<https://piazza.com/class_profile/get_resource/l1a9bvmlfoa462/l3e2dn4zqfb2o2>

Sprint 1

As a user of the website, I want the GUI to allow for dense layers, so I can generate basic networks.

As a user of the website, I want a GUI for creating perceptron tensorflow networks, so that I can make them in an intuitive fashion.

As a user of the website, I want a GUI to also create pytorch networks, so that I can choose different implementations for the same network.

Scenario:

1. Go to website; hover over ‘+’ button on toolbar on left side of canvas.
   1. Click ‘Add operator’ and select ‘Dense’ button.
   2. Dense layer should appear on canvas and can be selected/ dragged around the canvas.
   3. Layer can be adjusted in size and shape by dragging the corner boxes of the shape.
   4. Multiple layers can be added and will snap together if input/ output boxes are dragged over each other. Double-clicking the connection will disconnect the two operators.
   5. Grid of canvas can be toggled (automatically turned on). When grid is active, layers will snap to grid points. When grid is toggled off, layers can be moved in a freestyle manner.
   6. Connect leftmost tensor (dotted square) of dense operator to input tensor (orange tensor on left) (click and drag either)
   7. Connect rightmost tensor (dotted square) of dense operator to output tensor (blue tensor on right) (click and drag either)
   8. click on ‘Fully Connected’ in operator toolbar
   9. put any positive, non zero integer in the input box
   10. put any positive, non zero integer in the output box, the parameter box should have updated
   11. press the submit button
   12. press the generate code button, in top right corner
   13. make choices for each of the three dropdown menus, or leave them as defaults
   14. press the generate button
   15. click ok on the pop-up
   16. .py file should have downloaded

Sprint 2

As a user of the website, I want to generate the code that optimizes the network as well, as I want an intuitive interface for the entire machine learning pipeline.

As a user of the website, I want the GUI to allow for convolutional networks as well, so I can generate more types of networks.

Scenario:

1. Go to website; hover over ‘+’ button on the toolbar on the left side of canvas. Click ‘add operator’. add convolutional layer, move layer away from corner, add dense layer, move layer away from corner, add preLu layer, move layer away from corner
   1. Connect the leftmost convolutional layer tensor (dotted square) to the to input tensor (orange tensor on left) (click and drag either)
   2. Connect the dense layers leftmost tensor to the convolutional’s right most tensor. Connect the preLu’s leftmost tensor to the right most tensor on the dense layer. Connect the preLu’s rightmost tensor to the output tensor (blue tensor on right).
   3. Click on the convolution layer on the left side bar
   4. Fill in convolutional layer parameters
      1. Input image width = <28>
      2. Input image height = <28>
      3. Kernel width = <10>
      4. Kernel height = <10>
      5. Filter = <3>
      6. Click submit
   5. Fill in dense layer
      1. Input should already be filled in
      2. Output = <10>
      3. Parameter shape update
      4. Click submit
2. Click ‘generate code’
3. Click ‘generate’
4. Copy code in downloaded file
5. Navigate to <https://colab.research.google.com/drive/1OgM8bTDqNrCd5EgvtW0LWmPxWMleuHFG?authuser=2#scrollTo=HZP2EwWmsqh9>
6. Paste code into 3rd module box, titled ACTUAL MODEL deleting anything in there already
7. Click runtime at the top; click run all
8. User should see the result of the model being trained on the dataset at the very bottom of the colab file

Sprint 3

As a user of the website, I want it to tell me basic information about the network, such as number of parameters and number of layers, so I can have a better understanding of the network at a glance.

As a user of the website, I want a wide array of layers and activation functions to choose from

As a user of the website, I want the experience of creating the website to be simple and intuitive

Scenario:

1. Go to website; hover over ‘+’ button on the toolbar on the left side of canvas. Click ‘add operator’ add convolutional layer, move layer away from corner, add dense layer, move layer away from corner, add preLu layer, move layer away from corner
   1. Connect the leftmost convolutional layer tensor (dotted square) to the to input tensor (orange tensor on left) (click and drag either)
   2. Connect the dense layers leftmost tensor to the convolutional’s right most tensor. Connect the preLu’s leftmost tensor to the right most tensor on the dense layer. Connect the preLu’s rightmost tensor to the output tensor (blue tensor on right).
   3. Click on the convolution layer on the left side bar
   4. Fill in convolutional layer parameters
      1. Input image width = <28>
      2. Input image height = <28>
      3. Kernel width = <10>
      4. Kernel height = <10>
      5. Filter = <3>
      6. Click submit
   5. Fill in dense layer
      1. Input should already be filled in
      2. Output = <10>
      3. Parameter shape should update
      4. Click submit
2. User should see the # of parameters and # of layers in the bottom right corner
3. Click the ‘?’ in the top right corner
4. User should see a pop up on how to use the website
5. Click the X in the top right corner
6. Click ‘Clear Canvas’ on the top bar
7. Hover over ‘+’ button on the toolbar on the left side of canvas. Click ‘add operator’. Add a dense layer, move layer away from the corner, add softmax.
   1. Connect the dense layer to the start box (red box on the left side) using the left most tensor (the left square on the dense layer). Connect the right most tensor on the dense layer to the left most tensor on the softmax. Connect the right most tensor on the softmax to the blue box on the right side (output).
   2. Click on the dense layer on the left side bar
   3. Fill in dense layer parameters
      1. Input image width = <10>
      2. Input image height = <10>
      3. Click submit
   4. Click ‘generate code’
   5. Click ‘generate’
   6. Copy code in downloaded file
   7. Navigate to <https://colab.research.google.com/drive/1OgM8bTDqNrCd5EgvtW0LWmPxWMleuHFG?authuser=2#scrollTo=HZP2EwWmsqh9>
   8. Paste code into 3rd module box, titled ACTUAL MODEL deleting anything in there already
   9. Click runtime at the top; click run all
   10. User should see the result of the model being trained on the dataset at the very bottom of the colab file
8. Navigate back to the website
9. Click clear canvas
10. Repeat from 7 to 8, replacing softmax for Identity, Relu, Prelu, Softplus, Swish, Sigmoid, Softsign and Tanh. Should therefore be repeated 8 times.
11. Click clear canvas
12. Hover over ‘+’ on the toolbar on the left side of canvas. Click ‘add operator’. Click convolutional layer. Move layer away from the corner. Hover over ‘+’ on the toolbar on the left side of canvas. Click GlobalAvgPool.
13. Connect the left most tensor on the convolutional layer to the input (red box on the left of canvas). Connect the right most tensor on the convolutional layer to the left most tensor on the GlobalAvgPool layer. Connect the right most tensor on the GlobalAvgPool layer to the output (blue box on the right side)
14. Click on the convolutional layer on the left side bar
15. Fill in convolutional layer parameters
    1. Input image width = <28>
    2. Input image height = <28>
    3. Kernel width = <1>
    4. Kernel height = <1>
    5. Filters = <10>
    6. Click submit
16. Click on the globalavgpool layer on the left side bar
17. Fill in globalavgpool layer parameters
    1. Channels = <10>
18. Click ‘generate code’
19. Click ‘generate’
20. Copy code in downloaded file
21. Navigate to <https://colab.research.google.com/drive/1OgM8bTDqNrCd5EgvtW0LWmPxWMleuHFG?authuser=2#scrollTo=HZP2EwWmsqh9>
22. Paste code into 3rd module box, titled ACTUAL MODEL deleting anything in there already
23. Click runtime at the top; click run all
24. User should see the result of the model being trained on the dataset at the very bottom of the colab file
25. Navigate back to the website
26. Click ‘clear canvas’
27. Hover over ‘+’ button on the toolbar on the left side of canvas. Click ‘add operator’. Add a dense layer, move layer away from the corner. Hover over ‘+’ button. Click ‘add operator’. Click convolutional layer, move layer away from the corner. Hover over ‘+’ button. Click ‘add operator’. Click max pool.
    1. Connect the convolutional layer to the start box (red box on the left side) using the left most tensor (the left square on the dense layer). Connect the right most tensor on the convolutional layer to the left most tensor on the max pool layer. Connect the right most tensor on the max pool layer to the left most tensor on the dense layer. Connect the dense layer’s right most tensor to the output (red box on the right)
    2. Click on the convolutional layer on the left side bar
    3. Fill in convolutional layer parameters
       1. Input image width = <28>
       2. Input image height = <28>
       3. Kernel width = <3>
       4. Kernel height = <3>
       5. Filters = <3>
       6. Click submit
    4. Click on max pool layer on the left side bar
    5. Fill in max pool layer parameters
       1. Kernel width = <10>
       2. Kernel height = <10>
       3. strides = <3>
       4. Click submit
    6. Click on the dense layer on the left side bar
    7. Fill in dense layer parameters
       1. Output = <10>
    8. Click ‘generate code’
    9. Click ‘generate’
    10. Copy code in downloaded file
    11. Navigate to <https://colab.research.google.com/drive/1OgM8bTDqNrCd5EgvtW0LWmPxWMleuHFG?authuser=2#scrollTo=HZP2EwWmsqh9>
    12. Paste code into 3rd module box, titled ACTUAL MODEL deleting anything in there already
    13. Click runtime at the top; click run all
    14. User should see the result of the model being trained on the dataset at the very bottom of the colab file
28. Navigate back to the website
29. Repeat 24 - 25 replacing the max pool layer with avg pool layer
30. Click clear code
31. Click yes
32. Hover over ‘+’ button on the toolbar on the left side of canvas. Click ‘add operator’. Add a dense layer, move layer away from the corner. Hover over ‘+’ button. Click ‘add operator’. Click convolutional layer, move layer away from the corner. Hover over ‘+’ button. Click ‘add operator’. Click zero padding.
    1. Connect the zero padding layer to the start box (red box on the left side) using the left most tensor (the left square on the dense layer). Connect the right most tensor on the zero padding layer to the left most tensor on the convolutional layer. Connect the right most tensor on the convolutional layer to the left most tensor on the dense layer. Connect the dense layer’s right most tensor to the output (red box on the right)
    2. Click on the zero padding layer on the left side bar
    3. Fill in convolutional layer parameters
       1. Input image width = <28>
       2. Input image height = <28>
       3. Zero padding = <5>
       4. Click submit
    4. Click on convolutional layer on the left side bar
    5. Fill in max pool layer parameters
       1. Kernel width = <3>
       2. Kernel height = <3>
       3. Filters = <10>
       4. Click submit
    6. Click on the dense layer on the left side bar
    7. Fill in dense layer parameters
       1. Output = <10>
    8. Click ‘generate code’
    9. Click ‘generate’
    10. Copy code in downloaded file
    11. Navigate to <https://colab.research.google.com/drive/1OgM8bTDqNrCd5EgvtW0LWmPxWMleuHFG?authuser=2#scrollTo=HZP2EwWmsqh9>
    12. Paste code into 3rd module box, titled ACTUAL MODEL deleting anything in there already
    13. Click runtime at the top; click run all
    14. User should see the result of the model being trained on the dataset at the very bottom of the colab file
33. Navigate back to the website
34. Repeat 24 - 25 replacing the zero padding layer with BatchNorm layer
    1. BatchNorm only has input/output, so set that to <28> and disregard the zero padding parameter in the instruction

Sprint 4

As a user of the website, I want to be able to access the website across multiple machines.

As someone who is new to neural networks, I want some simple designs to get me started.

Scenario:

1. Click on this link, It will work for any computer connected to the internet
2. <https://neurula.onrender.com/>

Scenario:

1. Access our project through the following link: <https://neurula.onrender.com/>
2. To get started from there, you access a simple design under the plus symbol on the upper left hand of the screen
3. When you hover over it, you will see two options: “add operator” and “add pattern.” Since we are getting we will click add pattern.
4. Afterwards we will get a pop up with 3 options: “image processing head” “feature extractor” or “Classifier.” Pick the one that seems to fit best with what you are looking for and it will appear on the canvas
5. From there you can simply connect the input and output to both ends, fill out the inputs in each layer based on the data you are trying to process and
6. Then you can click generate code and pick the optimizers and loss functions that fits the data you want to process and afterwards when you click generate, it will begin downloading a file where you can find your neural network that you created.