Index_Advanced

March 5, 2024

Preliminaries

Week 0

Plan - Setting up your learning and programming environment

Getting started - Setting up your ML environment - Choosing an ML environment - Quick intro to the tools

1 Week 1

Plan

We give a brief introduction to the course.

We then present the key concepts that form the basis for this course - For some: this will be review

1.1 Intro to Advanced Course

- Introduction to Advanced Course

1.2 Review/Preview of concepts from Intro Course

Notation, concepts

Here is a *quick reference* of key concepts/notations from the Intro course - For some: it will be a review, for others: it will be a preview.
- We will devote a sub-module of this lecture to elaborate on each topic in slightly more depth. - For a more detailed explanation: please refer to the material from the Intro course (repo)

- Review and Preview

Colab

You may want to run your code on Google Colab in order to take advantage of powerful GPU's.

Here are some useful tips:

Google Colab tricks

Review: in-depth

1.2.1 Transfer Learning: Review

1.2.2 Transformers: Review

Suggested reading - Attention - Attention is all you need
- Transfer Learning
- Sebastian Ruder: Transfer Learning

Further reading - Attention - Neural Machine Translation by Jointly Learning To Align and Translate - Geron Chapter 16 - An Analysis of BERT's Attention

2 Week 2: Review/Preview (continued); Technical

Preview

There is lots of interest in Large Language Models (e.g., ChatGPT). These are based on an architecture called the Transformer. We will introduce the Transformer and demonstrate some amazing results achieved by using Transformers to create Large Language Models.

Attention is a mechanism that is a core part of the Transformer. We will begin by first introducing Attention.

We will then take a detour and study the Functional model architecture of Keras. Unlike the Sequential model, which is an ordered sequence of Layers, the organization of blocks in a Functional model is more general. The Advanced architectures (e.g., the Transformer) are built using the Functional model.

Once we understand the technical prerequisites, we will examine the code for the Transformer.

Plan

We continue the review/preview of key concepts that we started last week.

Our ultimate goal is to introduce the Transformer (which uses Attention heavily) in theory, and demonstrate its use in Large Language Models.

2.1 Review/preview continued

2.1.1 Transformers: Review continued

- Attention (continued)
- Transformer

2.1.2 Natural Language Processing: Review

2.2 Functional Models

Plan

Enough theory (for the moment) !

The Transformer (whose theory we have presented) is built from plain Keras.

Our goal is to dig into the **code** for the Transformer so that you too will learn how to build advanced models.

Before we can do this, we must - go beyond the Sequential model of Keras: introduction to the Functional model - understand more "advanced" features of Keras: customomizing layers, training loops, loss functions - The Datasets API

Basics

We start with the basics of Functional models, and will give a coding example of such a model in Finance.

- Functional API

2.2.1 Functional Model Code: A Functional model in Finance: "Factor model"

We illustrate the basic features of Functional models with an example - does not use the additional techniques of the next section (Advanced Keras)

Autoencoders for Conditional Risk Factors - code

3 Assignments

Your assignments should follow the Assignment Guidelines

4 Additional Deep Learning resources

Here are some resources that I have found very useful.

Some of them are very nitty-gritty, deep-in-the-weeds (even the "introductory" courses) - For example: let's make believe PyTorch (or Keras/TensorFlow) didn't exists; let's invent Deep Learning without it ! - You will gain a deeper appreciation and understanding by re-inventing that which you take for granted

4.1 Andrej Karpathy course: Neural Networks, Zero to Hero

- PyTorch
- Introductory, but at a very deep level of understanding
  - you will get very deep into the weeds (hand-coding gradients !) but develop a deeper appreciation

4.2 fast.ai

`fast.ai` is a web-site with free courses from Jeremy Howard. - PyTorch - Introductory and courses "for coders" - Same courses offered every few years, but sufficiently different so as to make it worthwhile to repeat the course ! - Practical Deep Learning - Stable diffusion - Very detailed, nitty-gritty details (like Karpathy) that will give you a deeper appreciation

4.3 Stefan Jansen: Machine Learning for Trading

An excellent github repo with notebooks - using Deep Learning for trading - Keras - many notebooks are cleaner implementations of published models

[1]:
```
print("Done")
```

```
Done
```