

CXResnet

Image classification for COVID-19 diagnosis

Zachary Wimpee

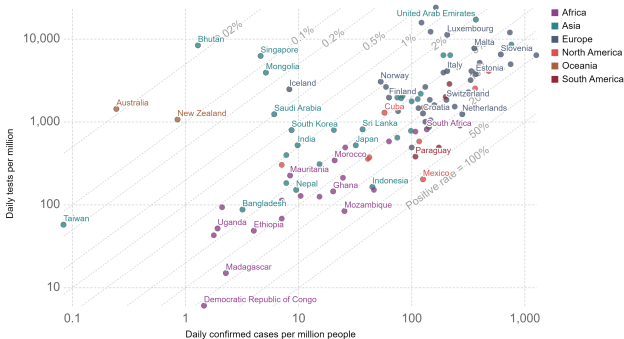
February 4, 2021

Motivating the problem

COVID-19: Daily tests vs. Daily new confirmed cases per million

The figures are given as a rolling 7-day average.

Our World
in Data



Source: Testing data from official sources collated by Our World in Data, confirmed cases from Johns Hopkins University CSSE

Note: Comparisons of testing data across countries are affected by differences in the way the data are reported. Daily data is interpolated for countries not reporting testing data on a daily basis. Details can be found at our Testing Dataset page.

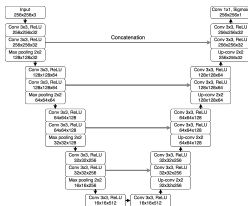
OurWorldInData.org/coronavirus • CC BY

Source: Our World in Data

Neural Networks and Computer Vision

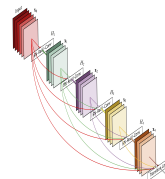
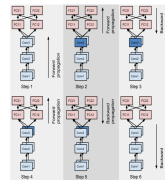


ResNet



U-Net

AlexNet

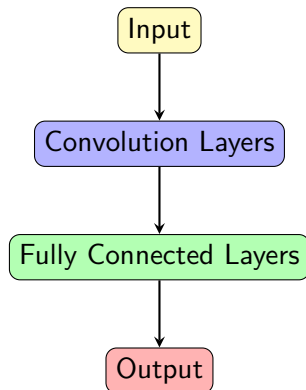


DenseNet

Creating the Blueprint for a New Architecture

I First consider a simple CNN architecture

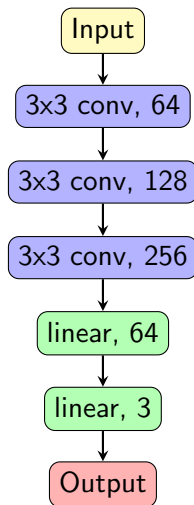
1. Several stacked convolution layers with structure:
 - i) 2d Convolution filter
 - ii) ReLU activation function
 - iii) Pooling layer
 - iv) Batch Normalization
2. Fully connected Linear layers
 - i) Sequence of linear layer, ReLU activation, and batch normalizations
 - ii) Final linear layer returns model output



Create the Blueprint for a New Architecture

II Next, consider how this network can be expanded

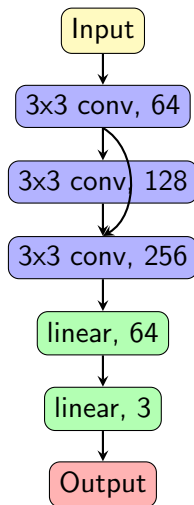
1. Increase the depth
 - Add more convolution and/or linear layers.
 - How many of each type of layer should be used?
2. Increase the width
 - Increase the number of convolutions



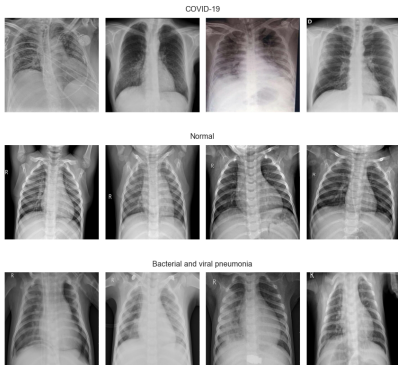
Create the Blueprint for a New Architecture

III Introduce residual connections between layers

1. Avoid encountering vanishing or exploding gradients
2. Effectively allows unneeded layers to be ignored



Examining the Dataset

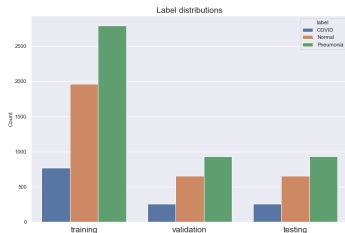


Dataset contains...

- 1281 COVID-19 X-rays
- 3270 Normal X-rays
- 4657 Pneumonia X-rays

Sample dataset images for each class

Splitting the Dataset



Distribution of image labels for each split

Dataset is split into 3 disjoint sets...

- 60% training
- 20% validation
- 20% testing

Choosing an Optimizer

Training script provides a choice between 2 optimizers...

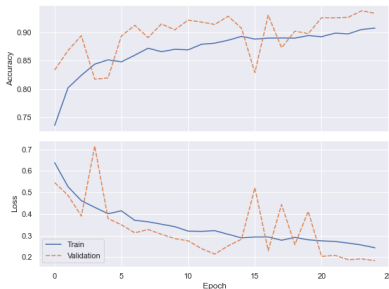
- SGD (Stochastic Gradient Descent)
 1. Avoids loss converging to a local minimum
 2. Slower and more computationally intensive than Adam
- Adam
 1. Adaptive learning rate increases training speed
 2. More likely for loss to converge to local minimum
 3. Default optimizer for training script

Choosing a Loss Function

- Cross Entropy Loss
 1. Combines LogSoftmax and NLLLoss functions
 2. Can define class weights to handle imbalanced classes

Tracking training progress

Training Accuracy and Loss over Epochs



Notice the 2 strange findings:

- i) Higher accuracy and smaller loss for validation steps.
- ii) Several pronounced fluctuations in both validation accuracy and loss between epochs