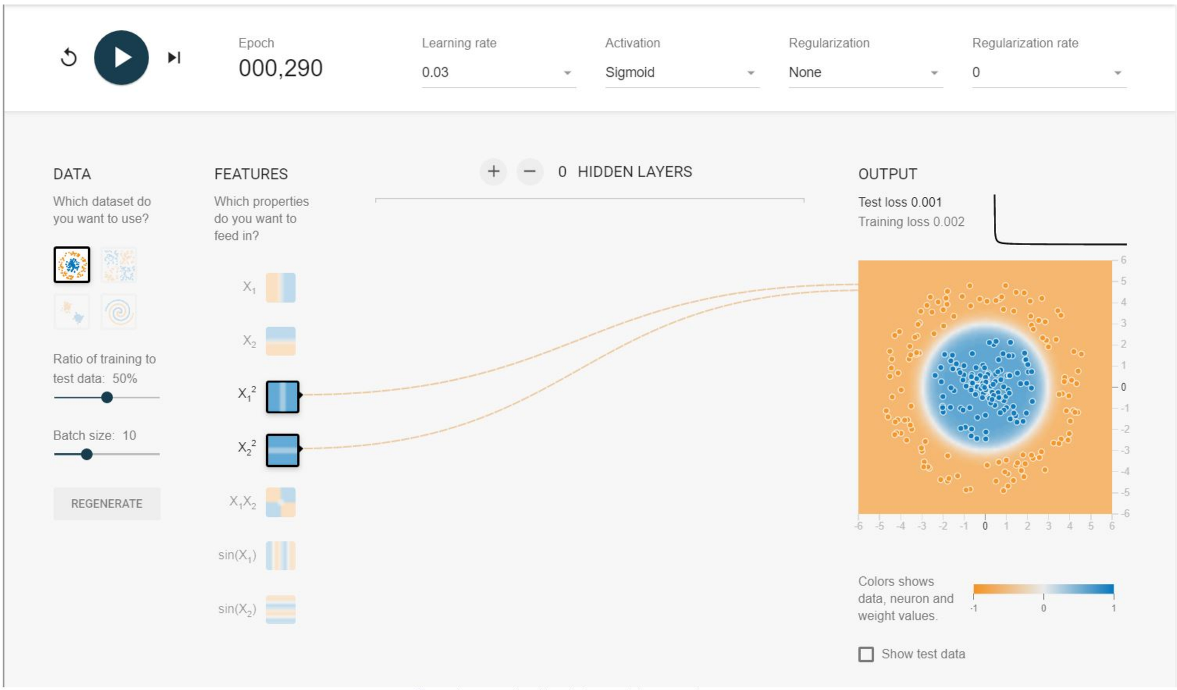
ML HW4

1.1 Hand-crafted Feature Engineering

Dataset	Selected Features	Iterations	Test Loss
Circle	x_1^2, x_2^2	290	0.001
Exclusive or	$x_1 x_2$	578	0.001
Gaussian	x_1, x_2	57	0.001
Spiral	x_2	1	0.465



Epoch 000,290

Learning rate 0.03

Activation Sigmoid

Regularization None

Regularization rate 0

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

0 HIDDEN LAYERS

OUTPUT

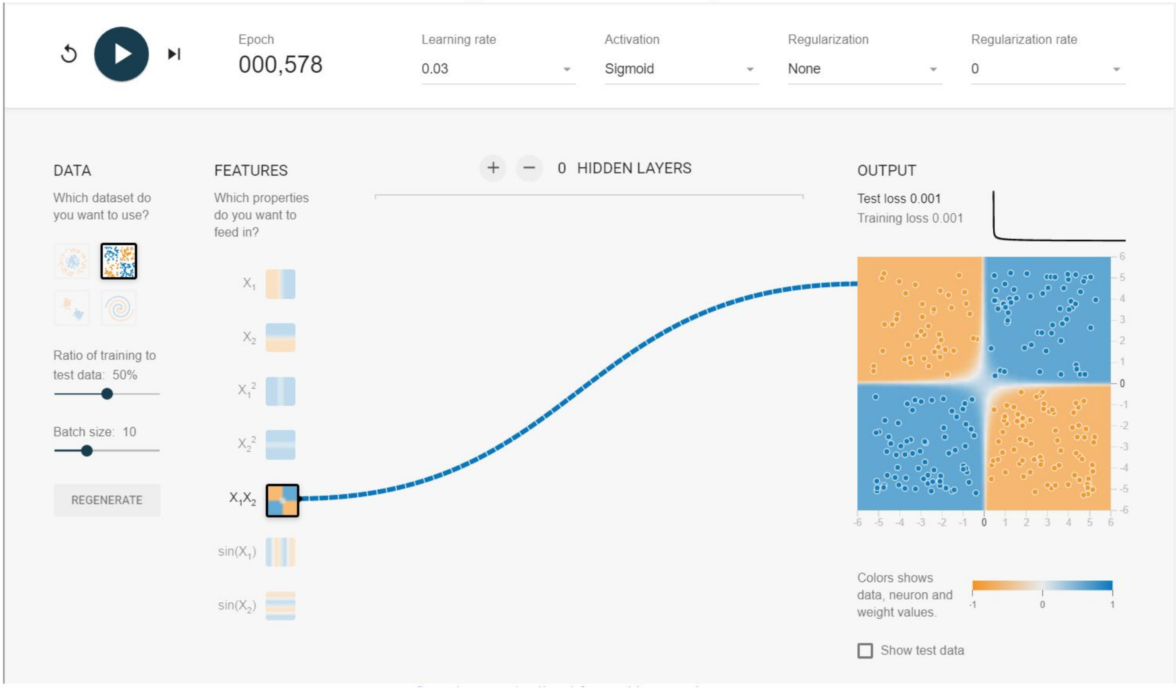
Test loss 0.001

Training loss 0.002

Colors shows data, neuron and weight values.

□ Show test data

[Tensorflow Playground](), tailored for Machine Learning Course at UVa.



Epoch 000,578

Learning rate 0.03

Activation Sigmoid

Regularization None

Regularization rate 0

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

0 HIDDEN LAYERS

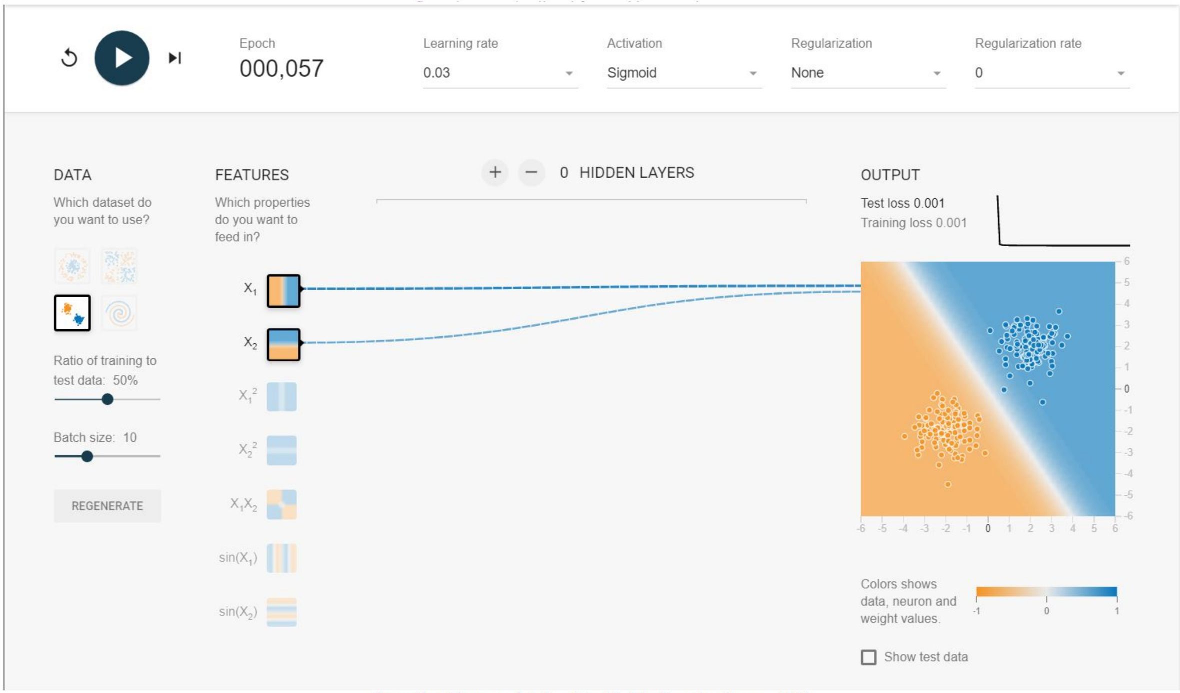
OUTPUT

Test loss 0.001

Training loss 0.001

Colors shows data, neuron and weight values.

□ Show test data



Epoch 000,057

Learning rate 0.03

Activation Sigmoid

Regularization None

Regularization rate 0

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

0 HIDDEN LAYERS

OUTPUT

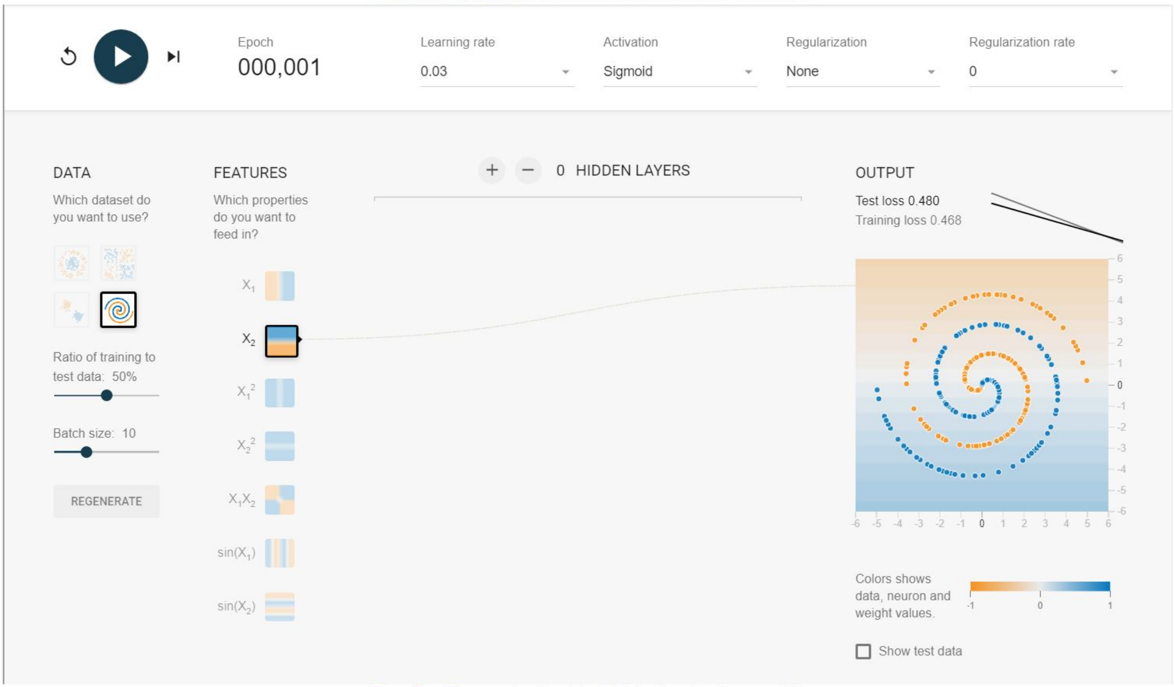
Test loss 0.001

Training loss 0.001

Colors shows data, neuron and weight values.

□ Show test data

[Tensorflow Playground](), tailored for Machine Learning Course at UVa.



Epoch 000,001

Learning rate 0.03

Activation Sigmoid

Regularization None

Regularization rate 0

Ratio of training to test data: 50%

Batch size: 10

0 HIDDEN LAYERS

Test loss 0.480

Training loss 0.468

Tensorflow Playground, tailored for Machine Learning Course at UVa.

Interpretation

For the circle, x_1^2 and x_2^2 work because we can infer the distance between the data point from the origin, which determines the label.

For xor, $x_1 x_2$ works well for classifying the data because the label is essentially the sign of $x_1 x_2$

For the Gaussian dataset, $x1$ and $x2$ works because we just need a linear separation for this dataset.

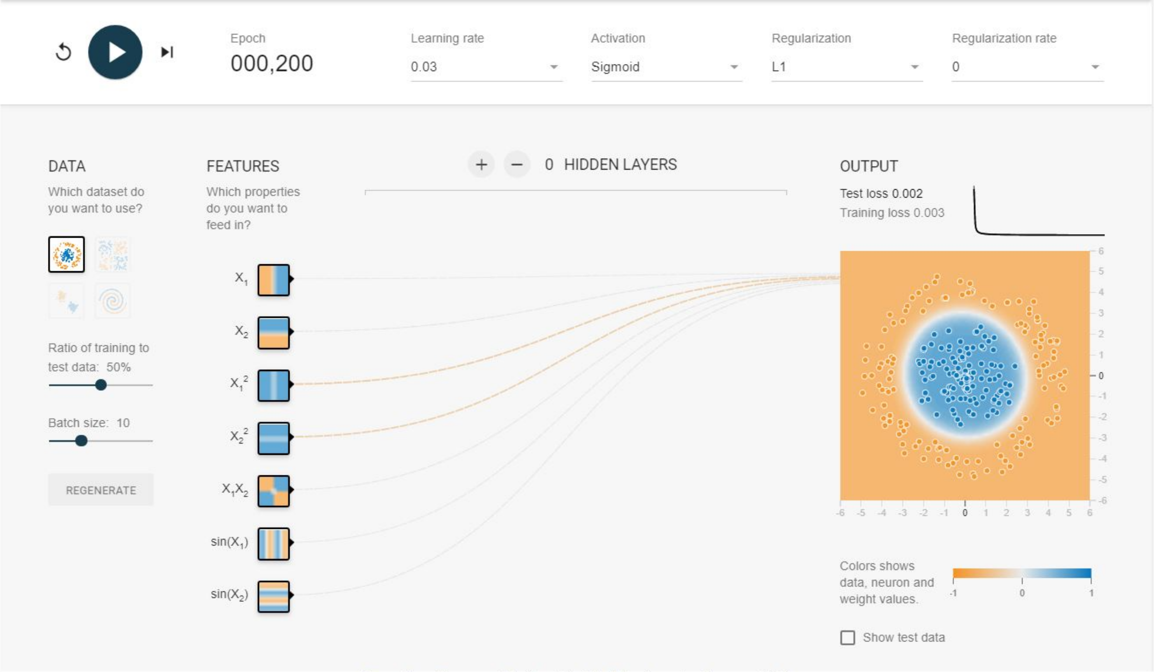
For the spiral dataset, I couldn't find a good selection of features with 0 hidden layers that yields a good classification.

1.2 Regularization

Task A

Circle

HW DNN - Question 1



Epoch 000,200

Learning rate: 0.03

Activation: Sigmoid

Regularization: L1

Regularization rate: 0

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

+ − 0 HIDDEN LAYERS

OUTPUT

Test loss 0.002

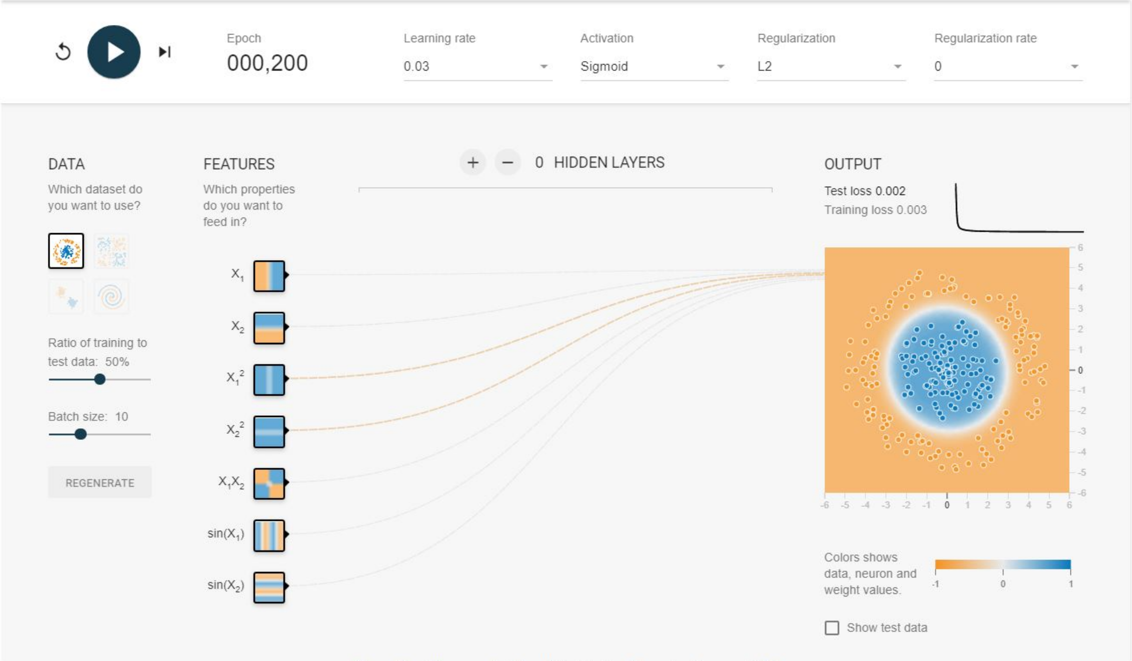
Training loss 0.003

Colors shows data, neuron and weight values.

□ Show test data

Tensorflow Playground, tailored for Machine Learning Course at UVa.

HW DNN - Question 1



Epoch 000,200

Learning rate: 0.03

Activation: Sigmoid

Regularization: L2

Regularization rate: 0

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

+ − 0 HIDDEN LAYERS

OUTPUT

Test loss 0.002

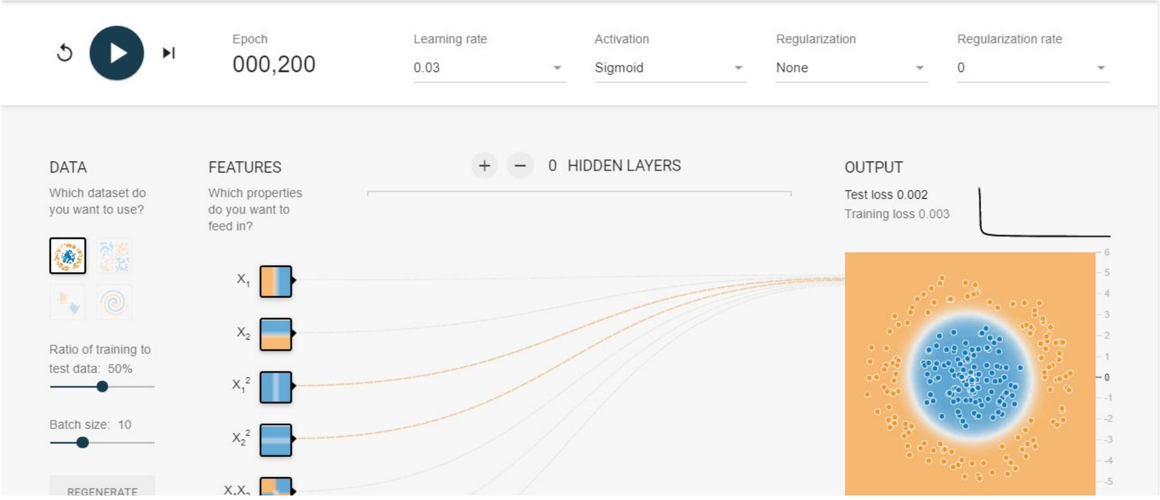
Training loss 0.003

Colors shows data, neuron and weight values.

□ Show test data

Tensorflow Playground, tailored for Machine Learning Course at UVa.

HW DNN - Question 1



Epoch 000,200

Learning rate: 0.03

Activation: Sigmoid

Regularization: None

Regularization rate: 0

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

REGENERATE

FEATURES

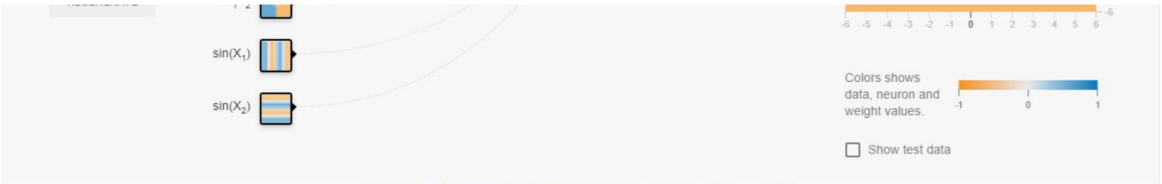
Which properties do you want to feed in?

+ − 0 HIDDEN LAYERS

OUTPUT

Test loss 0.002

Training loss 0.003

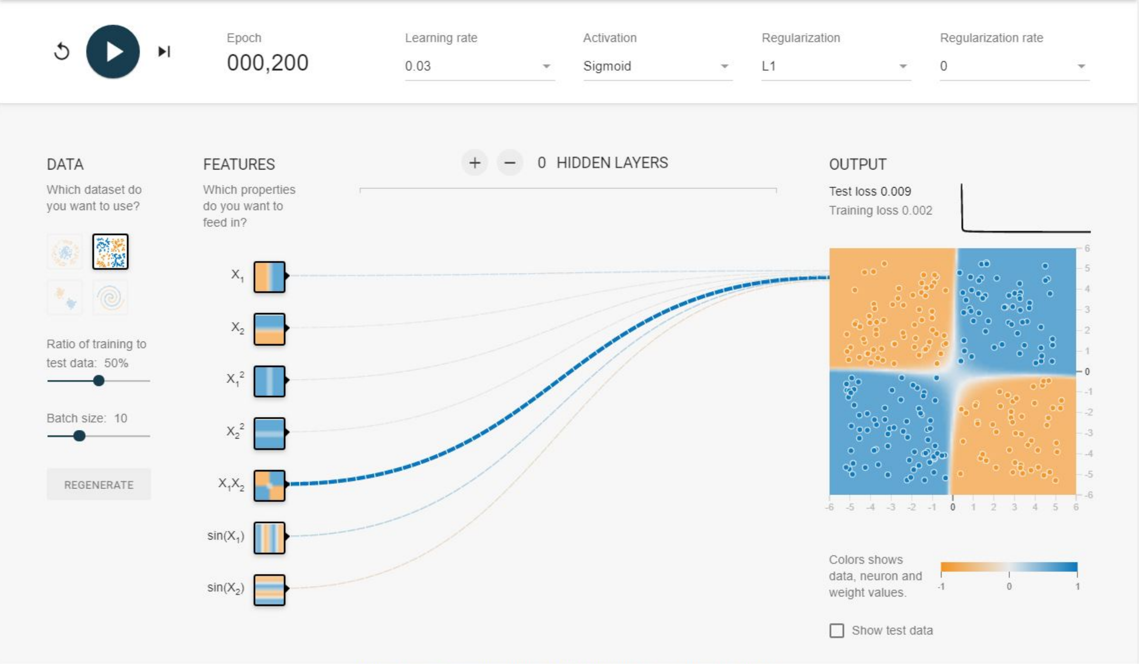


Tensorflow Playground, tailored for Machine Learning Course at UVa.

With the circle dataset, there isn't much of a difference between the three.

Exclusive or

HW DNN - Question 1



Epoch 000,200

Learning rate 0.03

Activation Sigmoid

Regularization L1

Regularization rate 0

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

+ − 0 HIDDEN LAYERS

OUTPUT

Test loss 0.009

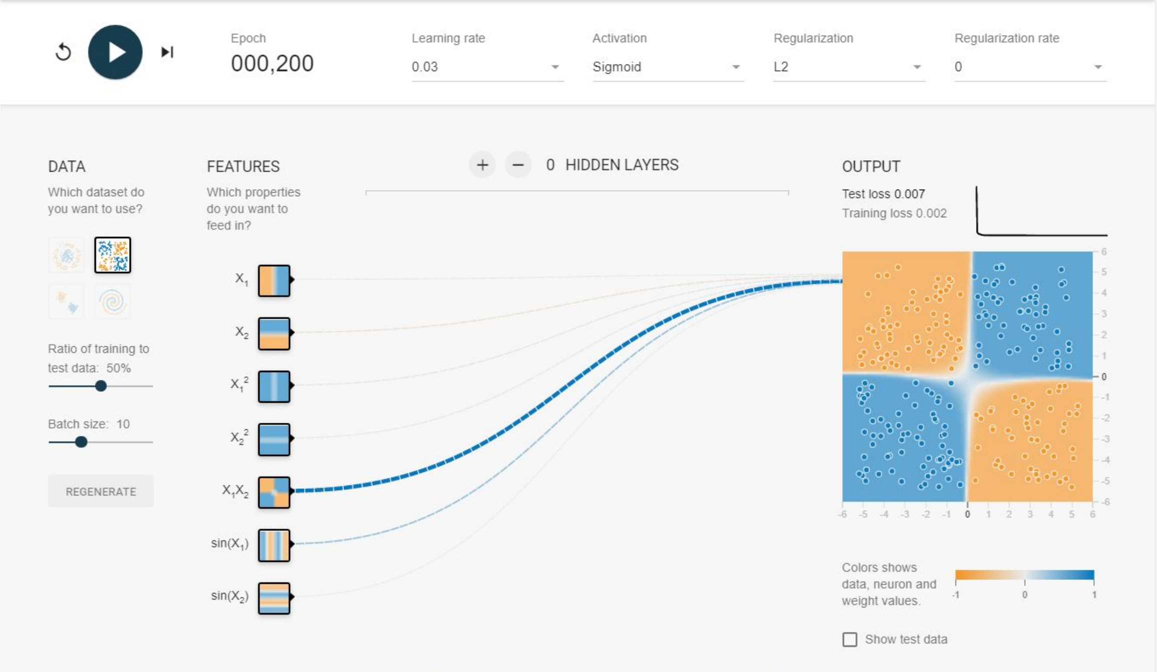
Training loss 0.002

Colors shows data, neuron and weight values.

□ Show test data

[Tensorflow Playground](), tailored for Machine Learning Course at UVa.

HW DNN - Question 1



Epoch 000,200

Learning rate 0.03

Activation Sigmoid

Regularization L2

Regularization rate 0

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

+ − 0 HIDDEN LAYERS

OUTPUT

Test loss 0.007

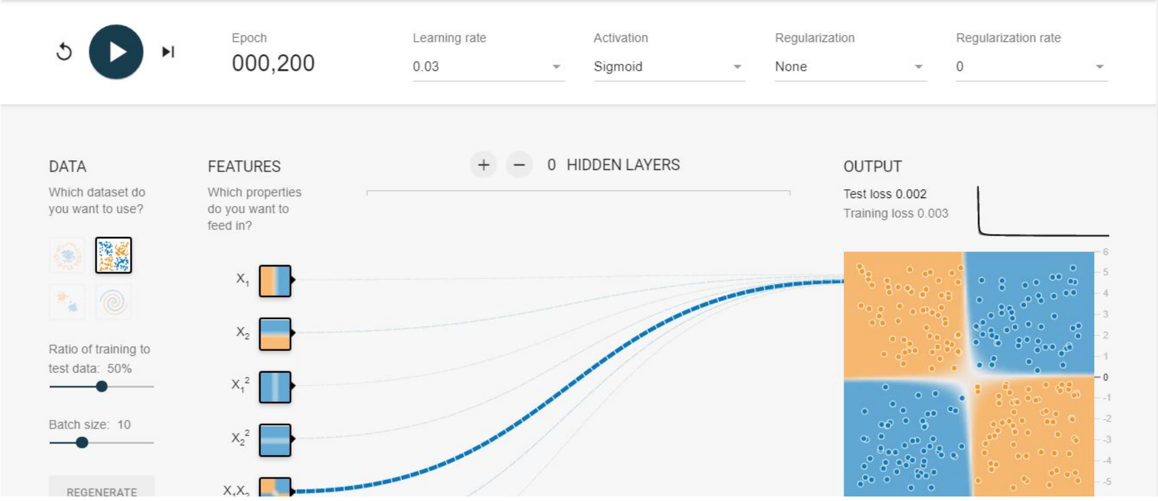
Training loss 0.002

Colors shows data, neuron and weight values.

□ Show test data

[Tensorflow Playground](), tailored for Machine Learning Course at UVa.

HW DNN - Question 1



Epoch 000,200

Learning rate 0.03

Activation Sigmoid

Regularization None

Regularization rate 0

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

REGENERATE

FEATURES

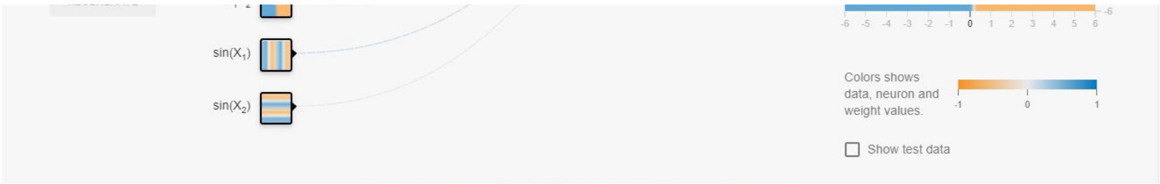
Which properties do you want to feed in?

+ − 0 HIDDEN LAYERS

OUTPUT

Test loss 0.002

Training loss 0.003

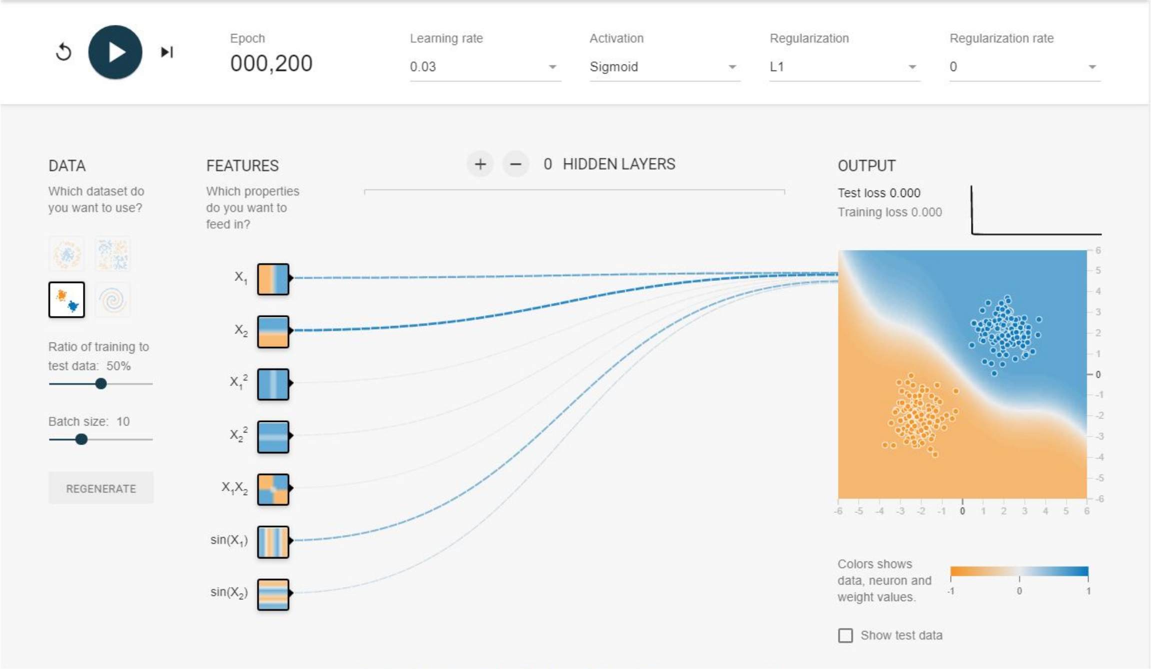


Tensorflow Playground, tailored for Machine Learning Course at UVa.

With the exclusive or dataset, different regularization methods led to slightly different decision boundaries.

Gaussian

HW DNN - Question 1



Epoch 000,200

Learning rate 0.03

Activation Sigmoid

Regularization L1

Regularization rate 0

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

FEATURES

Which properties do you want to feed in?

0 HIDDEN LAYERS

OUTPUT

Test loss 0.000

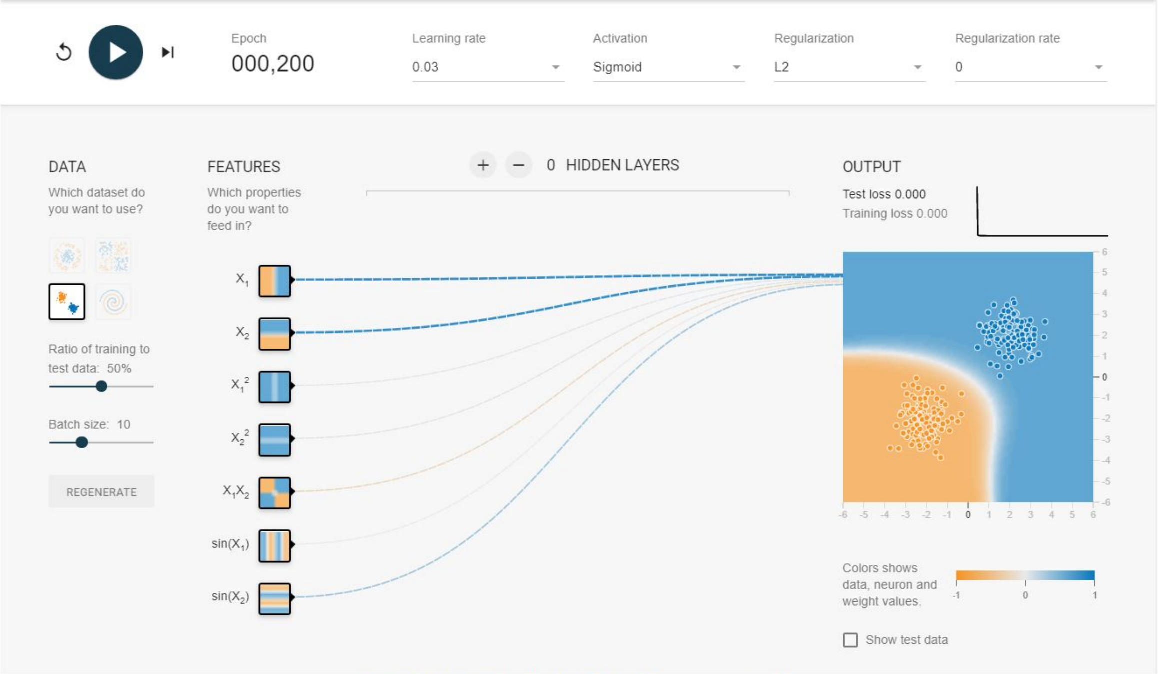
Training loss 0.000

Colors shows data, neuron and weight values.

□ Show test data

[Tensorflow Playground](), tailored for Machine Learning Course at UVa.

HW DNN - Question 1



Epoch 000,200

Learning rate 0.03

Activation Sigmoid

Regularization L2

Regularization rate 0

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

FEATURES

Which properties do you want to feed in?

0 HIDDEN LAYERS

OUTPUT

Test loss 0.000

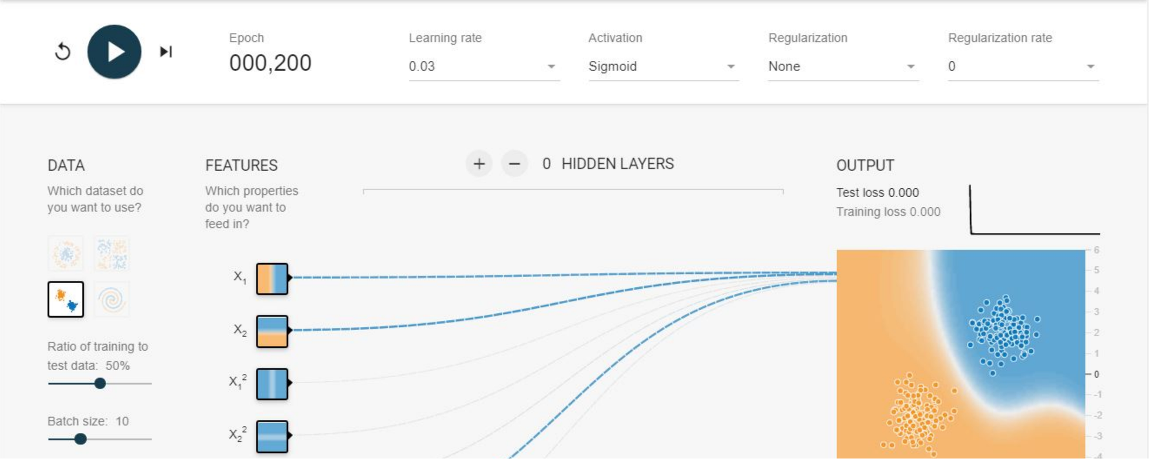
Training loss 0.000

Colors shows data, neuron and weight values.

□ Show test data

[Tensorflow Playground](), tailored for Machine Learning Course at UVa.

HW DNN - Question 1



Epoch 000,200

Learning rate 0.03

Activation Sigmoid

Regularization None

Regularization rate 0

DATA

Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

FEATURES

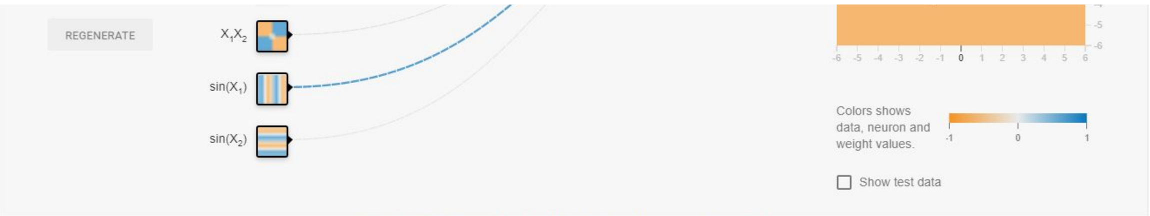
Which properties do you want to feed in?

0 HIDDEN LAYERS

OUTPUT

Test loss 0.000

Training loss 0.000



Tensorflow Playground, tailored for Machine Learning Course at UVa.

Interpretation

These configurations work because regularization helps filter out unimportant features by giving them lower weights.

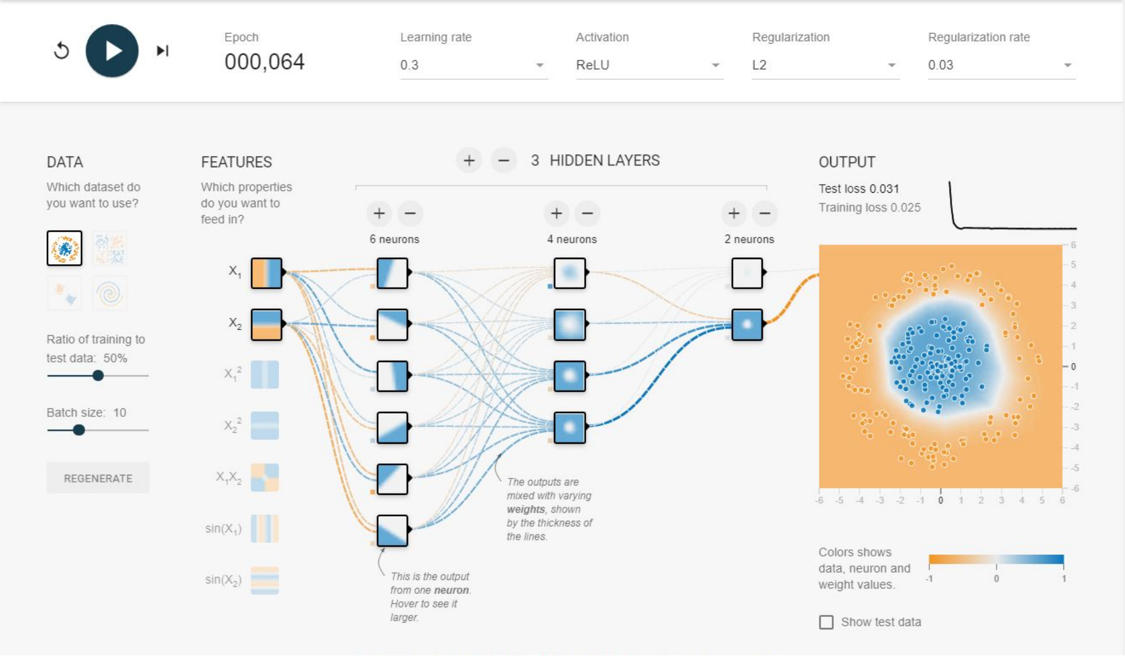
Task B

Yes. But there were some features that I didn't select that still got a little bit of weight.

1.3 Automated Feature Engineering with Neural Network

Circle

HW DNN - Question 1



Epoch 000,064
Learning rate 0.3
Activation ReLU
Regularization L2
Regularization rate 0.03

Ratio of training to test data: 50%
Batch size: 10

3 HIDDEN LAYERS: 6 neurons, 4 neurons, 2 neurons

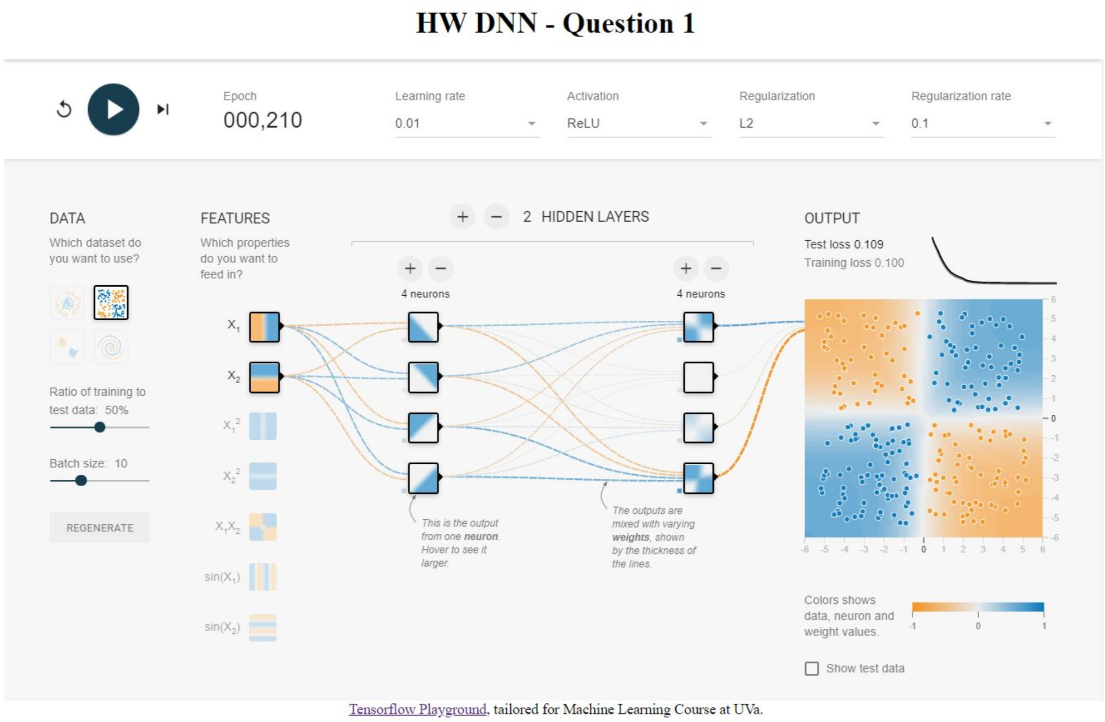
Test loss 0.031
Training loss 0.025

DATA
Which dataset do you want to use?
REGENERATE
FEATURES
Which properties do you want to feed in?
X_1
X_2
X_1^2
X_2^2
X_1X_2
$\sin(X_1)$
$\sin(X_2)$
+ −
The outputs are mixed with varying weights, shown by the thickness of the lines.
This is the output from one neuron. Hover to see it larger.
OUTPUT
-6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6
Colors shows data, neuron and weight values.
-1 0 1
□ Show test data

Tensorflow Playground, tailored for Machine Learning Course at UVa.

I used a total of 3 hidden layers, ReLU for the activation function, and L2 for regularization. This works because the first layer essentially learned boundaries at different slopes and the second layer learned the shape of a circle from those lines.

Exclusive or



HW DNN - Question 1

Epoch 000,210

Learning rate 0.01

Activation ReLU

Regularization L2

Regularization rate 0.1

DATA Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

FEATURES Which properties do you want to feed in?

2 HIDDEN LAYERS

4 neurons

4 neurons

OUTPUT

Test loss 0.109

Training loss 0.100

This is the output from one **neuron**. Hover to see it larger.

The outputs are mixed with varying **weights**, shown by the thickness of the lines.

Colors shows data, neuron and weight values.

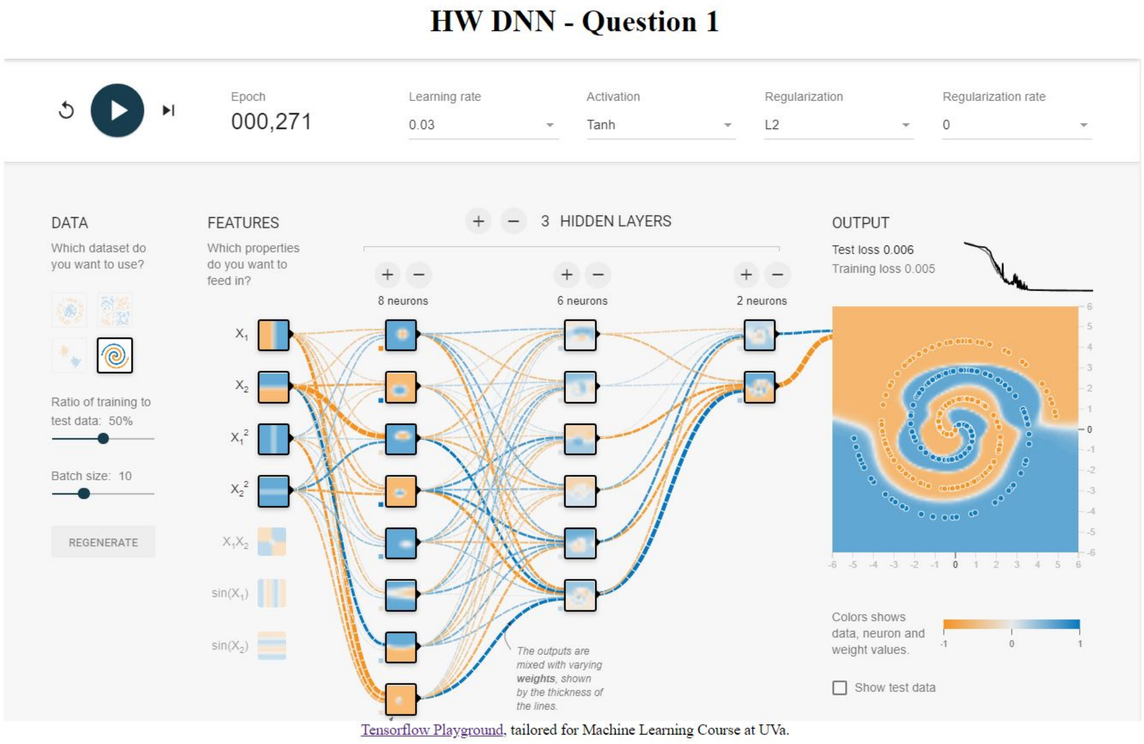
□ Show test data

Tensorflow Playground, tailored for Machine Learning Course at UVa.

Used 2 hidden layers, ReLU for activation, L2 for regularization, with a learning rate of 0.01 and the regularization rate being 0.1.

As we can see in the screen shot, this works because the first layer learned the subspaces separated by the diagonals of the space and the second layer just takes the intersections of them, which have the same shape as xor.

1.4 Spiral Challenge



HW DNN - Question 1

Epoch 000,271

Learning rate 0.03

Activation Tanh

Regularization L2

Regularization rate 0

DATA Which dataset do you want to use?

Ratio of training to test data: 50%

Batch size: 10

FEATURES Which properties do you want to feed in?

3 HIDDEN LAYERS

8 neurons

6 neurons

2 neurons

OUTPUT

Test loss 0.006

Training loss 0.005

The outputs are mixed with varying **weights**, shown by the thickness of the lines.

Colors shows data, neuron and weight values.

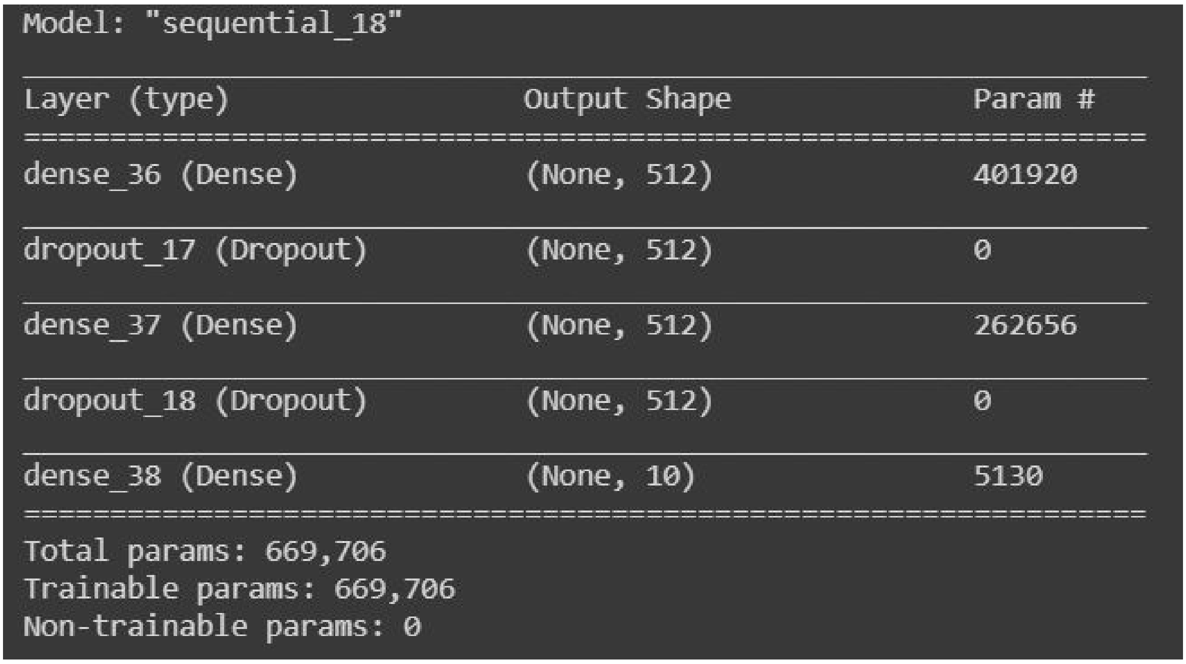
□ Show test data

Tensorflow Playground, tailored for Machine Learning Course at UVa.

My interpretation is that the first layer learned some basic/local features of the network and the second layer was then able to extract higher level features from the first layer, allowing the third layer to take advantage of that and learn features that was able to classify the data.

2.3 Multilayer Perceptron (MLP)

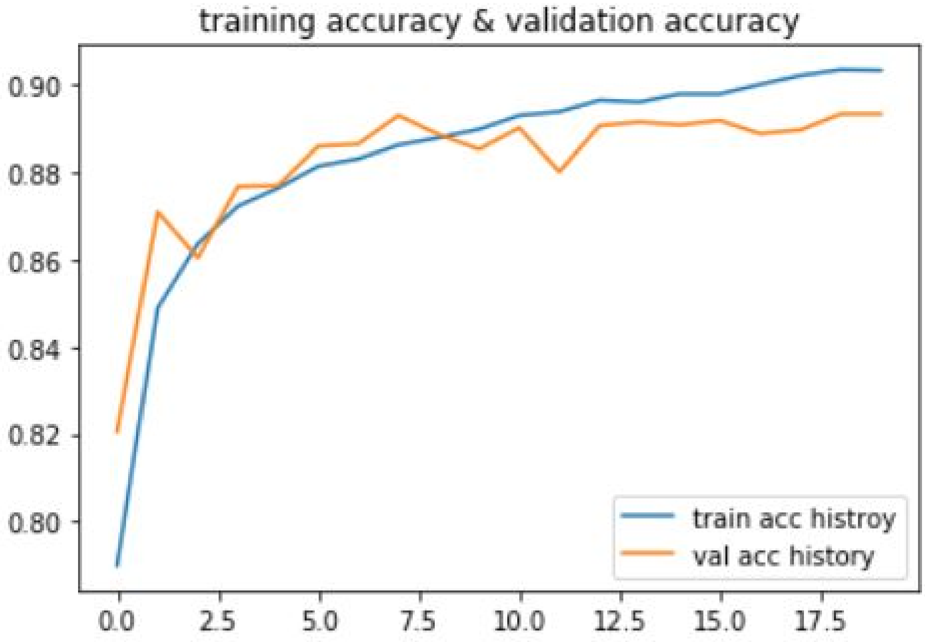
Summary

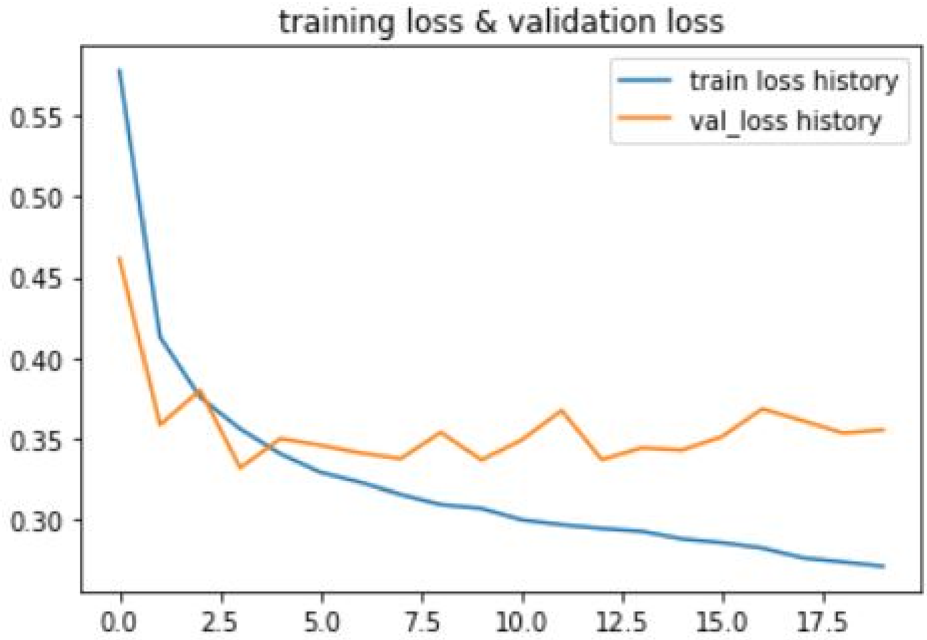


```
Model: "sequential_18"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_36 (Dense)             (None, 512)               401920
_________________________________________________________________
dropout_17 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_37 (Dense)             (None, 512)               262656
_________________________________________________________________
dropout_18 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_38 (Dense)             (None, 10)                5130
=================================================================
Total params: 669,706
Trainable params: 669,706
Non-trainable params: 0
```

Just like the previous DNN I built in tensorflow playgound, the the first layer captures some characteristics in the input image and the second layer infers higher level characteristics from the first layer. I also added dropout layers to prevent overfitting.

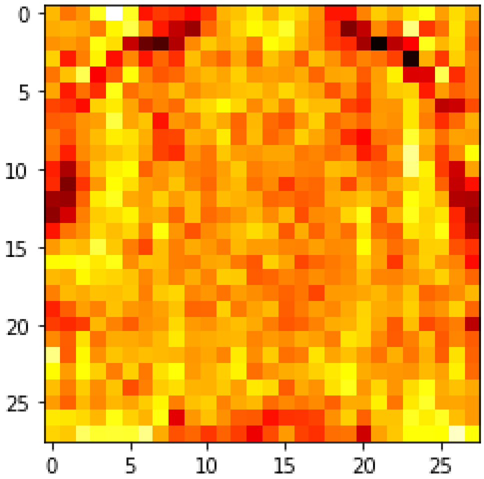
Plots

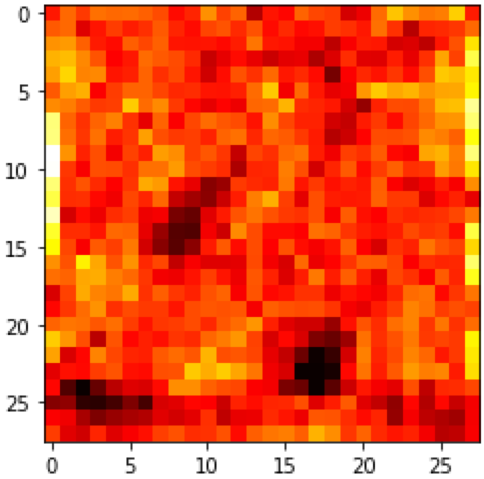




The validation accuracy being higher than the training accuracy suggests that my model might have overfit the data, meaning I reducing the number of epochs may increase the accuracy.

Weights Visualization





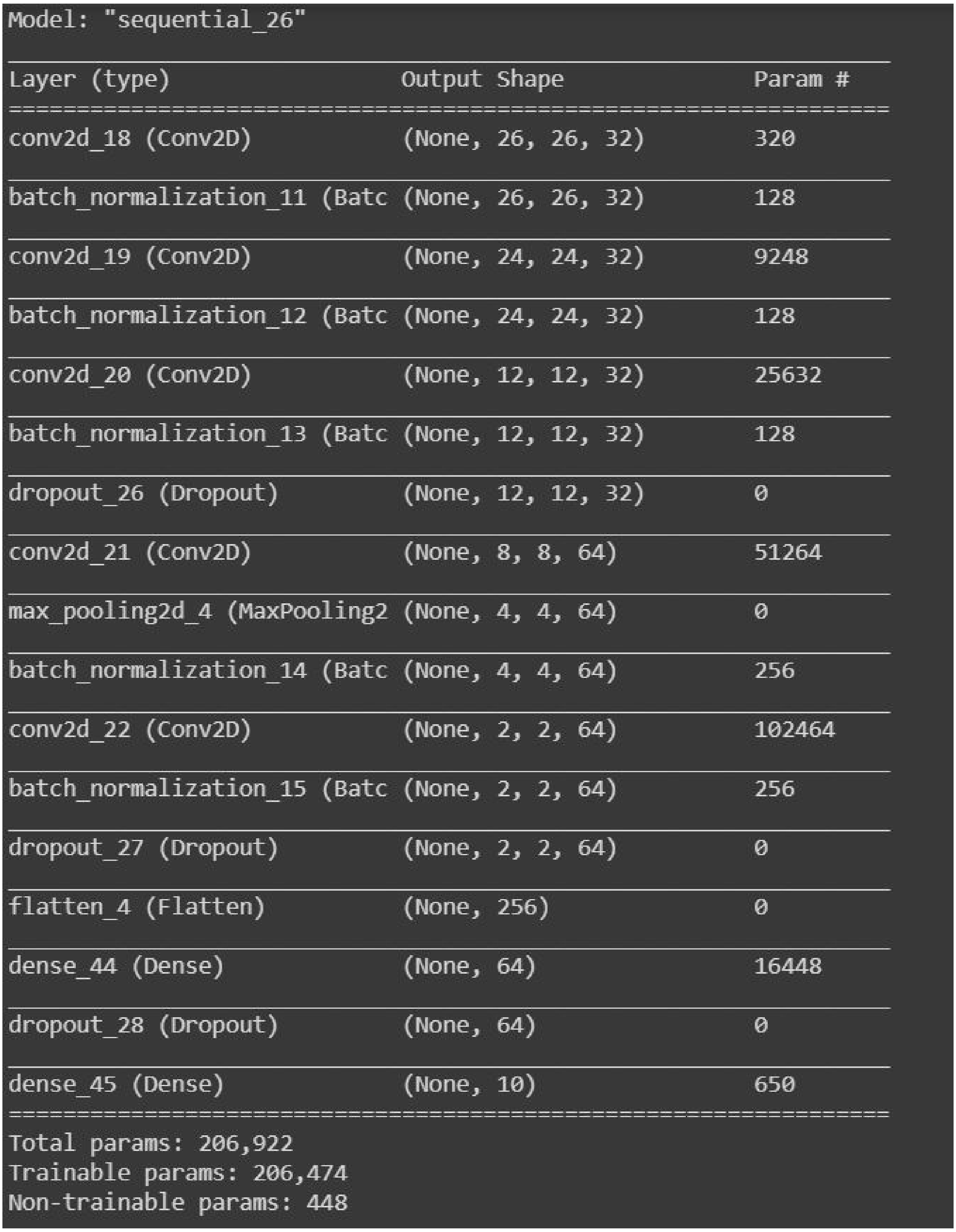
We can roughly see the outline of a piece of clothing or a top wear in the first picture and the outline of a high-heel in the second picture. This suggests the weights are capturing the certain features that help identify which class the input image is.

Why isn’ t MSE a good choice for a loss function in for this problem?

MSE doesn’ t punish misclassifications enough. We want the model to have the highest accuracy possible, so MSE isn't the best choice for classification problems.

2.4 Convolutional Neural Network (CNN)

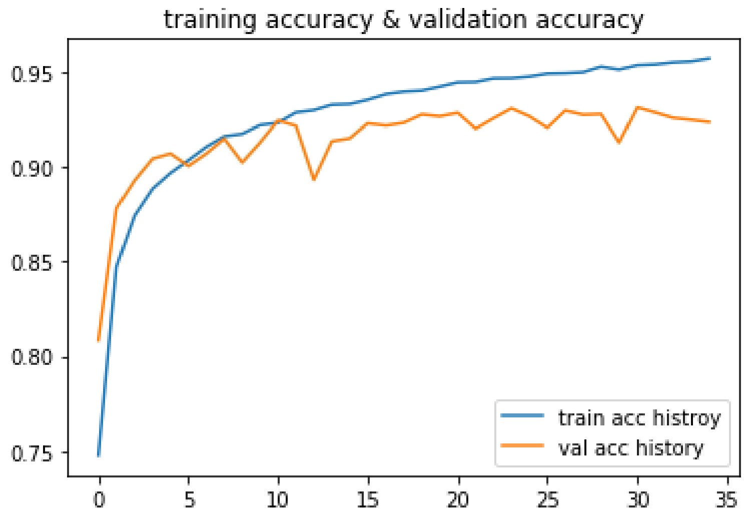
Summary

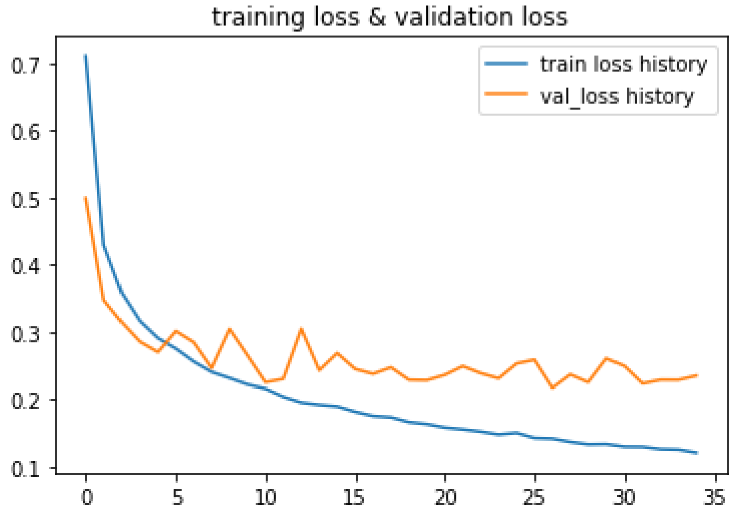


```
Model: "sequential_26"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_18 (Conv2D)           (None, 26, 26, 32)        320
_________________________________________________________________
batch_normalization_11 (Batc (None, 26, 26, 32)        128
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 24, 24, 32)        9248
_________________________________________________________________
batch_normalization_12 (Batc (None, 24, 24, 32)        128
_________________________________________________________________
conv2d_20 (Conv2D)           (None, 12, 12, 32)        25632
_________________________________________________________________
batch_normalization_13 (Batc (None, 12, 12, 32)        128
_________________________________________________________________
dropout_26 (Dropout)         (None, 12, 12, 32)        0
_________________________________________________________________
conv2d_21 (Conv2D)           (None, 8, 8, 64)          51264
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 4, 4, 64)          0
_________________________________________________________________
batch_normalization_14 (Batc (None, 4, 4, 64)          256
_________________________________________________________________
conv2d_22 (Conv2D)           (None, 2, 2, 64)          102464
_________________________________________________________________
batch_normalization_15 (Batc (None, 2, 2, 64)          256
_________________________________________________________________
dropout_27 (Dropout)         (None, 2, 2, 64)          0
_________________________________________________________________
flatten_4 (Flatten)          (None, 256)               0
_________________________________________________________________
dense_44 (Dense)             (None, 64)                16448
_________________________________________________________________
dropout_28 (Dropout)         (None, 64)                0
_________________________________________________________________
dense_45 (Dense)             (None, 10)                650
=================================================================
Total params: 206,922
Trainable params: 206,474
Non-trainable params: 448
```

I used 4 convolutional layers to reduce the size of data and extract features, 3 drop out layers to for regularization, and fully connected layers at the end.

Loss and Accuracy





Performance did not improve much on the validation data after 5 epochs despite continuous performance improvement in train accuracy. This suggests the model might've been overfitting and more iterations may not improve the accuracy of the model.

· How many matrices are outputted by your first convolutional layer when it receives a single testing image?

32

What are the dimensions of these matrices?

26×26

What are the dimensions of one of these matrices after it passes through your first maxpooling layer?

As can be seen in the summary after the first max pooling, the dimensions will be 4×4

Sample Questions

Question 1. Neural Nets and Regression

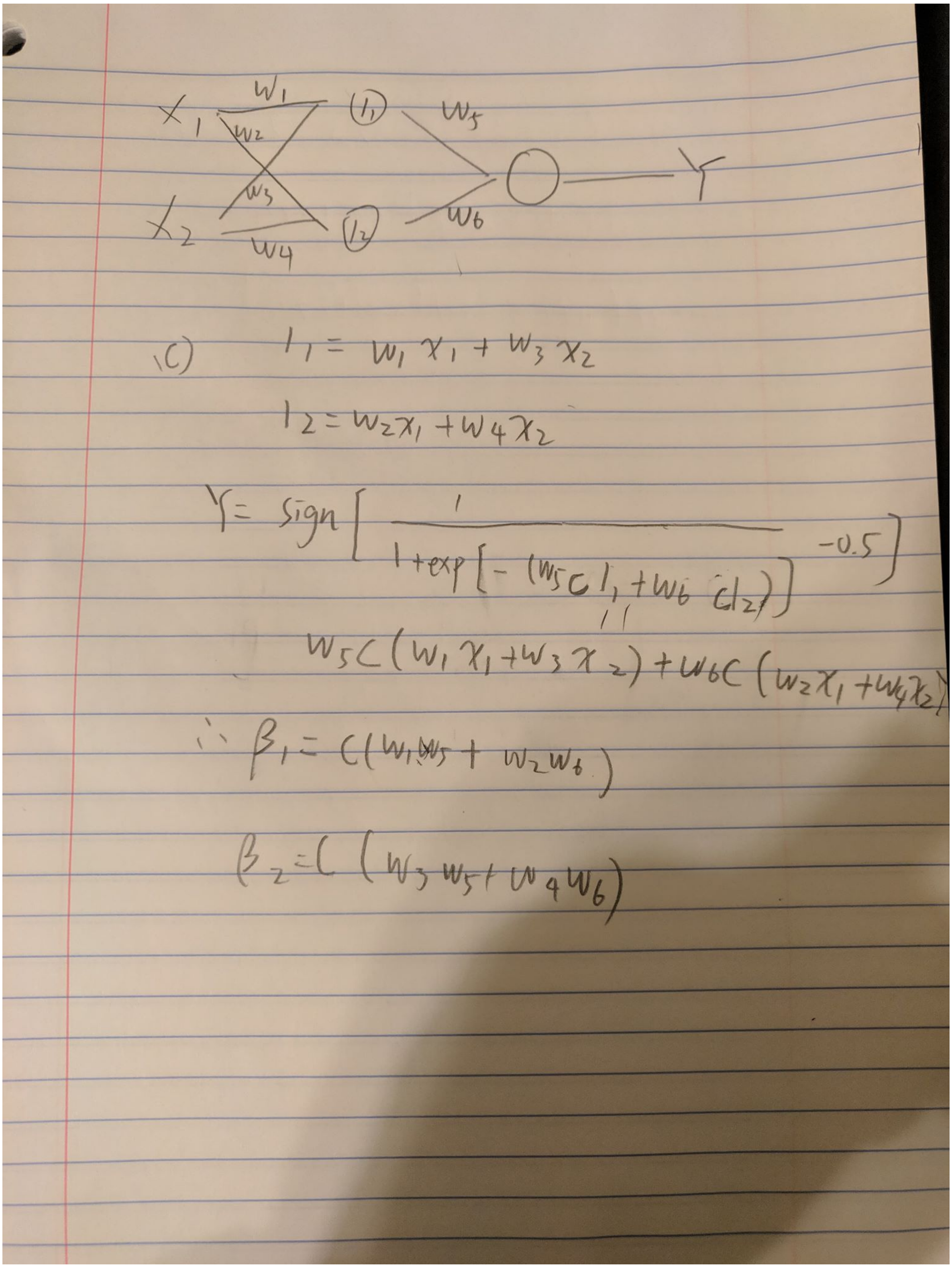
(a)

Assigned to all units L

(b)

First layer: L Last unit: S

(c)


x_1
x_2
w_1
w_2
w_3
w_4
I_1
I_2
w_5
w_6
Y

(c) $I_1 = w_1 x_1 + w_3 x_2$

$I_2 = w_2 x_1 + w_4 x_2$

$$Y = \text{sign}\left[\frac{1}{1+\exp[-(w_5 c I_1 + w_6 c I_2)]} - 0.5\right]$$

$$\| \quad w_5 c(w_1 x_1 + w_3 x_2) + w_6 c(w_2 x_1 + w_4 x_2)$$

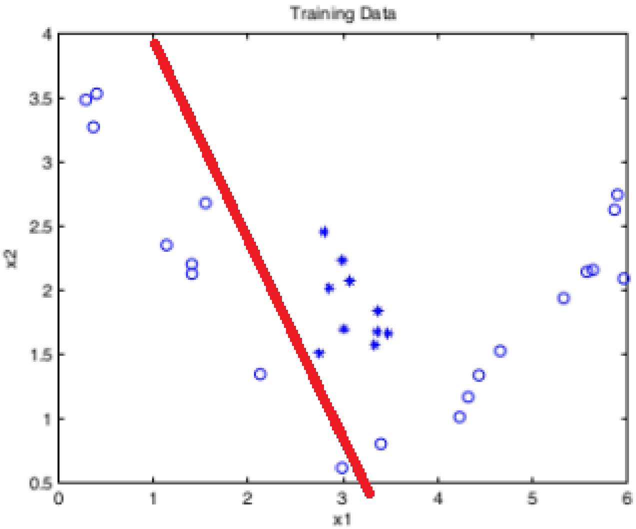
$\therefore \beta_1 = c(w_1 w_5 + w_2 w_6)$

$\beta_2 = c(w_3 w_5 + w_4 w_6)$

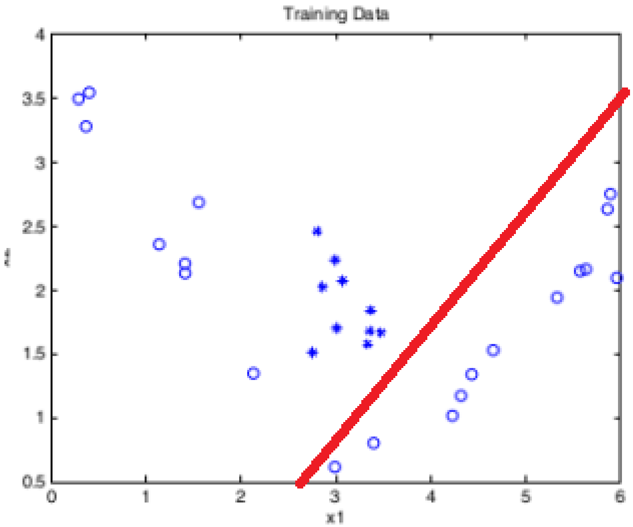
Question 2. Neural Nets

(a)

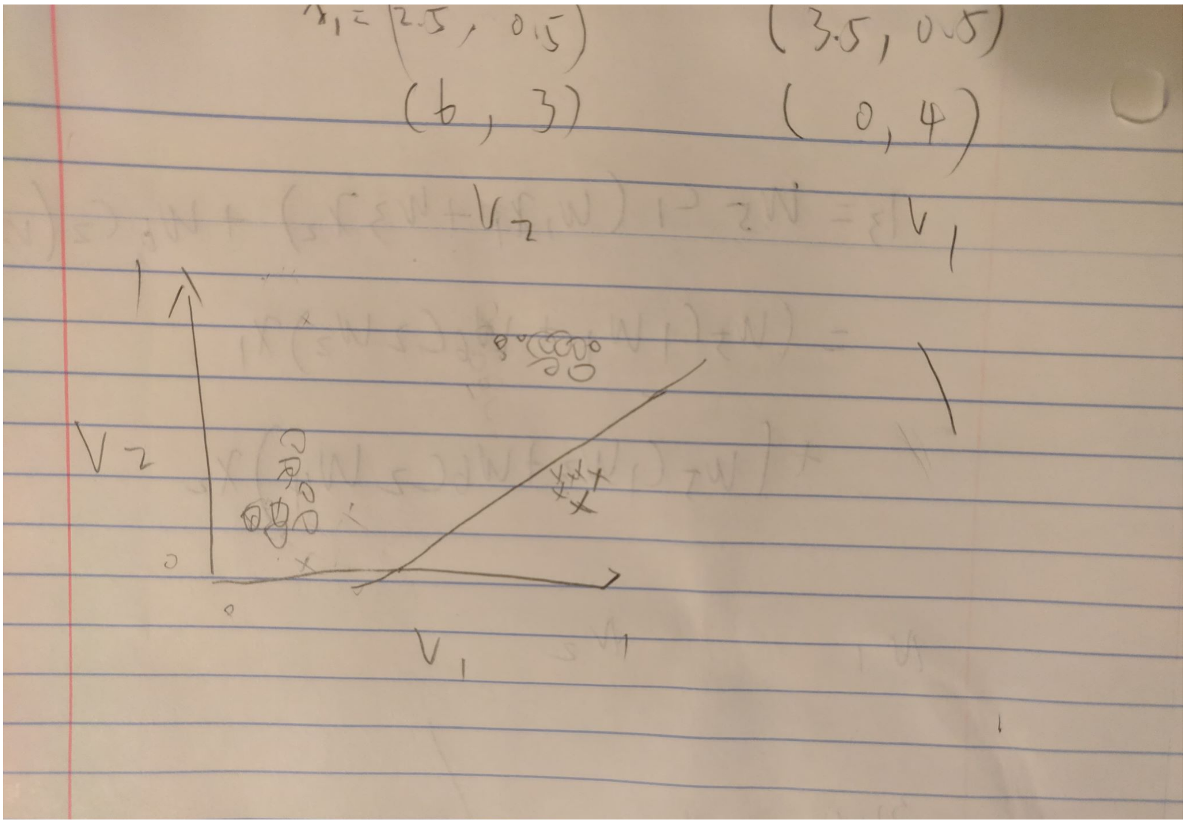
N1 (2 points)



N2 (2 points)



N3 (4 points)



$-x_1 + 3x_2 - 5 = 0$