Neural Networks - Assignment 3

Galaxy merger recognition

Michelle Willebrands & Eva van Weenen

Abstract

We apply a convolutional neural network to SDSS DR10 images of merging galaxies from the catalog of Darg et al. (2010a). These galaxies were identified as mergers with the Galaxy Zoo project (Lintott et al., 2011) using the weighted-merger-vote-fraction f_m and we investigate whether a neural network can learn to predict this fraction. Our data set consists of 3003 galaxy-pairs from the Galaxy Zoo Merger catalog (Darg et al., 2010a) and 10.000 non-merging galaxies of the Galaxy Zoo catalog. We use a neural network with two convolutional hidden layers followed by one fully-connected hidden layer. We test the performance of various initializations of the network on a data set consisting of both the mergers and non-mergers (the *complete sample*) and on a data set consisting of only mergers (the *merger sample*). We find that the neural network with the best initialization reaches a validation accuracy of 66% on the complete sample and a validation accuracy of 28% on the merger sample. When we define a *plus-minus accuracy*, the validation accuracy is increased to 70% on the merger sample. As this performance is still not very high, we test whether a neural network can identify the galaxy images as mergers or not, using binary class labels. In this case we reach an accuracy of 91% on the validation set. The main issue in this neural network application is the lack of merger data. With future observations, more merging galaxies might be identified and the performance of this network could be improved.

1 Introduction

The first Galaxy Zoo project launched in 2007 and was a revolutionary way of handling the large data sets involved in Astronomical research. Citizens could easily classify images of galaxies online based on their morphology, speeding up the process of analyzing each individual image by eye immensely. With the Galaxy Zoo project, over 800.000 galaxies have been classified according to the Hubble sequence[1]. This Hubble classification scheme can be found in Fig. 1. Multiple versions of the project have been established since, resulting in classifications of for example Active Galactic Nuclei, Green Peas and merging galaxies. The latter will be focused on in this project.

Two (or more) galaxies merge when they come so close by each other that they are influenced, or even disrupted, by the gravitational field of the other galaxy. The orderly orbits of the stars therefore change, as does the position of the gas and dust. This results not only in an irregular shape of the two galaxies during the merger, but also in a large increase of the star formation rate due to shocks in gas. Merging galaxies are therefore very interesting to study: they provide insight about the formation and evolution of stars, shock waves in galaxies, plasmas, ionized gas, Active Galactic Nuclei, the formation and evolution of super-massive black holes and many more topics. Thus, mergers are a unique environment in which the formation of stars can be studied. Studying mergers can provide insight in the evolution of our own galaxy, as it is predicted to merge with the Andromeda galaxy in 4 billion years (Sohn et al., 2012).

In 2010, the collection of merging galaxies obtained by Darg et al. (2010a) in the Galaxy Zoo project, was the largest ever. Their catalog consisted of 3003 visually selected pairs of merging galaxies. In a companion paper, Darg et al. (2010b), they investigated the physical properties of the mergers, using data of the Sloan Digital Sky Survey (SDSS). Even though citizen science seems a perfect way of classifying vast amounts of data, there are drawbacks. Many projects similar to Galaxy Zoo are created every year and eventually not enough people might be interested in participating. Therefore, applying a neural network on these data would be of great importance, as neural networks are known to be able to visually classify images on a very large scale. We will apply a convolutional neural network to images of SDSS and use the labels of the catalog of Darg et al. (2010a). Using different network layouts, different optimizers, batch sizes and training sets, we will research the extent to which neural networks can classify merging galaxies. In section 2, we will first describe the data and how we retrieved and processed it. Subsequently, we will describe the architecture of the convolutional neural network we used, as well as its performance on the data set in section 3. At the end, we will provide conclusions and a discussion in sections 4 and 5. The final section, 6, provides the acknowledgements.

If you are uncommon with astronomy and would like to read more, we kindly refer you to the footnotes for more information about specific astronomical concepts.

[1] The **Hubble sequence** is a classification system for galaxies, in which galaxies are classified based on their morphologies, such as the number of spiral arms and the size of the bar in the galaxy.

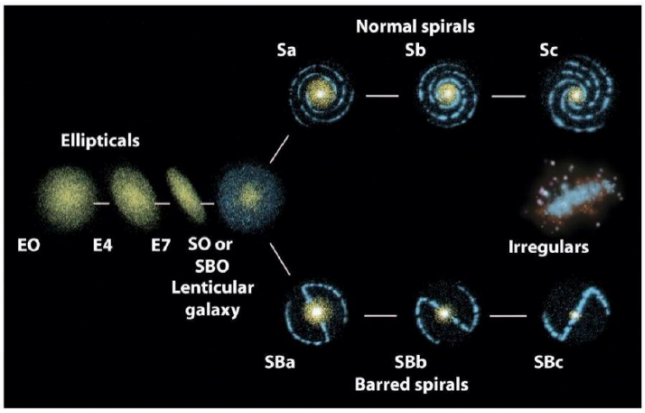


Figure 1: The Hubble classification scheme of galaxies. The main distinction is between elliptical and spiral galaxies. The latter can have a bar in the center. In the right side of the image, the irregular category is shown, these are merging galaxies. Imaged retrieved from Jenkinson et al. (2015).

2 Data

The Sloan Digital Sky Survey (SDSS) (York et al., 2000) is an astronomical survey that started in the year 2000 using the 2.5m optical telescope at Apache Point Observatory in New Mexico, United States. It has observed 500 million objects in 5 photometric filters[2] and has observed spectroscopic information of 3 million stars[3]. The photometric filters that are used for SDSS are called ultraviolet u (at a wavelength of 354.3 nm), green g (477.0 nm), red r (623.1 nm), near-infrared i (762.5 nm) and infrared z (913.4 nm). The main sample of galaxies from SDSS has a median redshift[4] of $z = 0.1$.

From the galaxy sample of SDSS Data Release 6 (DR6), 893 292 galaxies in the local Universe ($0.005 < z < 0.1$) have been visually classified with the Galaxy Zoo project Lintott et al. (2011). Galaxies were classified as *elliptical galaxies*, *clockwise/Z-wise spiral galaxies*, *anti-clockwise/S-wise spiral galaxies*, *other spiral galaxies* (that could not be further classified as their rotation only allowed us to look at the edge), *stars or don't know* (such as artifacts) and *mergers*. The decision tree for these classifications was based on morphological characteristics of galaxies such as the number of spiral arms. All irregular shaped galaxies are classified as mergers. However, the threshold for when a galaxy is irregular is not clearly defined. One person might classify an image as a merger, whereas someone else might decide that it is not. As each image in the Galaxy Zoo is classified by multiple people, the 'weighted-merger-vote-fraction' f_m can be defined. This parameter is defined as the fraction of people that classified a galaxy as a merger, multiplied by a weighing factor that corrects for the quality of the votes of an individual user. When this parameter is close to 1, the system is likely to be a merger, but a value close to 0 indicates that no merger is occurring.

Darg et al. (2010a) selected the galaxies with a high enough weighted-merger-vote-fraction: $0.4 < f_m < 1$ resulting in a total of 4198 galaxies. If only one of the two merging galaxies has an $f_m > 0.4$, both galaxies of the merger-pair are included in this set. Of this sample, they verified which galaxies were mergers and which were not, resulting in a catalog of 3003 mergers. This catalog consists of the names, astrometric positions and weighted-merger-vote-fraction of all galaxies involved in the merger, so for a total of 6006 galaxies as there are two galaxies involved in a merger. For our research, we use the galaxies from this catalog, and retrieve the SDSS images of these galaxies. We will retrieve the images of both galaxies involved in the merger, resulting in a double set of 3003 images. The image of a galaxy one of a merger is then slightly shifted with respect to the image of galaxy two of a merger as it is centered in a different astrometric position. This method might prove to be useful to prevent overfitting (Devries and Taylor, 2017). The images we will use are the images of a later data release of SDSS, DR10, because the images are of a better quality. The images from SDSS DR10 contain a wide field of stars, so we restricted this field to a width and height of 40 arcsec centered around the merger to ensure that the neural network only learns about the galaxy and not the stars around it. The resulting, cropped images contain almost no background stars. Keep in mind that galaxies that are far away usually cover a smaller area on the sky, so that not each galaxy is the same size within these 40 arcsec images. We retrieve the

[2]From observations in different **photometric filters** we can determine the color of an astronomical object (galaxies, stars, comets, etc.). The color of a star tells us its temperature, its age and it indicates different processes that are happening inside a star. The color of a region in a galaxy indicates for example the age of stars in that region, whether star formation is taking place at that region.

[3]**Spectroscopic information** gives us information about the age and composition of a star, and about the speed with which astronomical objects are receding away from us.

[4]The **redshift** means that an object is moving away from us and is essentially a way of comparing distances of far away celestial objects.

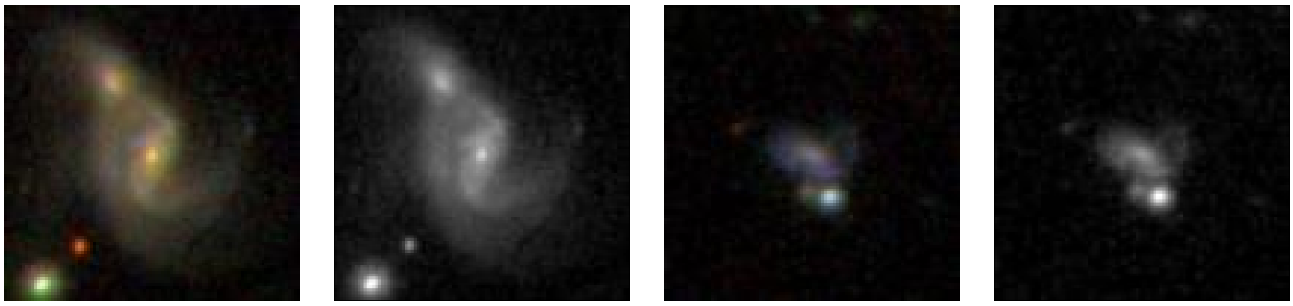


Figure 2: Two example images from our data set of merging galaxies. These images demonstrate the variation in size of the galaxies, as well as the presence of background stars. The colored and the final gray-scale images that are used as input for the network are shown side-by-side.

SDSS JPEG images using the JHU img cutout service (O'Mullane, 2004). The JPEG images are a composite of images in the g, r and i filters, thus creating a color image. The resulting images have 100×100 pixels and three channels. As the color of an image is unrelated to the probability of being a merger, we average over these channels to prevent any bias in our sample. The network does not need to use the colors in the learning process, but should only classify the images based on the morphology. The resulting images are then converted to gray scale.

Besides the sample of 6006 images with merging galaxies, we also want to add images of non-merging galaxies. In this way, the data set is more complete and is a more realistic representation of all galaxies on the sky. Therefore, we download another 10 000 images from SDSS DR10 at random, containing only non-merging galaxies that were also in the original Galaxy Zoo project. We manually label these images with $f_m = 0$. Of the 10 000 images, we manually remove approximately 100 images that contained strange artifacts such as cosmic rays or saturation bleeds, or images in which the galaxy was not visible due to a bright star closeby in the image.

From the images of merging- and nonmerging galaxies combined we create two different samples: the training- and test set. The training set consists of two thirds of the images of the samples and the test set consists of the remaining one third. With the data set complete, we now continue with encoding the class labels, y. The parameter f_m ranges from 0 to 1 and we bin the parameter in bins of size 0.1, resulting in 11 classes. We use one-hot encoding for the labels: class i is represented by an 11-dimensional vector that is all-zeros, except for a 1 at index i.

3 Convolutional neural network

3.1 Architecture

For the architecture of the convolutional neural network, we use the `mnist_cnn.py` code from Keras-team (2018) as a starting point and rebuild it for our own SDSS images. We use a convolutional neural network, consisting of an input layer with 100×100 nodes, followed by two convolutional layers and one dense layer and finally the output layer with the 11 classes. The complete architecture is as follows:

- ↓ Input layer with the 10000 (100×100) nodes
- ↓ Convolutional layer with N_1 nodes, 3×3 filter size and *relu* activation function
- ↓ Convolutional layer with N_2 nodes, 3×3 filter size and *relu* activation function
- ↓ Max pooling of 2×2
- ↓ Dropout of percentage D_1
- ↓ Dense layer with N_3 nodes and a *relu* activation function
- ↓ Dropout of percentage D_2
- ↓ Dense output layer with 11 nodes and a *softmax* activation function

The max pooling layer reduces the number of parameters in the model and is a form of down-sampling. The 2×2 pooling filter is moved over the previous layer and selects only the pixel with the maximum value within the area. By doing this it discards 75 % of the pixels and prevents over-fitting. The dropout disables a percentage of connections and is also added to prevent overfitting. The convolutional network produces a vector with the length of the number of classes, providing the probability that the input image belongs to each of these classes.

The predicted outcome for the image, $\hat{y}$, is the class with the highest probability.

Once the labels f_m of all data are one-hot encoded, the model can be trained. We train the network on two thirds of the images (4004 merging galaxies and 6700 non-merging galaxies) and validate it on the remaining data (2002 mergers and 3300 non-mergers). We make sure to use the 4004 images that contain the first 2002 galaxy pairs for the training data, so that there is no overlap in the training and test images.

The default settings of the Keras-team (2018) code is a batch-size of 128, a node-size of $N_1 = 32$ for the first convolutional layer, $N_2 = 64$ nodes for the second convolutional layer with a dropout of $D_1 = 25\%$ and $N_3 = 128$ nodes for the third hidden layer with a dropout of $D_2 = 50\%$. Furthermore, the loss function is *categorical cross-entropy*, the optimizer is *Adadelta* and the number of training epochs is set to 12. With these 'default' settings, we achieve a loss of 0.74 and a accuracy of approximately 73% on the training data. On the validation data, these values are 0.94 and 67% respectively. These relatively low values might be due to a variety of causes, so in the next section we describe how we adapt the network and the data set in order to improve the results.

3.2 Optimization of the network for the complete dataset

3.2.1 Adapting the network

We start out by experimenting with different initializations of the convolutional neural network. In each new run, we adapt one of the parameters slightly to see how this affects the resulting losses and accuracies on both the training and test data. In the first five runs, we only modify the model itself, not the data (see table 1). We change the batch size and the optimizer that is used. We do not change the loss-function, as the loss-function *categorical-crossentropy* is best when probabilities are returned in the output layer. The activation function *relu* remains unchanged as well. We use it because it speeds up the training process (Krizhevsky et al., 2012). We train the network for a total of 12 epochs and write down the accuracy and loss at the point where the validation loss is minimal. In later epochs, the model starts to over-fit the data, this becomes evident from a very high accuracy on the training data and a very low accuracy on the test data. The corresponding losses and accuracies are displayed in table 2.

Batch size We find that increasing the batch-size (run 2) does not change the accuracy on the training and test set. According to Keskar et al. (2017), using a larger batch size might even lead to a degradation in the quality of the model. Therefore, it might be interesting to see if using a smaller batch size can improve our network's ability to learn. Even though the current batch size is not very large (128), we change it to 32 and run the code again. We expect that there might be a trade-off between the batch size and the number of learning epochs that are necessary, so we increase the number of epochs to 20. However, we find that the minimal loss on the validation data has already been reached after 3 epochs. The achieved accuracy on the test data is now 27.9 %. Apparently, this change in the batch size does not improve (or worsen) the accuracy at all. This is not unexpected, because the batch size of 128 was not very large to begin with. The computation time has, however, increased from approximately 9 seconds per epoch to 13 seconds per epoch, so we decide to retain the original value of 128 for the batch size in the following runs.

Optimizer In the next run, we change the optimizer from *Adadelta* to *Adam*. This change does not seem to make a difference in accuracy, as we can observe in tables 1 and 2 for run 4, but we do change the optimizer from *Adadelta* to *Adam* for the next experiments. *Adam* should work better than *Adadelta*, because *Adam* is an extension of *Adadelta* and considers an exponential average of past gradients, the momentum (Kingma and Ba, 2014).

Number of nodes In run five, we experiment with the number of nodes in the hidden layers of the network. We double the number of nodes in all hidden layers, but find that the accuracy slightly decreases when we compare it with previous experiments.

3.2.2 Adapting the ratio of mergers to non-mergers

Overall, none of the modifications resulted in a significant improvement of the values of the loss and accuracy. The accuracies on the test set were not higher than 70% and were not improved by changing the batch-size, number of nodes or the optimizer. Therefore, we figure the data set itself might actually prove a challenge for the network. With the relatively high fraction of non-mergers in the data, the network might not be able to learn the mergers and their corresponding f_m well. Because of this, we include less non-mergers in the next two runs: the sixth run with the number of non-mergers reduced to one-third and the seventh run containing only merging galaxies. Furthermore, we keep the number of nodes to the increased settings, as it did not seem

Run	Batch size	Epochs	Nodes (N_1, N_2, N_3)	Filter size	Dropout (D_1, D_2)	Optimizer	Ratio train-test	# mergers	# no mergers	Accuracy
1	128	12	32, 64, 128	3×3	25%, 50%	Adadelta	2:1	4004	6700	Categorical
2	**256**	12	32, 64, 128	3×3	25%, 50%	Adadelta	2:1	4004	6700	Categorical
3	**32**	12	32, 64, 128	3×3	25%, 50%	Adadelta	2:1	4004	6700	Categorical
4	128	12	32, 64, 128	3×3	25%, 50%	**Adam**	2:1	4004	6700	Categorical
5	128	12	**64, 128, 256**	3×3	25%, 50%	Adam	2:1	4004	6700	Categorical
6	128	4	64, 128, 256	3×3	25%, 50%	Adam	2:1	4004	**2002**	Categorical
7	128	4	64, 128, 256	3×3	25%, 50%	Adam	2:1	4004	**0**	Categorical
8	128	4	32, 64, 128	3×3	25%, 50%	Adam	2:1	4004	0	**Plus-minus**
9	128	5	**32, 2·64, 2·128**	3×3	25%, 50%	Adam	2:1	4004	0	Plus-minus
10	128	5	32, 64, 128	3×3	25%, 50%	Adam	**4:1**	4004	0	Plus-minus
11	128	8	32, 64, 128	3×3	**50%, 75%**	Adam	2:1	4004	0	Plus-minus

Table 1: Initializations of the convolutional neural network and the data for each run. Each horizontal line indicates a different testing phase (adapting the network, adapting the data, adapting the accuracy measure). The parameter in bold text indicates the parameter that was changed for the initialization. The blue rows indicate the best found initializations as explained in sections 3.2 and 3.3.

Run	Training loss	Training accuracy	Validation loss	Validation accuracy	Plus-minus accuracy
1	0.74	73 %	0.94	67 %	
2	0.81	71 %	0.92	67 %	
3	0.78	71 %	0.92	67 %	
4	0.78	71 %	0.93	66 %	
5	0.59	84 %	0.99	60 %	
6	1.39	49 %	1.46	46 %	
7	1.81	28 %	1.81	25 %	
8	1.84	28 %	1.81	28 %	71 %
9	1.80	28 %	1.80	28 %	71 %
10	1.79	29 %	1.87	27 %	69 %
11	1.76	31 %	1.81	27 %	70 %

Table 2: Achieved losses and accuracies on the training and test data for the different initializations stated in table 1.

to make a big difference for the accuracy, and to ensure that the network can learn on these images. The exact settings and results of these runs can also be found in tables 1 and 2. It becomes clear from table 2 that the achieved accuracies in these runs is significantly worse than before: below 50% and 30% respectively. Note that these are the values after only 4 epochs, when the validation loss is minimal; if we train the network longer, it starts to over-fit the data.

This decrease in accuracy intuitively makes sense, because it might be easier for the network to predict an f_m of 0 for non-mergers, than to predict exactly the right value for merging galaxies. This is also visible in the vectors that the network gives as output: the probabilities calculated by the softmax activation function are often similar in value in neighboring classes. For example, when the input image has a true f_m of 0.6, the probabilities at indices 5, 6 and 7 are generally very similar. This should be taken into account, but is not. In the next subsections we will optimize the network for images of mergers and see if we can improve its performance.

3.3 Optimization of the network for mergers

In the previous subsection we found that changing various parameters of the network did not increase the performance of the neural network on the complete data set. Furthermore, we found that if we leave out the non-merging galaxies the accuracy reduces from 60 % to 25 %. It seems that the network is not able to learn well on merging galaxies. This might be due to an ill-defined accuracy. Therefore we will optimize the network on merging galaxies and investigate the extent to which it can classify these galaxies.

3.3.1 Introducing a new accuracy measure

The accuracies stated in table 2 are calculated as follows: when the predicted value $\hat{y}$ is equal to the label y, the network's prediction is correct and otherwise it is wrong. This is called categorical accuracy (see Keras-team (2016)) and is the default for networks with multiple class labels. However, for our purpose it might be considered sufficient when the network predicts an f_m that is only one class off. For example, when the label is 0.6, but the network predicts 0.7, this is still a fairly good prediction. Especially when considering that the

original values of f_m are not based on strict guidelines, but on an average of classifications by eye. Hence, we run the code again, but now with an additional calculated accuracy that defines a correct output as:

$$\hat{y} \in \{y - 0.1, y, y + 0.1\} \tag{1}$$

Hereinafter, we will call this accuracy the *plus-minus accuracy*.

If we now run the code again with the same settings as in run 7 (except that we change the number of epochs to 4 and reduce the number of nodes again to 32, 64, 128), we immediately see a substantial difference in both accuracies on the test data: 71.3 % plus-minus accuracy as opposed to 27.9 % categorical accuracy. In other words, for 71.3 % of the test images, the network is able to reproduce the correct label of f_m within a one class difference. With this new accuracy that we can use to evaluate the performance of the convolutional neural network, we can try again to optimize the model on the merger data.

3.3.2 Adapting the architecture

In the original keras code, the model has two convolutional layers and one fully connected layer before the output layer. This is a common structure for a convolutional neural network, but in many cases it is also conventional to include three convolutional layers and two fully connected ones (Karpathy, 2018). Each convolutional layer should then be followed by max pooling in order to reduce the number of trainable parameters in the network. We implement this new architecture and add the new layers to the keras model. Karpathy (2018) also explain that it is in general better to use multiple convolutional layers with a relatively small filter size, than one layer with a larger filter size. In this way, there are less trainable parameters and the stack of layers contain non-linearities that make the features in the filters more expressive. The downside to this method is that more memory is needed. We conclude from this, that we should stick to the 3×3 filter size in all the convolutional layers. This new neural network now turns out to reach the minimal validation loss after 5 epochs. This results in a categorical accuracy of 27.9% and a plus-minus accuracy of 71.3%. Again, we can conclude that this deeper network also did not affect the performance at all regarding the accuracy or loss.

3.3.3 Adapting the ratio of test to training data

Overall, none of the modifications to the network resulted in a higher accuracy on the validation data. Therefore, we will change the ratio of train and test data. Hitherto, we trained the network on two thirds of all the images and validated on the remaining third. Now, we will train it on approximately 80% of the complete data set (2403 galaxy pairs, or 4806 images), in order to see if this improves the performance of the network. With this change, 4 epochs of learning is enough for the network to reach the minimal loss on the test data, which is then 1.87. The categorical accuracy after 4 epochs is 26.7% and the plus-minus accuracy is 68.8%. So, dedicating more of our data to the training set results in a slightly lower achieved accuracy on the test set. Even though the network was provided more images to train on, it couldn't learn better features to classify the test images with a high accuracy. The only reason that we can think of to explain this result, is that the quality of the learning does not depend on the amount of images in the training set, but on the individual images that are fed to the network during training. As illustrated in Fig. 2, some images contain galaxies with a clear structure and a large size, whereas others are vague blobs in the background. The first type of image provides a more pronounced structure for the network to learn features, but is also easier to classify correctly in the validation set. Therefore, the accuracy on the training and test sets probably depends heavily on the specific images that both sets hold. And because we only have 3003 pairs of images of mergers in total and 11 classes, the division between training and test data influences the outcome significantly.

3.3.4 Adapting the dropout

In the previous experiments we noticed that the validation loss reached a minimum after a short amount of epochs, after which the network started over-fitting. The dropout in the model is a technique to prevent this, by disabling connections in the network (Srivastava et al., 2014). We will adapt the dropout percentage of the network in run 11, to see whether we can increase its performance. Effectively, it does not change the outcome, the losses and accuracies on the train and test data are still very similar to previous runs. The only difference is that these values are reached after 8 epochs of learning instead of 4 or 5. If the network continues after 8 epochs, it starts over-fitting the training data again.

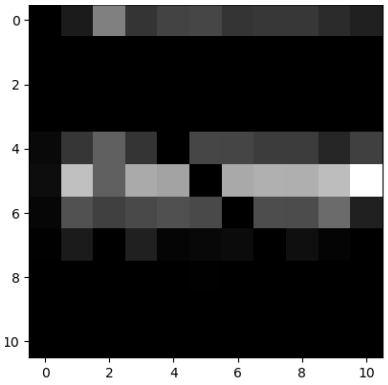
3.4 Application for predicting f_m

From the previous optimizations, we find that for both a data set including non-mergers and a data set excluding non-mergers, the initialization stated in Tab. 3 is the best. We apply the neural network to both the complete

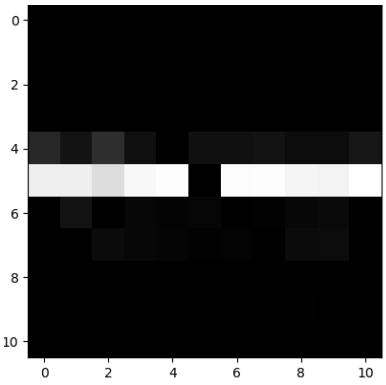
Parameter	Initialization
Batch size	128
Nodes (N_1, N_2, N_3)	32, 64, 128
Filter size	3×3
Dropout (D_1, D_2)	25%, 50%
Optimizer	Adam
Ratio train-test	2:1

Table 3: Best found initializations of the network, as analyzed in sections 3.2 and 3.3.

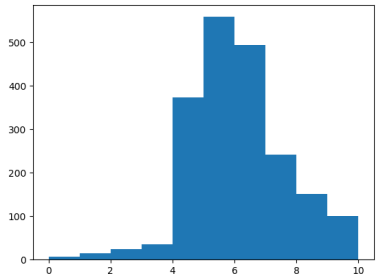
data set and the data set consisting only of mergers and evaluate the results using both the categorical and plus-minus accuracy. The results of these initializations can be found in tables 1 and 2, run 4 and 8 respectively. As we mentioned previously, the accuracy of both runs is quite low. To investigate which images the network is able to classify, we generate a misclassification matrix with elements $\{c_{ij}\}$, which tells us how often a galaxy with label j (column) is misclassified as i (row), normalized by the number of images in that class. The misclassification matrices for the complete sample and for the merger sample can be found in figures 3a and 3b. The color range in the misclassification matrices ranges from black -no misclassifications- to white -a lot of misclassifications. In both figures we observe a black diagonal, as galaxies with label i can not be misclassified as label i. In figure 3a we observe that some galaxies are misclassified with label 0, thus falsely classified as non-merging galaxies. This can happen when a galaxy resembles a non-merging galaxy. We then observe a black area in rows 1 to 3. This can be explained by the fact that there are little to no galaxies labeled with 0.1 to 0.3 in the first place. Darg et al. (2010a) made a selection based on the weighted-merger-vote-fraction f_m and only included galaxy pairs for which one of the two sufficed $0.4 \leq f_m \leq 1.$. Furthermore we observe that galaxies with label 1 are most often misclassified (completely certain that an image is a merger). This is in line with expectations as there are not many class 1 labels and the network has more trouble learning features of class 1. When we look at figure 3b, we can make the same observations for the merger sample. We observe that most galaxies are misclassified as label 0.5. We think that this caused by the fact that most galaxies in the sample have the label 0.5 and the network only recognizes features of this class. The distribution of labels in the merger sample can be observed in figure 3c.



(a) Misclassification matrix for the complete data set.



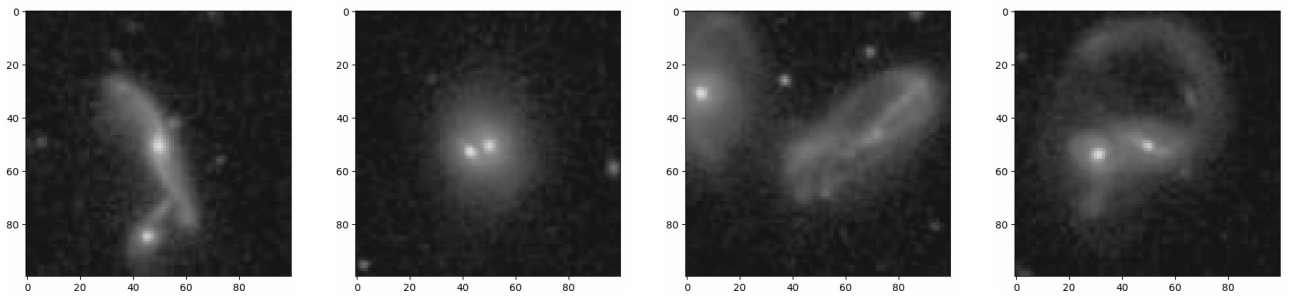
(b) Misclassification matrix for the merger sample.



(c) Distribution of the values of f_m for the merger sample. Clearly, most mergers are labeled with a value of 5.

Figure 3: The confusion matrices of misclassified images of the complete data set and of the data set containing only mergers. The squares range from white to black, where a white square visualizes a high fraction of misclassified images and a black square visualizes no misclassified images. The values in the matrices are normalized with the distribution of f_m values of the corresponding data set. Note: classes are multiplied by ten on the axes, so $f_m = 0.6$ is 6 in the matrix. Figure 3c shows the distribution of the class labels f_m for the merger sample, to help understand figures 3a and 3b.

Fig. 4 shows two examples of images that are classified correctly by the network, as well as two images that are misclassified. In the two correctly classified images there are some clear features that the network could identify in mergers: in image **(a)**, there is a non-spherical, chaotic structure and in image **(b)**, the two bright centers of the galaxies are very close together and surrounded by a roughly spherical and homogeneous halo (without this halo, it could be just two stars instead of galaxies). As for the first misclassified image, **(c)**, it is understandable that the network does not classify this with a high f_m. Neither of the two features (Chaotic, non-spherical structure or two bright centers very close together with a halo) is present here. However, in the second misclassified image, **(d)**, there is an evident, chaotic structure, that is characteristic for mergers, as well as two bright centers. It is not clear why the network is not able to predict a more realistic value of f_m for this



(a) Correctly classified as $f_m = 0.7$.

(b) Correctly classified as $f_m = 0.6$.

(c) Misclassified as $f_m = 0.4$, true value $f_m = 0.8$.

(d) Misclassified as $f_m = 0.5$, true value $f_m = 0.9$.

Figure 4: Images **(a)** and **(b)** are examples of images of the test set that were classified correctly with the network initialization as provided in Tab. 3. Images **(c)** and **(d)** are examples of misclassified images. The network was applied on the data set consisting of only merging galaxies.

image. In the sample of misclassified mergers, there are multiple such examples where we expected the network to recognize the merger features, but still predicted a low value of f_m.

3.5 Application for predicting binary classification

In the previous section we researched whether a neural network could predict a the probability that a galaxy is part of a merger. The accuracy was not as high as we hoped and therefore we decided to test the performance of a neural network using binary class labels. The labels we use for the galaxies are merger (1) and non-merger (0). We use the same network as in the previous section, with two instead of eleven output nodes. Furthermore, we use the initializations from table 3. We apply the network to the complete data set.

We find that the network achieves a much higher accuracy now. The resulting losses and accuracies can be found in table 4. The training accuracy reaches 96 % which is very high considering previous results and the accuracy on the test set reaches 91 %. We generate a small confusion matrix, only now we do not leave out the correctly classified images. The confusion matrix can be found in figure 5. The high accuracy can also be observed in the confusion matrix. As the confusion matrix is normalized over the number of images in that class, we observe that non-mergers are less often correctly classified than mergers. This could be caused by the fact that non-mergers have more influence of background stars, or that non-mergers do not have specific recognizable features, whereas mergers do.

Fig. 6 shows two examples of images that are correctly classified by the network, as well as two images that are misclassified. The correctly classified images consist of a merger and a non-merger and the misclassified images consist of a merger classified as a non-merger and a non-merger classified as a merger. In subfigures **(a)** and **(b)** it is relatively easy to recognize a non-merger and merger respectively. In **(c)**, it is understandable that

Run	Training loss	Training accuracy	Validation loss	Validation accuracy
12	0.12	96 %	0.24	91%

Table 4: Results of applying the neural network to our galaxy images using binary class labels. As one can observe, the accuracy is now much higher.

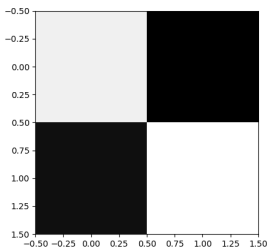
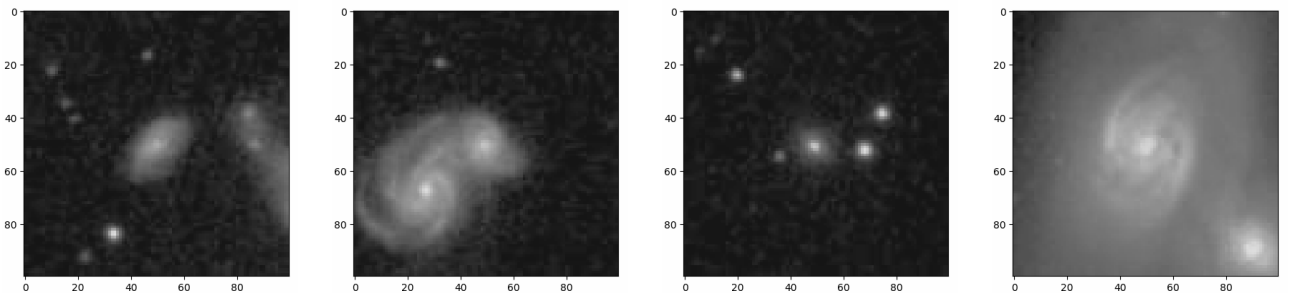


Figure 5: Confusion matrix for applying the neural network to our galaxy images using binary class labels. The color ranges from black (no classifications) to white (a lot of classifications). We observe that many images are correctly classified.



(a) Correctly classified as a non-merger.

(b) Correctly classified as a merger.

(c) Misclassified as a merger.

(d) Misclassified as a non-merger.

Figure 6: Images **(a)** and **(b)** are examples of images of the test set that were classified correctly and **(c)** and **(d)** are examples of misclassified images. These results are from the binary classification.

the network misinterprets the image as a merger, because there are background stars right next to the galaxy. This looks similar to the feature of two bright galaxy centers close to each other. In subfigure **(d)** from Fig. 6 too, it is difficult for the network to distinguish the two merging galaxies, because they are further away from each other than in most images and the central galaxy extends out of the image. This is also a clear example of an image that the human eye can classify without any problem, whereas the neural network falters in doing so.

4 Conclusion

We applied a convolutional neural network to a catalog of merging galaxies from the Galaxy Zoo project. We added images of non-merging galaxies to this catalog to create our complete data set. This complete set contains a more realistic representation of the fraction of mergers on the sky. All of these images are labeled with the weighted-merger-vote-fraction, f_m. This parameter is 0 for non-mergers and higher than 0.4 for mergers. The maximum value is 1, indicating a very high certainty that a galaxy is indeed part of a merger.

In section 3.2, we optimized the neural network by modifying several parameters, such as the amount of nodes in hidden layers and the batch size. This did not impact the accuracy of the predictions of the validation data significantly, but we did observe a drop in accuracy when we change the ratio of mergers to non-mergers in the data. Therefore, we introduced a new and more relevant accuracy measure to assess the performance of the network in section 3.3 and we optimized the network on just the merger data instead of the complete data set. Adding layers to the network did not improve the performance and neither did adapting the ratio of training to test data. Finally, in section 3.4, we evaluate how well the network performs on the complete data set and on the sample containing only mergers, using the best initialization found for the network (see Tab. 3). For the complete data set we find a 66% accuracy on the test set and for the merger sample we find a 28% accuracy. When we introduce a plus-minus accuracy, we find a 71% accuracy on the merger sample. This accuracy is acceptable, but not very high. We show examples of correctly and wrongly classified images from the test set containing only mergers in figures 4, but cannot conclusively explain why some images are not classified well.

Because we suspect that the network has difficulties predicting the right f_m due to a shortage of data, we decide to change the objective to classifying in only two classes. We now adapt the model to predict only 0 (non-merger) or 1 (merger). The resulting accuracies are now much higher, with an accuracy of 91 % on the test set.

5 Discussion

From experimenting with different initializations of the neural network, it became clear that the relatively low achieved accuracies are likely caused by limitations of the used data set and not by the architecture of the network. First of all, there are only 3003 unique images in the sample of mergers. Even though we could use 6006 images, because the two images of one merger-pair are slightly shifted, this is not a lot of data. Especially considering that there are 11 classes in the model. Moreover, not all of these images provide equally clear features. As illustrated in Fig. 2, some of the merging galaxies span a smaller area or exhibit less clear structure altogether. There can also be background stars in the images. This means that effectively, the network has a limited number of images that it can use to distract features.

Another issue with the data lies in the labels. First of all, we manually labeled all the non-mergers with 0, even though the could have any value below 0.4. We could not retrieve the original values from the Galaxy Zoo

project, so this choice was inevitable. Besides, the actual distribution of values of f_m was continuous, but we binned them into 11 classes to adapt them to the structure of the convolutional neural network. Therefore, the class labels of the data are not ideal and this might be a cause for concern.

In our data set, we included 6006 mergers and 10000 non-mergers. However, in reality the fraction of mergers is even way smaller. This fraction depends in a complex way on multiple astrophysical parameters (Rodriguez-Gomez et al., 2015), but estimates are that between 5 to 25 percent of the galaxies are ongoing mergers (Garner, 2011). Therefore, it might prove significantly more difficult for a neural network to identify merging galaxies in a data set consisting of all galaxies found in a patch of sky.

Altogether, we can conclude that a lot more data of mergers is necessary in order to learn a convolutional neural network to predict the probability that an image of a galaxy is a merger. With future observations, this might be realized and it would be interesting to implement this network again.

6 Acknowledgements

Funding for SDSS-III has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, and the U.S. Department of Energy Office of Science. The SDSS-III web site is http://www.sdss3.org/.

SDSS-III is managed by the Astrophysical Research Consortium for the Participating Institutions of the SDSS-III Collaboration including the University of Arizona, the Brazilian Participation Group, Brookhaven National Laboratory, Carnegie Mellon University, University of Florida, the French Participation Group, the German Participation Group, Harvard University, the Instituto de Astrofisica de Canarias, the Michigan State/Notre Dame/JINA Participation Group, Johns Hopkins University, Lawrence Berkeley National Laboratory, Max Planck Institute for Astrophysics, Max Planck Institute for Extraterrestrial Physics, New Mexico State University, New York University, Ohio State University, Pennsylvania State University, University of Portsmouth, Princeton University, the Spanish Participation Group, University of Tokyo, University of Utah, Vanderbilt University, University of Virginia, University of Washington, and Yale University.

Bibliography

D. W. Darg et al. Galaxy zoo: the fraction of merging galaxies in the sdss and their morphologies. *Monthly Notices of the Royal Astronomical Society*, 401:1043–1056, 2010a.

D. W. Darg et al. Galaxy zoo: the properties of merging galaxies in the nearby universe – local environments, colours, masses, star formation rates and agn activity. *Monthly Notices of the Royal Astronomical Society*, 401:1552–1563, 2010b.

T. Devries and G.W. Taylor. Improved regularization of convolutional neural networks with cutout. *CoRR*, abs/1708.04552, 2017. URL http://arxiv.org/abs/1708.04552.

R. Garner. Astronomers pin down galaxy collision rate. https://www.nasa.gov/mission_pages/hubble/science/collision-rate.html, note = Accessed: 2018-05-16, 2011.

J. Jenkinson, A. M. Grigoryan, and S.S. Again. Enhancement of galaxy images for improved classification. *Conference paper in Proceedings of SPIE*, 2015.

A. Karpathy. Notes of a stanford cs class: Module 2 - convolutional neural networks (cnns / convnets). http://cs231n.github.io/convolutional-networks/, 2018. Accessed: 2018-05-13.

Keras-team. keras/metrics.py. https://github.com/keras-team/keras/blob/c2e36f369b411ad1d0a40ac096fe35f73b9dffd3/keras/metrics.py, 2016. Accessed: 2018-05-14.

Keras-team. keras/examples/. https://github.com/keras-team/keras/tree/master/examples, 2018. Accessed: 2018-05-07.

N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P.T.K. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *CoRR*, 2017.

D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc. URL http://dl.acm.org/citation.cfm?id=2999134.2999257.

C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson, K. Masters, R.C. Nichol, M.J. Raddick, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg. Galaxy Zoo 1: data release of morphological classifications for nearly 900 000 galaxies. *Monthly Notices of the Royal Astronomical Society*, 410:166–178, January 2011. doi: 10.1111/j.1365-2966.2010.17432.x.

W. O'Mullane. Sdss jhu image cutout service. http://skyservice.pha.jhu.edu/develop/vo/imgcutoutclient.aspx, 2004. Accessed: 2018-05-13.

V. Rodriguez-Gomez et al. The merger rate of galaxies in the illustris simulation: a comparison with observations and semi-empirical models. *Arxiv*, 2015.

S.T. Sohn, J. Anderson, and R.P. van der Marel. The M31 Velocity Vector. I. Hubble Space Telescope Proper-motion Measurements. *The Astrophysical Journal*, 753:7, July 2012. doi: 10.1088/0004-637X/753/1/7.

N Srivastava, G Hinton, A Krizhevsky, I Sutskever, and R Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

D.G. York et al. The Sloan Digital Sky Survey: Technical Summary. *The Astrophysical Journal*, 120:1579–1587, September 2000. doi: 10.1086/301513.