Model Answer Exercise 1

For the first exercise of the day, it is proposed that you come back to the demo software: https://playground.tensorflow.org, and we will use the spiral dataset again.

On Wednesday we ignored the regularization parameter. Today, we are going to evaluate the role this parameter can play. The input data – or features - will be the two coordinates X_1 and X_2. The final map will not be displayed in "Discrete Output Mode" in order to better understand how the output of the neural network varies.

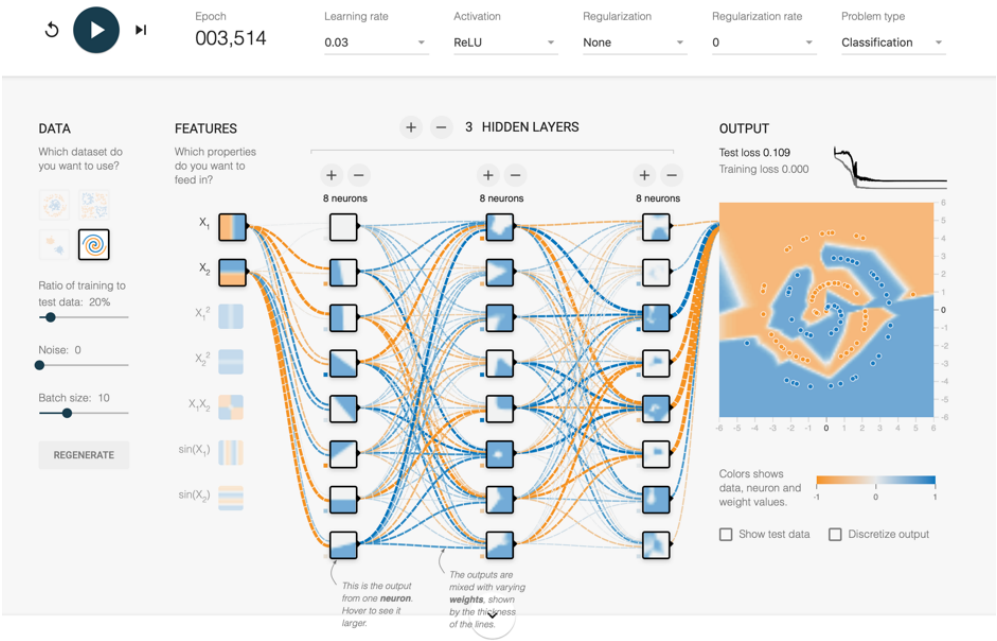
We will keep the following hyper-parameters fixed:

- Number of hidden layers (3)
- Number of neurons per layer (8)
- There is a bias term in each neuron calculation
- Learning rate: 0.03
- ReLU activation

1. How many neural networks parameters are fitted with these hyperparameters?
   The total number of parameters (there are bias terms) is:
   (2+1)x8 + (8+1)x8 + (8+1)x8 + (8+1)x1 = 177

2. If we take the ratio of training to test equal to 20%, how many training data points do we have? Do you expect overfitting?
   Since we have a total of 500 points, the training set is 20% of this, that is only 100 points. Considering that we have almost twice more parameters than the number of data points, we expect some overfitting.

3. Try one run to see what happens with the 20% ratio of training data if we calculate the neural network with no regularization. What do you observe?
   We observe a somewhat chaotic behaviour, which is a sign of overfitting (see image below). Actually, this is a great example of overfitting. Thanks to the large number of parameters and the small number of training data, the neural network manages to perfectly fit the training data after about 3500 epochs. But the test error remains significant, which confirms overfitting.



4. Now we are going to run the program with different regularization parameters. Write the mathematical expression of the loss function for the two configurations L1 and L2.
The mathematical expression is as given during the presentation this morning.

If the L2 norm is used:

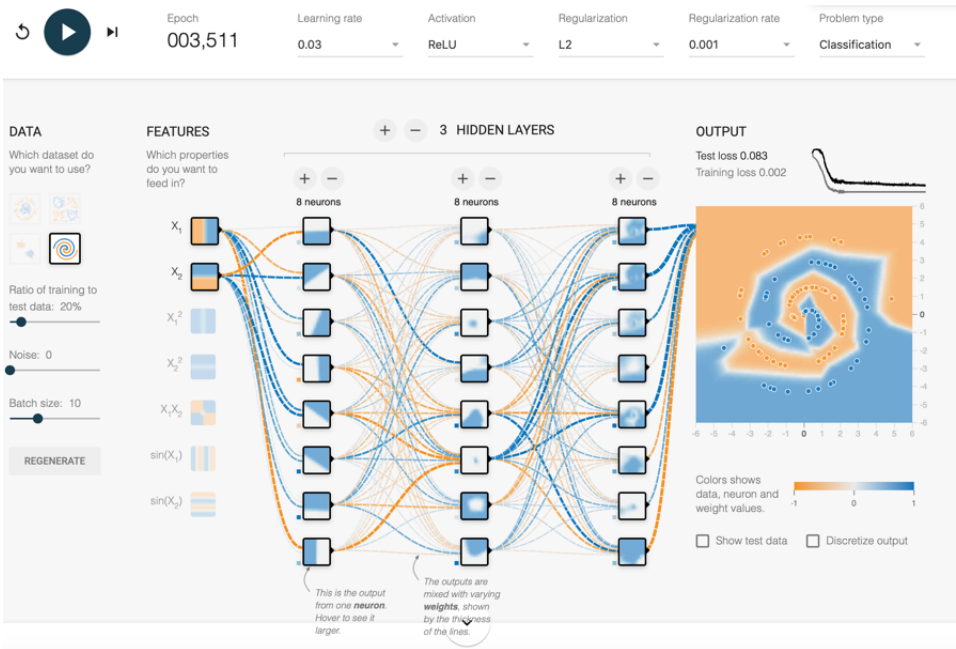
$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(h_\theta(x^{(i)})\right) + (1-y^{(i)})\log\left(1-h_\theta(x^{(i)})\right)\right] + \frac{\lambda}{2m}\sum_{j}\theta_j^2$$

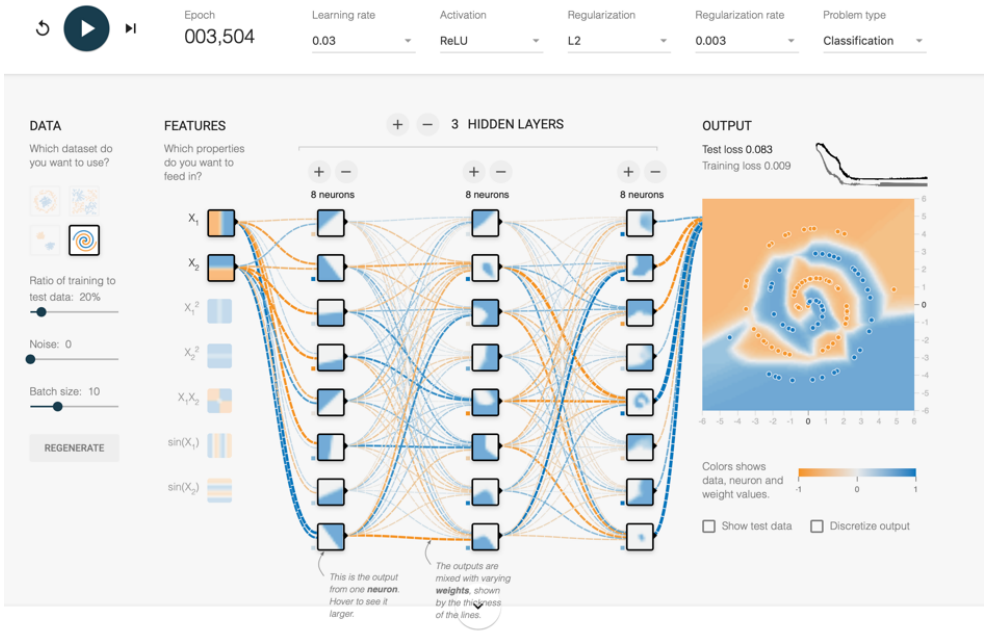
If the L1 norm is used :

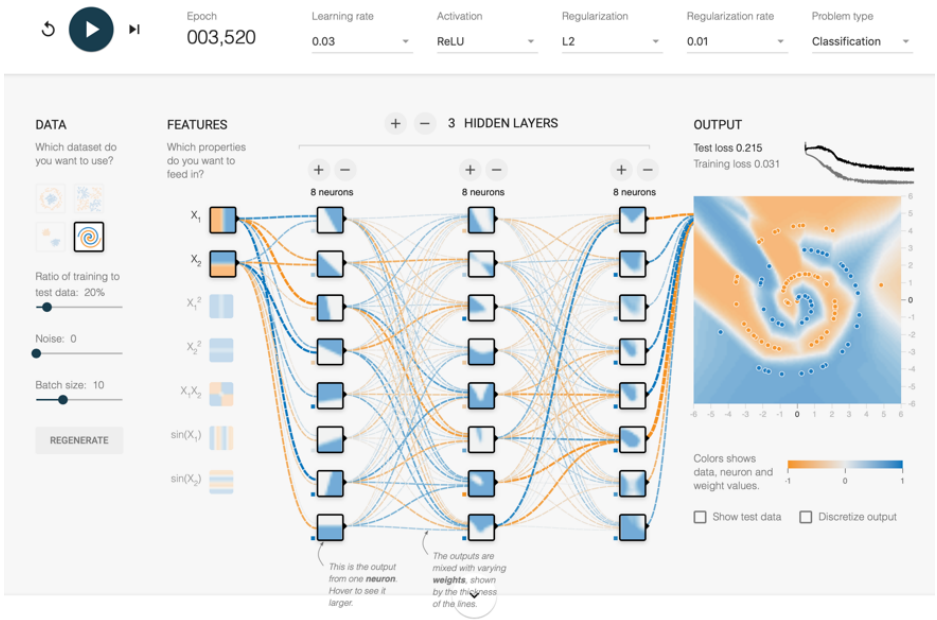
$$J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(h_\theta(x^{(i)})\right) + (1-y^{(i)})\log\left(1-h_\theta(x^{(i)})\right)\right] + \frac{\lambda}{2m}\sum_{j}|\theta_j|$$

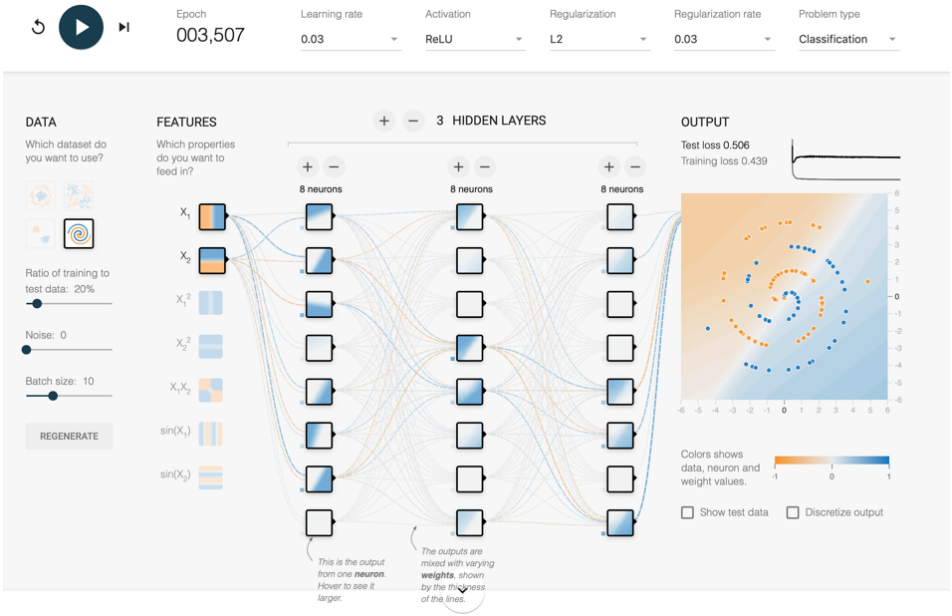
5. Recalculate the network for L2 optimization for values of the regularization parameter equal to 0.001, 0.003, 0.01 and 0.03. What do you observe in each case?
We observe that the rather chaotic behaviour obtained when there is no regularization is improved for values of the regularization parameter equal to 0.001 and 0.003, for which we see that the neural network is starting to understand the spiral shape of the dataset. The training data are not matched perfectly anymore, and the difference between the test and training error decreases as the regularization parameter changes from 0.001 to 0.003. Unfortunately the software does not allow testing values of the regularization parameter between 0.003 and 0.01. We observe that the weight decay term becomes too strong when the regularization parameter reaches values of 0.01 and 0.03, as it is clear that the hidden layers parameters (illustrated by the dotted lines

joining the neurons of different layers) do not vary enough to accommodate the data. Both the training loss and the test loss are significantly larger than for the two previous values of the regularization parameters. We are clearly in a situation of bias or underfitting.



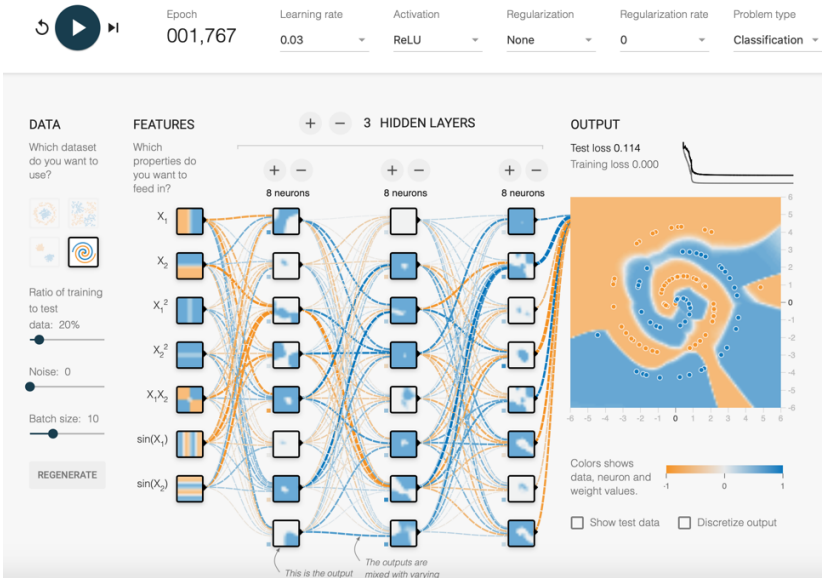


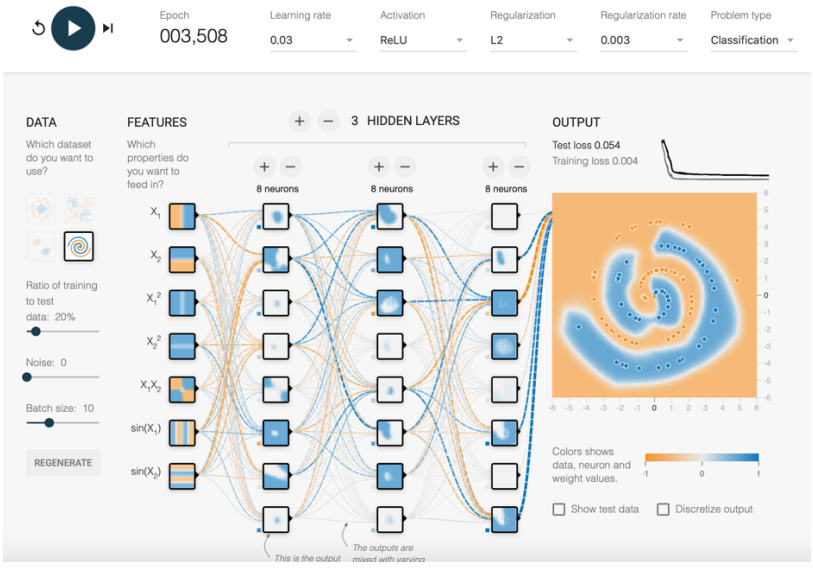




6. Now repeat the operation with seven instead of two input data (or features) : $X_1, X_2, X_1^2, X_2^2, X_1X_2, \sin X_1, \sin X_2$, keeping all the hyperparameters unchanged. First calculate the network without regularization, then calculate it with a 0.03 regularization coefficient. What do you observe?

The first image is obtained without regularization. With seven input features we have even more training parameters (8x8+9x8+9x8+9=217) than before, hence we expect even more overfitting than with just two input features. The training error reaches zero after about 1500 epochs, but the test error is quite high, which confirms overfitting. With L2 regularization and a regularization parameter equal to 0.003, the image is smoother, as expected. The training loss has increased and the test loss has decreased, as expected, and they are both lower that in the previous case of just two input features.





This is the link to the last run:
https://playground.tensorflow.org/#activation=relu&regularization=L2&batchSize=10&dataset=spiral&regDataset=reg-plane&learningRate=0.03&regularizationRate=0.003&noise=0&networkShape=8,8,8&seed=0.66894&showTestData=false&discretize=false&percTrainData=20&x=true&y=true&xTimesY=true&xSquared=true&ySquared=true&cosX=false&sinX=true&cosY=false&sinY=true&collectStats=false&problem=classification&initZero=false&hideText=false