Lifelong Machine Learning (Day-1)

28th June 2021

On completion of chapter 1 on lifelong machine learning, I understood the various aspects of Lifelong or Continual learning, or just the basic definition of it which goes as, for a given new task say Tn+1 for the given data Dn+1, we use the previous tasks upto Tn also the data till then Dn, the knowledge use from that plays an important role in understanding the new data and task. Now the major problem that occurs here is that of “Catastrophic Forgetting”, here when the model learns new data for its new task, it forgets the previously learnt parameters for the previous task, and since it is a relatively new area of research, some extensive research in this field could be very vital. Speaking about the architecture of LL, first is the primary Knowledge Base(KB) which is used for storing the data and model from the previously learnt tasks now this itself has several subcomponents, which are as follows:

1. Past Information System(PIS) – Storage of past data, intermediate result and model.
2. Meta Knowledge Miner (MKM) – It performs meta mining from the PIS(extraction of higher level knowledge).
3. Meta Knowledge Store(MKS) – The data mined by MKM in PIS is stored here.
4. Knowledge Reasoner (KR) – It makes inference based on the knowledge from MKM and PIS to generate more knowledge.

And other parts of the architecture following the Knowledge base are Knowledge Based Learner(KBL), Task-Based Knowledge Miner(TKM), Model, Application, Task Manager. These are all the major parts of a lifelong learning system, but in reality implementing the algorithm isn’t so sophisticated and some of the overall model are pretty simple, with few components from the mentioned above. The key challenges which are faced by LL is mainly two kinds of problems, one is Applicability of knowledge and the other one is correctness of the knowledge. Seeing the evaluation technique for LL is firstly running the data on previous tasks, running the data on new task , running a baseline algorithm( isolated learning) and finally comparing the results of both to show that LL has better prospects than Isolated learning because of learning from past knowledge and experience.

Implemented basic Tensor operations using PyTorch and also did basic linear regression for prediction of fruit prices with given input and target tensor values a model was built and the loss function used to evaluate the model was Mean Square Error.

Day – 2

29th June 2021

Completed the second chapter from Lifelong Machine Learning, we see various learning paradigms, from transfer learning, multi task learning, Online learning, reinforcement learning and finally meta learning. For transfer learning and multi task learning we also see their implementation with Deep Neural Networks and how these techniques are very different from Lifelong Learning, only two paradigms from the mentioned above come close the definition proposed for lifelong learning, Say we take Multi task learning which is closely performed for Supervised learning when it is joint with Online learning it is a form of Lifelong Learning, similarly even Meta learning, which exactly doesn’t represent the said definition but since it makes use of many tasks to help learn the new task. We go in depth in each method for example in transfer learning we take the example of Naïve Bayes Transfer Classifer(NBTC), it was a proposed method for transfer learning it is employed in two steps:

1. Building an initial Naïve Bayesian Classifer for the labelled Data(Dl) under the Distribution(dl) from the source domain.
2. Run an Expectation-Maximization Algorithm together with the data with the target unlabeled data to find a local optimal model under the target domain distribution(du).

This was also explained in the context of how Deep Neural Networks implementation takes place in Natural Language Processing Applications. Next followed by Multi task learning, how GO-MTL( Group and Overlap Multi Task Learning), which explains how the outliers was ignored in Traditional MTL, but in the proposed model of GO-MTL even the outliers having negative correlation have some information which could be also useful and be part of the cluster which are grouped together based on similar features. Same way for Reinforcement Learning, it is a way where the agent learns on trial and error basis and how it is different from LL.

Apart from this implemented a basic neural network using Pytorch, same as yesterday to predict the yield of fruits of apple and orange, considering three for atmospheric conditions, used loss function and Stochastic Gradient Descent as the optimizer, trained the model for 100 epochs. And able to get a better result than yesterday using neural networks.

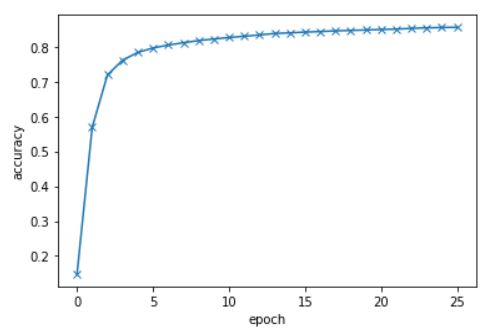
Day-3

30th June 2021

Started reading the third chapter of Supervised Machine Learning, it gives insights about Lifelong Memory Based Learning, the algorithm that were proposed for this task were, KNN( K Nearest Neighbors) and Shepard’s method, In KNN it takes only K values of data from the dataset whereas in the shepard’s method takes the entire dataset and does the mathematical computation, then we proceed to EBNN( Explaination Based Neural Networks) which works very similarly to KNN but here EBNN estimates the slope of the tangent function at each point and adds it to the vector representation of the data point.

In PyTorch Implementation, today implemented Logistic regression Neural Network model on MNIST dataset to predict the number from digits which were handwritten, the optimizer used here was Stochastic Gradient Descent(SGD) and the loss function used here was cross-entropy which has a advantage of applying softmax function internally, so even the tensor values( the probabilities ) that come in negative value are converted to positive by multiplying the result with -1. Learnt about nn.module class which is used for adding functionalities to custom model, trained the model for 25 epochs was able to achieve a accuracy upto 85.6 %, this value could be further pushed upto 90% and above by Hyperparameter tuning like the learning rate, and training it for more epochs. In the test set took four cases, out of 4 the model was correctly able to predict 3 of them, which it hasn’t seen before, the training images of 60000 was split as 50000 and 10000 to training data and validation data.

Started Reading about some unsupervised learning algorithms, where learnt about K Means clustering, how unsupervised learning is quite different from Supervised learning, and how by finding the centeroid of the cluster, the like instance can come together for the label of the instance and not labels for the entire dataset. This information could also be extended to the Lifelong Unsupervised Learning where the model can learn instances from the previous known clusters and group themselves and also with the knowledge gained because of the new cluster, can help in the previous data to.

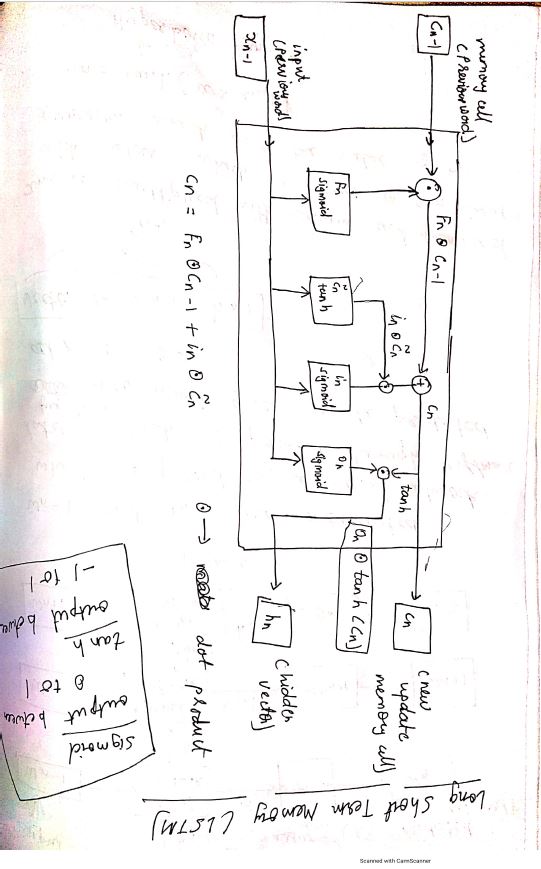
 The graph between accuracy and epoch.

Day -4

1st July 2021

Proceeding with book yesterday, the book heavily depended on banking with ideas from Natural Language Processing, Unsupervised Learning and also parts of Deep Belief Networks( Restricted Boltzmann Machines ) or also Semisupervised Algorithms. To get a idea of how these work, I spent reading about NLP got a very broad perspective and idea starting from Word vectors to very sophisticated state of the art model like Long Short Term Memory(LSTM), Architecture of RNN and how that model is extended to LSTM, and also how to caption a image, Image to text model, Firstly the image is given as a input to the Deep Convolutional Neural Network which extract the feature and gives the basic previous memory unit (c0), now this c0 along with the hidden vector is provided as the input to the neural network or the LSTM model, now the LSTM model generates the text for the given picture.

Working Of LSTM:



In the above picture f\_n, i\_n and o\_n are called as control networks, where their names stand as forget control, input control and output control respectively. c`\_n updates the memory cell and it is using Tanh non linear function in its network whereas the other three use sigmoid non linear function.

Got insights about Simple Word Embedding Model which can be used for sentiment analysis, that complex models like LSTM-CNN or RNN need not to prove best in every case, a efficient and a better accuracy for the model can be easily achieved by simple logistic regression model.

Day – 5

3rd July 2021

Started preparing with Convolutional Neural Networks and its implementation with TensorFlow, started reading about CNN from the basics and its evolution and spent sometime also looking at the Architecture of Unet, Convolutional Neural Networks is a deep neural network with special feature for analyzing images and for Computer vision based application, apart from computer vision it is also used for Natural Language Processing based applications to. Like weights in Neural Networks, in CNN there are filters which are equivalent to weights and CNN comprises of layers which are formed convolutional layers and Max pooling layers. Convolutional Layers perform a very similar operation to that of the Convolutional operation which is done in Signal Processing and it is very closely related to some high level mathematical concepts of Fourier and Laplace Transforms, The entire operation of CNN occurred from the understanding of humans brain as to how humans perceive images through a part in brain knows as the Visual Cortex.

Also covered the topic of UNet Architecture which has a primary application in Biomedical Image Processing but is also widely use for other computer vision applications such as Object Detection, Semantic Segmentation and also Instance segmentation. There are many other architecture which are developed under CNN which I would be reading and implementing in the coming days.

Apart from reading, did a very basic implementation of a 2d convolutional layer using TensorFlow having 2 filters of 4D tensor objects, and generated a feature map for the image which was generated using Scikit learn in-built function, and also converted the images into numpy array, and before the images were divided by 255, so that all images are scaled to floating values between 0 to 1.

Architecture Of UNet:

