Skin Lesions Detection using Convolutional Neural Network

Artificial Intelligence & Machine Learning (623.504, 20W)

Bhuwan Sharma

bhuwansh@edu.aau.at
Alpen-Adria-Universität Klagenfurt, Austria

March 15, 2021

Overview

The Dataset

We used HAM10000 as our dataset which is a large collection of 10015 multi-source dermatoscopic images of common pigmented skin lesions provided by the Harvard University.

- **lesion_id**: there can be lesions with multiple images;
- **image_id**: corresponding image file name;
- **dx**: lesion type. There are seven types of lesions;
- **dx_type**: more than 50% of lesions are confirmed through histopathology (histo), the ground truth for the rest of the cases is either follow-up examination (follow_up), expert consensus (consensus), or confirmation by in-vivo confocal microscopy (confocal);
- **age**: age of the patient;
- **sex**: sex of the patient;
- **localization**: describes where the skin lesion is located.

Dataset Analysis

Lesion types	Lesion ID	Number of Images
Melanocytic nevi (nv)	5	6705
Melanoma (mel)	4	1113
Benign keratosislike lesions (bkl)	2	1099
Basal cell carcinoma (bcc)	1	514
Actinic keratoses (akiec)	0	327
Vascular lesions (vasc)	6	142
Dermatofibroma (df)	3	115

Table: Number of images per lesion type

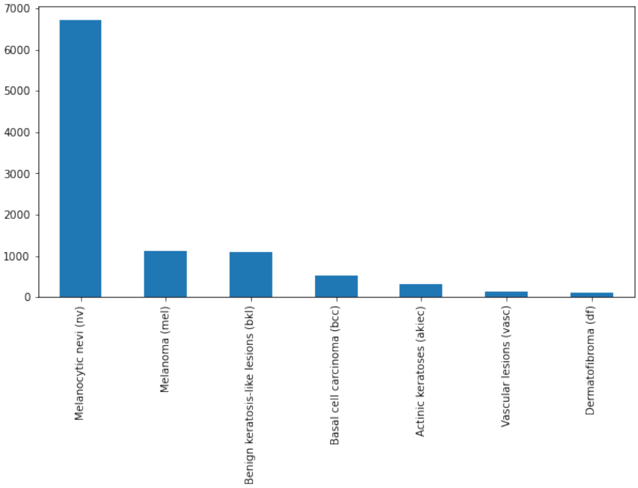


Figure: Distribution of images per lesion

Balancing the Imbalances

There are several ways to to overcome imbalances in a dataset, for instance, by collecting more data; resampling the dataset; create synthetic images of under-represented classes: SMOTE (Synthetic Minority Oversampling TEchnique), etc.

We chose **Median Frequency Balancing** where for each class we calculate the coefficient $\alpha_c = \mathit{median\_freq} \ / \ \mathit{freq}(c)$:

- **c**: class. In our case, each lesion type is a class;
- **freq(c)**: represents the division between the total number of pixels of class c across all images in the dataset and the total number of pixels in images where c is present;
- **median freq**: median frequency is the median of these frequencies.

Lesion types	**ID**	**N° images**	**MFB**
Melanocytic nevi (nv)	5	6705	0.076659
Melanoma (mel)	4	1113	0.461815
Benign keratosislike lesions (bkl)	2	1099	0.467698
Basal cell carcinoma (bcc)	1	514	1.000000
Actinic keratoses (akiec)	0	327	1.571865
Vascular lesions (vasc)	6	142	3.619718
Dermatofibroma (df)	3	115	4.469565

Table: Median Frequency Balancing per lesion type

Therefore, the dominant labels will be assigned with the lowest weight which balances the training process.

Artificial Neural Network (ANN)

Artificial Neural Network is a *synthetic* reproduction of humans brain's neural network to make computers learn and make decisions in a human-like manner.

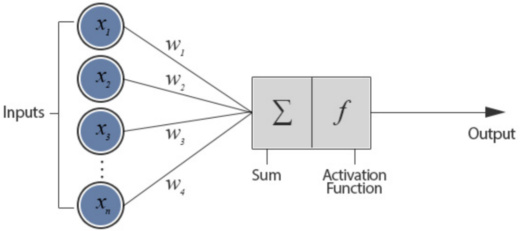


Figure: Perceptron, an artificial neuron

A perceptron takes binary inputs, $x_1, x_2, \ldots, x_n$ and produces a single binary output. Weights, $w_1, w_2, \ldots, w_n$, are used to set importance to the respective inputs.

The neuron's output, 0 or 1, is determined by whether the weighted sum $\sum_{i=1}^{n} x_i w_i$ is less than or greater than some threshold value.

The weighted inputs are added together and passed into the activation function, which decides when and what to output.

A **Deep Neural Network** is an ANN having an input layer, multiple hidden layers, and an output layer. It uses multiple layers to progressively extract higher-level features from the raw input. For instance, in image processing, lower layers may identify edges, while higher layers may identify more relevant concepts such as digits or letters, or faces.

Convolutional Neural Networks (CNNs)

A Convolutional Neural Network (**ConvNet/CNN**) is a Deep Learning algorithm, commonly applied in computer vision, which takes an image as input, assign importance (learnable weights and biases) to various aspects/objects in the image, and distinguish them from each other.

CNNs' leverage on the fact that nearby pixels are more strongly related than distant ones.

The CNN process begins with convolution and pooling, breaking down the image into features, and analyzing them independently. The result of this process is fed into a fully connected neural network structure that drives the final classification decision.

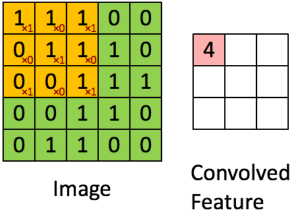


Figure: Example of a Filter/Kernel applied to a 2D input to create a Feature Map

In the context of a convolutional neural network, a convolution is a linear operation that involves the multiplication (dot/scalar product) of a set of weights with the input.
In the figure we can see the pixel values of an image, and we have a **filter** $\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$, which is a set of weights, and the filter is systematically applied to the input data to create a **feature map**.

AlexNet

AlexNet is a CNN deep neural network that was developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton in 2012. It won the ILSVRC 2012 competition.

AlexNet has 60 million parameters, 650,000 neurons and it consists of 8 layers:

- 5 Convolutional Layers
- 2 Fully Connected Layers
- 1 Softmax Layer (output)

Each layer is followed by the rectified linear activation function (**ReLU**) except for the last one (output layer) which uses **softmax**.

Max Pooling is applied after the 1st, 2nd and 5th convolutional layer.

The fully connected layers have 4096 neurons each and the second fully connected layer is fed into a softmax classifier. **Dropout** (a technique used to prevent a model from overfitting) is applied in the first two fully connected layers.

AlexNet Architecture

Figure: AlexNet Architecture

- **Convolution Layer** (CVL): applies a filter to an input image repeatedly (to detect a certain feature in the image), creating the feature map as result.
- **Subsampling Layer (Max Pooling Layer)** (MP): reduces the spatial size of the Convolved Feature (a.k.a Kernel) $\rightarrow$ decreases the computational power required to process the data.
- **Fully Connected Layer** (FC): takes the results of the Convolution or Pooling layers and use them to classify the image into a label (in a simple classification example).
- **Softmax Layer** (SM): used to get the probabilities of the input being in a particular class and do the final classification.

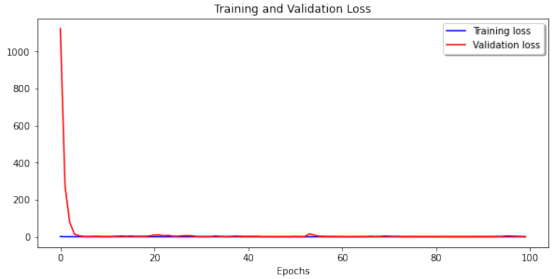
Training, Validation and Testing

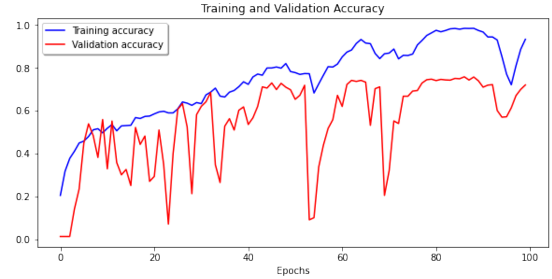
The dataset has been splitted into three parts, training, validation, and test dataset to avoid overfitting and improve the generalization of the model when training the neural network:

- **Training dataset** (60%): has been used to train the model by matching each image to the correct skin lesion;
- **Validation dataset** (20%): useful to check how well we trained our model;
- **Testing dataset** (20%): the moment of truth; useful to make an evaluation of the previously trained classifier. This will also tell us if we are under or overfitting our model.

We have to make reduce as much as possible the imbalances by distributing an equal percentage of every type of skin lesion. For instance, if we have 100 images of skin lesion type A, then we have to use 60 images of type A in the training phase, 20 for validation, and 20 for testing: **stratified sampling**.

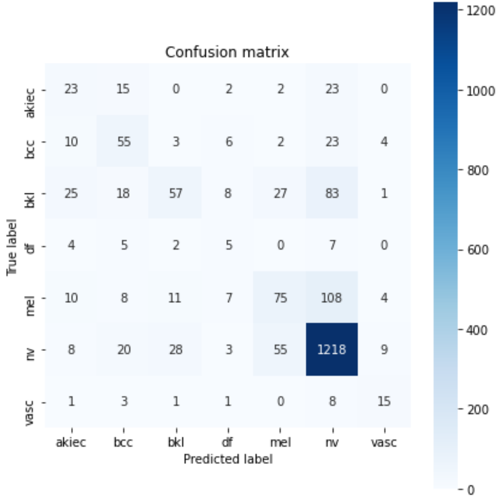
Results





- In the first plot we see that both training and validation losses are decreasing smoothly, meaning that the model is able to generalize on the validation set.
- The sudden drops in validation accuracy, in the second figure, occur due to the batch training. Another point to mention here is that we stopped our training in an unfortunate moment when the validation accuracy dropped. We should have used an epoch around 80-85 when both the curves reached a plateau.

Results Continued



	precision	recall	f1-score	support
akiec	0.28	0.35	0.32	65
bcc	0.44	0.53	0.48	103
bkl	0.56	0.26	0.36	219
df	0.16	0.22	0.18	23
mel	0.47	0.34	0.39	223
nv	0.83	0.91	0.87	1341
vasc	0.45	0.52	0.48	29
accuracy			0.72	2003
macro avg	0.46	0.45	0.44	2003
weighted avg	0.71	0.72	0.71	2003

Figure: Classification report

Conclusions

- Our training was not satisfactory at all because some label accuracies were even below 50%.
- Training a Neural Network is not an easy task. It involves a lot of challenges, starting from finding the dataset, analyzing it, balancing it, finding the right ML to use, and the right number of epochs.
- We have learned why data analysis is important and applied some techniques to overcome the imbalances in our dataset.
- Median Frequency Balancing did not contribute enough to mitigate this imbalance. Thus, our model became more accurate in classifying the lesion having the highest number of images, *nv* lesions, compared to other lesion types.

In future works, we could use image augmentation techniques to increase the number of images for the underrepresented lesion types. We could also use Synthetic Minority Oversampling Technique (SMOTE) as an antidote to imbalances. Replacing AlexNet with a more complex Neural Network could also be a solution to get better results.

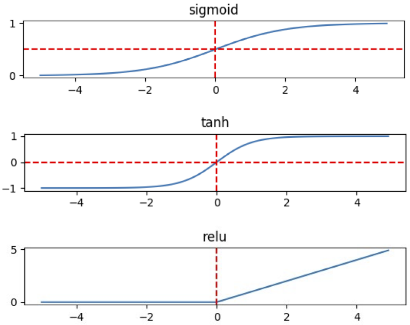
References

- https://github.com/niyazed/Dermatology-Image-Classification/blob/master/process_and_train.ipynb
- https://github.com/biagiom/skin-lesions-classifier/blob/master/skin_lesions_classifier.ipynb
- https://medium.com/miccai-educational-initiative/skin-cancer-image-classification-an-educational-guide-2a043a1beb59
- https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/
- https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/
- https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
- https://towardsdatascience.com/simple-introduction-to-convolutional-neural-networks-cdf8d3077bac
- https://towardsdatascience.com/implementing-alexnet-cnn-architecture-using-tensorflow-2-0-and-keras-2113e090ad98
- https://towardsdatascience.com/understanding-alexnet-a-detailed-walkthrough-
- https://learnopencv.com/understanding-alexnet/
- https://towardsdatascience.com/understanding-alexnet-a-detailed-walkthrough-20cd68a490aa
- https://medium.com/x8-the-ai-community/explaining-alexnet-convolutional-neural-network-854df45613aa
- https://wiki2.org/en/AlexNet
- https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

The End

Questions?

Backup - Sigmoid, Tanh and ReLU Activation Functions



The logistic activation function (**sigmoid**) was popular through the early 1990s. The input to the function is transformed into a value between 0.0 and 1.0.

The hyperbolic tangent function (**tanh**) outputs values between -1.0 and 1.0. Popular between 1990 - 2000. The *tanh* function was preferred over the *sigmoid* activation function as models that used it were easier to train and often had better predictive performance.

A general problem with both the sigmoid and tanh functions is that they saturate → they are sensitive only to the changes around their mid-point of their input[1].

Solution

The rectified linear activation function (**ReLU**) returns the value provided as input directly, or the value 0.0 if the input is 0.0 or less.