UNIVERSITÀ DELLA CALABRIA
DIPARTIMENTO DI MATEMATICA E INFORMATICA

Understanding neural network-based classifications: an application to COVID-19 diagnosis

Pierangela Bruno

post-Doc
Department of Mathematics and Computer Science,
University of Calabria, Italy

Artificial Neural Networks: definition

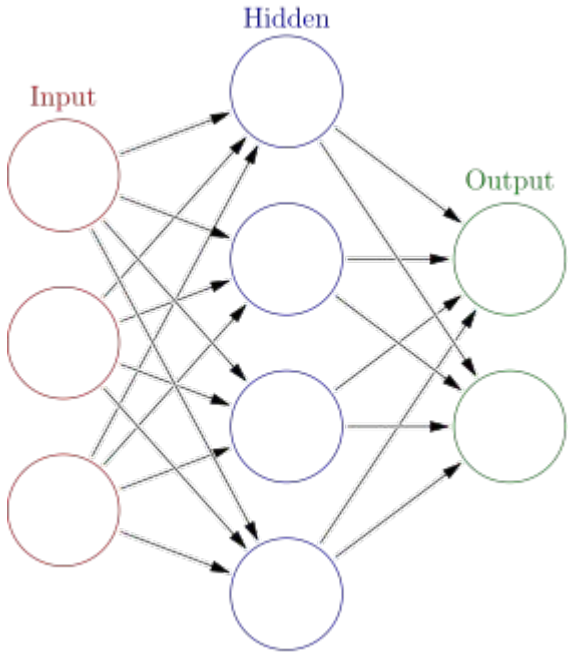
Artificial neural networks are

- computational models inspired by biological neural networks
- used to approximate functions that are generally unknown

Artificial neural networks are composed of various layers of interconnected artificial neurons powered by activation functions

- to decide whether a neuron should be selected (i.e., activated) or not.

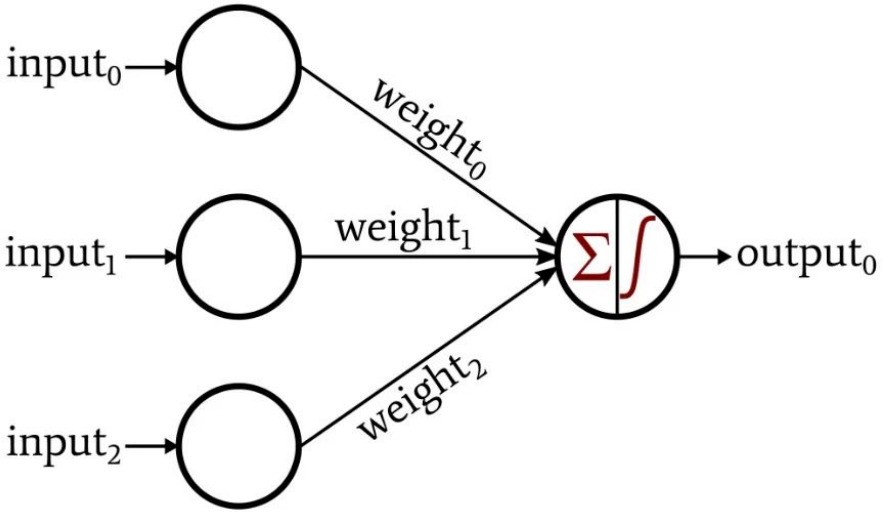
Like traditional machine algorithms, neural networks learn the features of data in the training phase.



ANNs: Perceptron

A perceptron is an algorithm for supervised learning of binary classifiers. It takes several binary inputs and produces a single binary output.

- Each connection between input and output has an associated weight which expresses the importance of the input to the output
- The neuron's output is determined by whether the weighted sum of the inputs is less (or greater) than some threshold value

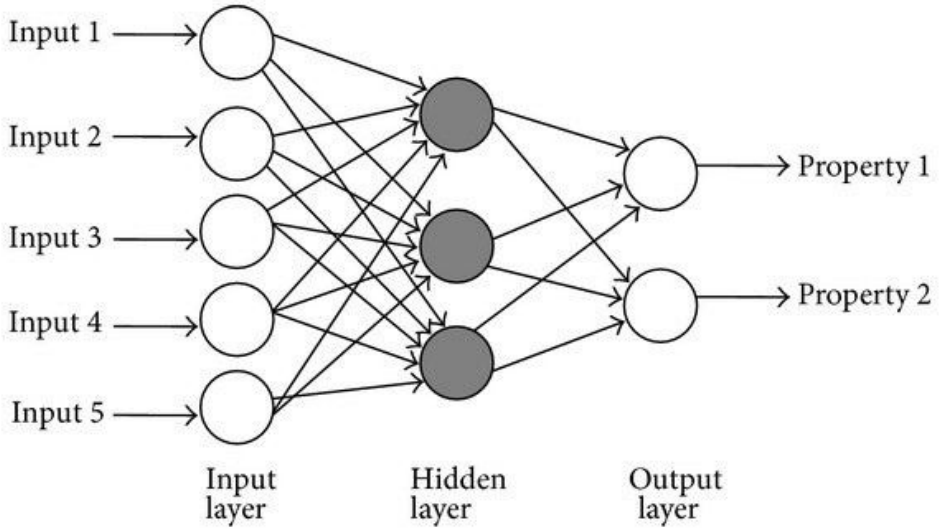


ANNs: Multilayer Perceptrons

It is composed of more than one perceptron, stacked in several layers, to solve complex problems.

It consists of three types of layers:

- The input layer that contains input neurons
- The output layer that contains output neurons
- The hidden layer that contains hidden neurons



Each perceptron in the first layers sends outputs to all the perceptrons in the second layer and all perceptrons in the second layer send outputs to the final layer (output layer)

ANNs: Perceptron Learning Process

The output of a network is strictly related to its **weights** and **biases**; they can influence the overall behaviour of the network.

- The weight allows the perceptron in evaluating the relative importance of each of the outputs
- The bias allows the classifier to turn its decision boundary around

Perceptron learning process relies on:

- finding weights and biases able to approximate any kind of input
- a cost function to evaluate the performance in achieving the goal - e.g., mean squared error (MSE)

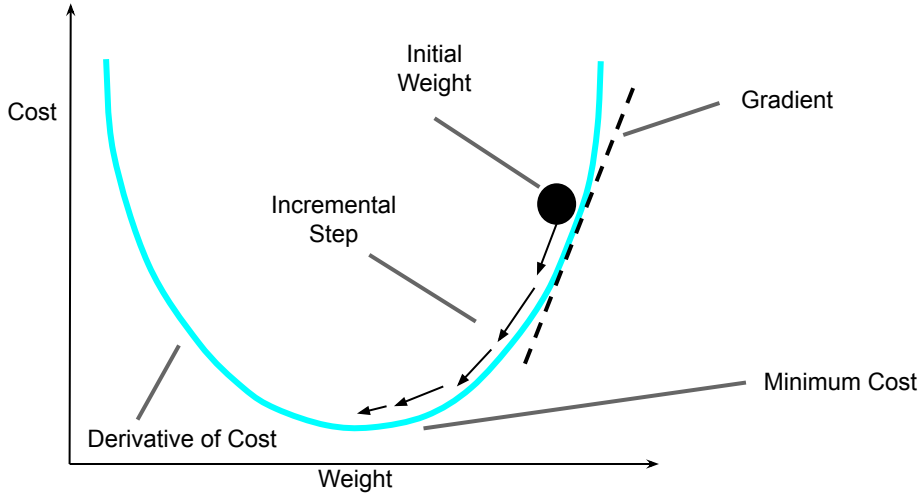
There are several algorithms used to fine tune the weights, the most common in supervised learning is called Backpropagation with gradient descend.

ANNs: Backpropagation

Backpropagation helps to adjust the weights of the neurons to obtain a result closer to the known true result and to find the minimum of the error function using gradient descent.

Imagine a ball rolling down the slope of the valley.

- Gradient descent simulates the motion of the ball by computing the derivative points, indicating in which direction the ball should roll to
  - reach a local minimum in our search space



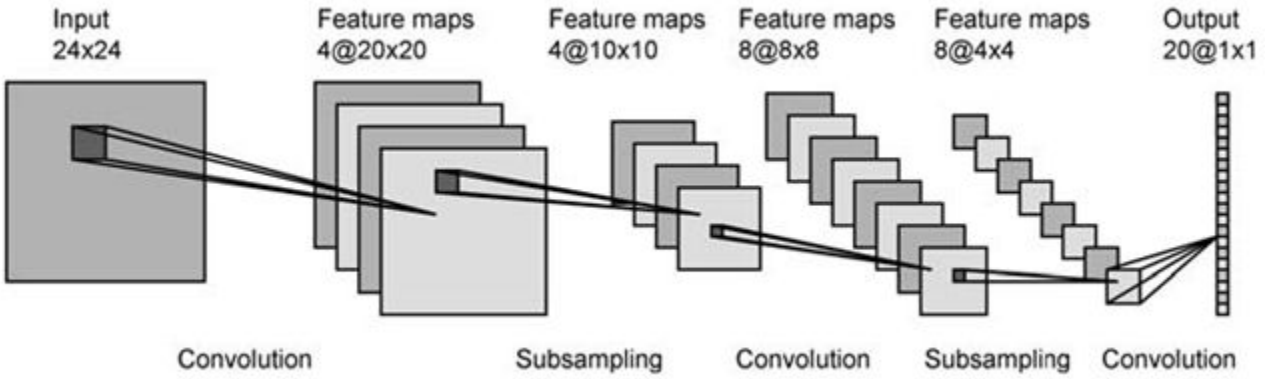
ANNs: Convolutional Neural Networks

Fully connected networks were found to be inefficient for computer vision tasks - e.g. classify images.

Limitations

- Too many connections - each neuron is connected to all neurons in the next layer
- The spatial structure of the images are not considered
- Hard to train

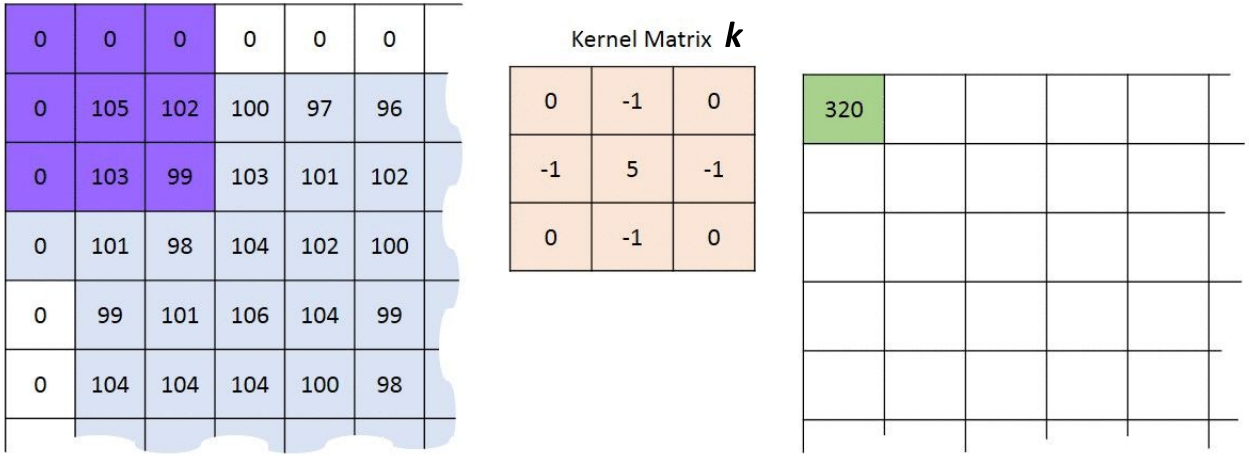
Convolutional neural networks (CNNs) leverage spatial information and, therefore, they are well suited for classifying images.



CNNs

Convolution is one of the main building blocks of a CNNs. Convolution is performed on the input data with the use of a **kernel** to then produce a **feature map**. The convolution of an image *x* with a kernel *k* is computed as:

$$(x * k)_{ij} = \sum_{pq} x_{i} + p,j+q$$



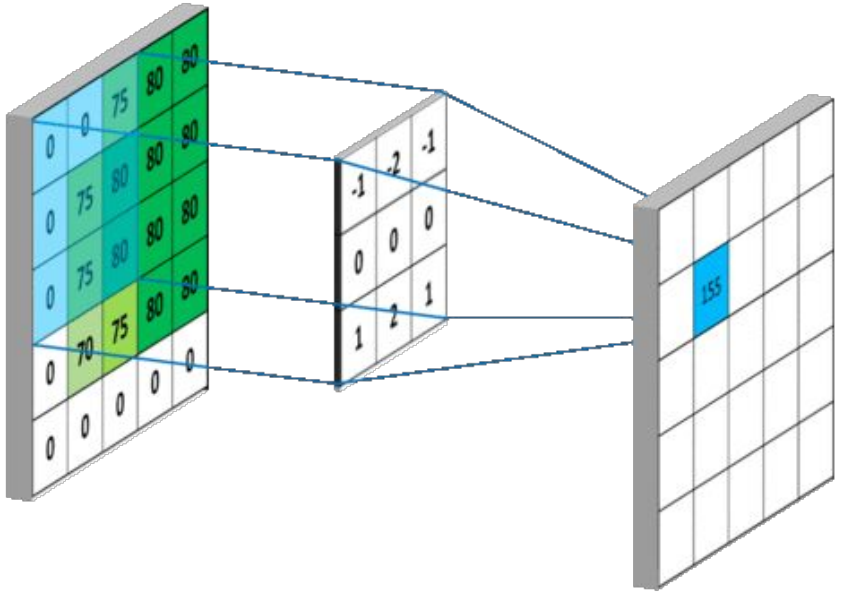
CNNs

CNNs are made of many simple units, whose behaviors are determined by their weights and biases.

Goal: use training data to train the network's weights and biases to maximize the performance of CNN.

CNNs use three main concepts:

- **Pooling**
- **Local receptive fields**
- **Shared weight**

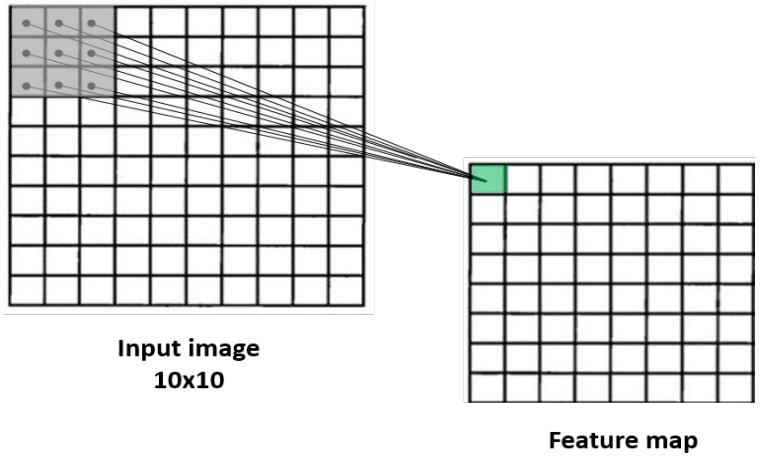


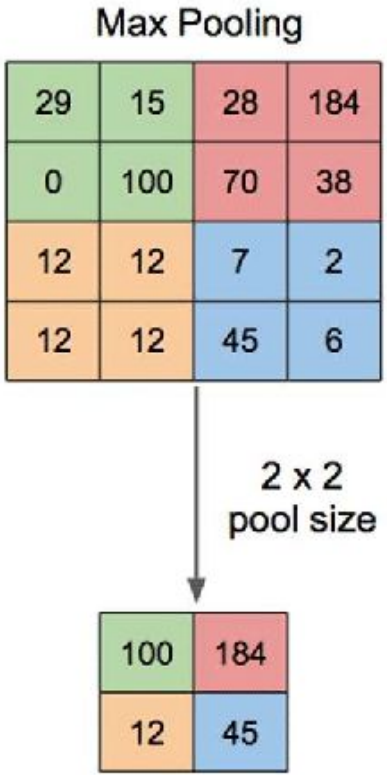
CNNs: Pooling Layers

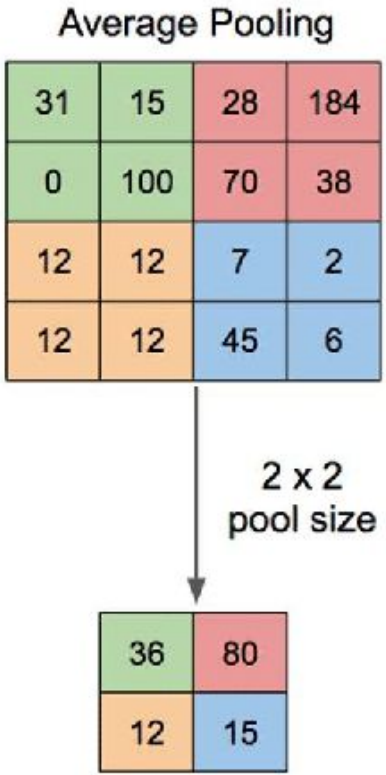
- Usually placed after convolutional layers
- Simplify the information output from the convolutional layer

The most common form of pooling is max-pooling which takes the maximum value in each window to

- decrease the feature map size
- keep the significant information.



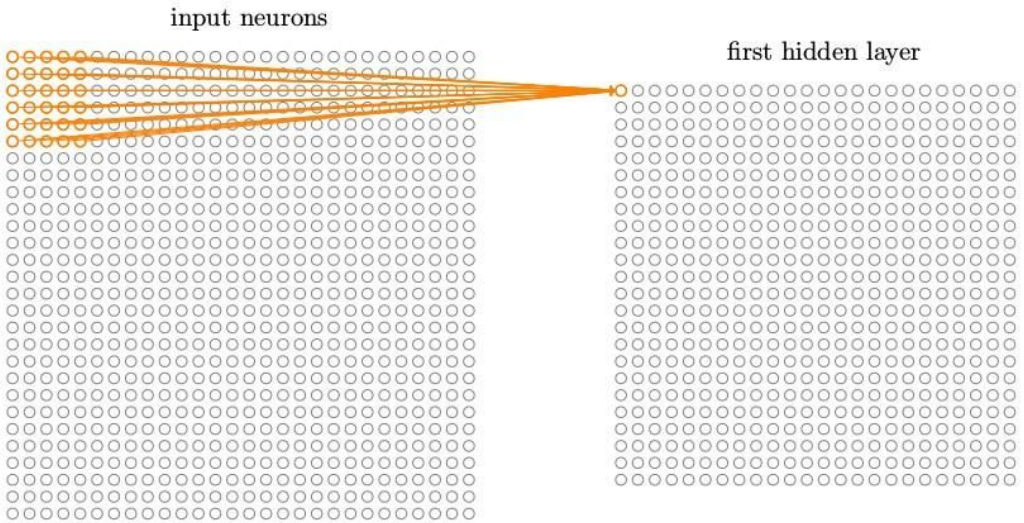




CNNs: Local receptive fields

All input neuron are just pixel intensities of an input image.

- Each neuron in the hidden layer is connected to a small region of input neurons. That region in the input image is called the local receptive field
- Each connection with local receptive field learns a weight
- Each hidden neuron learns to analyze its particular local receptive field



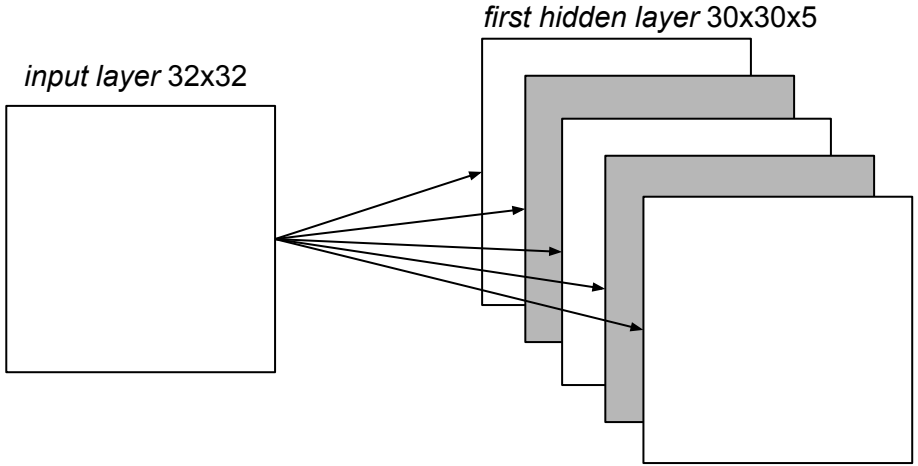
CNNs: Shared weight

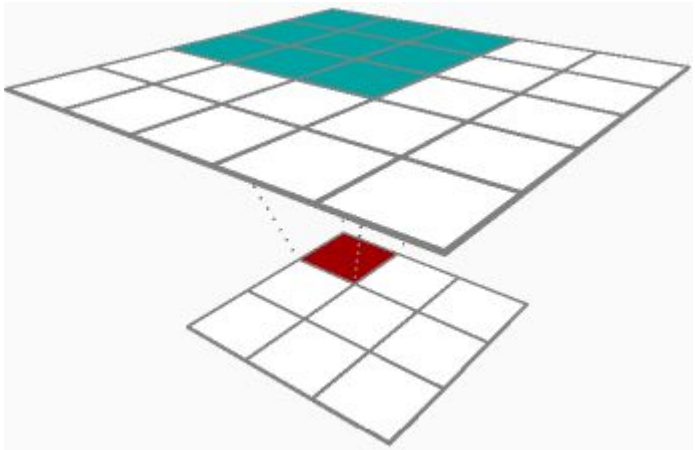
The same weights are used for more layers in the model: the same matrix elements may be updated multiple times during back propagation from varied gradients.

Forcing the **shared weights** among spatial dimensions:

- the number of parameters is drastically reduced and
- the convolution kernel is used as a learning framework

The mapping from the input to the hidden layer is called feature map and the shared bias are called kernel or filter.

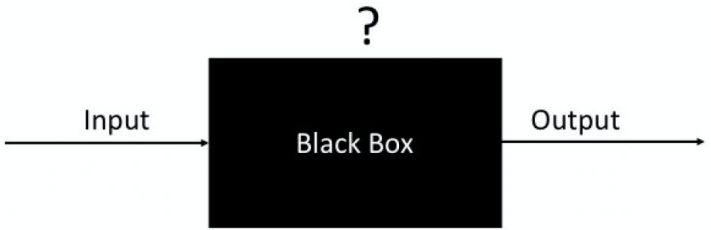




CNNs: Learning

Problem

CNN-based methods suffer from lack of sufficient explanation of proposed solutions and decisions. These techniques output complex information networks, which make the decision process difficult to explain (**“black box”** problem).



Goal

Open the black box to:

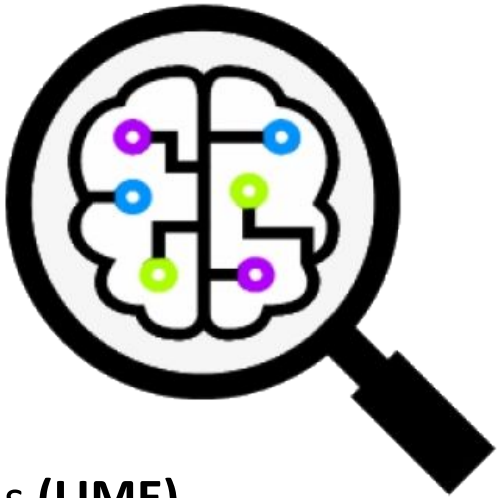
- provide an **explanation** and **interpretation** of the decision performed by a neural network during the training phase.
- validate the results

Solution

Explainable Artificial Intelligence (XAI)

Explainable Artificial Intelligence (XAI)

1. **Localization/Attention mechanisms**
   - Class Activation Map **(CAM)**
   - Gradient-Class Activation Map **(GradCAM)**



2. **Deconvolution**

3. **Feature importance and interactions**
   - Local Interpretable Model-Agnostic Explanations **(LIME)**

XAI: Localization/Attention mechanisms

- One of the most widely used in Deep Learning research in the last decade
- Used to analyze the model's capability and highlight the most relevant information used in the prediction and classification task

CAM

- Used to indicate the discriminative regions of an image used by a CNN to identify the category of the image
- A feature vector is created by computing and concatenating the averages of the activations of convolutional feature maps that are located just before the final output layer

XAI: CAM

Identify last convolutional layer

Let $f_k(x,y)$ be the activation map of unit k in the last convolutional layer at spatial location (x,y)

For each unit *k*, the Global Average Pooling (GAP) is computed as:

$$F_k = \sum_{x,y} f_k(x,y)$$

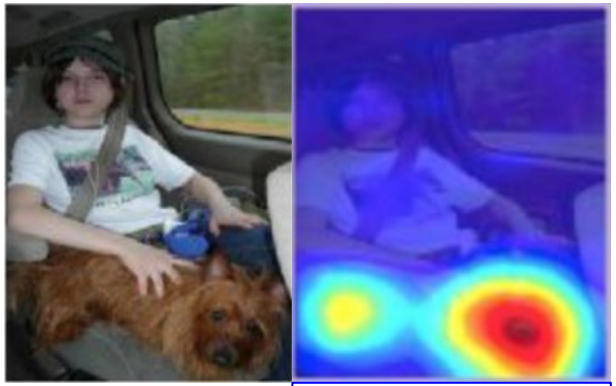
For a class c, the input to the softmax is computed as:

$$S_c = \sum_k w_k^c F_k$$

where $w_{(c,x)}$ is the weight corresponding to class c for unit k

the final equation for an activation map of class c is:

$$M_c(x,y) = \sum_k w_k^c f_k(x,y)$$



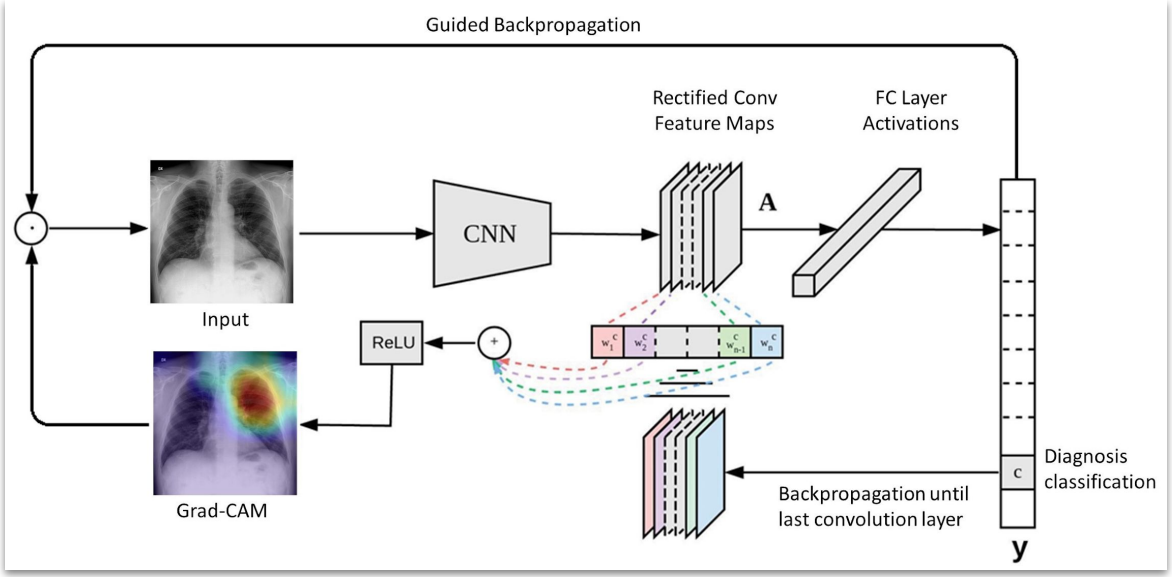
CAM result of Australian terrier classification

XAI: GradCAM

Generalization of CAM - (1) it can produce visual explanations for any CNN, regardless of its architecture, **(2)** does not need to have a GAP layer in architecture

GradCAM uses the gradient information flowing into the last convolutional layer of the CNN to:

- assign importance values to each neuron
- identify visual features in the input able to explain result process achieved during the classification

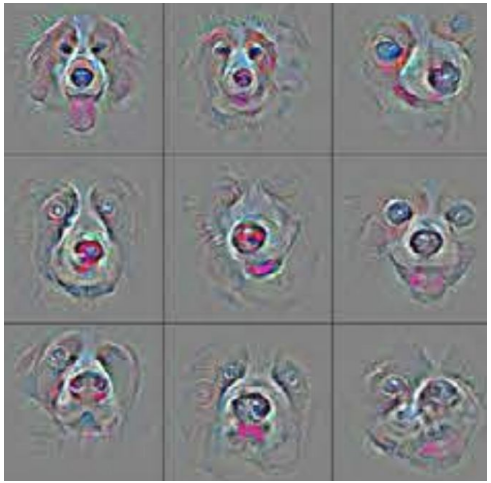


XAI: Deconvolution

Used to approximately project the activations of an intermediate hidden layer back to the input layer

- Projections can provide an insight into what details the hidden layer has captured from the input image
- To observe the evolution of features

It doesn't invert the CNN exactly: it only **projects** the pixels which favor the activation of a hidden layer.



https://towardsdatascience.com/explainable-neural-networks-recent-advancements-part-2-8cce67833ba

XAI: Feature importance and interactions

Analyze the level of contribution of the input features to the output prediction to provide interpretability to DL models

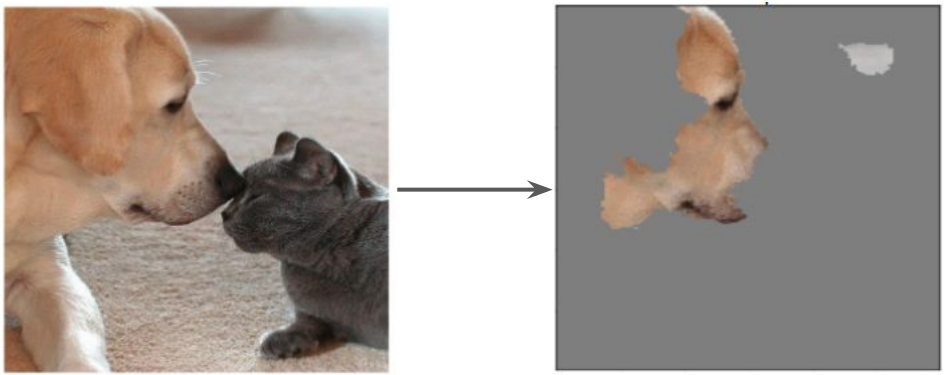
LIME

- It provides an explanation of a decision task
- It can be applied to any model in order to produce explanations for its predictions
- Based on perturbing the inputs (i.e., modifies a single data sample) and watching how doing so affects the model's outputs
    - Enabling to construct a picture of which inputs the model is focusing on and using to make its predictions
- The output of LIME is a list of explanations that reflects the contribution of each feature/pixels to the prediction of a data sample

XAI: LIME

Workflow

- Given a model trained on images to produce a probability distribution over the classes
- Perturb the input (e.g., hiding pixels by coloring them grey) and apply the model to see how the probabilities for the class change
- Use an interpretable (usually linear) model like a decision tree on the newly-created dataset of perturbed instances to extract the key features which explain the changes
    - The model is locally weighted — meaning that we care more about the perturbations that are most similar to the original image we were using
- Output the pixels with the greatest weights as our explanation



https://www.kdnuggets.com/2019/04/introduction-explainable-ai.html

DL-based approach to provide pneumonia diagnosis and explainability

Understanding Automatic COVID-19 Classification

The Novel Coronavirus (SARS-CoV-2) started to infect human individuals at the end of 2019.

- Rapidly caused a **pandemic**;
- Signs of infection include:
  - respiratory symptoms, fever, cough and dyspnea;
  - in more serious cases, Pneumonia, severe acute respiratory syndrome, multi-organ failure, and death.

Early and **automatic** diagnosis is crucial to:

- facilitate timely referral of patients to quarantine;
- perform rapid intubation of serious cases in specialized hospitals;
- better control the epidemic.

Diagnosis classification and interpretability

Obtain accurate diagnosis of diseases, as well as proper interpretations/explanations of decisions from artificial networks, is crucial in medicine and healthcare.

In particular, new methods are needed to:

- **discriminate** and **classify** pathological images from other/normal ones;
- **interpret** and **explain** the internal processes performed by the artificial networks during the classification.

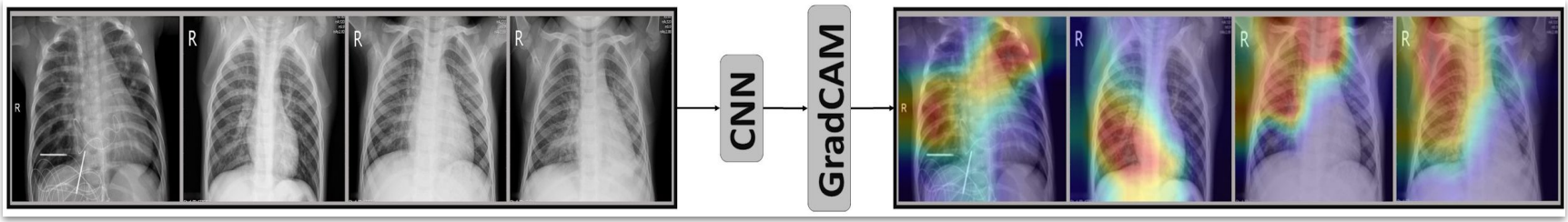
Challenges

The similarity of COVID-19 disease with other lung infections might impact on

- proper diagnoses
- treatment plans.

Proposed Approach

- Investigate the use of convolutional neural networks (CNNs) to perform multiple-disease classification from Chest X-ray images;
- Use **visual explanation** techniques to identify the mechanisms behind networks and therefore highlight discriminative areas;
- Analyze the correlation between these areas and classification accuracy, evaluating a possible performance worsening after their removal.



Methods: Deep Learning and Visual explanation

DenseNet-121, DenseNet-169 and DenseNet-201

- Network made of dense blocks, where for each layer the inputs are the feature maps of all the previous layers.
  - 121, 169, and 201 denote the depth of the ImageNet models.

Inception-v3

- Network composed of *inception modules* (i.e., well-designed convolution modules) that can both generate discriminatory features and reduce the number of parameters.

GradCAM

- identify visual features in the input able to explain result process achieves during the classification. In general, the warmer the color, the more important to the network are the highlighted features.

Experimental analysis

Evaluation

Recall (*Rec*) coefficient is used to minimize False Negatives (i.e., the disease is present but not identified)

$$Rec = \frac{TP}{TP+FN}$$

The **Area Under the Curve** (AUC), which is 1 for a perfect system, is a single measure to quantify this behavior.

Dataset

Experiment I

The datasets specifically include images of Chest X-ray of:

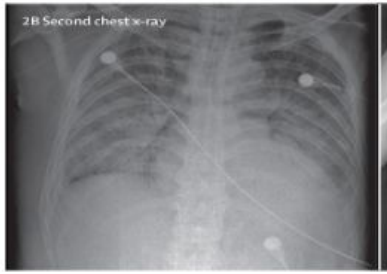
COVID-19

Viral Pneumonia

Streptococcus Pneumonia

Healthy patients

Experiment II

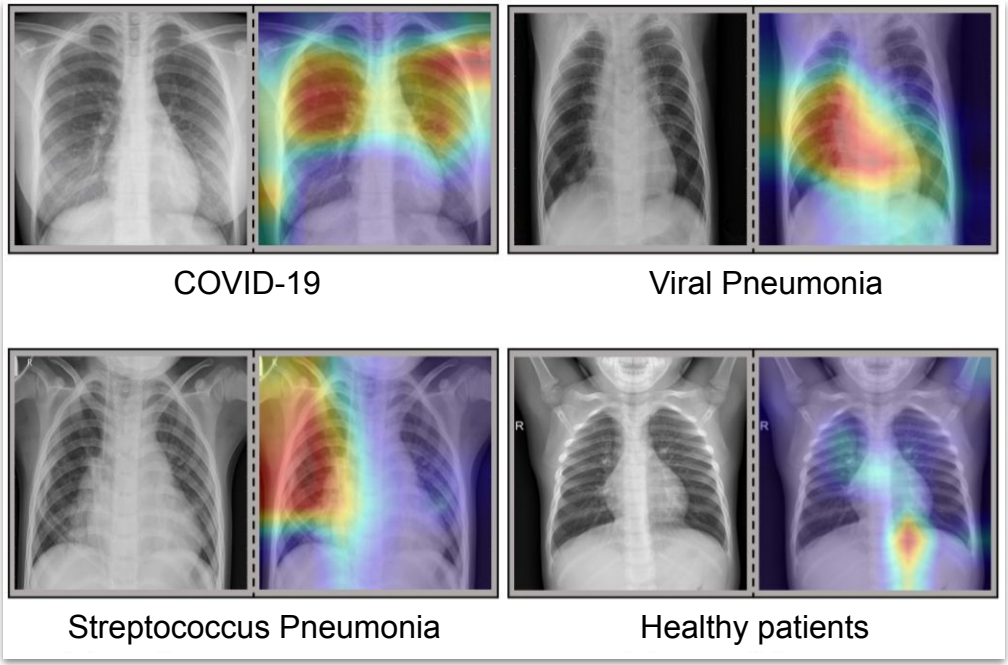


COVID-19

Tuberculosis

Healthy patients

Experiment I: Results Explainability



DenseNet 169 achieves the best performance on all datasets (Rec mean value of 0.84)

We selected and removed the 40% of highlighted elements by GradCAM

- Recall **decreases** by 10% on COVID-19, Viral Pneumonia, and Streptococcus Pneumonia dataset;
- p-value<0.05 for paired t-test computed before and after images cutting;
- no statistical changes are shown using dataset of Healthy patients.

GradCAM is able to identify the important features involved in the training process.
Classification performance is lower when highlighted regions are removed from the input images.

Experiment II: Results

Recall

DATASET	DenseNet 121	DenseNet 169	DenseNet 201
COVID-19	0.94 (0.02)	**0.95** (0.01)	0.90 (0.02)
TB Pneumonia	0.75 (0.03)	**0.84** (0.02)	0.66 (0.06)
Healthy patients	**0.94** (0.02)	0.88 (0.02)	0.87 (0.02)

AUC

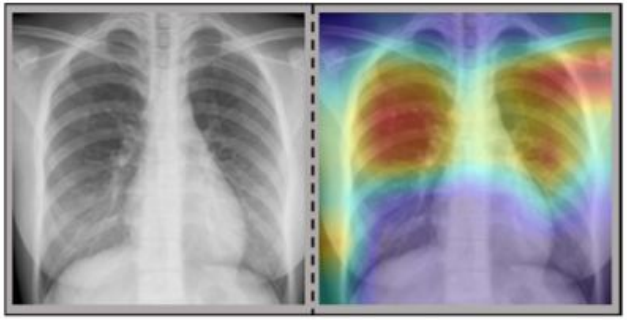
DATASET	DenseNet 121	DenseNet 169	DenseNet 201
COVID-19	0.96	**0.97**	0.92
TB Pneumonia	**0.95**	0.92	0.94
Healthy patients	0.95	**0.96**	0.93

DenseNet 169 achieves the best performance on COVID-19 dataset

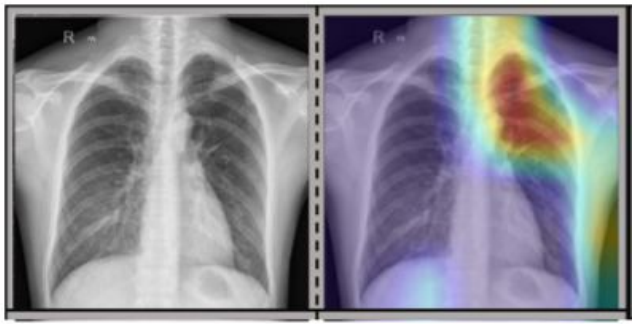
- Rec 0.95 and AUC 0.97

*CheXNet model as pre-trained weights

Experiment II: Explainability



(a) COVID-19



(b) TB Pneumonia

We selected and removed the 40% of highlighted elements by GradCAM

- Recall **decreases** by 5%
- p-value<0.05 for paired t-test computed before and after images cutting;

Assessing explanations from GradCAM

We compared the highlighting regions obtained by our approach to the labels assigned by expert clinicians to the areas of the images they considered relevant for the classification.

- e.g., in Figure (b) clinicians assigned the label "L upper"
- our approach identifies at least the 60% of the area suggested by clinicians

References

- *Zeiler MD et al.,* **Adaptive deconvolutional networks for mid and high level feature learning**. In International Conference on Computer Vision. 2011.
- *Zeiler, MD et al.,* **Visualizing and understanding convolutional networks.** European conference on computer vision. 2014.
- Nielsen M., ***Neural Network and Deep Learning*** *[http://neuralnetworksanddeeplearning.com].* 2016
- *Zhou, Bolei, et al.* **Learning deep features for discriminative localization.** Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- Chilamkurthy S., **Deep Learning Crash Course** [https://chsasank.github.io/deep-learning-crash-course-2.html]. 2017
- *Bruno P*., **Understanding Automatic COVID-19 Classification using Chest X-ray images**. 2020
- *Linardatos P et al.,* **Explainable AI: A Review of Machine Learning Interpretability Methods**. Entropy. 2021.

Thank you!