Deep Learning

Guide to AI and Data Courses · Part I: Core

Representation learning | Convolutional networks | Recurrent networks | Transformers | Training techniques | Neighboring courses

Sercan KÜLCÜ · Giresun University · Computer Engineering

© Sercan Külcü. All rights reserved. · Interactive HTML version of these slides: https://sercankulcu.github.io/files/ai_courses/slides/Chapter_05_Deep_Learning.html

What Question Does It Ask?

> **How do many-layered networks learn useful representations from raw data?**

Deep learning studies how multilayer neural networks learn useful intermediate representations from raw data (pixels, audio, text) on their own, layer by layer. The key idea is **representation learning** instead of hand-designed features; big data, GPUs and good architectures make it possible.

At a glance

APPROACH multilayer representation learning	**INPUT** raw data: images, text, audio	**OUTPUT** learned representation and prediction
SUCCESS MEASURE accuracy, loss, scaling	**MATH LOAD** high: derivatives, matrices, probability	**DATA NEEDS** large; needs GPUs

A Short History

1997
Hochreiter and Schmidhuber propose the LSTM: a cure for vanishing gradients on long sequences.

1998
LeCun et al.'s LeNet-5 convolutional network recognizes handwritten digits.

2006
Hinton et al. train deep belief networks with layer-wise pretraining; interest revives.

2012
AlexNet, a convolutional network trained on GPUs, wins ImageNet by a wide margin.

2015
He et al.'s residual networks (ResNet) win ImageNet with 152 layers.

2017
Vaswani et al. introduce the transformer architecture: attention only.

THE COURSE

Core Concepts

Representation learning
Layers learn ever more abstract features, from edges to objects.

Convolution
Small filters slide over the image; weights are shared.

Pooling
Shrinks spatial size, adds robustness to small shifts.

Recurrent network
RNN and LSTM: process a sequence step by step, carrying a state.

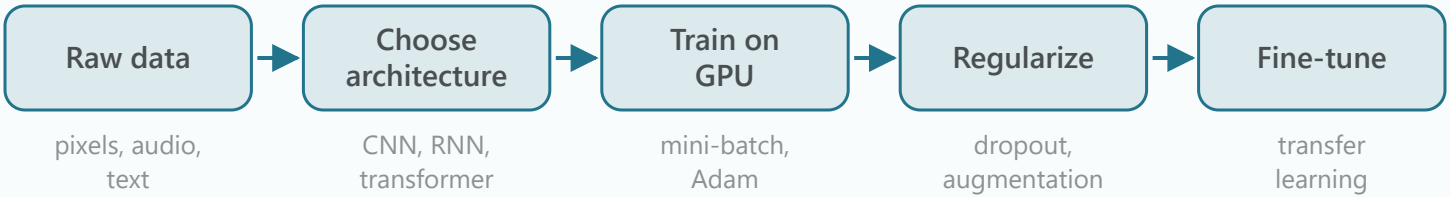
Attention
Each element weighs which other elements of the sequence to look at.

Residual connection
A path that skips a layer; makes very deep networks trainable.

Normalization
Batch and layer normalization stabilize training.

Transfer learning
Adapting a model trained on big data to a new task.

A Typical Workflow



Example: An app that recognizes plant diseases collects leaf photos, takes a ResNet pretrained on ImageNet, replaces its last layer with the disease classes and fine-tunes it on a GPU with data augmentation. No feature is defined by hand.

THE COURSE

What Does the Course Cover?

1. Deep feedforward networks and computational graphs
2. Optimization: SGD, momentum, Adam
3. Regularization: dropout, weight decay, augmentation
4. Batch and layer normalization
5. Convolutional networks: convolution and pooling
6. CNN architectures: LeNet, AlexNet, VGG, ResNet
7. Transfer learning and fine-tuning
8. Recurrent networks: RNN, LSTM, GRU
9. Sequence-to-sequence models and attention
10. Transformers and self-attention
11. Autoencoders and self-supervised learning
12. Interpreting deep networks: saliency maps

Skills you will gain

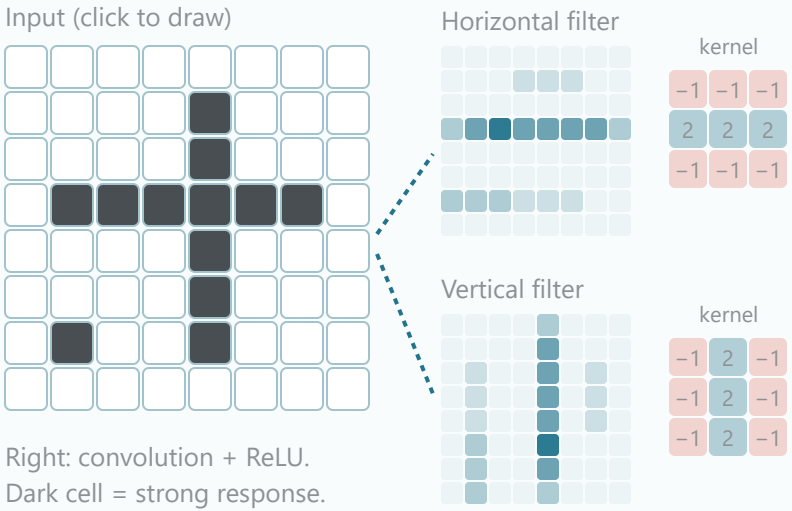
building and training models in PyTorch | GPU training and mini-batch tuning | fine-tuning pretrained models | monitoring training with TensorBoard | using Hugging Face models | designing hyperparameter experiments

A typical semester; the order and coverage vary between universities.

TRY IT

How Do Filters See a Line?

A school app will recognize signs drawn by children. Click the left grid to draw; two convolution filters scan the picture, and class scores come out of the feature maps.



Right: convolution + ReLU.
Dark cell = strong response.

Horizontal | Vertical | Plus | Clear

Horizontal map sum **42**, vertical **42**.
Prediction: **plus** (95%)

Class probabilities (softmax)

horizontal	2%
vertical	2%
plus	95%

Draw a diagonal line: both filters **barely respond**. The network needs other filters for other patterns.

The nine numbers in each filter were written by hand; a deep network **learns them from data** and, by stacking layers, climbs from edges to shapes and from shapes to objects.

CONTEXT

Prerequisites

Courses worth knowing before this one, and why they are needed.

Course	Why it is needed
Artificial Neural Networks	multilayer networks, backpropagation, activation functions
GPU Programming and Parallel Computing	parallel training of large networks, tensor operations
Calculus	multivariable derivatives and the chain rule
Programming	comfort with Python, NumPy and a deep learning library

CONTEXT

Are You Ready for This Course?

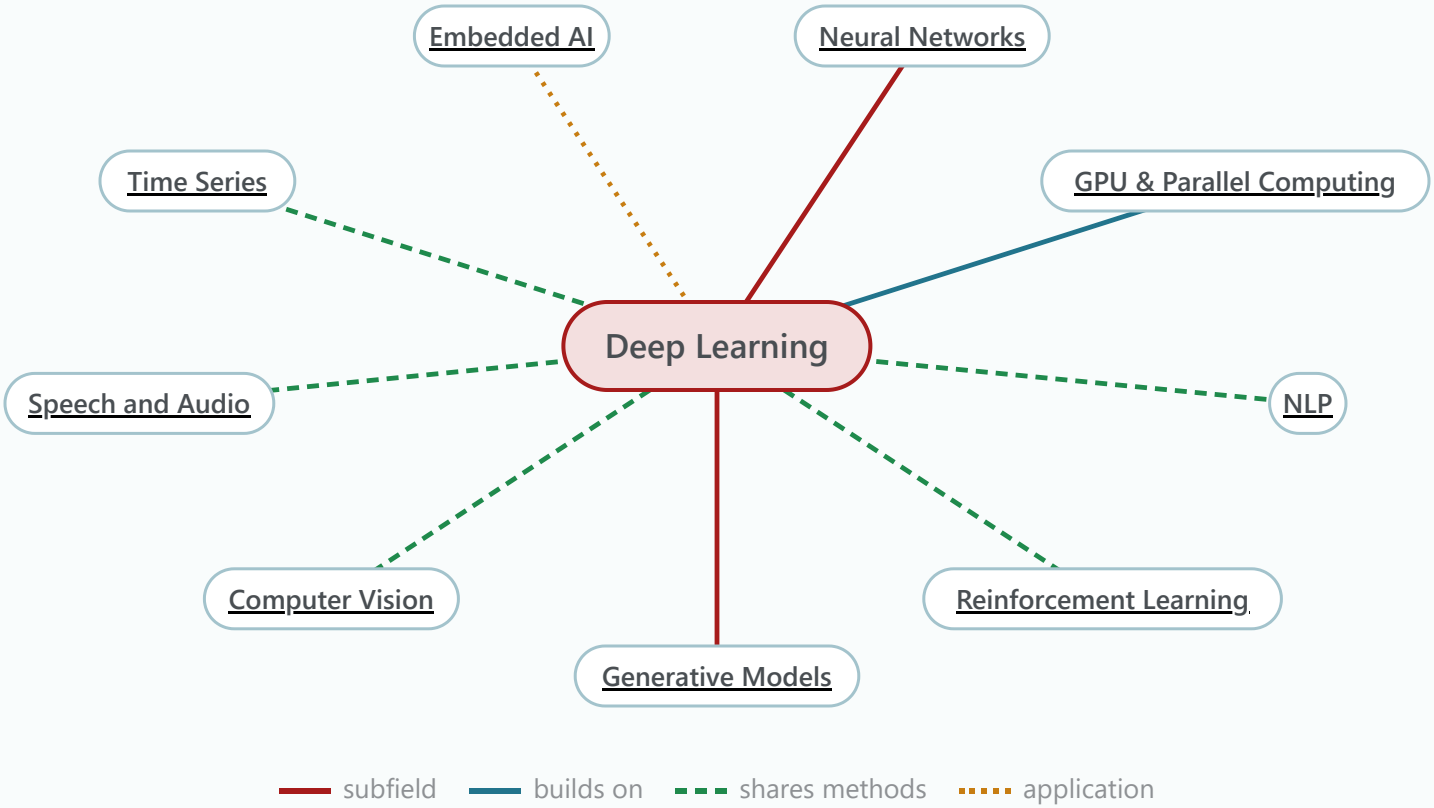
Ask yourself; for every "no", the right-hand column shows where to start.

Ask yourself	If not, start with
Can you explain backpropagation step by step in a small network?	Artificial Neural Networks
Can you compute the gradient of a multivariable function?	Partial derivatives and the chain rule from calculus
Can you say how shapes must match in a matrix product?	Linear Algebra
Can you write a class in Python and work with NumPy arrays?	An intro programming course, then a short NumPy tutorial

If you can say "yes" to most of these, you are ready; one or two gaps can be closed during the semester.

CONTEXT

Neighbouring Courses



Overlapping Courses (1/2)

Relation	Course	Shared ground
subfield	This course is a subfield of Artificial Neural Networks	multilayer networks, backpropagation, activation functions
subfield	Generative Models is a subfield of this course	GANs, VAEs, diffusion, large language models
builds on	This course builds on GPU Programming and Parallel Computing	parallel training of large networks, tensor operations
application	Embedded AI applies the methods of this course	quantization, pruning, small models
shares methods	Shares methods with Natural Language Processing and Text Mining	word embeddings, transformer architecture
shares methods	Shares methods with Reinforcement Learning	deep Q-networks, deep policy networks
shares methods	Shares methods with Computer Vision	convolutional neural networks, vision transformers
shares methods	Shares methods with Speech Recognition and Audio Processing	end-to-end speech recognition

Overlapping Courses (2/2)

Relation	Course	Shared ground
shares methods	Shares methods with Time Series Analysis and Forecasting	recurrent networks, attention-based forecasting

DIFFERENCES

Where It Differs (1/2)

Criterion	Deep Learning	Neural Networks	Machine Learning
Core question	How do many-layered networks learn useful representations from raw data?	How do networks of simple units learn by adjusting their weights?	How does a program generalize from examples?
Approach	multilayer representation learning	network of simple weighted units	statistical learning
Input	raw data: images, text, audio	numeric feature vector	labeled or unlabeled examples
Output	learned representation and prediction	class or numeric prediction	a model that makes predictions
Success measure	accuracy, loss, scaling	loss and accuracy	error on unseen data
Math load	high: derivatives, matrices, probability	medium-high: derivatives, matrices	high: linear algebra, probability, optimization
Data needs	large; needs GPUs	moderate	moderate to large

- **Neural Networks:** ANN builds neurons, loss and backpropagation on small networks; this course covers deep architectures and the practice of training them.
- **Machine Learning:** Classic ML works with hand-designed features; deep learning learns features from raw data by itself.

Where It Differs (2/2)

Criterion	Deep Learning	Computer Vision	Generative Models
Core question	How do many-layered networks learn useful representations from raw data?	How does a computer understand what is in an image or a video?	How does a model generate new samples that look like real data?
Approach	multilayer representation learning	extracting meaning from images	learning the data distribution
Input	raw data: images, text, audio	image or video	huge, mostly unlabeled data
Output	learned representation and prediction	label, box, depth, 3D structure	new samples: images, text, audio
Success measure	accuracy, loss, scaling	accuracy, mAP, IoU	realism and diversity (FID etc.)
Math load	high: derivatives, matrices, probability	high: geometry, linear algebra	high
Data needs	large; needs GPUs	many labeled images	very large

- **Computer Vision:** Computer vision is the problem of understanding images; deep learning is a toolkit shared by vision, language and speech.
- **Generative Models:** Generative models aim to sample from the data distribution; this course mostly focuses on discriminative, predictive networks.

PRACTICE

Applications

Image classification
Recognizing an object or a disease in a photo.

Speech recognition
Turning an audio recording into text.

Machine translation
Transformer-based translation systems.

Object detection
Finding pedestrians and vehicles for a self-driving car.

Protein structure
Predicting 3D structure from an amino acid sequence.

Game playing
Atari and Go with deep reinforcement learning.

PRACTICE

Risks and Ethics

Compute and energy cost

Training large models takes thousands of GPU hours and real carbon emissions, concentrating research in a few large organizations.

Adversarial examples

Imperceptible pixel changes fool a classifier; researchers showed the same with stickers placed on a stop sign.

Data scraped from the web

Web-scraped datasets such as ImageNet can contain photos taken without consent and offensive labels; ImageNet later removed many of them.

Illusory progress

Hyperparameters tuned on the test set and results reported from a single seed make small differences look like real gains.

In depth: Explainable AI and AI Ethics.

PRACTICE

Tools and Datasets

Tools and libraries

PyTorch | TensorFlow | Keras | JAX | Hugging Face | CUDA | TensorBoard

Well-known datasets

ImageNet | CIFAR-10/100 | MNIST | COCO | LibriSpeech | Wikipedia text

PRACTICE

Common Misconceptions

Common belief	In fact
Deep learning is best for every problem.	On small tabular data, gradient-boosted trees are often better and cheaper.
A deep network is a black box that cannot be inspected.	Tools such as gradient-based saliency maps and attention weights give partial explanations.
Deep learning was invented in 2012.	The ideas are decades old; big data and GPUs brought the 2012 leap.
More layers always mean better results.	Without residual connections and normalization, very deep networks may not train.

NEXT STEPS

After This Course

Courses that build on or extend this one

Generative Models | Embedded AI

Related careers

Deep learning engineer | Computer vision engineer | Researcher | ML infrastructure engineer | Applied scientist

Deep learning is the most widely used tool on the map today: computer vision, natural language processing, speech recognition, generative models and deep reinforcement learning are built on it. On the systems side, GPU programming, embedded AI and MLOps teach how to speed these models up and take them to production.

NEXT STEPS

Further Reading

- Goodfellow, Bengio and Courville, *Deep Learning*, MIT Press, 2016.
- Zhang, Lipton, Li and Smola, *Dive into Deep Learning*, Cambridge University Press, 2023.
- Prince, *Understanding Deep Learning*, MIT Press, 2023.
- Chollet, *Deep Learning with Python*, 2nd ed., Manning, 2021.
- LeCun, Bengio and Hinton, "Deep learning", *Nature* 521, 2015.

The End

Chapter 5: Deep Learning

← Previous chapter · Map of the field · Next chapter →