INTELLIGENCE · TAUGHT WITH PYTORCH

Deep Learning

Neural networks from the training loop up. You write PyTorch by hand, fine-tune pretrained models on hardware you actually have, and read the loss curve when it goes wrong.

Foundational | 5 phases | 24 sessions | 36 contact hours

Syllabus

5 phases, 24 sessions. Each phase ends in something you have built.

1 Before Anything Learns

Get a real model running, then learn to read the numbers going through it.

5 sessions
$58 alone

1.1 Run a Model First

Load a pretrained network, classify a photo you took, and see the whole thing work before any theory.

FREE PREVIEW

1.2 Just Enough Python

Functions, loops, lists, and the imports every notebook in this course starts with.

1.3 Tensors as Containers

Shapes, dtypes, devices, indexing, and reading an error that says it expected 4 dimensions.

1.4 Matmul Without Linear Algebra

Weighted sums, broadcasting, and the two rules that decide whether your shapes line up.

1.5 Where Your Compute Comes From

Kaggle's 30 weekly GPU hours, Colab's unpublished limits, and sizing a job to fit inside them.

By the end: **A notebook that classifies your own photos with a pretrained model and explains every shape in it.**

2 The Training Loop

Write the loop that turns data into weights, then work out why it isn't learning.

6 sessions
$58 alone

2.1	**Data In, Batches Out** Dataset, DataLoader, transforms, and why most training bugs turn out to live in here.
2.2	**Autograd** The graph PyTorch builds behind your back, and what .backward() actually fills in.
2.3	**Your First Training Loop** nn.Module, logits, loss, optimizer, zero_grad: the five lines you will write forever.
2.4	**Making It Learn** Learning rates, schedules, warmup, and what to do about a loss curve that has gone flat.
2.5	**Overfitting & Regularization** The train and validation gap, dropout, weight decay, augmentation, and knowing when to stop.
2.6	**The Diagnosis Checklist** Loss at init, overfit one batch, train versus eval mode, and the four mistakes everyone makes.

By the end: **A classifier you trained from scratch, plus a written diagnosis of a run you deliberately broke.**

3 Architectures That Earned It

Match an architecture to the shape of your data, and build the one that won.

5 sessions
$58 alone

3.1	**Convolutional Networks** Local connectivity, parameter sharing, and what a filter actually learns to look for.
3.2	**Why Recurrence Lost** One unrolled RNN, the hidden-state bottleneck, and the problem attention was invented to fix.
3.3	**Attention From Scratch** Q, K, V, scaling, causal masks, and multi-head written out by hand exactly once.
3.4	**The Transformer Block** Residuals, LayerNorm, position, and stacking blocks until text comes out the other end.
3.5	**Tokenizers & a Tiny Language Model** Byte-level tokenization and a 10M-parameter model you train end to end on a free T4.

By the end: **A character-level transformer you wrote by hand and trained inside one free GPU session.**

4 Real Data, Borrowed Weights

Stop training from scratch and get a pretrained model working on data you collected yourself.

4 sessions
$58 alone

4.1	**Transfer Learning**	Pretrained backbones, what to freeze, and why you will rarely train from scratch again.
4.2	**Building Your Own Dataset**	Collect, label, split honestly, and find the leak before it flatters your numbers.
4.3	**Fine-Tuning on One Free GPU**	bf16 autocast, gradient accumulation, and fitting the run inside 16GB and one session.
4.4	**Evaluating Honestly**	Per-class error, a test set you touch once, and actually looking at what the model got wrong.

By the end: **A fine-tuned model on your own dataset, with a sealed test set and a written error analysis.**

5 Ship It

Make training reproducible and inference cheap enough for somebody else to run.

4 sessions
$58 alone

5.1	**Speed & Memory**	Profile first, then torch.compile and bf16, and measure whether anything actually got faster.
5.2	**Export**	torch.export instead of the dead TorchScript path, ONNX via dynamo, and a file that runs without your code.
5.3	**Quantization**	int8 and int4 with torchao, the accuracy you trade away, and hitting a latency number.
5.4	**Reproducible Runs**	Seeds, config files, tracked experiments, and a result someone else can reproduce next month.

By the end: **An exported, quantized model inside a stated latency budget, from a run a classmate can rerun.**

Tools you will use

PyTorch 2.13 · torch.compile · torch.export · TorchVision · Hugging Face Transformers · torchao · Weights & Biases · Kaggle / Colab

What you will build

1. **Image Classifier From Scratch**
   A CNN you wrote, trained on a free T4, and debugged
2. **Tiny Transformer**
   Char-level model, attention by hand, one free GPU session
3. **Shipped Model**
   Fine-tuned, quantized with torchao, exported, under budget

Registration

Phase 1: Before Anything Learns	$58
Phase 2: The Training Loop	$58
Phase 3: Architectures That Earned It	$58
Phase 4: Real Data, Borrowed Weights	$58
Phase 5: Ship It	$58
Whole track, all 5 phases	**$240**

Buying every phase separately costs $290, so the whole track saves $50.

No payment is taken online. Registering creates an order and payment instructions are sent with it. Access opens once payment is confirmed.