|  |  |  |
| --- | --- | --- |
|  | **Experiments with Machine Learning** |  |

# 

## For the following activities you will be making observations about various machine learning applications and experiments.

[Teachable Machine](https://teachablemachine.withgoogle.com/)

Go through the teachable machine tutorial.

* Does your machine successfully show different gifs for different motions?

Yes it does.

* What happens if you only have one field trained? Why does it do this?

You get very skewed results because it doesn't know what the other field is.

* In the input field, the experiment is showing your web camera. In the learning field the machine is taking still images. How does it use the still images to compare to the input of a video to produce an output? (Don’t overthink this, what is a video? )

The video changes and is constantly moving causing the machine trouble because it isn't still.

* Once the network is trained, could you replace the Green, Red, and Orange fields with the word neurons and have this be an accurate model? Explain your answer:

Yes because you can train it to know the different new fields.

[Slice of Machine Learning](https://sliceofml.withgoogle.com/#/)

Run through the experiment twice.

On the second run, make sure to vary the training data/ test data.

* Describe the relationship between training and test data. How does it affect accuracy?

When you train it more than testing it, it fails. When you test it more than training it, it fails. When you do about 60-40 for the training, it works.

* Run the experiment again and test altering the features. Make two lists of features that improve the accuracy of your model and ones that do not.
  + Improve

Round, Sauce

* + Do not improve

Cheese, Sausage

* What do you believe are features that help this neural network to be successful?

The fact that it can best identify the size and colors of toppings.

[TensorFlow Playground](https://playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=circle&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=0&networkShape=4,2&seed=0.32293&showTestData=false&discretize=false&percTrainData=50&x=true&y=true&xTimesY=false&xSquared=true&ySquared=true&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=classification&initZero=false&hideText=false)

Try altering the features, neurons, and hidden layers in the model. Use only 10 and 25 Epochs for these tests. Set the activation field to Sigmoid for these tests.

* What does altering the features of the neural network do to the output test loss and training loss fields?

They cause more loss in both areas. It causes more test loss.

* What does adding hidden layers do to the output output test loss and training loss fields?

It causes about even test and training loss.

* What does adding or removing neurons do to the output output test loss and training loss fields?

It causes the loss of both test and training to be the same and even out.

* Use these three fields and try to optimize the loss to be as low as you can make them. Record the lowest loss you achieve and the settings that you used.
  + Lowest loss: 0
  + Number of features: 1
  + Number of hidden layers: 1
  + Number of neurons in each layer: 1
* Using what you learned from the Slice of machine learning lab. Change the ratio of training to test data. Try to optimize the network further. Explain what happened:

It caused a short increase of loss and then evened out to 0.

* Was this ratio as successful in this model as in the Slice of machine learning?

No it didn’t cause much change except in loss.

* Run another test using a Regression Problem Type. What is different about this output?

This test caused lots of loss that got smaller, but never got to 0.

[Style Transfer and Deep Art Explanation Video](https://www.youtube.com/watch?v=du_YnoC6dKU)

[Google Infinite Patterns/Deep Dream Video](https://experiments.withgoogle.com/infinitepatterns)

[DeepArt](https://deepart.io/#) [TensorFire](https://tenso.rs/demos/fast-neural-style/)  [Infinite Patterns](https://colab.research.google.com/github/tensorflow/lucid/blob/master/notebooks/differentiable-parameterizations/appendix/infinite_patterns.ipynb) [DeepDream](https://dreamscopeapp.com/deep-dream-generator)  [Fast Style](https://deepai.org/machine-learning-model/fast-style-transfer) [Neural Style](https://deepai.org/machine-learning-model/neural-style)

Experiment with DeepArt, TensorFire, and DeepDream. Infinite Patterns, Fast and Neural style are alternatives to DeepArt. Paste two examples of images that you have made in each platform below. For DeepDream use the “SuperTrippy” filter.



* Examples

* Using what you have learned about neural networks thus far, how do you believe that these platforms produce their output? To the best of your ability describe what the neural networks are doing. DeepDream is different from the other two. 

I believe they try to run different weights between the two pictures. I think they are looking for similarities on each picture and then applying them to each other.

* Are these networks examples of supervised or unsupervised learning? Explain your answer:

Unsupervised because you feed in the content but you don’t tell them what each thing means.