the features within a fully connected layer, which consists of the entire image. In this manner, the feature maps are matched with identical items in the full image. (Krizhevsky, Sutskever, and Hinton 2017)

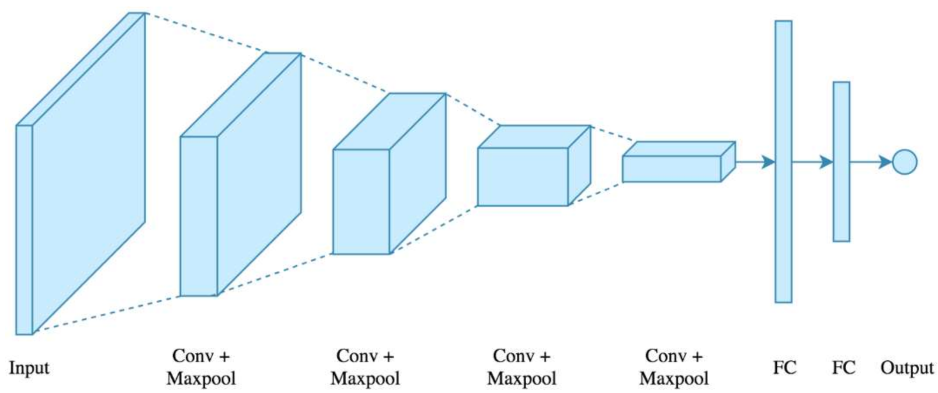

Figure 6: Convolution and max pooling data (Krizhevsky, Sutskever, and Hinton 2017)

2.7 Recurrent neural networks

A recurrent neural network (RNN) is an NN where the output from a previous step is fed as input to the current step. The main and most important feature of an RNN is the hidden state, which remembers some information about a sequence. The network can be a long string of time series data with outputs of equal length. (Hochreiter and Schmidhuber 1997) RNNs by themselves are limited to the dependencies between each cell, which is called the long-term dependency problem. To overcome this, a Long-Short Term Memory (LSTM) cell is introduced to retain information indefinitely. (Vinyals et al. 2015) This is accomplished by using three different gates, which will let the cell read or write information according to parameters that open or close these gates. They are cells with logistic and linear units, where information gets into the cell when the "write" gate is open. The information stays in memory until the "keep" gate is opened. The "read" gate gives access. It is also possible to back-propagate through the memory cells. Each gate is a layer of neural nets governed by a sigmoid function. (Gerstner 2001)

`

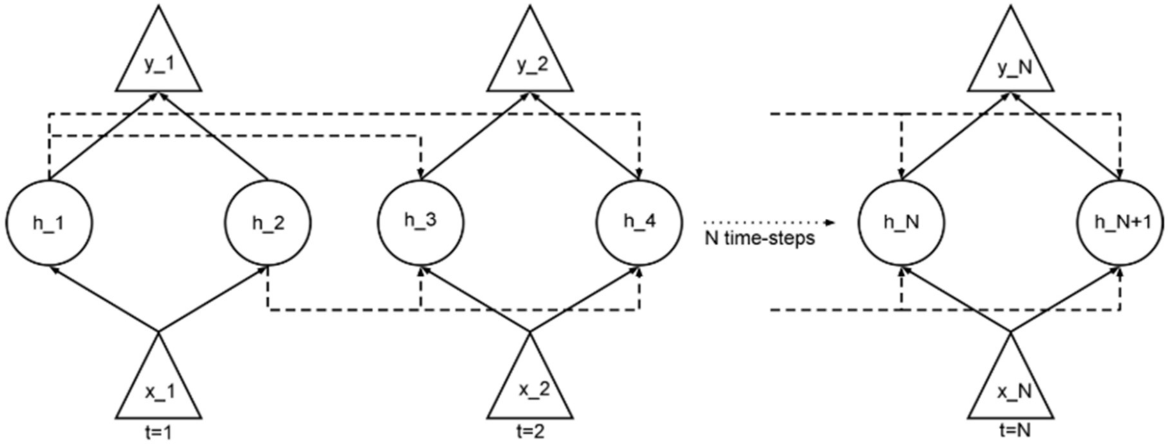

Figure 3: Recurrent Neural Network. (Engler 2017)

The LSTM-RNN is an architecture designed as an improvement on the RNN with the introduction of simple memory cells with a gating architecture. For each memory cell, the network computes the output of four gates: an update, input, forget, and output gates, as seen in *Figure 4.* Long-Term Short Memory is suitable for time series applications, as well as for images and signal processing. The training is through back-propagation and there are three different types of memory gates within the LSTM cell:

- **Input gate**: Uses a sigmoid function or tanh function to give weights to the values passing through. If they hit a threshold the value will pass into the memory cell.
- **Forget gate**: Allows the memory cell to discard values by giving them weight with a sigmoid function and comparing against a threshold.
- **Output gate**: Using the input gate value, it is compared against a sigmoid and tanh function by multiplying the tanh against it.

3. Types of Astronomical Observations and Data

Astronomy has a large variety of observational tools at its disposal. Telescopes collect light in the range from large dishes collecting radio waves, to the optical telescopes collecting in the visible spectrum, to satellites collecting gamma rays. Each of these collection methods come with their own type of data and methods of data reduction. This literature review covers a number of these methods which will be described in general here.

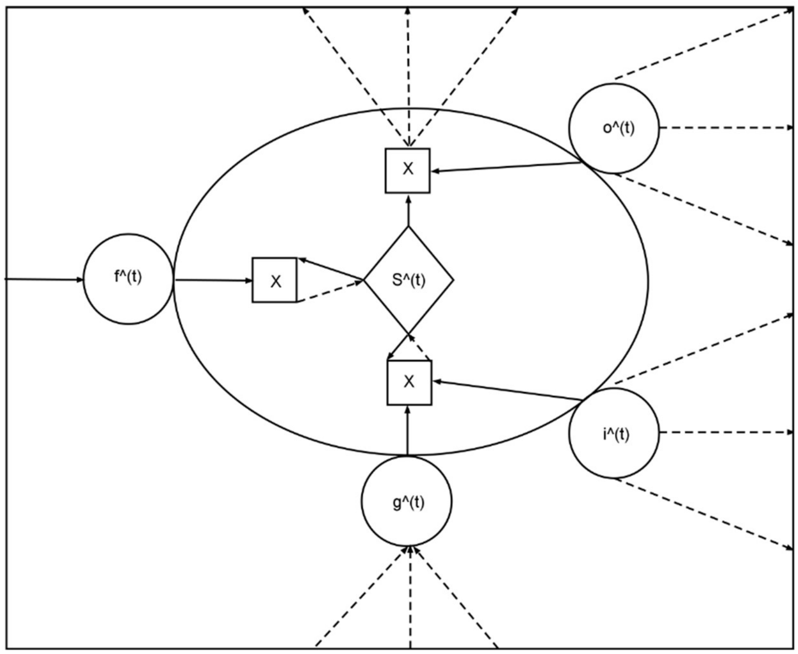

Figure 4: Diagram of an LSTM Cell. Forget gates as f^(t) and output gates as o^(t) Here, S^(t) is activated by x^(t) values from the hidden layer h^(t−1). The input gate i^(t) is multiplied by the input node g^(t) to give X. (Engler 2017)

3.1 Photometry

Photometry is one of the most popular form of data analysis from data collected by almost all observational methods. Dominantly used in the visible spectrum, light from the telescope will fall onto a photon collector such a charged couple device (CCD) camera. (Amelio, Tompsett, and Smith 1970) Well studied standard stars are used to compare to the unknown starts to obtain the stars magnitude. (Landolt 1992) The magnitude is a measure of the brightness of a star. In comparison to the standard star and adjusting for distance, the absolute magnitude of the star is obtained. With the absolute magnitude one can observe the accurate brightness of the star. Watching this brightness over time can reveal a lot of

`

information about the star over time. (Kuntzer, Tewes, and Courbin 2016)

3.1.1 Photometric planet detection

One of the more prominent uses of photometry is for planet detection. Satellite telescopes such as Kepler and TESS are designed to look at a section of the sky for extended periods of time. The light from the stars in the field are collected onto numerous highly sensitive CCD cameras which can measure the absolute magnitude of the stars to high precision. (Hinners, Tat, and Thorp 2018) When watching the star over time, if an extrasolar planet passes in front of it, it will cause the light from the star to dip and then brighten. By examining the light curve generated from the eclipsing planet, one can determine the distance, mass, and orbital period of the extra-solar planet. (Gomez Gonzalez, Absil, and Van Droogenbroeck 2018)

DL comes into play in planetary detection because these observations become exceptionally complex if you have more than once exo-planet which generates some very complicated light curves. A combination of simulations and DL with some supervised learning can discover these exoplanets within the data automatically. (Pearson, Palafox, and Griffith 2018)

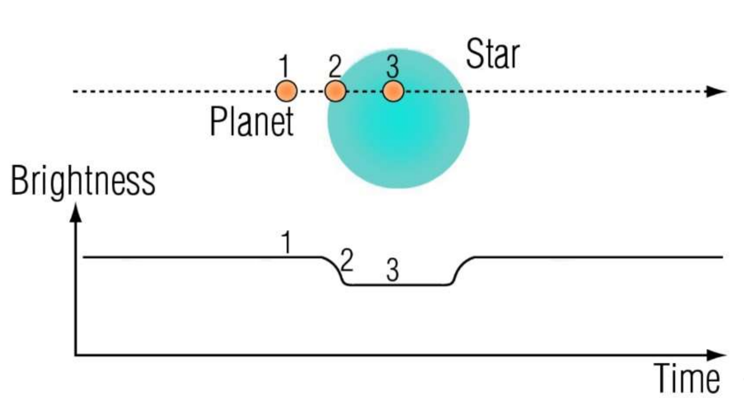

Figure 5: The brightness light curve of a planet passing a star. (NASA, ESA, G. Bacon (STSci), 2018)

`

3.1.2 Variable star photometry

Stars have a long and complex lifespan. Depending on its mass, a star will go through many stages from a spectral classification of G2, a yellow star like our sun, to a red giant, to its death in a nova. The stages of a blue giant, B2, its death will be in a super nova, and remaining years as a neutron star or black hole depending on its mass. One of these stages is a variable star phase. During this time a star will continuously and periodically change its radius. Other stars will go through random fluctuations in their sizes, such as wolf-rayet stars. (Crowther 2007) These physical radial changes translate into changes in brightness seen from Earth, and can be measured with photometry. Deep Learning can take wide field images and extract the photometric values of these stars over time, and then classify them.

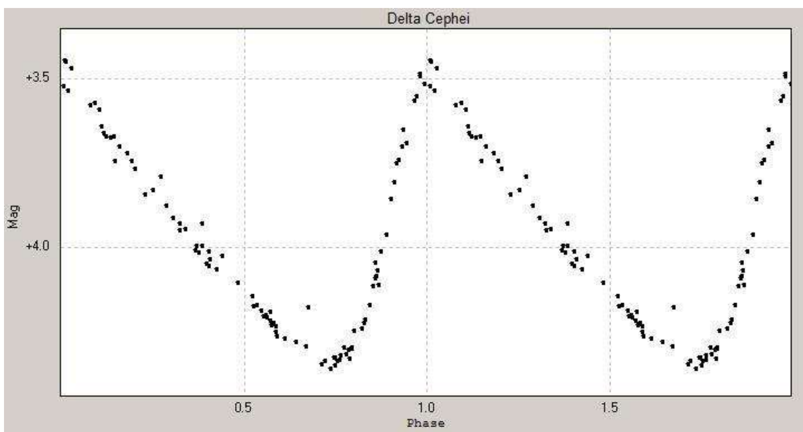

Figure 6: Periodic variation in brightness in star Delta Cephei. Three are numerous reasons stars dim and brighten. Classification methods can discover what is happening to each start. ThomasK Vbg / CC BY-SA (https://creativecommons.org/licenses/by-sa/3.

3.2 Spectroscopy

When Isaac Newton first split light in a prism he probably could not have guessed the vast amount of information this technique would give not only about stars, but about the galaxies and the grand scale of the universe. The technique would evolve over the years to include any interaction with radiative energy as a function of its wavelength or frequency, predominantly in the electromagnetic spectrum. It has allowed us to investigate the physical nature (elements) not only at the atomic scale, but over astronomical distances. (Skoog, Holler, and Crouch 2007)

3.2.1 *Stellar Spectral Type*

The light spectrum of a star in physics is analogous to DNA in biology, as it reveals a tremendous amount of the chemical composition and internal behavior of the star. This spectral type will change over its life cycle. In the spectrum one can extract the elements that are present in a star through the spectral absorption and emission lines due to the changing of electron positions relative to their atomic nucleus host. Deep learning can be used to take complex stellar spectra and classify them into stellar spectral type. The spectral type of a star is a category with a letter identifier (O, B, A, F, G, K, or M) based on the color of the star, with numerical subcategories ranging from 0 – 9. So, for example our Sun is currently a star of spectral type G2. (Russell 1914) DL can also provide other information, such as the velocity of turbulence within a star, and how fast the star is moving relative to the Earth.

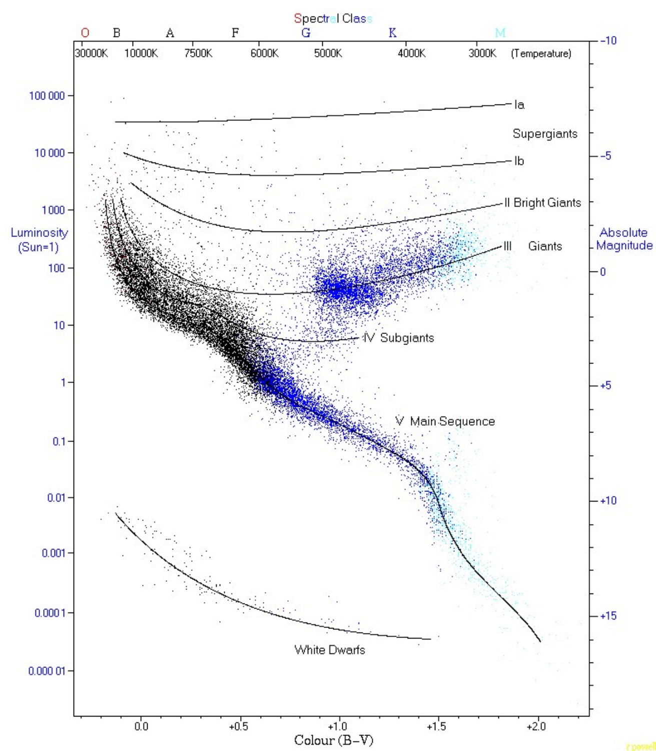

Figure 7: By Richard Powell - The Hertzsprung Russell Diagram, CC BY-SA 2.5,

`

https://commons.wikimedia.org/w/index.php?curid=1736396

3.2.2 Galactic red shift

If an object is observed to have a red or blue shifted spectrum then it is moving away or towards an observer respectively. In 1929, Edwin Hubble in 1929 discovered that all the galaxies were red-shifted in the spectrum. (Hubble 1929) This led to the discovery that the universe itself is in a state of expansion, and led to the big bang theory, which is about origin of the universe itself. Research into galaxy redshifts are being conducted to this day with observations being conducted across the electromagnetic spectrum. Deep Learning is helpful in this exploration as it can examine the spectrum of a large amount of data to help measure and categorize the galaxies.

3.2.3 Galaxy dynamics

The spectra of galaxies is much like the spectra of stars in that they reveal the behavior of these distant objects. The galaxy rotation rates show the changing velocities of stars which allow us to study the galaxy dynamics. Also, spectra show where star formation and deaths are occurring over time. Since there are many billions of galaxies in the universe, Deep Learning is a powerful tool to use since it can categorize and analyze single to multiple galactic spectra to examine behaviors of interest. (Hála 2014)

3.3 Optical telescopes

Large mirrored telescopes on Earth and space telescopes such as the Hubble space telescope are used to capture light in the visible spectrum. One research area where DL can be applied is the evolution of galaxies and their shapes, known as galactic morphology (discussed below). Another field of interest is watching the surface of the Sun to understand stellar interiors with DL helping to process the data. Another application of optical data is wide field detection, which looks at a very large piece of the sky in hope of discovering a new object. DL can help process the data and spot such objects unsupervised. (Iafrate and

`

Ramella 1990)

3.3.1 Galactic morphological classification

There are many billions of galaxies visible from Earth with the largest telescopes. The fainter the light we can see, the farther back in time we can view these galaxies. Subsequently, by viewing galaxies farther and farther away we can see how galaxies have evolved over time. Galaxies are initially classified into three different categories based on their shapes, elliptical, spiral, and lenticular. Each of these have subcategories with additional features. (Iafrate and Ramella 1990) DL can take wide field images containing many thousands of galaxies and classify them to high accuracy unsupervised, which provides many advantages for processing and collecting vast amounts of galaxy data. (Hocking et al. 2018)

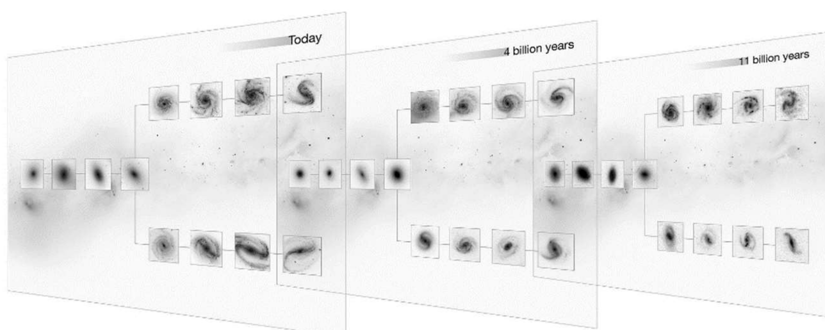

Figure 8: The Hubble galaxy morphological classification. (M. Kornmesser 2013)

4. Deep Learning Research in Astronomy and Astrophysics

How Deep learning is applied to astrophysics and astronomy is dependent on the observation methods, and some typical examples are listed in *Table 1*. literature review by (Hordri et al. 2017) outlines a way to analyze the effectiveness and quality of deep learning models using five features of the DL architecture:

- **Hierarchical layer**: DL has multiple learning layers and is designed to extract low-level features to high-level features in a hierarchical structure. DL research should employ this carefully in classification applications.

- **High-level abstraction**: This enables the extraction of data in unsupervised learning without external input. The architectures should represent the algorithms through the composition of non-linear transformations.
- **Processing of high volume of data**: How well does the DL method utilize large quantities of raw data for processing into results, and what is the amount of computer resources required?
- **Robust model**: A useful measure of a DL learning model is its ability to be applied to different types of problems.
- **Overfitting:** DL algorithms should not overfit to the training dataset.

Here, we apply the same criteria to eleven research papers on DL and astronomy/astrophysics.

New generation telescopes, such as the Atacama large millimeter array (ALMA), very large telescope (VLA), and sky survey telescope will produce enormous data that will require automatic processing. It is not enough to process the data – the processed data needs to be capable of answering scientific questions. One research area in astronomy is the classification of astronomical sources. Classification tasks are well suited to CNN; however, there are challenges in dealing with bias, misclassified data, and portability of the CNN to other datasets it is not trained on. (Chen and Lin 2014)

Unsupervised classification relies on the statistical properties of objects while supervised classification is inherently biased through inconsistent human assessment Research in *Table 2* shows classified variable stars and exoplanet transits using a recurrent-CNN, which uses the raw data to extract features. This enables data analysis directly from the raw data without the typically tedious data preparation conducted through human supervision. This study also simulates synthetic image sequences based on instrumentation and observations to adapt the training of the recurrent-CNN. Using this simulation for training allows the system to be fine-tuned to the data from the actual experiment using only a small number of real data points. (Carrasco-Davis et al. 2019)

`

Table 1: DL methods typically applied to astronomical data. (Fluke and Jacobs 2019)

	Convolutional Neural Network	**Recurrent Neural Network**	**Long-Short Term Memory (NN or RNN)**
Classification	• Stellar spectral type • Stellar photometric • Galactic morphological • Solar activity • Wide field detection	• Stellar photometric • Solar activity	• Stellar photometric
Regression		• Photometric redshifts • Spectral line parameter measurements	
Clustering	• Stellar spectral classification • Galaxy classification		
Time-Series Analysis		• Photometric planet detection • Variable star photometry	• Spectral planetary detection • Novel detection • Extrasolar planet detection

Table 2: Feature evaluation for deep learning for image sequence classification of astronomical events. (Carrasco-Davis et al. 2019)

DL Features	Deep Learning for Image Sequence Classification of Astronomical Events
Hierarchical layer	This study used a recurrent-CNN architecture with four layers of 21×21×64 data cubes followed by four layers of 11×11×64 and one layer of 6×6×64, then one layer of size 2304 followed by a 1024 layer. This is fed into an LSTM RNN of size 512. It then outputs data of size 7 for the classification. The architecture in this study is unusual as

	there are four identical layers. A good CNN should have multiple layers that are smaller than the previous layers to take advantage of the features. This model likely does not pick up nearly as many features as it could.
High-level abstraction	The data used in this study have a high level of abstraction and are suitable for this type of DL model.
Data volume processing	The volume of data for this DL architecture is questionable owing to the hierarchical layers of unusual sizes and the LSTM at the end of the system. It is likely a very slow model, but this is not discussed in the research article.
Robust model	This model in this study is claimed to be universal.
Overfitting	It is unlikely that this model would overfit due to the structure of the model and size of the dataset.

Unsupervised classification in several papers is based on the structural features of astronomical objects, such as galaxies. In *Table 3*, a CNN is used to classify galaxies into structural categories such as spheroid, irregular, point source, or unclassifiable using a Hubble super novae survey. One problem in the research was with the accuracy of category labels used in the training process of the CNN, as the labels were selected by humans, who were possibly biased. To reduce this bias, the authors used five experts to classify 100 randomly selected galaxies to help fine-tune the CNN training. (Pérez-Carrasco et al. 2019)

Table 3: Feature evaluation for multiband galaxy morphologies for CLASH: A convolutional neural network transferred from CANDELS. (Pérez-Carrasco et al. 2019)

DL Features	Multiband Galaxy Morphologies for CLASH: A Convolutional Neural Network Transferred from CANDELS

Hierarchical layer	The inception program uses a convolutional method that creates a computationally efficient program. For example, instead of creating a 5×5 grid with a filter, it creates two 3×3 grid with two filters, which is computationally more efficient. Ultimately, the program created a very large hierarchal structure through Inception and ConvNet code.
High-level abstraction	The data used in this study have a high level of abstraction and are suitable for this type of DL model.
Data volume processing	The inception code should create a highly efficient hierarchy, which would enable an effective data volume processing.
Robust model	This model uses considerable specialized data in the code that is specific for the application.
Overfitting	Article indicates that overfitting is not a problem with this study.

A similar study by (Longo, Merényi, and Tiňo 2019) The researchers used this dataset to train and compared with real data using a quantile regression forest that provided a matrix of similarities between the image and training data. Another study seen in *Table 4* in unsupervised classification of galaxies was conducted using a large-scale dataset of over 30,000 images of galaxies for deep k-means clustering and combining the data with feature selection. (Chattopadhyay, Fraix-Burnet, and Mondal 2019)Interestingly, the authors used only the observed data from photometer and spectroscopy, and they used no non-observational data. This approach was successful in identifying clusters of galaxies using different star formation histories. This study showed that DL was successful in categorizing data with strictly physical data and alleviated label bias in categorical applications.

Table 4: Feature evaluation for unsupervised classification of galaxies. I. Independent component analysis feature selection. (Chattopadhyay, Fraix-Burnet, and Mondal 2019)

`

DL Features	Unsupervised Classification of Galaxies. I. Independent component analysis feature selection
Hierarchical layer	Here, independent component analysis (ICA) is used to determine a set of derived independent components of the galaxies. K-means cluster analysis is then applied on nine independent components to obtain 10 distinct and homogeneous groups.
High-level abstraction	There is no high-level abstraction within k-means analysis.
Data volume processing	This approach has limited data processing owing to the limited computational processing of ICA.
Robust model	Once the ICA is conducted, the model is specific to the observational dataset. Hence, it is not universal.
Overfitting	Article indicates there is some risk of overfitting for large datasets

Table 5: Feature evaluation for identifying complex sources in large astronomical datasets using a coarse-grained complexity measure. (Segal et al. 2019)

DL Features	Identifying Complex Sources in Large Astronomical Datasets Using a Coarse-grained Complexity Measure
Hierarchical layer	A filter was applied to a 10×10 pixels and calibration with 256×256 radio image and 8-bit pixel intensity values. A 64×64 center of the image is cropped and filtered with window size h=1-.
High-level abstraction	There is very little high-level abstraction in this method.
Data volume processing	Data volume processing is good, but the data preparation required is excessive.
Robust model	There is little robustness in the model.
Overfitting	Article indicates there is risk of overfitting depending on the calibration

`

	method.

Deep CNN has also been applied to astrophysical simulations of stellar asteroseismology data as seen in *Table 6*. (Hon, Stello, and Yu 2017) It enabled the research group to interpolate between different stages of stellar evolution within the complex evolutionary models. The space telescopes, Kepler and TESS, which are both photometric measurements of stars over a large section of the sky, are predominantly time-series datasets extracted from telescopes hunting for exoplanets. There is an extremely large amount of data that requires the detection of transit events within these datasets.

Table 6: Feature evaluation for deep learning applied to the asteroseismic modeling of stars with coherent oscillation modes. (Hon, Stello, and Yu 2017)

DL Features	Deep Learning Applied to the Asteroseismic Modeling of Stars with Coherent Oscillation Modes
Hierarchical layer	This study used a fully connected dense network with a 6-dimensional input vector, a 1-dimensional output vector, and two *n*-dimensional layers in-between. The layers have 6, *n*, *n*, and one neuron.
High-level abstraction	The architecture is used to train and predict oscillation frequencies within stellar models. The hierarchical structure is easy to modify for various other applications.
Data volume processing	Data are processed at very high volumes.
Robust model	Article indicates that the model can be universally applied.
Overfitting	Article indicates that overfitting will be governed by the amount of data available.

One of the key points of interest in the hunt for exoplanets is the hunt for earth-like planets, and the

`

detection of such transits is the presence of red (correlated) noises in the light curves as seen in *Table 7*. Researchers in (Zucker and Giryes 2018) The researchers simulated the red noise through the application of parameters and found that the noise can be filtered by RNN to extract data for earth-like planets. Research conducted in *Table 8* (Yu et al. 2019) expanded an exoplanet DNN that had a 95% accuracy. The work illustrates the importance of including expert knowledge in state-of-the-art DL models when applying them to scientific research problems that seek to identify weak signals in noisy data. This research paved the way for DL to be employed as a major tool in astronomy and the hunt for exoplanets.

Table 7: Feature evaluation for Shallow transits—deep learning. I. Feasibility study of deep learning to detect periodic transits of exoplanets. (Zucker and Giryes 2018)

DL Features	Shallow Transits—Deep Learning. I. Feasibility Study of Deep Learning to Detect Periodic Transits of Exoplanets
Hierarchical layer	This project focused on a CNN with an input of 6144×2, which breaks into four hidden layers connected by 1527×40 followed by another four layers of size 377×32, 89×32, 17×32, and finally an output of 1×2.
High-level abstraction	The larger receptive field enables more global decisions at the earlier layers of the network.
Data volume processing	The data volume in this architecture should be enough.
Universal model	The model is universal to images as input.
Overfitting	The object is not subject to overfitting.

Table 8: Feature evaluation for identifying exoplanets with deep learning: a five-planet resonant chain around Kepler-80 and an eighth planet around Kepler-90. (Shallue and Vanderburg 2018)

DL Features	Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90

Hierarchical layer	This CNN is broken into five multiple layers: a single input of 6144×1, four hidden 1527×40, 277×32, 89×2, and 17×32, as well as an output of 1×2.
High-level abstraction	The larger receptive field enables more global decisions at the earlier layers of the network.
Data volume processing	The data volume in this architecture.is >10GB and is enough for DL problems
Universal model	Article indicates the model is universal to images as input.
Overfitting	Article indicates object is not subject to overfitting.

In cosmology, deep learning has been applied in research shown in *Table 9*. (Krachmalnicoff and Tomasi 2019), which developed a pixel-based approach to implement convolutional and pooling layers on the spherical surface estimation of cosmological parameters from simulated maps of the cosmic microwave background (CMB). This showed the applicability of the CNN to CMB parameter estimation using a simple NN architecture consisting of four convolutional and pooling layers. The researchers were able to sample value temperature and polarization parameters directly from simulated maps using this method, which achieved better accuracy than simulations using Bayesian methods.

Table 9: Feature evaluation for convolutional neural networks on the HEALPix sphere: a pixel-based algorithm and its application to CMB data analysis. (Krachmalnicoff and Tomasi 2019)

DL Features	Convolutional neural networks on the HEALPix sphere: a pixel- based algorithm and its application to CMB data analysis
Hierarchical layer	This group developed a pixel-based approach to implement convolutional on the spherical surface using a fully connected NN. The filter size swept across the image surface starts at 16 and reduces to 8,

`

	4, 2, and 1, where it then hits a dropout layer followed by a fully connected 48-neuron layer. This is passed to an activation function and finally an output layer.
High-level abstraction	The design can be adjusted to almost any image classification problem.
Data volume processing	High-volume data can be processed on this system.
Universal model	The model is highly universal.
Overfitting	The model does not overfit.

Deep learning is applied to Cosmological redshift is an important measurement in work by (Beck and Sadowski 2019) using data from Sloan Digital Sky Survey (SDSS), which contains spectrum from galaxies with about 2 million objects. The galaxies are mapped onto a Graph convolutional network, where each galaxy (or point on the geodesic sphere) is an activation function. Each galaxy has five photometric measurements through different color filters. The DL system is used on three different architectures, a single model using the five color inputs trained with a mean square error, a 'MoNet' architecture that utilized galaxy neighbor information, and recurrent versions of the previous two architectures adding hyper-parameter tuning using SHERPA software. (Hertel et al. 2018)

*Table 1*0 shows the single-object model, which consists of 512 ReLU units and three hidden layers was run with ten percent dropout (dropout is a regularization method to prevent overfitting) in the final layer and had a single output with the calculated mean square error. It was found that the model was unable to incorporate neighborhood information and the method was shelved. (Beck and Sadowski 2019) The MoNet was trained to incorporate neighbor information using the first predictions of the NN. It uses weights from photometry of the neighbor galaxies, their position from the graph convolutional network, and the distance between each other. The method showed a 10% increase in performance over the single-object model (ibis). The SHERPA experiments showed no improvements in performance but show promise once the architecture can be pass to multiple Graphics Processing Units. (ibis)

`

Table 10: Refined Redshift Regression in Cosmology with Graph Convolution Networks (Beck and Sadowski 2019)

DL Features	DL applied to a convolutional graph with a Single object model, MoNet architecture, and hyper-parameter tuning
Hierarchical layer	All three models are built on a convolutional graph having each node of the graph as an ReLU activation function. • The single object model three hidden layers of 512 ReLU units with 10% dropout in last layer, followed by soft plus output. • MoNet architecture has three layers of 256 ReLU units and 10 linear outputs. Weighted averages are calculated using nearest neighbors. • SHERPA hyper-parameter tuning took the weights for nearest neighbors and the single object model and put them in an RNN architecture.
High-level abstraction	Abstraction of the data for this model can be put into context of other systems which can be represented by Graph convolution network.
Data volume processing	No information on amount of data processed
Robust model	The structure of these models is very specific to data from SDSS, and would be difficult to port to another system. Data would need to be represented on a graph convolution network.
Overfitting	Overfitting is avoided using dropout at 10% on the final output.

The latest solar telescope, the Daniel K. Inouye Solar Telescope (DKIST) is providing the large quantities of data at high resolution, and (Dodds et al. 2020) look to DL as a computational method to create approximate inference on photosphere data in 4D. Using parameters of the magnetic Naiver-Stokes equations and the magnetic field vectors at each pixel 3D convolutional NN were created. Table 11 shows they found that the DL method was able to speed up the inversion calculations by about 100 times compared to previous methods. [ibis]

Table 11: Inverting Solar Spectro polarimetric Observations with Deep Learning (*Dodds et al. 2020)*

DL Features	Calculates 3D inference of the photosphere over time
Hierarchical layer	Four convolution models with 22 layers containing 3D layers with ReLU activation functions and a 3x3x3 kernel. Followed by 5% dropout, 3D max-pooling 2x2x2, four transposed convolutional models.
High-level abstraction	Data from any magneto hydrodynamic system should be able to be applied to this model.
Data volume processing	Training set used 9.2TB of DKIST data. Computation was seen to be approximately 7.4x104 pixels/s which is up from 6x102 pixel/sec.
Robust Model	Can be applied to model using any data set
Overfitting	Overfitting is not discussed in article.

5. Discussion and Conclusion

Out of the 65 papers found in the literature search eleven papers were selected for their different applications, representing 17% of the population, which is a good sample size. Of these 17%, 63% use spectroscopy, 36% use photometry, and one paper used optical observations. Also, 54% use convolutional networks of some type, 18% use sequential learning (LSTM), and 28% fit into an 'other' category.

Over the previous eleven tables this literature review examined the qualities of the research through a method outlined in a similar literature review by (Hordri et al. 2017). It provides a way to analyze the effectiveness and quality of deep learning models using hierarchy, high-level abstraction, data volume processing, robustness, and possibility of overfitting. To gain an understanding to how these qualities are related to the method of observation*, Table 12* compares the method of observation to the DL features in the eleven papers selected. Each percentage represents how many papers using the three observational methods of spectroscopy, photometry, and optical have the qualities of a good deep learning system.

Table 12: Percentage of studies using DL features compared to observation method

	All observations	Photometry	Optical	Spectroscopy
Hierarchical layer	81%	29%	0.81%	51%
High-level abstraction	72%	25%	0.72%	45%
Data volume processing	45%	16%	0.45%	28%
Robust Model	54%	19%	0.54%	34%
Overfitting	36%	13%	0.36%	22%

Table 12 shows that the bulk of the studies used enough hierarchical layers in their models, with 54% of those being CNN. This indicates the importance of the hierarchical layer in all forms of DL. The DL abstraction was present in 72% of the studies with the majority of those using spectroscopic methods. Data volume processing scored low at 43% with each of the categories scoring low in all the studies, the highest being 22%. The robustness of the models is only 54%, with the highest being spectroscopic observations. Overfitting scores low in all the observations, with the highest observation method (spectroscopy)having 22% of the studies with enough countermeasures to avoid overfitting.

The information in *Table 12* shows that the weakness in DL lies most significantly in the data volume processing and the overfitting areas. This could be an indication of the direction research should take with regards to constructing DL systems for astronomy. Higher data volume processing and mitigating overfitting could make these models significantly more effective.

`

This literature review examined the origins of DL at the basic level. CNN and RNN were defined in the LSTM form. The review further examined how DL has been influencing astronomy. We found that DL is still at an early stage of application in astronomy. Further, we found that astronomers are considering DL as the main tool for processing data from the next generation of telescopes. The literature on DL in astronomy applications highlights transfer learning as a major concern for future research. (Vilalta et al. 2019) Transfer learning is the ability to take data from one type of data and apply it to another. For example, taking photometric data from TESS and applying it to spectroscopic data, the authors (Ansdell et al. 2018) used a new approach to the time domain adaptation of the problem. The literature also reveals additional problems within the application of DL to astronomy. Learning algorithms are not ideal for astronomical datasets, as there is considerable variation in the quality of astronomical data, and currently, there is no straightforward way to handle this within the DL algorithms. Specifically, astronomical measurements vary in terms of signal-to-noise ratios and measurement uncertainties, and training simulations having modelling uncertainties. The construction of classifiers that can handle taking a ML model trained on one dataset and applying it to another, is of a high priority. Additionally, the new telescopes will be required to process data as it is being collected. Hendriks and Aerts (2019) also call for deep CNN that can present data in scientifically interpretable ways and are physically meaningful within the astrophysical models being tested.

References

Alom, Md Zahangir, Tarek M Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S Awwal, and Vijayan K Asari. 2018. “The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches.” *CoRR* abs/1803.0 (March). http://arxiv.org/abs/1803.01164.

Amelio, G F, M F Tompsett, and G E Smith. 1970. “Experimental Verification of the Charge Coupled Device Concept.” *The Bell System Technical Journal* 49 (4): 593–600. https://doi.org/10.1002/j.1538-7305.1970.tb01791.x.

Ansdell, Megan, Yani Ioannou, Hugh P. Osborn, Michele Sasdelli, Jeffrey C. Smith, Jon M. Jenkins, Chedy Raissi, and Daniel Angerhausen. 2018. “Scientific Domain Knowledge Improves Exoplanet Transit Classification with Deep Learning,” no. Borucki 2016: 3–10. https://doi.org/10.3847/2041-8213/aaf23b.

Baron, Dalya, Dovi Poznanski, Darach Watson, Yushu Yao, Nick L.J. Cox, and J. Xavier Prochaska. 2015. “Using Machine Learning to Classify the Diffuse Interstellar Bands.” *Monthly Notices of the Royal Astronomical Society* 451 (1): 332–52. https://doi.org/10.1093/mnras/stv977.

Beck, Robert, and Peter Sadowski. 2019. “Refined Redshift Regression in Cosmology with Graph Convolution Networks,” no. NeurIPS.

Carrasco-Davis, Rodrigo, Guillermo Cabrera-Vives, Francisco Förster, Pablo A. Estévez, Pablo Huijse, Pavlos Protopapas, Ignacio Reyes, Jorge Martínez-Palomera, and Cristóbal Donoso. 2019. “Deep Learning for Image Sequence Classification of Astronomical Events.” *Publications of the Astronomical Society of the Pacific* 131 (1004): 108006. https://doi.org/10.1088/1538-3873/aaef12.

Chattopadhyay, Tanuka, Didier Fraix-Burnet, and Saptarshi Mondal. 2019. “Unsupervised Classification of Galaxies. I. Independent Component Analysis Feature Selection.” *Publ. Astron. Soc. Pac.* 131 (1004): 108010. https://doi.org/10.1088/1538-3873/aaf7c6.

Chen, Xue Wen, and Xiaotong Lin. 2014. “Big Data Deep Learning: Challenges and Perspectives.” *IEEE Access* 2: 514–25. https://doi.org/10.1109/ACCESS.2014.2325029.

Crowther, Paul A. 2007. “Physical Properties of Wolf-Rayet Stars.” *Annual Review of Astronomy and Astrophysics* 45 (1): 177–219. https://doi.org/10.1146/annurev.astro.45.051806.110615.

Dodds, S Curt, Ian Cunnyngham, Lucas Tarr, Sarah Jaeggli, Tom Schad, and Peter Sadowski. 2020. “Inverting Solar Spectropolarimetric Observations with Deep Learning,” no. NeurIPS 2019: 1–6.

Engler, Simon T. 2017. “Forecasting of Energy Requirements for Planetary Exploration Habitats Using a Modulated Neural Activation Method.” *Masters Thesis University of Calgary*. University of Calgary. https://doi.org/http://dx.doi.org/10.11575/PRISM/26208.

Fluke, Christopher J., and Colin Jacobs. 2019. “Surveying the Reach and Maturity of Machine Learning and Artificial Intelligence in Astronomy." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*. https://doi.org/10.1002/widm.1349.

Fukushima, Kunihiko. 1988. "Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition." *Neural Networks* 1 (2): 119–30. https://doi.org/10.1016/0893-6080(88)90014-7.

Gerstner, Wulfram. 2001. "Coding Properties of Spiking Neurons: Reverse and Cross-Correlations." *Neural Networks*. https://doi.org/10.1016/S0893-6080(01)00053-3.

Gomez Gonzalez, C. A., O. Absil, and M. Van Droogenbroeck. 2018. "Supervised Detection of Exoplanets in High-Contrast Imaging Sequences." *Astronomy & Astrophysics* 613: A71. https://doi.org/10.1051/0004-6361/201731961.

Gupta, Ankita, Gurunath Gurrala, and P. S. Sastry. 2018. "An Online Power System Stability Monitoring System Using Convolutional Neural Networks." *IEEE Transactions on Power Systems* 34 (2): 864–72.

Hála, Pavel. 2014. "Spectral Classification Using Convolutional Neural Networks" arXiv:1412 (December). http://arxiv.org/abs/1412.8341.

Hendriks, L., and C. Aerts. 2019. "Deep Learning Applied to the Asteroseismic Modeling of Stars with Coherent Oscillation Modes." *Publications of the Astronomical Society of the Pacific* 131 (1004): 108001. https://doi.org/10.1088/1538-3873/aaeeec.

Hertel, Lars, Peter Sadowski, Julian Collado, and Pierre Baldi. 2018. "Sherpa : Hyperparameter Optimization for Machine Learning Models." *NIPS Workshop*, no. Nips. https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/.

Hinners, Trisha A., Kevin Tat, and Rachel Thorp. 2018. "Machine Learning Techniques for Stellar Light Curve Classification." *The Astronomical Journal* 156 (1): 7. https://doi.org/10.3847/1538-3881/aac16d.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9 (8): 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

Hocking, Alex, James E. Geach, Yi Sun, and Neil Davey. 2018. "An Automatic Taxonomy of Galaxy

Morphology Using Unsupervised Machine Learning.” *Monthly Notices of the Royal Astronomical Society* 473 (1): 1108–29. https://doi.org/10.1093/mnras/stx2351.

Hon, Marc, Dennis Stello, and Jie Yu. 2017. “Deep Learning Classification in Asteroseismology.” *Monthly Notices of the Royal Astronomical Society* 469 (4): 4578–83. https://doi.org/10.1093/mnras/stx1174.

Hordri, N. F., A. Samar, S. S. Yuhaniz, and S. M. Shamsuddin. 2017. “A Systematic Literature Review on Features of Deep Learning in Big Data Analytics.” *International Journal of Advances in Soft Computing and Its Applications* 9 (1): 32–49.

Hubble, Edwin. 1929. “A Relation between Distance and Radial Velocity among Extra-Galactic Nebulae.” *Proceedings of the National Academy of Science* 15 (3): 168–73. https://doi.org/10.1073/pnas.15.3.168.

Iafrate, G, and M Ramella. 1990. “The Hubble.” *Nature* 344 (6269): 885–885. https://doi.org/10.1038/344885b0.

Krachmalnicoff, N., and M. Tomasi. 2019. “Convolutional Neural Networks on the HEALPix Sphere: A Pixel-Based Algorithm and Its Application to CMB Data Analysis.” *Astronomy & Astrophysics* 628: A129. https://doi.org/10.1051/0004-6361/201935211.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. 2017. “ImageNet Classification with Deep Convolutional Neural Networks.” *Communications of the ACM* 60 (6): 84–90. https://doi.org/10.1145/3065386.

Kuntzer, T., M. Tewes, and F. Courbin. 2016. “Stellar Classification from Single-Band Imaging Using Machine Learning.” *Astronomy and Astrophysics* 591. https://doi.org/10.1051/0004-6361/201628660.

Landolt, Arlo U. 1992. “UBVRI Photometric Standard Stars in the Magnitude Range 11.5 < V < 16.0 Around the Celestial Equator.” *\aj* 104 (July): 340. https://doi.org/10.1086/116242.

Longo, Giuseppe, Erzsébet Merényi, and Peter Tiňo. 2019. “Foreword to the Focus Issue on Machine Intelligence in Astronomy and Astrophysics.” *Publications of the Astronomical Society of the*

`

Pacific 131 (1004): 100101. https://doi.org/10.1088/1538-3873/ab2743.

M. Kornmesser. 2013. "The Hubble Sequence throughout the Universe's History." *Spacetelescope.Org*. ESA/Hubble. http://www.spacetelescope.org/news/heic1315/.

Maclaurin, Dougal, David Duvenaud, and Ryan P. Adams. 2015. "Gradient-Based Hyperparameter Optimization through Reversible Learning" 37. http://arxiv.org/abs/1502.03492.

Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. MIT press.

Nair, V. (U of Toronto), and G.E. (U of Toronto) Hinton. 2010. "Rectified Linear Units Improve Restricted Boltzmann Machines." *27th International Conference on Machine Learning*. https://doi.org/10.1123/jab.2016-0355.

Pearson, Kyle A., Leon Palafox, and Caitlin A. Griffith. 2018. "Searching for Exoplanets Using Artificial Intelligence." *Monthly Notices of the Royal Astronomical Society* 474 (1). https://doi.org/10.1093/mnras/stx2761.

Pérez-Carrasco, M., G. Cabrera-Vives, M. Martinez-Marin, P. Cerulo, R. Demarco, P. Protopapas, J. Godoy, and M. Huertas-Company. 2019. "Multiband Galaxy Morphologies for CLASH: A Convolutional Neural Network Transferred from CANDELS." *Publications of the Astronomical Society of the Pacific* 131 (1004): 108002. https://doi.org/10.1088/1538-3873/aaeeb4.

Russell, Henry Norris. 1914. "Relations Between the Spectra and Other Characteristics of the Stars." *Popular Astronomy* 22 (May): 275–94.

Sadowski, Peter. 2016. "Deep Learning for Experimental Physics." *ProQuest Dissertations and Theses*, 83. http://libproxy.library.wmich.edu/login?url=https://search.proquest.com/docview/1847984109?accountid=15099%0Ahttp://primo-pmtna01.hosted.exlibrisgroup.com/openurl/01WMU/01WMU_SERVICES??url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mtx:dissertation&genr.

Segal, Gary, David Parkinson, Ray P Norris, and Jesse Swan. 2019. "Identifying Complex Sources in Large Astronomical Data Sets Using a Coarse-Grained Complexity Measure." *Publications of the*

Astronomical Society of the Pacific 131 (1004): 108007. https://doi.org/10.1088/1538-3873/ab0068.

Shallue, Christopher J., and Andrew Vanderburg. 2018. "Identifying Exoplanets with Deep Learning: A Five-Planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90." *The Astronomical Journal* 155 (2): 94. https://doi.org/10.3847/1538-3881/aa9e09.

Skoog, D A, F J Holler, and S R Crouch. 2007. *Principles of Instrumental Analysis*. International Student Edition. Thomson Brooks/Cole. https://books.google.com/books?id=GrOsQgAACAAJ.

Vilalta, Ricardo, Kinjal Dhar Gupta, Dainis Boumber, and Mikhail M. Meskhi. 2019. "A General Approach to Domain Adaptation with Applications in Astronomy." *Publications of the Astronomical Society of the Pacific* 131 (1004): 108008. https://doi.org/10.1088/1538-3873/aaf1fc.

Vinyals, Oriol, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. "Show and Tell: A Neural Image Caption Generator." *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 07-12-June: 3156–64. https://doi.org/10.1109/CVPR.2015.7298935.

Xu, Danfei. 2020. "Convolutional Neural Networks for Visual Recognition." 2020. http://cs231n.stanford.edu/.

Yang, Zhilin, Ye Yuan, Yuexin Wu, Ruslan Salakhutdinov, and William W. Cohen. 2016. "Review Networks for Caption Generation." *Advances in Neural Information Processing Systems*, no. Nips: 2369–77.

Yann, LeCun, and Bengio Yoshua. 1995. "Convolutional Networks for Images, Speech, and Time-Series" 4 (April 2016): 2571–75. https://doi.org/10.1109/IJCNN.2004.1381049.

Yu, Liang, Andrew Vanderburg, Chelsea Huang, Christopher J. Shallue, Ian J. M. Crossfield, B. Scott Gaudi, Tansu Daylan, et al. 2019. " Identifying Exoplanets with Deep Learning. III. Automated Triage and Vetting of TESS Candidates ." *The Astronomical Journal*. https://doi.org/10.3847/1538-3881/ab21d6.

Zucker, Shay, and Raja Giryes. 2018. "Shallow Transits—Deep Learning. I. Feasibility Study of Deep Learning to Detect Periodic Transits of Exoplanets." *The Astronomical Journal* 155 (4). https://doi.org/10.3847/1538-3881/aaae05.