Society for Artificial Intelligence and Deep Learning

www.saidl.in

Summer 2023 Induction Assignment

Introduction

This is the official assignment for SAiDL inductions. Please join our Slack Workspace here and fill this form in order to register.

Deadline is **Thursday 20/04/2023 11:59 PM IST**.

A Note from Us

Through this assignment we aim to introduce you to key ideas in Machine Learning, Deep Learning and Artificial Intelligence. We value your time and have designed each question such that the process of solving it should add to your skill set while also giving us an idea of your abilities. We have also provided a collection of resources to help you learn the required tools and concepts along the way.

Please go through the instructions before attempting the questions. The questions might seem intimidating at first glance. Don't panic! They are meant to be challenging and will take some time and effort to complete. Try looking around to understand the concepts involved in the questions and try digging into these topics. It is expected that around 10% - 40% of time (depending on your prior familiarity) on each question will be spent on learning new concepts. The submissions will be judged primarily on the basis of the approach and the thought that went into attempting the answer, so be sure to document everything. Also make sure to submit your attempt irrespective of whether or not you have completed the assignment as a whole.

If you have any doubts, try to search for solutions on the internet first before asking doubts on the slack. We will also be assigning everyone mentors so keep them updated with your progress.

All the best!

Logistics

Submission Guidelines

The submission deadline is **Thursday 20/04/2023 11:59 PM IST**
Submission is to be made via: *TBA*

The submission should be primarily in the form of a Git repo titled *SAiDL-Summer-Assignment-2023*. The repo should be **kept private up until the submission deadline**. Make sure you make the repo public while submitting the assignment.

You are required to document your approach and results for each question you attempt and analysis wherever required, in detail. The documentation should be included in the github repository along with the code and can be made in Markdown, LaTeX etc… Make sure to make your final report for each task as detailed and clear as possible.

Any form of plagiarisation or collaboration with others is strictly prohibited and will be penalized.

You'll be allotted a mentor for the assignment and you are required to inform them whenever you complete a question (submit the google form only after you have completed the assignment or after the deadline). All of this will be done through our Slack Workspace so make sure you join it for receiving all the assignments and for asking queries that you might have. You can also contact those mentioned below for queries.

SAiDL holds the rights to the questions asked in the assignment, if you choose to use this outside the assignment, you need to cite/acknowledge SAiDL.

Contact Details

In case of any doubts, clarifications, or guidance, you can contact one of us. We request that you stick to slack as the medium of communication for all the questions that you have. However, you can reach out to us through other means in case we fail to respond on slack.

- Shreyas Bhat (Core ML) - 9082080984
- Aditya Agarwal (Core ML / RL) - 8879459970
- Shrey Pandit (NLP) - 9820320640
- S I Harini (NLP) - 7021247073
- Rishav Mukherji (CV) - 9920741703
- Ashmit Khandelwal (CV) - 7977519061
- Hardik (CV) - 9427925103

Resources

The following is a list of resources to tools and topics you will need to be familiar with in order to solve the assignment questions. This section is more a selection of resources for things you will need along the way. It is not mandatory to complete, you should treat it more as a roadmap of things to learn and as a reference to get back to.

Python

Python is the Lingua Franca of AI (at the moment). Check out this guide by Google to start or the tutorial series by Sentdex. These books (1, 2) and the official documentation are good references.

Coursera Course on Machine Learning

This course is the launchpad for most AI enthusiasts and gives you your first hands-on approach to Machine Learning along with the maths behind it. The course requires implementation of ML algorithms in Octave/MATLAB. However, it is recommended that you attempt the Python versions of these exercises which can be found here.

Linux Terminal

This is one of the fundamental requirements for any computer scientist. The following will help you get started (1, 2).

Numpy

Crudely speaking this is MATLAB for Python. We suggest you do this after the Andrew NG course. You can go to its official tutorial or learn it with hands-on deep learning experience via deeplearning.ai's first course.

Pandas

This is one of the most crucial and powerful libraries in data science. To begin with, you end up learning how to read and write CSV and JSON files as well as how to manipulate Data Frame rows, columns, and contents. Again Sentdex to the rescue.

Matplotlib

Learn how to plot basic graphs. The pyplot submodule should be enough for the beginning. Sentdex is your saviour again. The docs are useful too!

PyTorch

PyTorch is a Machine Learning framework built to provide a toolkit for writing Deep Learning research and application code. We would suggest that you get hands-on experience with PyTorch by following the official tutorials.

Git

An integral part of most software projects, Git is a tool for version control and managing changes to code. This is a cool guide for a quick overview, for a more detailed intro check out this course. Make sure to sign up for the Student Developer Pack on GitHub.

Conda

Managing different packages when you are working on multiple projects can be a pain (especially in Python). Conda (and various other similar tools) is a highly featured package manager which makes life much easier. Check it out at the docs and learn how to use it here or here.

Resources for ML and AI

For Core Machine Learning, we recommend Bishop's *Pattern Recognition and Machine Learning* Textbook [Link] and for Core AI, Russel and Norvig's *Artificial Intelligence: A*

Modern Approach [Link]. Both of these are very comprehensive textbooks that cover the fundamentals of both areas with easy to understand explanations. You do not need to complete them cover to cover in one go, but can use them more as reference books, learning about specific topics at a time.

For those who are new to Deep Learning, we recommend you start with the Stanford CS231 (Computer Vision): [YouTube Link] [Course Link] course which covers everything from loss functions and basics of Neural Networks to Computer Vision. Go through the video lectures and attempt the accompanying assignments to get a good idea of various fundamental concepts. The Deep Learning Book [Link] is also a great resource for detailed introductions to a wide range of concepts.

For those interested in getting into specific areas -
Natural Language Processing:

- Stanford CS224 [YouTube Link] [Course Link]

Reinforcement Learning:

- Berkeley CS285: [YouTube Link] [Course Link]
- NPTEL [YouTube Link]
- Sutton and Barto's Classic Introductory Text [Book Link]

Assignment

The assignment is made up of the following sections

1. Core ML: Variations of Softmax (Compulsory)
2. Domain Specific Questions
    a. Natural Language Processing
    b. Reinforcement Learning

    c. Computer Vision
3. Paper Implementation

Section 1 is compulsory. Additionally, you should attempt **EITHER** two tasks out of the three domain-specific tasks in Section 2 **OR** attempt the Paper Implementation Task which makes up Section 3. Please note that the difficulty of the questions varies, and you will be judged primarily on the effort you have put in.

Along with how well you perform the tasks, you will also be evaluated on how clearly and thoroughly you document your approach and results. So make sure to make your final README or report for each task as detailed and clear as possible.

1. Core ML: Variations of Softmax

Variations of Softmax

The softmax function is a commonly used activation function in neural networks, particularly for classification tasks. It takes in a vector of K real numbers and outputs a probability distribution of K possible outcomes. This function is useful because it allows us to map probabilistic distributions using neural networks.

However, the softmax function can become computationally expensive when the number of classes, denoted as C, is large. This is because the denominator of the softmax function involves a summation over all C classes, taking O(C) time. This can be problematic as the number of classes increases, as it can significantly slow down the training and evaluation of the model.

Overall, this experiment will provide insights into the trade-offs between different softmax implementations in neural networks, particularly in the context of classification tasks with many classes. By understanding the impact of softmax implementations on model performance and training time, we can make more informed decisions when designing and training neural networks for classification tasks.
This is useful for many tasks and is used extensively in classification tasks.

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Though the function looks simple, it can get computationally expensive. This is because the summation of the denominator takes O(C) time, where C is the number of classes. This is a non-issue when the number of classes is small but can cause problems as the number increases.
Report the results/analysis, observations and inferences in a technical report in addition to the code.

You are tasked with the following:

1. Develop a convolutional neural network (CNN) model on the [CIFAR 100] dataset for image classification with the standard softmax.
2. Make a second model with the same architecture but different softmax implementations that reduce the computational complexity of the softmax function (eg: hierarchical. Gumbel-Softmax which is an alternative to the standard softmax that uses a Gumbel distribution to simulate the process of sampling from a categorical distribution, Adaptive Softmax). Report the time complexity for each of the softmax functions you are using.
3. Evaluate the performance of the models using a range of evaluation metrics, including accuracy, precision, recall, F1 score, and confusion matrix.
4. Compare the performance and epoch time between the methods.

BONUS: Develop a transformer-based architecture that integrates these variations of softmax function for image classification using the CIFAR-100 dataset and evaluate the performance across the above-mentioned metrics and also compare the results with the above question using a CNN-based architecture.

2. Domain Specific Questions

Natural Language Processing

Current applications of machine learning are immense, and some include using machine learning for monitoring of social media content for auto flagging of abuse data. Usually, these content are written in the “non-formal” language, often in the way of code-mixed framed sentences. These types of sentences are often difficult to categorize by any ML model. The task for this assignment is to create code-mixed sentences, given a set of monolingual English corpus, and test the performance against a standard code-mixed dataset.

Task: Implement a data creation strategy of creating code mixed sentences using any standard translator from a monolingual corpus (HASOC Dataset, choose English 2021 dataset). Create code mixed data with different code-mixing index (CMI Paper), and then using finetune a pre-trained LLM, (both BeRT and m-BeRT), compare the performance for various (CMI vs Accuracy & BeRT vs mBeRT performance) and justify the observation. (Imp: make sure to do all necessary preprocessing like data cleaning etc. before fine-tuning the LLM)

Bonus: Pick any number of bonus parts depending upon time availability -

1) **Theoretical + Practical** (Easy)**:** How does aligned embedding spaces affect the performance on code-switched languages. Choose any code-switched dataset and verify this observation by evaluating the two LLMs (aligned and non-aligned, eg: InfoXLM vs mBERT).
2) **Practical** (Medium) - create a basic triplet generator (for Knowledge graphs) using any library available, on the newly created code-mixed dataset. After observing the results, comment on what can be done to improve on creating better triplets for code-mixed language.
3) **Practical** (Difficult) - Recent work in using Adapters in LLMs have shown to improve the model performance by significant margin. Use the pre-trained adapter (Link) on the mBERT LLM specifically fine tuned for Hinglish social media data, and compare the performance improvement.

Resources to read:

1) https://aclanthology.org/2020.acl-main.329/
2) https://wiprotechblogs.medium.com/a-tale-of-two-languages-the-code-switch-mix-story-d88e105b07f4
3) CMI - https://umagunturi789.medium.com/a-primer-on-code-mixing-code-switching-9bbde2a15e57

Resource for Bonus Questions:

1) Adapters (just for knowledge) - https://www.youtube.com/watch?v=_Z9qNT-g14U
2) Triplets - https://www.youtube.com/watch?v=80BXKyzCp0k
3) Aligned embeddings - https://arxiv.org/abs/2007.07834 (InfoXLM), https://arxiv.org/pdf/2010.13688

Reinforcement Learning

Studying RL you are bound to come across the term the deadly triad. It is the three conditions such that if all are met, makes an agent extremely unstable. These 3 conditions are as follows:

1. Using function approximation (eg: using neural networks).
2. Using off-policy learning (Offline learning).
3. Using Temporal Difference Learning (Bootstrapping).

These 3 are good properties to use but we must avoid using all 3 together.
Function approximation is a must since it allows for generalisation when the state space is large.
Off-Policy learning allows you from running tedious and unfeasible simulations over and over and allows you to learn from a fixed bank from the data.
TD allows you to calculate the value functions on the go without completing the complete process and can allow for continual learning.

TD normally have a discount factor which biases the agent towards short term goals which may not be ideal. Though an issue this is a necessity for bootstrapping and hence TD learning. Newer works have looked at RL tasks through the lens of sequential learning and have developed a new method called decision transformer.

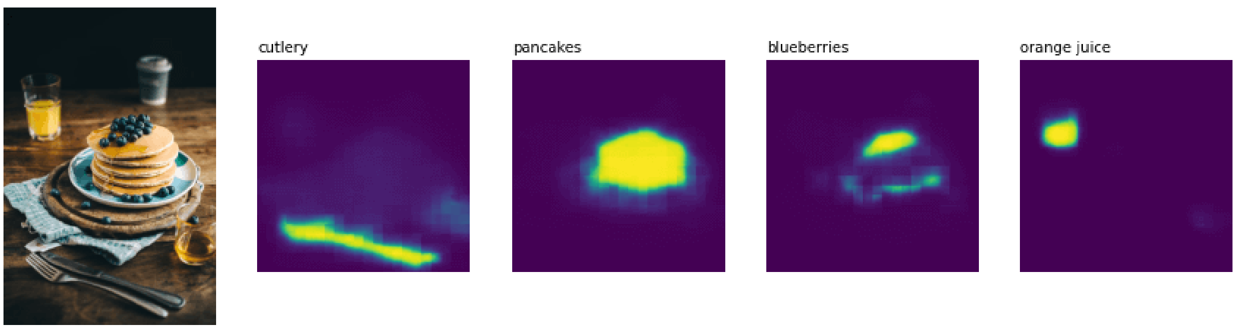
It is a model which takes in a sequence of (R,S,A) tokens learn and predict the next step and reward. The paper '**Decision Transformer: Reinforcement Learning via Sequence Modeling**'. This paper uses transformers to model these sequences.
This leads us to wonder the obvious question of how other sequential models would perform. This is exactly what you would find out in the task.

Formally, you must do the following:

1. Read the paper mentioned above.
2. Implement their methodology but change the architecture with other sequential models other than transformers.
3. Train the model on the Hopper environment available on gymnasium.
4. Compare the results and give your thoughts on why this may be the case.

Computer Vision

Image segmentation is the process of partitioning an image into multiple image regions (segments), with each region corresponding to a certain class of object (eg: car, signboard, person), or having a certain characteristic (eg: colour, intensity, or texture). In other words, it involves assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.



In this task, instead of segmenting images based on a fixed set of classes, or having a certain static characteristic, you will instead segment the image for arbitrary objects based on a text prompt on a subsegment of the Phrasecut dataset. [Dataset: Train+Val] Use gdown or wget to download the dataset within the notebook. [Official paper]

The task is to carry out Zero Shot Segmentation with the help of OpenAI's CLIP. Briefly, CLIP is a network trained to associate similar text and images, and create a contrast between dissimilar ones (contrastive learning). CLIP produces similar vector representations of similar text and images, and vice versa. Read more about CLIP :

- CLIP: Connecting Text and Images
- CLIPSeg: Image Segmentation Using Text and Image Prompts

Task: Implement code to perform the following experiments and provide thorough documentation of your approach (model architecture, hyperparameters etc…) as well as the results in the form of a detailed README or report.

1) Create a model such that it accepts a text prompt and an image as input and uses CLIP to create an embedding of the text prompt and the input image. Train a decoder on top of this to produce a binary segmentation map.

2) Train different models of different complexity using different loss functions and see how they compare for different evaluation metrics.

Bonus:
An important aspect of machine learning is testing of hypotheses. There can be several ways one can come up with a hypothesis, intuition or extending results from previous work.

For the current task at hand i.e. Image Segmentation, consider the following hypothesis: "*Pixel intensities in a grayscale image vary smoothly within an object, as opposed intensities from pixels that cross object boundaries or belong to different objects.*" Effectively, in a grayscale image, properties of pixels that lie within the same object would be much similar when compared to pixels belonging to different objects and this is expected intuitively.

Construct a statistical test to test the above-mentioned hypothesis for segmented grayscale images. For testing this hypothesis, you have access to RGB images, the corresponding grayscale conversion, and object boundaries. You may use the find boundaries function from skimage to get object boundaries from a segmented image. Use your model from the main task above that uses CLIP, for getting a semantic map of your input image.

Hint:
There is no one correct way to test this hypothesis. A lot of statistical methods exist that can be used for hypothesis testing. Below is a very simple test using which we can prove/disprove our intuition.

Consider two populations: absolute pixel intensity differences between pixels belonging to the same object, and absolute pixel intensity differences between pixels belonging to different objects.

Use the data you generated from CLIP to sample from these two populations. Use the Welch T-Test to comment about the means of the two populations from their corresponding samples. Think about how a result on the mean can finally translate to a decision on our hypothesis.

One can extend this idea to RGB images as well. Try out other statistical tests that give results on the means of populations which can help you prove/disprove our hypothesis.

3. Paper Implementation

Your task is to examine the central claim of a paper by implementing it and performing additional empirical experiments.

1. Select any one of the papers published in an A or A* conference (NeurIPS, ICML, ICLR, ICCV, AAAI, ACL, etc). First, read the abstract and give a rough reading of the paper you have chosen and identify its central claims. At this point inform us that you are attempting to implement the paper and tell us about the paper chosen.
2. Once you get approval from us, you can go ahead and give a thorough reading of the paper and implement the paper using a framework of your choice (we highly recommend PyTorch).
3. Formulate and perform additional experiments and ablations (such as trying on a different dataset, with different experimental parameters etc...) to test the central claims of the paper. You can discuss these with your mentor.
4. Document your code with a README and a short review of the paper (preferably in LaTeX). The documentation should contain the results of your evaluation and explain how your additional experiments are evaluating the claims of the paper.

You will be evaluated primarily on your understanding of the paper and the additional evaluations you have performed on its central claims. We are not expecting very clean code, however, it should be able to replicate the results described in the paper and be well-documented. While this task has not been designed for beginners, anyone is welcome to give it a go.