Colorectal Tumor Classification with Neural Networks

Katrina Wheeler

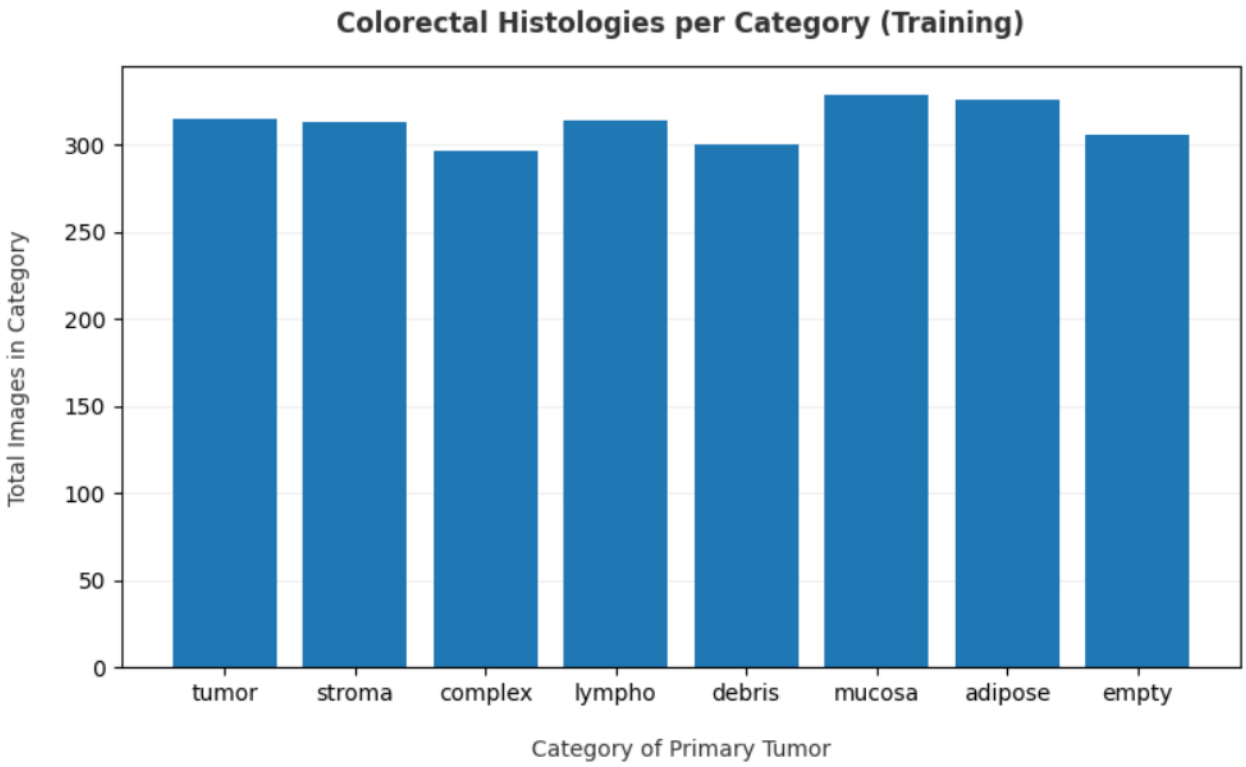
Department of Engineering & Computer Science: University of Denver

COMP 4449: Data Science Capstone

November 17, 2023

Diagnosis of colorectal cancer in people under the age of 55 has increased from 11% in 1995 to 20% in 2023 (Masciadrelli.) This cancer has a mortality rate of 33.33% (Ghosh). Diagnosis of colorectal cancer involves manual analysis that can be 'vulnerable to observational variations that reduce the overall efficiency of cancer grading' (Ghosh). Utilizing neural networks to assist in the identification of colorectal cancer can assist in reducing these observational variations by creating a more standardized classification system. In this project, I am interested in utilizing TensorFlow and neural networks to classify images from textures in colorectal cancer histology. These images represent primary tumors from the pathology archive of the Institute of Pathology at the University Medical Center in Mannheim Germany. The data is labeled with 7 different classifications, plus some blank images which are labeled as 'blank.' The categories include: lympho, stroma, debris, adipose, tumor, mucosa, and complex. The goal of my analysis is to use a neural network within TensorFlow to classify the images in this dataset into one of the 7 categories, and tuning the model to improve accuracy as much as possible.

The dataset can be found within the TensorFlow Datasets collection, which is a series of ready to use data sets to use within the TensorFlow framework. When loading in the dataset, I split the data into training, validation, and testing sets to utilize while building my neural network model. The dataset features 5,000 images of colorectal histologies, with the corresponding label representing the classification. The label feature is of type string, with a number representing the category and a label for that category (for example '6 (adipose)'. Plotting the count of each category for the training, validation, and testing sets, I found that the categories were somewhat evenly distributed, as shown by the image below:



This image is representative of the training dataset, but the validation and testing data sets were similarly distributed. For the model, I programmed a function to one hot encode the labels that were previously of type ‘string’, converting these to just a number, rather than a number and a label.

I will be using Keras to assist in training the model. Keras is a library in Python that is often used in neural networks to assist in training neural network models. I will first be using Keras to scale the images within the dataset. The images within the colorectal histology dataset are a 150x150x3 RBG image. I will be using Keras to scale the RGB channels to between 0-1. I will be utilizing a layer to scale the dataset, so that the same scale can be easily used for each different dataset.

Neural networks fall under the umbrella of machine learning, and are composed of an input layer, hidden layers, and an output layer. These layers contain nodes, which have associated weights and thresholds (IBM). Data is passed between the nodes when the input data is above the specific threshold, which activates the next node by moving to the next layer. There are different types of neural networks that are used for different situations. For this image classification model, I will be using a convolutional neural network (CNN). This type of neural network is not only inspired by the neural systems of humans, but the optical system as well (Boesch). CNNs are able to train and learn from data without much human assistance, and only need a minimal amount of preprocessing (Boesch). They use various principles of linear algebra to identify patterns within an image (IBM). CNNs are made up of three types of layers: convolutional layer, pooling layer, and fully-connected (FC) layer (IBM). Increasing the amount of layers can increase the complexity of the model (IBM). These layers start out by focusing on general features of the input data, such as colors and edges, and slowly move onto larger elements, ultimately painting a full picture of the input data (IBM). A CNN can have anywhere between 3 and 150+ layers (Boesch). A CNN model is the best option for the colorectal classification because it can process images, and is able to classify those images into multiple different categories.

The model starts with the convolutional layer, which requires input data, a filter, and a feature map (IBM). Convolution is an ‘element-wise multiplication of two matrices followed by a sum’ (Rosebrock). For image classification, the input data for the convolutional layer, the image, has three dimensions: the height, the width and the depth (which represents the RGB, or color model, of the image). The image itself is essentially a matrix. The convolution is a matrix that transforms the image matrix via matrix transformations. The layer will also use a kernel, or

filter, which will check if a specific feature is present in the image based on a small area of the image, moving until it goes through the entire image. Once complete, the output of this process is a feature map (IBM). Additional convolutional layers can be added, which creates a hierarchical structure with smaller pieces of the image at the bottom, and the full picture at the top.

The pooling layer reduces the dimensionality by reducing the number of parameters in the input, which can "help reduce complexity, improve efficiency, and limit risk of overfitting" (IBM). Finally, there is the fully-connected layer, where each node from the output layer connects directly to the nodes in the previous layers (IBM). This layer performs the classification based on the feature extraction conducted by the previous layers. This layer utilizes the activation functions to produce a probability for each classification category.

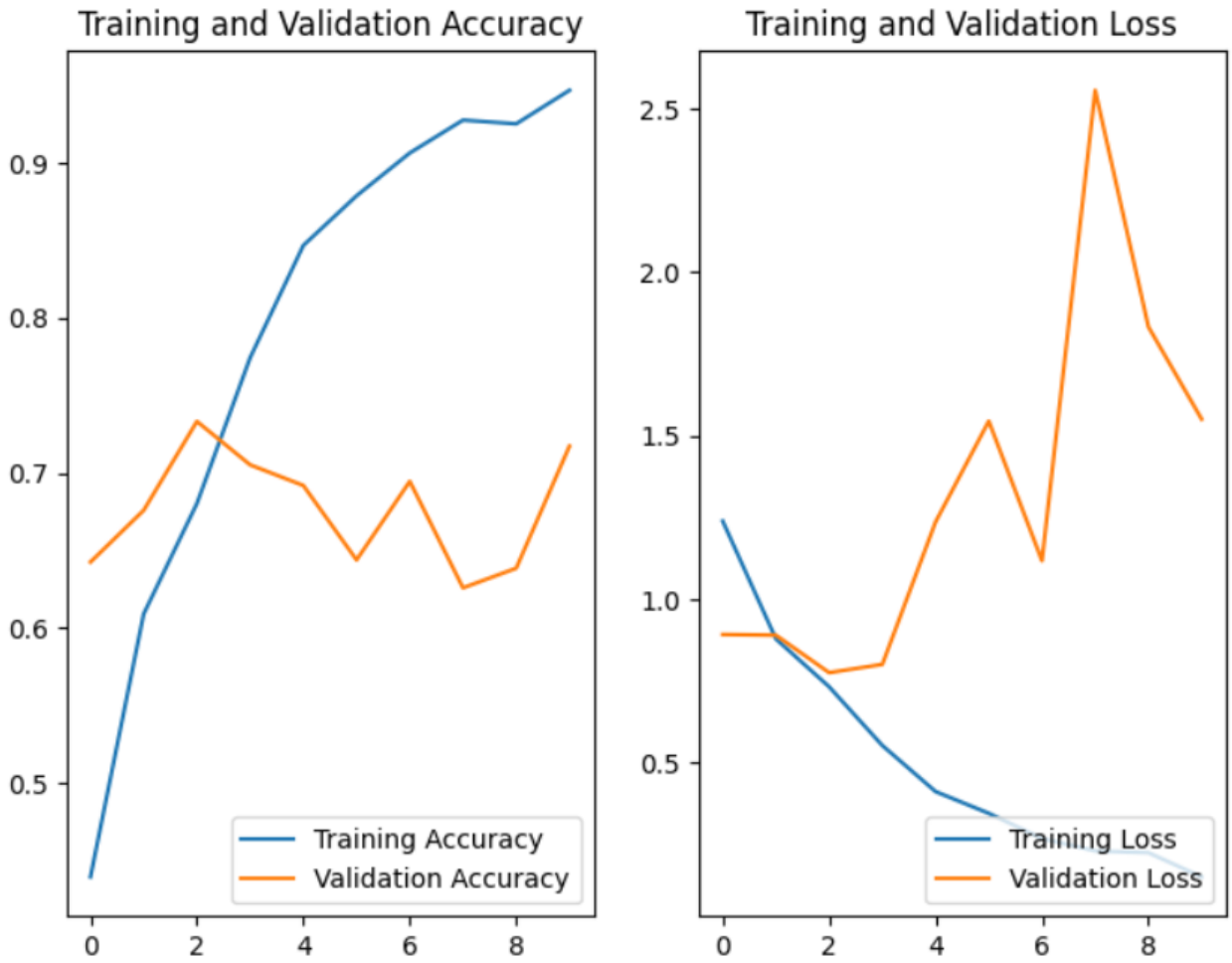
An activation function must be selected for specific layers in the neural networks. These activation functions assist with backpropagation, which uses gradient descent to reduce the loss function with respect to each weight in the model (Backpropagation in Data Mining). The activation function allows for non-linear transformation to the input data, which allows for more complex classification (Tiwari). There are different options for activation functions depending on the purpose of the model. These include linear, sigmoid, tanh, RELU, and softmax. For my model, I will be using RELU and softmax for different layers within the model. The RELU activation function is a widely used function in image classification neural networks. The RELU function ranges from 0 to infinity, and provides a number X if X is positive, otherwise it produces a 0. The RELU function is more efficient then other activation functions as it learns faster compared to other possible functions (Tiwari). In addition, I will also be using a softmax activation function. The softmax activation function is a nonlinear function similar to the

sigmoid function, and is often used for multi-class classification. It is often used in the output layer to provide the probabilities of the input being in any of the categories. I used the softmax activation layer for the final dense layer as this turns the output vector into a vector of probabilities, making the output easier to read (Priya C). This activation function is ideal for multiclass classification problems (Priya C).

I will be using a standard neural network, and will hyperparameter tune to increase accuracy. The model is made up of 3 convolution layers, each with its own pooling layer. The very first layer defines the input shape of the tensors. The first convolutional layer will start with 16 layers, since convolutional layers early in the model learn fewer filters, while later layers learn more (Rosebrock). It is recommended by multiple articles, including Rosebrock and Tensorflow's own documentation, to begin with 32,64,128 filters to the first 3 convolutional layers respectively, and work up from there. Using a RELU activation function will convert any negative features in the feature map to 0. The max pooling layer will select the highest feature in the feature map after the RELU function is applied.

A loss function represents a metric that the model is trying to improve on, such as mean squared error, precision, recall, etc. Classification models in neural networks commonly use cross entropy (Kumar). Keras has built-in loss functions for cross entropy, the one I will be using for this model is categorical cross entropy, which is used for classification of multiple classes.

After running the model with 10 epochs on the training and validation datasets, I got a decent accuracy score of 96% and 72% respectively. Since there is a somewhat large gap between the training and validation scores, it's possible my model is overfitting the data. The graph below shows the training and validation accuracy scores, as well as the loss:



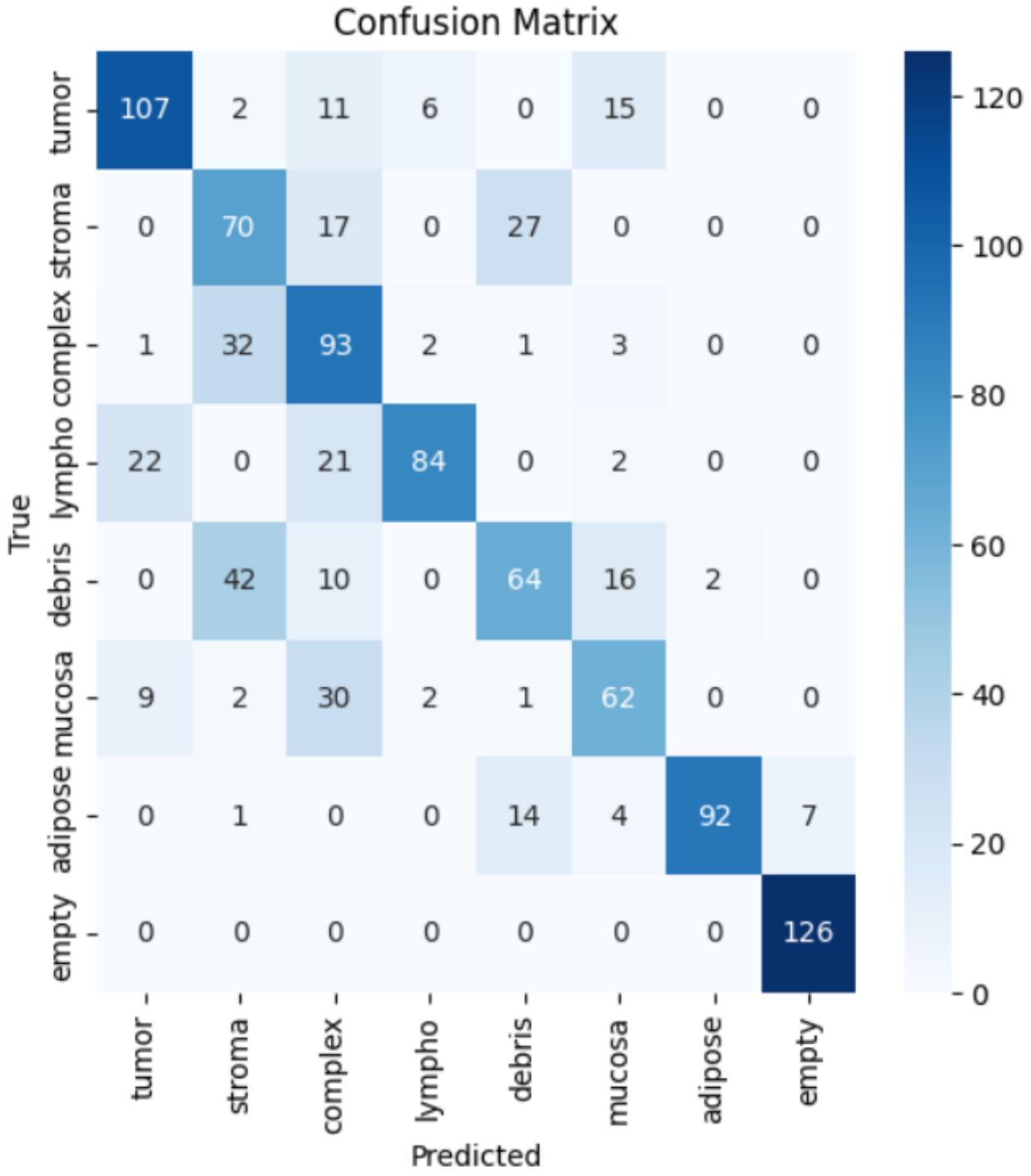
With the hope to improve the validation accuracy and reduce the likelihood of overfitting, I will add a drop layer to the model. Overfitting is when the model essentially memorizes the training data, and does not apply well to data that it has not been trained on. Adding a drop layer will help to avoid overfitting by randomly dropping a secession of output units from the layer during training (Martin et al.). The parameters within the dropout layer include the percentage of outputs that will be dropped during training. I selected 20%, and will update if the overfitting continues.

After adding the dropout layer, my training and validation scores were 95% and 71% respectively, so the dropout layer reduced the accuracy of the model. I will be using the Keras

Tuner to hypertune the model with the goal of improving the accuracy of the model. Similar to a Grid Search for an ML model, the Keras Tuner is a ‘library that helps you pick the optimal hyperparameters for your TensorFlow program’ (Martin et al.). I will use this tuner to find the optimal parameters for my image classification model. Parameters I am adjusting include the learning rate of the model, as well as the number of units in the dense layers. I also tuned the drop layer to see if a different parameter would improve the accuracy score from 71%.

The best parameters identified by the Keras Tuner were 64 nodes for the dense layer, a learning rate of 0.001, and a dropout rate of 0.4. I reran the model to find the optimal amount of epochs, which was 3. Running the training dataset on the optimal parameters got a training accuracy score of 74% and a validation accuracy score of 68%.

While the issue of overfitting seems to have been addressed, the model's accuracy did not improve much. When running the optimal parameters on the testing dataset, I got an accuracy of 67%. A confusion matrix, showing the predicted vs true labels of each category can be found below:



Since the Keras Tuner is slow to run, taking about an hour and a half for the parameters I provided, I was not able to examine as many parameters as I would have liked. I believe that

having additional parameters for the Keras Tuner to test could potentially improve the accuracy of the model.

Overall, I believe incorporating a neural network learning model into classification of colorectal histologies could assist in early diagnosis of colorectal cancer, which has the capacity to save lives due to the aggressive nature of the cancer, and the importance of early treatment. I enjoyed completing this project, and found it to be a good introduction to TensorFlow. If I had additional time and computer bandwidth I would like to tune the parameters more, and test out different combinations of layers.

Works Cited

Kather JN, Weis CA, Bianconi F, Melchers SM, Schad LR, Gaiser T, Marx A, Zollner F: Multi-class texture analysis in colorectal cancer histology (2016), Scientific Reports (in press)

"| Notebook.community." *Notebook.community*, notebook.community/GoogleCloudPlatform/ml-design-patterns/04_hacking_training_loop/transfer_learning. Accessed 2 Nov. 2023.

Boesch, Gaudenz. "A Complete Guide to Image Classification in 2023." *Viso.ai*, 30 Jan. 2023, viso.ai/computer-vision/image-classification/#:~:text=Today%2C%20the%20use%20of%20convolutional. Accessed 5 Nov. 2023.

Dasa, Ashraf. "TensorFlow - Image Classification of Colorectal Cancer Histology - 92.5% Accuracy." *Medium*, 21 Oct. 2021, medium.com/@ashraf.dasa/tensorflow-image-classification-of-colorectal-cancer-histology-92-5-accuracy-8b8b40ac775a. Accessed 2 Nov. 2023.

Ghosh, Sourodip, et al. "Colorectal Histology Tumor Detection Using Ensemble Deep Neural Network." *Engineering Applications of Artificial Intelligence*, vol. 100, Apr. 2021, p. 104202, https://doi.org/10.1016/j.engappai.2021.104202. Accessed 7 May 2022.

"How to Normalize, Center, and Standardize Image Pixels in Keras?" *GeeksforGeeks*, 23 Feb. 2021,

www.geeksforgeeks.org/how-to-normalize-center-and-standardize-image-pixels-in-keras/.

IBM. “What Are Convolutional Neural Networks? | IBM.” *Www.ibm.com*, 2023, www.ibm.com/topics/convolutional-neural-networks.

Kohir, Virajdatt. “Starting with TensorFlow Datasets -Part 2; Intro to Tfds and Its Methods.” *Medium*, 10 Jan. 2022, kvirajdatt.medium.com/starting-with-tensorflow-datasets-part-2-intro-to-tfds-and-its-methods-32d3ac36420f.

Kumar , Ajitesh. “Keras - Categorical Cross Entropy Loss Function.” *Data Analytics*, 28 Oct. 2020, vitalflux.com/keras-categorical-cross-entropy-loss-function/.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

Masciadrelli, Michael. “Why Are Colorectal Cancer Rates Rising among Younger Adults?” *Medicine.yale.edu*, 29 Mar. 2023, medicine.yale.edu/news-article/with-colorectal-cancer-rates-rising-among-younger-adults-a-yale-cancer-center-expert-explains-there-may-be-more-factors-behind-this-worrisome-trend/.

“Neural Networks Part 8: Image Classification with Convolutional Neural Networks.” *Www.youtube.com*, www.youtube.com/watch?v=HGwBXDKFk9I. Accessed 24 Aug. 2021.

Priya C, Bala Priya C. “Softmax Activation Function: Everything You Need to Know.” *Pinecone*, www.pinecone.io/learn/softmax-activation/.

Parashar, Abhay. “4 Techniques to Tackle Overfitting in Deep Neural Networks.” *Comet*, 11 Nov. 2022, www.comet.com/site/blog/4-techniques-to-tackle-overfitting-in-deep-neural-networks/. Accessed 12 Nov. 2023.

Team, Keras. “Keras Documentation: Rescaling Layer.” *Keras.io*, keras.io/api/layers/preprocessing_layers/image_preprocessing/rescaling/.

“TensorFlow 2.0 Complete Course - Python Neural Networks for Beginners Tutorial.” *Www.youtube.com*, www.youtube.com/watch?v=tPYj3fFJGjk&t=11361s. Accessed 14 Nov. 2023.

Tiwari, Sakshi. “Activation Functions in Neural Networks - GeeksforGeeks.” *GeeksforGeeks*, 6 Feb. 2018, www.geeksforgeeks.org/activation-functions-neural-networks/.