OPTIMIZING A CONVOLUTIONAL NEURAL NETWORK TO DETECT MELANOMA IN IMAGES

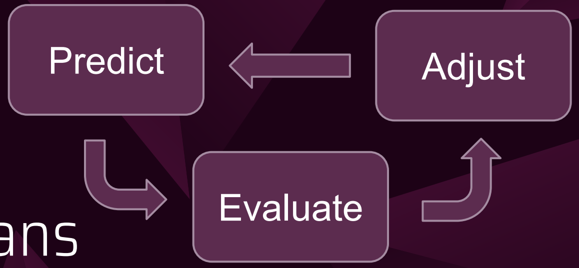
MA Capstone Final Presentation 2021

Christian Shadis

INTRODUCTION AND BACKGROUND

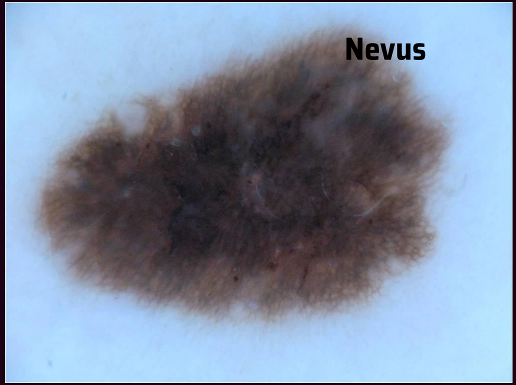
WHAT IS MACHINE LEARNING?

- Iterative function to minimize error
  - Predict: Multiply some input by some weight
  - Evaluate: Calculate **loss** (prediction error)
  - Adjust: Use loss to adjust weight
  - Repeat
- Learns a pattern from data that humans otherwise couldn't



MELANOMA

- Most deadly type of skin cancer
- Looks very similar to seborrheic keratosis and other harmless nevi



APPLIED ML: IMAGE RECOGNITION

- Images are represented numerically
  - Mathematical operations allowed
- Grids of pixels transformed into a single prediction vector
- Trained on many images, each one labelled as a distinct class
  - I.e. every image in this dataset is labelled either melanoma, nevus, or seborrheic keratosis
- Model 'learns' patterns in the data with many iterations (epochs)
- Model's ability to predict the class of an input image increases

GOAL

Develop a convolutional neural network that can detect the presence of melanoma in images of skin lesions at an accuracy rate of at least 70%

CONVOLUTIONAL NEURAL NETWORKS

CONVOLUTIONAL LAYERS: BASICS

- Receives input
- Iterates small kernel ('window') over image
    - Computes sum of products of corresponding cells

Image

0.604	0.545	0.553	0.62	0.601	0.522
0.639	0.573	0.541	0.573	0.616	0.6
0.659	0.631	0.573	0.541	0.589	0.635
0.651	0.675	0.643	0.561	0.537	0.584
0.639	0.675	0.678	0.616	0.545	0.541
0.627	0.639	0.659	0.655	0.612	0.561

Kernel

0.202	-0.757	-0.667
-0.118	-0.595	-0.482
-0.51	0.132	-0.492

Result

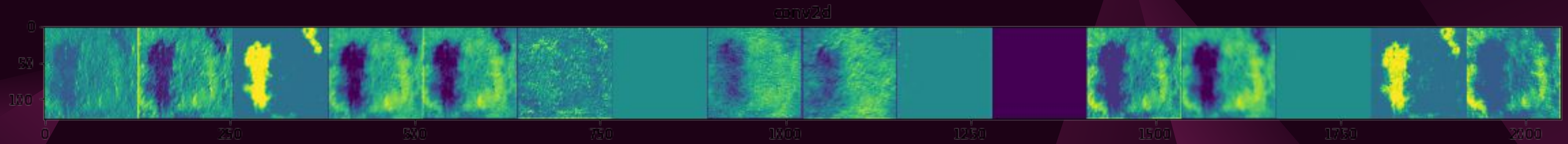
-1.87122	-1.9	-1.971	-1.912
-1.954	-1.888	-1.927	-1.974
-2.085	-1.958	-1.828	-1.936
-2.172	-2.066	-1.912	-1.87

(0.604 *0.202) + (0.545*-0.757) +
(0.553*-0.667)+(0.639*-0.118) +
(0.639*-0.118)+(0.573*-0.595) +
(0.541 * -0.482)+(0.659*-0.51) +
(0.631 * 0.132)+(0.573*-0.492)

= -1.87122

CONVOLUTIONAL LAYERS: FEATURE MAPS

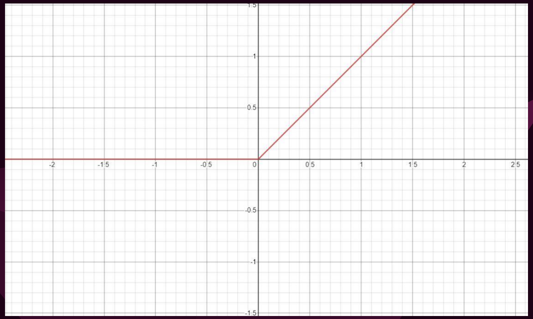
- For each different filter(kernel) applied, a “feature map” is generated
- Ex: a single image passed through a 16-filter Convolutional Layer might generate the following 16 feature maps:
  - As shown, the convolutional layer extracts high-level patterns in the images
    - For example, the round, darkened center of the lesion in the input image
  - 16 different images for 16 different ‘kernels’



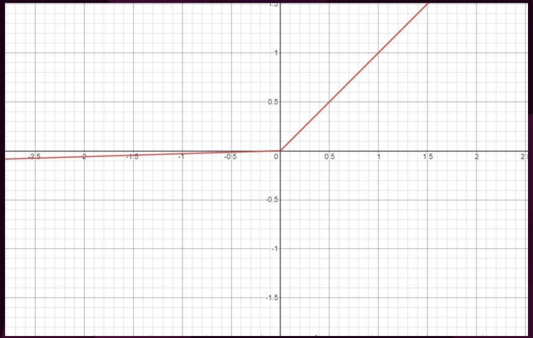
CONVOLUTION LAYERS: ACTIVATION FUNCTIONS

- Non-linear function applied to every neuron
  - Emphasizes relationship between similar pixels
- Properties of Activation Functions:
  - Differentiable
  - Symmetry about x=0 preferred
  - Computationally inexpensive
- Common examples of Activation Functions:
  - ReLU - sets negative input values to 0
  - LeakyReLU - sets negative input values to very small value
  - Sigmoid (outdated for inner layers)
  - Tanh

ReLU

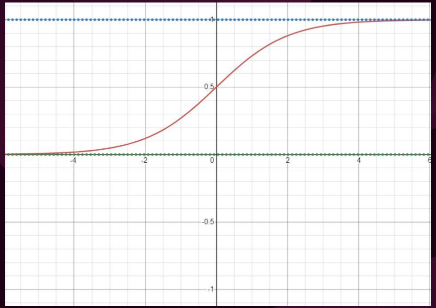


LeakyReLU (alpha=0.03)



CONVOLUTIONAL NEURAL NETWORK STRUCTURE

- **Input Layer:** Image represented as 256 x 256 x 3 image
- **Convolution Layers:** Extract patterns and important features
- **Max-pool Layers:** Retain most impactful features
- **Dropout Layers:** Randomly deactivate neurons (reduce overfitting)
- **Flatten Layers:** Reduces dimensionality from 3 to 1
- **Dense Layers:** traditional neural network structure
  - Each pixel's relationship with one another examined
- **Sigmoid Dense Layer**: calculate probability of match for each class
  - Transforms data to interval [0, 1] and select corresponding class
  - Ex. [0.01, 0.99] would select class 1 (melanoma) as the prediction for the input
- **Output Layer:** Vector containing predicted class of image
  - Ex. [0, 1]

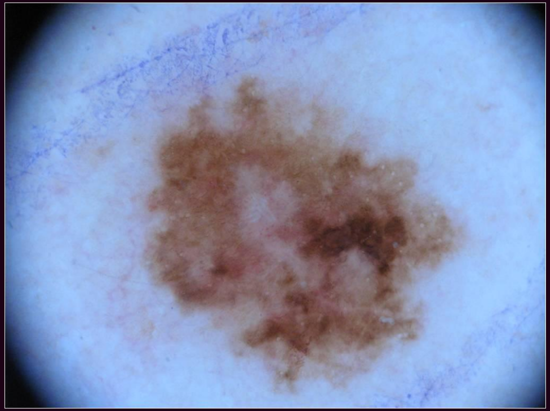


DATA PROCESSING

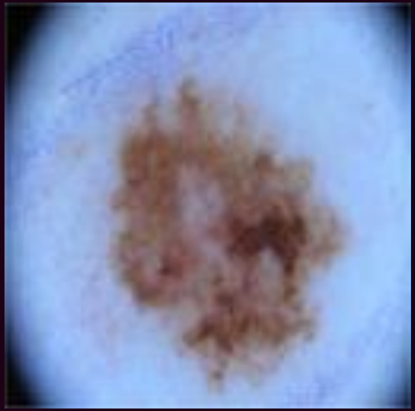
TRAINING DATASET

- Dataset of 2750 images
    - Augmented to create set of over 100,000 images
    - Originally high-definition, re-sized to 256 x 256

Original Image



Resized Image



Augmented Image

DATA SELECTION DETAILS

1. Validation set selected and separated *before augmentation* (n = 200)
2. From pool of 100,000 images, select a *random sample* of images from each class
    a. Training Set n = 2000
        i. 1,000 melanoma, 1,000 non-melanoma
3. 90/10 train/validation split
    a. Wide split due to data limitations
    b. Training set is then split into 80/20 train/test
        i. 20% of training images are used to evaluate model accuracy *each iteration*
    c. Model training occurs
    d. 10% of original data that the model **has not seen** is now used to evaluate accuracy
        i. This assures the model can generalize and is not simply ‘memorizing’ the data

REFACTORING OF CLASSIFICATION

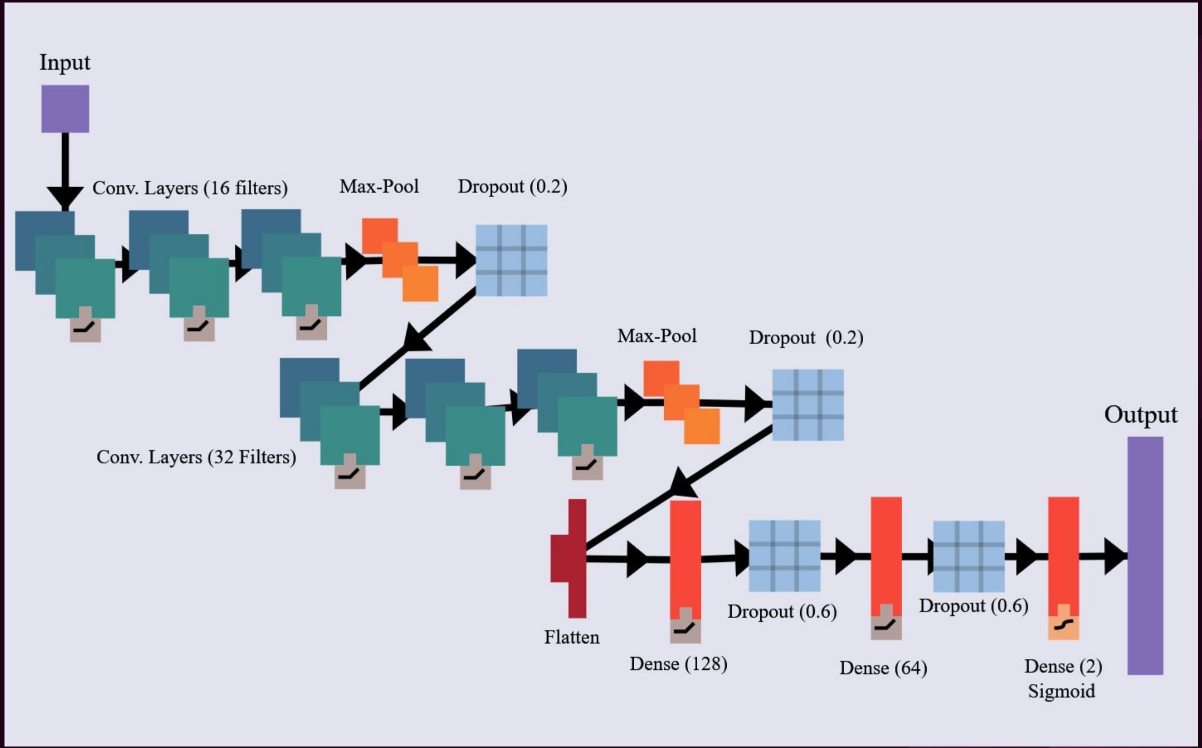
In an effort to address poor results, I reframed the question from classifying between 3 lesions to just detecting the presence of melanoma, which was the original project goal.

CATEGORICAL → BINARY

"Melanoma, Seborrheic Keratosis, or Nevus?" → "Melanoma or not?"

RESULTS

FINAL MODEL STRUCTURE



FINAL MODEL SPECS AND RESULTS

- Total Layers: 16
- Conv Layers: 6
- Dense Layers: 3
- Activation Function: LeakyReLU (0.3)
- Dropouts: 4
- Optimizer LR: 0.00001
- Batch Size: 32

Accuracy: 62.5%

CONCLUSION

- Computational limitations cripple the dataset
  - Classification would likely be easier with HD images than downsampled, blurry ones
- Improved a model from a 38%-accurate categorical classifier to a 62%-accurate binary classifier

REFERENCES

[1] J. Koushik (2016). Understanding Convolutional Neural Networks. ArXiv. abs/1605.09081.

[2] I. Goodfellow and Y. Bengio, and A. Courville (2016). Deep Learning. MIT Press.

[3] A. Trask (2019). Grokking Deep Learning. Manning Publications Co..

[4] Glorot, X., Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (pp. 249–256). PMLR.

[5] Santori, Pablo Lopez. (May 2020). Melanoma Detection Dataset, Version 1. Retrieved 10/14/2021 from https://www.kaggle.com/wanderdust/skin-lesion-analysistoward-melanoma-detection/.

[6] Mishkin, D., Sergievskiy, N., Matas, J. (2016). Systematic evaluation of convolution neural network advances on the Imagenet. Computer Vision and Image Understanding, 161. https://doi.org/10.1016/j.cviu.2017.05.007

[7] Davis, L. E., Shalin, S. C., Tackett, A. J. (2019). Current state of melanoma diagnosis and treatment. Cancer biology therapy, 20(11), 1366–1379. https://doi.org/10.1080/15384047.2019.1640032

[8] Zheng, Alice (2015). Evaluating Machine Learning Models. O'Reilly Media, Inc..

[9] Yedlin, M., Jafari, M. (2020, February 4). Binary cross-entropy. YouTube. Retrieved December 15, 2021, from https://www.youtube.com/watch?v=wpPkDSMzdKo