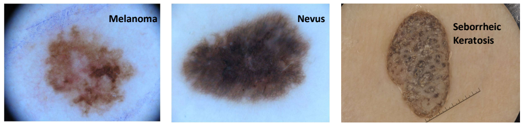
Optimizing a Convolutional Neural Network to Detect Melanoma in Images

Christian Shadis

Goal: Implement machine learning knowledge to tune a model that detects melanoma in images

The Problem

Melanoma is a rare form of skin cancer that accounts for most skin-cancer-related deaths. Given a set of approximately 2700 images of benign and malicious skin lesions, can we develop a convolutional neural network to detect melanoma with 70% accuracy?



Machine Learning in a Nutshell

Given:
- A random input **X** in *n* dimensions
- A random output **Y**
- Loss (error) function L(**Y**, *f*(**X**)), where *f*(**X**) is the model's output

Want:
- A mapping from **X -> Y** minimizing the **expected value** of the loss function

$$\hat{g} = \arg\min \mathrm{E}[\mathrm{L}(\mathrm{Y}, \mathrm{g}(\mathrm{X}))]$$

CNNs: Convolutional Layers

- Foundation of Convolutional Neural Networks
  - What separates CNNs from other Machine Learning models
- Purpose: extract high-level features from an image
- Method:
  - Iterate a small 'window' over our image
    - Window (a.k.a. 'filter', 'kernel') has a (*random*) decimal value at each pixel
    - Each pixel multiplied with kernel value corresponding, add all products
    - Window shifts to next pixel

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

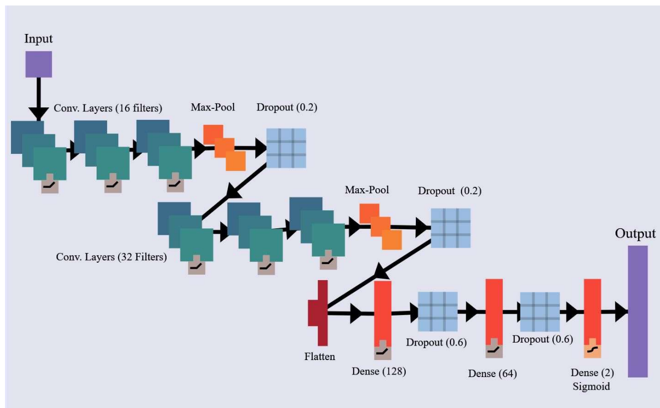
Values in the kernel are determined with the Xavier Glorot initialization
- Random distribution dependent on the size of multiple layers in the network

$$W_{i,j} \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}\right]$$

CNNs: Other Layers

- **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
- **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]

Final Model Structure



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-analysis-toward-melanoma-detection/.

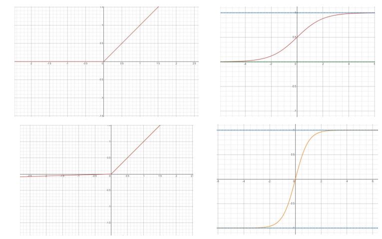
[6] Mishkin, D., Sergievskiy, N., Matas, J. (2016). Systematic evaluation of convolution-neural network advances on the Imagenet. Computer Vision and Image Understanding, 161. https://doi.org/10.1016/j.cviu.2017.05.007

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

CNNs: Parameters

- **Optimizer Learning Rate**: Extremity of weight adjustments per iteration
- **Batch Size**: How many images are fed to the model simultaneously
- **Activation Function**: Functions that find nonlinear patterns in data
- **Epochs**: Number of training iterations
- **Sample Size**: Number of images in training dataset



Activation Functions: ReLU, LeakyReLU, Sigmoid, Tanh

Results

Summary:

- ***Achieved 60 – 65% accuracy consistently***
- Was **not** able to develop model to reliably detect melanoma in an image
- **Was** able to develop model to classify between melanoma and non-melanoma images with 60 – 65% accuracy

Final Model Specs:

- **Total Layers: 16**
- **Conv Layers: 6**
- **Dense Layers: 3**
- **Activation Function: LeakyReLU (0.03)**
- **Dropouts: 4**
- **Optimizer LR: ?????**
- **Batch Size: ?????????**