Beyond MLPs: Modern Architectures

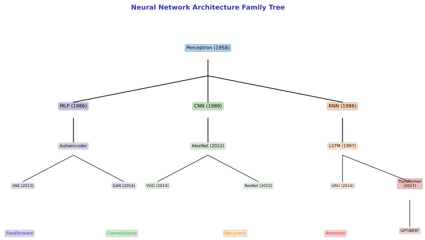
The MLP Foundation:

- Everything we learned applies to modern architectures
- Backpropagation: same algorithm
- Activation functions: same choices
- Regularization: same techniques

Key Modern Architectures:

1. **CNN**: Convolutional Neural Networks
2. **RNN/LSTM**: Recurrent Networks
3. **Transformer**: Attention-based

Common Thread:
All contain feedforward (MLP) components!



../.

architecture_family_tre

MLPs are the foundation for everything that followed

Convolutional Neural Networks

Key Idea: Learnable pattern detectors

- Convolutional filters slide over input
- Detect local patterns (edges, shapes)
- Weight sharing reduces parameters
- Hierarchical feature learning

For Time Series:

- 1D convolutions over time
- Detect patterns in price/volume sequences
- Filter learns what to look for
- E.g., "head and shoulders" pattern

Architecture:

Conv1D → ReLU → Pool

→ Conv1D → ReLU → Pool

→ Flatten → MLP → Output

Finance Use Cases:

- Technical pattern recognition
- Order book analysis
- Multi-asset correlation patterns

CNNs: Finding patterns with learnable filters

Recurrent Neural Networks

Key Idea: Memory for sequences

- Process sequences one step at a time
- Maintain hidden state (memory)
- Output depends on current + past inputs
- Natural for time series

RNN Update:

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b)$$

Problem: Vanishing gradients over long sequences

LSTM Solution (1997):

- Forget gate: what to discard
- Input gate: what to add
- Output gate: what to reveal
- Cell state: long-term memory

Finance Use Cases:

- Time series forecasting
- Sequence-to-sequence (prices)
- Combining with attention

RNNs and LSTMs: Designed for sequences

Transformers and Attention

Key Innovation: Self-attention

- Each position attends to all others
- No recurrence needed
- Parallelizable (fast training)
- Captures long-range dependencies

Components:

- Multi-head attention
- Feedforward layers (MLPs!)
- Layer normalization
- Positional encoding

Attention Formula:

$$\text{Attn} = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Finance Use Cases:

- News/sentiment analysis (NLP)
- Document understanding
- Multi-asset attention
- Temporal attention for prices

The architecture behind GPT and modern NLP

RAG Formula: A Concrete Example

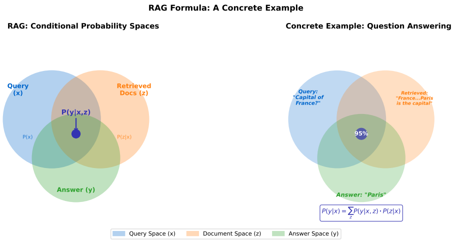
Retrieval-Augmented Generation (RAG):

- Combines retrieval with generation
- Query x retrieves relevant documents z
- Model generates answer y conditioned on both

The RAG Formula:

$$P(y|x) = \sum_z P(y|x,z) \cdot P(z|x)$$

- $P(z|x)$: Retriever finds relevant docs
- $P(y|x,z)$: Generator produces answer
- Marginalizes over retrieved documents



../../quantlet_to

Finance Application: Retrieve relevant filings/news, then generate analysis conditioned on context.

The MLP Foundation

Every Modern Architecture Contains MLPs:

CNN:

- Conv layers: shared MLPs
- Final classifier: MLP
- Same activation functions

RNN/LSTM:

- Gate computations: MLPs
- Output layer: MLP
- Same backprop algorithm

Transformer:

- FFN after attention: MLP
- Position-wise: MLP
- 2/3 of parameters in MLPs!

What you learned in this course is the foundation for all of deep learning.

Perceptron → MLP → CNN/RNN/Transformer

Every modern architecture contains feedforward components

Discussion Question

"If you were building a financial AI startup today,

what architecture and problem would you focus on?"

- Direct price prediction vs risk management?
- Traditional features vs alternative data?
- Simple MLP vs complex Transformer?
- Retail product vs institutional tool?

Think-Pair-Share: 3 minutes

The Black Box Problem

The Interpretability Challenge:

- Neural networks: millions of parameters
- No simple explanation for decisions
- "Why did you sell?" - "Because weight 47,823 was 0.0032"

Why This Matters in Finance:

- Regulatory requirements (explainability)
- Risk management needs understanding
- Client trust requires explanation
- Debugging requires insight

Partial Solutions:

- SHAP values, LIME
- Attention visualization
- Simpler models where possible

Trade-off:

Simple	Complex
Interpretable	Black box
Linear	Non-linear
Stable	May overfit
Lower accuracy	Higher accuracy

Question:
Is 1% more accuracy worth losing all interpretability?

Neural networks are often difficult to interpret

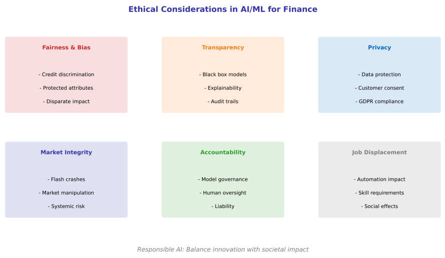
Key Regulations:

- **MiFID II** (EU): Best execution, transparency
- **GDPR**: Right to explanation for automated decisions
- **SR 11-7** (US): Model risk management
- **Basel III**: Capital requirements, risk models

Explainability Mandates:

- Credit decisions must be explainable
- Trading algorithms need documentation
- Model validation required
- Audit trails essential

Trend: Increasing regulation of algorithmic decision-making



../.

ethical_consideration

Regulations increasingly demand explainable AI

Systemic Risk from Correlated AI

What if everyone uses similar models?

The Problem:

- Similar training data
- Similar architectures
- Similar features
- $\Rightarrow$ Similar predictions
- $\Rightarrow$ Correlated trades
- $\Rightarrow$ Amplified market moves

Historical Example:

- August 2007: Quant meltdown
- Many funds used similar strategies
- All deleveraged simultaneously
- Massive losses in days

Flash Crash Risk:

- May 6, 2010: Dow dropped 1000 points in minutes
- Algorithmic trading implicated
- Feedback loops between systems

Mitigations:

- Circuit breakers
- Position limits
- Diversity requirements
- Human oversight
- Stress testing for crowding

Correlated AI trading could amplify market instability

Model Risk Management

"All models are wrong, some are useful" - George Box

Types of Model Risk:

- **Specification risk**: Wrong model type
- **Implementation risk**: Coding bugs
- **Data risk**: Bad inputs
- **Usage risk**: Misapplication

Famous Failures:

- LTCM (1998): Model assumptions failed
- Knight Capital (2012): $440M in 45 minutes
- London Whale (2012): VAR model issues

Governance Framework:

- Independent model validation
- Documentation requirements
- Regular backtesting
- Stress testing
- Change management
- Clear ownership

Key Principle:
Never deploy a model you don't understand well enough to know when it might fail.

Responsible deployment requires proper oversight

Historical Lesson: Responsible Innovation

AI Has Seen Hype Cycles Before:

The Pattern:

1. Breakthrough discovery
2. Excessive optimism/funding
3. Over-promising
4. Failure to deliver
5. "AI Winter" backlash
6. Quiet progress
7. Next breakthrough...

Examples:

- 1960s: "Machines will think in 20 years"
- 1980s: Expert systems will replace experts
- 2010s: "Deep learning solves everything"
- Today: "AGI is imminent"

Lesson:
Hype damages the field. Responsible claims and honest assessment help it grow sustainably.

Your Responsibility: Be honest about what neural networks can and cannot do.

Hype cycles damage the field; responsible claims help it grow

Where Neural Networks Add Value

Realistic Assessment of Neural Networks in Finance:

High Value Applications:

- **Risk Management**
  - Fraud detection
  - Credit scoring
  - Anomaly detection
- **Alternative Data**
  - Satellite imagery analysis
  - News sentiment
  - Social media signals
- **Execution**
  - Optimal order routing
  - Market making
  - Transaction cost analysis

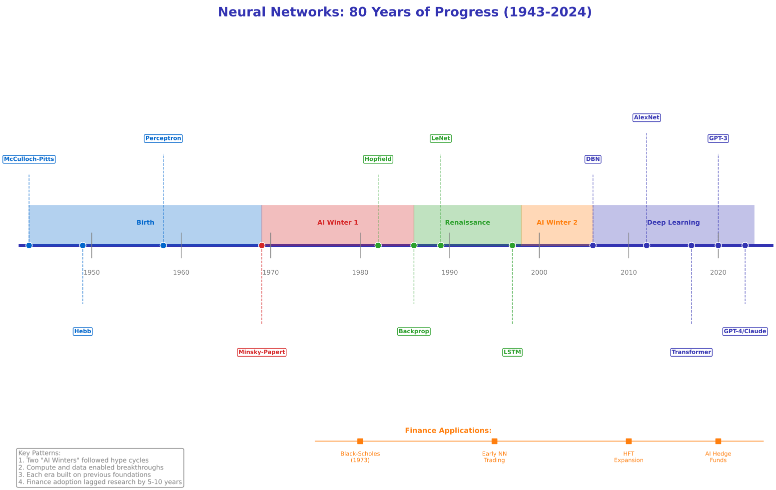
Lower Value (Often Overhyped):

- Direct price prediction
- "AI-powered" retail trading apps
- Fully automated strategies
- Complex models on limited data

Key Insight:
Neural networks work best when:

- Abundant data available
- Clear signal exists
- Domain expertise integrated
- Proper validation done

Risk management, alternative data, market making



full_timeline_1943_202

From McCulloch-Pitts to GPT: 80 years of progress

Course Summary: Key Takeaways

Module 1 - Perceptron:

- Neuron as weighted voting
- Linear separability limits
- XOR problem $\rightarrow$ AI Winter

Module 2 - MLPs:

- Hidden layers solve XOR
- Activation functions enable non-linearity
- Universal Approximation Theorem

Module 3 - Training:

- Gradient descent finds minimum
- Backprop: efficient gradient computation
- Overfitting is the main enemy

Module 4 - Practice:

- Regularization fights overfitting
- Financial data is uniquely challenging
- Honest assessment of capabilities

The Foundation: Everything in modern AI builds on these concepts.

The essential concepts from all four modules

Where to Go from Here

Suggested Learning Path:

Theory:

- Deep Learning (Goodfellow et al.)
- Neural Networks and Deep Learning (Nielsen) - free online
- Stanford CS231n (CNNs)
- Stanford CS224n (NLP)

Practice:

- PyTorch or TensorFlow tutorials
- Kaggle competitions
- Personal projects
- Open-source contributions

Finance-Specific:

- Advances in Financial ML (de Prado)
- Machine Learning for Asset Managers
- QuantConnect, Zipline (backtesting)
- Academic papers (SSRN, arXiv q-fin)

Key Advice:

- Build things!
- Start simple, add complexity
- Focus on fundamentals
- Be skeptical of claims
- Domain knowledge matters

Suggested resources for continued learning

Thank You

Neural Networks for Finance
BSc Lecture Series

Questions?

See Mathematical Appendix for full derivations