Homework and Projects:

1. **Homework one. Due: 11:59pm (Central Time) on Friday 5/30/2025 in Canvas.**

Check out the Jupyter notebook for Chapter 2 at <https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/chapter02_mathematical-building-blocks.ipynb>. Then:

1) For the neural network in the Jupiter notebook for the MNIST task (which has 2 layers, where the first layer has 512 neurons and the second layer has 10 neurons), adjust the size of the first layer (namely, the number of neurons in the first layer) as: 16, 32, 64, 128, 256, 512. For each size, train the neural network and record its test accuracy (namely, accuracy on the test set). Draw a figure, where the x-axis is the size of the first layer, and the y-axis is the test accuracy. Submit your figure.

2) For the neural network in the Jupiter notebook for the MNIST task (which has 2 layers, where the first layer has 512 neurons and the second layer has 10 neurons), adjust the number of layers to be: 2, 3, 4, 5. (The last layer has to have size 10 since we have 10 classes. But you can decide on the sizes of the other layers.) For each case, train the neural network and record its test accuracy. Draw a figure, where the x-axis is the number of layers, and the y-axis is the test accuracy. Submit your figure, and specify the sizes of your layers for each case.

1. **Homework two. Due: 11:59pm (Central Time) on Tuesday 6/3/2025 in Canvas.**

1) Check out the Jupyter notebook for Chapter 3 at <https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/chapter03_introduction-to-keras-and-tf.ipynb>. Then, use the "GraidentTape API" to find the derivative of the function f(x) = sin(x) for x = 0, 0.1, 0.2 and 0.3. Submit your Jupyter notebook that shows both the code and the result you got.

2) Check out the Jupyter notebook for Chapter 4 at <https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/chapter04_getting-started-with-neural-networks.ipynb>. Then, for the task "Classifying movie reviews: A binary classification example", tune the hyper-parameters of the model (such as changing the number of layers, changing the sizes of layers, changing the optimizer, changing the learning rate, etc.), and see if you can improve the model's performance. Submit a Jupyter notebook where you clearly show the code with the best hyper-parameters that you have found, along with its performance on training, validation and test sets.

1. **Homework three. Due: 11:59pm (Central Time) on Friday 6/6/2025 in Canvas. (**Submit all answers below in a single Jupyter notebook file.)

1) Check out the Jupyter notebook for Chapter 5 at <https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/chapter05_fundamentals-of-ml.ipynb> . Answer the question: what regularization techniques were mentioned in that Jupyter notebook?

2) The MNIST dataset has 60,000 training images and 10,000 test images. Each image is a 28x28 array, where each array element is between 0 and 255. The images have 10 labels: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

 We now create a new dataset of about 30,000 training images, about 5,000 test images and 5 labels (which are 0, 1, 2, 3, 4) as follows. First, randomly pair up the training images of label 0 with the training images of label 1, to get about 6,000 such pairs. (The actual number may be a little less than 6,000, because the number of images for some label may be less than -- although close to -- 6,000. In this case, just get as many pairs as you can.) Then, for each pair (A,B) where A is an image of label 0 and B is an image of label 1, we create a new image of size 28x28, where each element's value is the "average" of the two corresponding pixel values in A and B. (So the new image is a "mixture" of the two original images.) This way we create about 6,000 new "mixture" images for training. In a similar way, we create about 1,000 new "mixture" images for testing. We give all these about 6,000+1,000=7,000 new "mixture" images the new label 0. Then, in the same way, we create about 6,000 new training images and about 1,000 new test images by mixing the original images of label "2" and "3", and give them the new label 1; create about 6,000 new training images and about 1,000 new test images by mixing the original images of label "4" and "5", and give them the new label 2; create about 6,000 new training images and about 1,000 new test images by mixing the original images of label "6" and "7", and give them the new label 3; create about 6,000 new training images and about 1,000 new test images by mixing the original images of label "8" and "9", and give them the new label 4.

    Your task: submit your code that creates the above new dataset; then for each of the 5 new labels, randomly select 2 images of that label from your new dataset, and display them in your submitted Jupyter notebook.

3) Design a neural network model to classify the 5 classes in the new dataset, and optimize it by tuning its hyper-parameters and trying our learned regualization techniques (such as L1 regularization, L2 regularization, dropout).

    Your task: For your final (namely, optimized) neural network model, submit its code, and show the model's performance (including loss value and accuracy) for training, validation and testing. (For training performance and validation performance, illustrate them using figures, where the x-axis is the number of training epochs, and the y-axis is the loss or accuracy. For testing performance, just show the values of loss and accuracy.) Also, answer the questions: in this process of optimizing your model, did you observe underfitting? Did you observe overfitting? Did you try the reguliazation techniques? Did they help?

1. **Homework four. Due: 11:59pm (Central Time) on Friday 6/13/2025 in Canvas.** (Submit all answers below in a single Jupyter notebook file.)

1) Check out the Jupyter notebook for Chapter 8 at <https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/chapter08_intro-to-dl-for-computer-vision.ipynb>. It uses the convolutional base of VGG16 for an image classification task, and also tries fine tuning. Your task: use another existing trained neural network (which is different from VGG16, such as ResNet) for the same task, and also try fine tuning. Submit your complete code, draw figures on the training/validation performance, and show the testing performance. (Note: the Dogs-vs-Cats dataset originally used in the textbook's Jupyter notebook is no longer available. But there are still many other datasets available on image classification. Students are encouraged to search for them and use one of them. But note that the MNIST dataset should not be used. The dataset you choose should be more difficult than the MNIST dataset, and more like the original dogs-vs-cats dataset. The content of the dataset can be other objects, such as cars, desks, houses, etc. By the way, here is actually a dogs-vs-cats dataset: https://www.kaggle.com/competitions/dogs-vs-cats/data. Accessing this dataset may require logging in and joining the competition beforehand. )

1. **Homework five. Due: 11:59pm (Central Time) on Tuesday 6/17/2025 in Canvas.** (Submit all answers below in a single Jupyter notebook file.)

Check out the Jupyter notebooks for Chapter 9 at <https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/chapter09_part01_image-segmentation.ipynb>, <https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/chapter09_part02_modern-convnet-architecture-patterns.ipynb>, and <https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/chapter09_part03_interpreting-what-convnets-learn.ipynb>. Then:

1) Try the code on image segmentation. Show the segmentation result on a randomly selected image.

2) Try the code on "Visualizing intermediate activations". Show the visualization result on a randomly selected image from the cats-and-dogs dataset.

3) Try the code on "Visualizing convnet filters". Show the visualization result for a randomly selected filter.

4) Try the code on "Visualizing heatmaps of class activation". Show the visualization result on a randomly selected image from the cats-and-dogs dataset.

1. **Homework six. Due: 11:59pm (Central Time) on Friday 6/20/2025 in Canvas.** (Submit all answers below in a single Jupyter notebook file.)

Check out the Jupyter notebook for Chapter 10 at <https://github.com/fchollet/deep-learning-with-python-notebooks/blob/master/chapter10_dl-for-timeseries.ipynb>. It has tried 7 methods for the temperature prediction problem: Try 1 (A common-sense, non-machine learning baseline method), Try 2 (A fully connected neural network), Try 3 (1-d convolutional neural networks), Try 4 (Recurrent neural network), Try 5 (LSTM with recurrent dropout), Try 6 (stacking RNN layers), Try 7 (Bidirectional RNN).

Your task: use the above 7 methods to predict the temperature in 48 hours (instead of 24 hours). In the Jupyter notebook, include your code as well as the performance of the 7 methods.

1. [Project 1](https://drive.google.com/file/d/1oxBb2Gki4RB9WKmK39-LiqK034TBp98P/view?usp=share_link). Due: 11:59pm (Central Time) on Friday 6/20/2025 by email. (Please email your trained DNNs and Jupyter notebook to our project email address [moore.research.education@gmail.com](mailto:moore.research.education@gmail.com). The title of the email should be "CSCE 636-700 Summer 2025 Project 1 + {YOUR NAME} + {YOUR UIN}".)

We provide a training set here: "[input file for (n,k,m,P)](https://drive.google.com/file/d/1JRIhwhOV2B-XzDNLHbHtVUuN9j9Kic0-/view?usp=sharing)" and "[output file for m-heights](https://drive.google.com/file/d/1We21EnZVBRiIw7MJzZpf4tFdkqV_vRdq/view?usp=sharing)". (If you would like a larger training set, the LP-based algorithm described in the project file can be used to create more samples.) Both files in the training set contain a list saved using pickle.dump(), which you can open using pickle.load(). In the list of the "input file for (n,k,m,P)', each sample is a list  [n,k,m,P], where n=9 is an integer, k is an integer in {4, 5, 6}, m is an integer in {2, 3, ...,, n-k}, P is a 2-dimensional numpy array of k rows and n-k columns. In the list of the "output file for m-heights", each element is an m-height, which is a real number. For r=0,1,2..., the r-th number in the list of the "output file for m-heights" is the m-height of the r-th sample in the list of the "input file for (n,k,m,P)". (Note: the above dataset is posted on 6/16/2025. It has 32,087 samples.)

The file "input file for (n,k,m,P)" above is also an example of a test set. It has the same format as the test set that will be used to test your model. MAKE SURE that your submitted code works with the test set, and outputs a list of corresponding predicted m-heights that has EXACTLY THE SAME FORMAT as the list in the "output file for m-heights".

Note: To ensure correct grading of your submitted model, a Sample Test Set will be posted here on Saturday 6/21/2025. Please run your already-submitted model for this project (without ANY change to the model) on the posted Sample Test Set to get the list of corresponding predicted m-heights, and submit the list as a single file in Canvas. (The posted Sample Test Set is a part of the test set used to evaluate your models. Your submitted results --- namely, the m-heights predicted by your model for this project -- will be compared to our testing results on your model for verification.)

What to submit: Save the list of m-heights (which are your model's predicted m-heights for the Sample Test Set) as a file using pickle.dump(). The name of the file should be "CSCE 636-700 Summer 2025 Project 1 Test Results + {YOUR NAME} + {YOUR UIN}".  Submit the above file in Canvas by Monday 6/23/2025. (Note: without this file, the project will not be graded.)

1. Project 2: Same as [Project 1](https://drive.google.com/file/d/1oxBb2Gki4RB9WKmK39-LiqK034TBp98P/view?usp=share_link). Students are encouraged to explore new methods to further improve the results. Due: 11:59pm (Central Time) on Tuesday 6/24/2025 by email. (Please email your train DNNs and Jupyter notebook to our project email address [moore.research.education@gmail.com](mailto:moore.research.education@gmail.com). The title of the email should be "CSCE 636-700 Summer 2025 Project 2 + {YOUR NAME} + {YOUR UIN}".)

 Note: To ensure correct grading of your submitted model, a new Sample Test Set will be posted here on Wednesday 6/25/2025. Please use your submitted model for this project on the new Sample Test Set (without ANY change to the model) to get the corresponding predicted m-heights, and submit them as a single file in Canvas. (The new Sample Test Set is a part of the test set used to evaluate your models. Your submitted results --- namely, the m-heights predicted by your model for this project -- will be compared to our testing results on your model for verification.)

What to submit: Save the list of m-heights (which are your model's predicted m-heights for the new Sample Test Set) as a file using pickle.dump(). The name of the file should be "CSCE 636-700 Summer 2025 Project 2 Test Results + {YOUR NAME} + {YOUR UIN}".  Submit the above file in Canvas by Friday 6/27/2025. (Note: without this file, the project will not be graded.)