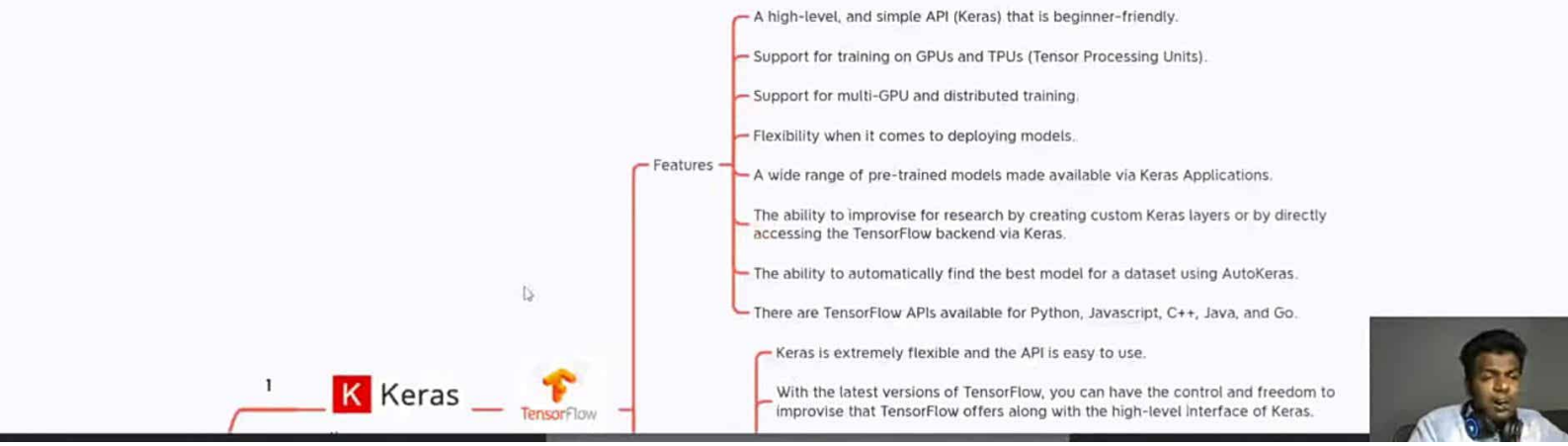
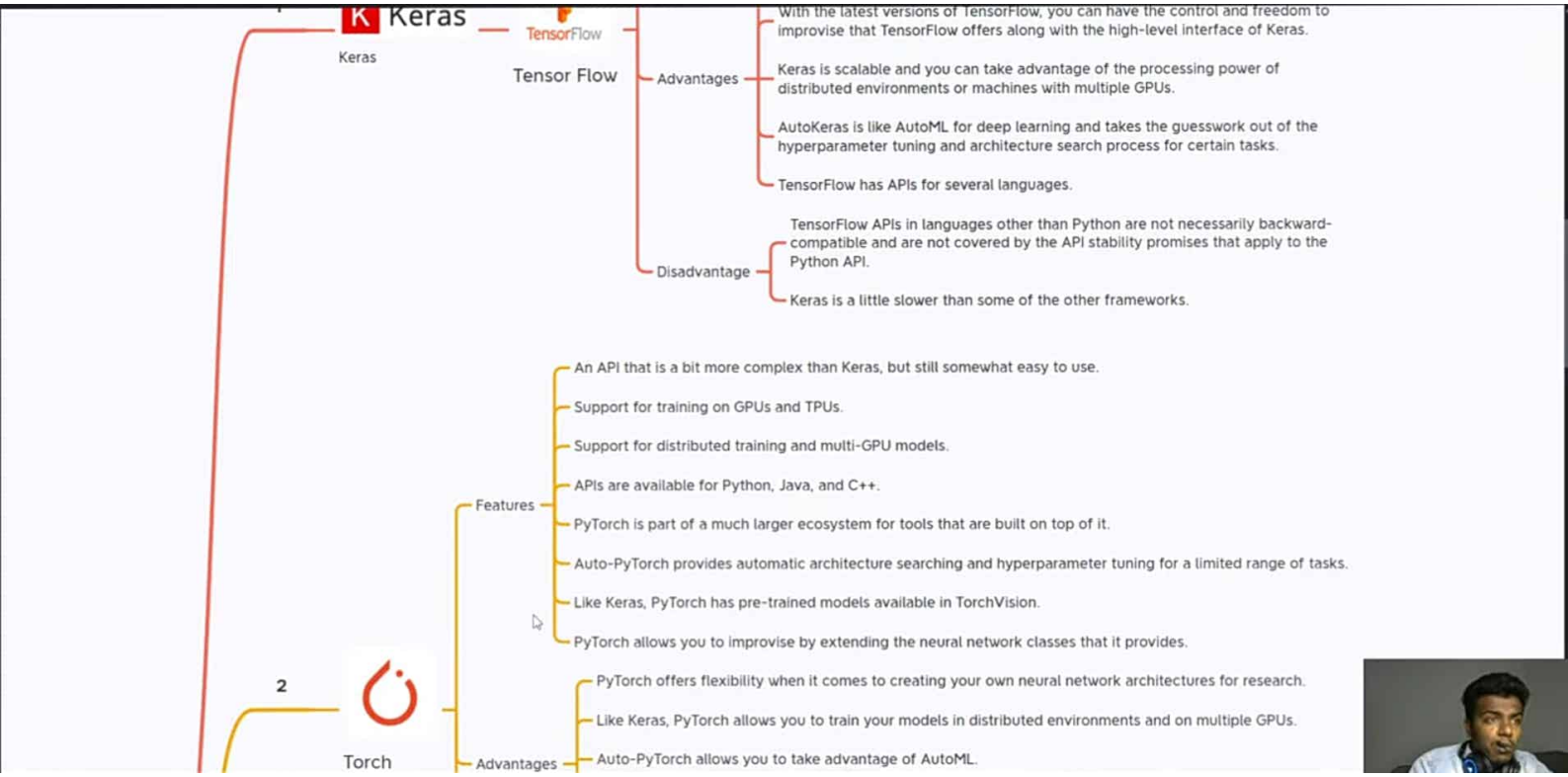
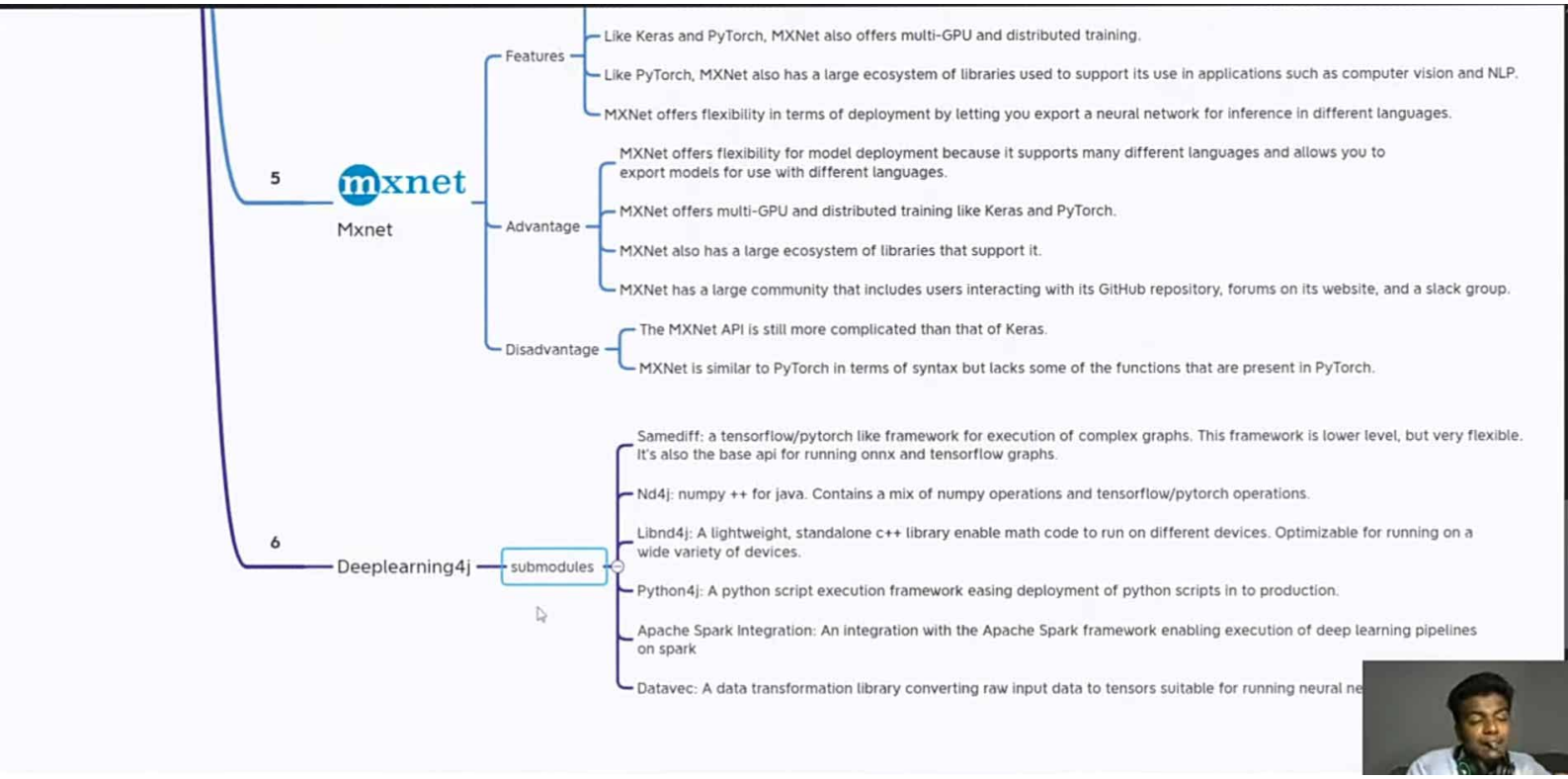
FRAMEWORKS

Frameworks	Written Language	CUDA Support	Pretrained Model	Release Year
TensorFlow	C++, Python	Yes	Yes	2015
Keras	Python	Yes	Yes	2015
PyTorch	Python, C	Yes	Yes	2016
Caffe	C++	Yes	Yes	2013
Deeplearning4j	C++, Java	Yes	Yes	2014


1
Keras
TensorFlow
Features
A high-level, and simple API (Keras) that is beginner-friendly.
Support for training on GPUs and TPUs (Tensor Processing Units).
Support for multi-GPU and distributed training.
Flexibility when it comes to deploying models.
A wide range of pre-trained models made available via Keras Applications.
The ability to improvise for research by creating custom Keras layers or by directly accessing the TensorFlow backend via Keras.
The ability to automatically find the best model for a dataset using AutoKeras.
There are TensorFlow APIs available for Python, Javascript, C++, Java, and Go.
Keras is extremely flexible and the API is easy to use.
With the latest versions of TensorFlow, you can have the control and freedom to improvise that TensorFlow offers along with the high-level interface of Keras.


Keras
Keras
TensorFlow
Tensor Flow
Advantages
With the latest versions of TensorFlow, you can have the control and freedom to improvise that TensorFlow offers along with the high-level interface of Keras.
Keras is scalable and you can take advantage of the processing power of distributed environments or machines with multiple GPUs.
AutoKeras is like AutoML for deep learning and takes the guesswork out of the hyperparameter tuning and architecture search process for certain tasks.
TensorFlow has APIs for several languages.
Disadvantage
TensorFlow APIs in languages other than Python are not necessarily backward-compatible and are not covered by the API stability promises that apply to the Python API.
Keras is a little slower than some of the other frameworks.
2
Torch
Features
An API that is a bit more complex than Keras, but still somewhat easy to use.
Support for training on GPUs and TPUs.
Support for distributed training and multi-GPU models.
APIs are available for Python, Java, and C++.
PyTorch is part of a much larger ecosystem for tools that are built on top of it.
Auto-PyTorch provides automatic architecture searching and hyperparameter tuning for a limited range of tasks.
Like Keras, PyTorch has pre-trained models available in TorchVision.
PyTorch allows you to improvise by extending the neural network classes that it provides.
Advantages
PyTorch offers flexibility when it comes to creating your own neural network architectures for research.
Like Keras, PyTorch allows you to train your models in distributed environments and on multiple GPUs.
Auto-PyTorch allows you to take advantage of AutoML.


PyTorch has a large ecosystem of additional frameworks built on top of it, such as Skorch, which provides full Scikit-learn compatibility for your Torch models.
Disadvantage
The API is not as simple as Keras and a little tougher to use.
Auto-PyTorch is arguably not as sophisticated as AutoKeras.
arning
works
4
Caffe
Features
Caffe is extremely fast. In fact, with a single Nvidia K40 GPU, Caffe can process over 60 million images per day.
The Caffe Model Zoo features many pre-trained models that can be reused for different tasks.
Caffe has great support for it's C++ API.
Advantages
Caffe is really fast, and some benchmark studies have shown that is even faster than TensorFlow.
Caffe was designed for computer vision applications.
Disadvantages
The documentation is not as easy to follow compared to Keras and PyTorch.
The community is not as big as the ones for Keras and PyTorch.
The API for Caffe is not as beginner-friendly as the ones for Keras and PyTorch.
Caffe is great for computer vision applications but not suited for NLP applications involving architectures such as recurrent neural networks.
As a framework, it is outdated in 2020 and not as popular as Keras or PyTorch.
Features
MXNet is flexible and supports eight different languages, including Python, Scala, and Julia.
Like Keras and PyTorch, MXNet also offers multi-GPU and distributed training.
Like PyTorch, MXNet also has a large ecosystem of libraries used to support its use in applications such as co
MXNet offers flexibility in terms of deployment by letting you export a neural network for inference in differe
MXNet offers flexibility for model deployment because it supports many different languages and allows yo


Like Keras and PyTorch, MXNet also offers multi-GPU and distributed training.
Features
Like PyTorch, MXNet also has a large ecosystem of libraries used to support its use in applications such as computer vision and NLP.
MXNet offers flexibility in terms of deployment by letting you export a neural network for inference in different languages.
5
mxnet
Mxnet
MXNet offers flexibility for model deployment because it supports many different languages and allows you to export models for use with different languages.
Advantage
MXNet offers multi-GPU and distributed training like Keras and PyTorch.
MXNet also has a large ecosystem of libraries that support it.
MXNet has a large community that includes users interacting with its GitHub repository, forums on its website, and a slack group.
Disadvantage
The MXNet API is still more complicated than that of Keras.
MXNet is similar to PyTorch in terms of syntax but lacks some of the functions that are present in PyTorch.
Samediff: a tensorflow/pytorch like framework for execution of complex graphs. This framework is lower level, but very flexible. It's also the base api for running onnx and tensorflow graphs.
Nd4j: numpy ++ for java. Contains a mix of numpy operations and tensorflow/pytorch operations.
6
Deeplearning4j
submodules
Libnd4j: A lightweight, standalone c++ library enable math code to run on different devices. Optimizable for running on a wide variety of devices.
Python4j: A python script execution framework easing deployment of python scripts in to production.
Apache Spark Integration: An integration with the Apache Spark framework enabling execution of deep learning pipelines on spark
Datavec: A data transformation library converting raw input data to tensors suitable for running neural ne

Deep Learning Terminology - 1

Gradient Clipping

- Gradient Clipping is one way to solve the problem of exploding gradients.
- Exploding gradients arise in deep networks when gradients associating weights and the net's error become too large.
- Exploding gradients are frequently encountered in RNNs dealing with long-term dependencies.
- One way to clip gradients is to normalize them when the L2 norm of a parameter vector surpasses a given threshhold.

Deep Learning Terminology - 2

GRU (Gated recurrent unit)

- A GRU is a pared-down LSTM. GRUs rely on gating mechanisms to learn long-range dependencies while sidestepping the vanishing gradient problem.
- They include reset and update gates to decide when to update the GRUs memory at each time step.
- GRU is simplified LSTM

Deep Learning Terminology - 3

Word2Vec

- Word2vec is a two-layer neural net that processes text.
- Its input is a text corpus and its output is a set of vectors: feature vectors for words in that corpus.
- While Word2vec is not a deep neural network, it turns text into a numerical form that deep nets can understand