

# INAIO 2026 Resource List

Below is a curated set of resources for INAIO/IOAI preparation. These materials may overlap and may include topics beyond the syllabus, so students should use their judgment to build strong foundational understanding rather than treat this as a fixed, board-style syllabus.

Adaptability and effective use of AI tools are key to the assessment. Students may need to solve advanced problems using information from the problem statements, supported by AI assistants and whitelisted documentation, even if such content is not covered in the resources listed below.

---

## Resource Short List

- [Python](#)
  - [PyTorch](#)
  - [Machine Learning](#)
  - [Deep Learning](#)
  - [Natural Language Processing](#)
  - [Word Embeddings](#)
  - [Transformers](#)
  - [Generative Modeling](#)
  - [Computer Vision](#)
  - [Recent Developments in CV](#)
- 

## Recommended General Courses

- **Practical Deep Learning for Coders (fast.ai)** — neural networks, optimization, regularization, vision
  - <https://course.fast.ai>

- **CS224n: Natural Language Processing with Deep Learning (Stanford)** — embeddings (Word2Vec, GloVe), RNNs, attention, Transformers
    - <http://web.stanford.edu/class/cs224n/>
  - **HuggingFace LLM Course** — Transformers, Vision Transformers (ViT), CLIP, multimodal models
    - <https://huggingface.co/learn>
  - **CS231n: Deep Learning for Computer Vision** — CNNs, object detection, segmentation, representation learning
    - <http://cs231n.stanford.edu>
- 

## SECTION 1 — Foundational Skills & Classical ML

### Programming Fundamentals

- **Practical Python Programming** — loops, functions, core Python  
<https://dabeaz-course.github.io/practical-python/>
- **NumPy Documentation** — array operations, broadcasting  
<https://numpy.org/doc/stable/user/index.html>
- **Pandas Documentation** — dataframes, manipulation  
<https://pandas.pydata.org/docs/>
- **Pandas Tutorials (Kaggle)** — introductory notebook series  
<https://www.kaggle.com/learn/pandas>
- **NumPy Illustrated (visual guide)** — visual explanations  
<https://medium.com/@jalammar/numpy-illustrated-the-visual-guide-to-numpy-3d6f7f39c0d6>
- **Matplotlib Tutorial** — plotting basics  
<https://matplotlib.org/stable/tutorials/>
- **Seaborn Tutorial** — statistical visualization  
<https://seaborn.pydata.org/tutorial.html>
- **Intro to Scikit-Learn** — API, models, workflow  
<https://jakevdp.github.io/PythonDataScienceHandbook/05.02-introducing-scikit-learn.htm>

!

- **PyTorch Basics** — tensors, autograd, modules  
<https://pytorch.org/tutorials/beginner/basics/intro.html>
  - **PyTorch Debugging Guide**  
[https://pytorch.org/tutorials/beginner/debugging\\_tutorial.html](https://pytorch.org/tutorials/beginner/debugging_tutorial.html)
  - **Tensors Explained (Medium)** — multidimensional arrays  
<https://medium.com/towards-data-science/tensors-explained-8c399ecf4270>
  - **PyTorch Randomness & Reproducibility Notes** — seeds and determinism  
<https://pytorch.org/docs/stable/notes/randomness.html>
  - **Training Models on GPU (PyTorch Tutorial)**  
[https://pytorch.org/tutorials/beginner/blitz/cifar10\\_tutorial.html#training-on-gpu](https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html#training-on-gpu)
- 

## Supervised Learning

- **Linear Regression Lecture Notes (CMU)**  
<https://www.stat.cmu.edu/~cshalizi/mreg/15/lectures/02/lecture-02.pdf>
- **Logistic Regression (Speech & Language Processing chapter)**  
<https://web.stanford.edu/~jurafsky/slp3/5.pdf>
- **K-Nearest Neighbors (Scikit-Learn Guide)**  
<https://scikit-learn.org/stable/modules/neighbors.html>
- **Decision Trees (Scikit-Learn Guide)**  
<https://scikit-learn.org/stable/modules/tree.html>
- **Understanding Random Forests (arXiv)**  
<https://arxiv.org/pdf/1407.7502>
- **XGBoost Model Tutorial**  
<https://xgboost.readthedocs.io/en/stable/tutorials/model.html>
- **Support Vector Machines (Scikit-Learn Guide)**  
<https://scikit-learn.org/stable/modules/svm.html>

---

## Unsupervised Learning

- **K-Means (Scikit-Learn Guide)**  
<https://scikit-learn.org/stable/modules/clustering.html#k-means>
  - **PCA Explained (arXiv overview)**  
<https://arxiv.org/pdf/1404.1100.pdf>
  - **t-SNE (Distill.pub Visual Explanation)**  
<https://distill.pub/2016/misread-tsne/>
  - **UMAP Documentation**  
<https://umap-learn.readthedocs.io/en/latest/>
  - **DBSCAN Beginner Guide (Medium)**  
<https://towardsdatascience.com/clustering-like-a-pro-dbscan-55ef8d37d8df>
  - **Hierarchical Clustering (Scikit-Learn Guide)**  
<https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering>
- 

## Model Evaluation & ML Practice

- **Precision, Recall, Classification Metrics (Google ML Crash Course)**  
<https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall>
- **Underfitting & Overfitting (Scikit-Learn Example)**  
[https://scikit-learn.org/stable/auto\\_examples/model\\_selection/plot\\_underfitting\\_overfitting.html](https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html)
- **Hyperparameter Tuning (Grid Search, Randomized Search)**  
[https://scikit-learn.org/stable/modules/grid\\_search.html](https://scikit-learn.org/stable/modules/grid_search.html)
- **Cross-Validation (Scikit-Learn Guide)**  
[https://scikit-learn.org/stable/modules/cross\\_validation.html](https://scikit-learn.org/stable/modules/cross_validation.html)

- **Confusion Matrix, ROC, AUC (Scikit-Learn Evaluation)**  
[https://scikit-learn.org/stable/modules/model\\_evaluation.html](https://scikit-learn.org/stable/modules/model_evaluation.html)
- 

## SECTION 2 — Neural Networks & Deep Learning

### Neural Network Theory

- **Backpropagation (Colah's Blog)**  
<https://colah.github.io/posts/2015-08-Backprop/>
  - **Backprop from Scratch (Roelants)**  
<https://peterroelants.github.io/posts/backprop/>
  - **Activation Functions Overview**  
<https://amansari.info/primers/activations>
  - **Loss Functions Overview**  
<https://amansari.info/primers/losses>
  - **Karpathy's "A Recipe for Training Neural Networks"**  
<https://karpathy.github.io/2019/04/25/recipe/>
- 

### Deep Learning Techniques

- **MLPs (Deep Learning Statistical View)**  
<https://blog.shakirm.com/2015/01/a-statistical-view-of-deep-learning/>
- **Stochastic Gradient Descent (D2L Chapter)**  
[https://d2l.ai/chapter\\_optimization/sgd.html](https://d2l.ai/chapter_optimization/sgd.html)
- **Optimization Algorithms Overview (Ruder)**  
<https://ruder.io/optimizing-gradient-descent/>
- **Regularization Methods (ML Cheatsheet)**  
<https://ml-cheatsheet.readthedocs.io/en/latest/regularization.html>

- **Early Stopping (Keras Documentation)**  
[https://keras.io/api/callbacks/early\\_stopping/](https://keras.io/api/callbacks/early_stopping/)
  - **Dropout (Original Paper)**  
<https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf>
  - **Weight Initialization (Xavier / Kaiming Overview)**  
<https://amansari.info/primers/initialization>
  - **Batch Normalization (Primer Explanation)**  
<https://amansari.info/primers/batchnorm>
  - **Autoencoders (Deep Learning Book)**  
<https://www.deeplearningbook.org/contents/autoencoders.html>
- 

## SECTION 3 — Computer Vision

### Vision Fundamentals

- **Convolutional Neural Networks (CS231n Chapter)**  
<https://cs231n.github.io/convolutional-networks/>
- **Pooling Layers (D2L Chapter)**  
[https://d2l.ai/chapter\\_convolutional-neural-networks/pooling.html](https://d2l.ai/chapter_convolutional-neural-networks/pooling.html)
- **Image Classification (PyTorch CIFAR Tutorial)**  
[https://pytorch.org/tutorials/beginner/blitz/cifar10\\_tutorial.html](https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html)

### Object Detection & Segmentation

- **YOLO / RCNN / SSD Overview (CS7015 Lecture Video)**  
<https://www.youtube.com/watch?v=Nk5d-36iZQE>
- **U-Net Paper (Segmentation)**  
<https://arxiv.org/abs/1505.04597>

### Transfer Learning & Features

- **Transfer Learning in PyTorch**  
[https://pytorch.org/tutorials/beginner/transfer\\_learning\\_tutorial.html](https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html)
- **Image Augmentation with torchvision**  
<https://pytorch.org/vision/stable/transforms.html>

## Generative & Self-Supervised Vision

- **DCGAN Tutorial (PyTorch)**  
[https://pytorch.org/tutorials/beginner/dcgan\\_faces\\_tutorial.html](https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html)
  - **SimCLR Self-Supervised Learning Paper**  
<https://arxiv.org/abs/2002.05709>
  - **Vision Transformers (ViT Overview)**  
<https://towardsdatascience.com/the-vision-transformer-vit-explained-5f5a9fd4b0>
  - **CLIP (OpenAI Overview)**  
<https://openai.com/research/clip>
  - **Stable Diffusion (HuggingFace Blog)**  
[https://huggingface.co/blog/stable\\_diffusion](https://huggingface.co/blog/stable_diffusion)
- 

## SECTION 4 — NLP & LLMs

### Foundations

- **Word Embeddings (CS224n Notes)**  
<https://web.stanford.edu/class/cs224n/readings/cs224n-2019-notes1.pdf>
- **Illustrated Transformer (Jay Alammar)**  
<https://jalammar.github.io/illustrated-transformer/>
- **Text Classification (Scikit-Learn Guide)**  
[https://scikit-learn.org/stable/tutorial/text\\_analytics/working\\_with\\_text\\_data.html](https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html)

## Transformers & Pre-trained Models

- **HuggingFace Transformers Documentation**  
<https://huggingface.co/docs/transformers/index>
- **HuggingFace QA Tutorial**  
[https://huggingface.co/docs/transformers/tasks/question\\_answering](https://huggingface.co/docs/transformers/tasks/question_answering)

## LLM Core Concepts

- **A Hacker's Guide to Language Models**  
<https://newsletter.pragmaticengineer.com/p/hackers-guide>
- **1-Hour Introduction to LLMs (Karpathy)**  
[https://www.youtube.com/watch?v=zjkBMFhNj\\_g](https://www.youtube.com/watch?v=zjkBMFhNj_g)
- **"Let's Build GPT From Scratch" (Karpathy Video)**  
<https://youtu.be/kCc8FmEb1nY>

## Fine-Tuning & Agents

- **PEFT / LoRA (HuggingFace PEFT Library)**  
<https://huggingface.co/docs/peft/index>
  - **LLM Agents Primer**  
<https://amansari.info/primers/agents>
- 

## SECTION 5 — Audio & Speech

### Audio Processing & Speech Recognition

- **TorchAudio Documentation**  
<https://pytorch.org/audio/stable/index.html>
- **HuggingFace Audio Course (Wav2Vec2, HuBERT, Whisper)**  
<https://huggingface.co/learn/audio-course>



## Pre-trained Audio Models

- **HuBERT Paper (Self-Supervised Speech Representation Learning)**  
<https://arxiv.org/abs/2106.07447>
- **Whisper (OpenAI Speech Recognition)**  
<https://openai.com/research/whisper>
- **Whisper GitHub Implementation**  
<https://github.com/openai/whisper>
- **Qwen-Audio Model Page**  
<https://huggingface.co/Qwen/Qwen-Audio>

## Practical Audio Modeling

- **Audio Classification Tutorial (PyTorch)**  
[https://pytorch.org/tutorials/beginner/audio\\_classifier\\_tutorial.html](https://pytorch.org/tutorials/beginner/audio_classifier_tutorial.html)
- **Speech-to-Text Task Guide (HF Transformers)**  
[https://huggingface.co/docs/transformers/tasks/speech\\_recognition](https://huggingface.co/docs/transformers/tasks/speech_recognition)
- **Text-to-Speech Task Guide (HF Transformers)**  
[https://huggingface.co/docs/transformers/tasks/text\\_to\\_speech](https://huggingface.co/docs/transformers/tasks/text_to_speech)