TFMG Onboarding Assignment

Introduction

Welcome to our team! Our goal is to create templates and sample implementations of recently published machine learning methods using the TensorFlow platform.

Table of contents**:**

Salutation

This is the onboarding activity for the TensorFlow Model Garden (TFMG) VIP team! This document outlines an activity that is intended to take you about 4 weeks to complete (though feel free to finish early, depending on how much background knowledge you have).

Terms and conditions

To succeed on the team (*read as: get a good grade this semester*), you need to complete this assignment (or at least give it a rum go – we expect more from juniors than fresh-persons).

There is an optional component that we will use for a friendly competition.

Try your best. The older team members are here to help.

FAQ

Q: What do I do if I get stuck?

A: Post in the #tf-errors channel in the TFMG Slack.

Q: I’m lost and afraid. What should I do?

A: Post in the #tf-errors channel.

Q: It’s already week 3 of the semester and I have not started the assignment. What should I do?

A: Don’t do that. Start it now, make time for steady progress, and you’ll be able to complete it without trouble.

Q: To whom should I reach out for help?

A: Well, you can head to the #tf-errors channel.

Main point of contact

Mateusz Romaniuk

Recourses to study

Machine learning

From scratch

We are aware that people with different experience join our team. If you have no experience with machine learning,Andrew Ng offers a free online course, in the form of many ~10-minute videos available on YouTube. This course starts from the basics and covers many kinds of statistics and learning. The videos can be helpful to clarify a particular concept. Other ways to access Andrew Ng’s deep learning materials here:

- https://www.coursera.org/specializations/deep-learning
- https://youtu.be/OGxgnH8y2NM
- https://www.deeplearning.ai/
- Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

Deep learning

MIT intro to deep learning course: We recommend watching the first four lectures. You should also complete the first two labs. This course is taught using TensorFlow, and the content will help you understand our onboarding task.

- MIT intro to deep learning course
- Lab material:

is another great book to learn machine learning with TensorFlow. Section two of the book is particularly important.

Python

If you don't have any experience with Python, check out these introductory courses.

- https://exercism.io/tracks/python
- https://software-carpentry.org/lessons/

Software engineering

Coding style

- Review the Google Python Style Guide.
- We use tools to automatically enforce some of this style. Take a few minutes and read about linting (pylint) and unit testing (pytest).

Version control

On this team, we use git and GitHub to manage our software. The folks at Software Carpentry have a ~3-hour git tutorial here. It includes exercises. If you have not used git before, take time to complete the exercises.

- For deeper study, there's a git book: https://git-scm.com/book/en/v2

Testing

- All of our code is tested. You will need to write tests, too. Read the chapters “Testing Overview” and “Unit Testing” from the book “Software engineering at Google”. The book is available as an e-copy through the Purdue library.

Using Stack Overflow well

- If you are troubleshooting a program, you should *not* just aimlessly search the web hoping for a magic solution. Prof. Davis has some advice on using Stack Overflow effectively. His notes apply extra for specialized code like what we are writing.

- Testing deep learning code is a little different from testing “traditional” software. This blog post has some guidelines: Unit Testing in Tensorflow 2.0.

Getting started

Make a post in #tf-errors that says “**I have reviewed the Onboarding Assignment and am starting the activity!**” (You should do this in the first week of classes).

Activities

For this assigment you use Google Colab notebook. This is a free Jupiter notebook created by Google. It is optimized for TensorFlow.

Save a copy of this Colab template and put your solutions in the copy.

The rest of this document has links to the Colab template to help you see which parts to change in your copy.

ResNet Residual Block

1. Build a data pipeline to process images from the MNIST dataset using the tensorflow_datasets api (i.e. using a tf.data.Dataset). The pipeline should:
    a. Cast the image to a float32 tensor
    b. Scale all values in the image between 0 and 1. Remember that MNIST handwritten digit dataset are all black and white images. So the single value represents the pixel intensity.
    c. Generate a random number in the closed interval [26, 28]
    d. Apply a random_crop to the image to use a size (randint, randint), where randint is the random number generated above
    e. Resize the image to (28, 28, 1) after cropping
    f. Use tf.print to print the random number that is generated if the disp flag is set

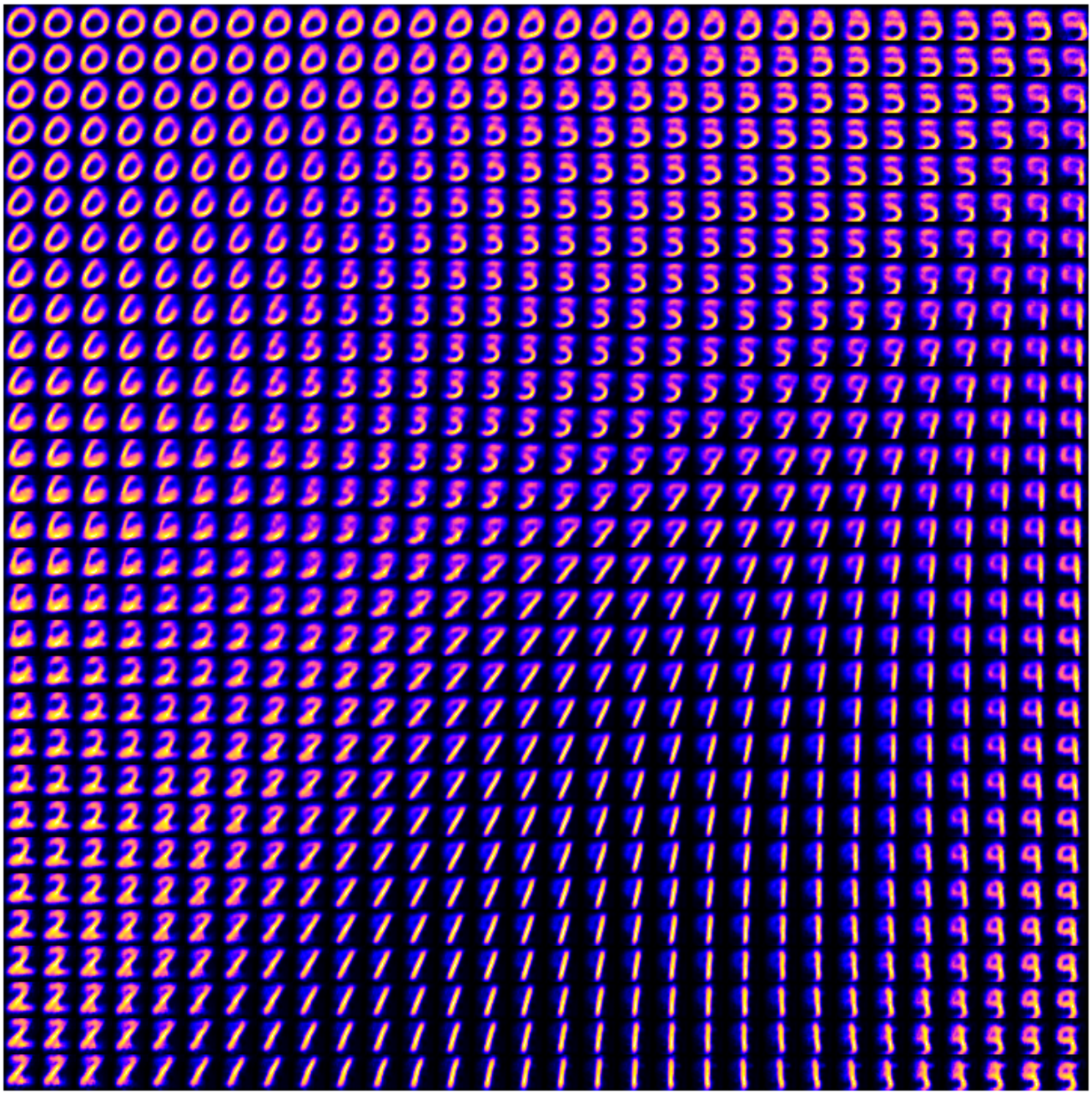
g. Throw away the label attribute and return preprocessed image as the label
    i. This model is unsupervised, so the input image is used as the label
    ii. So instead of a vector <image, label> to train with a vector <image, image>

Template in Cell 2

1. Read the ResNet Paper and implement the core residual learning building block (the non-bottleneck building block)Template in Cell 3
2. Test the model, we are providing the testing script:

Template in Cell 4

Example Output: (the output should look something like this, not the exact same, it is very unlikely that both images are exactly the same)



Debugging Loss Functions

1. **This step is optional**, but it will be extra-credit for the competition (see step 7).

Look at the Deep learning MIT lecture 4, and Use the testing script below to build the kld divergence loss. If the loss value hits NAN or INF, provide your best explanation for

why? If it does not, same thing, why? Because of this, the model will not train. Fix the issue and write down what you did to fix the loss function.

Use the display function to show the output of the VAE (variational auto-encoder) for both your own loss function definition and for the provided/incorrect loss function definition (see loss equations below).

hints :

$$loss = ||x - \hat{x}||^2 + (regualrization\ term)$$

$$(regualrization\ term) = D(p(z|x)\ ||\ p(z))$$

$$D(p(z|x)\ ||\ p(z)) = -\frac{1}{2}\sum_{j=0}^{k-1}(\sigma_j^2 + \mu_j^2 - 1 - log(\sigma_j))$$

- Excuse spelling errors:
  - @MIT Generative Models lecture 4
- There is a known error in this loss function, and the model will not train effectively with this regularization term.
  - Do not find a new one as this is a part of the exercise, modify the regularization term, and indicate above the loss function what you tried in order to get it to work, and why?
  - I should make clear, it is not a programming(compiling) error, the loss should compile, but it is a mathematical error.

    - If done correctly the output should look like the one above, but should not be exactly the same, it would be pretty sus if it looked that same.
- Write you answer either as a doc string in the loss function

Provided Loss function: Template in Cell 5

Your Loss function:

Template in Cell 6

Rock-Paper-Scissors CNN

1. Copy the pipeline from problem 1 (Cell 2) into the correct location in Cell 6. Add the following preprocessing steps to the pipeline:
    a. random flip
    b. random hue, max delta = 0.3
    c. random brightness, max delta = 0.1
    d. random_contrast, min = 0.9, max = 1.1

We will use this pipeline to load the rock_paper_scissors dataset from the tfds API.

Template in Cell 7

1. Create a convolutional neural network using the Sequential model class to classify hand pose images as rock, paper, or scissors. Use a softmax Activation layer as the last layer of the model in order to convert the logits outputted by the CNN into a prediction.

You will see that the term loss_fn is Not implemented, it is your job to determine the loss function to use, and in order to gauge your thought process, please include an explanation of why you choose the given optimization functionLet’s have a friendly competition. The aim of the competition is to use the least number of parameters in order to achieve at least 90% accuracy on the validation set. The top three competitors will win $30, $20 and $10, respectively.. Yay!

Template in Cell 8

1. Build a simple training and validation loop (i.e. we are not using tf.keras.Model.fit to train the model anymore). tf.GradientTape is indispensable for this problem. To help you understand the training loop and understand what happens behind the scenes in tf.keras.Model.fit, consider the following questions:
2. How do you interact with a tf.data.Dataset object?
3. Is the tf.data.Dataset object an iterator?
4. Does tf.data.Dataset require you to store the whole dataset in memory?

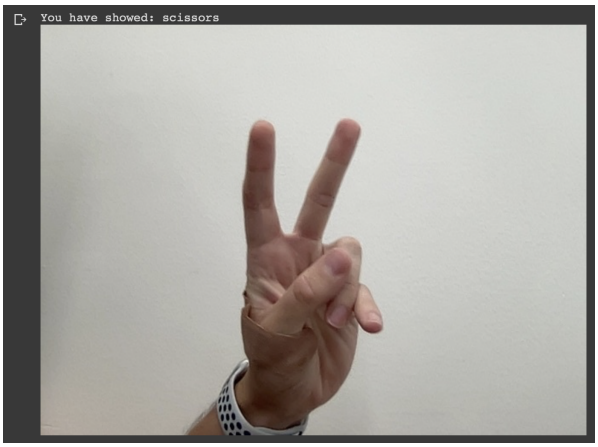
Template in Cell 9 To test the model that you trained with your custom training loop, you can run the code in Cell 9.

Template in Cell 10

Let’s have some fun

As we know, this club focuses on computer vision applications. You've put a lot of work into it, so let's test our application in the real world. The code in cell 11 will allow you to take a picture of your hand thanks to the camera on your laptop. Remember to make the background in the photo white.

See examples below:



We trained the model using 3D generated images and tested it using the same database. Now we are using the model in a real world scenario. I invite you to share your thoughts on the pros and cons of this approach.

Template in Cell 11

Analyze research papers

A big part of this team is analyzing research papers and implementing solutions. We are currently working on two projects: YOLO family and Mesh RCNN. Your next task will be to read the papers below and answer the questions in cell 12.

YOLOv3 and YOLOX

YOLO stands for "you only look once." There are many variations of YOLO. Our goal now is to make our implementation of YOLOX tensorflow work with similar results as shown in the article. Your task will be to read the YOLOv3 article and the YOLOX article and list 3 differences between the two versions.

Mesh R-CNN

Mesh Predictor is the core of the Mesh R-CNN. It is a long, complicated and interesting algorithm. Try to explain and show that you understand what Voxel Branch is and its role in Mesh R-CNN.

Links to papers:

- YOLOX - https://arxiv.org/abs/2107.08430
- YOLOv3 - https://arxiv.org/abs/1804.02767
- Mech R-CNN - https://arxiv.org/abs/1906.02739

Template in Cell 12

TFMG Tech Report

1. Read the TFMG Tech Report – An Experience Report on Machine Learning Reproducibility: Guidance for Practitioners and TensorFlow Model Garden Contributors

Please answer the questions in Cell 10.

Template in Cell 13

How to Submit

Please download your Colab as a Jupyter Notebook (File > Download > Download .ipynb). Save the file as “TFMG Onboarding {INSERT YOUR NAME}”.

Send an email with your final submission to Mateusz Romaniuk (mromaniu@purdue.edu) with the following subject line: TFMG Fall 2022 Onboarding - {INSERT YOUR NAME}.

This assignment is due on September 29th at 11:59pm.