

LSM4213

Practical 2

Digit classification website

Code is used and adapted with permission mainly from
Deploying a Simple Machine Learning Model into a WebApp using TensorFlow.js by Carlos Aguayo
<https://towardsdatascience.com/deploying-a-simple-machine-learning-model-into-a-webapp-using-tensorflow-js-3609c297fb04>

This material is **not required** for the final exam.

Overview

The Practical 2 folder contains:

- [Digit_classification_website.ipynb](#) (Colab notebook)
- [digit_classification_website](#) (folder with HTML code)
- [Digit_classification_website_with_data_augmentation.ipynb](#) (Colab notebook)
- [digit_classification_website-with_data_augmentation](#) (folder with HTML code)

In this practical we will:

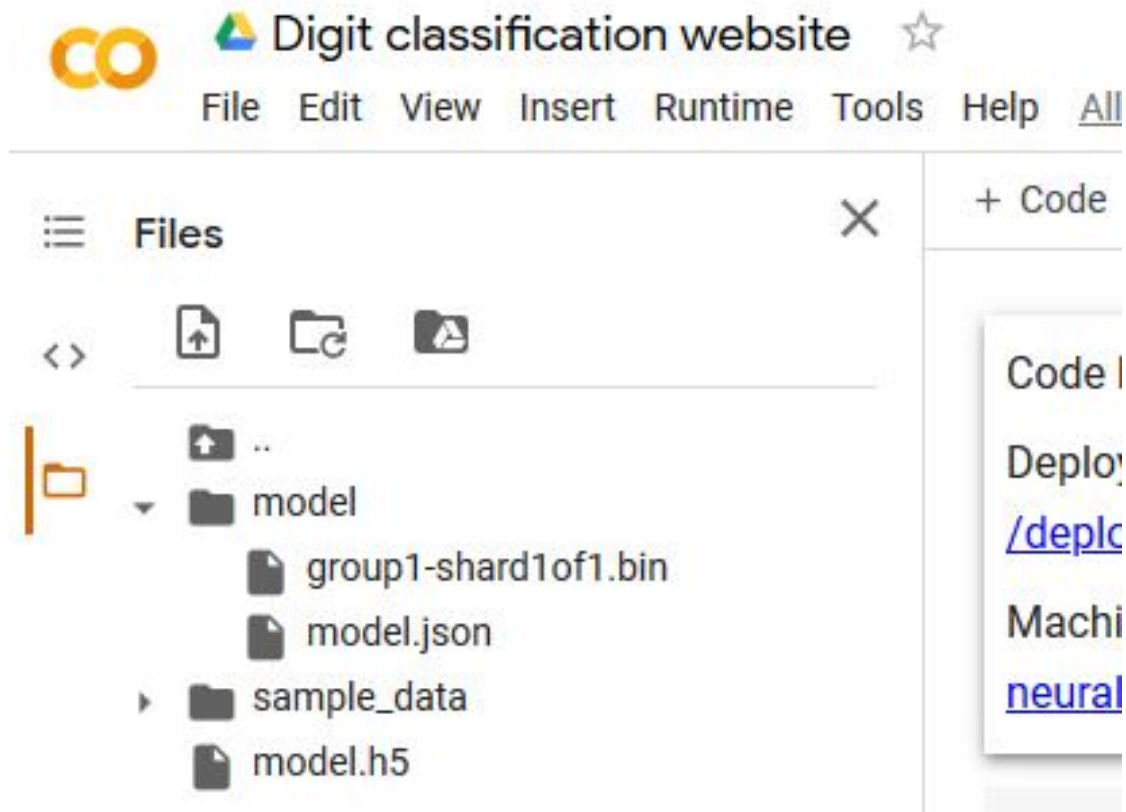
1. Train a neural network model to recognize handwritten digits. This code is written in Python and uses the Tensorflow package. We will run the code online using Google Colab.
2. Add the trained model from Step 1 to the code for a web app that creates a web interface for a handwritten digit to be inputted, and interfaces with the model to recognize the handwritten digit. The main code for the web app is written in HTML.
3. Deploy the web app using GitHub, which includes a web hosting service.
4. Repeat steps 1-3 with data augmentation during neural network training to obtain a web app with better digit recognition performance.

1. Train the neural network model (1)

- Register for a Google account at <https://www.google.com/account/about/>. Registration is **free** and will give you access to Google Drive and Google Colab, which we will use. You already have a Google account if you have Gmail.
- Sign-in to your Google account.
- Go to Google Drive <https://www.google.com/drive/>.
- Upload the [Digit_classification_website.ipynb](#) Colab notebook to your Google Drive.
- The [Digit_classification_website.ipynb](#) notebook is written mainly in Python and makes use of TensorFlow, a Python package that makes it easy to build and train artificial neural networks.
- Open [Digit_classification_website.ipynb](#) in Colab, by right clicking or double clicking on its filename in Google Drive, then choosing "Open with" followed by "Google Colaboratory".
- Go to the [Digit_classification_website.ipynb](#) notebook that has been opened in Colab. From the "Runtime" menu, go to "Change Runtime type" and choose the "GPU" hardware accelerator option. The GPU is not always available (since you are sharing Google's computers with many other Colab users). The code will run without a GPU, but it will be very much faster if one is available.

1. Train the neural network model (2)

- Run the code by going to the "Runtime" menu, and choosing "Run all".
- Scroll down as the code runs to see printouts indicating the progress of the program.
- The outputs from running the [Digit_classification_website.ipynb](#) notebook are 2 files describing the trained neural network model: [group1-shard1of1.bin](#) and [model.json](#).
- Download both files by clicking on the folder icon in the left margin, then clicking on the "model" folder, which will show that it contains (see screenshot on right) [group1-shard1of1.bin](#) and [model.json](#). For each of the files, right click on the filename, and choose the download option.



2. Package the model with the web app code

- The code for the web app is contained in the `digit_classification_website` folder which contains the following:
 - `tjfs.html` (main code for the web app, written in HTML)
 - `spinner.gif` (gif animation used by tjfs)
 - `README.md` (file explaining folder contents, does not contain code)
 - `model` (empty folder)
- The main code for the web app is written in HTML. It also uses code written in JavaScript and uses the TensorFlow.js JavaScript library, which allows the web browser to interface with the `group1-shard1of1.bin` and `model.json` files describing the neural network model that we trained.
- Add both `group1-shard1of1.bin` and `model.json` files that were downloaded from Colab to the initially empty `model` folder.
- The `digit_classification_website` folder now has the complete code for the web app, and is ready to be uploaded to and deployed with GitHub

3. Deploy the web app using GitHub (1)

- We follow the instructions at <https://pages.github.com/> there for setting a user site.
- Register for a **free** GitHub account at <https://github.com/>.
- Sign-in to your GitHub account.
- Create a new repository *username.github.io*, where *username* is your username on GitHub. Choose the options for a public repository, and to initialize the repository with a README.
- You will have now set up
 - GitHub repository: <https://github.com/username/username.github.io>
 - GitHub user page: <https://username.github.io>
- Minor diversion I: To make your GitHub user page look prettier, go to the GitHub repository <https://github.com/username/username.github.io>, then go to "Settings", scroll down to "GitHub Pages", click on "Choose a Theme". This will bring you to a page with various page designs or themes. Click on the theme you would like to have and click on "Select a Theme". This will bring you to a page with an edited README. Scroll to the bottom and click on "Commit Changes". Go to your GitHub user page <https://username.github.io> to see its new design. We will update the README text later.

3. Deploy the web app using GitHub (2)

- Install the GitHub Desktop app from <https://desktop.github.com/>.
- Open the GitHub Desktop app, choose "Clone a repository", then choose *username/username.github.io*, which is the repository you have just created, to clone. This will create a local copy of the repository on your computer.
- Open the folder of the local copy of the repository in File Explorer (Windows) or Finder (Mac) by choosing "Show in Explorer" (Windows) or using the "Ctrl+Shift+F" (Windows) or "Cmd+Shift+F" (Mac) keyboard shortcut in GitHub Desktop.
- Place a copy of the [digit_classification_website](#) folder (that you completed by adding files to) in the folder of the local copy of the repository.
- The updates you have just made to the local copy of the repository will be reflected in GitHub Desktop under "Changes". Fill in the "Summary" field with a brief description on the changes, eg. "Add digit classification website code". Click on "Commit to master".
- After the changes are committed to the local repository, the "Push origin" option will appear, which you should choose to push the changes to the online GitHub repository at <https://github.com/username/username.github.io>.

3. Deploy the web app using GitHub (3)

- The web app is now deployed at https://username.github.io/digit_classification_website/tfjs.html
- You will find that the app's performance is not terrible, but leaves much to be desired. We will improve performance by training the neural network with distortion added to the data, so it learns to recognize a wider range of handwritten digits. Training with distortion added to the data is a form of **data augmentation**.
- The next steps are similar to those just completed, so we state them briefly.
- Run the [Digit_classification_website_with_data_augmentation.ipynb](#) notebook in Google Colab. Download the [group1-shard1of1.bin](#) and [model.json](#) files.
- Go to the [digit_classification_website_with_data_augmentation](#) folder, and add the [group1-shard1of1.bin](#) and [model.json](#) files to the [model](#) folder.
- Deploy the [digit_classification_website_with_data_augmentation](#) folder by adding it to the <https://github.com/username/username.github.io> repository.
- The web app will be deployed at https://username.github.io/digit_classification_website_with_data_augmentation/tfjs.html

3. Deploy the web app using GitHub (4)

- **Minor diversion 2: Let us complete our GitHub user page. Go to the GitHub repository <https://github.com/username/username.github.io>. Scroll down to the README and click on the edit symbol. Delete the text in the README and replace it with the following text.**

Welcome

Systems neuroscience and artificial intelligence

- [Digit classification website](https://andrew-tan-yy.github.io/digit_classification_website/tfjs.html)

- [Digit classification website with data augmentation](https://andrew-tan-yy.github.io/digit_classification_website_with_data_augmentation/tfjs.html)

Code is used and adapted with permission mainly from

- [Deploying a Simple Machine Learning Model into a WebApp using TensorFlow.js](<https://towardsdatascience.com/deploying-a-simple-machine-learning-model-into-a-webapp-using-tensorflow-js-3609c297fb04>) by Carlos Aguayo.

- **Scroll down and click on "Commit changes".**
- **We should now have**
 - **GitHub user page** <https://username.github.io> (with links to the following apps)
 - **Digit classification website** https://username.github.io/digit_classification_website/tfjs.html
 - **Digit classification website with data augmentation**
https://username.github.io/digit_classification_website_with_data_augmentation/tfjs.html

References

- **Code is used and adapted with permission mainly from Deploying a Simple Machine Learning Model into a WebApp using TensorFlow.js by Carlos Aguayo**
<https://towardsdatascience.com/deploying-a-simple-machine-learning-model-into-a-webapp-using-tensorflow-js-3609c297fb04>
- **Code is also used and adapted from Handwritten Digit Recognition using Convolutional Neural Networks in Python with Keras by Jason Brownlee**
<https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/>
- **Data augmentation code is based on the Step 7: Improve model accuracy with data augmentation by The TensorFlow Authors**
<https://codelabs.developers.google.com/codelabs/digit-classifier-tflite/#6>
- **GitHub user page set up is based on Train a model in tf.keras with Colab, and run it in the browser with TensorFlow.js by Zaid Alyafeai**
<https://medium.com/tensorflow/train-on-google-colab-and-run-on-the-browser-a-case-study-8a45f9b1474e>

Further explorations

- **To deploy a similar app on an Android smartphone, follow the TensorFlow Lite tutorial - Build a handwritten digit classifier app with TensorFlow Lite by The TensorFlow Authors**
<https://codelabs.developers.google.com/codelabs/digit-classifier-tflite/#0>
- **The TensorFlow Lite example apps (including a digit classifier) for Android and iOS Smartphones are available at**
<https://www.tensorflow.org/lite/examples>