COMPANION HANDOUT

Foundations of AI

Montana AI Summit September 2026

Through-line: AI did not suddenly become smart in 2022. Capability followed data, compute, and a software stack that could use both.

A short timeline

1950 — Turing
Ask whether a machine can behave as if it thinks. A test instead of a soul.

1956 — Dartmouth
McCarthy names the field. The bet: intelligence described precisely enough to simula

1958 — Perceptron
Learning as weighted connections — then a proof that one layer is not enough.

1970s — First winter
Overpromise and combinatorial explosion. Funding retreats.

1980s — Expert systems
Useful in narrow domains, brittle at scale. A second winter follows.

1986–99 — Backprop + stats
Deep nets become trainable in principle. Speech and translation go probabilistic.

2012 — AlexNet
ImageNet + better algorithms + two consumer GPUs. Vision changes in a year.

2017+ — Transformer & scale
Attention makes language GPU-friendly. Pretrain once; adapt many times.

Five things to keep

01 A 70-year program, not a product.
Named in 1956, failed twice, returned when data and compute arrived.

02 Learn the function if you cannot write the rules.
Modern systems fit patterns from examples. Rules still help — they no longer carry everything.

03 GPUs won because the math is linear algebra.
Thousands of simple cores do matrix multiplies. CUDA turned a chip lead into a platfor

04 Software is layered on purpose.
CUDA → libraries → PyTorch → trainers → servers. Each layer buys speed and hides mise

05 The live constraints are physical.
Energy, memory bandwidth, interconnect, data quality, and people who can run the machines.

CubCloud AI

What's on a modern GPU

Tensor Cores
Mixed-precision matrix units (FP16, BF16, FP8, FP4). This is the throughput.

HBM
Stacked high-bandwidth memory. If weights live too far away, the cores starve.

Interconnect
NVLink in the rack; InfiniBand or Ethernet across racks. Big models are distributed computers.

The rack
The unit of AI is no longer a card. It is a cooled rack with a power budget.

The software stack

Apps & products	Chat, search, agents, copilots, science tools
Serving	vLLM, SGLang, TensorRT-LLM, Kubernetes, queues
Training frameworks	PyTorch, JAX, TorchTitan, DeepSpeed, Megatron
Libraries & compilers	cuDNN, NCCL, Transformer Engine, Triton, torch.compile
Runtime	CUDA / ROCm / XLA — talk to the device, allocate memory, launch kernels
Hardware	GPU / TPU / interconnect / HBM

Pocket glossary

Token
A chunk of text (often a word piece) the model reads and writes.

Parameter / weight
A number the model learned. Modern models have billions to trillions.

Training
Adjust weights so predictions match targets. A long distributed job.

Inference / serving
Use a trained model on new requests. Latency and cost per token matter.

Attention
Every token can look at every other token. Powerful; costly as context grows.

KV cache
Memory of past tokens during generation. Often what fills the GPU.

CUDA
NVIDIA's programming model. Why most research still starts on their chips.

Quantization
Store weights in fewer bits (8, 4) so more model fits and each token costs less.