Neurons Saving Neurons: Development of a Machine Learning Model for Prediction of Ischemic Stroke using Non-Contrast Enhanced Computed Tomography (NCCT) Imaging

By Josh Lazar

1. Abstract

Determining whether a patient presenting with stroke symptoms is experiencing an ischemic event is a very difficult problem with large implications. Deciding whether or not to treat a suspected stroke patient must be performed quickly in order to maximize the potential for saving brain tissue, so using what little data is available to its maximum potential is of paramount importance. Using the most common data collected on a suspected stroke patient, non-contrast enhanced computed tomography (NCCT) images, a single-layer feedforward neural network (FNN) was developed to provide an informed decision to a caregiver whether a patient is experiencing an ischemic stroke. The model achieved a 78.0% accuracy on the test set with 77.4% sensitivity and 78.4% specificity.

2. Introduction

In 2021, stroke and other cerebrovascular diseases were the fifth leading cause of death in the United States (162,890 deaths), behind heart disease (695,547), cancer (605,213), COVID-19 (416,893), and accidents (224,935)[1]. A stroke can occur either through the blockage of a blood vessel in the brain via a clot, which is known as an ischemic stroke, or through the rupture of an aneurysm, which is known as a hemorrhagic stroke. Globally, ischemic strokes account for 87% of all strokes[2]. The focus of this paper will be on ischemic stroke.

There is a life-saving thrombolytic drug for treating ischemic stroke patients called intravenous tissue plasminogen activator (IV-tPA), but this drug must be administered within 4.5 hours of symptom onset in order to be considered safe and effective[3]. However, patients suspected of having a stroke are not immediately given this drug because the symptoms of ischemic and hemorrhagic stroke are very similar, and administering IV-tPA to a hemorrhagic stroke patient can be fatal as it thins the blood. Therefore, patients suspected of having a stroke are normally put through a head computed tomography (CT) scan. Hemorrhage is very obvious from a CT scan (see images below), so typically if this is not observed and the patient is within the IV-tPA window, they will immediately be administered the drug. However, if a bleed is not observed and they are outside the window, the situation becomes difficult as it is difficult to identify an ischemic stroke from CT images. Typically, this means further studies including magnetic resonance imaging (MRI) or contrast-enhanced CT imaging before intervention through mechanical thrombectomy (removal of blood clot causing blockage), which can take time and mean the death of millions of brain cells in the meantime.

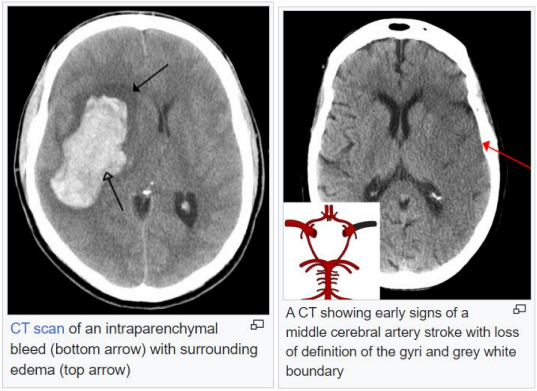


Figure 1: CT Scan Images of Hemorrhagic (left) and Ischemic (right) Stroke Patients[4]

Therefore, the goal of this project is to use the images initially gathered on suspected stroke patients to make an informed decision as to whether or not to intervene. A CT scan provides a 3D representation of brain tissue by taking individual slices at a specified thickness. For head and neck CT scans, this thickness is recommended to be no more than 3mm but can vary depending on the X-ray machine manufacturer's specifications and the settings selected[5]. This results in a set of images of fixed resolution (but variable number) that represent a single patient's scan. These images are the input to the model, and the output of the model is a binary decision: whether or not an ischemic stroke has occurred.

3. Related Work

The state-of-the-art for NCCT image analysis is provided by a company called Rapid AI, who offer a broad suite of products for stroke, aneurysms, pulmonary embolism, and other applications. Their Rapid NCCT Stroke application is an FDA-approved AI-based medical device that determines suspicion of intracranial hemorrhage (ICH) and large-vessel occlusion (LVO) based on NCCT imaging. They claim to be the first to give integrated suspicion output based on multiple stroke-related indicators. They also claim to have demonstrated a 55% increase in sensitivity when compared to the intuition of general radiologists[6]. According to their 510(k) summary published on the FDA website, this translates to a sensitivity of 0.635 (95% CI: 0.544 - 0.717) and specificity of 0.951 (95% CI: 0.891 – 0.979). Subgroup analyses were also performed, which did not show significant differences in sensitivity or specificity across patient gender, age, slice thickness, or X-ray machine manufacturer[7]. The only details provided about their implementation is that the technology used is "AI/ML/Neural Network."

There have also been attempts in academia to develop similar prediction algorithms:

1. Shinohara et. al developed a deep convolutional neural network for classifying NCCT images as hyperdense middle cerebral artery positive or negative and achieved an 82.9% sensitivity and 89.7% specificity[8].
2. Lisowska et. al proposed a 3D convolutional neural network designed to exploit contralateral features and anatomical atlas information and achieved a ROC AUC of 0.996 and a Precision-Recall AUC of 0.563 in a voxel-level evaluation, which was deemed not at a level for routine clinical use[9].
3. Abedi et. al developed an artificial neural network to recognize acute cerebral ischemia and differentiate stroke mimics in an emergency setting and achieved an average sensitivity of 80.0% and specificity of 86.2% in a 10-fold cross-validation analysis. This study utilized diffusion-weighted imaging and apparent diffusion coefficient data, which are outputs of MRI[10].
4. The Ischemic Stroke Lesion Segmentation (ISLES) challenge in 2015 pitted 21 research teams against each other to develop an algorithm for ischemic stroke lesion segmentation from multi-spectral MRI images. The top ranking teams used convolutional neural networks, as noted by Litjens et al[11,12].

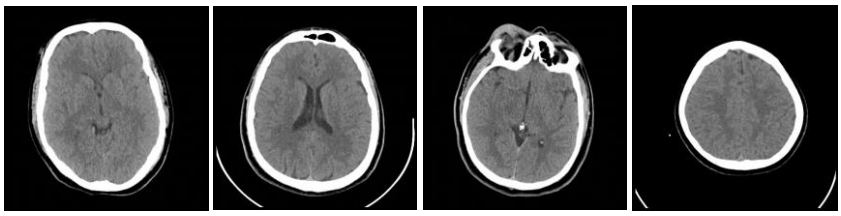
4. Dataset and Features

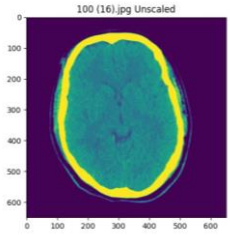
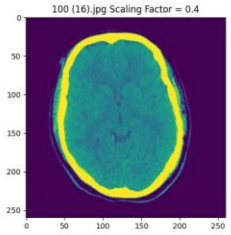
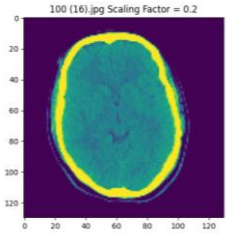
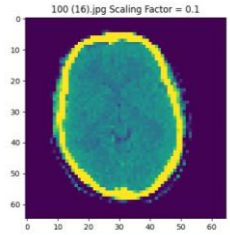
A publicly available dataset was pulled from Kaggle that contains categorized images of brain CT scans[13]. The images are labeled with the patient number followed by the slice number, meaning that it is known which set of images belong to a single patient. However, upon inspection of the training dataset, not every slice from the scan is always included and the slice numbers usually increase sequentially from inferior to superior, but this is not always the case. The following table shows the number of images and patients in each set and class in the pre-divided data.

Set	Class	Number of Images	Number of Examples	% of Total (Images)	% of Total (Examples)
Train	Normal	1087	51	73.3%	35.2%
	Stroke	756	31		
Dev	Normal	157	43	9.3%	29.6%
	Stroke	78	26		
Test	Normal	307	51	17.4%	35.2%
	Stroke	130	31		
Total	Normal	1551	145	100%	100%
	Stroke	964	88		
	All	2515	233		

Each image is a square 650-pixel x 650-pixel black and white image, where each pixel has a value from 0 to 1. A few examples of slice images from the training set can be seen below, where the black border is the space that surrounds the patient, the bright white ring is the skull, and the gray in the middle of the ring is the brain tissue of interest. It was immediately apparent that it would be necessary to scale down the images, since the unscaled image contains 650*650=422,500 pixels, resulting in 422,500 parameters for the model to handle. Thus, a hyperparameter was introduced called a "scaling factor," which makes use of Scikit Image's rescale function[14] to size the image down and reduce the

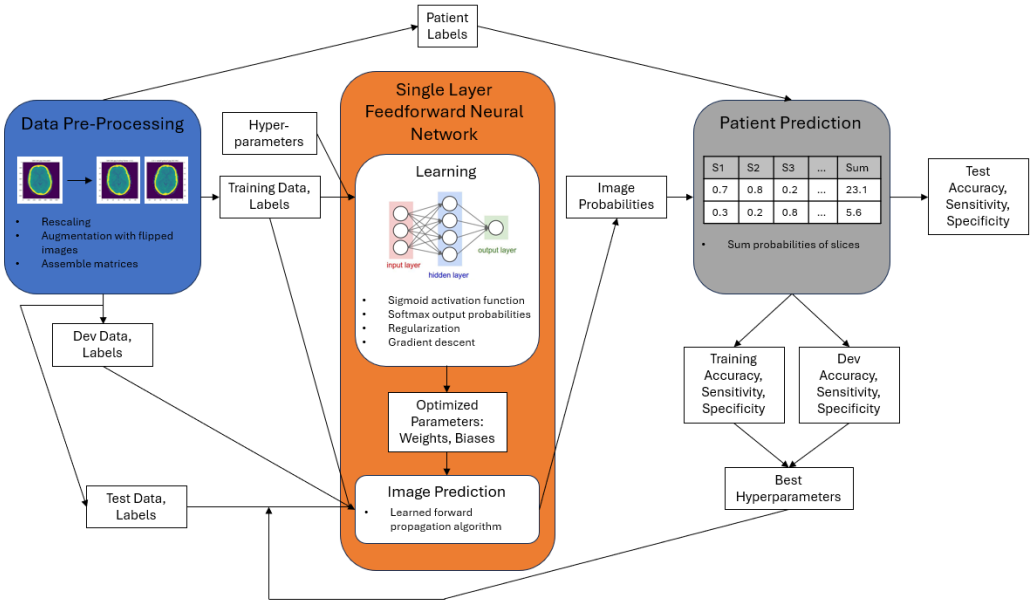
number of parameters by the scaling factor squared. The table below the images shows the first example slice image with various scaling factors.



SF	1	0.4	0.2	0.1
Image				
Params (pixels)	422,500	67,600	16,900	4,225

The training and validation data sets were read in and the images were scaled and formed into a vector using numpy.ravel[15], then this vector was loaded into a matrix. Label vectors were also assembled, where 0 corresponds to an image from a normal patient and 1 corresponds to an image from a stroke patient. Patient label and slice vectors were also assembled based on the file names. Then, taking advantage of the lateral symmetry of the brain, left-right mirrored images were added to the training set, effectively doubling the size of the training set.

5. Methods



The above graphical representation shows how the machine learning model was developed and tested. First, data pre-processing as described in section 4 was performed. Then, the training data and labels were passed into a single layer feedforward neural network. Then, the probabilities from this prediction are fed to a patient predictor.

The goal of a single layer feedforward neural network is to learn the parameters of the model, the weights and the biases. There is one less weight matrix and bias vector than there are layers in the neural network, as these help to translate from one layer to the next, so for a single hidden layer there are two weight matrices and two bias vectors. During forward

propagation, the input data is multiplied by the first weight matrix and the first biases are added, then the activation function is applied in order to produce non-linearity, producing the values of the hidden layer. Then, the values of the hidden layer are multiplied by the second weight matrix and the second biases are added. When the softmax function is applied, this produces a probability for each classification, the output layer. Cross entropy loss is calculated based on the training labels, then, in a process called backpropagation, the derivative of the loss with respect to each parameter is calculated, and gradient descent is used to adjust each of the parameters in the direction that will minimize loss. This is repeated over several epochs in order to continually improve the parameters to minimize the loss.

This neural network was used to create an image prediction, and this probability was fed into a secondary classification predictor. This predictor sums the probabilities for each slice in a patient's scan to determine which classification has the highest overall probability, then makes a binary decision based on these summed probabilities.

The metrics selected for this model are accuracy, sensitivity, and specificity. Sensitivity and specificity are traditionally used in the medical field as these metrics are commonly used in drug trials. These metrics are based on the confusion matrix and can be written as:

$$Se = \frac{\#\ True\ Positive}{\#\ True\ Positive + \#\ False\ Negative} \qquad Sp = \frac{\#\ True\ Negative}{\#\ True\ Negative + \#\ False\ Positive}$$

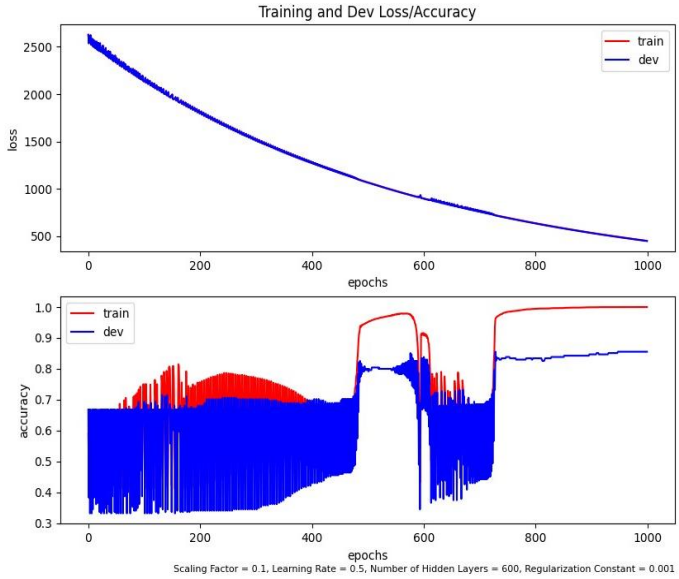
The accuracy, sensitivity, and specificity of the model on the training data could then be calculated. The validation (dev) data could also be fed through the two predictors to obtain a dev accuracy, sensitivity, and specificity. The dev accuracy, sensitivity, and specificity were then used to select the model hyperparameters, and the final model was tested on the test set.

6. Experiments/Results/Discussion

Several initial experiments were conducted to determine the right ballpark for the hyperparameters. The scaling factor was fixed at 0.1 because it did not appear that a higher scaling factor yielded improved results and significantly increased computational cost. The number of epochs was also fixed at 800, although it was noted that some models did not appear to converge by this point. Then, a full factorial experiment was conducted with the remaining hyperparameters: the learning rate, the number of nodes in the hidden layer, and the regularization constant. 3 different learning rates, 2 different numbers of hidden layer nodes, and 5 different regularization constants were selected for a total of 3*2*5=30 different models. The results can be seen below.

Inputs			Outputs						
			Training			Dev			
Learning Rate	Hidden Layer Nodes	Regularization Constant	Accuracy	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity	Average
0.10	300	0	96.3%	80.6%	100.0%	62.3%	30.8%	81.4%	58.2%
0.25			100.0%	100.0%	100.0%	73.9%	61.5%	81.4%	72.3%
0.50			100.0%	100.0%	100.0%	68.1%	50.0%	79.1%	65.7%
0.10	600		89.0%	41.9%	100.0%	66.7%	30.8%	88.4%	61.9%
0.25			100.0%	100.0%	100.0%	72.5%	50.0%	86.0%	69.5%
0.50			100.0%	100.0%	100.0%	72.5%	53.8%	83.7%	70.0%
0.10	300	0.1	81.1%	0.0%	100.0%	62.3%	0.0%	100.0%	54.1%
0.25			81.1%	0.0%	100.0%	62.3%	0.0%	100.0%	54.1%
0.50			81.1%	0.0%	100.0%	62.3%	0.0%	100.0%	54.1%
0.10	600		81.1%	0.0%	100.0%	62.3%	0.0%	100.0%	54.1%
0.25			68.9%	100.0%	61.7%	31.9%	76.9%	4.7%	37.8%
0.50			68.9%	100.0%	61.7%	31.9%	76.9%	4.7%	37.8%
0.10	300	0.01	81.7%	3.2%	100.0%	63.8%	3.8%	100.0%	55.9%
0.25			81.7%	3.2%	100.0%	63.8%	3.8%	100.0%	55.9%
0.50			82.3%	6.5%	100.0%	65.2%	7.7%	100.0%	57.6%
0.10	600		81.1%	0.0%	100.0%	63.8%	3.8%	100.0%	55.9%
0.25			81.7%	3.2%	100.0%	63.8%	3.8%	100.0%	55.9%
0.50			81.1%	0.0%	100.0%	62.3%	0.0%	100.0%	54.1%
0.10	300	0.001	90.9%	51.6%	100.0%	69.6%	30.8%	93.0%	64.5%
0.25			94.5%	100.0%	93.2%	56.5%	69.2%	48.8%	58.2%
0.50			73.8%	100.0%	67.7%	39.1%	76.9%	16.3%	44.1%
0.10	600		100.0%	100.0%	100.0%	60.9%	42.3%	72.1%	58.4%
0.25			76.2%	100.0%	70.7%	39.1%	73.1%	18.6%	43.6%
0.50			100.0%	100.0%	100.0%	82.6%	69.2%	90.7%	80.8%

0.10	300	0.0001	90.2%	48.4%	100.0%	66.7%	26.9%	90.7%	61.4%
0.25			99.4%	100.0%	99.2%	73.9%	65.4%	79.1%	72.8%
0.50			100.0%	100.0%	100.0%	73.9%	53.8%	86.0%	71.3%
0.10	600		94.5%	100.0%	93.2%	53.6%	61.5%	48.8%	54.7%
0.25			100.0%	100.0%	100.0%	76.8%	57.7%	88.4%	74.3%
0.50			100.0%	100.0%	100.0%	82.6%	69.2%	90.7%	80.8%



The model with the highest average dev accuracy/sensitivity/specificity was selected, with a learning rating of 0.50, 600 hidden layer nodes, and 0.001 regularization constant (tied with 0.0001). The relatively high learning rate appears to make the accuracy jump up and down in earlier epochs, but this eventually stabilizes around 750 epochs. Interestingly, the training accuracy takes a large dive at around 600 epochs but then makes a recovery where the dev accuracy is also stable. There is likely significant overfitting in the model seeing that there is a large gap between the converged training and dev accuracies. The loss function performs as expected with a constant decrease over the number of epochs.

This model was then tested on the test set and resulted in an accuracy of 78.0%, a sensitivity of 77.4%, and a specificity of 78.4%.

7. Conclusion/Future Work

A neural network was developed for classification of CT images into normal and stroke categories that achieved an accuracy of 78.0%, a sensitivity of 77.4%, and a specificity of 78.4% on the test set. While the sensitivity achieved is higher than the average benchmark of Rapid AI by over 10%, the specificity is significantly lower. Further model refinement is necessary before moving forward into clinical applications.

In the future, given more time, resources, and expertise, there are a few improvements that I believe would strengthen this model.

- First, improving the patient prediction model I believe would add value and help improve accuracy. Certain slices may be more important for prediction than others (e.g. those with more brain tissue), so I would like to explore this rather than summing all the probabilities. This task is made somewhat complicated by the fact that each scan contains a different number of images, and the images in the dataset are not consistently ordered. There would likely need to be some manual ordering before the dataset is ready for weighting or learning based on each specific slice in the scan.
- Secondly, I think that further supplementing the training set with images that are slightly rotated could help the model learn better. This could add a large number of training examples and thus likely improve the model accuracy.
- I also think that there is room for improvement in the explainability of the model. Currently, the model only outputs a 1 or a 0 based on the scan images. However, there is value in pointing to which specific image(s) in the scan contributed most to this conclusion and which superpixels within the image(s) contributed. This would identify specific regions of the brain that are undergoing an ischemic event and could provide caregivers with more information that would allow them to determine whether or not treatment is necessary.
- Finally, I think that gathering more data would be helpful, including more scans as well as more features including the patient's gender, age, and the time from symptom onset when the scan was taken. Because the early signs of ischemia are very faint, if the model had the time from symptom onset to when the scan was taken, it could learn to look for more subtle features in those where this measure is less.

8. Appendices - N/A for Applications project

9. Contributions

This project was performed solely by Josh Lazar, Master's Student in Mechanical Engineering at Stanford University.

10. References/Bibliography

1. "FASTSTATS - Leading Causes of Death." Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 18 Jan. 2023, www.cdc.gov/nchs/fastats/leading-causes-of-death.htm.
2. Experts, PNI. "What Is the Difference between Hemorrhagic and Ischemic Stroke?" Pacific Neuroscience Institute, 3 Mar. 2023, www.pacificneuroscienceinstitute.org/blog/stroke/what-is-the-difference-between-hemorrhagic-and-ischemic-stroke/.
3. Guidelines for the Early Management of Patients with Acute Ischemic Stroke:, www.stroke.org/-/media/Stroke-Files/Ischemic-Stroke-Professional-Materials/AIS-Toolkit/Guidelines-for-Mangaging-Patients-with-AIS-2019-Update-to-2018-Guidelines.pdf. Accessed 9 Dec. 2023.
4. "Stroke." Wikipedia, Wikimedia Foundation, 6 Dec. 2023, en.wikipedia.org/wiki/Stroke.
5. AAPM Medical Physics Practice Guideline 12.A ... - Wiley Online Library, aapm.onlinelibrary.wiley.com/doi/10.1002/acm2.13526. Accessed 9 Dec. 2023.
6. iSchemaView, Inc. "Faster Triage & Transfer Decisions Based on CT Imaging." RapidAI NCCT Stroke, www.rapidai.com/rapid-ncct. Accessed 8 Dec. 2023.
7. "510(k) Premarket Notification." Accessdata.Fda.Gov, www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfpmn/pmn.cfm?ID=K222884. Accessed 8 Dec. 2023.
8. Shinohara Y;Takahashi N;Lee Y;Ohmura T;Kinoshita T; "Development of a Deep Learning Model to Identify Hyperdense MCA Sign in Patients with Acute Ischemic Stroke." Japanese Journal of Radiology, U.S. National Library of Medicine, 31 Oct. 2019, pubmed.ncbi.nlm.nih.gov/31673998/.
9. Lisowska, Aneta, et al. "Thrombus Detection in CT Brain Scans Using a Convolutional Neural Network." SCITEPRESS, SCITEPRESS, 21 Feb. 2017, www.scitepress.org/Papers/2017/61146/.
10. Abedi, Vida, et al. "Novel Screening Tool for Stroke Using Artificial Neural Network." Stroke, U.S. National Library of Medicine, 24 Apr. 2017, pubmed.ncbi.nlm.nih.gov/28438906/.
11. "Isles Challenge 2015 Ischemic Stroke Lesion Segmentation." ISLES, www.isles-challenge.org/ISLES2015/. Accessed 8 Dec. 2023.
12. Litjens, Geert. "A Survey on Deep Learning in Medical Image Analysis." Medical Image Analysis, Elsevier, 26 July 2017, www.sciencedirect.com/science/article/pii/S1361841517301135.
13. Rahman, Afridi. "Brain Stroke CT Image Dataset." Kaggle, 18 June 2021, www.kaggle.com/datasets/afridirahman/brain-stroke-ct-image-dataset.
14. "Rescale, Resize, and Downscale#." Rescale, Resize, and Downscale - Skimage 0.22.0 Documentation, scikit-image.org/docs/stable/auto_examples/transform/plot_rescale.html. Accessed 8 Dec. 2023.
15. "Numpy.Ravel#." Numpy.Ravel - NumPy v1.26 Manual, numpy.org/doc/stable/reference/generated/numpy.ravel.html. Accessed 8 Dec. 2023.