# Project 3: You Always Cut the Red Wire

01:198:440

It is another day on the deep space vessel *Archaeopteryx*, and someone has accidentally activated the self-destruct mechanism. It's up to you to save the ship. Fortunately, you have the manual.

The manual contains pages and pages of complex wiring diagrams, each labeled as 'safe' or 'dangerous', and for the 'dangerous' ones, which wire needs to be cut. But there is no general rule listed in the manual - you'll need to try to learn what makes a wiring safe or dangerous, and what needs cutting, based on the examples in the manual.

## 1 The Data Set

Each diagram is a 20x20 pixel image, that has been generated in the following way:

- Starting with rows, pick a random number from 1 to 20 and a random color from {Red, Blue, Yellow, Green}, and color that row that color.
- Switching to columns, pick a random number from 1 to 20 and a random color (different from the first), and color that column that color, coloring over the previously colored row at the point of intersection.
- Switching back to rows, pick a random number from 1 to 20 (different from the first randomly selected row), and a random color (different from the first two), and color that row that color, coloring over the previously colored column at the point of intersection.
- Switching back to columns, pick a random number from 1 to 20 (different from the first randomly selected column), and take the remaining unselected color, and color that column that color, coloring over the previously colored columns at the points of intersection.
- *Note: The same instructions should be applied starting with columns instead of rows, with* 50% *chance.*

According to my math, there are 6931200 possible wiring diagrams given the above instructions. Your training data set will be based on much fewer than that.

If the Red wire is laid before a Yellow wire, this image is marked as 'Dangerous', otherwise it is marked as safe.

Among the images marked 'Dangerous', the wire color to cut is always the 3rd wire laid down.

## 2 The First Task

Build a train a model on the data set to take as input a wiring diagram and give as output whether or not it is dangerous. Your writeup should include the following:

- How are you defining your input space?
- How are you defining your output space?
- What model space are you considering, and what parameters does it have? Be sure to specify any design choices you make here.
- How are you measuring the loss or error of a given model?

- What training algorithm are you using to find the best model you can? Include any necessary math to specify your algorithm.
- How are you preventing overfitting?

Your writeup should also include:

- Performance of your model on a training set of 500 examples, $1k$ examples, $2.5k$ examples, and $5k$ examples.
- A graph of your model's loss over time, demonstrating learning.
- Assessment of your trained model, and evidence that it is not overfit to the data, and instead generalizes well.

- For full credit, your model should include some non-linear features, determined in a non-ad hoc way.
- You must code and train everything yourself - no prebuilt libraries for this project. You are welcome to use a library for vectors and linear algebra (e.g., numpy) - but what you do with those vectors and linear algebra, you must write yourself.
- You **do not** need to build a fully generalized neural network library with layer objects etc for this project. It is unnecessary for the project, it will overcomplicate implementing and debugging, and I will take off points if you do.

## 3 The Second Task

Build a model on the data set to take as input a wiring diagram that is dangerous, and give as output which of the four colors of wire should be cut. Your writeup should include the following:

- How are you defining your input space?
- How are you defining your output space?
- What model space are you considering, and what parameters does it have? Be sure to specify any design choices you make here.
- How are you measuring the loss or error of a given model?
- What training algorithm are you using to find the best model you can? Include any necessary math to specify your algorithm.
- How are you preventing overfitting?

Your writeup should also include:

- Performance of your model on a training set of 500 examples, $1k$ examples, $2.5k$ examples, and $5k$ examples.
- A graph of your model's loss over time, demonstrating learning.
- Assessment of your trained model, and evidence that it is not overfit to the data, and instead generalizes well.

- Again - for full credit, your model should include some non-linear features, determined in a non-ad hoc way.
- Again - you must code and train everything yourself - no prebuilt libraries for this project. You are welcome to use a library for vectors and linear algebra (e.g., numpy) - but what you do with those vectors and linear algebra, you must write yourself.
- Again - you **do not** need to build a fully generalized neural network library with layer objects etc for this project. It is unnecessary for the project, it will overcomplicate implementing and debugging, and I will take off points if you do.

# 4 Bonus:

As a bonus, you may do the above in an established machine learning framework (pytorch is my personal preference right now, tensorflow or jax are also popular choices).

Additionally: in the machine learning framework of your choice, experiment, and find the smallest model that you can that successfully solves the problem when trained on minimal data. Your writeup should include evidence to validate your claim.