Deep Learning application to proton radiography analysis

Adrien Descamps
Stanford University
Aeronautics and Aerospace Department
adescamp@stanford.edu

Malo Marrec
Stanford University
MS&E Department
malo@stanford.edu

Abstract

To reconstruct the structure of the magnetic field, the deflection of charged particles through this field is recorded on a detector. The distribution of the particles on the screen are a consequence of the existing magnetic field. By analyzing these radiographs, we can infer the magnetic field that caused these deflections. However, the analysis is challenging when the amplitude of the field increases. In this paper we will propose a new way to analyze these radiographs based on Convolutional Neural Networks, which have demonstrated their ability to extract features from images. Two different approaches will be considered: (a) retrieve the integrated-line potential which is extremely challenging given the size of the dataset and (b) classify each radiograph based on the number of Gaussian blobs used to describe $\vec{B}$ in the object.

1. Introduction

Investigating high energy dense matter, a state where matter undergoes high energy stimulation, is crucial to understanding nature and develop a theory for matter at extreme temperature and pressure conditions. To generate this state of matter, we usually send high intensity short pulse laser on a target to heat it on a short time scale. The interaction between light and matter leads to the generation of a plasma inside the target. We can then retrieve information of the matter in extreme conditions from the analysis of this plasma.

However, one significant drawback is that under certain conditions, the plasma is opaque to light which makes the analysis difficult. One way to tackle this problem is to use protons rather than photons as probe particles. Indeed generating a plasma leads to charge separation inside the target, creating electric and magnetic fields. The interaction between these fields and the probe particles (protons) leads to a modification of the distribution of proton trajectories, which can be recorded on a detector, generating a *radiograph*.

For now, the analysis of the radiographs is done by guessing the 3-D shape of the electric and magnetic fields inside the target and use that guess to simulate a radiograph. The simulated radiograph is then compared to the experimental one, the guess for the fields is updated until both radiographs are identical.

As a first approach, we will analyze radiographs by training a neural network on simulated radiographs to retrieve the line-integrated potential ϕ, defined as:

$$\Phi(x, y) = \int_{-\infty}^{+\infty} A_z(x, y, z)dz$$

Where A_z is the potential vector such as $\vec{B} = -\nabla\Phi$ In a second part, the analysis will be restricted to getting the structure of the field $\vec{B}$ by classifying radiographs based on the number of blobs in the magnetic field.

2. Related work

Up to now there is only a few attempt to use Machine Learning to analyze data coming from experiments [1]. Even though it is possible to retrieve the integrated-line potential ϕ directly from a radiograph in the linear regime, there is in general no model that accurately describe the relation between a radiograph and ϕ. The capability of Machine Learning to generalize from data makes it compelling for this task.

Even though Machine Learning can bring a solution to this retrieval problem, there are a few difficulties that have to be considered. First, the difficulty to acquire data from these experiments reduces the amount of data realistically available to train a model. Second, radiographs coming from real experiments cannot be used to train any Machine Learning algorithms. Indeed, to train a model we need a training set and the corresponding labels. Those labels are unavailable in general as they are not available to the experimenter and are usually estimated through the trial and error process described above. Thus, all the training set and labels will come from simulations, where full knowledge of the ground truth is available.

From the wide range of simulated radiographs, we use the backpropagation property of neural networks to come up with a new analysis method.

3. Data Generation Process

3.1. Generation of radiographs using simulation

As described in [1], the target is radiated with a proton beam and proton deflection is recorded on the detector. To simulate our data, an object with a known a magnetic field $\vec{B}$ is radiated with a source (object in Figure 1), then the protons distribution is recorded in the image plane which leads to a radiograph.

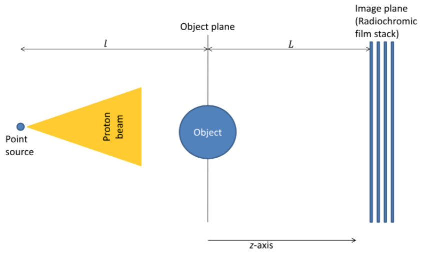


Figure 1. Scheme of a proton radiography imaging system

During the analysis, one might be tempted to predict the field $\vec{B}$ that generated the radiograph. This is in general not possible. Indeed, the 2D-radiograph is the result of the course of protons through a complex 3-D magnetic field. Consequently, many different fields can generate the same 2-D projection, which makes this problem highly non bijective. One approach to retrieve $\vec{B}$ is generating a 3-D radiograph by moving the proton source around the geometry. It is not explored here. However, by only using 2-D radiographs, the line-integrated potential Φ can be recovered, that corresponds to the total deflection of the proton throughout its course across the field. That is a necessary step to the 3-D approach.

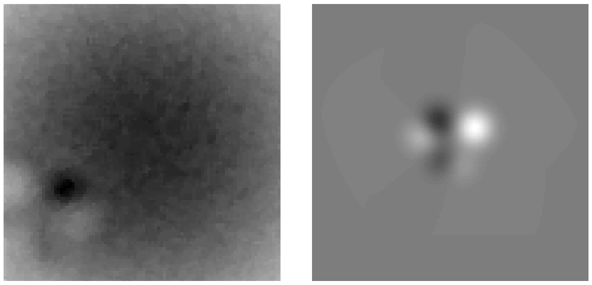
The scope of this study is restricted to magnetic field $\vec{B}$, even though the same method can be apply to an electric field. The following assumptions are made: the energy of protons is constant and the relative positions of the detector and the source.

To generate the data, different magnetic fields are explored over different shapes and intensities. The stimulated magnetic fields are a constellation of Gaussian blobs.

$$B_\Phi = B_0 \frac{r_0}{a} exp(-\frac{r_0^2}{a^2} - \frac{z_0^2}{b^2})$$

Where B_Φ is the only non zero component in spherical coordinates. In this case, the magnetic field is modeled as spherical blob (a = b). [2]

The parameters of this simulations will be the relative position in space of each blob as well as the intensity and size of each blob. We then simulate the radiograph corresponding to each field, and the line-integrated potential map, our label. The same simulator is used to generate training and test sets (see Figure.1).



(a) (b)

Figure 2. Legend: 691x691 pixels images (a) Simulated radiograph used for training, (b) Line-integrated potential Φ used as label, ($B_\Phi = 30T$)

The novelty of the approach as compared to as done in [1] consists of (a) using convolutional neural networks instead of simple feed forward neural network and (b) using radiographs with more complex magnetic field geometry making our approach more applicable.

Note that the dataset generation process introduces a bias since all the simulated radiographs come from the same simulator. Since all the simulators Maxwell Equations, there is a natural bias in all the radiographs. The value of this approach is to speedup the retrieval of Φ in a first place and to come up with a model that classify each radiograph based on the number of blobs used to describe $\vec{B}$, even though the model will be applicable under the assumption that we made on $\vec{B}$. We can then loosen those assumptions and try to to build a general model, but this is out of the scope of this paper.

3.2. Two different regimes

Based on the amplitude of the magnetic field, the perturbation undergone by the protons follow two different regimes. In the *linear regime*, where amplitudes of $\vec{B}$ are ”small”, the expected trajectory should be closed to the non-perturbed trajectory.

When the amplitude is ”high”, we enter the *non-linear regime* and the interpretation of the radiograph gets difficult as the perturbations are large. There is also no known analytical relationship between Φ and the intensity I of the radiograph in that regime.

We will first train our model on simulated data in the linear regime. We will then try to extent this approach to the linear regime, which we expect will be more difficult as it requires a larger dataset to ensure generalization.

3.3. Data Augmentation

As it is quite costly to generate data using our simulator, we augment our data by rotating the images by 90°, 180° and 270°. This makes sense from a physics perspective as the experiment is rotation invariant: turning the features results in a rotated radiograph.

Finally, the full process of generating data is summarized in the following pipeline.

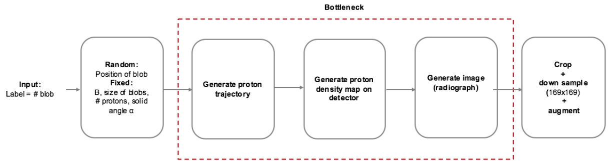


Figure 3. Pipeline for the simulation of the training data

4. First approach: Predict the integrated line potential Φ for each radiograph

4.1. Approach

To train our model, we first need a training set and a test set. We simulated as many radiographs as possible to get a dataset of reasonable size (10000 + images). This proved to be a major bottleneck as (a) each image generation takes a minute (b) the simulation process may run into numerical instability in some edge cases, the protons are deflected too much and are not recorded on the detector anymore. Once the dataset was created, we implemented the pipeline to feed our data to our neural network. That involved preprocessing and data augmentation.

To estimate the computation time required to train a model on our task, we considered the following baseline model: a 4 layers ConvNet with batch-normalization. We first didn't use max-pooling layers since we wanted to preserve the dimensions of our images throughout all the layers. Indeed we want in the end to produce a 2-D map of the integrated line potential which is also a 691 × 691 image.

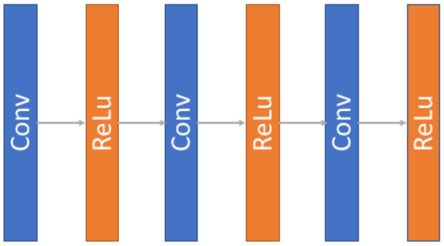


Figure 4. Architecture of the Neural Network used for sanity-check

4.2. Experimental Results

Training our neural network on this task is very challenging, and all of our losses looked very bumpy. We can explain these difficulties by (a) the high dimensionality of the problem and the small amount of data that reduce our capacity to explore the state space and (b) the size of our images constraints the batch-size to be 10 because of memory limitations.

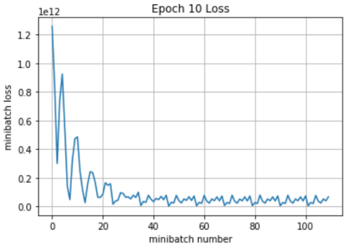


Figure 5. Evolution of the loss for different batch

The best loss we obtained is plotted below: note that the final loss is still extremely high. As we aim at predicting the line-integrated potential in the form of a 691 × 691 × 3. We opted for the L_2 norm to compare the features and the labels, this leads to a value for the loss that is unrealistically high. Creating an accuracy metric is also something that proves to be a challenge for that problem. To tackle the problem of dimensionality, we changed our approach and we reduced the task to a classification problem where each radiograph is classified based on the number of blobs used to simulate it. This step is important to characterize the field $\vec{B}$ in this case because it gives an idea of the complexity of the field in the target object.

As our data is based on simulation, we can generate labels with a lower dimension. Indeed, we can parametrize them by the number of Gaussian blobs we used in the simulation, their relative position as well as their intensity. That would restrict the applicability of the method since we are basically retrieving the magnetic field structures and not the integrated line potential anymore. Even though, this approach has been studied in [1], they limited their analysis to only one Gaussian blob. We want to extend this analysis to multiple blobs. Considering the following approach will reduce the output space size from a 691 × 691 × 3 image to only a few parameters ($\approx$ 10). In the part that follows, we focus on classifying the radiographs depending on the number of blobs they contain.

5. Second approach: Predict the number of blobs to describe $\vec{B}$ for each radiograph

5.1. Approach

As discussed in the previous paragraph, the amount of data available is limited and thus makes the retrieval of Φ

delicate. A second approach is to classify each radiographs based on the number of blobs used to describe $\vec{B}$. For the sake of this analysis, we fixed the intensity of the magnetic field used so that we can observe modifications in the distribution of the protons on the detector ($||\vec{B}|| = 40T$). Finally the training dataset consists in simulated radiographs for different number of blobs ($\leqslant 5$) and for random position of the blobs in the object. Even though the quality the image is an increasing function of the number of protons, we decrease it to cut down the computation time ($N_{Protons} = 5.10^6$).

In addition to the computation time, the classification of each image based on the number of blobs is a challenging task since the relationship between the number of blobs and the radiograph is highly non-linear (see the example below).

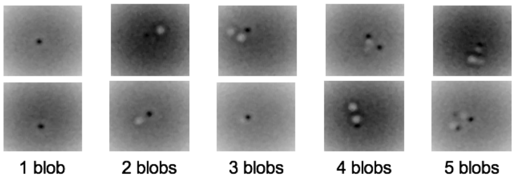


Figure 6. Illustration of the non-linear relationship between the images and the labels, high variance

From the above samples, one can easily understand that this task is extremely challenging even for a human since the number of blobs is not directly encoded in the images. A naive approach would be to count the number of circles we see on the picture. However this approach is only valid if the deflection of a proton only results from the interaction with only one blob. This corresponds to radiographs generated from small blobs with no overlapping. But in the most general case, the protons interact with many blobs which results in the images presented above.

To make a prediction about the number of blobs, we consider a CNN with a Fully Connected layer to compute the scores of each class for all training examples.

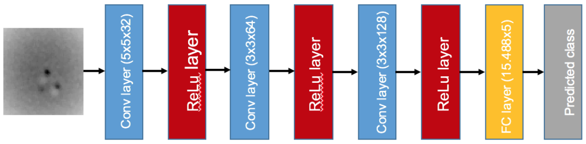


Figure 7. Architecture of the classifier used to predict the number of blobs

To ensure the robustness of the model and to minimize the impact of the background of the image on the prediction, we normalize all the training data by retrieving the mean value of the training dataset. Since we want to evaluate our model on similar images, we also retrieve the mean of the training set to the validation and test sets. (see Fig.8)

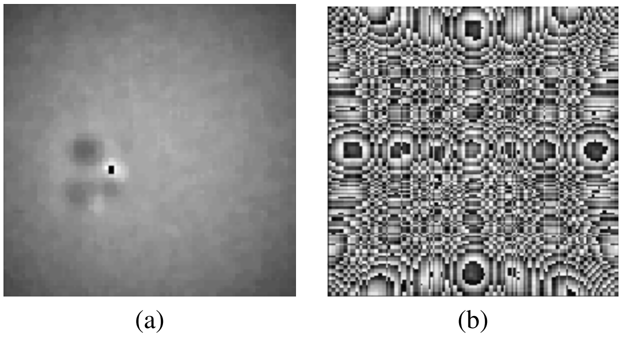


Figure 8. (a) Original sample from the training set, (b) Same sample after normalization by the mean of the training set

Before training our model, we resize all the images in order to speed up the computation time during the forward and the backward pass through the images. In the end the images are 169x169x3.

5.2. Experimental Results

The following model design characteristics and hyperparameters are tuned:

- Number of convolutional layers
- Size (depth) of convolutional layers
- Size and strides of filters
- Number of fully connected (FC) layers
- Size of FC layer
- Use of Batch Normalization
- Loss choice (cross-entropy, hinge loss, L_2 loss) + L_1 regularization
- Regularization scope and strength
- Learning rate, decay and annealing

Intuitively, adding many fully connected layers (FC) would not be optimal as that would very quickly increase over-fitting on our relatively small dataset. That intuition turned out to be true as adding more than one FC dramatically increased over-fitting. A number of filter size (3×3 to 10×10), strides and layer numbers (2 to 7) were tried for the convolutional layers. A model with 3 convolutional layers and one FC achieved the best results. Adding batch-normalization decreased performances in the best-performing model, though it slightly increased the performance of a deeper model. That makes intuitive sense as the deeper the stack of convolution, the more we can expect the input parameters of the deeper layers to change.

In terms of loss choice, the hinge loss performed in general slightly worse than the cross-entropy loss by a few % accuracy on the validation set.

The training is completed on 25 epochs, after which we see the validation accuracy starting to decrease. At this stage, the training set is nearly over-fitted. Adding more regularization strength does not yield better results than this model. The training is stopped when either the accuracy on the training set is 100% either the accuracy on the validation test starts decreasing.

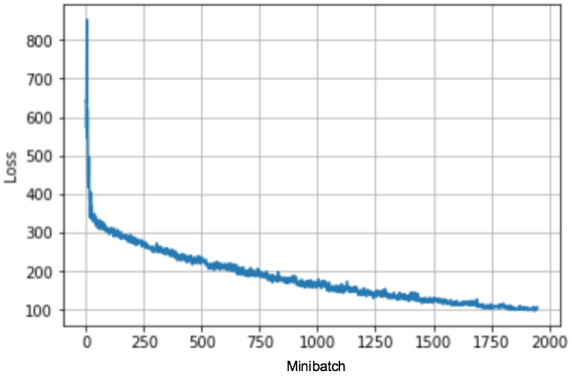


Figure 9. Evolution of the loss as a function of the number of minibatches during training

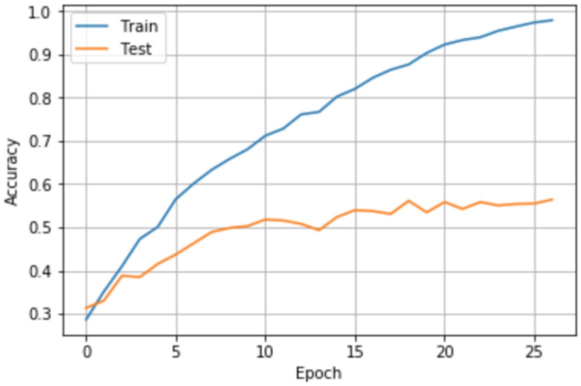


Figure 10. Train and Validation Accuracy as a function of the number of epoch at training time

The results on the best performing models are as follow. For reference, we achieved our best validation accuracy on $169 \times 169 \times 3$ images with a learning rate of $\lambda = 1.10^{-4}$, and a regularization parameter of $r = 1.10^{-1}$.

Image Size	Validation Accuracy	Test Accuracy
169x169x3	56.4%	54.8%
67x67x3	54.2%	54.1%

Figure 11. Best validation and test accuracies

Human accuracy on a sample size of 50, with balanced radiograph classes, is 38% (p-value vs null of picking classes at random stands at 0.1%). That means our method performs significantly better than humans.

Note that a slightly lower accuracy is achieved on smaller image sizes, however training is much faster. For future work on a larger dataset, having a deeper model with smaller images might be efficient in terms of both training speed and memory requirement (the training set becomes rapidly too big to fit in memory for larger images).

6. Limitations and Future Work

The size of the dataset has been a key limitation in this project. As we double the size of the training set, we saw a significant increase in performance as the validation accuracy went from 40% to more than 50%. The computation time at training time has to been tackled by reducing the size of the images.

It might be interesting to evaluate the performance of our model with a bigger dataset to prevent early over-fitting in the training phase. Once the dataset is ”big enough”, we can then increase the complexity of the task. Up to now we only developed a Neural Network for the number of blobs, this network would be coupled to a second network that would predict the position of the center of the blobs in the object. Indeed, based on the number of blobs predicted by the first CNN, we will output of 3-D vector corresponding to the center of the blob in space. This task is even more challenging than the first one since the predictions would in $\mathbb{R}^3$ (restricted to the position of the object). To complete this task, a larger dataset is needed since the dimensionality of the predictions is larger.

7. Conclusion

The study presented in [1] has been extended to more complex geometry for the $\vec{B}$ field (constellation of Gaussian blobs). ”Realistic” radiographs have been generated using simulation. The CNN trained on this dataset reaches a test accuracy of 54% which is better than human accuracy.

8. Acknowledgement

We would like to thank Joe Chen for his advice on the project, Paulo Alves for his help on the Physics of the problem and Maxence Gauthier for this help on the simulation of the radiographs.

References

[1] Nicholas F. Y. Chen et al. “Machine learning applied to proton radiography of high-energy-density plasmas”. In: *Phys. Rev. E* 95 (4 Apr. 2017), p. 043305. DOI: 10.1103/PhysRevE.95.043305. URL: https://link.aps.org/doi/10.1103/PhysRevE.95.043305.

[2] NL Kugland et al. “Invited Article: Relation between electric and magnetic field structures and their proton-beam images”. In: *Review of scientific instruments* 83.10 (2012), p. 101301.