Jupyter Notebook Written Assessment 2023-24

Create a Fully Convolutional PyTorch Model for Protein Secondary Structure Prediction

Getting Started

The coursework is set as a Kaggle Competition. Kaggle is one of the leading data science competition sites and it is worthwhile getting familiar with it.

Our Kaggle competition will be private and can only be viewed by course lecturers and your classmates - **so we will use the University of Glasgow email addresses so that I can identify you for assigning marks.**

So you will first need to register for this competition using your UofG email address using the Kaggle registration link from the Kaggle website: https://www.kaggle.com/

If you are unfamiliar with Kaggle then you should probably get started by doing the Titanic Tutorial:

https://www.kaggle.com/alexisbcook/titanic-tutorial

Once you are logged into Kaggle (with your UofG email address) then you can join the private "Deep Learning for MSc 2022-23" competition using the link: https://www.kaggle.com/t/59d729144530466b9513a7528ea8c462

(Do not share this link with others outside this class.)

You will need to choose a Team Name to take part in the Kaggle competition. Often Kaggle competitions are done in Teams but this will be **individual coursework** so you will only have yourself in your Team !

Overall Goal

Your overall goal is to write a **Fully Convolutional PyTorch model** that can input protein sequence data (often called the Protein Primary Structure, or additionally using PSSM Profiles to predict the protein secondary structure (H = Helix, E = Extended Sheet, C = Coil symbols). The PDB Database contains the protein structures of over 200,000 proteins. Each has a unique PDB_ID code such as 1A0S (the first one in the training data) which is the structure shown above (sucrose-specific porin of salmonella) which is used to transfer sucrose across the cell membrane of salmonella bacteria which causes food poisoning. The protein has a 3D Structure which shows that most of this protein is extended beta sheet (flat arrows) and coil (random lines).

The Data Tab on Kaggle will allow you to browse the available data used for training. You should use this Data Tab to browse through the data so you understand what it is like. You will find a seqs_train.csv file which is a CSV file that gives the PDB_ID (unique identifier) and the SEQUENCE of each protein. You will also find a train.zip file which contains a large collection of

<PDB_ID>_train.csv files containing residue number, amino acid and PSSM profiles for each residue in that particular protein. The labels_train.csv file contains the secondary structure labels for the different training proteins (given as H = Helix, E = Extended Sheet, C = Coil symbols). The seqs_test.csv and test.zip contains similar data for the test sequences for which you need to predict the secondary structure.

IN ADDITION - you will also need to submit your Jupyter Notebook that produces these outputs via the Moodle web page.

How should I develop my code (and where do I get the GPU/TPU power from ?)

So far you have mostly used the Google Colab Notebook for Labs but this would involve transferring fairly large data files and also you may find that you run out of GPU time on Google Colab (particularly if you are doing other courseworks as well using it).

For this coursework we will use Kaggle Notebooks ! This not only gets you familiar with another Jupyter Notebook system – but it means you can directly access the data files for this competition without transferring them. It also allows you to keep your files in order since it has a Versioning system built in (it is important that you submit the same notebook as you used to generate predictions for your best attempt at Kaggle and this can be difficult unless you keep track of what versions of the notebook you used for different submissions).

If you go onto the “Code Tab” then do “New Notebook” – this will create a competition notebook where you have direct access to the competition data. Please see the Kaggle Notebook documentation for more information about their notebook system:
https://www.kaggle.com/docs/notebooks

You use GPUs in the same manner as you would with normal PyTorch code (you need to turn them on though in a similar manner to Google Colab). If you want to experiment with using TPUs then a good getting started course is:
https://www.kaggle.com/competitions/tpu-getting-started

With a more specific tutorial on using TPUs with PyTorch given here:
https://www.kaggle.com/code/tanlikesmath/the-ultimate-pytorch-tpu-tutorial-jigsaw-xlm-r/notebook

Steps to Success !

This is very much a “Capstone” project where you will bring together a lot of the material you have understood from different labs and lectures in the first 5 weeks.

After getting familiar with the Kaggle infrastructure – the first stage of Notebook development would be to write a custom data loader for the PDB ID csv data and PSSM data in a similar manner to Lab 5.

You first want to make sure you understand the key ideas in Lecture 4 – Machine Learning Workflow. A lot of these concepts will be essential in terms of splitting the data into suitable training and validation datasets (you should be evaluating your performance using the validation dataset and not relying on resubmission to Kaggle to assess your performance – you are only allowed 5 submissions per day – and doing more will result in overfitting to the test set).

Then the first stage would be writing a custom data loader for this particular data similar to Lab 5. Then possibly synthesize the material in Lab 3 for ConvNets – but modifying this to work with the new type of data and turning the model into a fully convolutional network to predict many residue labels for a protein at the same time. At this point you may want to include code from Lab 4 for Ray Tune or Ax hyperparameter optimization.

You must:

1. **Develop a model in PyTorch ! (I shouldn't need to say this … but each year we get Keras and TensorFlow models submitted … usually just taken from GitHub!)**
2. **You need to design and implement a Fully Convolutional Model for this task which will take a tensor of inputs (either a complete sequence as a tensor or a complete PSSM sequence profile as a tensor) and then putting it through the model generates a complete tensor of output secondary structure labels. Please look at how Fully Convolutional Models are used to do segmentation of images into a number of labelled regions. You will be doing a similar thing but "segmenting" a sequence into a number of secondary structure labels.**
3. **You need to demonstrate carrying out some suitable hyperparameter optimization using Ray Tune or Ax as in Lab 4. Clearly you need to choose a sensible approach to hyperparameter optimization given your limited Kaggle GPU/TPU resources.**

You should:

4. **Have your notebook generate loss and accuracy curves to assess how your training is working and diagnosing any issues.**
5. **As a stretch challenge, try using Captum to understand the features your model is using to predict alpha helix, beta sheet and coil regions !**

Clearly you should also submit your predictions to Kaggle for each model and determine which of them might be doing best (usually this is done by creating and submitting a submission.csv file). This can be done directly from the "Output" directory for Kaggle Notebooks.

Submission

Please submit the results of your method (submission.csv) to the Kaggle site. This will be tested and the accuracy on the **unseen test set** will be given on the leader board. You can make up to 5 submissions per day to assess what might be the best approach. The final private leader board (only released once the competition ends) will show the score of your best submission **and this will constitute 50% of the marks** (this will be based on getting a score better than particular thresholds rather than a direct conversion of the accuracy score !)

IMPORTANT - also submit your final Jupyter Notebook for your results to Moodle (exported from Kaggle Notebooks system).

Your Jupyter Notebook will be marked on a number of aspects such as showing the key components mentioned above (use of training and validation data, plotting and interpreting loss curves, hyperparameter tuning, stretch challenge in terms of interpretation using Captum and discussion of both your models in terms of these aspects). Your Jupyter

Notebook file should be well annotated as a data science laboratory notebook – explaining what you are doing and why, interpretating your results and what they mean. **Your submitted notebook needs to have all the cells run so that they are all showing output to get you marks!** The submitted Notebook will constitute the other 50% of the coursework marks.

Again - your submitted Jupyter Notebook should have all output visible so it can be read as a data science notebook *without running it again.*

PLEASE USE THE **TEAMS JUPYTER WRITTEN COURSEWORK CHANNEL** TO CLARIFY ANY INFORMATION ABOUT THE COURSEWORK.