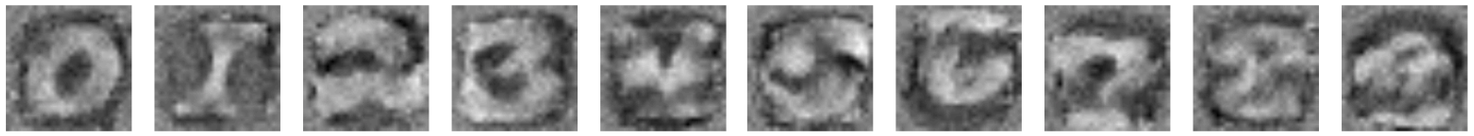
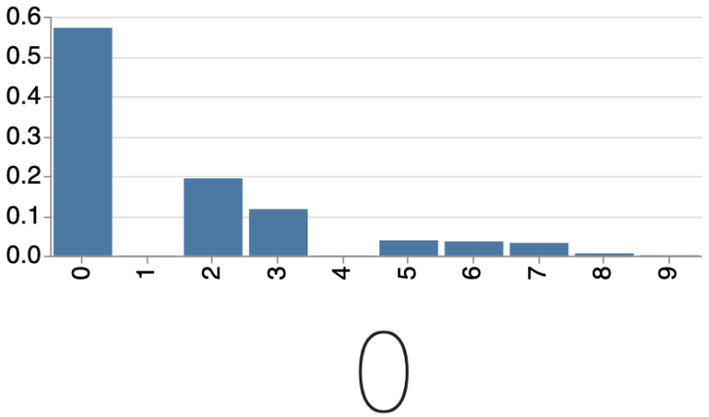
first layer weights



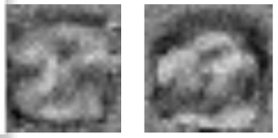
draw

clear

prediction



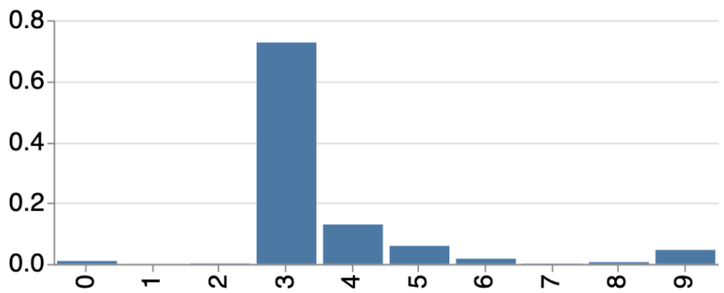
0



draw

clear

prediction



3

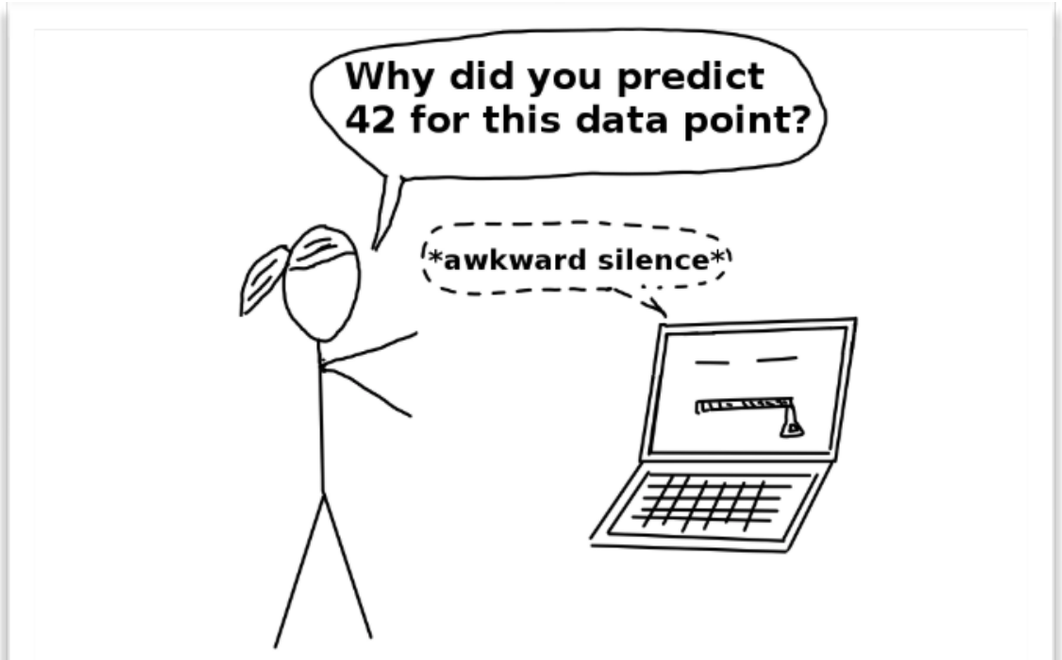
CS 6965 - Final Project
Exploring neural nets with MNIST

Michael Young

background

- neural nets are cool & useful, but complexity can make interpretation of models difficult

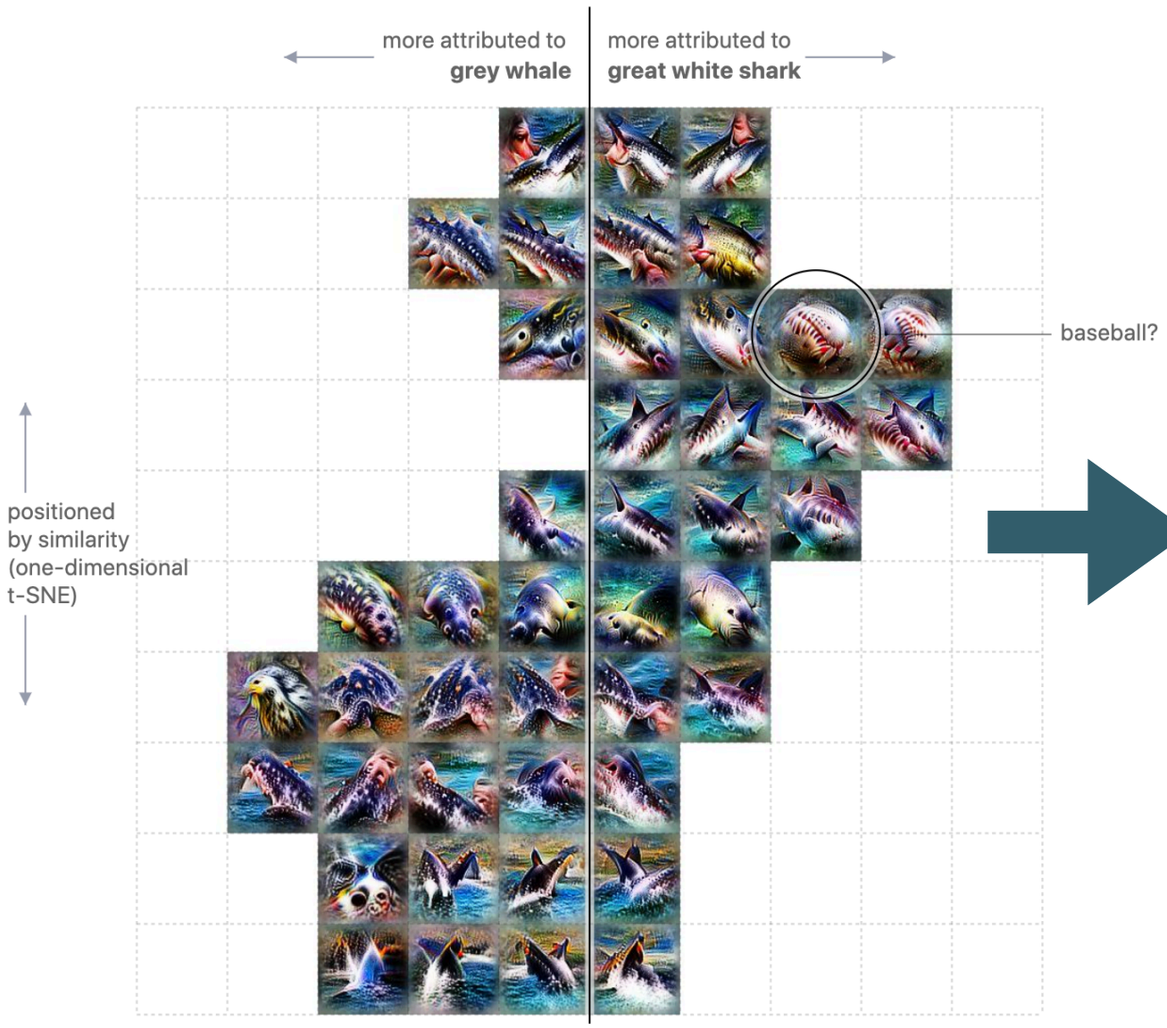
- efforts are being made to crack open models and allow users to peer under the hood in accessible ways



https://towardsdatascience.com/interpretable-machine-learning-1dec0f2f3e6b

background cont.

- an example we've seen of this is the activation atlas

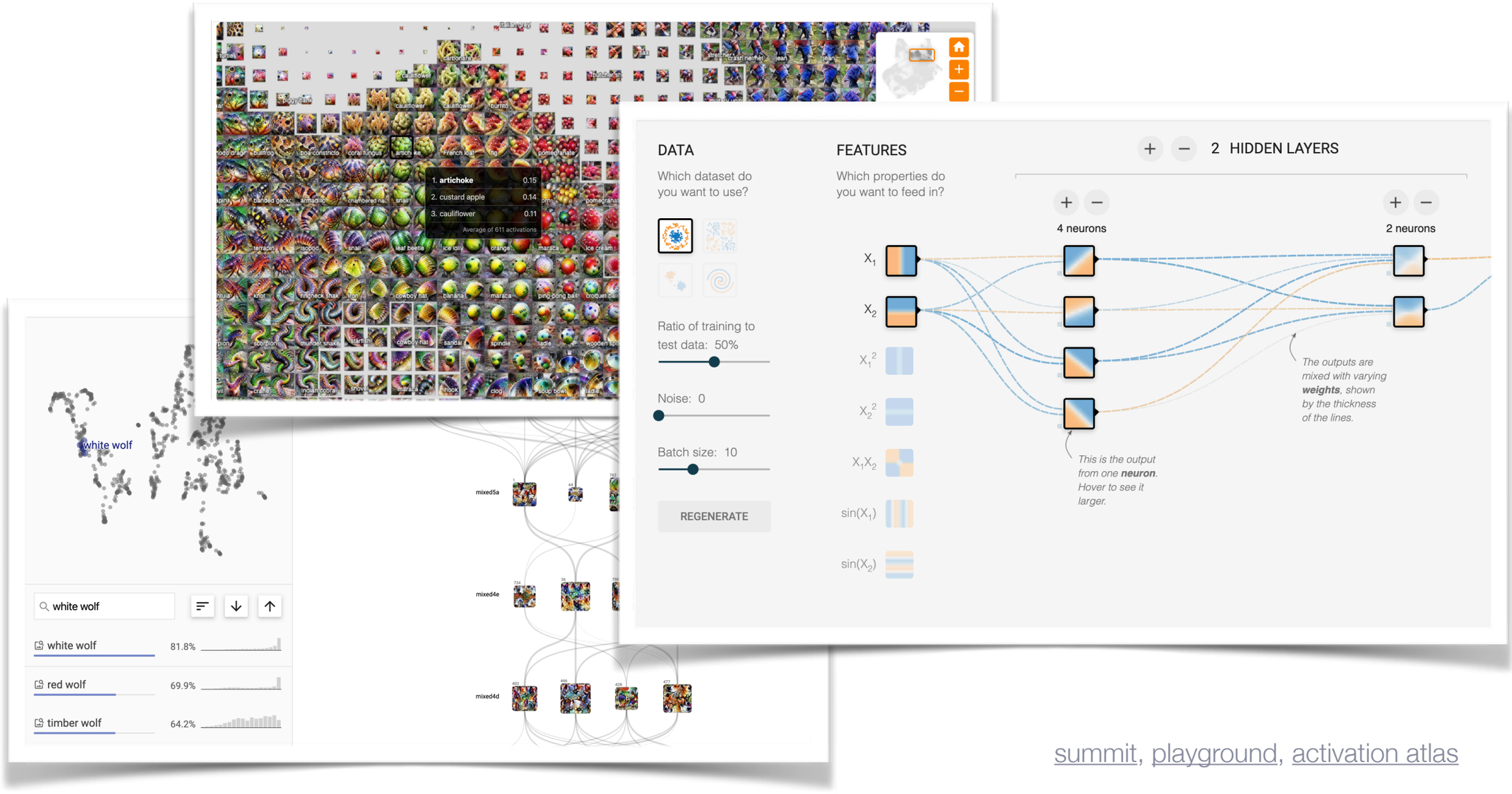


1.	**grey whale**	**91.0%**
2.	killer whale	7.5%
3.	great white shark	0.7%
4.	gar	0.4%
5.	sea lion	0.1%
6.	tiger shark	0.1%

1.	**great white shark**	**66.7%**
2.	baseball	7.4%
3.	grey whale	4.1%
4.	sombrero	3.2%
5.	sea lion	3.1%
6.	killer whale	2.7%

objective + motivations

- create an educational/exploratory tool that incorporates visualizing aspects of neural networks



summit, playground, activation atlas

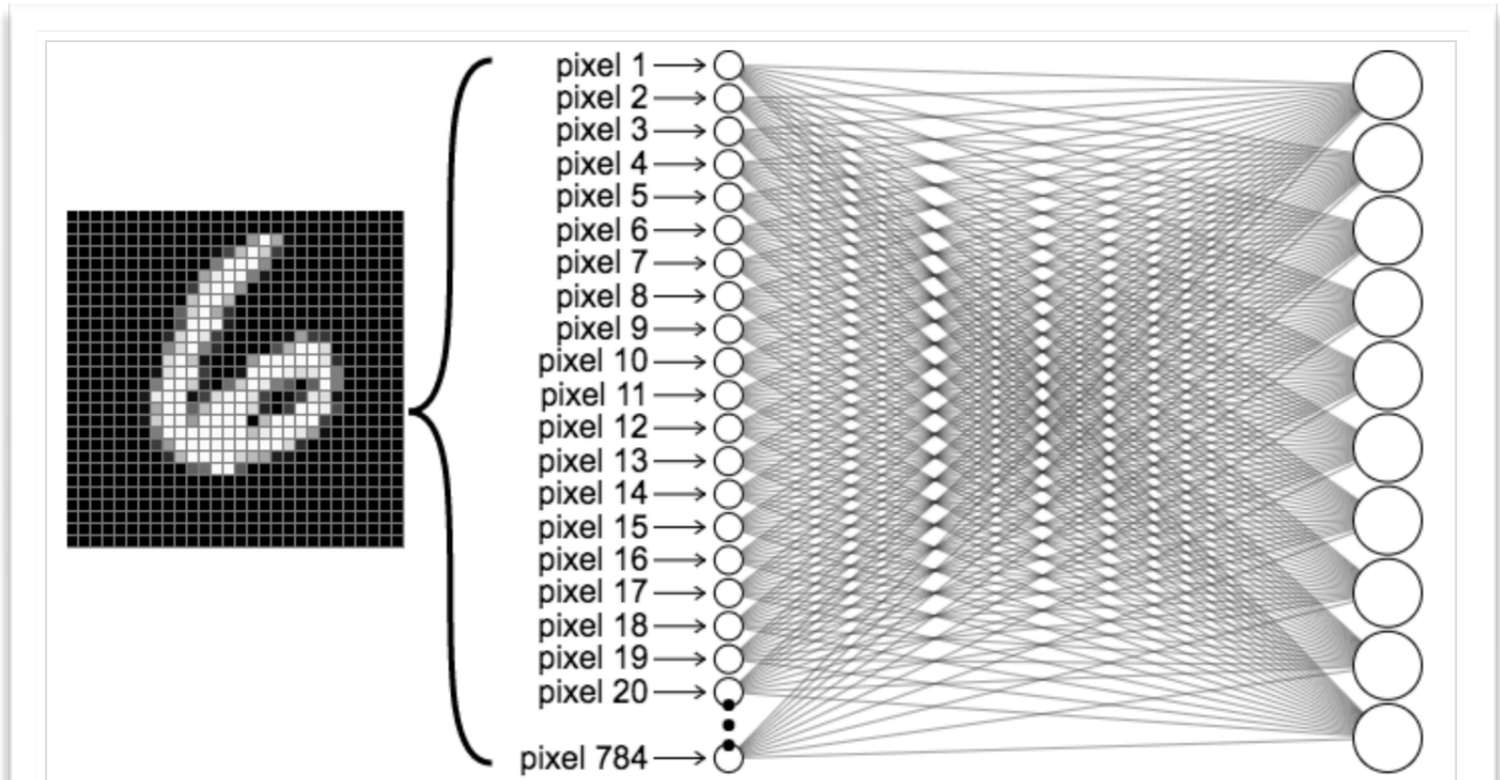
the data

- the classic MNIST handwritten digit dataset

- 60,000+ images of 28x28 pixel images, labelled and ready to rumble

the data cont.

- we can “unroll” these images into 784 element vectors which we then feed into a model - each pixel becomes an input “feature.”



1-layer neural network for MNIST. The 10 output neurons correspond to our classes, the 10 digits from 0 to 9.

https://ml4a.github.io/ml4a/looking_inside_neural_nets/

the vis

- consists of 3 parts

    1. building the model

    2. training the model

    3. exploring the model

building the model

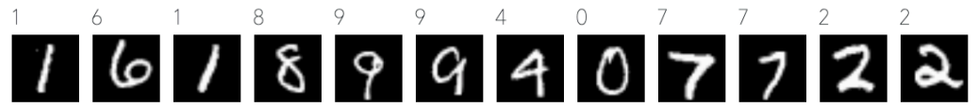
1. select model type
2. adjust parameters
3. initialize the model to examine architecture

MNIST exploration with neural nets

Explore the MNIST handwritten data set using neural nets.

the data

A collection of 60,000+ 28x28 pixel images of labeled, handwritten digits.



build & train model

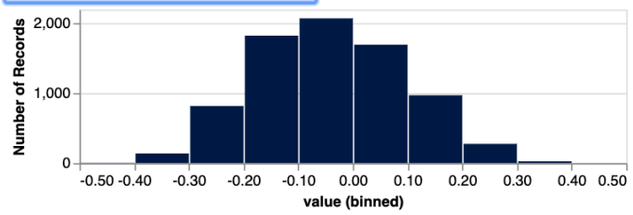
First select the type of neural net you would like to use. Tweak model parameters to see what effect they have on the model. Initialize the model to examine the architecture, then train and explore.

model architecture

Layer Name	Output Shape	# Of Params	Trainable
flatten_Flatten1	[batch,784]	0	true
dense_Dense1	[batch,10]	7,850	true

Weight Name	Shape	Min	Max	# Params	# Zeros	# NaNs	# Infinity
dense_Dense1/kernel	[784,10]	-0.1003	0.1003	7,840	0	0	0
dense_Dense1/bias	[10]	0	0	10	10	0	0

Show Values Distribution for: dense_Dense1/kernel



select parameters

basic | convolution

initialize

train

learning rate: 0.01

size of training set: 5500

epochs: 5

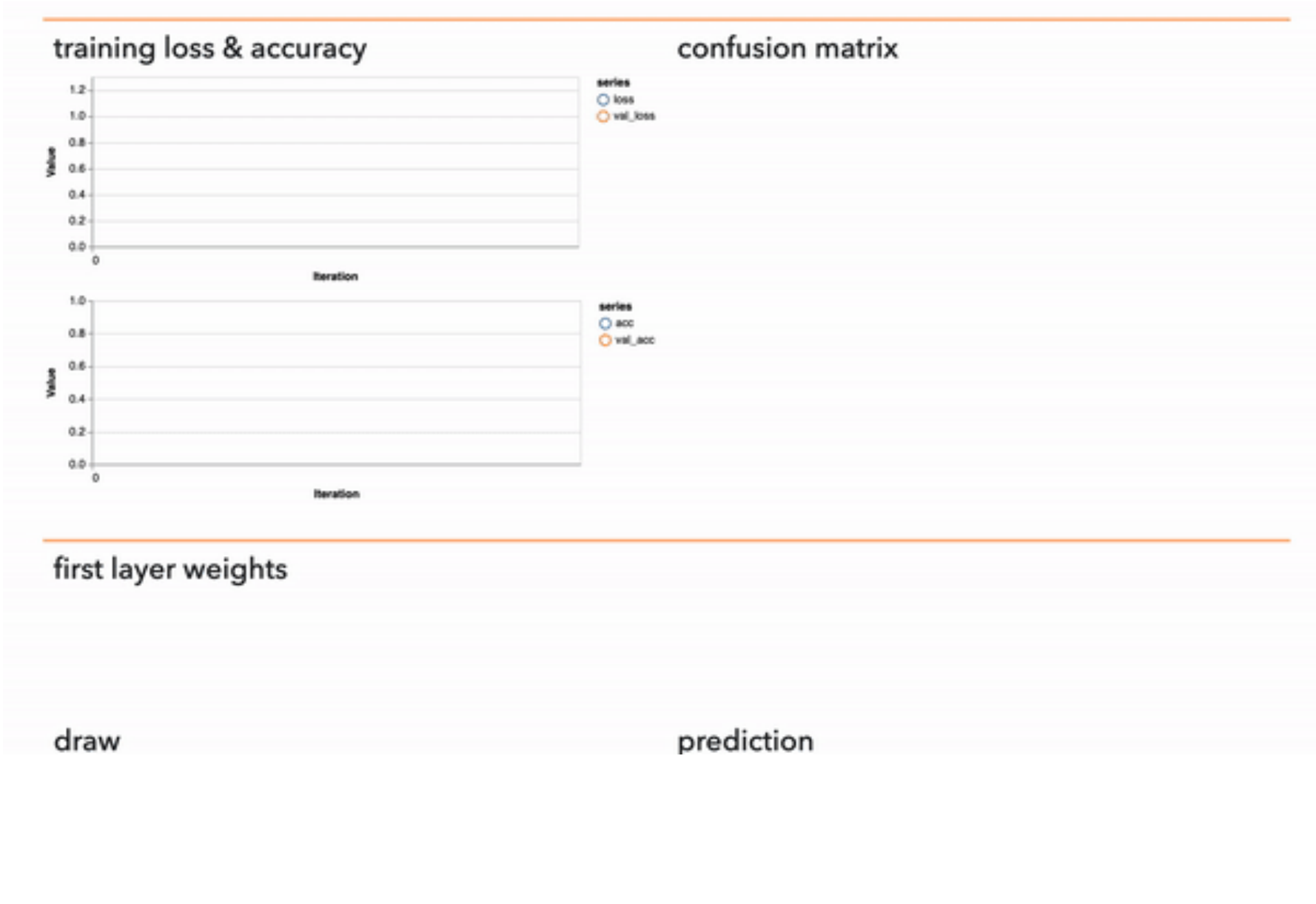
size of test set: 1000

batch size: 512

number of layers: 1

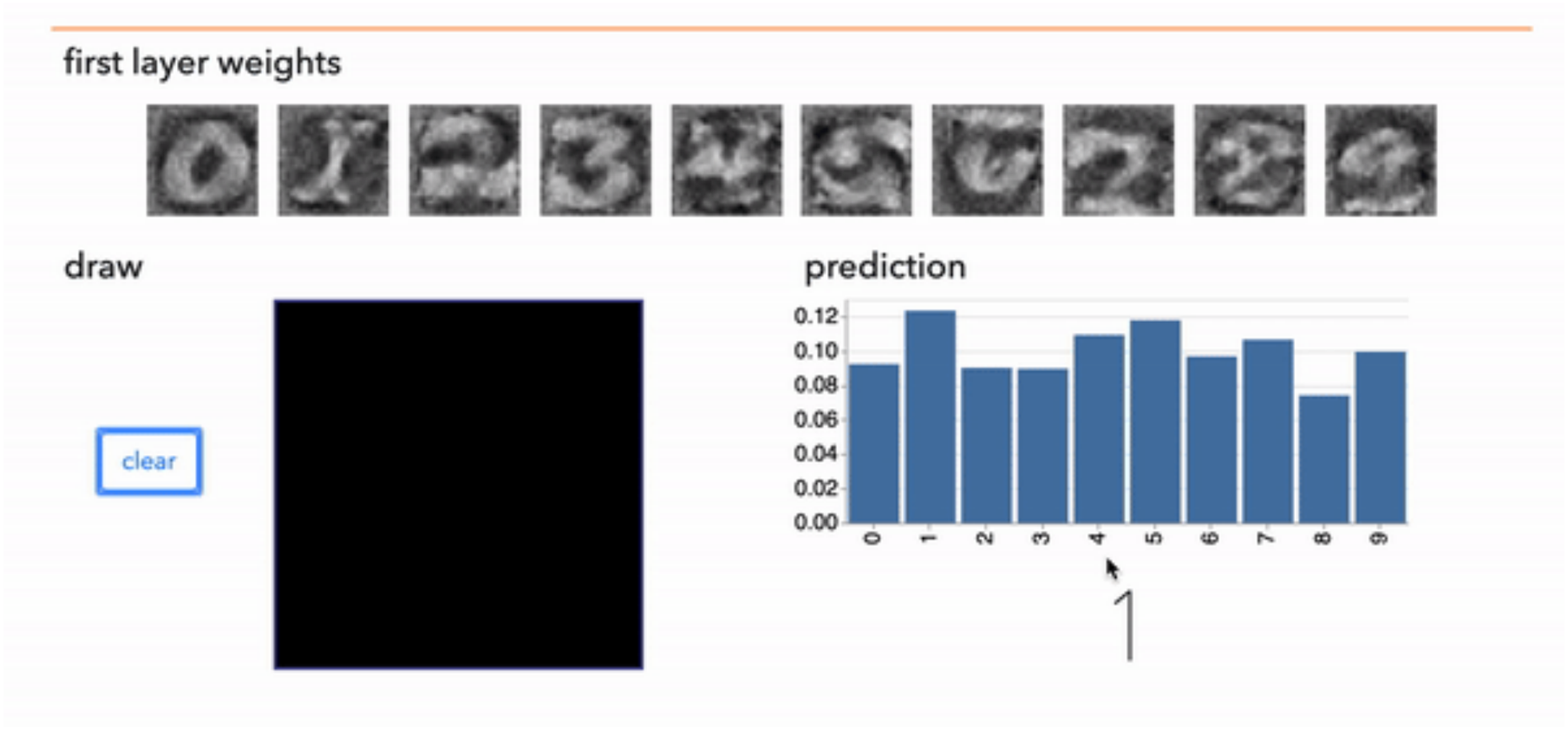
training the model

- select ‘train’ and watch stuff happen



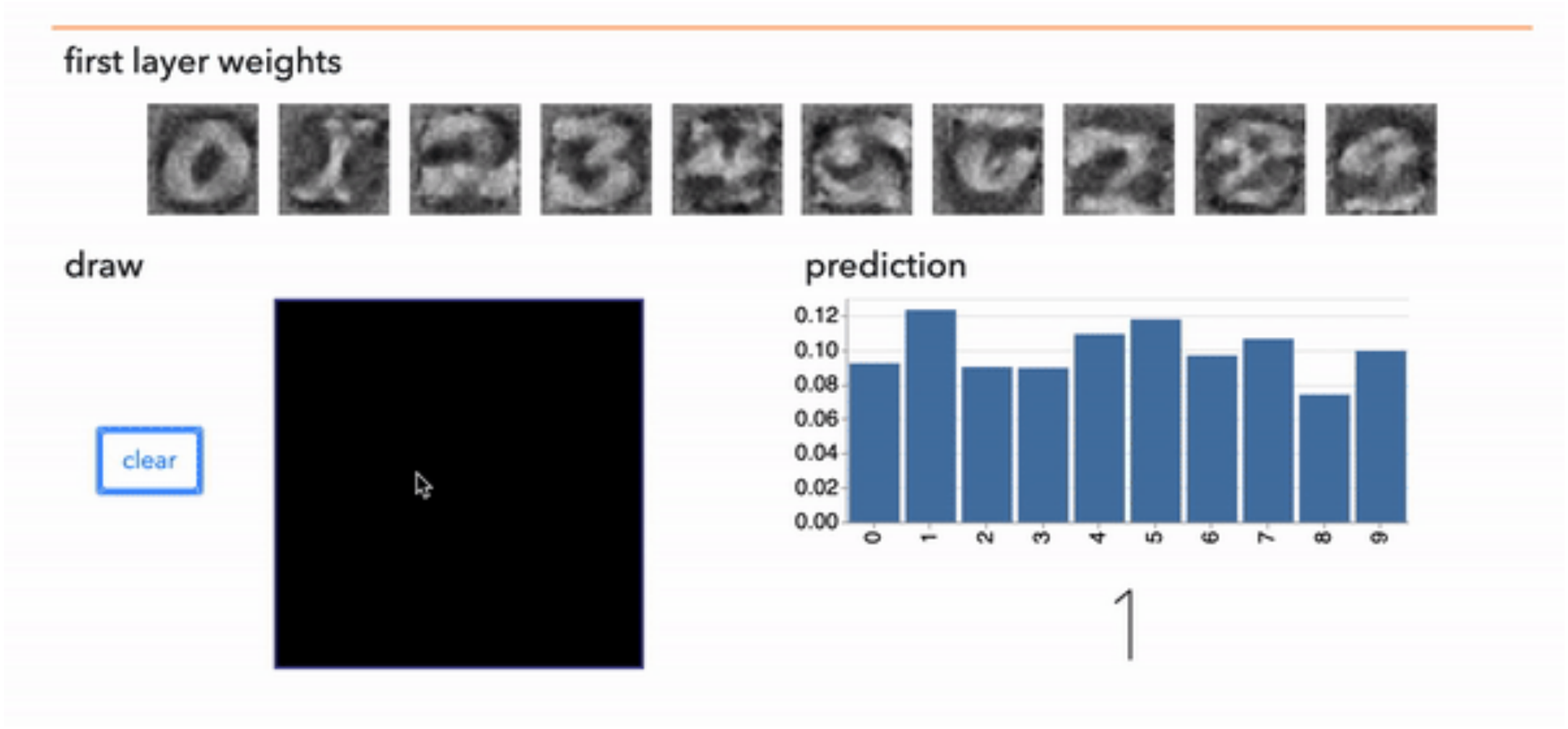
exploring the model

- user can draw own digit in the browser to see how the model performs



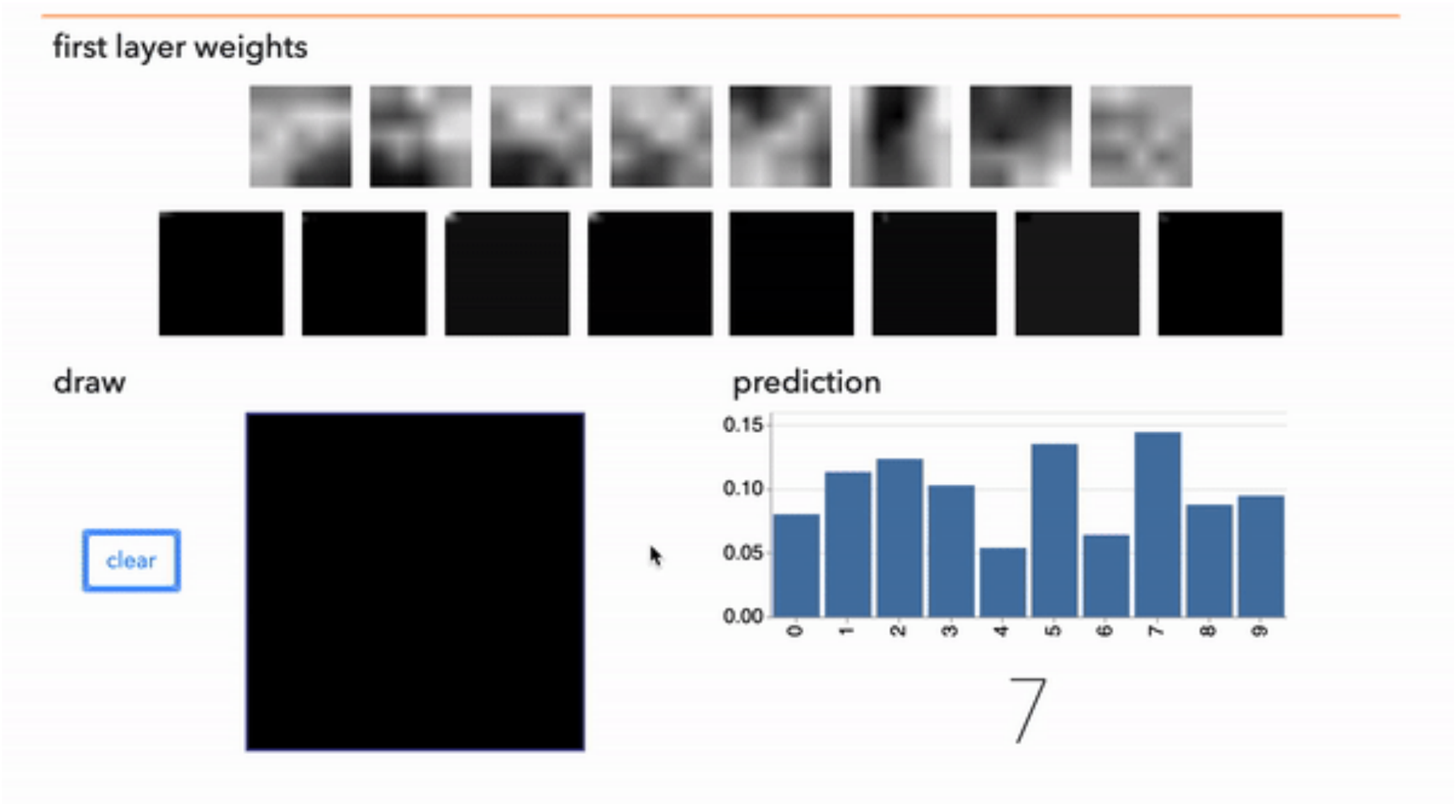
adversarial examples

- observing the weights of a 1-layer network can show user potential ways to create adversarial inputs



exploring a convolutional model

- If the user trains a convolutional model, filters are visualized along with corresponding activation maps of user input



demo

- Vis deployed here: https://mkcyoung.github.io/ML-vis/

evaluation + future directions

- useful as a supplemental tool, but perhaps not as a stand-alone educational experience

- Future directions:
  - Give the user the ability to visualize more aspects of the network, not just the first layer.
  - Add a tutorial

Questions?