Deep learning classification of solar images using MXNet

Universiteit van Amsterdam

Centrum wiskunde & informatica

Author: Martijn Dortmond
10740406

Supervisor:
Enrico Camporeale
second assessor:
Anna Watts

Report Bachelor Project Physics and Astronomy
size 15 EC
conducted between 04-04–2017 and 04-07-2017

1 Summary

1.1 Scientific summary

A strong solar wind has a major impact on the earth because it can interfere with radio communications, damage satellites and in the worst case can cause power blackouts. A solar wind is a plasma of protons, electrons and other charged particles which are released by solar flares near sunspots. A strong solar wind can be predicted by IMP-8 and ACE up to one hour before it arrives. We wanted to create an earlier warning system for large solar winds. To achieve this goal, solar images are classified into two classes. The first class contains large solar winds while the other class contains normal solar wind activity. [Kamide, 1992] showed that the strength of a solar wind is mostly based on the interplanetary magnetic field of the plasma. Therefore the classification is done based on the interplanetary magnetic field of the solar wind plasma. Whenever the plasma has a field vector of -5 the solar wind is considered as large. The strength of the field is obtained from the OMNI2 dataset. Magnetograms of the sun are used for the classification because sunspots are clearly visible from them. These magnetograms originate from SOHO and SDO. A convolutional neuron network is trained to classify the images. An 80% validation accuracy has been achieved in the case where a magnetogram as well as a extreme ultraviolet image has been used. However the validation accuracy is greatly influenced by the proportion of the test and validation set. Therefore the achieved accuracy cannot be immediately accepted. Further research with a larger dataset is needed to give a clear conclusion.

1.2 Dutch summary

Een sterke zonnewind kan grote invloed hebben doordat het radio comminucatie aantast, satelieten beschadigd of in het ergste geval een grote stromingstoring tot gevolg heeft. Een zonnewind is een plasma van elektronen, protonen en andere geladen deeltjes. De plasma kan de zon verlaten rondom de zonnevlekken. De plasma kan, indien het een inwendig zuidwaards magnetisch veld heeft, grote verstoringen aanbrengen in het magnetisch veld van de aarde. Een zonnewind kan momenteel uur van te voren voorspeld worden met behulp van sattalieten. Wij hopen echter een zonnewind eerder te kunnen voorspellen door deze te classificeren met behulp van een neuraal netwerk. Deze classificatie wordt gedaan met behulp van magnetisatiediagrammem van de zon.

Het network Het netwerk wordt getraind met ongeveer 2000 fotos van de zon. Hiervan wordt 70% gebruikt voor de test set en 30% voor de controleset. In een neuraal network wordt de test set gebruikt om het netwerk te trainen terwijl de controleset wordt gebruikt om te kijken hoe goed het netwerk preseteerd op data die nog nooit gezien is door het netwerk. De grootste nauwkeurigheid is behaald met een testset waarbij een combinatie van zowel een plaatje van licht met een golflengte van 171 angstrom als een magnetisatiediagram gebruikt is. De nauwekeurigheid van controleset bij deze dataset is 80% terwijl de nauwekeruigheid van de test set bijna 100% bedraagd. Na verder onderzoek blijkt dat de training van het netwerk te veel afhangt van de proporties tussen de test en validation set. Dit komt voornamelijk doordat de testset bijna perfect gefit is. Dit kan voorkomen worden door meer plaatjes te gebruiken, echter zijn deze niet beschikbaar in de databases doordat er maar weinig data beschikbaar is van grote zonnewinden. Hiernaast is gebleken dat de karakterstieke eigenschappen waarop het netwerk de plaatjes moet classifiseren niet altijd aanwezig zijn of juist aanwezig zijn wanneer deze niet verwacht worden. Hierdoor kan de behaalde nauwkeurigheid niet met zekerheid worden aangenomen.

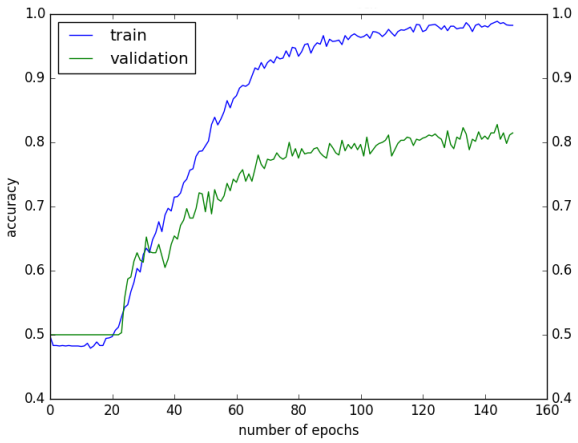


Figure 1: De nauwkeurigheid van een neuraal netwerk bij gebruik van plaatjes met een golflengte van 171 angstrom en een magnetisatiediagram

Contents

1 Summary **1**
1.1 Scientific summary . . . 1
1.2 Dutch summary . . . 2

2 Introduction **4**
2.1 Motivation . . . 4
2.2 Scientific questions . . . 4
2.3 Theory . . . 4
2.3.1 Solar wind origin . . . 4
2.3.2 Solar wind interaction . . . 5
2.4 Approach . . . 5
2.4.1 The classes . . . 5
2.4.2 The Network . . . 6
2.4.3 Databases . . . 6
2.4.4 Collecting images . . . 7

3 Results **10**

4 Discussion **13**

5 Conclusion **15**

6 Appendices **16**
6.1 Convolution Neural network . . . 16

2 Introduction

2.1 Motivation

A sufficiently large solar wind can cause a geomagnetic storm which is visible as a temporary disturbance of the Earths magnetosphere. This disturbance can have large impact on society because it could damage satellites, interfere with radio communication and in the worst case can cause power blackouts. Solar wind properties can be predicted by satellites such as IMP-8 and ACE, which are located at the Lagrangian L1 point, 1,500,000 km from Earth. However the solar wind is measured when it is already close to Earth and a warning can only be made up to one hour before its arrival. This could pose a problem to power stations that cannot be shut down immediately. An earlier prediction of solar winds could possible be made by detecting a solar wind from images of the outer layer of the Sun.

2.2 Scientific questions

It would be a tedious task for humans to check the solar images on characteristic features every hour of the day. Therefore we would like to invent a system that is able to classify a solar wind from images of the Sun surface. In this thesis we would like to find out whether it possible to derive the existence of a solar wind from pictures of the sun by using the classification. For this purpose a neural network will be used. The network should classify the solar wind with a sufficiently high accuracy. To create the network we will use the neural network framework MXnet.

2.3 Theory

2.3.1 Solar wind origin

A solar wind consists of a plasma of electrons, protons and a small fraction of other charged particles with speeds of 300-400 km/s. The properties of the solar wind like speed, magnetization, pressure and temperature change over time. Therefore fast solar winds can reach speeds of 500-800 km/s [Cranmer, 2001]. Within the plasma there exist a interplanetary magnetic field which originates from the surface of the sun. This field plays an important role in the impact of the solar wind. The plasma can be released by either coronal holes or sunspots. Sunspots are regions of lower temperature on the sun's surface [Harvey and Sheely, 1979]. Sunspots appear near the equator of the sun and do not surpass latitudes greater than 70 degrees. The lifetime of the sunspots is a few days to a few months. Sunspots only appear in pairs with opposite polarities. The magnetic field near the sunspots is not bound to remain inside the sun but arches away from one sunspot and connects with a sunspot with an opposite polarity by forming a corona loop. It is suggested by [Tsuneta et al., 1992] that magnetic reconnection within these loops generate solar flares. Therefore solar flares originate mostly from sunspots. Whenever magnetic reconnection occurs, the field lines within a plasma break and merge with other field lines to create a new magnetic configuration. During this reconnection some of the energy stored in the magnetic field is converted to heat and kinetic energy. The sudden release of energy from the magnetic reconnection generates enough energy to accelerate the particles to escape velocity.
Coronal holes are regions in the sun surface with a low density. Corona holes appear mostly near the poles of the sun but occasionally also exist near the equator. The

corona holes near the equator only exist for a few solar rotations. These holes have a open magnetic field that leads into space [Cranmer, 2001]. Charged particles will flow along these magnetic field lines and reach speeds up to 800 km/s.

2.3.2 Solar wind interaction

Whenever the solar wind reaches the magnetosphere, most of the particles are deflected by Lorentz force caused by the Earths magnetic field. These particles will therefore travel around the earth and are unable to penetrate into the magnetosphere. However some particles are able to partly travel into the magnetosphere and transfer their energy to the magnetosphere by magnetic reconnection between the interplanetary magnetic field and the Earth magnetic field[Dungey, 1961]. The released energy causes an increase in energy and particles in the ring current in the magnetosphere. The current in the ring is induced from electron drifting eastwards and protons drifting westwards which induces a total westward current. This current induces a magnetic field which changes the earth magnetic field. [Kamide, 1992] showed that the reconnection of the fields is the most efficient for plasma with a southward magnetic field due to the Earth magnetic field pointing northward. Therefore magnetic storms are the most likely to develop whenever the plasma contains a southward interplanetary magnetic field.

2.4 Approach

2.4.1 The classes

As described by [King, 1986] typical quiet solar winds have a magnetic field vector with a size less than 5nT. Any solar wind with a magnetic field vector with a size larger than 5nT are large solar winds. It also known from [Tsurutani et al., 1992] that the magnetic field vector and not the velocity of the particles plays a dominant role in the influence of the solar wind on the magnetic ring current. Therefore the boundary of the Bz value for the classification on the data is set at 5nT without taking the solar wind speed into account. The data is classified into two classes. The first class contains images with a Bz value lower than -5. This class is characterized by a higher than normal solar activity and therefore have a large impact on the Earths magnetic field. The second class contains images with a Bz value higher than -5 but lower than 0. These images are associated with a normal solar activity and contain no solar storms. The images with a Bz value higher than 0 are not used because negative and positive area's of magnetization on the sun only occur in pairs. There are not big distinct features between images with large negative and positive values. The program will never be able to detect the difference between a large positive and a large negative Bz value based on the characteristic features in the image. Therefore we decided to only look at the negative values. This boundary for the classification leads to 1016 images in class 1 as well as 1016 images in class 2. The images are taken between 1996 and 2016. The images from 1996 until 2011 originate from the Solar & Heliospheric Observatory and the images from 2011 until 2016 originate from the Solar Dynamics Observatory.

2.4.2 The Network

The network will import all the images from class 1 and class 2 with their appropriate labels. The network will try to find the best weights for the classification by training on the given dataset. The training will be done by using stochastic gradient descent. The appendix will discuss neural networks in more detail. The structure of the network consists of one convolution layer with 10 convolution filters with a 5x5 kernel. A Relu activation function is used for the convolution layer. The output of the convolution is fed into a maxpool layer with a 2x2 kernel and a 2x2 stride. This serves as the input for hidden layer with 30 neurons and a dropout of 80%. The dropout layer is added to reduce over fitting on the test set. A Relu activation layer is used for this hidden layer. The output of the hidden layer is then fed into a softmax layer which classifies the 2 classes.

2.4.3 Databases

Omni
The omni data set contains a hourly mean value of solar wind plasma parameters [King and Papitashvili, 2005]. The data from the Omni database is measured by several satellites in the first Lagrange (L1) point of the sun about 1,500,000 km from the earth. The data between different the satellites is cross compared and validated. The interesting data from omni for this project are the Bz value measured in Geocentric Solar Magnetospheric (GSM) coordinate system and the flowspeed of the particles. In the GSM coordinate system the X axis points towards the Sun while the Z axis is parallel to the earth's magnetic dipole axis which is perpendicular to the X axis. The effects of a solar wind on the magnetic field in the magnetosphere and ionosphere can be easily studied by using this coordinate system.

The Solar & Heliospheric Observatory database
The Solar & Heliospheric Observatory (SOHO) was a satellite that studied the Sun from its deep core to the outer corona and the solar wind. [soh, 1997]. SOHO has measured the magnetic properties of the sun since December 2, 1995 until the end of 2010. SOHO is located at the L1 point of the earth due to the constant visibility of the sun. In this way the properties of the sun can be measured at any time. The SOHO database consist out of gray scale and colored magnetogram, large angle Spectrometric Coronagraphs and extreme ultraviolet images. To study the solar wind properties, the magnetograms will be used. The magnetograms are chosen because they will most likely show the most clearest properties of a sunspot.

A magnetogram of the sun is taken every 15 minutes. The magnetograms are available in different resolutions. In this research a resolution of 512 by 512 will be used. This resolution shows enough detail of the possible solar spot but is not too large for the neural network.

Solar Dynamics Observatory database

The Solar Dynamics Observatory(SDO) was launched on February 11, 2010 and is the successor of SOHO. SDO resides in an inclined geosynchronous orbit and is able to observe the sun almost continuously. SDO observation will be disrupted a small amount of time due to eclipses. Due to a higher uplink stream to earth, SDO is able to make images of the sun more frequently and with a higher resolution. The SDO database consists of gray scale and colored magnetograms and extreme ultraviolet images. The magnetograms of SDO will be used for the same reason as described earlier.

2.4.4 Collecting images

Before any image can be downloaded from the database, the program has to calculate the right time from which the solar wind left the sun surface. Therefore the program start off by reading the south component of the magnetic vector (Bz) in Geocentric solar magnetospheric (GSM) coordinates and the flowspeed of the particles in km/s from the omnidata. The data from omni is measured with a timespan of 1 hours between consecutive measurements. The timestamp of the data is written in the format year/day/hour, with the days between 0 and 365. The time object that will be used, requires the input format year/month/day/hour instead. Therefore a function will map all the days of the year to a month with the appropriate leftover days. The function does take into account the leap years.
Thereafter the image will be classified based on the Bz data from omni. Class1 is defined as Bz $<$ -5, while class2 is defined by -5 $<$ Bz $<$ 0

The Bz and the flowspeed are both averaged over an interval of 10 consecutive measurements. The flowspeed will therefore be more smooth for consecutive measurements rather than spiky which leads to a better approximation for time of the image that is used for the classification. Bz values that are smaller than -5 for one measurement are not classified as class 1 by taking the average of the Bz value. Therefore only large storms with multiple consecutive large negative measurements of Bz end up in class 1 and any single measurement that is classified as class 1 get removed because these are not large solar storms.

Calculating the solar wind release time
The time at which the solar wind particles were released from the sun is calculated based on their flowspeed measured by omni. It is approximated that the particles maintain their speed on the way from the Sun to the Earth. The time at which the plasma was released is given by subtracting the travel time of the particles from the sun to the L1 point from the timestamp at which OMNI measured its data. The Earths ecliptic movement has to be taken into account when calculating the the distance from the Sun to the L1 point. The distance of the Sun to Earth can be calculated by

$$r = a\frac{1-e^2}{1+ecos(\theta)} \tag{1}$$

with a the semi-major axis, e the eccentricity of the Earths orbit and θ the true anomaly.

Since the eccentricity of the earth is really small the approximation

$$\frac{1}{1+x} = 1 - x \tag{2}$$

can be used to rewrite equation 1 as

$$r = a(1 - e^2)(1 - ecos(\theta)) \tag{3}$$

Because the eccentricity of Earths orbit is low ,the difference between the mean and true anomaly is really small and thus the true anomaly gets approximated by the mean anomaly. The mean anomaly

$$\theta = daynumber \cdot \frac{360}{365.256} = daynumber \cdot 0.9856 \tag{4}$$

is given by the total degrees of a circle divided by 365.256 which is the number of days in a sidereal year. The mean anomaly should be zero whenever the earth reaches perihelion. This happens around the 4th of January. So there should be 4 days subtracted from the original day. Combining equation 3 and 4 gives

$$r = a(1 - e^2)(1 - cos((daynumber - 4) \cdot 0.9856)) \tag{5}$$

Thereafter the distance from the Earth to the L1 point is calculated by using the formula for the hill sphere of the Earth.

$$earth\_L1 = a\sqrt[3]{\frac{m_{earth}}{3m_{sun}}} \tag{6}$$

The distance from the Sun to the L1 points is the difference between 5 and 6. Based on the date it will calculate the time when the particles left the Sun.

Lookup in the database

The program will take the time at which the particles left the Sun and it will look in the SOHO database for the closest match that is available for that particular day. It could be that two calculated timestamps will map on the same image in the database. In that case the program will only download one image and select the lowest class as the label for the image.

It could be that Bz measured by omni is lower than -5 for only hour. In this case there is probably no storm going on because a storm last will likely last longer than 1 hour. The program will remove any image from class 1 if there are not two consecutive images in this class. This will remove any remaining random dips in Bz from the data.

Whenever there is only one hour between two magnetograms the images are almost identical. Therefore the program selects the images that are at least 4 hours apart from each other. Hereby the program prevents that nearly similar images end up in the database. These images do not increase the generalization of the network and thus do not contribute positively to the classification of the images.

The program will now download all the images in class 1. The images in class 2 are randomly selected from the pool of images because there are more images in class 2 than class 1. If the images in class 2 are downloaded in order, the images will only originate from the early years in the database of SOHO and will not be spread out over all years. This causes less spread in the shape of the images in class 2.

Enhancing the data

The images from SOHO and SDO contain a lot of black padding around the sun that is of no use for classification. We would like to get rid of these parts because the computation time is lower for a smaller input vector. Most of the characteristic properties of a solar wind reside in the region around the equator of the sun and not at the polar regions. To reduce the unnecessary pixel only the middle region of the sun is selected as shown in figure 2

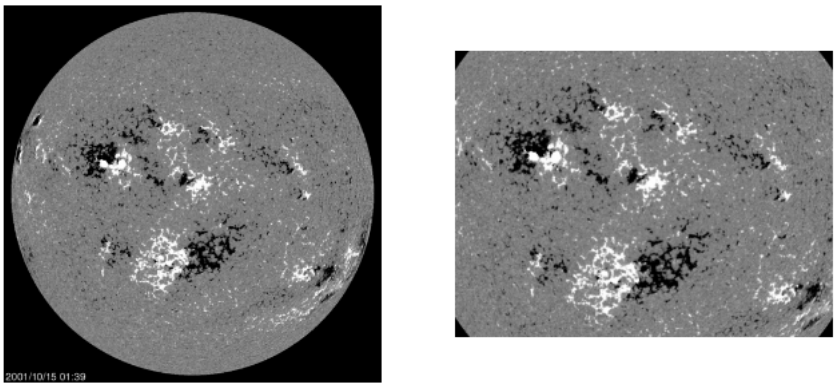


Figure 2: The original magnetogram on the left and the resized image with the region of interest outlined on the right

The dataset is quite small due to the lack of solar storms.Therefore the data needs to be enhanced in such a way that more images with the same characteristics are generated. Normally this is done by rotating the images. The key features are kept while the image appears totally different for a normal neural network. A convolution neural network on the other hand should learn the characteristic features even if they are rotated by some angle. A problem arises when the image is rotated because the slicing described earlier would remove the key characteristic from the pictures as they are now possibly located at the top side of the picture. Instead the images will be horizontally flipped as show in figure 3. Therefore the characteristic features still exist but the image is different from the original picture.

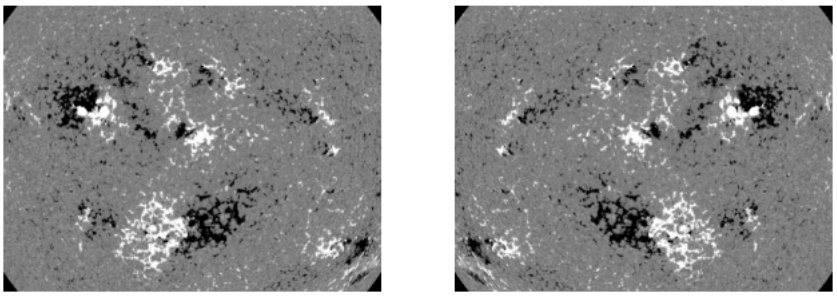


Figure 3: The original image on the left and the vertical flipped image on the right

This doesn't enhance the dataset by a large amount but still double the amount of images available. To further increase the data points available for the neural network, two pictures are stacked on top of each other. On top of the magnetogram an extreme ultra violet with a wavelength of 171 angstrom is stacked. The extreme ultra violet, which is shown in figure 4, contains 3 data channels for red, blue and green. The shape of the ultraviolet image is a 512x512x3 matrix which is stacked on top of the grayscale image with dimension of 512x512x1. This creates a input matrix of 512x512x4 which contains 4 times as much data as the original input.

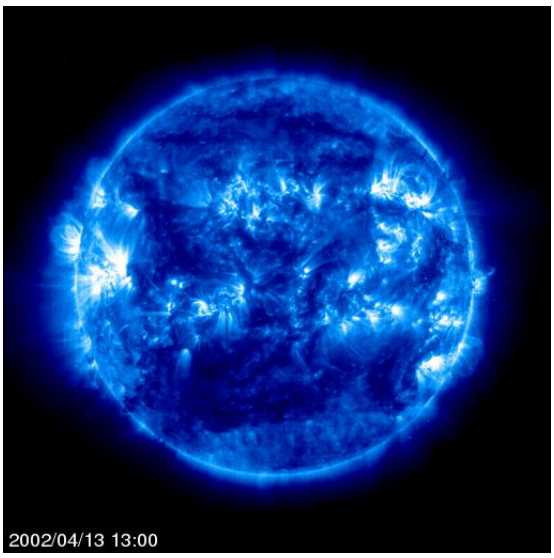
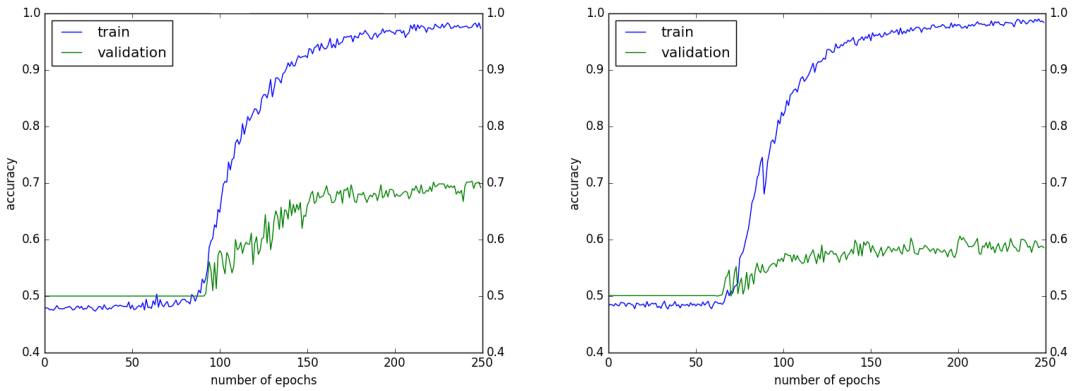


Figure 4: An extreme ultra violet image with a wavelength of 171 angstrom

3 Results

All the results are generated by using the network described in the introduction. The training for every model is done over 250 epochs with a learning rate of 0.025 and a batch size of 10. The weights optimization algorithm is stochastic gradient descent.
In figure 5 the training and validation accuracy are plotted for a network where the dataset is divided into 70% of the dataset for training and 30% of the dataset for validation. In figure 5a the original dataset with the magnetograms from SOHO and SDO is plotted. In figure 5b the data set consists out of the original images as well as their vertical flipped relatives, which doubles the original data set. The validation accuracy of the normal dataset reaches 70% while the validation accuracy of the enhanced dataset is only 60%.

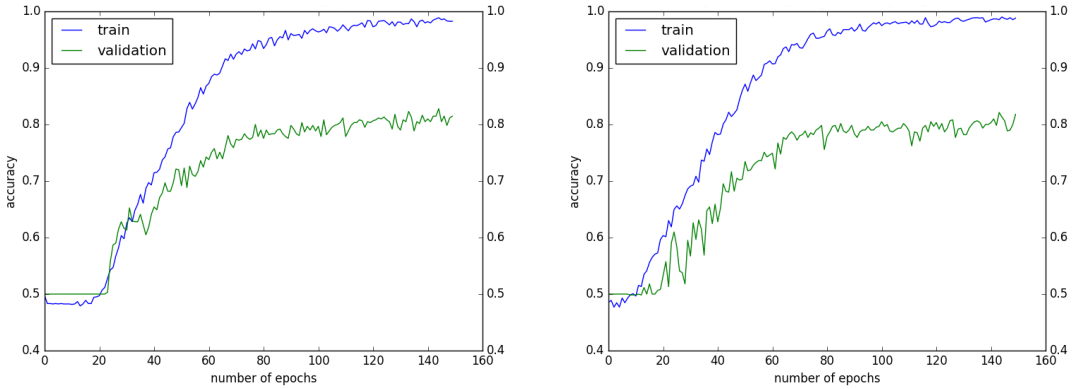


(a) original dataset

(b) dataset enhanced with the vertical flipped images

Figure 5: The train and validation accuracy for the network described in the introduction. The training is done over 250 epochs with a learning rate of 0.025 and a batch size of 10.

Figure 6 is made by using the magnetograms from SDO and SOHO. On parallel channels an extreme ultra violet image with a wavelength of 171 Angstrom is added to the dataset. In figure 6a the extreme ultraviolet image is added as a grayscale image, which create only 1 extra channel because each pixel is given by one grayscale value. In 6b the extreme ultra violet image is added as a red-blue-green(RGB)n figure 6 both the images reach a training accuracy of nearly 100% and a validation accuracy of 80%. This shows that there is extra information in the extreme ultraviolet pictures to help the network to identify the classes. However it also shows that a single grayscale image is as effective as a RGB image. Therefore it would be wise to use the grayscale images as they require half the amount of computations.



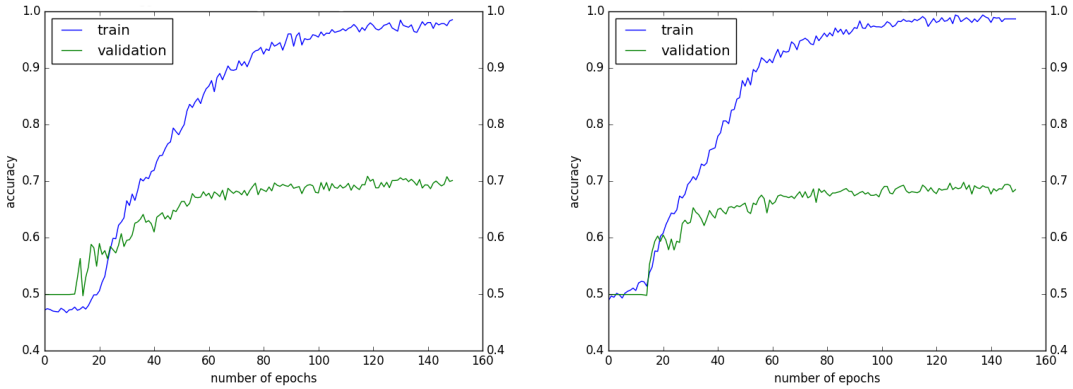
(a) Both images in grayscale

(b) Magnetogram in grayscale and extreme ultraviolet in color

Figure 6: The train and validation accuracy for the network described in the introduction. The dataset consist of an magnetogram stacked on top of an extreme ultra violet image with a wavelength of 171 angstrom. The training is done over 250 epochs with a learning rate of 0.025 and a batch size of 10.

To find out whether the same relation between the size of the dataset and the accuracy exist as shown in figure 5a and figure 5b, the test and validation accuracy for a dataset, which contain the stacked images as well as their flipped counterparts, is checked. As shown in figure 7 the validation accuracy is 10% lower than the dataset without the flipped images. This is the same accuracy drop as was previously observed. This suggest that the size of the dataset is influencing the accuracy of the network.

To gain further insight on the influence of the size of the training dataset on the validation accuracy of the network, the network is trained with different proportions of the training and validation sets. Hereby the size of the total dataset remains the same. The result is shown in figure 8. The validation accuracy for datasets with 60% or less data in the training set is around 55% which is very low. This suggests that the network is just doing slightly better than just guessing. However a network with 70% of more data in the training set archives a quite high validation accuracy of 70% or more.



(a) Both images in grayscale

(b) Magnetogram in grayscale and extreme ultraviolet in color

Figure 7: The train and validation accuracy for the network described in the introduction The dataset consist of an magnetogram stacked on top of an extreme ultra violet image with a wavelength of 171 angstrom. The data also contains their respective vertical flipped images. The training is done over 250 epochs with a learning rate of 0.025 and a batch size of 10.

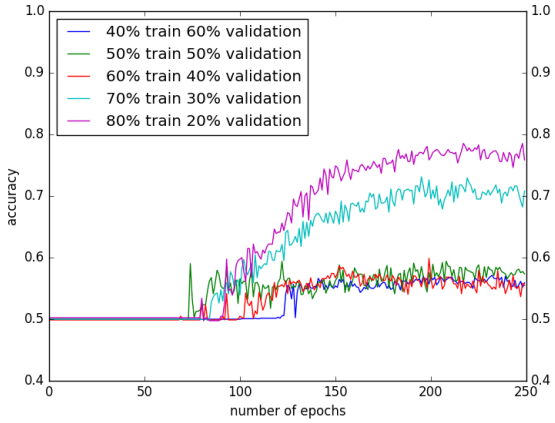


Figure 8: The validation accuracy for datasets with different proportions of images in the training and validation set. The training is done over 250 epochs with a learning rate of 0.025 and a batch size of 10.

4 Discussion

As shown in figure 6 the network that uses a database with a combination of a magnetogram and a extreme ultra violet image combined gives the best validation accuracy of 80%. This network does however overfit the test data quite significantly with a test accuracy of almost 100%. However if the size of the database is increased by adding the flipped images to it, the validation accuracy is lowered to 70%. The same relation can be seen between figure 5a and figure 5b. This suggest that either the network does not generalize enough, the key features are not present in every image or that there are not enough images from training and validation.

Whenever the image is flipped, the characteristic features remain in the image yet they are mirrored. Only the relative positions of characteristics are changed. In our case the filter should recognize a part of the white or black spot and output a high signal. When an image is mirrored these white and black spots still exist. Therefore the filters from the convolution layers should recognize these features and output a high signal even when they are mirrored. It could be that the filters do not generalize well enough or that the hidden layer does not train well for different relative positions. This lack of generalization is also present in high training rate. Due to this lack of generalization the network will give a higher accuracy if the images are all in a certain orientation of the image.

Besides the generalization, the key characteristics for the sunspots are not present in every image that is classified in class 1. There are images classified as class 1 that are just an uniform gray disk, without the black and white spots which indicate increased magnetization. Therefore the network cannot train perfectly on these key characteristics as it has false positives in the dataset. These false positives could arise from a solar wind with a different origin than a sunspot, for example a coronal hole. These coronal holes become more visible in the extreme ultra violet images. This could be the reason why figure 6 has a higher accuracy than figure 5. The problem with the key characteristics is also visible for images in class 2. In this class there are some images that contain sunspots. This is probably due to the relative large lifespan of sunspots. If an image is taken within a day of a large solar wind the chance that the sunspots are still present is quite large. However there is no large solar wind present at the current time. Therefore a picture with sunspots

end up in class 2. This also leads to the question whether a sunspot always lead to a solar flare. There could easily be a large sunspot without the existence of a solar flare. The magnetic fields between the sunspots first need to reconnect with each other before bursting out some plasma into space. The problem with images in class 2 is that the images show signs of sunspot and thereby the characteristic features on which the images are classified on, yet there is no large solar wind present. Therefore the best thing the network could do is to give a warning whenever there are sunspots on the surface of the sun. The sunspots however are not directly linked to a solar wind. Therefore the network cannot classify the solar wind based only on the magneto grams.

Figure 8 shows that the validation accuracy changes whenever the proportion of pictures between the training and validation set is different. The validation accuracy rises whenever the validation set relatively contain less pictures. However the training accuracy is large for the given networks. Therefore the weights of the network are shaped in such a way that almost all training images are classified correctly. Even if relatively more pictures are in the test set, the network still reaches an training accuracy of nearly 100%. It is likely that an almost similar copy of the training images is classified correctly by this set of weights. Because the network trained on more training pictures, there are more images that are classified correctly. The change to find an almost similar image in the validation set grows because the size of the validation set decreases and the size of correctly classified test set increases. This causes a higher validation accuracy whenever there are relatively more pictures in training set.

To avoid the dependency of the validation accuracy on the proportion of the test and validation sets, the amount of overfitting should be reduced. The amount of over fitting could be reduced by either using dropout layers, by using less parameters in the network or by using a larger dataset. In the network a dropout layer of 80% is already used between the convolutional layer and the hidden layer. This dropout is already huge and increasing the size even more would lead to an almost full reset of the learned weights after each epoch. Therefore the network would not be able to learn anything with an even bigger dropout layer. Secondly the amount of parameters is already quite low with a hidden layer of only 30 neurons. Therefore it would be unwise to significantly reduced the amount of neurons in the hidden layer. The input for this layer is already reduced by reshaping the images to a dimension of 112x112. By reducing the dimensions of the input vector even more the image will lose its characteristic features and the images becomes one uniform gray picture. Applying more limitation on the amount of parameters in the network would be challenging due to these two problems. Furthermore the amount of images could be increased to reduce over fitting on the dataset. However the databases only contain 20 years of images from the sun surface. There are not a lot large solar winds within this timespan which can be used as classification data for class 1. All images from SOHO and SDO that can be classified in class 1 are already used so there is no way the dataset can be increased.

5 Conclusion

It is not possible to classify solar winds from images of the sun outer layer. The validation accuracy is too reliant on the proportion of the training and validation set. As shown in figure 8 the validation accuracy is around 50% whenever the training and validation set are nearly the same size. The validation accuracy increases whenever the training set consist of more than 70% of the total data set. However the training accuracy is nearly 100% for every proportion of the validation and training set. All the training images are correctly classified by the given set of weights. The high validation accuracy most likely originates from near similar pictures in the training and validation set rather than from a generalized pattern that is learned by the network. To verify this behavior of the network a larger dataset should be used, however the databases from SOHO and SDO do not contain more pictures in class 1 due to the lack of large solar storms. Furthermore the dataset contains false positives in the large solar wind class which originates from other solar wind sources than sunspots such as coronal holes. The regular solar wind class also contains false positives since sunspots do not disappear rapidly and sunspots do not always lead to large solar winds. This makes it hard to give a high accuracy on the classification. Due to these reason the achieved accuracy cannot be immediately accepted and further research is needed.

References

[soh, 1997] (1997). soho documentation. Available at: https://sohowww.nascom.nasa.gov/publications/soho-documents/sop/sop.html.

[Cranmer, 2001] Cranmer, S. (2001). *Corona Holes*. Institute of Physics Publishing.

[Dungey, 1961] Dungey, J. W. (1961). Interplanetary magnetic field and the auroral zones. *Phys. Rev. Lett.*, 6:47–48.

[Guo Y., 2016] Guo Y., Liu Y., e. a. (2016). Deep learning for visual understanding: A review. page 4.

[Harvey and Sheely, 1979] Harvey, J. and Sheely, N. (1979). Coronal holes and solar magnetidc fields. *Space Science Reviews*, 23:139–158.

[Kamide, 1992] Kamide, Y. (1992). is a substorm occurence a necessary condition for a magnetic storm? *Journal of Geomagnetism and Geoelectricity*, 44:109–117.

[karpathy,] karpathy. neural network. Available at: *http* : *//cs*231*n.github.io/assets/nn*1*/neural_net.jpeg*.

[King, 1986] King, J. H. (1986). Solar wind parameters and magnetospheric coupling studies. In Kamide, Y. and Slavin, J. A., editors, *Solar Wind Magnetosphere Coupling*, volume 126 of *Astrophysics and Space Science Library*, pages 163–177.

[King and Papitashvili, 2005] King, J. H. and Papitashvili, N. E. (2005). Solar wind spatial scales in and comparisons of hourly wind and ace plasma and magnetic field data. *Journal of Geophysical Research: Space Physics*, 110(A2).

[Tsuneta et al., 1992] Tsuneta, S., Hara, H., Shimizu, T., Acton, L. W., Strong, K. T., Hudson, H. S., and Ogawara, Y. (1992). Observation of a solar flare at the limb with the YOHKOH Soft X-ray Telescope. *Publications of the Astronomical Society of Japan*, 44:L63–L69.

[Tsurutani et al., 1992] Tsurutani, B. T., Gonzalez, W. D., Tang, F., and Lee, Y. T. (1992). Great magnetic storms. *Geophysical Research Letters*, 19(1):73–76.

6 Appendices

6.1 Convolution Neural network

The idea of a neural network is based on information processing of our brain. The brain is made out of billions of connected neurons. These neurons are connected by dendrites and axons. A neuron receives an input signal from one of its dendrites and based on the strength of these input signals, the neuron send an signal to other neurons with it axons. In the interconnected network of the brain a lot of neurons are used to determine the output due to an input signal at one of the neurons.

Neural network
A neural network is created by connecting several neurons after each other where the output of one neuron is the input for another neuron as shown in figure 9

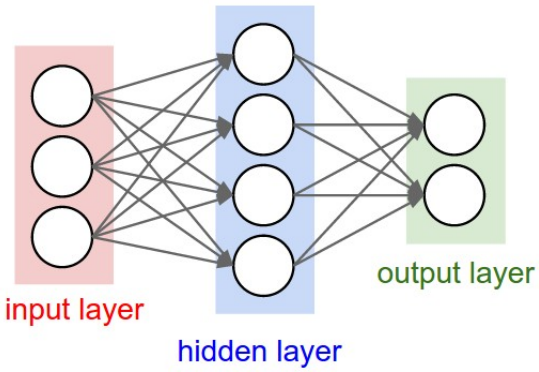


Figure 9: An example of a neuron network with 1 hidden layer. Used from [karpathy,]

The neural network consist of neurons in an input layer, one or several hidden layers and a output layer. The input layer consists of a vector with values between 0 and 1. The hidden layers receive data from neurons of the previous layer and apply the corresponding weight which determines the input for that particular link. The total inputed is summed up and and an activation function determines the output for the next layer. The output layer determines the output of the whole network. Depending on the system this could be a classification or number.

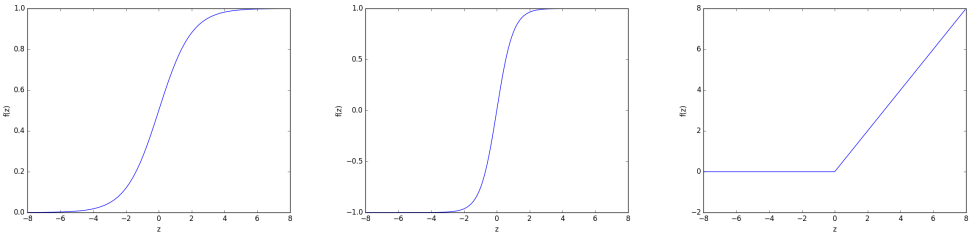
The neuron of a neural network has $x_1, x_2..x_n$ input links. All of the links have a weight W_i. This weight determines the contribution towards the input of the neuron. The total input z for a neuron

$$z = \sum_{i=1}^{n} x_i W_i + b \tag{7}$$

is given by the sum over all links x_i times their weight W_i plus a bias term. This bias term is usefull whenever the total input of the neuron due to other links is 0. The bias term then makes sure that an output is generated. The output O of a neuron will be determined by an activation function on the total input as

$$O = f(\sum_{1}^{n} x_i W_i + b) = f(z) \tag{8}$$

with f the activation function. The most commonly used activation function are the sigmoid (10a), hyperbolic tangent (10b) and the rectified linear activation function (10c) (Relu). The Relu function is found to work the best in deep neural networks. The advantage of Relu is that the output value of the neuron is not bound for large values. Furthermore for large input values the sigmoid and tanh functions have small derivatives whereas the derivative for the Relu function for $z > 0$ is constant and sufficiently large. The learning of a network is based on the derivative of the activation function and therefore using the Relu function speeds up the training.



(a) $\frac{1}{1+e^{-z}}$ (b) $tanh(z)$ (c) $max(0, z)$

Figure 10: Three types of activation functions for a neuron

The output O of the j^{th} neuron in the l^{th} layer is given by

$$O_j^l = f(\sum_i O_i^{l-1} W_{ij}^l + b^l) \tag{9}$$

with O^{l-1} the output of the neurons from the previous $(l-1)^{th}$ layer. The sum is taken over all i links of the $(l-1)^{th}$ layer to the current neuron. The weight W_{ij}^l is a matrix to make use of fast matrix multiplication methods in python and the gpu.

There are several possible activation functions for the output layer depending on the task of the network. For a multiple classification problem the softmax function can be used. The output O_j

$$O_j = \frac{e^{z_j}}{\sum_k e^{z_k}} \tag{10}$$

of the softmax classifier for the j^{th} class out of k possible classes with z given by 7 is used to determine the accuracy of the classification. The sum makes sure that

$$\sum_{j=0}^{n} O_j = 1 \tag{11}$$

such that the total of all outputs in the output layer sum up to one. Therefore the softmax function gives the accuracy of a given class as a probability.

Training

The training of a network is done to find the best set of weights W_{ij} such that the network has the highest accuracy for the classification of an image. When training the weights, the links between the neurons, will be optimized. This optimization is done with gradient descent.

Gradient descent requires to know how well the network is doing by defining a cost function C. The cost functions gives a quantification of the error on the classification. A high value of the cost function means that the network preforms poorly while a low value informs us that the network is achieving a high performance. For a network with the softmax classifier, the cross-entropy cost function is used. The cross-entropy cost function is defined by

$$C = -\sum_{i=1}^{n}\sum_{0}^{k}\{label_i = k\} ln(\frac{e^{z_j}}{\sum_k e^{z_k}}) \tag{12}$$

with n the number of neuron in the output layer. The first sum of the cross entropy sums over all neurons and adds up the cross entropy of the individual neurons. The second sum goes over all k classes. The term $\{label_i = k\}$ evaluates to 1 if $label_i$ is equal to k and zero otherwise, where $label_i$ is the label given to the input. This term only select the entropy of the label that correspond to the given input. The entropy of the other labels is not used in the calculation.

Gradient descent

The goal of gradient descent is to minimize the cost function. The gradient of a function shows in which direction the derivative is the largest and thus the function increases the most. Gradient descent uses the negative gradient to determine the direction in which the function decreases the fastest. In a neural network gradient descent is used to update the weights of the links such that the cost function reduces. The weights W_{ij} are update to the new weights W_{ij}' by using the negative gradient of the cost function as

$$W'_{ij} = W_{ij} - \eta\frac{\delta C}{\delta W_{ij}} \tag{13}$$

Where η is the learning rate of the network. The learning rate determines how fast a network updates its weights. If the learning rate is too large the minimum could never be reached because the gradient will overshoot the minimum of the cost function. If the learning rate is too small, one could be stuck in a local minimum because the steps of the gradient descent are too small and would never be big enough to go out of the local minimum. By iteratively using gradient descent, the minimum of the cost function will eventually be reached and the optimal set of weights are found.

In neural network the amount of parameters are large and the gradient need to be calculated for each parameter W_{ij} in W. For a single update of all the weights the network needs to calculate a lot of gradients and the computation time would be large.

To minimize the computation time, stochastic gradient descent(SGD) is used. A mini-batch of n weights is taken out of all the weights. SGD then calculates the average of the n gradients of the weights as

$$\nabla C = \frac{1}{n} \sum_{p=1}^{n} \frac{\delta C}{\delta W_p} \tag{14}$$

The average is then used to update all the weights in the matrix W by

$$W'_{ij} = W_{ij} - \eta \nabla C \tag{15}$$

This method circumvents calculating all the gradients which is the part that is the most computational intensive. In practice this turns out to be an efficient method that works quite well for updating the weights.

Back propagation

Whenever the cost function is calculated the weights need to be updated according to 13. The gradient however can not be easily calculated because the cost function does not linearly depend on the weights. This non-linearity already appears in a single neuron. A neuron collects all the inputs, passes them to an activation function and then output is passed to the next layer as shown in figure 11.

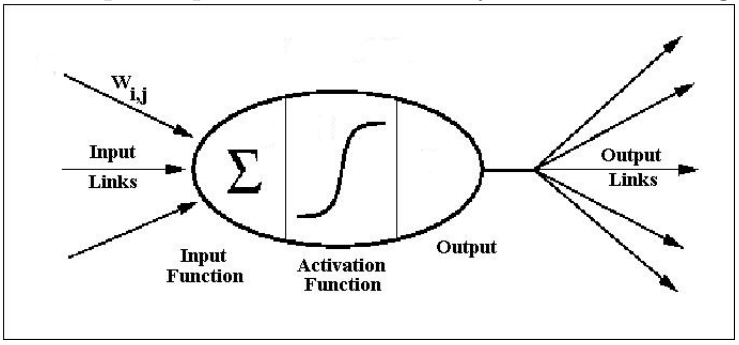


Figure 11: Forward propagation through a single neuron

Therefore to calculate the gradient with respect to a certain weight W_{ij} the chain rule has to be used. For the backpropagation on the last set of weights the chain rule

$$\frac{\delta C}{\delta W_{ij}^l} = \frac{\delta C}{\delta O^l} * \frac{\delta O^l}{\delta z^l} * \frac{\delta z^l}{\delta W_{ij}^l} \tag{16}$$

has to be used to calculate the gradient with respect to a weight of a link between the last hidden layer and the output layer. Here $\frac{\delta C}{\delta O^l}$ is the partial derivative of the cost function to the output of the l^{th}layer. To gain the output of a neuron one passes the total input through an activation function. Therefore $\frac{\delta O^l}{\delta z^l}$ gives the derivative based on this activation function. At last the derivative between the input and the contribution of the weights is taken in $\frac{\delta z^l}{\delta W_{ij}^l}$

To calculate the gradient of any weight in a lower layer one must repeatedly use the chain rule

$$\frac{\delta W^l}{\delta W_{ij}^{l-1}} = \frac{\delta W^l}{\delta O^{l-1}} * \frac{\delta O^{l-1}}{\delta z^{l-1}} * \frac{\delta z^{l-1}}{\delta W_{ij}^{l-1}} \tag{17}$$

However O^{l-1} does not depend on a single weight. All the links that the output contribute to must be taken in to account. For any weight in the system 17 is replaced by

$$\frac{\delta W^l}{\delta W_{ij}^{l-1}} = \sum_i \frac{\delta W_{ij}^l}{\delta O^{l-1}} * \frac{\delta O^{l-1}}{\delta z^{l-1}} * \frac{\delta z^{l-1}}{\delta W_{ij}^{l-1}} \tag{18}$$

with the sum over all links from the neuron in the l^{th} layer to the $(l+1)^{th}$ layer. So to calculate the gradient with respect to a single weight formula 16 in combination with 18 has to be used.

Calculating all the partial derivatives with the chain rule for a gradient would take a lot of computations. However a large proportion of the derivatives of the chain rule can be used in other calculations as well. This will highly speed up the calculation of the gradients.

Convolutional neural network

A convolutional neural network is similar in architecture as normal neural networks but they contain convolutional layers as well as hidden layers. A convolutional layer uses spatially information of an image by applying convolution

$$convolution = i * f \tag{19}$$

between a small subset of input image i and a filter f as shown in figure 12. The filter will look at a subset around the current point. This subset is the same size as the size of the filter. When using convolution, the filter maps this subset of points onto a single point. The filter slides along the width and height of the image until every pixel has been mapped.

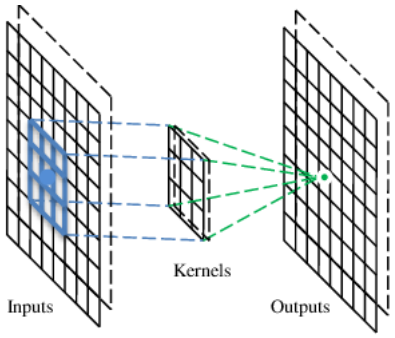


Figure 12: An example of a convolution layer. Picture originates from [Guo Y., 2016]

A convolution layer consists of several filters where each filter is used to recognize a different pattern in the dataset. If a filter detects a pattern in the dataset it will output a high signal to the next layer. This generates and activation map with recognized patterns. The hidden layers will then collect all the recognized patterns and output a signal based on the relative positions of the patterns.

Pooling layer Between several convolutional layers a pooling layer is often used. Most of the networks make use of a maxpool layer. The maxpool layer reduces the amount of parameters by selecting the highest activation for a filter in a small region.The relation between the features is more important than the exact location of these features. Therefore one can minimize the amount of data points but keep the relation between the features. By using maxpool the amount of parameters is greatly reduced but the relation between key features remains.

The pooling layer contains two parameters, a kernel and stride. The kernel with dimensions m x n defines the size of the filter and the stride determines the stepsize with which the filter slides across the image. For each kernel the maximum activation is selected out of all the values in the kernel as shown in fig 13.

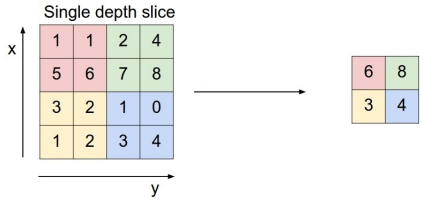


Figure 13: An example of a maxpool layer with a kernel of 2x2 and a stride of 2x2. Original from [karpathy,]