Deep Regression Techniques for Decoding Dark Matter with Strong Gravitational Lensing

Personal Details

Name: Yurii Halychanskyi
University: Clark College, Vancouver, WA, United States
Email: renessmi2018@gmail.com
Phone: +13603564387
Github: https://github.com/Claussss
LinkedIn: https://www.linkedin.com/in/yurii-halychanskyi-a57590169/
Country of Residence: United States
Timezone: Pacific Standard Time

I am a sophomore Computer Science student at Clark College, Vancouver, WA. My semester ends on Thursday, June 17, therefore I will be able to work only part-time (~25 hours per week) the first week of the internship. I will also be working at SEH America part-time this summer, so If I am selected, I will be able to dedicate 30+ hours per week to Google Summer of Code.

Technical Knowledge

I am a sophomore at American college, but I had been studying at Chernivtsi Industrial College, Ukraine in the Computer Science department for 3 years. I didn't finish my last year due to emigration to the U.S. My course work was dedicated to a computer vision problem (face recognition).

I have participated in Fellowship.AI, where I gained a lot of experience as a data scientist.

I performed the following tasks:

- Worked with YOLOV3, person ReID models for surveillance video analytics (Pytorch was used for this project).
- Implemented OCR for analytics of workforce meeting records.
- Researched, modeled, and benchmarked classifiers for prediction of pizza doneness.

Currently, I am working as a Computer Science Intern at SEH America, where I improve my software development skills by developing .Net applications, working with SQL queries.

Programming languages I have previously worked with include C#,C++, Java, Python.

I have also worked on the following Kaggle Competitions:

- “Catch Me If You Can” Kaggle Competition (anomaly detection on time series)
- Lyft Motion Prediction for Autonomous Vehicles (real-time segmentation, classification, and regression techniques to predict motion of other cars)

Project

Abstract

This project aims to use deep regression techniques for estimating dark matter properties, including population-level quantities and properties of dark matter particle candidates.
Currently, DeepLense supports the following models for unsupervised dark matter classification:

- Adversarial Autoencoder
- Convolutional Variational Autoencoder
- Deep Convolutional Autoencoder
- Restricted Bolzmann Machine

Most of the listed models are surrogate base approaches, which means that they try to first learn feature representation with an **unsupervised alternative objective function**, and then assume that model will lead to poor performance to the anomalous data, which are not exposed to training, so that they can be distinguished from the normal data. While surrogate approaches have become the mainstream method for anomaly detection in recent years and have shown promising results, it is hard to ensure that the surrogate tasks share a consistent optimization direction with anomaly detection. For example, in AAE we try to minimize the reconstruction error to learn a proper representation of images without superfluid dark matter (the “no_sub” class). We try to optimize an alternative objective function (reconstruction loss) other than optimizing anomaly detection directly.

We could return to a **direct objective function for anomaly detection**, which maximized the distance between normal and anomalous data in terms of the **joint distribution for image and feature representation**. The above objective function is decomposed into the following fours components:

1. Mutual information between image space and latent space of normal data.
2. Entropy of normal data in latent space.
3. Expectations of cross-entropy between normal and anomalous samples in latent space.
4. Distribution distance between normal and anomalous samples in image space.

The first two components can be calculated with normal data only. The fourth component is a constant once normal data is selected. Only optimizing the third term requires anomalous data. To optimize the objective function, the third element could be bypassed, and then we can get a lower bound on the objective function which can be considered as trade-off between the mutual information and the entropy. **This anomaly detection objective function can be end-to-end optimized under the unsupervised setting.**

Technical Details

The objective function I am going to describe consists of two parts: **semi-supervised** and **unsupervised**. I will show formulas from both functions since the unsupervised one is derived from the semi-supervised.

Semi-supervised

$X\ -\ domain\ images$

$Z\ -\ representation\ of\ domain\ images\ in\ the\ latent\ space$

$X_n, Z_n -\ normal\ images$

$X_a, Z_a -\ anomalous\ images$

$p_n(Z)\ and\ p_a(Z)\ -\ marginal\ distribution\ of\ the\ latent\ representation$

$for\ normal\ and\ anomalous\ images.$

$p_n(Z\,|\,X)\ and\ p_a(Z\,|\,X)\ -\ conditional\ distribution\ between\ normal\ and\ anomalous$

$anomalous\ data\ in\ the\ latent\ space.$

$p_n(Z,\ X)\ and\ p_a(Z,\ X)\ -\ joint\ distribution\ for\ image\ and\ feature\ representation.$

$E_\theta -\ function\ that\ encodes\ the\ data\ to\ the\ low\ dimension\ latent\ space\ (E(X)\ =\ Z).$

$In\ our\ case,\ this\ is\ a\ neural\ net\ architecture\ with\ weights\ \theta.$

$KL\ -\ Kullback\ -\ Leibler\ divergence$

I will skip the proof part to save some space. All the proofs are provided in the paper.
Semi-supervised objective function:

$$max_\theta\, KL\, [p_n(Z,\ X)\,||\,p_a(Z,\ X)\,]\ =$$

$$\{I_n(X,Z)\ -\ H_n(Z)\ +\ E_{p_n(X)}[H(p_n(Z\,|\,X),\ p_a(Z\,|\,X))]\ +\ KL\,[p_n(X)]\,||\,p_a(X)]\}$$

Where:

1. $I_n(X, Z) \ = \ E_{p_n(X,Z)}[log(\frac{p_n(Z, X)}{p_n(X) \cdot p_n(Z)})]$, which refers to the mutual information between data sample X and its representation Z for normal data.

2. $-\ H_n(Z) \ = E_{p_n(Z)}[log(\frac{1}{p_n(Z)})]$, which is the negative entropy of Z.

3. $E_{p_n(X)}[H(p_n(Z \mid X),\ p_a(Z \mid X))] \ = \ E_{p_n(X)} E_{p_n(Z \mid X)}[-\ log(p_a(Z|X))]$, which is expected value of the cross entropy between the conditional distributions $p_a(Z \mid X)\ and\ p_n(Z \mid X)$.

4. $KL\ [p_n(X)] \mid\mid p_a(X)] \ = \ log(\frac{p_n(X)}{p_a(X)}) \ = \ C,$ $p_n(X)\ and\ p_a(X)$ are fixed, so the fourth component is a constant.

The decomposed objective function is an essential general formula for optimizing anomaly detection, which combines unsupervised learning (the first and second equation) and supervised learning (the third equation).

As **the first and the second components** can be **trained through an unsupervised** fashion and force the encoder to extract effective features, the demand for anomalous data is greatly reduced.

Unsupervised

$If\ p_n(X, Z)\ and\ p_a(X, Z)\ have\ a\ certain\ distance\ such\ that\ for\ most\ samples$ $X, Z\ \sim p_n(X, Z)\ the\ evaluated\ density\ p_a(Z \mid X)\ is\ small\ enough,\ such\ that$ $p_a(Z \mid X)\ <= p_n(Z)\ and\ p_a(Z \mid X)\ <=\ 1\ almost\ everywhere,\ then\ we\ can\ derive$ $a\ lower\ bound\ of\ the\ semi\ -\ supervised\ objective\ function$:

$$max_{\theta}\ \{I_n(X, Z)\ -\ H_n(Z)\}$$

This objective function can be optimized with an entropy-regularized Lagrange multiplier:

$$max_{\theta} \{I_n(X, Z) - \beta \cdot H_n(Z)\},$$

where $\beta >= 0$ is a positive coefficient for the weight of the entropy regularization.
To maximize the mutual information between X and Z $(I_n(X, Z))$, we apply the **Contrastive Predictive Coding lower bound with Noise Contrastive Estimation:**

$$I(X, Z) >= E[\frac{1}{K}\sum_{i=1}^{K} log(\frac{e^{f(X_i, Z_i)}}{\frac{1}{K}\sum_{j=1}^{K} e^{f(X_i, Z_j)}})],$$

Where f is the critic function that maps the inputs into a value R, which can be modeled with various methods, such as a similarity function or a neural discriminator. Here, X_i and Z_i are called positive pairs; while X_i and $Z_j = E_{\theta}(X_i \varrho_{[i \neq i]}),\ where\ \varrho_{[i \neq i]} \in \{0, 1\}$ is an indicator function evaluation to 1 if $i \neq i$, are called negative pairs. Empirically, two independently randomly augmented versions of the same sample ($\bar{Z}$ is augment

To minimize the entropy (second element), we seek the lower bound for version of Z),

e.g. an image and its rotated view, are often used as positive pairs. the negative entropy as $- H(Z) >= E_{p(X,Z)}[log(r(Z))]$, where $r(Z)$ is a reference distribution. One common choice is to let $r(Z) = N(0, I)$. Then, this lower bound is proportional to the L2 norm:

$$E_{p(X,Z)}[log(r(Z))] \propto - \frac{1}{N}\sum_{i=1}^{N} ||Z_i||_2^2,$$

Now our objective function can be optimized in an unsupervised manner:

$$\frac{1}{N}\sum_{i=1}^{N}[- log(\frac{e^{f(Z_i, \bar{Z}_i)}}{\frac{1}{K}\sum_{j=1}^{K} e^{f(X_i, \bar{Z}_j)}} + \beta \cdot ||Z_i||_2^2]$$

Methods that use this objective function outperform state-of-the-art methods. Results on different datasets are shown in the paper.

Timeline

Pre GSOC

Learn more in-depth theory about strong lensing and different kinds of WIMP. Practice simulating different kinds of strong lensing images by using PyAutoLens.

Community Bonding

Exploring DeepLense code written last year and defining a set of models for unsupervised anomalous detection that can be implemented with the objective function described above. The objective function can be modified for different algorithms (like Autoencoders or SVDD), so It wii worth spending some time to find the best combination.

Week 1

Start designing the logic for the chosen algorithms. In the paper, they described algorithms mostly in an abstract way, so we will need to choose some specifications. For example, $||Z_i||_2^2$ is proportional to L2 norm if the reference distribution is the normal distribution, $(r(Z) \ = N(0, I))$ but if $r(Z) \ = L(0, I)$ (standard Laplace distribution), the entropy is proportional to L1 norm, $||Z_i||_1$.

Week 2

Implement the algorithms in Pytorch.

Week 3-5

Compare results of the implemented algorithm. Choose the best combination of an anomalous detection algorithm and objective function version.

Week 6

Tune algorithms hyperparameters. For example β, which is the weight of entropy regularization in the equation of the objective function.

Week 7-8

Improve training loop (expand dataset, add augmentations, tune training hyperparameters, add Pytorch schedulers, experiment with optimizers).

Week 9

Edit the model based on the feedback from the community; document the code, clean repositories.

Week 10

Get/give final feedback.

References

Most of the content of the document is based on the following paper:

[Deep Unsupervised Image Anomaly Detection: An Information Theoretic Framework (Fei Ye, Huangjie Zheng, Chaoqin Huang, Ya Zhang)]