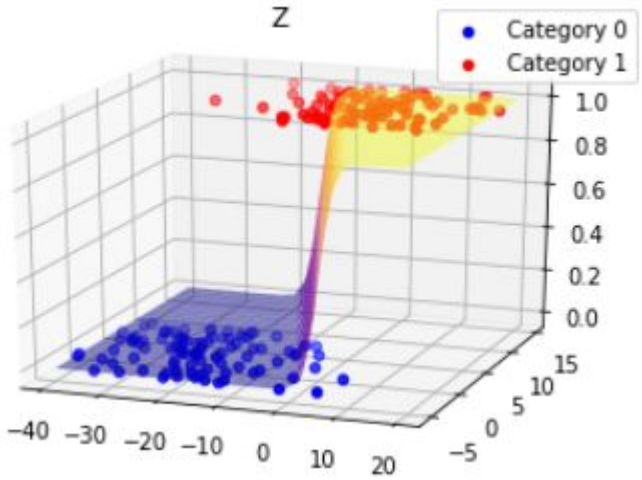
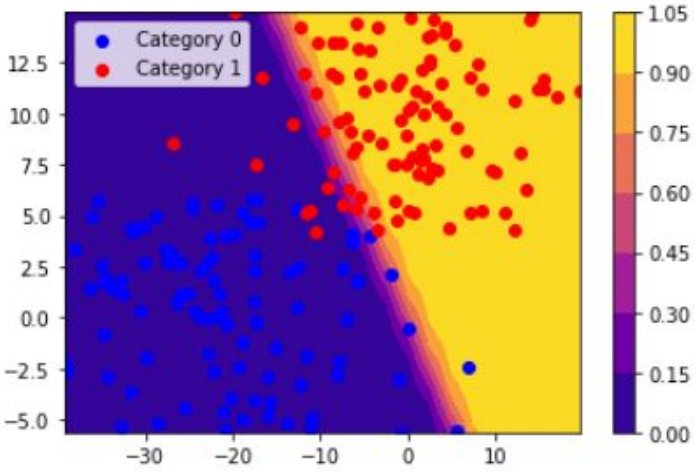
Multilayer Neural Networks

Machine Learning Crash Course

Logistic Regression

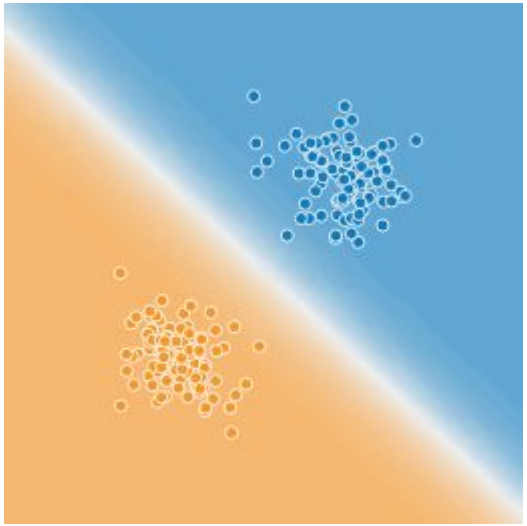
- Output is binary (or 0 to 1)
- We pass linear equation through an activation function (sigmoid / logistic function)





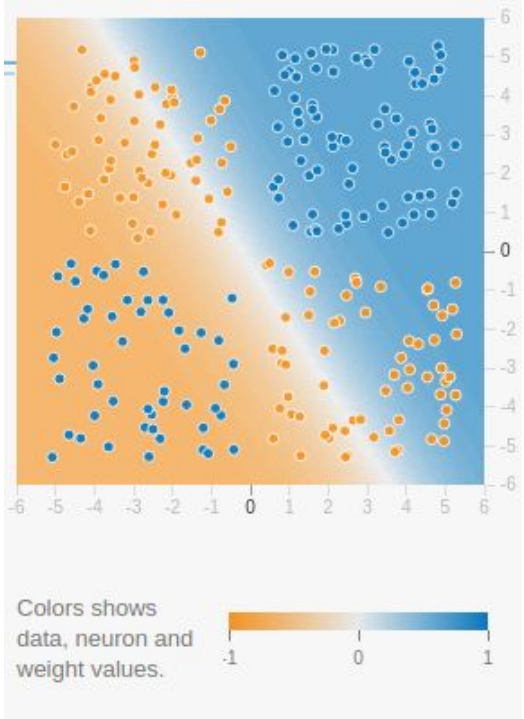
What is the problem with logistic regression?

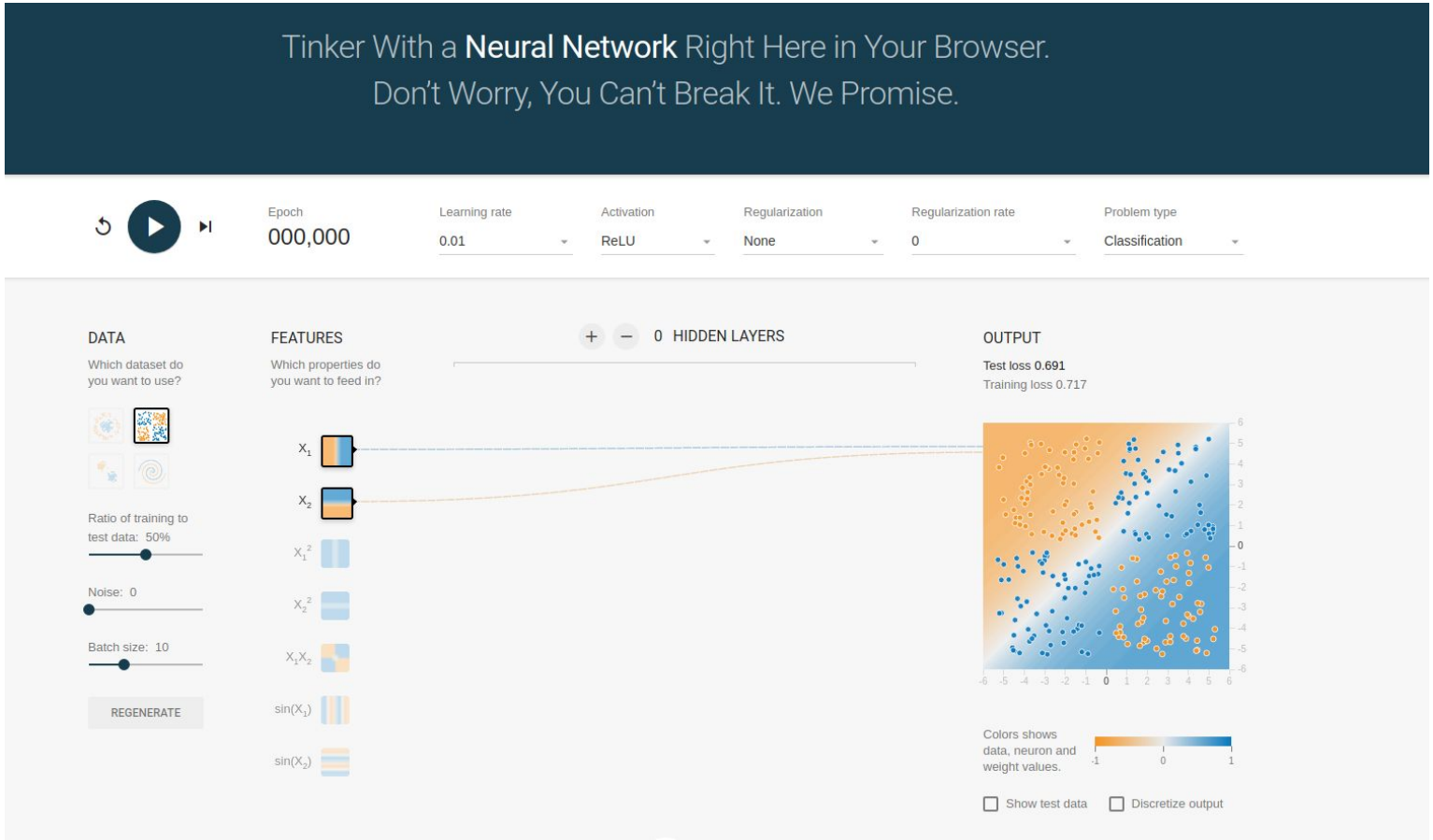
- The decision boundary for logistic regression needs to be a straight line.
- So, data needs to be **linearly separable**



What can we do if our dataset is not linearly separable?

- It is not possible to use plain old linear regression to split a complex dataset.
- Some datasets cannot be split using a straight line
- Example) XNOR function
- f(x, y)
  - +1 if x > 0, y > 0 or x < 0, y < 0
  - -1 if x > 0, y < 0 or x < 0, y > 0

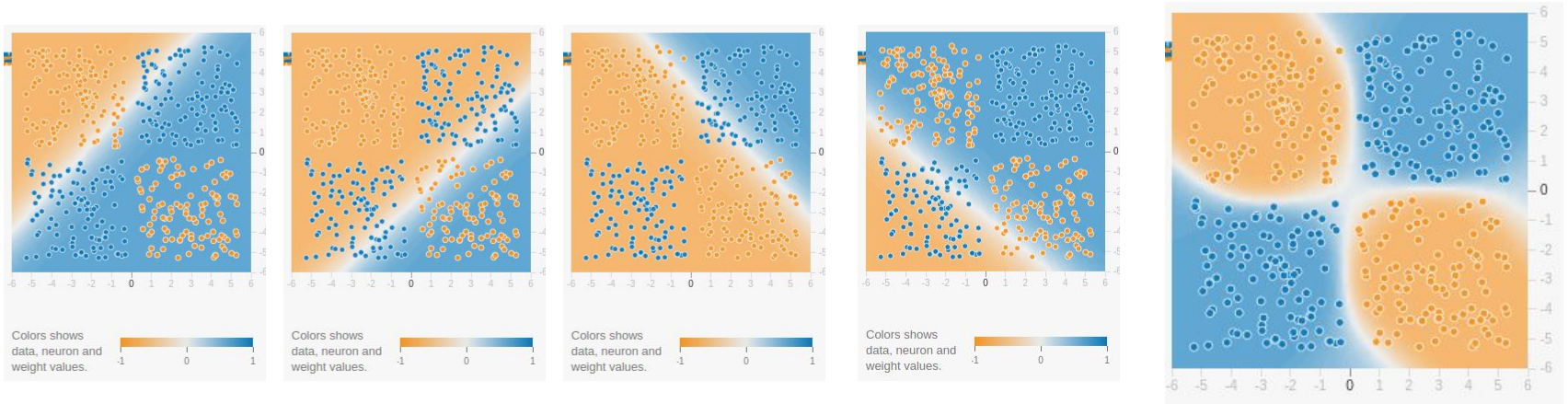


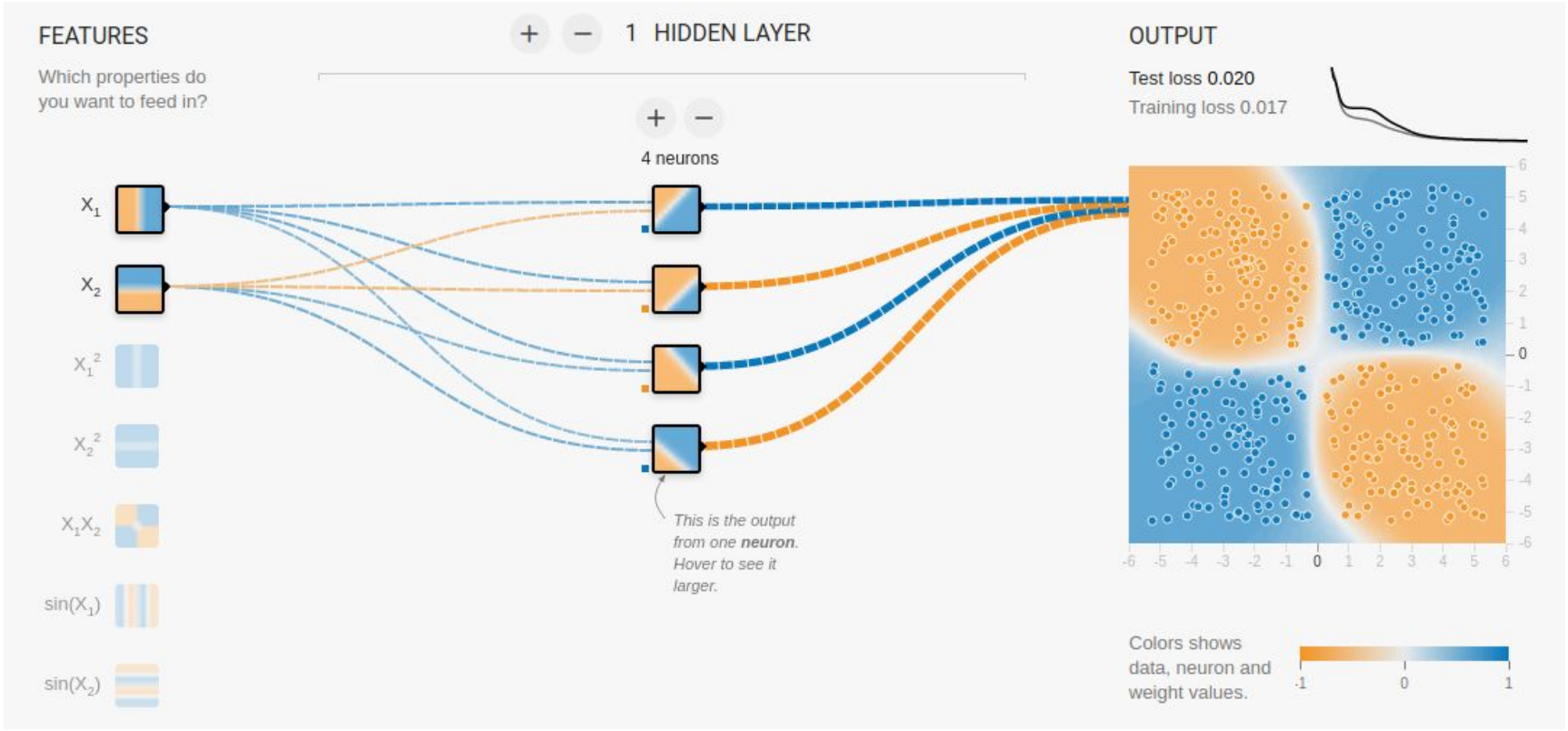


Tensorflow playground

What can we do?!

- We can detect multiple areas using independent logistic regression units.
- Then, we can combine the information from these using another logistic regression unit

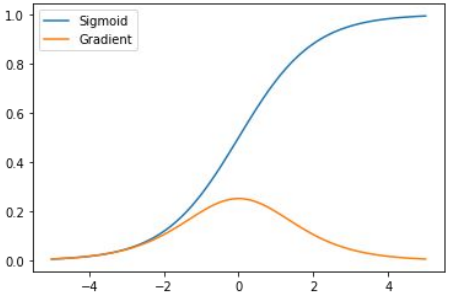



FEATURES
Which properties do you want to feed in?
1 HIDDEN LAYER
4 neurons
X1
X2
X1²
X2²
X1X2
sin(X1)
sin(X2)
This is the output from one neuron. Hover to see it larger.
OUTPUT
Test loss 0.020
Training loss 0.017
Colors shows data, neuron and weight values.
-1
0
1

Activation functions

- In logistic regression, the logistic / sigmoid function was used to convert a plane into a non-linear surface
- Some other functions are also available for different situations
  - Sigmoid
  - Tanh
  - ReLU
  - Leaky ReLU

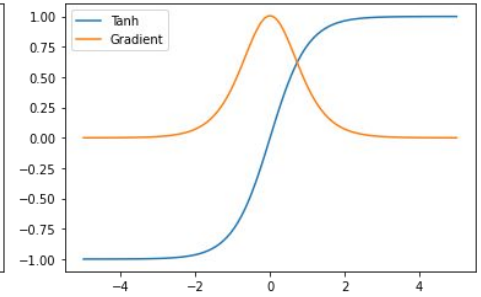
Activation functions



Sigmoid

Output in 0 to 1 range

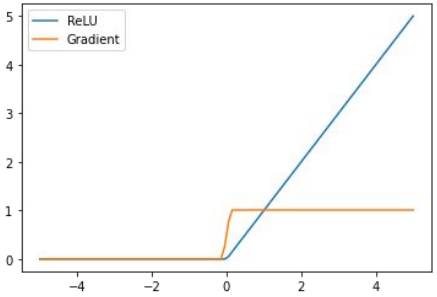
$$sigmoid(x) = \frac{1}{1+e^{-x}}$$



Tanh

Output in -1 to 1 range

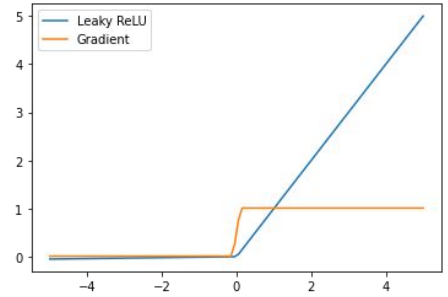
$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$



ReLU

Like a perfect diode :
Rectified Linear Unit

$$ReLU(x) = \begin{cases} x, x > 0 \\ 0, x \leq 0 \end{cases}$$



Leaky ReLU

Like an imperfect diode

Has some slope when inp < 0. Avoid neurons dying because of high -ve input.

Categorical Data

One-Hot encoding

- Sometimes, ML algorithms need to work on categorical data
- Animal : Cat / Dog / Mice
- Language : Hindi / English / Malayalam
- Information indicating presence of only one among a list
- **Problem** : How to encode this?

One-Hot encoding

- As integers?
  - Cat = 0, Dog = 1, Mice = 2
- The problem : This imposes some sort of ordering on data

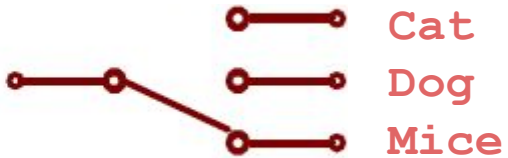
Cat < Dog < Mice

- But, there is no ordering for most categorical data.
- **Solution** : We use a vector with only one element 1 and the rest 0

[1 0 0]	[0 1 0]	[0 0 1]
Cat	Dog	Mice

One-Hot encoding

It is easy if we imagine it like a switch



$$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

What if we need categorical output from neural network

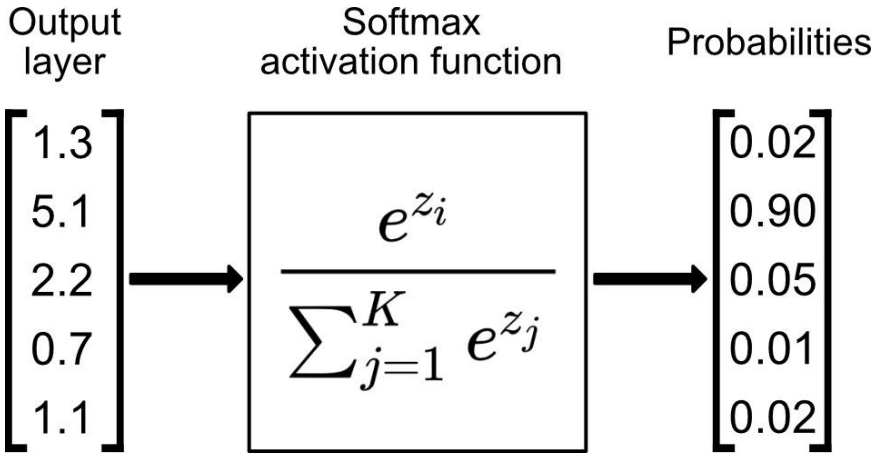
- We use sigmoidal function to get a binary output (in the range 0 to 1)
- For getting categorical, one-hot encoded data as output, we use **softmax function**

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

- Take a vector, apply exponential function on all of them
- Normalize it (such that sum of the vector elements = 1)

How to interpret softmax output?

- Exponential function : Ensures that output is always > 0
- Normalization : Ensures that sum of outputs = 1
- The output vector can be interpreted as probability or confidence



[0.9, 0.1, 0]
"Im 90% sure its a cat"

[0.5, 0, 0.5]
"It could be a cat or a mice"

[0, 1, 0]
"Im 100% sure its a dog"

Iris dataset

Iris dataset

- Inputs
  - Sepal length
  - Sepal width
  - Petal length
  - Petal width
- Output
  - Iris versicolor
  - Iris setosa
  - Iris virginica