



Narrative Machine Learning Technical Test

Overview

This test is designed to be an opportunity for you to display your technical ability in architecture, training, and validation of an image vision/model. The test is broken down into three levels to make it a bit less overwhelming and guide you in the steps to take when working on the solution.

You have up to 3 days to submit a solution after receiving this email.

- Do not work on this for 3 days: The time-frame is designed to create equal opportunity and reduce bias and influence. It is not meant to represent 3 days of total effort, rather it is to give you flexibility and some reflection time along the way. Please don't spend too much time on this. You'll see significant diminishing returns.
- Words are stronger than CNN: In other words, we are not looking for a revolutionary ground-breaking solution to this problem. Rather, we want to see your thinking process, decision making, trade-offs, etc. A page of notes holds a ton of weight with us. Please reference any academic research as it relates to your solution. ChatGPT is not acceptable as a reference.
- ChatGPT is not going to solve this for you: ChatGPT is an amazing tool that can write code really well. You are welcome to use it as one of many tools in your arsenal when working on this test, just as you might use it once you are employed. However, if you simply take the solution offered by ChatGPT that will be: (a) extremely obvious, (b) produce a solution that will be far from optimal, (c) will not contain the nuance and clever thinking born from years of experience in this field that you certainly can bring to the table.

- This is not a trick: We are not trying to play games with you and there are no hidden traps for you to find. It is simply meant to be a concise contained example for you to demonstrate how you think and operate.
- Life happens: You are a busy person, with deadlines, and hard problems to solve (plus life in general). If something comes up, please just reach out and explain, we can easily find a solution.
- Feel free to ask questions: If you want to clarify something while doing the test, please feel free to ask at any time. Email or text Julian on: 02102994705

Task

Design, train, and validate a model based on the [CIFAR10 dataset](https://www.cs.toronto.edu/~kriz/cifar10.html) (Canadian Institute For Advanced Research). You can view a getting started guide to downloading the dataset and training a simple CNN model using PyTorch here: https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

The model should be trained exclusively on the data available in the cifar-10 training set, and all validation and testing done with the cifar-10 test set.

You are expected to deliver:

LEVEL 1

Available via a public git repo (pick your flavour):

- A python script(s) which trains a model from scratch and saves it to a file
- A python script(s) which tests the generated model
- Accompanying documentation on how to run both scripts
- A document that describes the chosen model and structure, and the result of the model, i.e. several measures of accuracy and performance

LEVEL 2

- **Optimise the model to achieve high accuracy with a small model. The optimised model must contain less than 1 million parameters.**
- Expand on the details of the model performance with a confusion matrix and breakdown accuracy for each label/class
- Measure the inference time of the model in milliseconds per image per CPU
- Measure the number of parameters of the model

Ideally all of the measured details are computable dynamically when running the test script.

LEVEL 3

- Show your work: Document your thinking process, findings and experiments.
- Imagine if you had 3 months to work on this model: Describe areas of future research, techniques you might implement, experiments you might run, and specific areas where you believe the model could be improved.
- Reference any academic research that you have consulted when working on this test.

Requirements

- The test solution must be entirely your own work.
- The classifier must be capable of running on a standard, consumer laptop, and be CPU driven.
- The inference will also run on the laptop, so size (≤ 1 million parameters), and inference time of the model is important (≤ 20 ms inference time on a single CPU thread).
- You may use any preferred framework, e.g. PyTorch, TensorFlow, Keras, MXNet, Caffe, Theano, etc.
- Use of a pre-trained model or transfer learning is not allowed for this task; the model must be entirely trained by you.
- You may base the model architecture of your solution on an open-source model, but please reference which model architecture you are using, and document any changes you make on top of the open-source model.
- You may use ChatGPT to help you with this project, but we would suggest that you do a traditional literature review before consulting ChatGPT, as, for this particular problem, it will limit your thinking and result in a subpar solution.
- If you use ChatGPT, please include a transcript of your conversation with the Large Language Model as a text file with your test submission. We want to see how you prompt the AI and how you use it to expand your thinking.
- Structure your solution in a manner that demonstrates how you would approach a problem in a professional context. Show us how you structure your project, code, and other resources.

Evaluation criteria

We will evaluate your solution against the following criteria with roughly this order of importance:

1. Model has $< 1\text{M}$ parameters
2. Model accuracy
3. Model performance analysis
4. Documentation quality
5. Research quality and future work ideas
6. Code quality