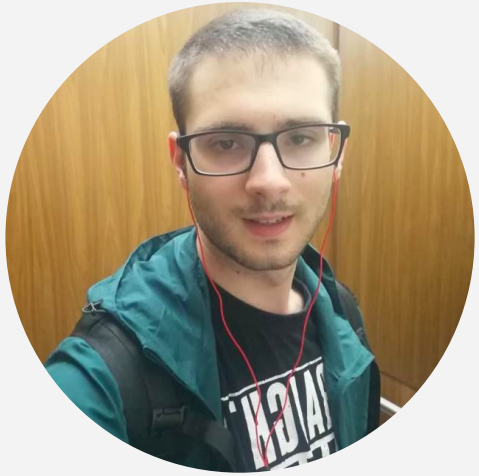
Team 22 : Prediction model using COVID-19 chest X-ray and clinical notes



Andi/高安迪, T08902144

- LogisticRegression / Decision tree (clinical notes analysis)
- Link platform to X-ray chest analysis

Elena, T08902135

- LogisticRegression / Decision tree (clinical notes analysis)
- Platform front-side
- Link platform to clinical note analysis

洪筱慈, R08922A20

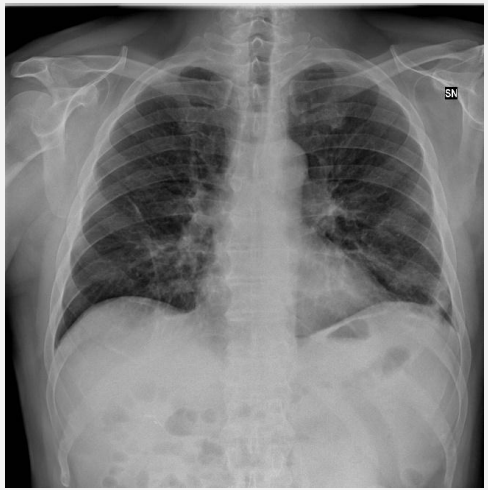
- Reference search
- Dataset preprocessing
- X-ray prediction model
- focal loss solution to imbalanced data in X-ray data

Communication : e-mail, messenger, face-to-face team meeting once per week

OUTLINE

1. Motivation/Background
2. Related works
3. Target problem
4. Proposed solution
5. Experiments
6. Reference
7. Q&A

Although there are already some existing works on using Chest X-ray to identify COVID-19 Infection, the sensitivity to COVID-19 still needs to be improved to be practical.



MOTIVATION & BACKGROUND

MOTIVATION & BACKGROUND

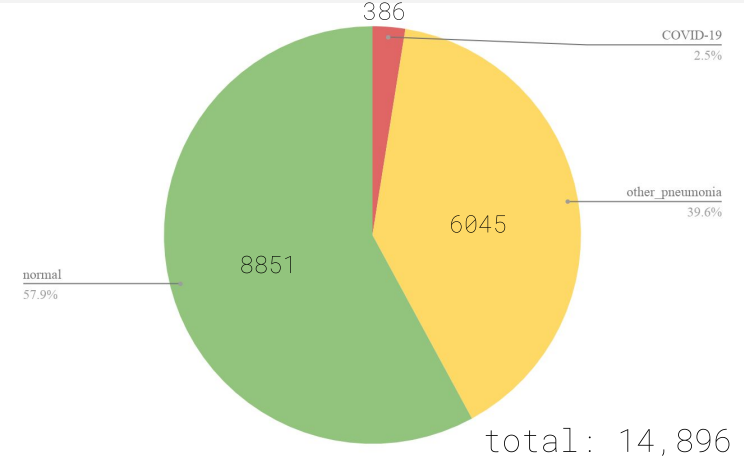
When digging into the datasets, we found two things:

1. Some datasets contain clinical notes of patients. This might help the identification of COVID-19, because lung damages are not a necessary symptom of COVID-19.

'On January 20, 2020, a 55-year-old woman who worked in Wuhan, China, arrived at Taiwan Taoyuan International Airport and presented to quarantine officials immediately, with a history of sore throat, dry cough, fatigue, and low-grade subjective fever since January 11, 2020. Apart from a history of hypothyroidism with regular medical follow-up, she had no other underlying disease before this onset. Chest X-ray showed progression of prominent bilateral perihilar infiltration and ill-defined patchy opacities at bilateral lungs, which slowly resolved on the follow-up image.'

When digging into the datasets, we found two things:

2.Current X-ray data is imbalanced:



MOTIVATION & BACKGROUND

Based on our findings, we decided to focus on:

1. Using the clinical notes for prediction.
2. Dealing with data imbalance problem existed in X-ray prediction model.

MOTIVATION & BACKGROUND

RELATED WORKS

- Clinical notes analysis

- X-ray prediction

  COVID-Net[1] and covid-cxr

  are two existing COVID-19 prediction models using chest X-ray.

TARGET PROBLEM

- Clinical notes analysis

- X-ray prediction:

  Handle Imbalanced Data in X-ray dataset

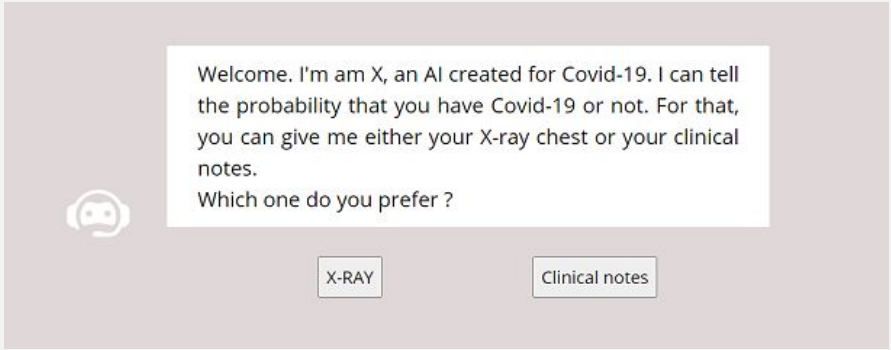
PROPOSED SOLUTION

- Two applications :

1. Clinical notes analysis :
   - Input : patients symptoms
   - Output : probability to have covid-19 based of symptoms
2. X-ray chest analysis :
   - Input : patients X-rays
   - Output: classification result and confidence

PROPOSED SOLUTION

- A website platform linked to Python algorithms :



- Usable in real-life for every users
- Easy to use without specific knowledge

PROPOSED SOLUTION

I. Clinical notes analysis

Decision Tree	Logistic Regression
bisects the data into smaller and smaller groups	separates the data into exactly two group through a single line
can capture the division of classes better but is bad if the classes aren't well distinctable	better, when training data is not well-separable
good for categorical data (f.e. string format)	cannot handle categorical data, needs to be converted into numbers
But: because of our small dataset the outcome of those two was not different	

PROPOSED SOLUTION

I. Clinical notes analysis

Limitations :

- Covid-19 is new, the datasets are not as big as others
- Accuracy of the model is then questionable (i.e : no one in the dataset has fatigue without having covid-19, so model automatically link fatigue to covid-19

PROPOSED SOLUTION

II. Imbalanced classes in chest X-ray dataset

Common ways to deal with imbalanced classes:

First kind of methods are **data-driven**, which means increasing minority or decreasing majority.

1. Over-sampling
2. Under-sampling

However, these methods might cause overfitting or discard useful information.

PROPOSED SOLUTION

II. Imbalanced class on chest X-ray dataset

Another kind of methods are **Loss-driven**, which is controlling the loss for classes during training stage. The goal is to force the model to focus more on minority class or hard samples during training stage.

1. class weight
2. Focal loss[2]

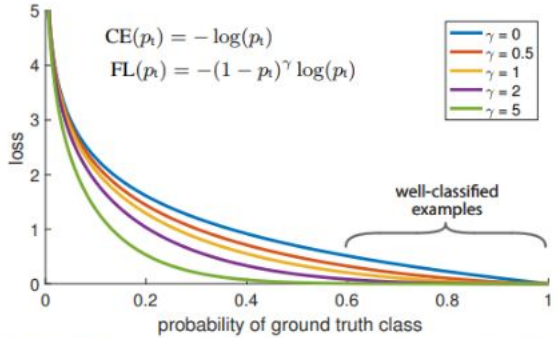


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^{\gamma}$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.

We focus on implementing **focal loss**[2] on the existing model structure.

It forces model to focus more on difficult samples.

Original cross-entropy: $CE(p,y) = CE(p_t) = -\log(p_t), \quad p_t = \begin{cases} p, & if\ y=1 \\ 1-p, & O.W. \end{cases}$

α-balanced cross-entropy: $CE(p_t) = -\alpha \log(p_t)$

focal loss: $FL(p_t) = -(1-p_t)^{r}\log(p_t)$

I. Clinical notes analysis

Decision tree data :

- X : symptoms of each patient under array form [fever, cough, sore throat, ards, fatigue] ([0, 1, 0, 0, 0] = having cough)
- y : array of patients based on covid-19 status ([0, 1, 0] = 2nd patient has covid-19

EXPERIMENT

I. Clinical notes analysis

- Decision tree algorithm :

```
class Predictor(Resource):
    def __init__(self, dataset_path='mila_metadata.csv'):
        self.test = 0
        self.dataset_data, self.dataset_target = self.read_csv(dataset_path)
        self.model = self.train()

    @staticmethod
    def read_csv(file_path='mila_metadata.csv'):
        mila_df = pd.read_csv('mila_metadata.csv')
        mila_df.head()

        coronaOrNot = mila_df['finding']
        data = mila_df['clinical_notes']

        # Creates a list containing x patients, each of 6 items, all set to 0
        h, w = coronaOrNot.size, 5
        symptoms = [[0 for x in range(w)] for y in range(h)]
        # symptoms [[fever, cough, sore throat, ards, fatigue]]
        # ie : [1, 0, 0, 0, 0] : only have fever

        target = []
        # target [0, 1, 0.....]
        # 2nd patient has corona
```

EXPERIMENT

I. Clinical notes analysis

- Decision tree algorithm :

```
for i in range(data.shape[0]):
    if ("covid-19" in str.lower(coronaOrNot[i])):
        target.append(1)
    else:
        target.append(0)

    if ("fever" in str.lower(str(data[i]))):
        symptoms[i][0] = 1
    else:
        symptoms[i][0] = 0
    if ("cough" in str.lower(str(data[i]))):
        symptoms[i][1] = 1
    else:
        symptoms[i][1] = 0
    if ("sore throat" in str.lower(str(data[i]))):
        symptoms[i][2] = 1
    else:
        symptoms[i][2] = 0
    if ("ards" in str.lower(str(data[i]))):
        symptoms[i][3] = 1
    else:
        symptoms[i][3] = 0
    if ("fatigue" in str.lower(str(data[i]))):
        symptoms[i][4] = 1
    else:
        symptoms[i][4] = 0
```

EXPERIMENT

I. Clinical notes analysis

- Decision tree algorithm :

```
        return np.array(symptoms), np.array(target)

    def train(self):
        reg = tree.DecisionTreeClassifier().fit(self.dataset_data, self.dataset_target)
        return reg

    def predict(self, x):
        symptoms = []
        # symptoms [[fever, cough, sore throat, ards, fatigue]]
        if ("fever" in str.lower(x)):
            symptoms.append(1)
        else:
            symptoms.append(0)
        if ("cough" in str.lower(x)):
            symptoms.append(1)
        else:
            symptoms.append(0)
        if ("sore throat" in str.lower(x)):
            symptoms.append(1)
        else:
            symptoms.append(0)
        if ("ards" in str.lower(x)):
            symptoms.append(1)
        else:
            symptoms.append(0)
```

EXPERIMENT

I. Clinical notes analysis

- Decision tree algorithm :

```
        if ("fatigue" in str.lower(x)):
            symptoms.append(1)
        else:
            symptoms.append(0)

        return self.model.predict_proba(np.array(symptoms).reshape(1,-1))


    def get(self, clinicalNotes):
        pred = self.predict(clinicalNotes)
        proba_no_covid = pred[0][0]
        proba_covid = pred[0][1]
        proba_covid = round(proba_covid,2) * 100
        return jsonify(proba_covid)
```

EXPERIMENT

I. Clinical notes analysis

Decision tree vs Logistic regression results

```
i have fever
Decision tree :  94.69999999999999
Logistic Regression :  93.8


i have fever and cough
Decision tree :  92.80000000000001
Logistic Regression :  92.5
```

- decision tree more precise

EXPERIMENT

II. X-ray chest analysis

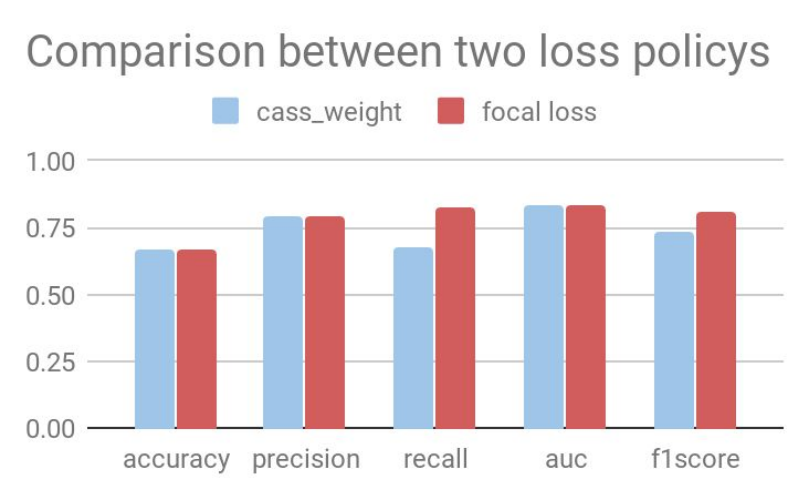
- Datasets used for experiment:
  - covid-chestxray-dataset[3]
  - Figure1-COVID-chestxray-datase[4]
  - RSNA Pneumonia Detection Challenge[5]
- Model used for experiment:
  - DCNN-Resnet in covid-cxr
  - # of convolutional blocks: 4
- Fixed training parameters:
  - Number of Epochs: 200
  - Learning rate: 0.0002
  - Batch size: 32
  - Optimizer: adam
  - Classification threshold for prediction: 0.5
- Data imbalance strategy parameters:
  - Class weight: (Normal: 0.4, Pneumonia: 0.4, COVID-19: 1.0)
  - Focal loss: gamma=2.0, alpha=4.0

EXPERIMENT

EXPERIMENT RESULT

Due to the limited time, we decided to use a small subset for the experiment.

COVID-19	Pneumonia	Normal
274	532	500



Focal loss do improve both the recall and f1-score, and we believe it also work on the whole dataset.

REFERENCES

[1]Linda Wang, Alexander Wong. 2020. COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest **X-Ray** Images. arXiv:2003.09871. Retrieved from https://arxiv.org/abs/2003.09871

[2]Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollár. 2017. **Focal Loss** for Dense Object Detection. *In Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[3]Joseph Paul Cohen and Paul Morrison and Lan Dao. 2020. **COVID-19** image **data** collection, arXiv:2003.11597. Retrieved from https://github.com/aildnont/covid-cxr

[4]COVID-Net Team. 2020. **Figure1-COVID-chestxray-dataset**. Retrieved June 1, 2020 from https://github.com/agchung/Figure1-COVID-chestxray-dataset

[5]Radiological Society of North America. 2018. **RSNA Pneumonia Detection Challenge**. Retrieved June 1, 2020 from https://www.kaggle.com/c/rsna-pneumonia-detection-challenge

Q & A

Q : Can you talk more about the decision-tree chat-bot? It seems like a cool idea. Is separate from the x-ray check?

A : Thank you. The platform is an Angular project linked to the Python code. The clinical analysis is based on the model Decision Tree. It was originally a help for x-ray check, but it happens to be seperate.

Q : Maybe it's a better way to consider the patient's location?

A : We can. Here is an example of a clinical note in the dataset “50-year-old man was sent to the fever clinic for fever, chills, [...] He reported the travel history of Wuhan from January 8 to 12, and the first symptoms appeared on January 14[...]” We can take localisation like Wuhan in account. As long as it’s an information in the notes, we can.

Q : What is the model used to calculate the probability of getting COVID-19 when entering clinical notes?

Q : What is the size of your clinical notes data?

Q : how did you obtain the result 67%?

A : The model is decision tree. The major problem of our model is the dataset, being quite new and small (400+ cases). It calculates based on Dataset but for example every person that have fatigue have covid and nobody has fatigue and another sickness, so the model associates fatigue 100% to covid. That's how it works for the probability we only need a bigger dataset but couldn't find one big enough. If we have 67%, it means that in our dataset, most of patient who have COVID-19 also have cough while patients that have others sickness don’t have cough.

Q : How could you differentiate between covid19 and other chest/lung related diseases?

A: The dataset is labeled by professional doctors, not by ourselves. So the diagnosis may based on the knowledge of X-ray reading.

Q : What is the model you choose to classify X-ray image data? It seems like your data is too small to train a deep neural network model.

A: We use ResNet for the training. A brief introduction about ResNet: https://reurl.cc/pdRe6Q

The size of the subset we used for experiment is 1306. It's true that the size is a bit small so we cannot get good f1-score, but we only want to see if the "focal-loss" method could work. We can still train the model on the whole dataset, which contain 14,896 images, we should receive much higher performance on prediction.

Q : Covid-19 is 7.5% of data, what is the real number of dataset

Q : How does the covid-19 percentage calculate?

A: The size of the whole dataset is 14,896, and the number of COVID-19 positive is 386. As a result, the percentage of covid-19 is 7.5.

The dataset is from COVID-net[1], and they gathering these datasets together to make 14,896 samples:

- https://github.com/ieee8023/covid-chestxray-dataset
- https://github.com/agchung/Figure1-COVID-chestxray-dataset
- https://github.com/agchung/Actualmed-COVID-chestxray-dataset
- https://www.kaggle.com/tawsifurrahman/covid19-radiography-database
- https://www.kaggle.com/c/rsna-pneumonia-detection-challenge (which came from: https://nihcc.app.box.com/v/ChestXray-NIHCC)

Thanks for reading !