



Learning GANs: A Roadmap

Machine Learning India

What is a GAN?

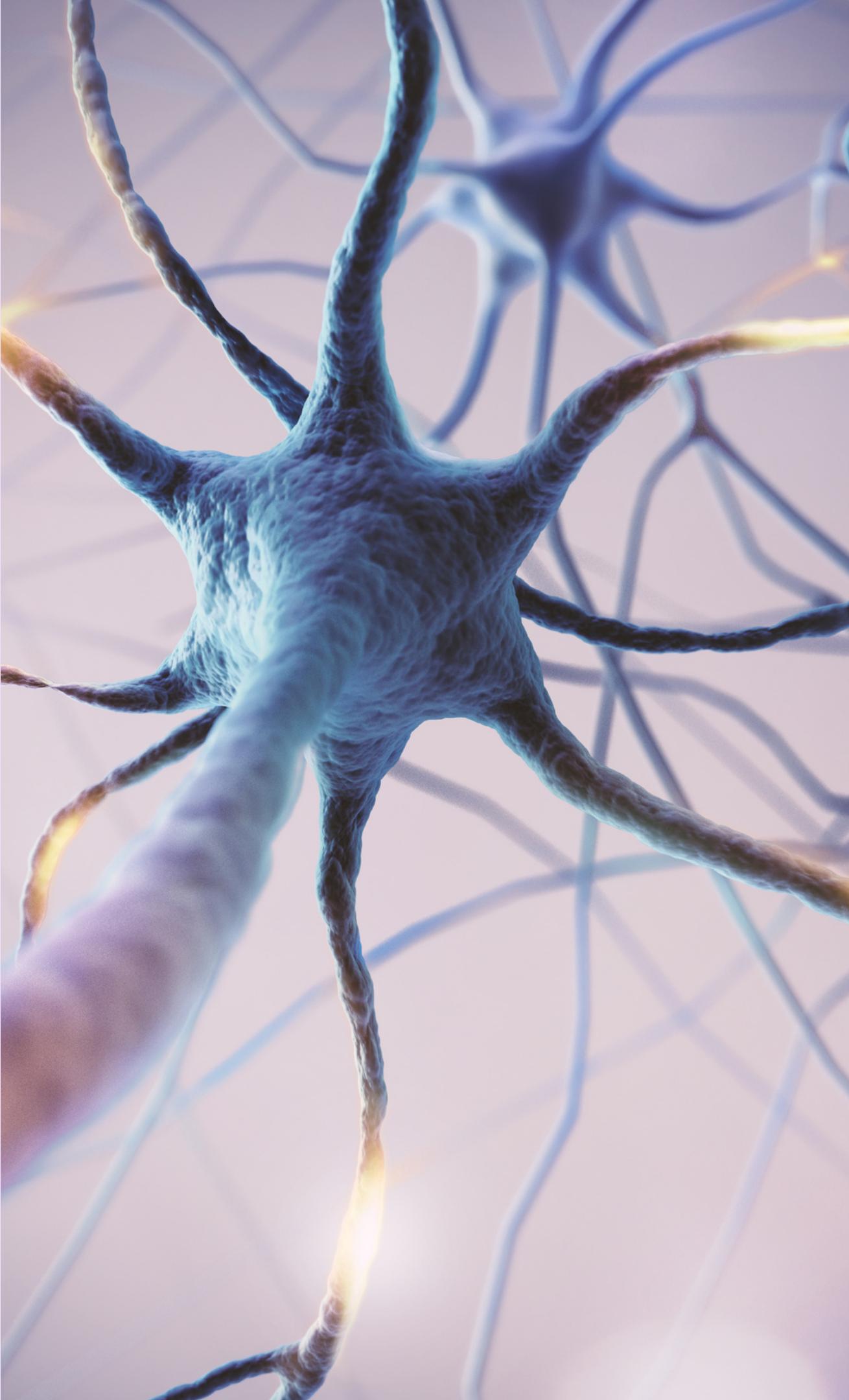
A generative adversarial network is a class of machine learning frameworks designed by Ian Goodfellow and his colleagues in 2014. Two neural networks contest with each other in a game. Given a training set, this technique learns to generate new data with the same characteristics and statistics as the training set.

The purpose of this roadmap is to introduce a shortcut to developers and researchers for finding useful resources about Generative Adversarial Networks.



Week 1: Getting Started

- To get a basic understanding what a Neural Network is, watch this excellent playlist by [3Blue1Brown - Neural Networks](#).
- Now, to build your own Neural Network, try completing this short course by Andrew NG - [Neural Networks and Deep Learning](#). You can opt for Financial aid, if you need to.
- It is sometimes overwhelming to visualise how a neural network improves its performance over time. This website will allow you to do just the same - [Neural Network Playground](#).
- Exhausted by all the math? Here's an article to get you motivated - [Applications of GANs](#).



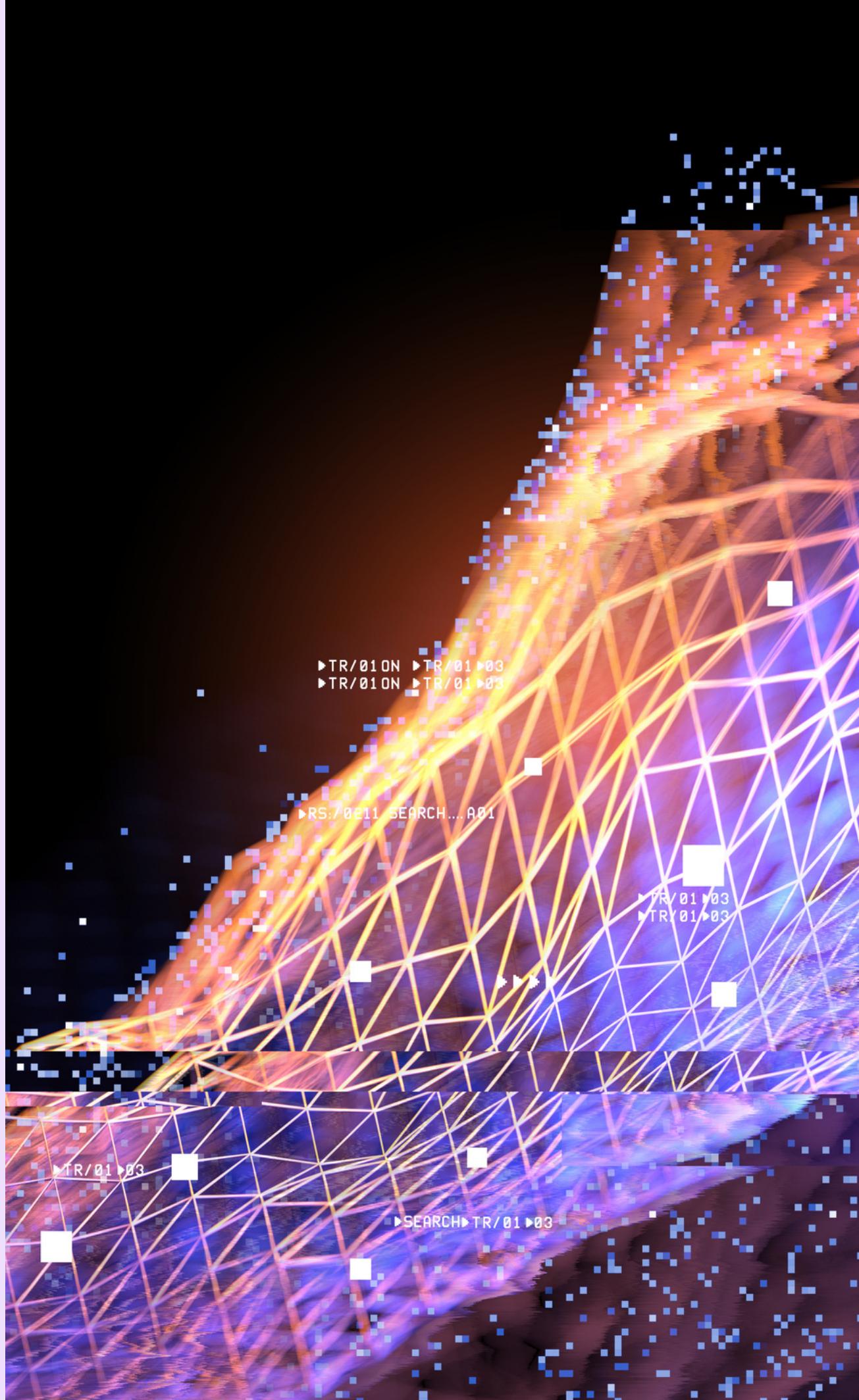
Week 2: Learning PyTorch

- Libraries like PyTorch and Tensorflow make implementing neural nets a bliss. PyTorch's [60 Minute Blitz](#) will help you get started. It's recommended that you type your own code as well.
- Hopefully you would have got a clear understanding of what a neural network is. It is now time to tinker around with them to decrease training time, and improve accuracy. Do this course on [Hyperparameter Tuning](#) to know more. You can skip the TensorFlow part if you wish to, since you already got an idea of PyTorch.
- You can now do further PyTorch [tutorials](#). The above course would help you understand these examples better. Make your own [Google Colab](#) notebooks and tinker around. It's important to try out various values of hyperparameters for better practical learning.



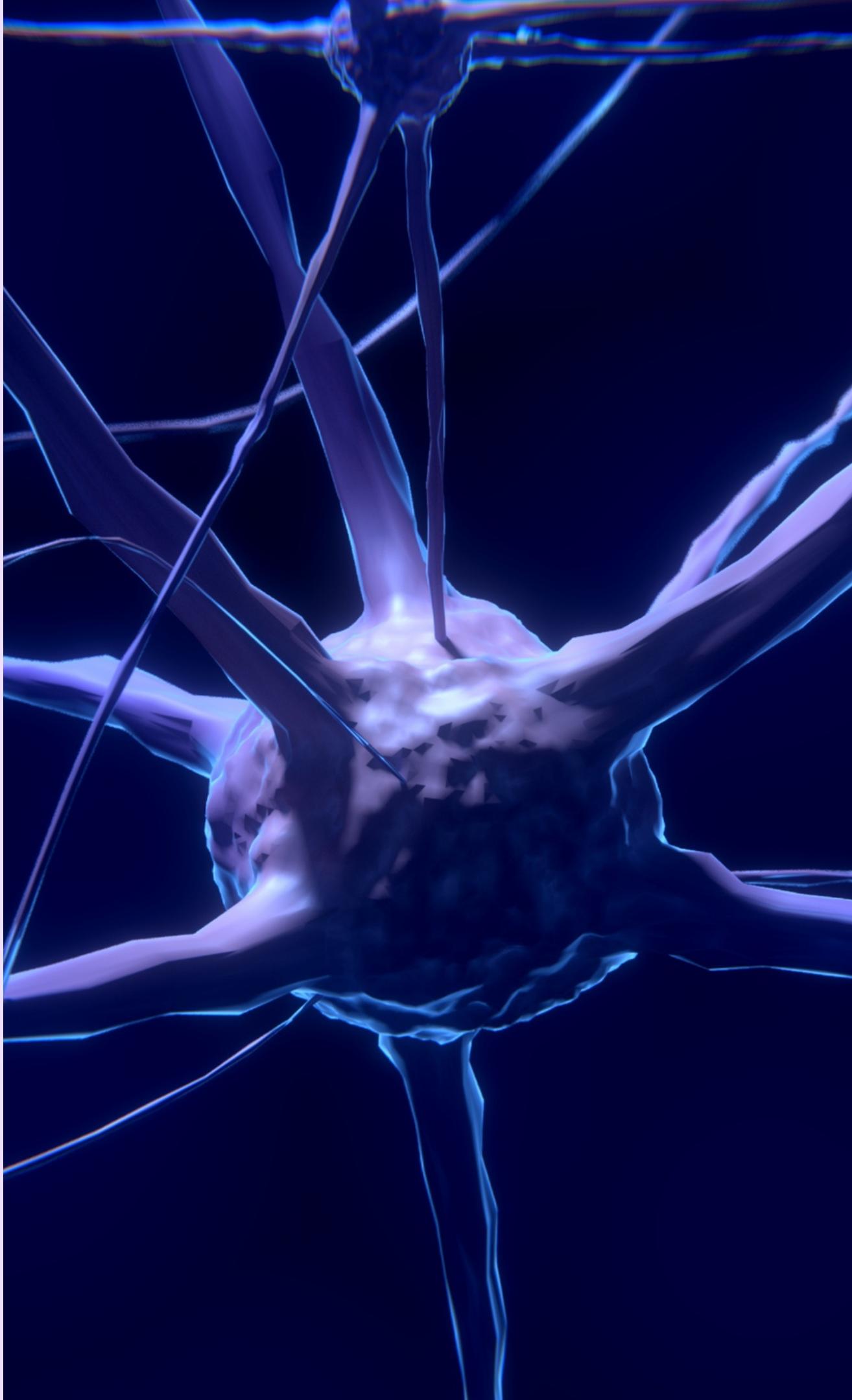
Week 3: Attempting a Kaggle Challenge

- MNIST dataset is a large database of handwritten digits. Pytorch has a tutorial to train your NN on the MNIST dataset. You can leave the CNN part for now.
- Kaggle is a community of data scientists where you can find a vast variety of datasets and competitions to hone your skills. Try attempting this Kaggle Challenge to get started - Digit Recognizer.



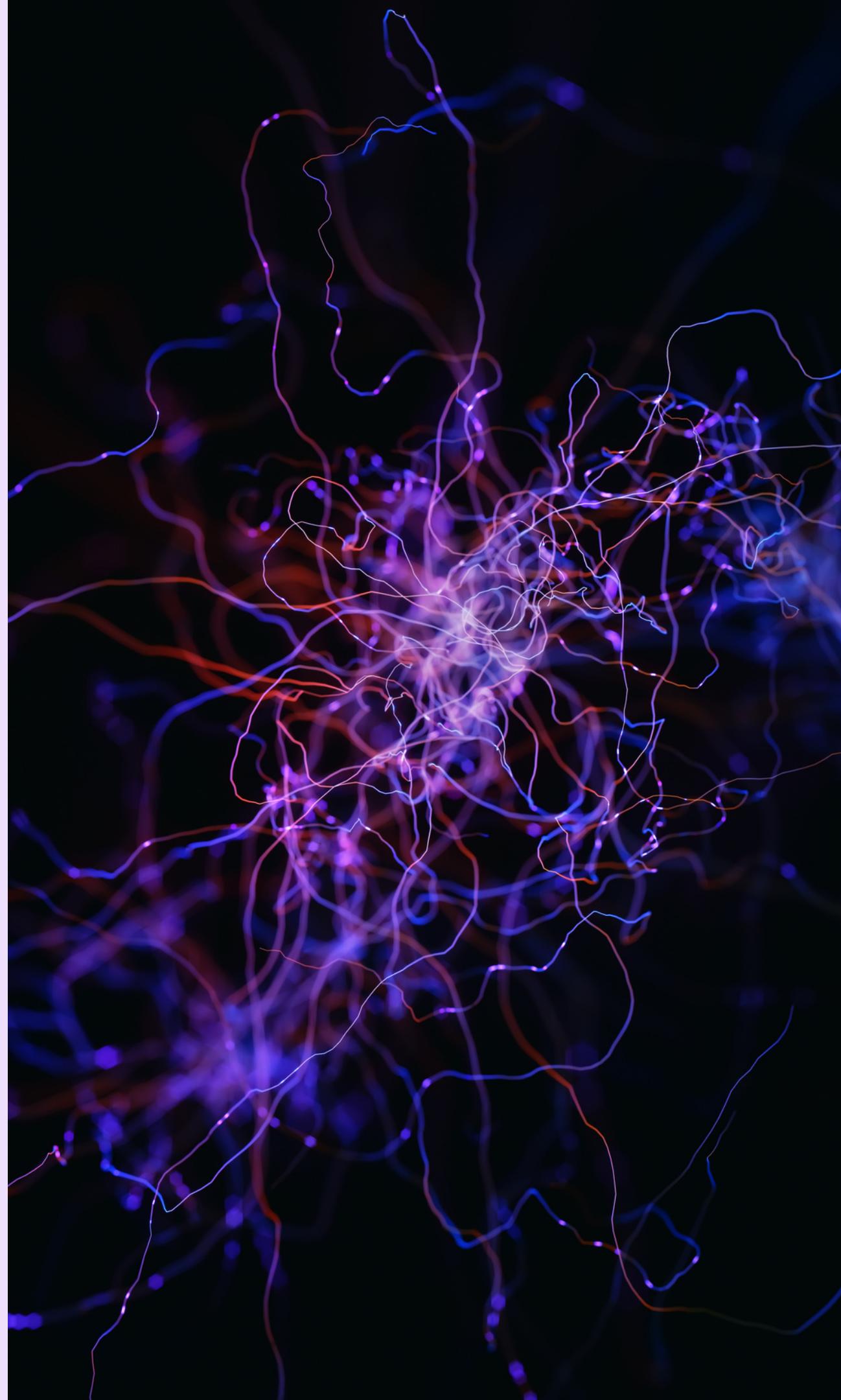
Week 4: Convolutional NNs

- Convolutional Neural Networks have been considered revolutionary in processing images. Read either of these articles to get an understanding of how they work -
 - [CNN in PyTorch](#)
 - [CNN in PyTorch \(2\)](#).
- CIFAR-10 is an established computer-vision dataset. Attempt a related challenge on Kaggle - [Object Recognition](#).
- Try implementing CNN models for classification problems on your own. This article will guide you as to how you can [Create your own dataset](#).



Week 5: Generative Adversarial NNs

- At last, we will now start with GANs. In case you have never read a research paper before, here is a guide to get you started - [How to Read a Research Paper](#).
- It might be overwhelming to read this paper but it is strongly recommended that you do - [GANs](#).
- It is okay even if you do not understand all of it. These articles might come handy -
 - [What are GANs](#)
 - [Understanding GANs](#)
- Now that you have understood how a GAN works, you can try implementing GANs for simple datasets. You can refer to the code given here - [PyTorch GANs](#). You can leave DCGANs for now.
- Also read this article for some [Tips to make GANs work](#).



Week 6: Miscellaneous

- Researchers have developed various types of GANs for specialised applications. We have listed some of the most popular ones -
 - [DCGAN](#)
 - [CycleGAN](#)
 - [StackGAN](#)
 - [InfoGAN](#)
 - [WassersteinGAN](#)
- You can try implementing these from their research papers to get a better understanding. If you find it difficult to get started, PyTorch provides an [Implementation](#) of DCGANs as well.
- Finally, try attempting this [Assignment](#). Here are the [Solutions](#).



Thank you!

**Questions? Clarifications?
Let us know!**