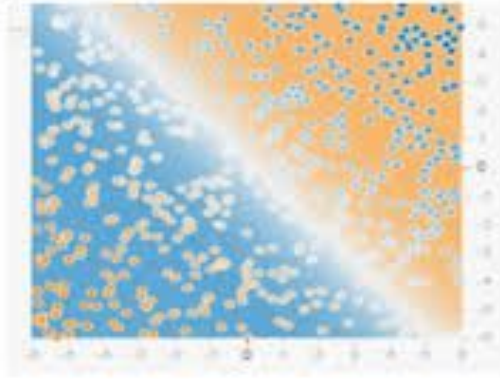


TensorFlow

Playground



www.educba.com

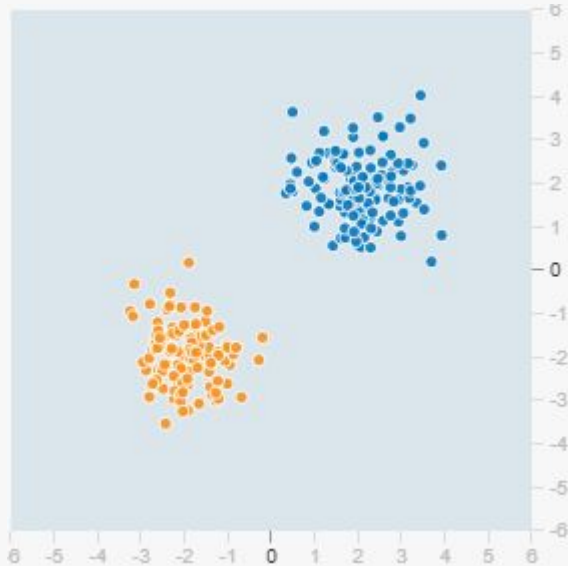


Tests de paramétrage des classifieurs à base de réseaux de neurones sur les différentes distributions de données

Distribution Gaussienne

Distribution Gaussienne

Visualisation de la distribution

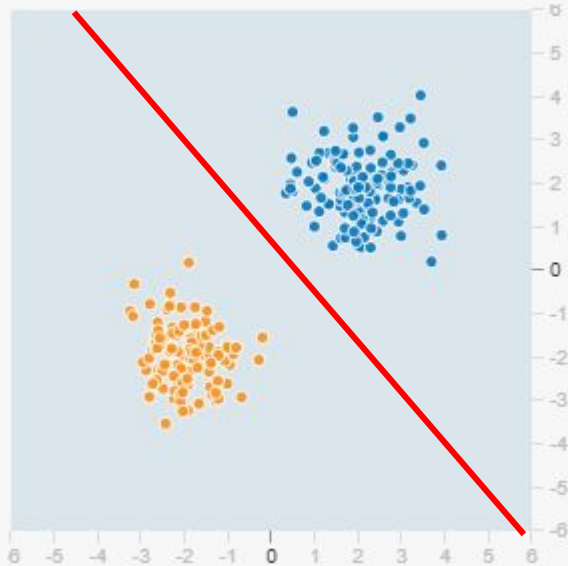


Observation

Les deux classes sont séparées de façon linéaire.

Distribution Gaussienne

Visualisation de la distribution



Observation

Les deux classes sont séparées de façon linéaire.

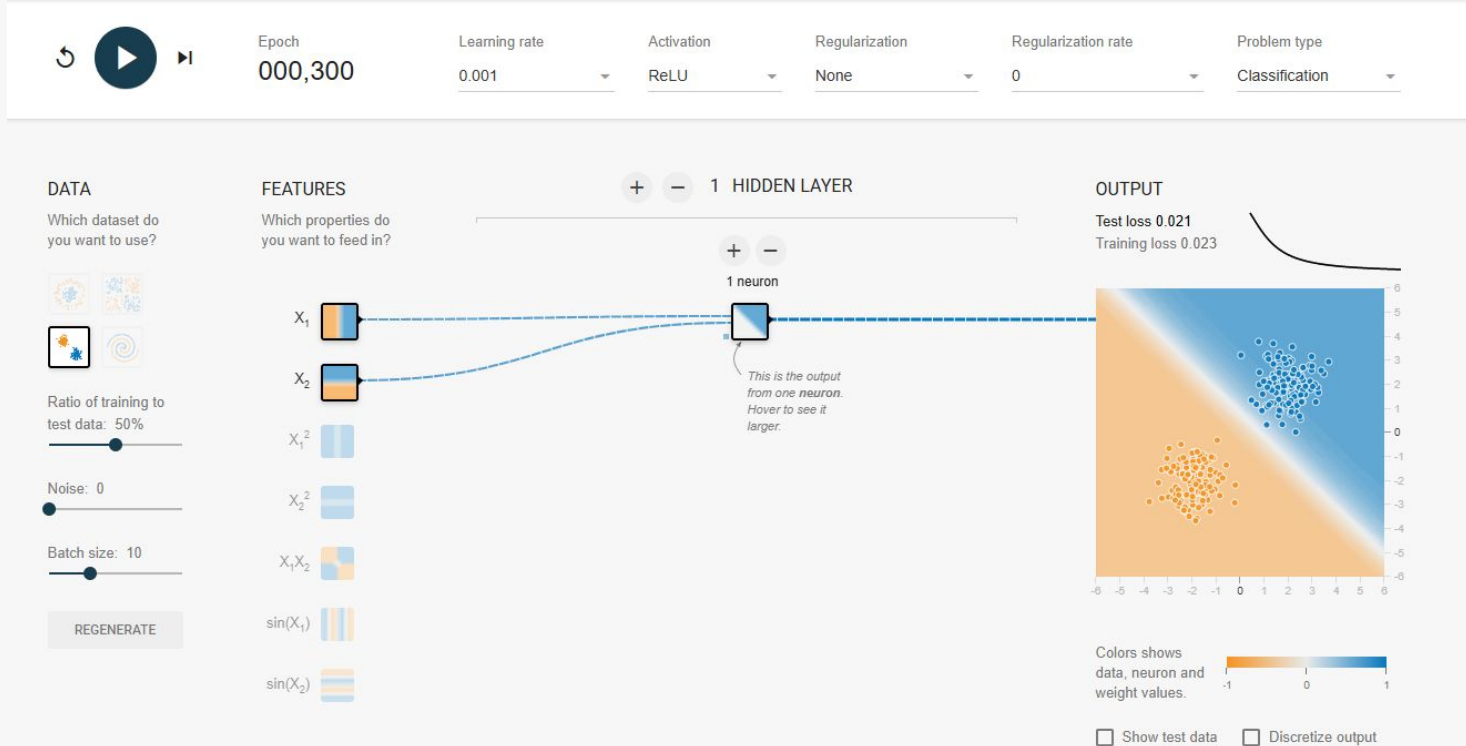
Les variables les plus susceptibles de permettre au réseau de neurones de converger sont donc X_1 , X_2 .

Distribution Gaussienne

Choix des variables

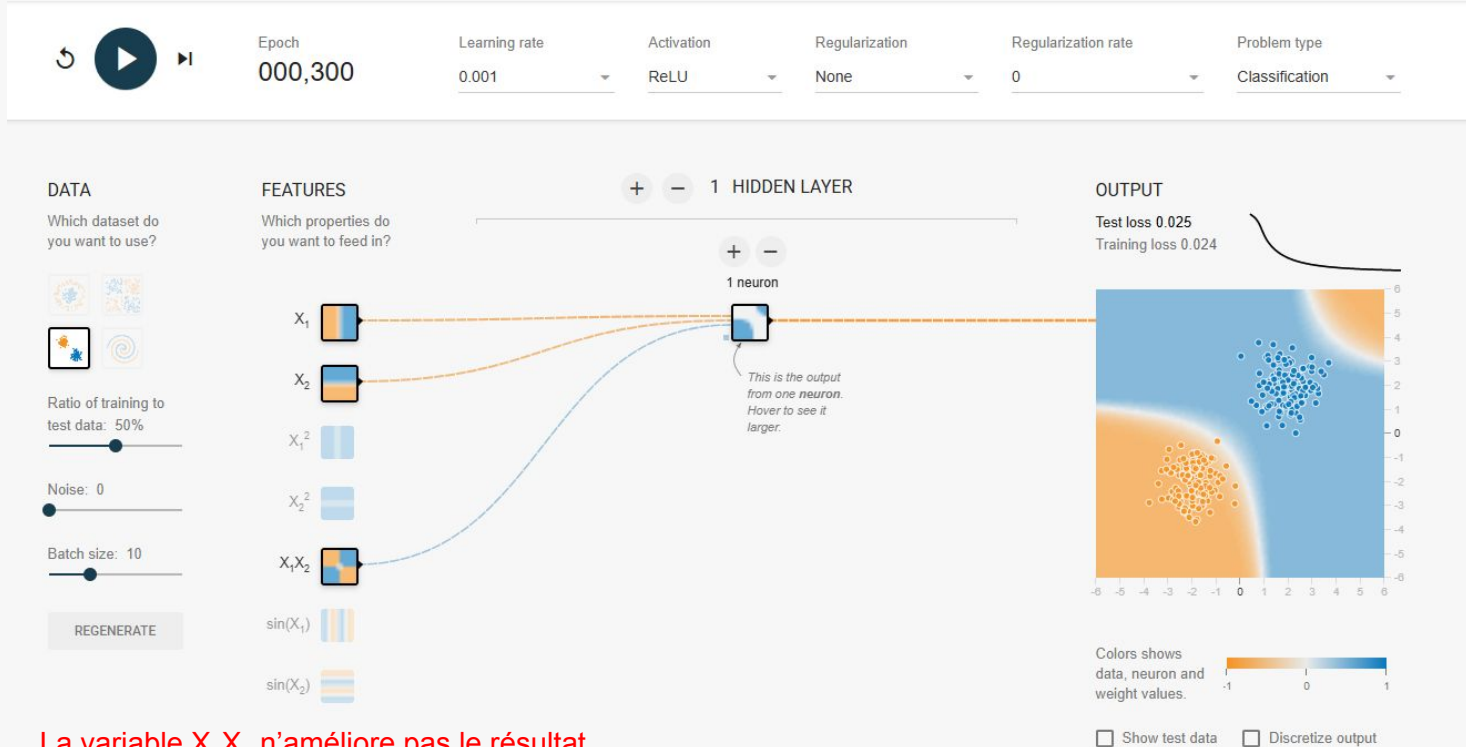
Distribution Gaussienne

Variables : X_1 et X_2



Distribution Gaussienne

Variables : X_1 , X_2 et X_1X_2



La variable X_1X_2 n'améliore pas le résultat

Distribution Gaussienne

Choix de la fonction d'activation

Distribution Gaussienne

Activation: ReLU

↺

▶

Epoch
000,300

Learning rate
0.001

Activation
ReLU

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%



Noise: 0



Batch size: 10



REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

$X_1 X_2$

$\sin(X_1)$

$\sin(X_2)$

+ - 1 HIDDEN LAYER

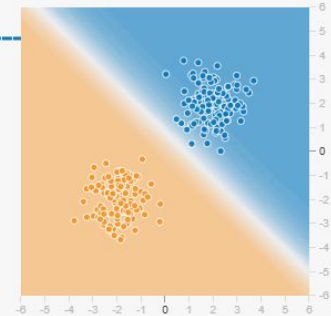
+ -

1 neuron

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

OUTPUT

Test loss 0.021
Training loss 0.023



Colors shows data, neuron and weight values.

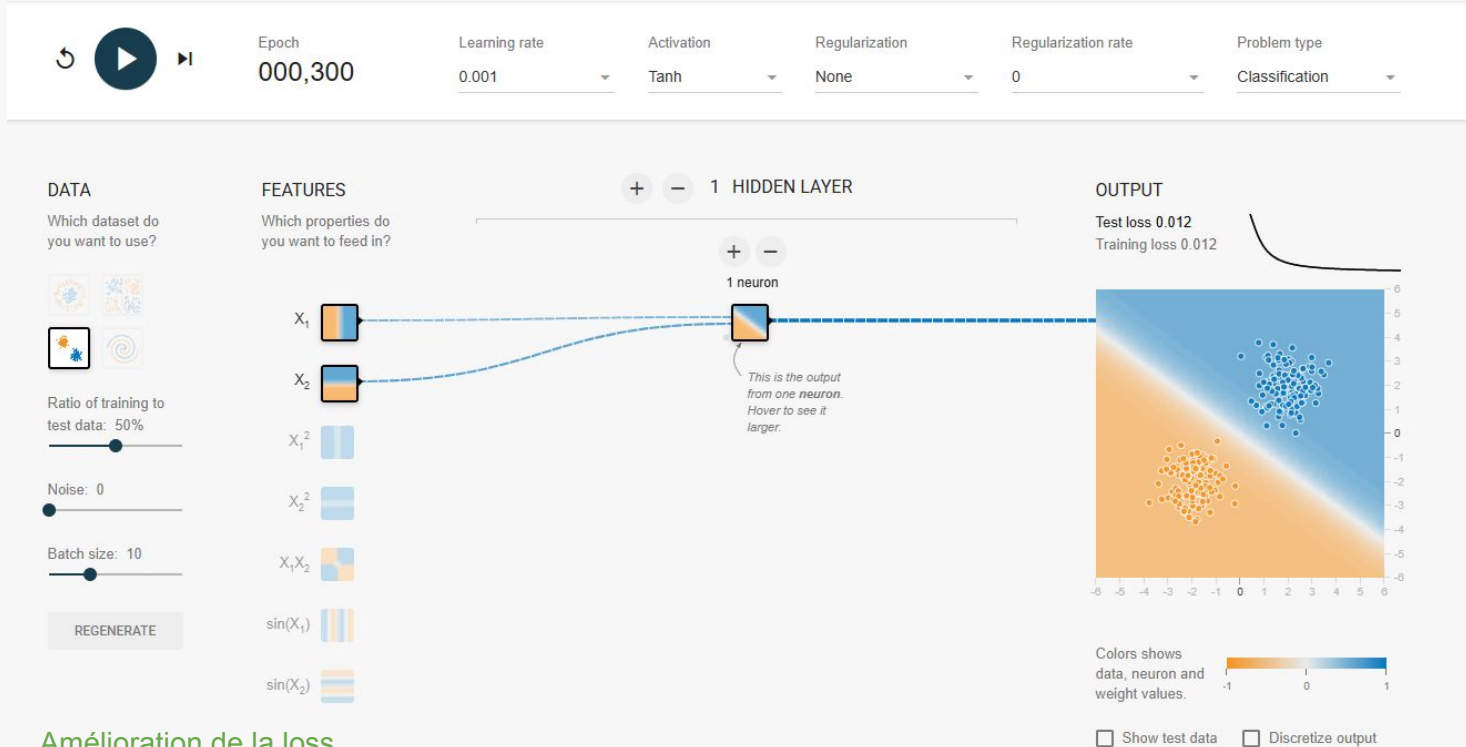


☐ Show test data

☐ Discretize output

Distribution Gaussienne

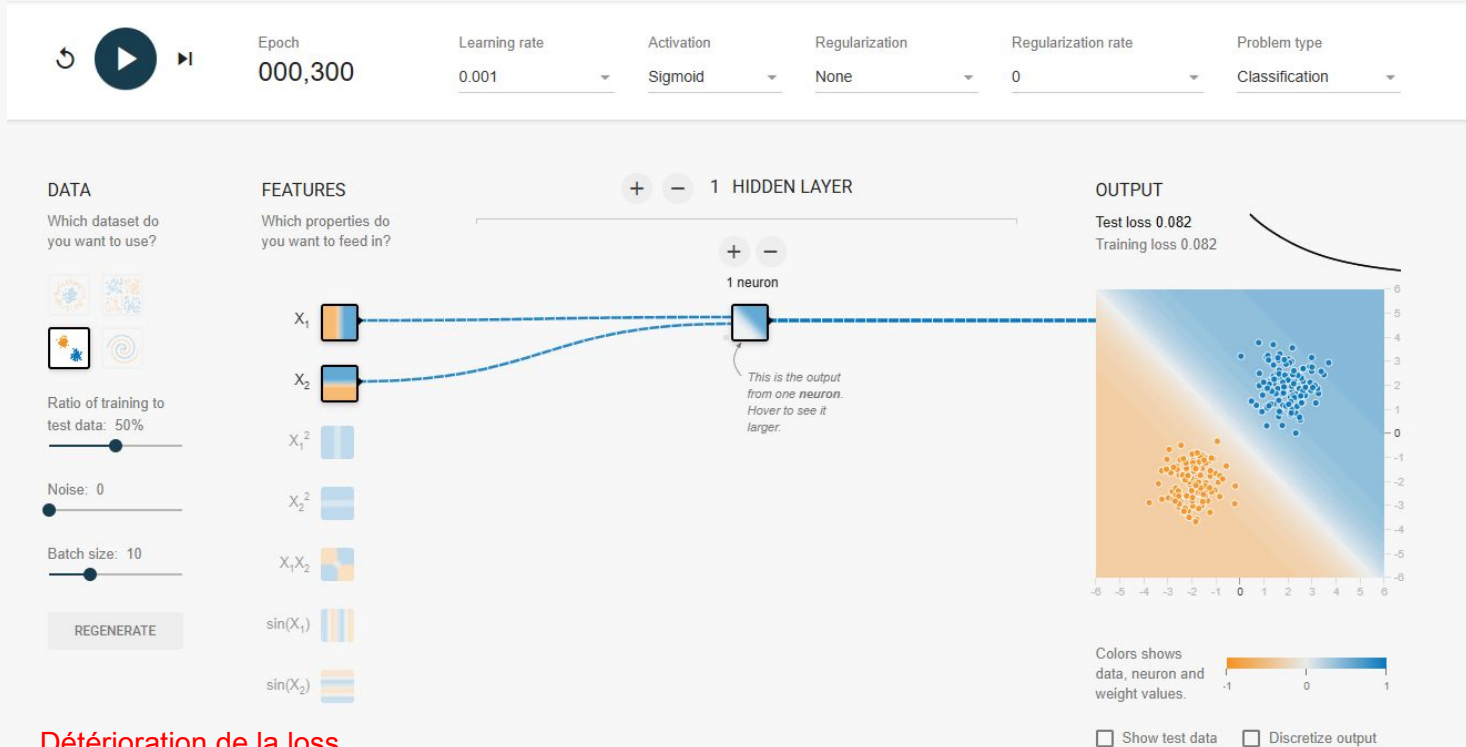
Activation: Tanh



Amélioration de la loss.

Distribution Gaussienne

Activation: Sigmoid



Détérioration de la loss.

Distribution Gaussienne

Activation: Linear

↺

▶

▶

Epoch
000,300

Learning rate
0.001

Activation
Linear

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

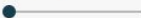
Which dataset do you want to use?



Ratio of training to test data: 50%



Noise: 0



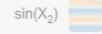
Batch size: 10



REGENERATE

FEATURES

Which properties do you want to feed in?



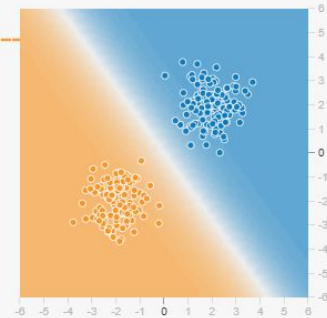
1 HIDDEN LAYER

1 neuron

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

OUTPUT

Test loss 0.002
Training loss 0.003



Colors shows data, neuron and weight values.



☐ Show test data

☐ Discretize output

Meilleure loss.

Distribution Gaussienne

Choix du learning rate

Distribution Gaussienne

Learning Rate : 0.001

↺

▶

▶

Epoch
000,300

Learning rate
0.001

Activation
Linear

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

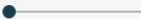
Which dataset do you want to use?



Ratio of training to test data: 50%



Noise: 0



Batch size: 10



REGENERATE

FEATURES

Which properties do you want to feed in?

- X_1
- X_2
- X_1^2
- X_2^2
- $X_1 X_2$
- $\sin(X_1)$
- $\sin(X_2)$

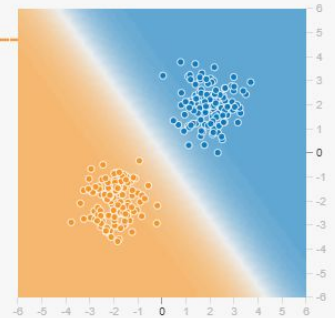
1 HIDDEN LAYER

1 neuron

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

OUTPUT

Test loss 0.002
Training loss 0.003



Colors shows data, neuron and weight values.

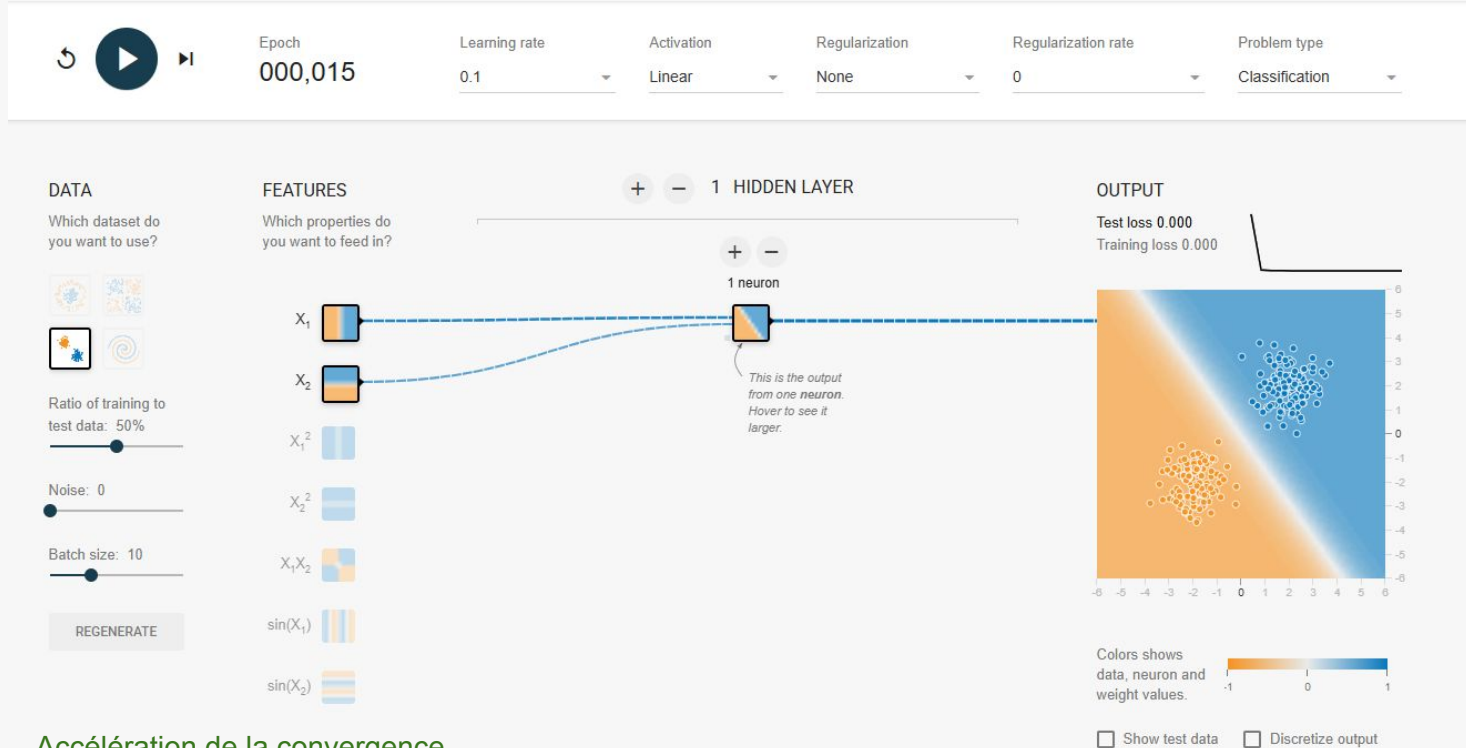


☐ Show test data

☐ Discretize output

Distribution Gaussienne

Learning Rate: 0.1



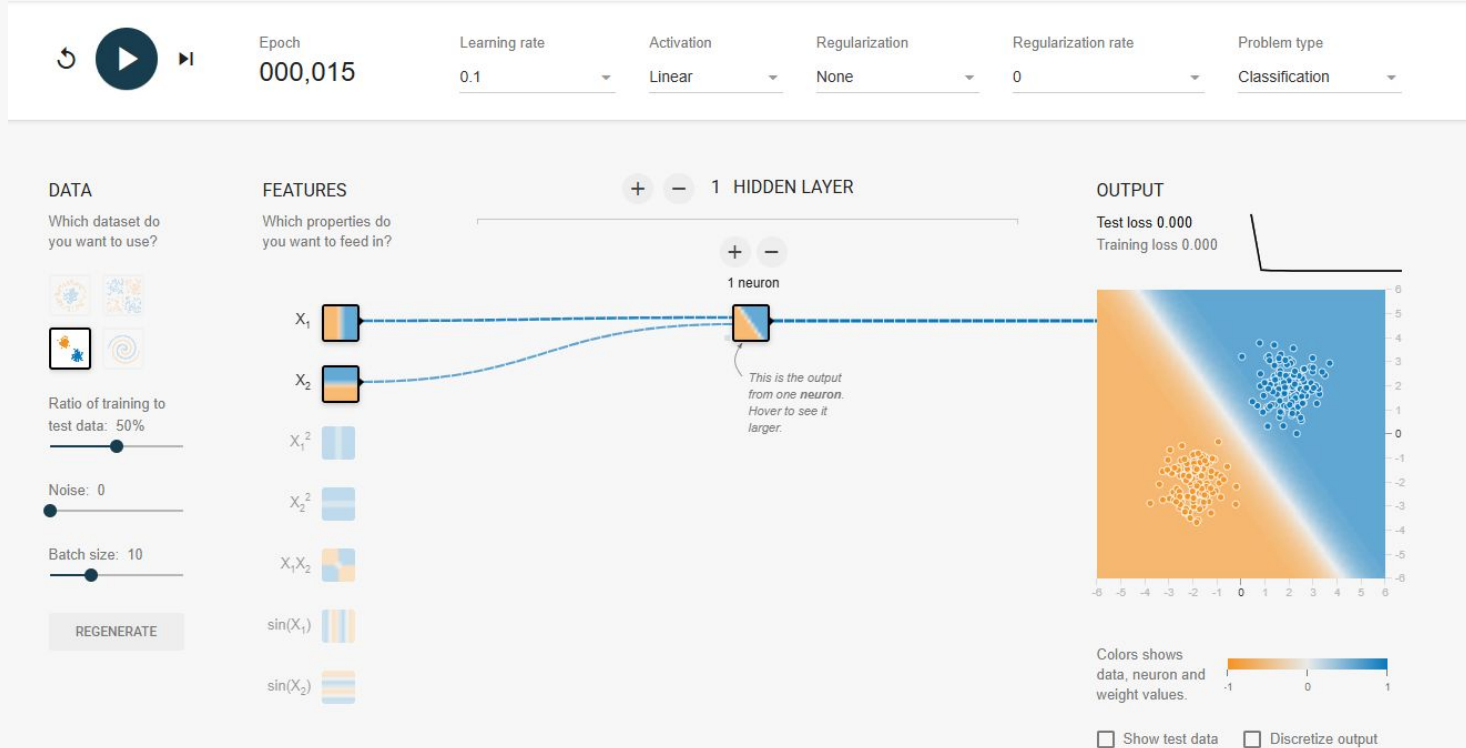
Accélération de la convergence.

Distribution Gaussienne

Choix du nombre de neurones

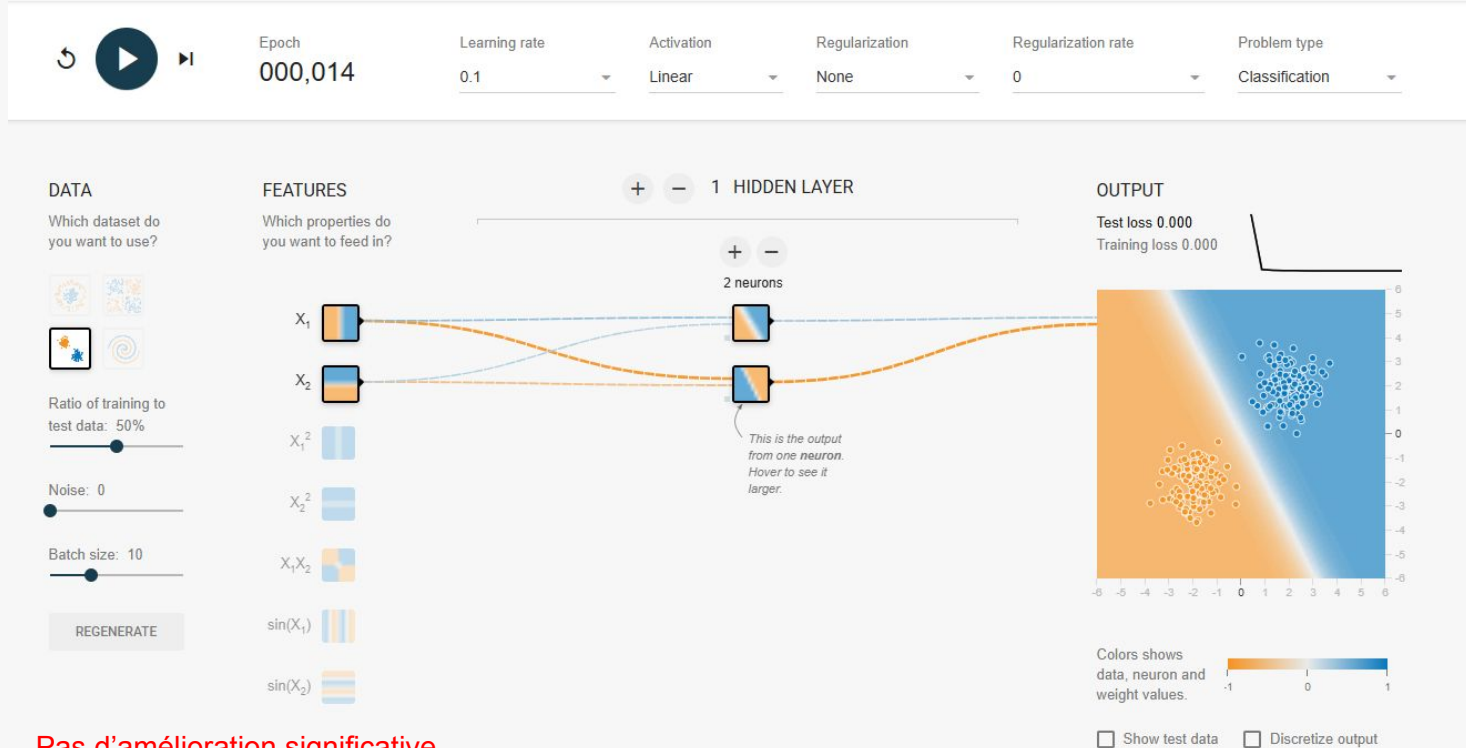
Distribution Gaussienne

Nombre de neurones : 1



Distribution Gaussienne

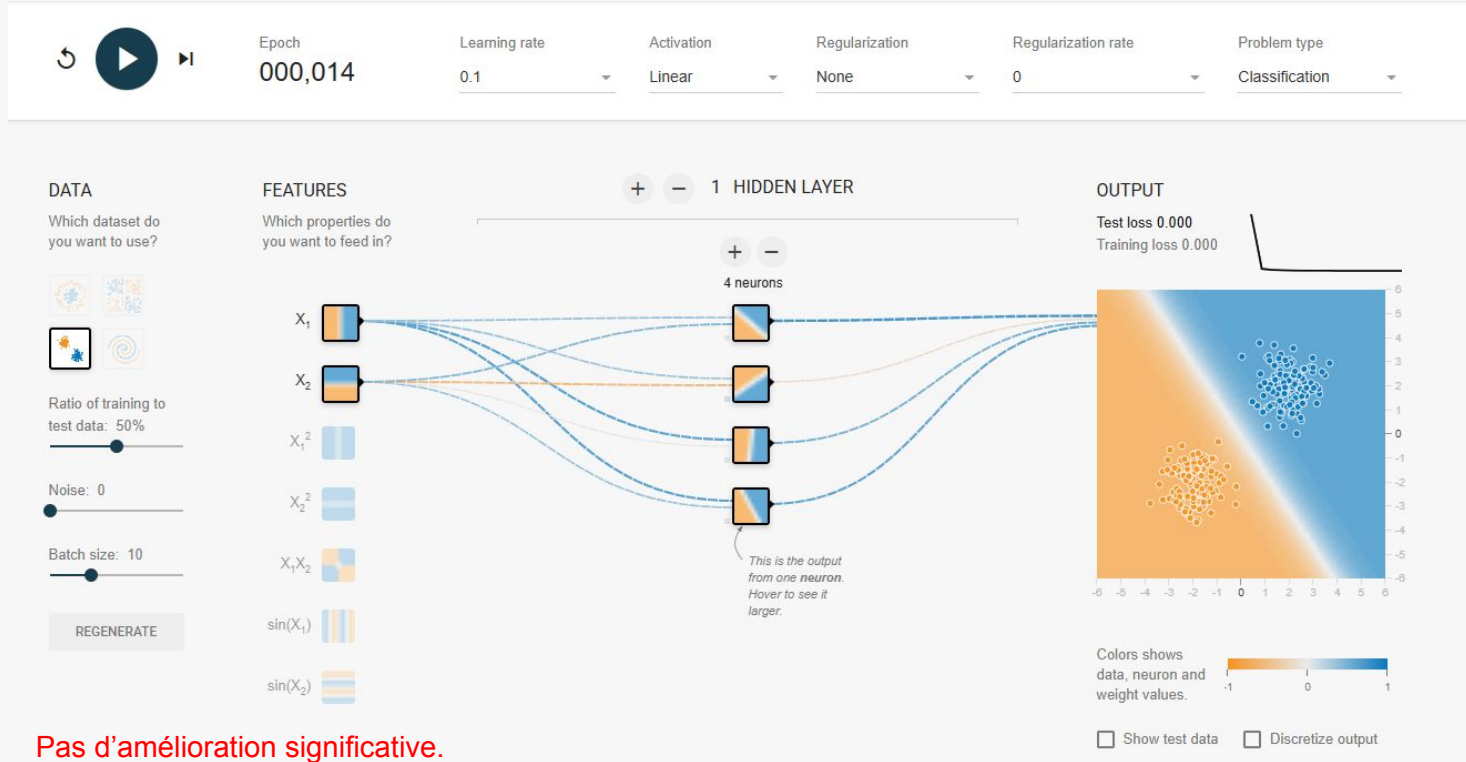
Nombre de neurones : 2



Pas d'amélioration significative.

Distribution Gaussienne

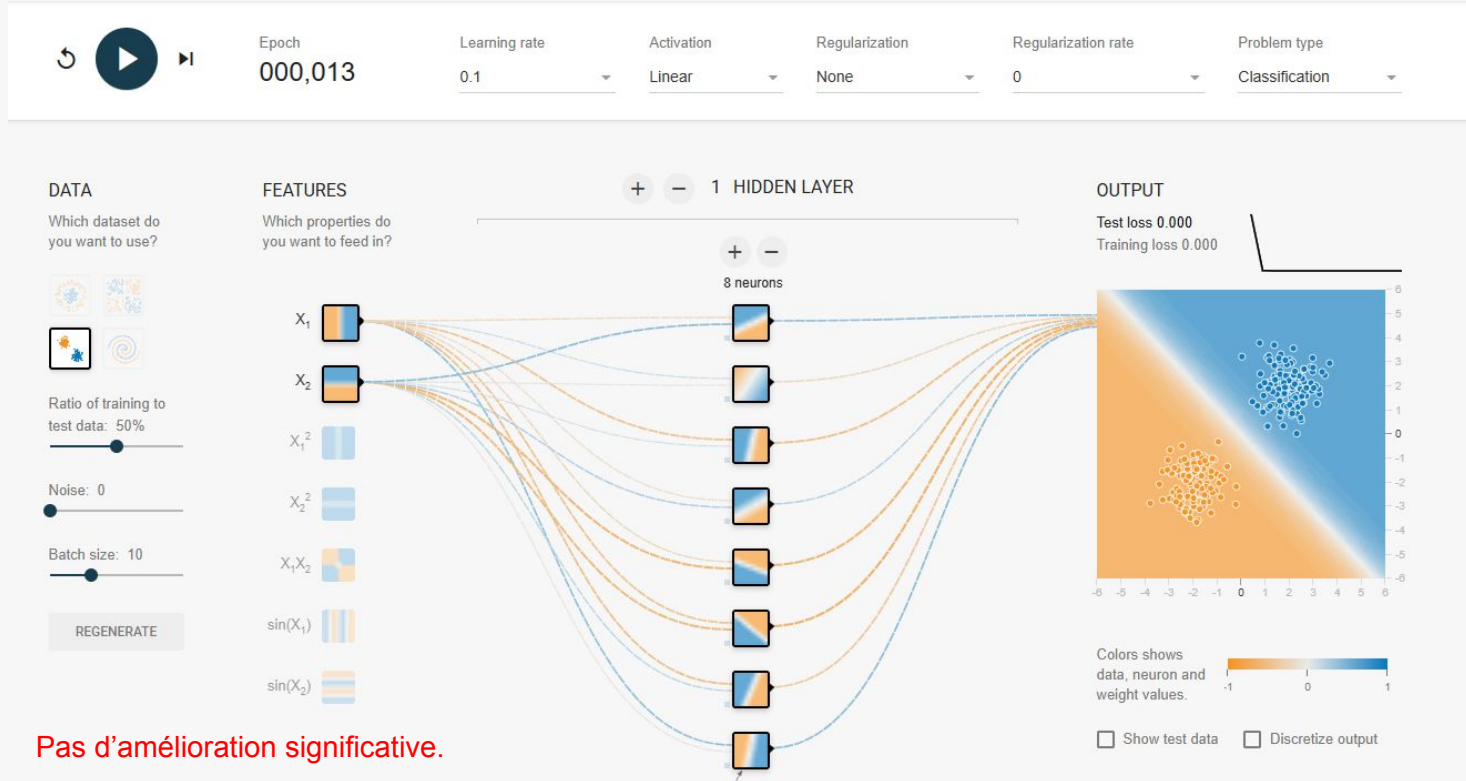
Nombre de neurones : 4



Pas d'amélioration significative.

Distribution Gaussienne

Nombre de neurones : 8



Pas d'amélioration significative.

Distribution Gaussienne

Conclusions

Dans le cadre d'une classification binaire de données linéairement séparables, nous savons que le perceptron est adapté. C'est donc sans trop de surprise que l'on constate qu'un seul neurone suffit pour venir à bout de la classification de cette distribution de points.

Le meilleur choix de variables est X_1 et X_2 .

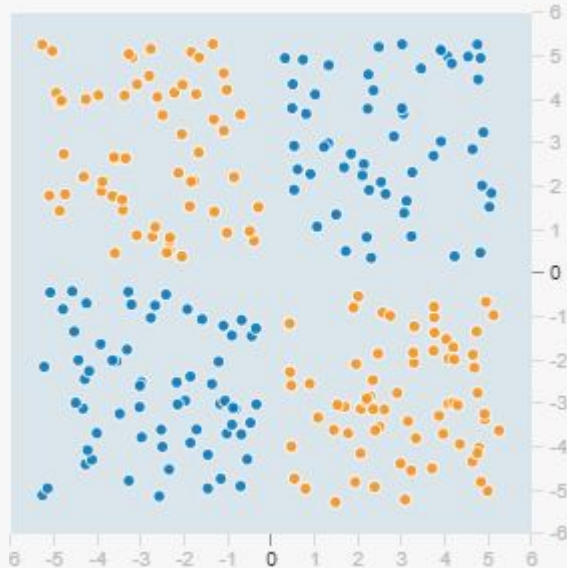
La fonction d'activation la plus adaptée est la fonction Linear.

La diminution du learning rate permet de faire converger le modèle plus rapidement.

Distribution Ou Exclusif

Distribution Ou Exclusif

Visualisation de la distribution

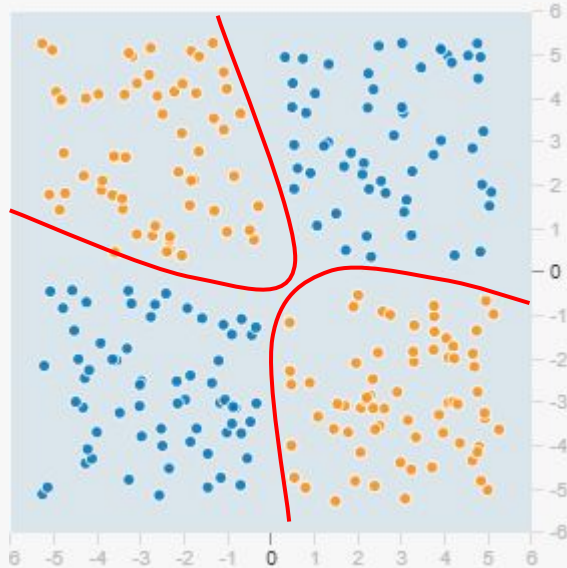


Observation

Chaque classe est répartie dans deux cadrans. La séparation n'est pas linéaire.

Distribution Ou Exclusif

Visualisation de la distribution



Observation

Chaque classe est répartie dans deux cadrans. La séparation est de forme hyperbolique.

Les variables les plus susceptibles de permettre au réseau de neurones de converger sont donc X_1X_2 .

Distribution Ou Exclusif

Choix des variables

Distribution Ou Exclusif

Variables : $X_1 X_2$

Epoch: 001,000 Learning rate: 0.001 Activation: ReLU Regularization: None Regularization rate: 0 Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

- X_1
- X_2
- X_1^2
- X_2^2
- $X_1 X_2$
- $\sin(X_1)$
- $\sin(X_2)$

+ - 1 HIDDEN LAYER

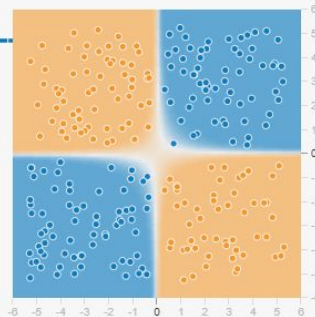
+ -

1 neuron

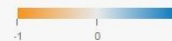
This is the output from one neuron. Hover to see it larger.

OUTPUT

Test loss 0.012
Training loss 0.010



Colors shows data, neuron and weight values.

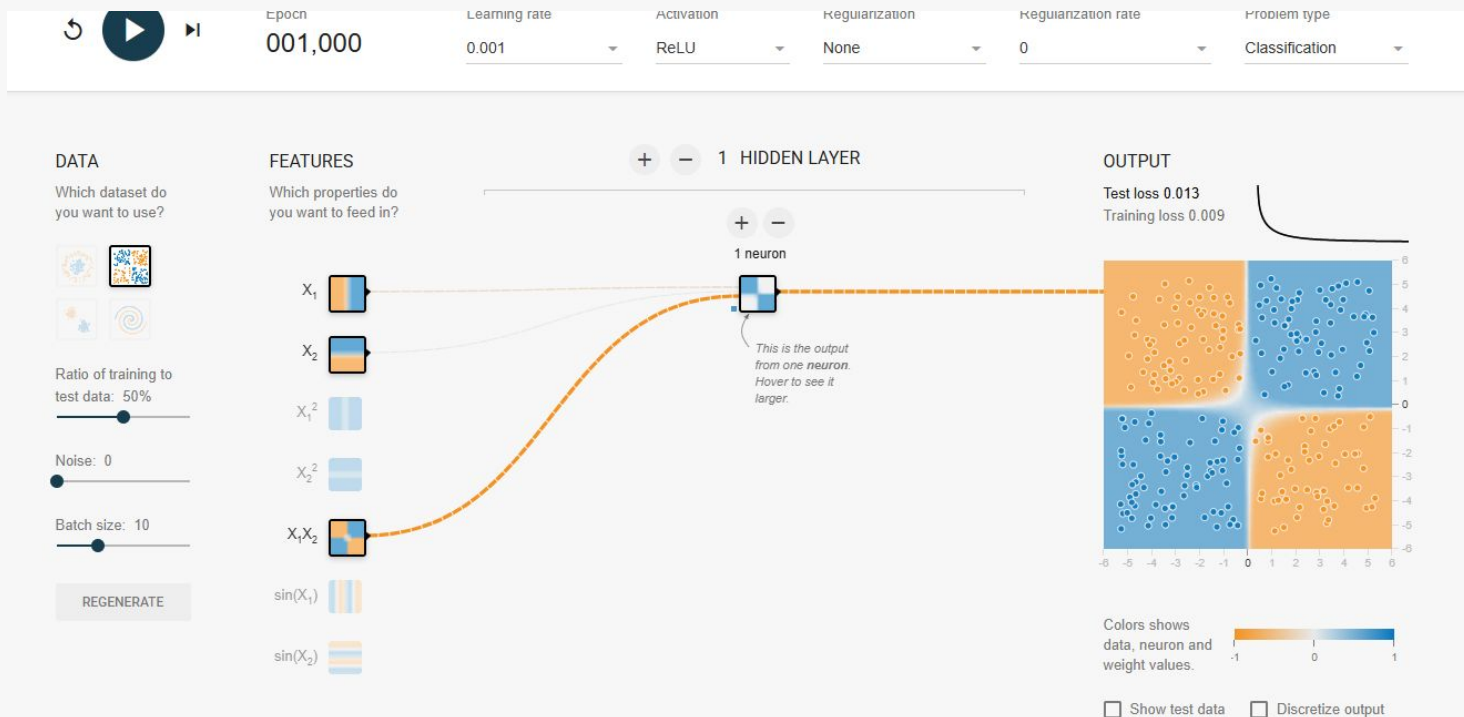


☐ Show test data

☐ Discretize output

Distribution Ou Exclusif

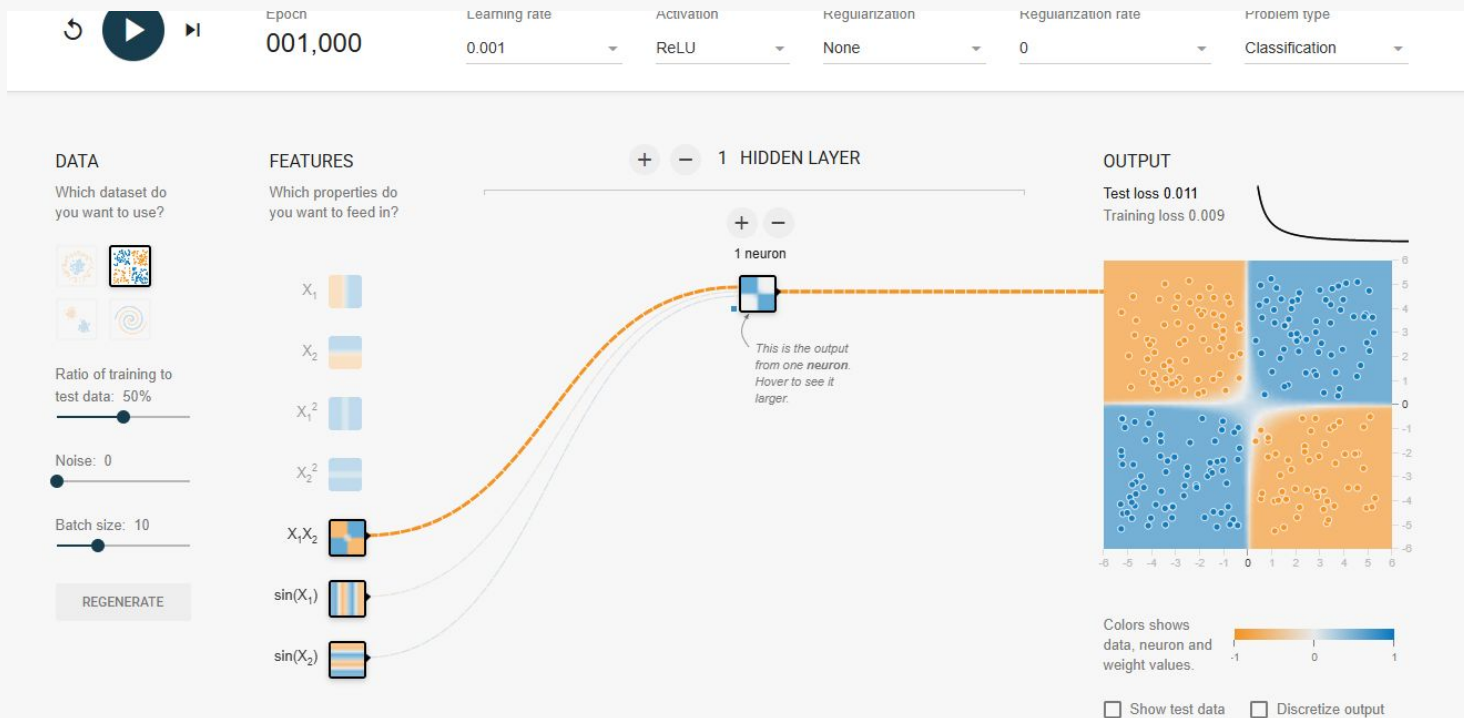
Variables : X_1 , X_2 et X_1X_2



Les nouvelles variables n'ont pas d'influence.

Distribution Ou Exclusif

Variables : $\sin(X_1)$, $\sin(X_2)$ et X_1X_2



Les nouvelles variables n'ont pas d'influence.

Distribution Ou Exclusif

Choix de la fonction d'activation

Distribution Ou Exclusif

Activation: ReLu

Epoch: 001,000 Learning rate: 0.001 Activation: ReLU Regularization: None Regularization rate: 0 Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?



+ - 1 HIDDEN LAYER

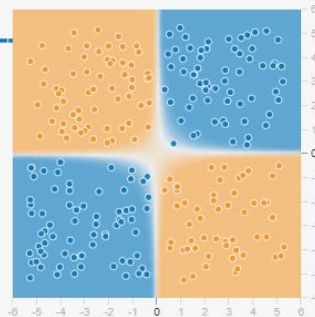
+ -

1 neuron

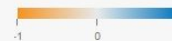
This is the output from one neuron. Hover to see it larger.

OUTPUT

Test loss 0.012
Training loss 0.010



Colors shows data, neuron and weight values.

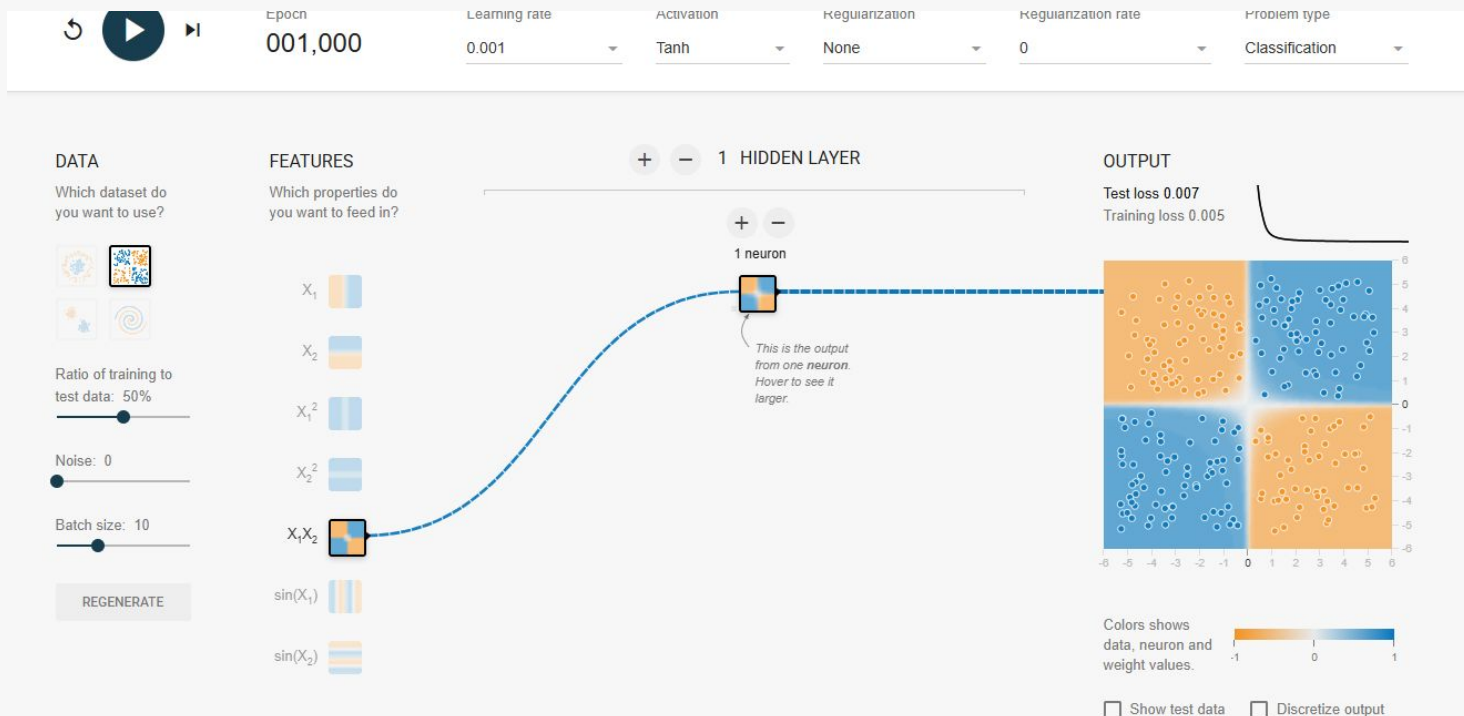


☐ Show test data

☐ Discretize output

Distribution Ou Exclusif

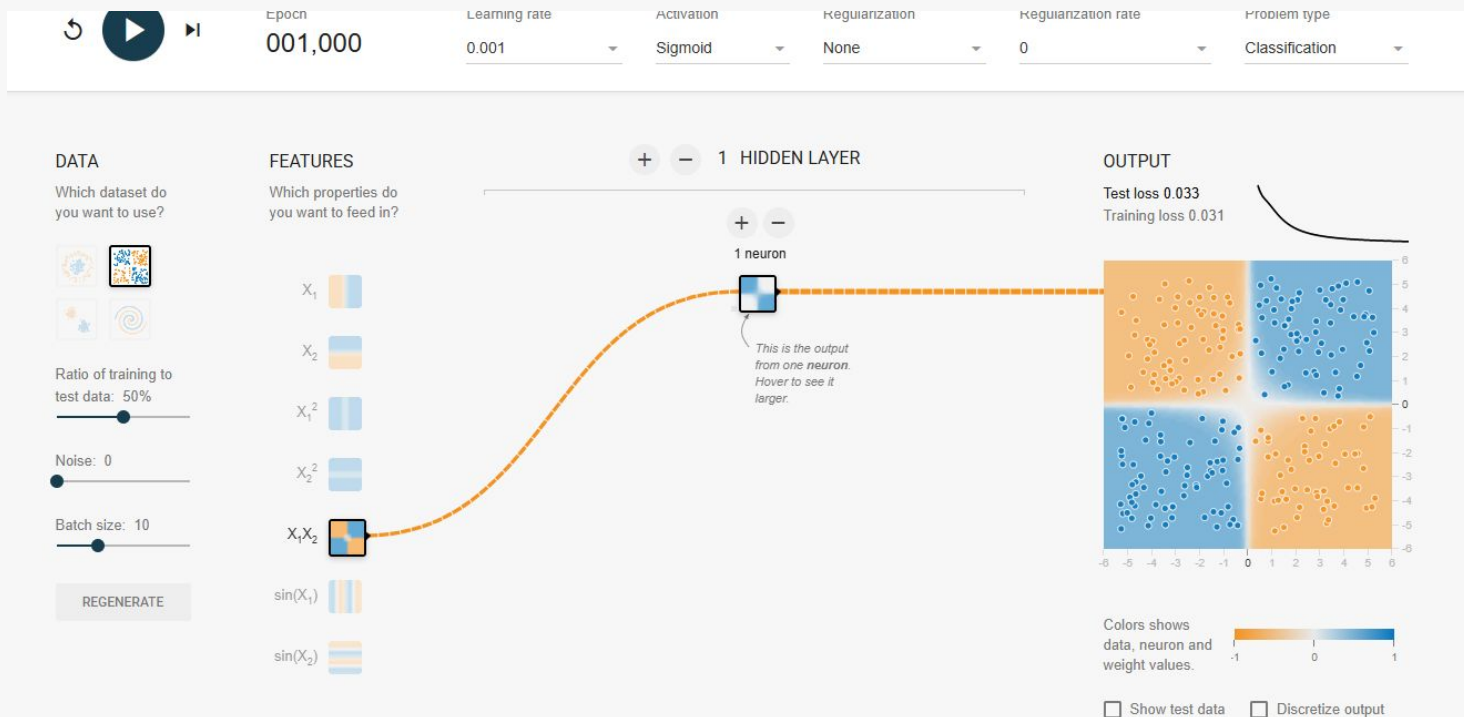
Activation: Tanh



Amélioration de la loss.

Distribution Ou Exclusif

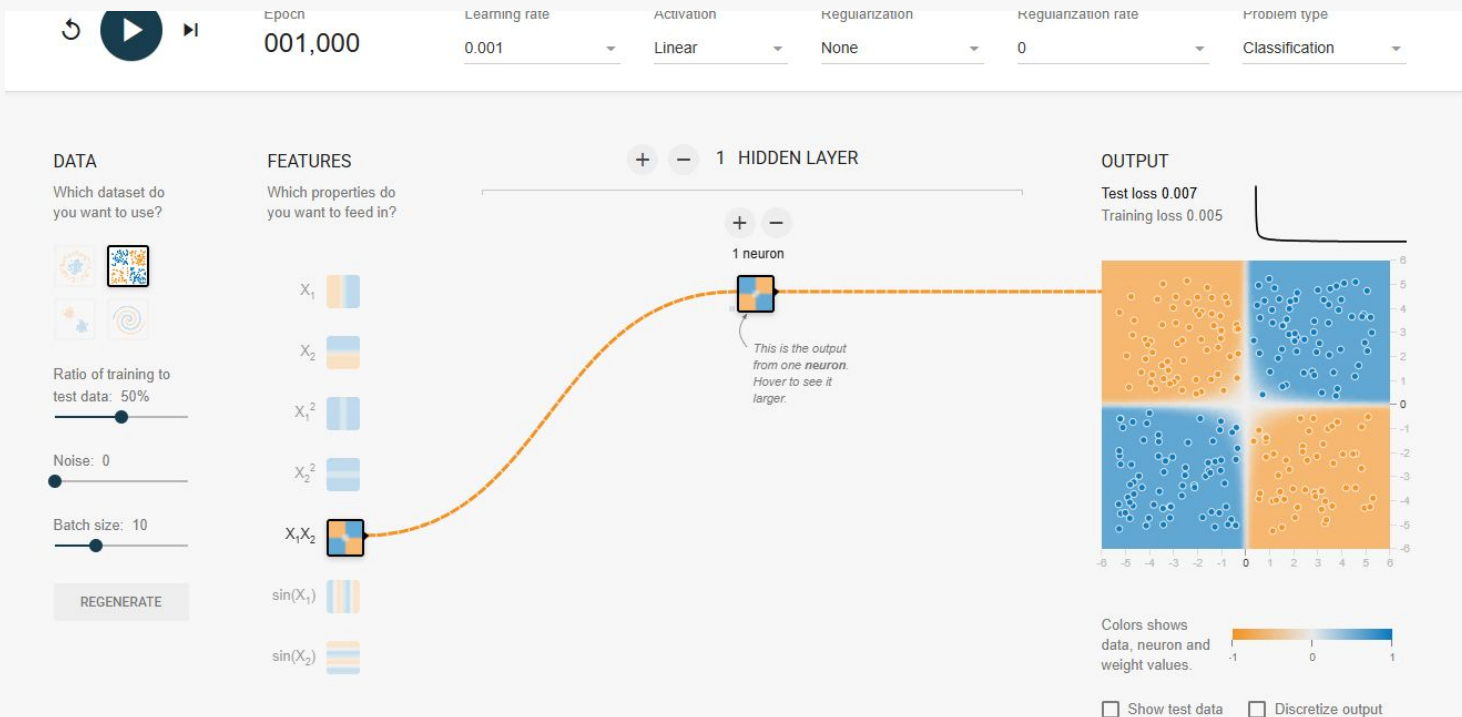
Activation: Sigmoid



Augmentation de la loss.

Distribution Ou Exclusif

Activation: Linear



Même résultat que Tanh.

Distribution Ou Exclusif

Choix du learning rate

Distribution Ou Exclusif

Learning Rate: 0.001

Epoch: 001,000 Learning rate: 0.001 Activation: Tanh Regularization: None Regularization rate: 0 Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

$X_1 X_2$

$\sin(X_1)$

$\sin(X_2)$

+ - 1 HIDDEN LAYER

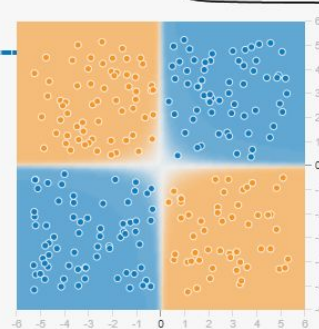
+ -

1 neuron

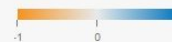
This is the output from one neuron. Hover to see it larger.

OUTPUT

Test loss 0.007
Training loss 0.005



Colors shows data, neuron and weight values.



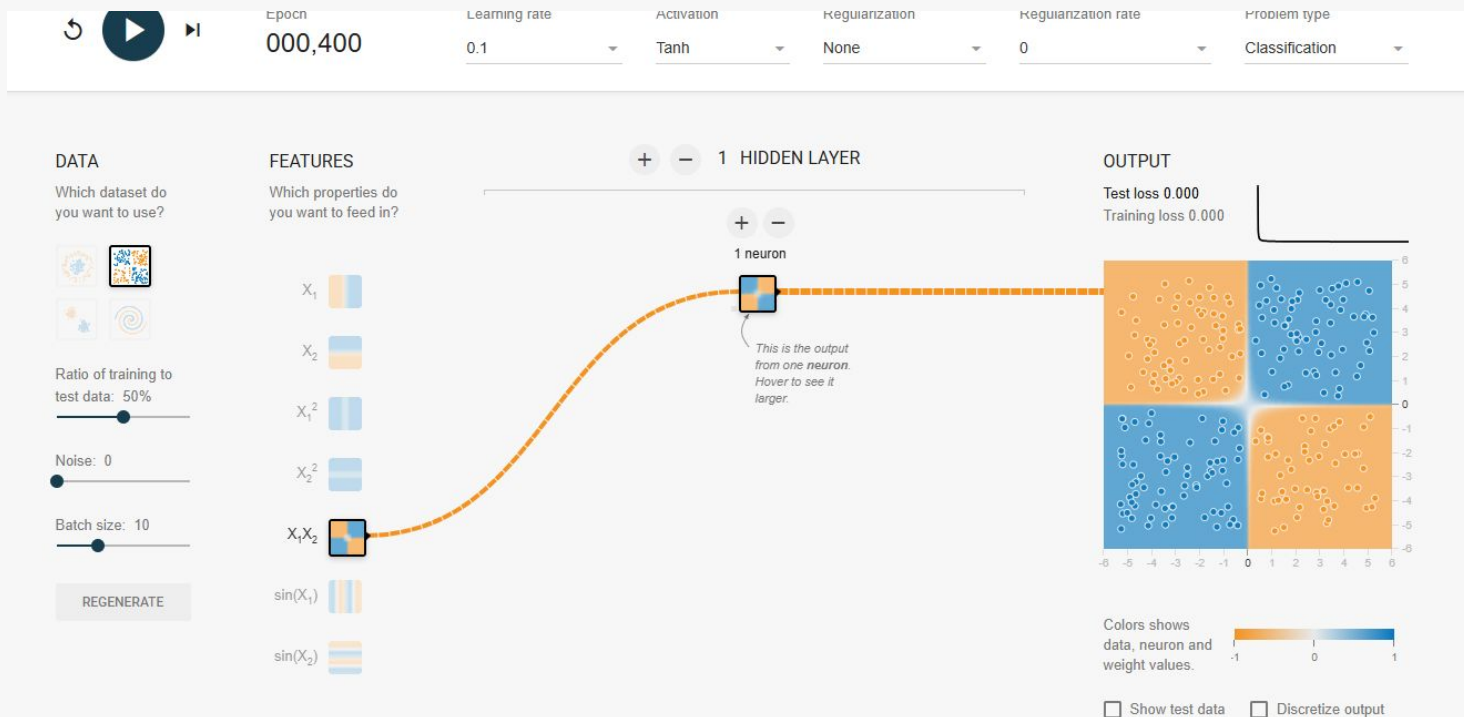
☐ Show test data

☐ Discretize output

Amélioration de la loss.

Distribution Ou Exclusif

Learning Rate: 0.1



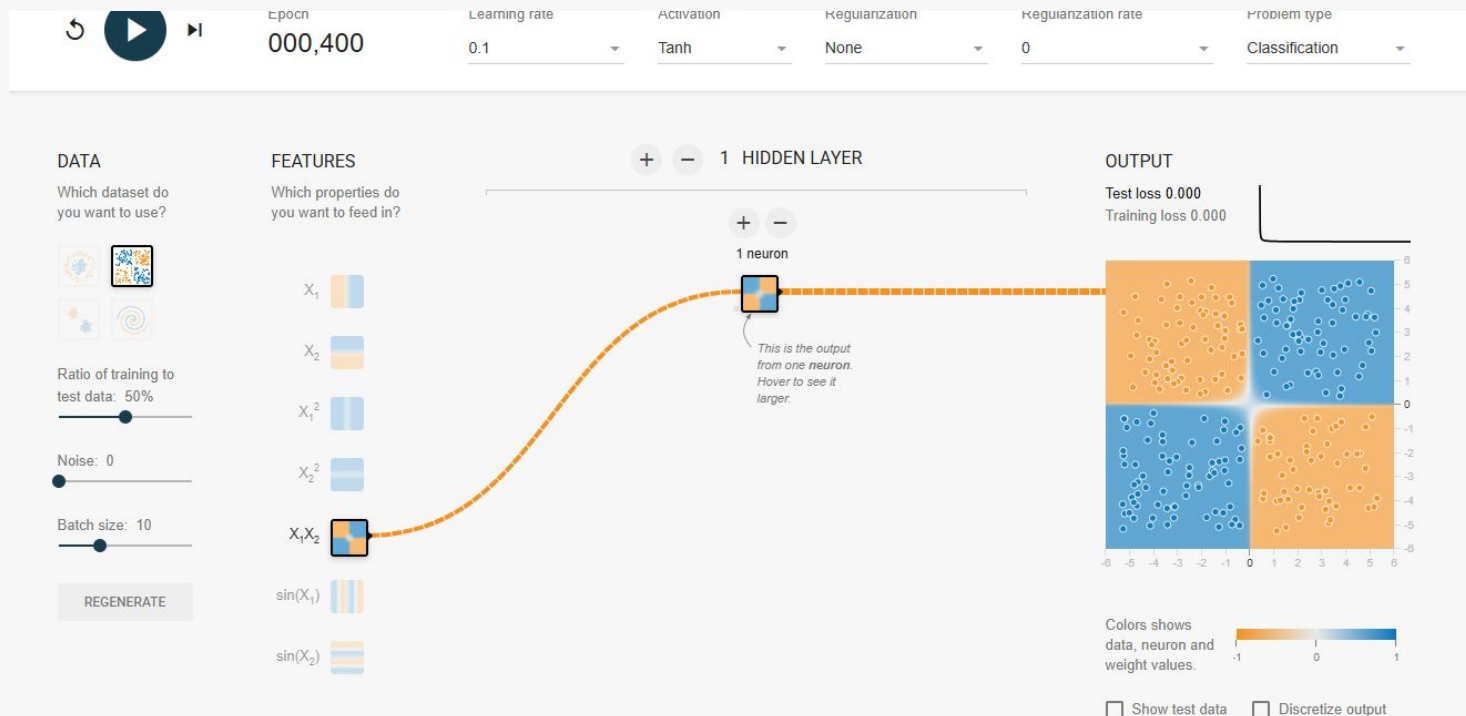
Accélération de la convergence et amélioration de la loss.

Distribution Ou Exclusif

Choix du nombre de neurones

Distribution Ou Exclusif

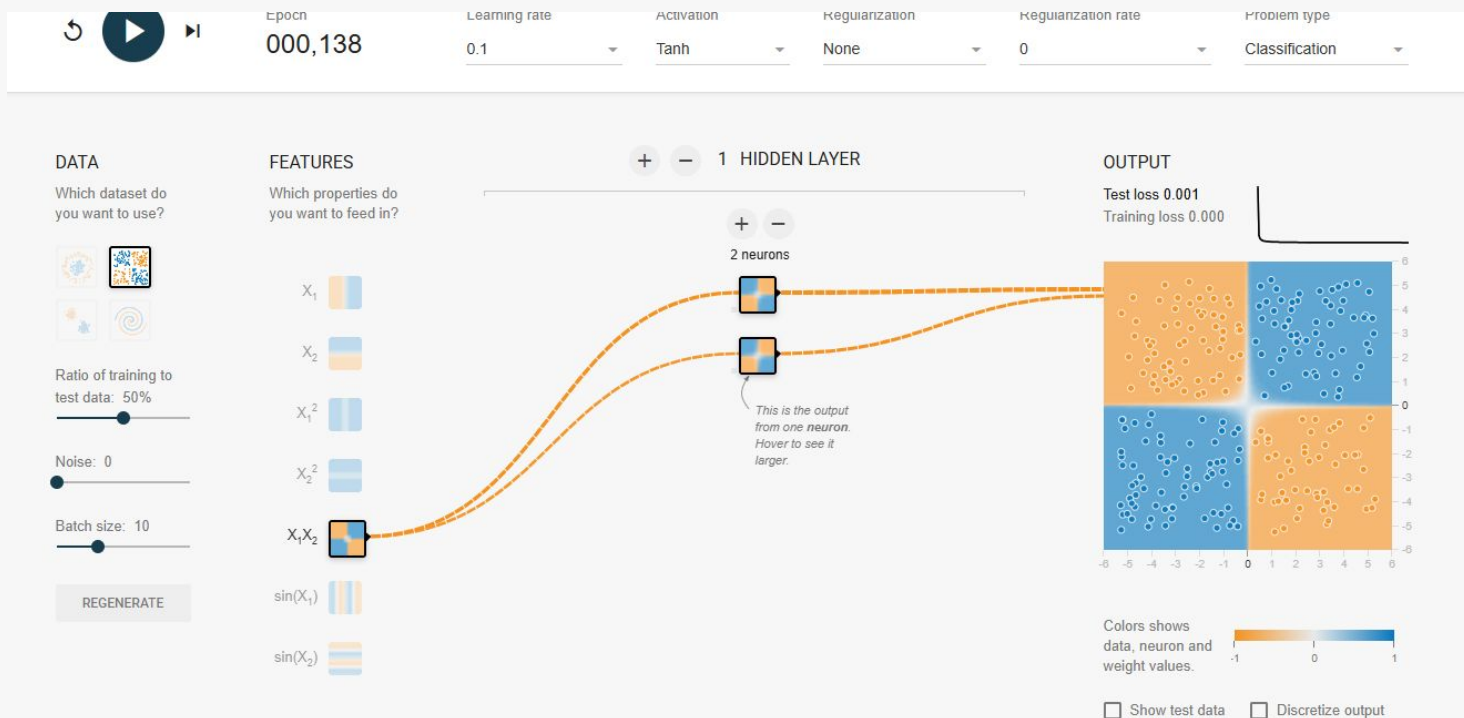
Nombre de neurones: 1



Accélération de la convergence et amélioration de la loss.

Distribution Ou Exclusif

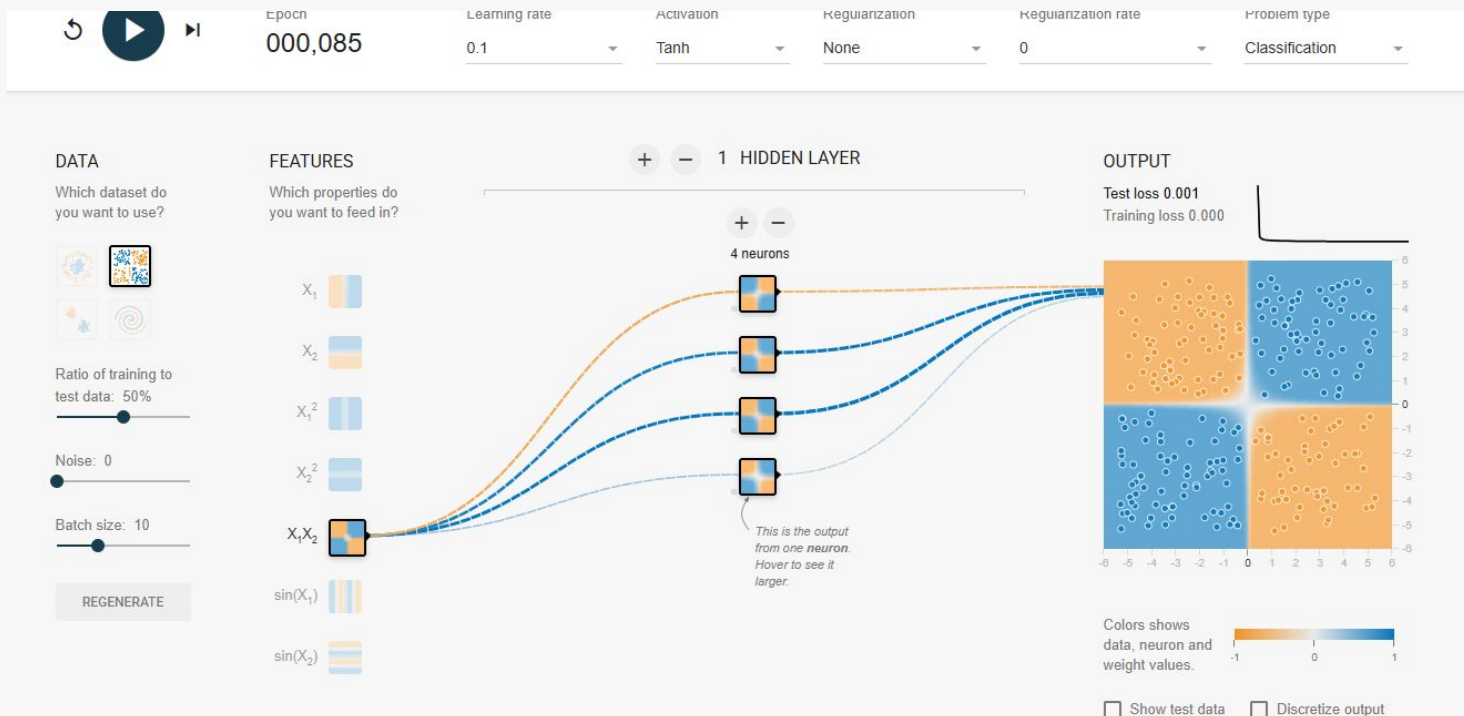
Nombre de neurones: 2



Accélération de la convergence.

Distribution Ou Exclusif

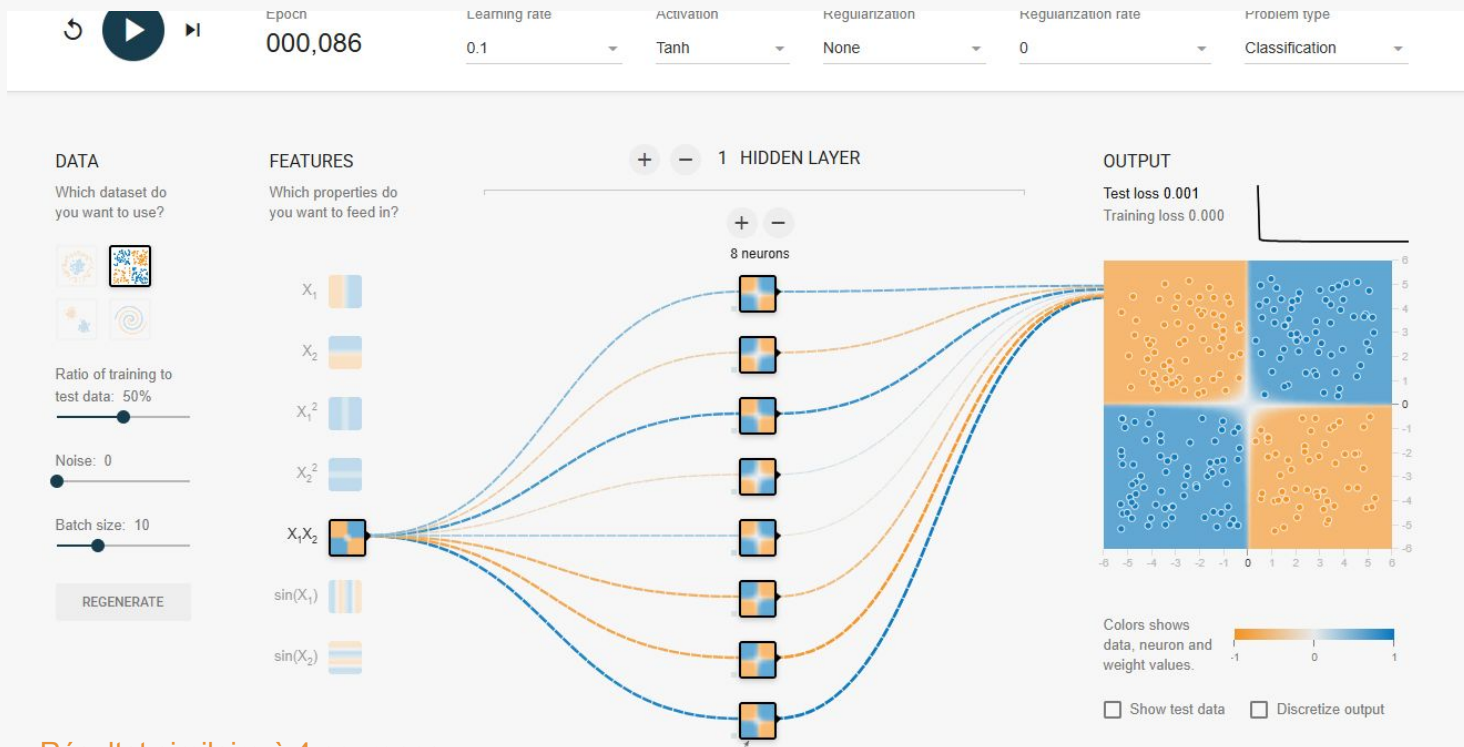
Nombre de neurones: 4



Accélération de la convergence.

Distribution Ou Exclusif

Nombre de neurones: 8



Résultat similaire à 4 neurones.

Distribution Ou Exclusif

Conclusions

Le meilleur choix de variables est X_1X_2 .

La fonction d'activation les plus adaptées sont les fonctions Tanh et Linear qui donnent des résultats équivalents.

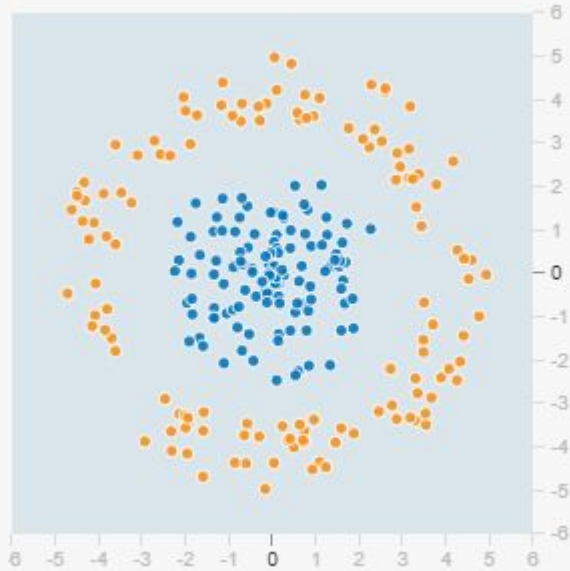
Le nombre de neurones optimal est 4.

La diminution du learning rate permet de faire converger le modèle plus rapidement.

Distribution Circulaire

Distribution Circulaire

Visualisation de la distribution

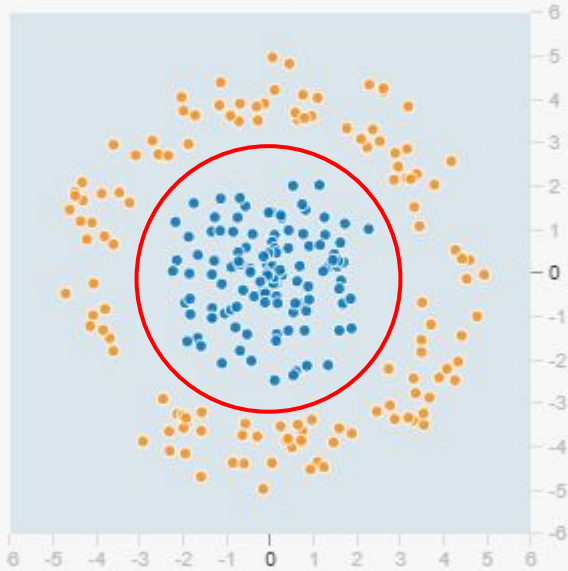


Observation

Les classes sont séparables par un cercle.

Distribution Circulaire

Visualisation de la distribution



Observation

Les classes sont séparables par un cercle.

Les variables les plus adaptées sont probablement X_1^2 et X_2^2 .

Distribution Circulaire

Choix des variables

Distribution Circulaire

Variables : X_1^2 et X_2^2

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

$X_1 X_2$

$\sin(X_1)$

$\sin(X_2)$

+ - 1 HIDDEN LAYER

+ -

1 neuron

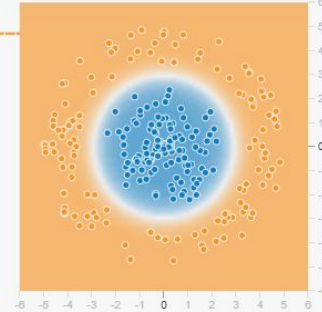


This is the output from one neuron. Hover to see it larger.

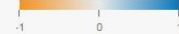
OUTPUT

Test loss 0.006

Training loss 0.004



Colors shows data, neuron and weight values.

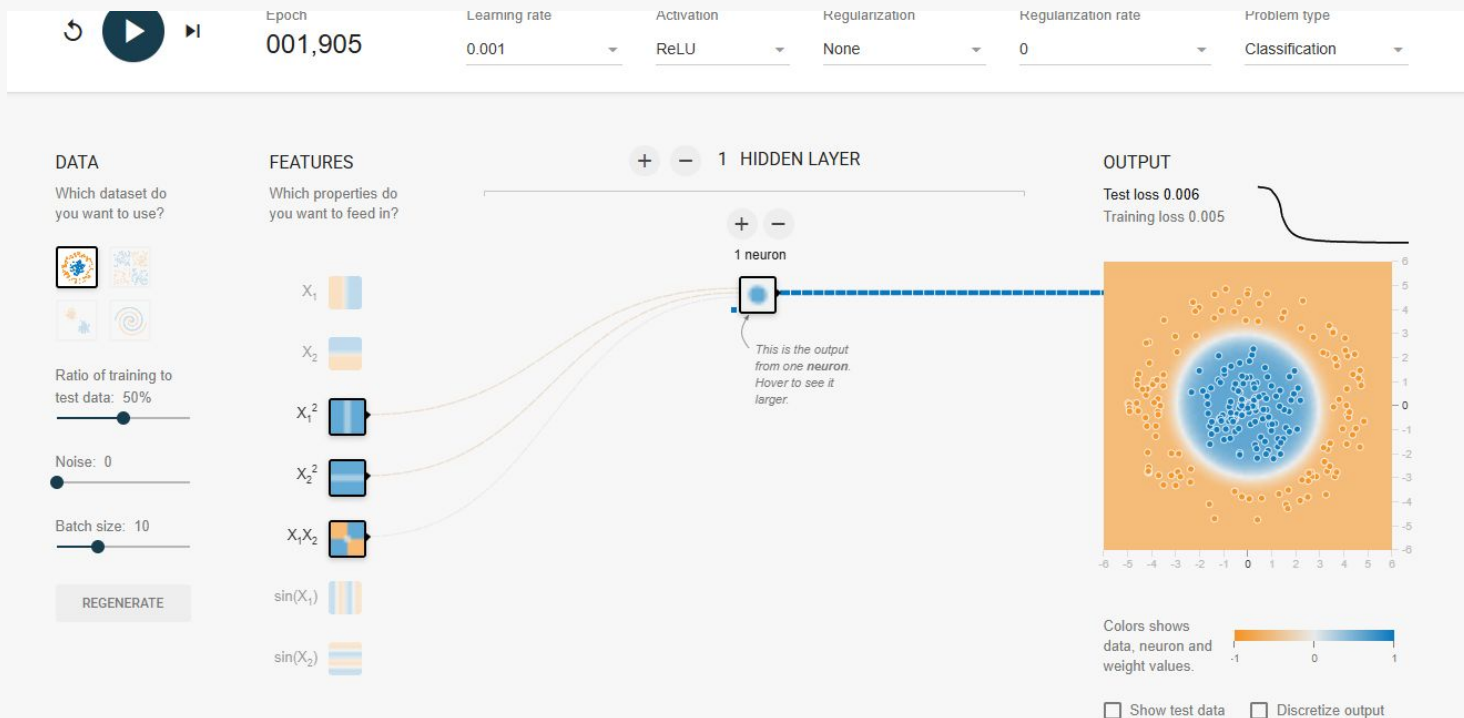


☐ Show test data

☐ Discretize output

Distribution Circulaire

Variables : X_1^2 , X_2^2 et X_1X_2



La nouvelle variable n'a pas d'influence.

Distribution Circulaire

Choix de la fonction d'activation

Distribution Circulaire

Activation: ReLu

Epoch: 001,932 Learning rate: 0.001 Activation: ReLU Regularization: None Regularization rate: 0 Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

$X_1 X_2$

$\sin(X_1)$

$\sin(X_2)$

+ - 1 HIDDEN LAYER

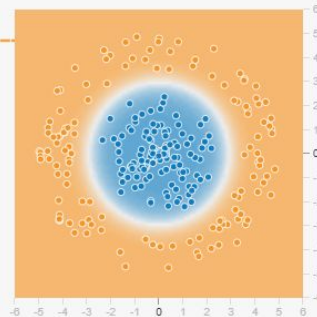
+ -

1 neuron

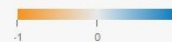
This is the output from one neuron. Hover to see it larger.

OUTPUT

Test loss 0.006
Training loss 0.004



Colors shows data, neuron and weight values.



☐ Show test data

☐ Discretize output

Distribution Circulaire

Activation: Tanh

Epoch: 001,478 Learning rate: 0.001 Activation: Tanh Regularization: None Regularization rate: 0 Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

$X_1 X_2$

$\sin(X_1)$

$\sin(X_2)$

+ - 1 HIDDEN LAYER

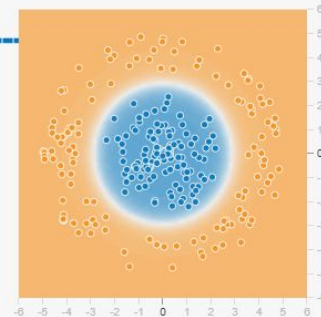
+ -

1 neuron

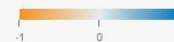
This is the output from one neuron. Hover to see it larger.

OUTPUT

Test loss 0.005
Training loss 0.004



Colors shows data, neuron and weight values.



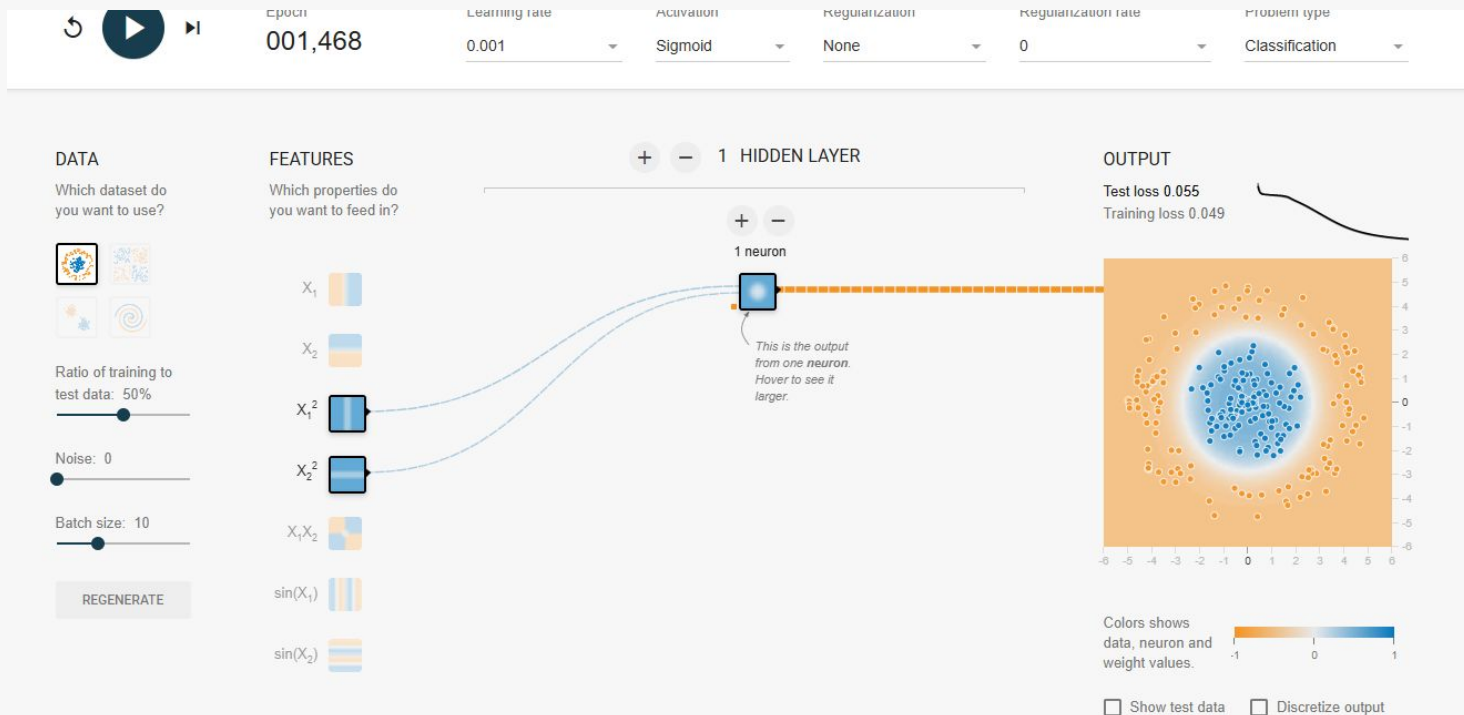
☐ Show test data

☐ Discretize output

La convergence est plus rapide.

Distribution Circulaire

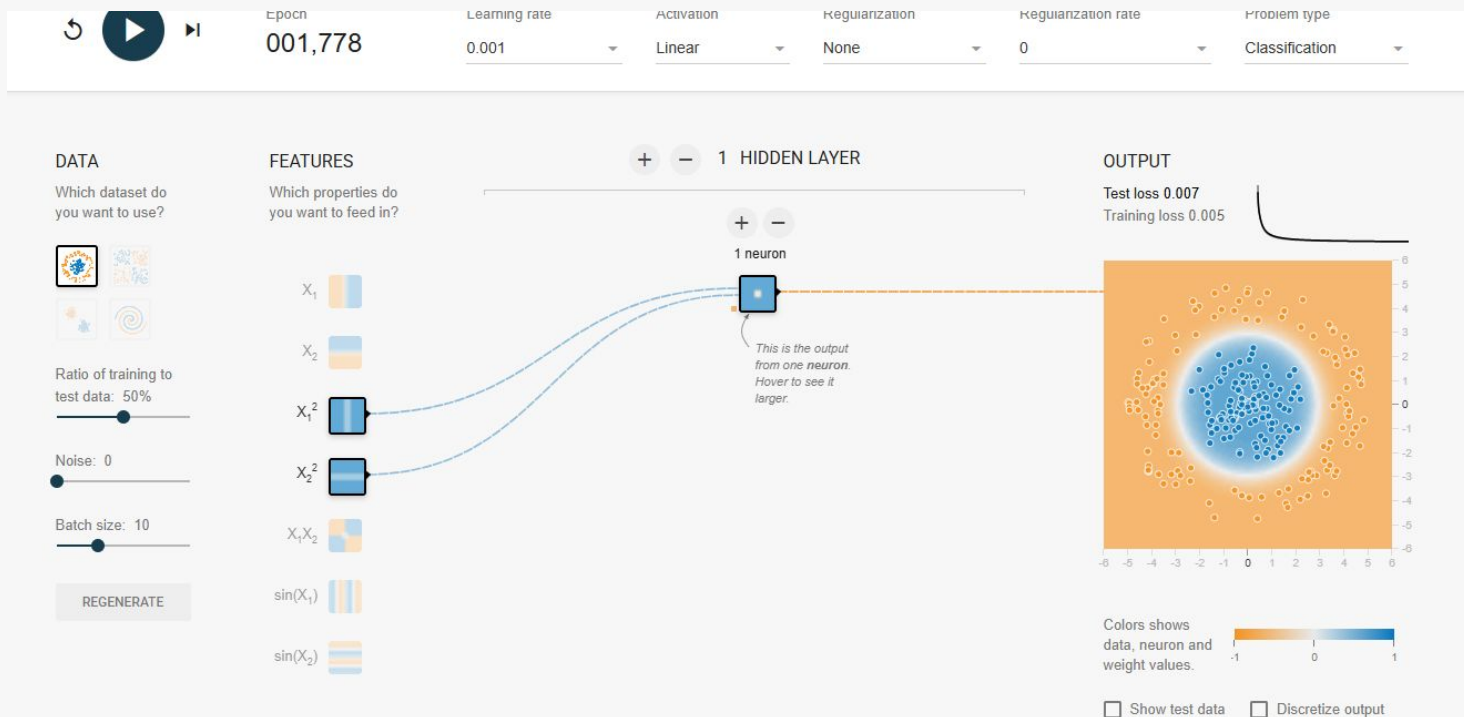
Activation: Sigmoid



Augmentation de la loss.

Distribution Circulaire

Activation: Linear



Vitesse de convergence et loss équivalentes à ReLu.

Distribution Circulaire

Choix du learning rate

Distribution Circulaire

Learning Rate: 0.001

Epoch: 001,932 Learning rate: 0.001 Activation: ReLU Regularization: None Regularization rate: 0 Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

$X_1 X_2$

$\sin(X_1)$

$\sin(X_2)$

+ - 1 HIDDEN LAYER

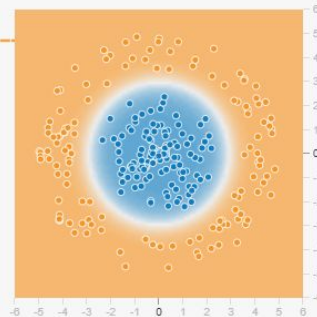
+ -

1 neuron

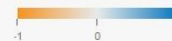
This is the output from one neuron. Hover to see it larger.

OUTPUT

Test loss 0.006
Training loss 0.004



Colors shows data, neuron and weight values.

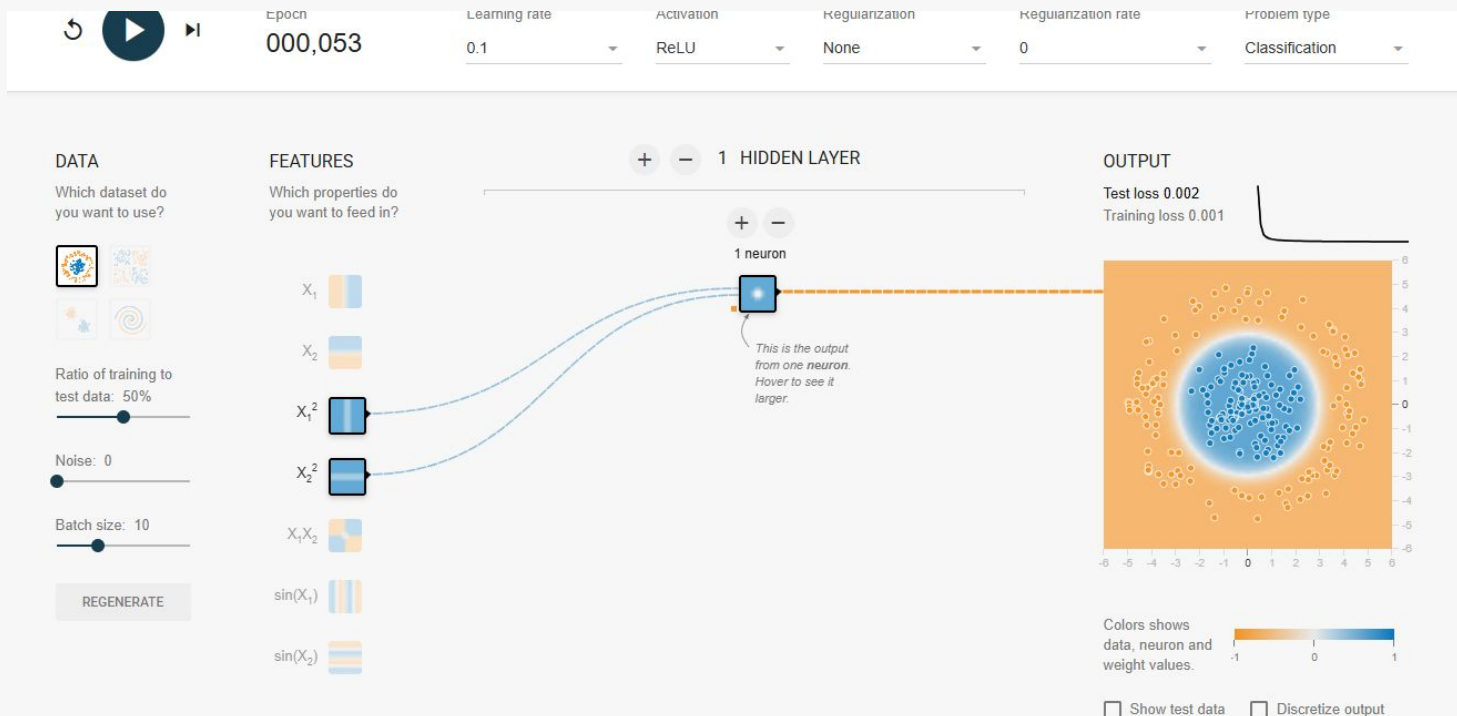


☐ Show test data

☐ Discretize output

Distribution Circulaire

Learning Rate: 0.1



Accélération de la convergence et amélioration de la loss.

Distribution Circulaire

Choix du nombre de neurones

Distribution Circulaire

Nombre de neurones: 1

Epoch: 000,053 Learning rate: 0.1 Activation: ReLU Regularization: None Regularization rate: 0 Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1

X_2

X_1^2

X_2^2

$X_1 X_2$

$\sin(X_1)$

$\sin(X_2)$

+ - 1 HIDDEN LAYER

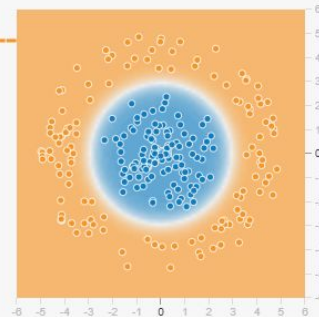
+ -

1 neuron

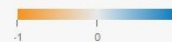
This is the output from one neuron. Hover to see it larger.

OUTPUT

Test loss 0.002
Training loss 0.001



Colors shows data, neuron and weight values.

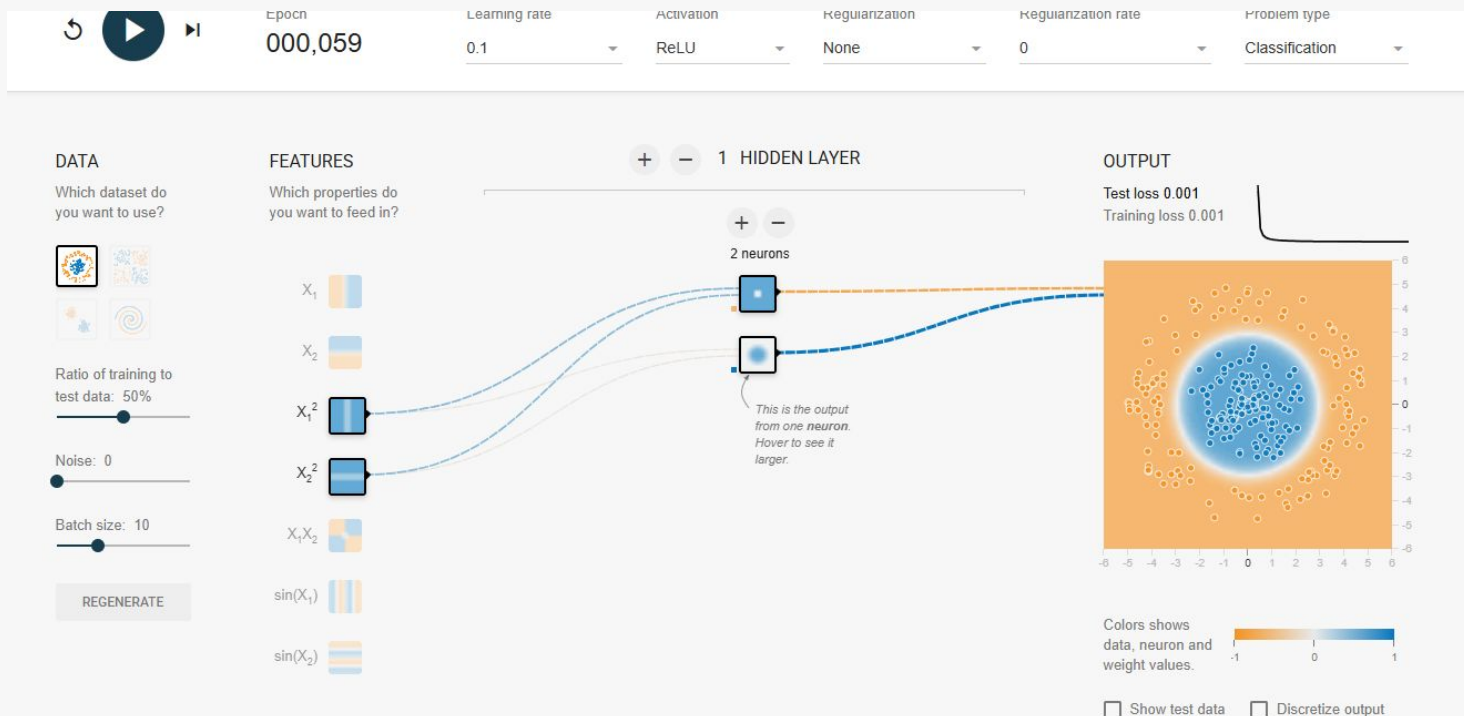


☐ Show test data

☐ Discretize output

Distribution Circulaire

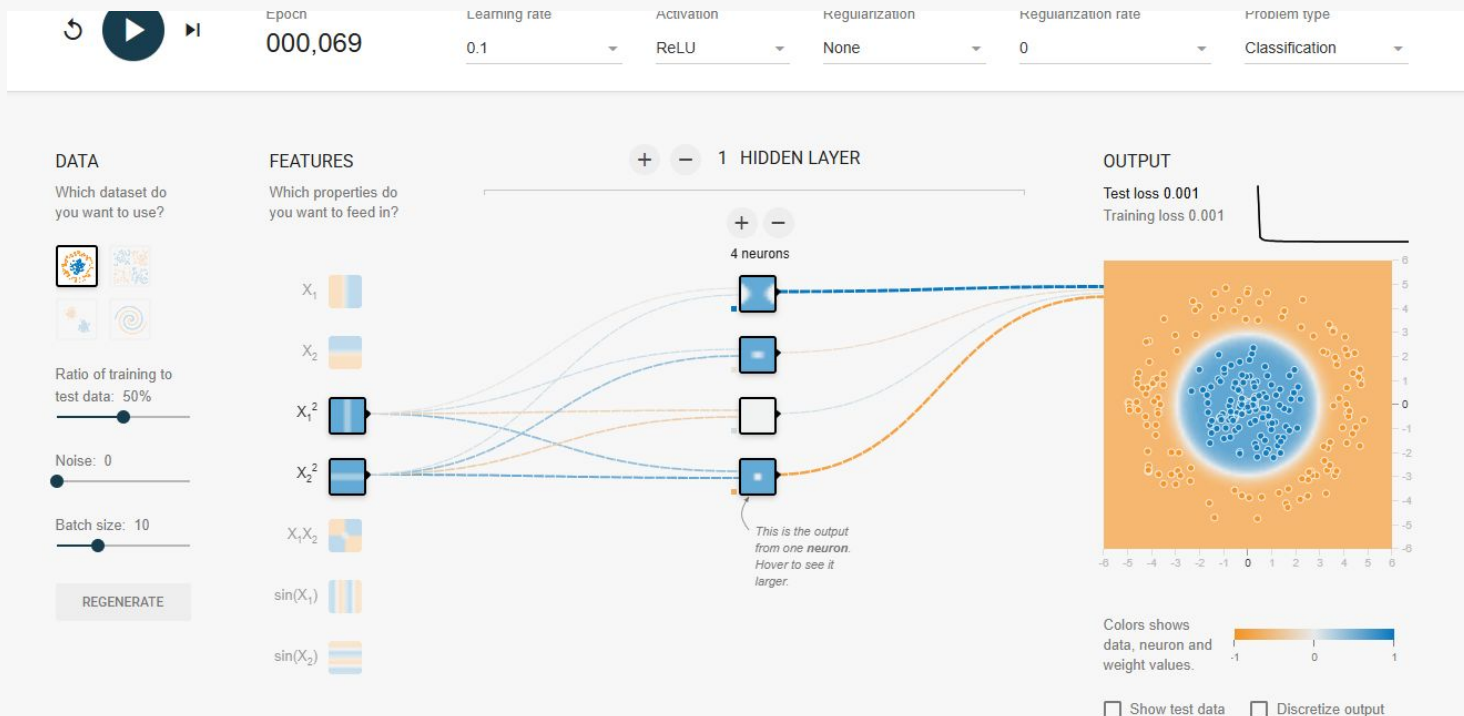
Nombre de neurones: 2



Amélioration de la loss.

Distribution Circulaire

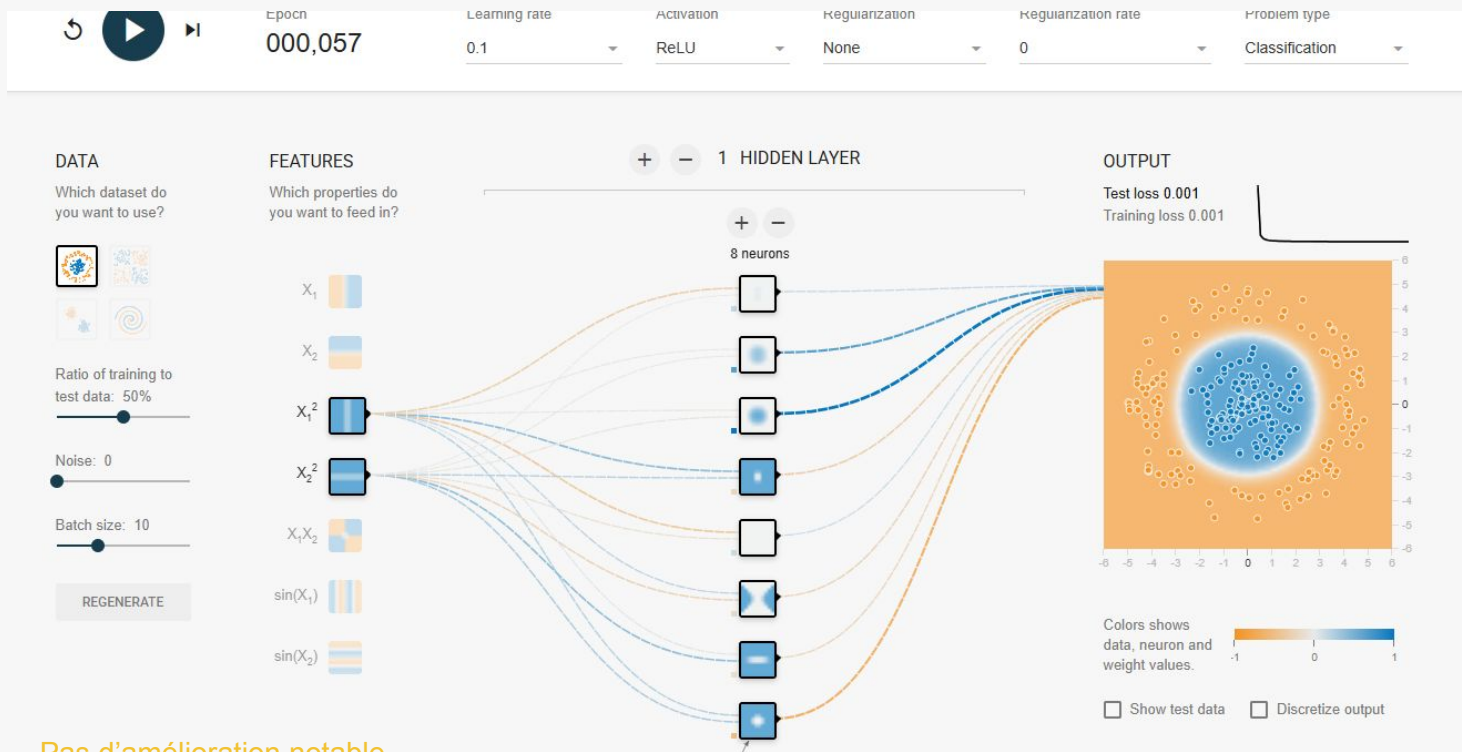
Nombre de neurones: 4



Pas d'amélioration notable.

Distribution Circulaire

Nombre de neurones: 8



Pas d'amélioration notable.

Distribution Circulaire

Conclusions

Le meilleur choix de variables est X_1^2 et X_2^2 .

La fonction d'activation la plus adaptée est Tanh.

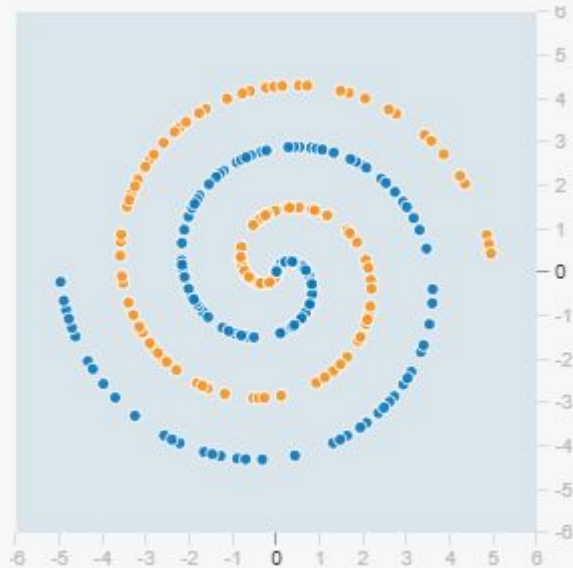
Le nombre de neurones optimal est 2.

La diminution du learning rate permet de faire converger le modèle plus rapidement.

Distribution en Spirale

Distribution en Spirale

Visualisation de la distribution

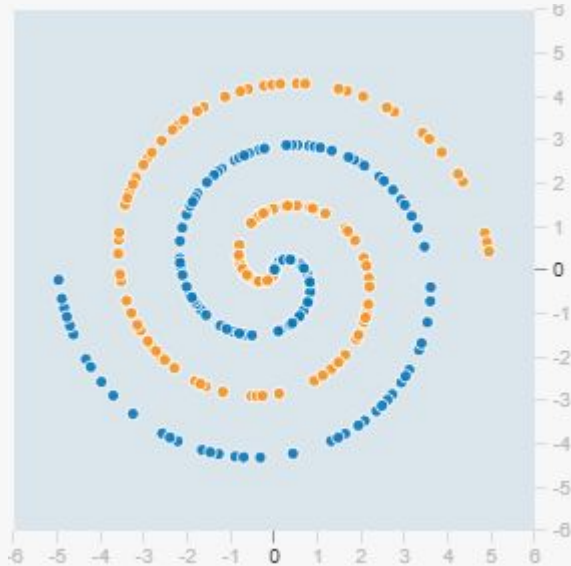


Observation

Les classes sont séparables par une spirale.

Distribution en Spirale

Visualisation de la distribution



Observation

Les classes sont séparables par une spirale.

On va essayer directement avec un couche de 8 neurones.

Distribution en Spirale

Choix des variables

Distribution en Spirale

Variables : X_1^2 et X_2^2

Epoch: 000,811 Learning rate: 0.001 Activation: ReLU Regularization: None Regularization rate: 0 Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

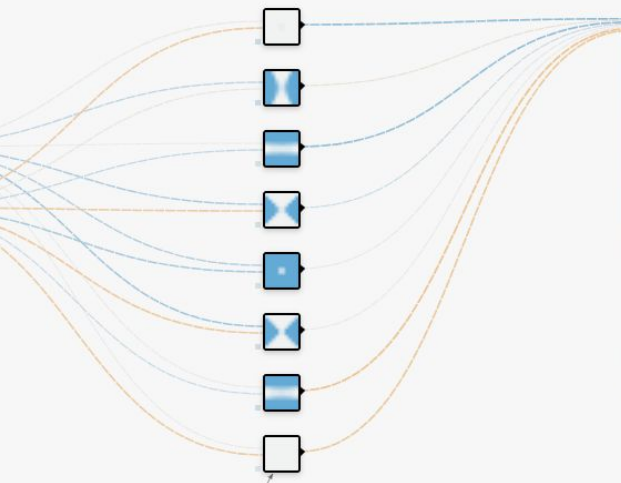
FEATURES

Which properties do you want to feed in?

- X_1
- X_2
- X_1^2
- X_2^2
- $X_1 X_2$
- $\sin(X_1)$
- $\sin(X_2)$

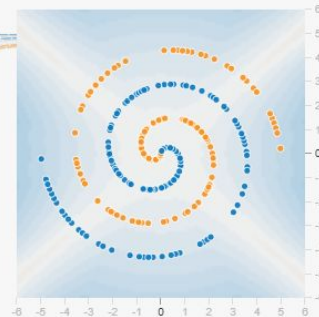
1 HIDDEN LAYER

8 neurons

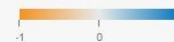


OUTPUT

Test loss 0.510
Training loss 0.498



Colors shows data, neuron and weight values.



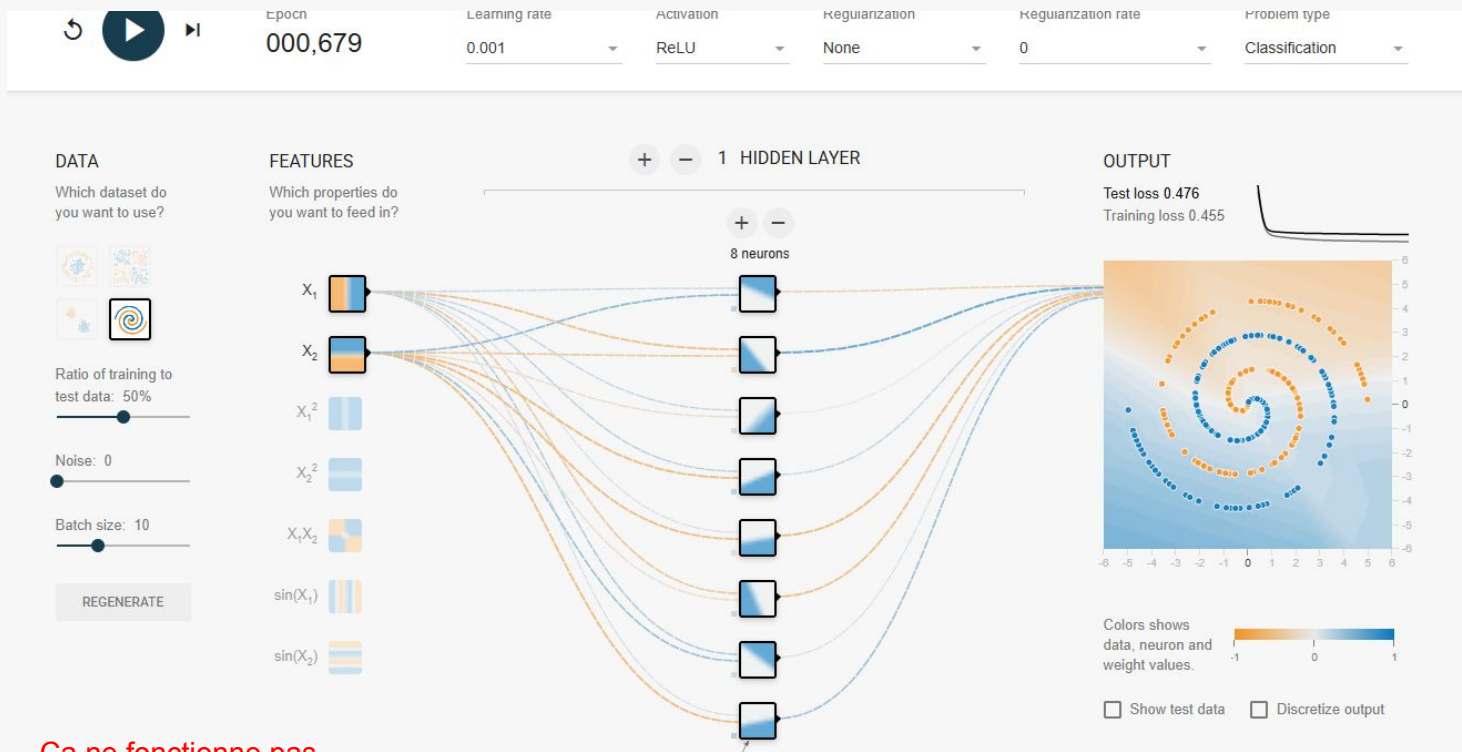
☐ Show test data

☐ Discretize output

Ca ne fonctionne pas.

Distribution en Spirale

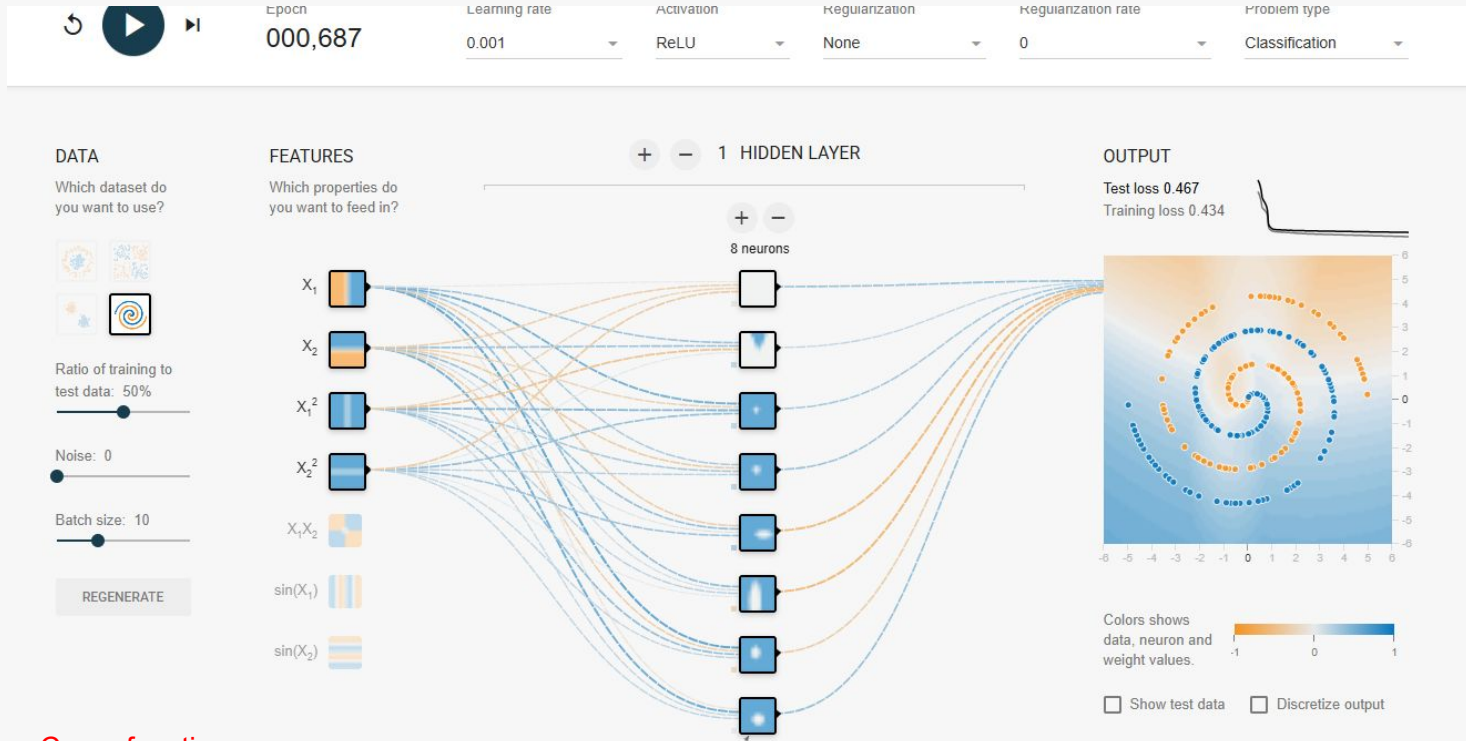
Variables : X_1 et X_2



Ca ne fonctionne pas.

Distribution en Spirale

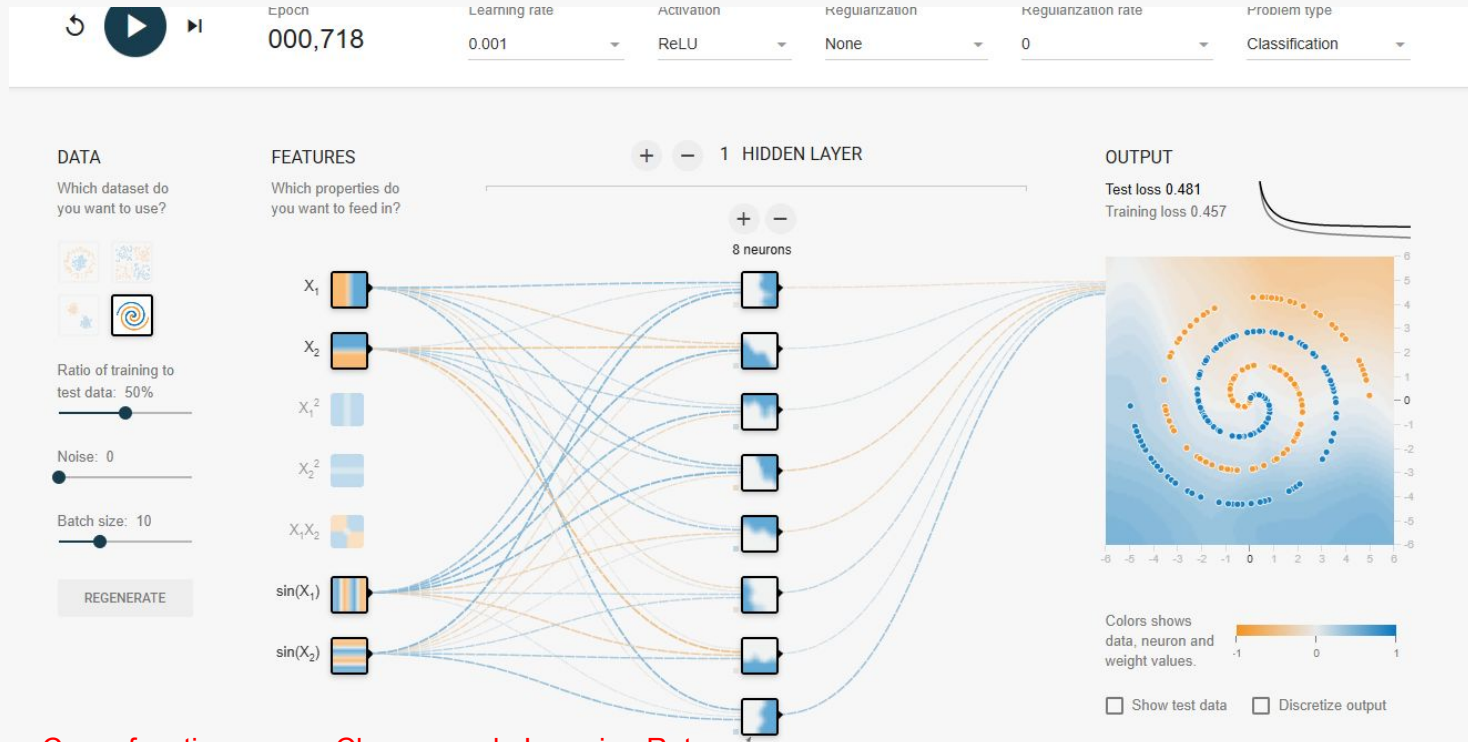
Variables : X_1 , X_2 , X_1^2 et X_2^2



Ca ne fonctionne pas.

Distribution en Spirale

Variables : X_1 , X_2 , $\sin(X_1)$ et $\sin(X_2)$



Ca ne fonctionne pas. Changeons le Learning Rate.

Distribution en Spirale

Learning Rate: 0.01

Epoch: 001,187 Learning rate: 0.01 Activation: ReLU Regularization: None Regularization rate: 0 Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

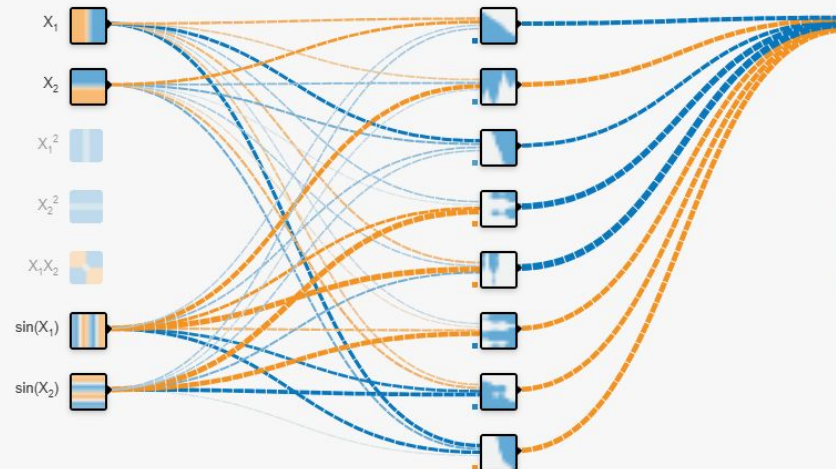
FEATURES

Which properties do you want to feed in?

X_1
 X_2
 X_1^2
 X_2^2
 $X_1 X_2$
 $\sin(X_1)$
 $\sin(X_2)$

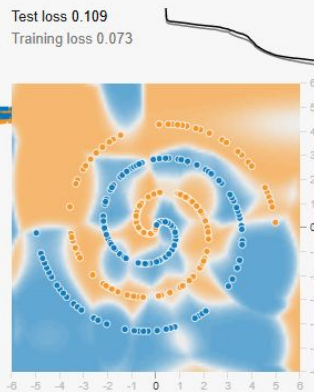
+ - 1 HIDDEN LAYER

+ -
8 neurons

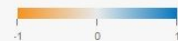


OUTPUT

Test loss 0.109
Training loss 0.073



Colors shows
data, neuron and
weight values.



☐ Show test data

☐ Discretize output

Amélioration de la loss et de la forme de la frontière. Changeons la fonction d'activation pour essayer de la lisser.

Distribution en Spirale

Activation: Tanh

Epoch: 000,990 Learning rate: 0.01 Activation: Tanh Regularization: None Regularization rate: 0 Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

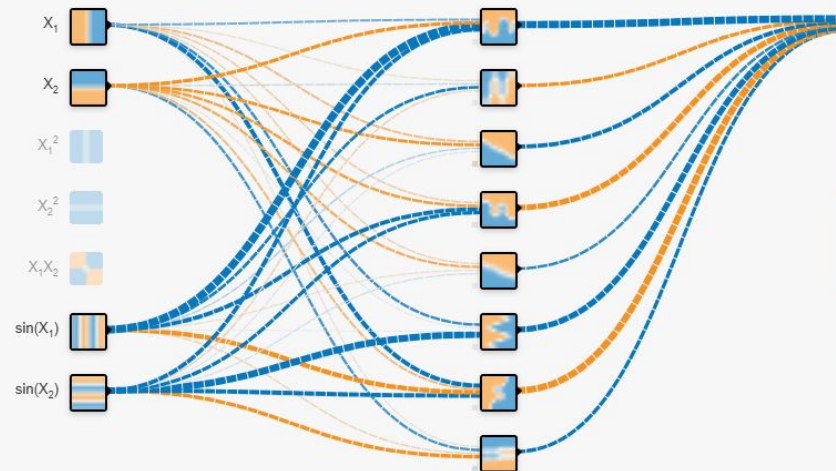
Which properties do you want to feed in?

X_1
 X_2
 X_1^2
 X_2^2
 $X_1 X_2$
 $\sin(X_1)$
 $\sin(X_2)$

+ - 1 HIDDEN LAYER

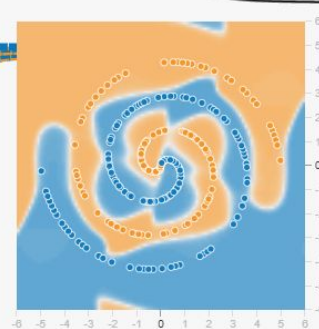
+ -

8 neurons

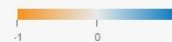


OUTPUT

Test loss 0.024
Training loss 0.016



Colors shows data, neuron and weight values.



☐ Show test data

☐ Discretize output

Amélioration de la loss et de la forme de la frontière.
Augmentons le nombre de couches de neurones.

Distribution en Spirale

Nombre de couches: 2

Epoch: 001,166 Learning rate: 0.01 Activation: Tanh Regularization: None Regularization rate: 0 Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

Batch size: 10

REGENERATE

FEATURES

Which properties do you want to feed in?

X_1
 X_2
 X_1^2
 X_2^2
 $X_1 X_2$
 $\sin(X_1)$
 $\sin(X_2)$

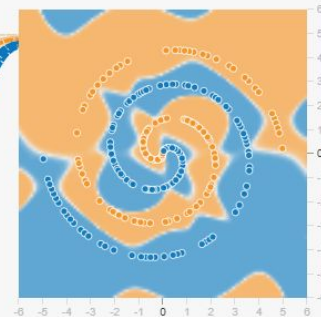
+ - 2 HIDDEN LAYERS

+ -
8 neurons

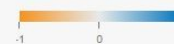
+ -
8 neurons

OUTPUT

Test loss 0.009
Training loss 0.005



Colors shows data, neuron and weight values.



☐ Show test data

☐ Discretize output

Amélioration de la loss et de la forme de la frontière.
Augmentons le nombre de couches de neurones.

Distribution en Spirale

Nombre de couches: 3



Amélioration de la vitesse de convergence.
Diminuons le Learning Rate.

Distribution en Spirale

Learning Rate: 0.1

Epoch: 000,802 Learning rate: 0.1 Activation: Tanh Regularization: None Regularization rate: 0 Problem type: Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%

Noise: 0

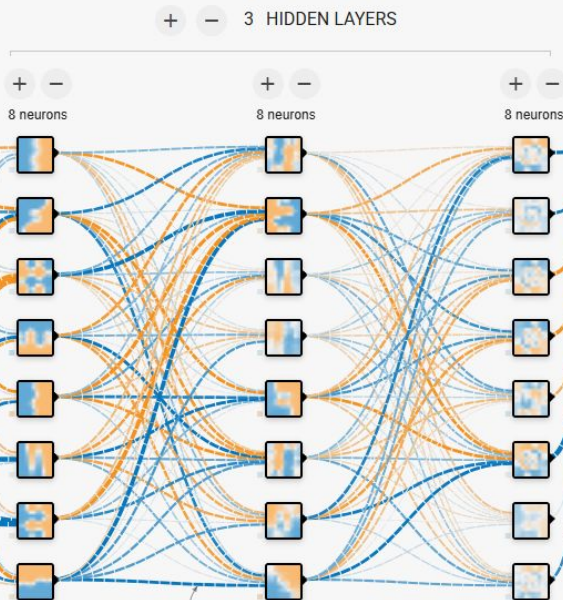
Batch size: 10

REGENERATE

FEATURES

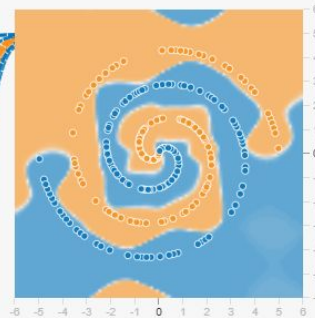
Which properties do you want to feed in?

X_1
 X_2
 X_1^2
 X_2^2
 $X_1 X_2$
 $\sin(X_1)$
 $\sin(X_2)$

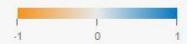


OUTPUT

Test loss 0.010
Training loss 0.000



Colors shows data, neuron and weight values.



☐ Show test data

☐ Discretize output

La vitesse de convergence a chuté
et le modèle sur-apprend.

Distribution en Spirale

Conclusions

Le meilleur choix de variables est X_1 , X_2 , $\sin(X_1)$ et $\sin(X_2)$.

La fonction d'activation la plus adaptée est Tanh.

Le nombre de couches de neurones optimal est 3.

La diminution du learning rate permet de faire converger le modèle plus rapidement mais il ne faut pas que sa valeur soit trop élevée.