
Vibin
Visionaries

Understanding Neural Networks: From Questions to Intelligence

Exploring Key Concepts and Applications

01
Why neural networks ?

Why Do We Need Neural Networks?

Real world problems

They aren't linear, trying to solve them using only if and else is futile. Neural networks have the ability to learn non linear relationships.

Handling high dimensional data

Real data can have thousands, or millions of inputs, say pixels of an image. Neural networks scale well with large inputs

Neural networks can learn complex patterns

Neural networks can learn complex and non-linear patterns in data that are difficult or impossible to model using traditional algorithms.

Type something or '/' for commands

Neural networks as AI's backbone

Neural networks are the core of modern AI because they mimic how our brains learn, helping machines recognize patterns and solve complex problems Once trained, neural networks can make accurate predictions on unseen data by generalizing what they have learned from training examples.

02
What actually goes on under the hood of an ANN ?

How Does an ANN Actually Think?

Artificial neurons as weighted decision units

Artificial neurons act like decision-makers, weighing inputs differently to decide what’s important, helping the network learn patterns and make smarter predictions over time.

Structure: input, hidden, output layers

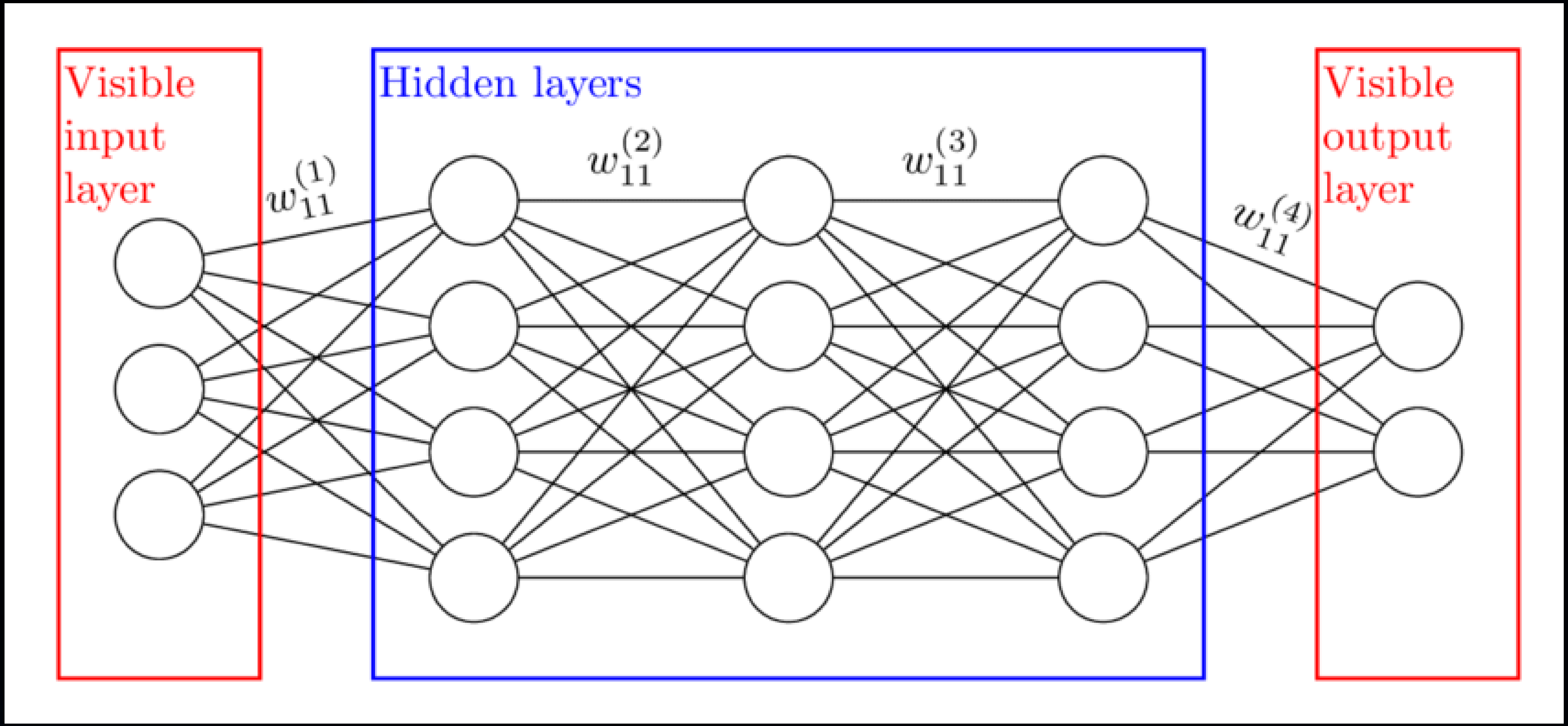
Neural networks think by passing info through layers: inputs receive data, hidden layers detect patterns, and output layers make decisions—kind of like teamwork in your brain!

Role of weights and activation functions

Weights adjust the strength of signals between neurons, while activation functions decide if a neuron “fires,” helping the network learn complex patterns and make smart decisions.

Learning process via error correction

Neural networks learn by tweaking themselves whenever they make a mistake, using error correction to get better at tasks step by step—kind of like how we learn from trial and error.



Type something or '/' for commands

Instances from Task

	precision	recall	f1-score	support
0.0	0.85	0.96	0.90	1607
1.0	0.67	0.31	0.42	393
accuracy			0.83	2000
macro avg	0.76	0.63	0.66	2000
weighted avg	0.81	0.83	0.81	2000

Classification report of predictions on churning data. An accuracy of 83.40% was obtained which is good for a simple ANN. Although some bias is seen for 0 class.

05

The Working Behind Convolutional Neural Network

The Pipeline behind CNN

1. Input and Layers

Convolutional Neural Networks → Excel at processing images, and visual inputs as these inputs are 2 dimensional and can be processed with ease using convolutions.

Core Idea: Extracting features → Learning from them → Making predictions

In order to make accurate predictions we first train our model on some data

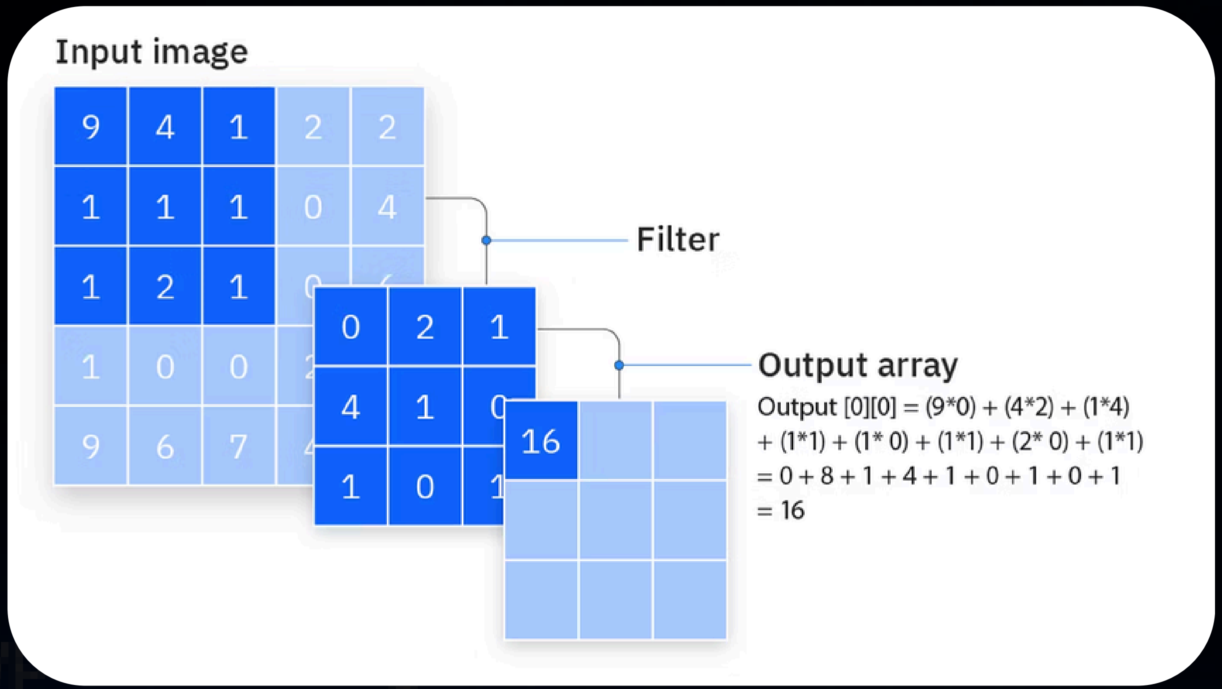
Data

- **Training Data :** The data which our model learns the patterns from.
- **Testing Data :** The data on which the model will predict after learning to get a perspective of our model

Layers in our network

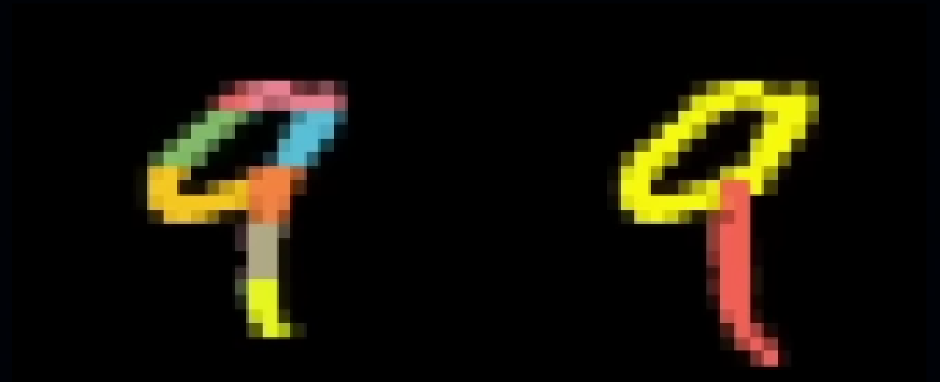
For processing images effectively our input goes through different layers. With these layers CNN increases complexity and identifies clearer features in the image. Lets look at them one by one.

Convolution Layer → **Pooling Layer** → **Full Connection Layer**



In this layer different kernels/filters are passed through the image to extract features like edges, faces etc.

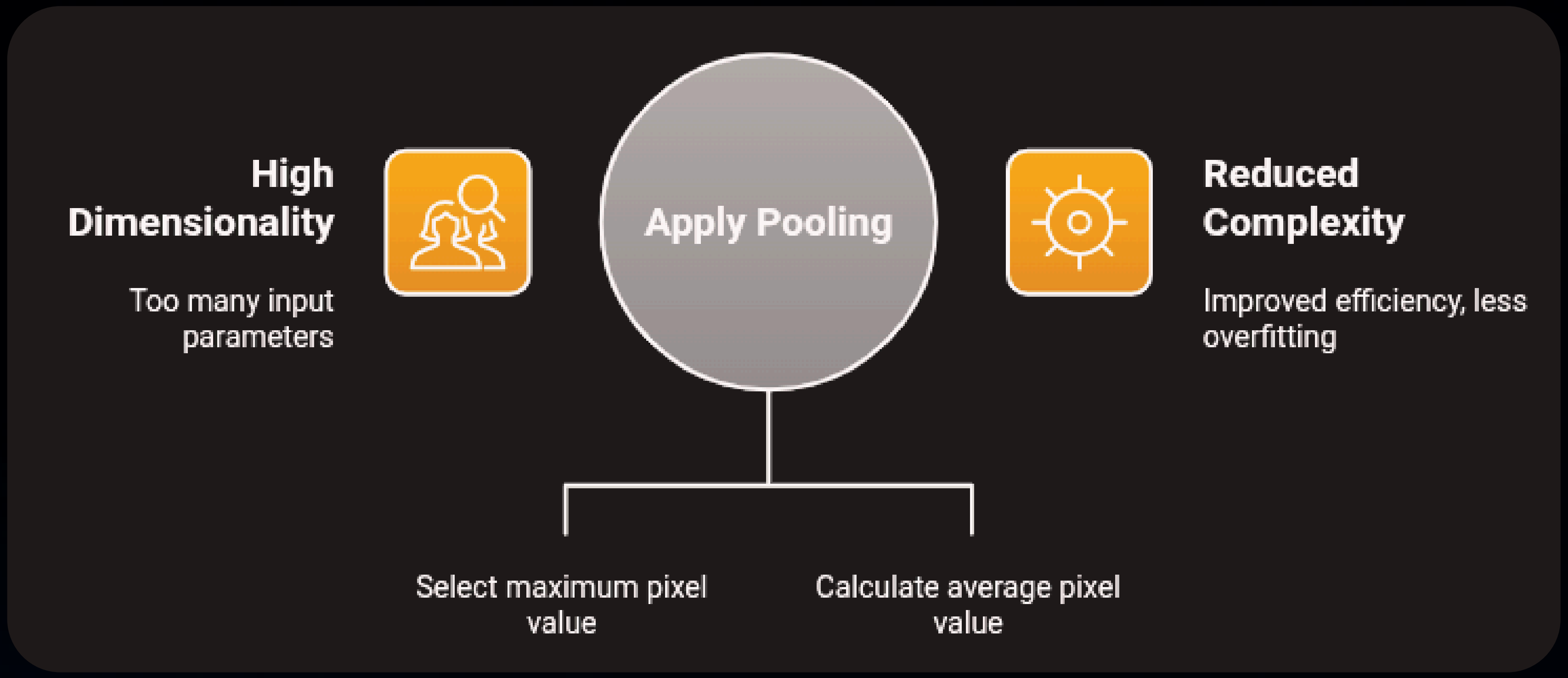
Additional convolutional layers derive a better picture of the features.



With additional layer **Without additional layer**

Layers in our network

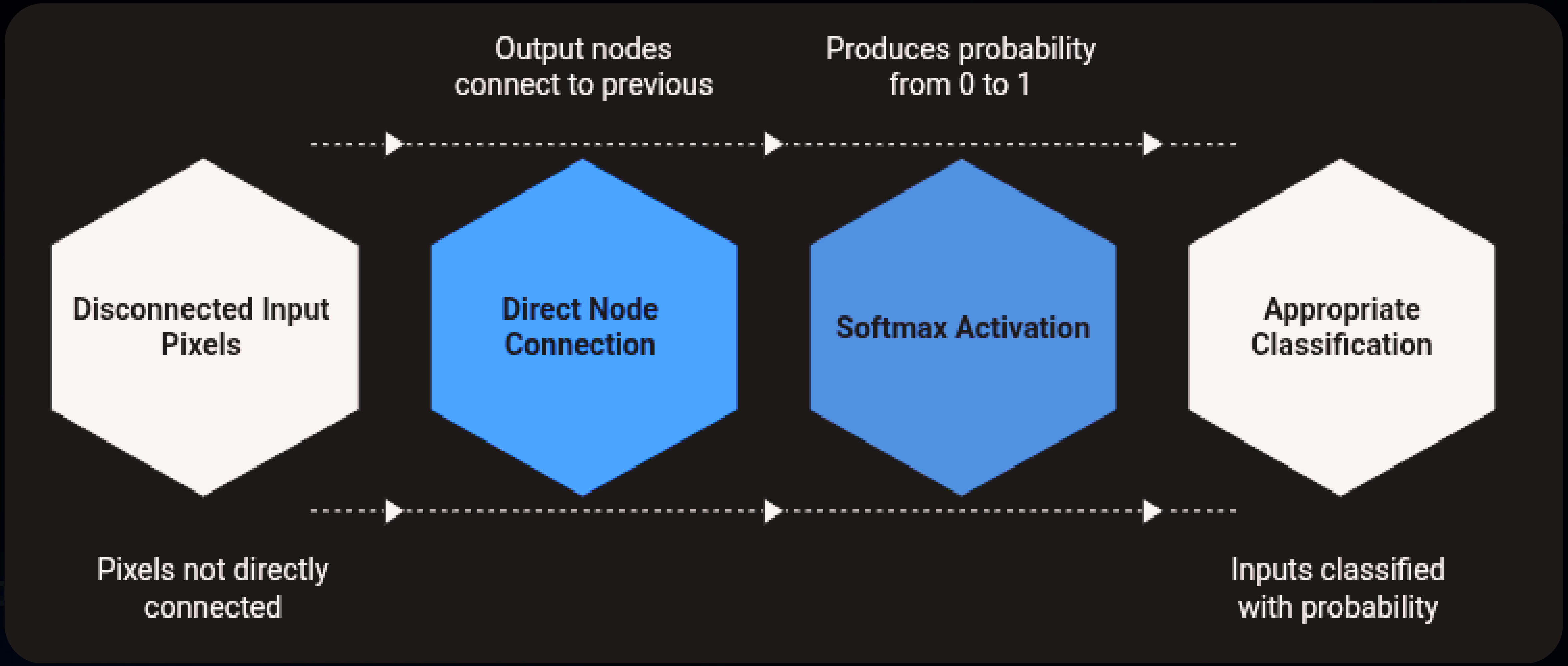
Convolution Layer → Pooling Layer → Full Connection Layer



After the extraction of features, we still have data with high dimensions. To improve efficiency of our network we introduce pooling which reduces dimensions and limits risk of overfitting.

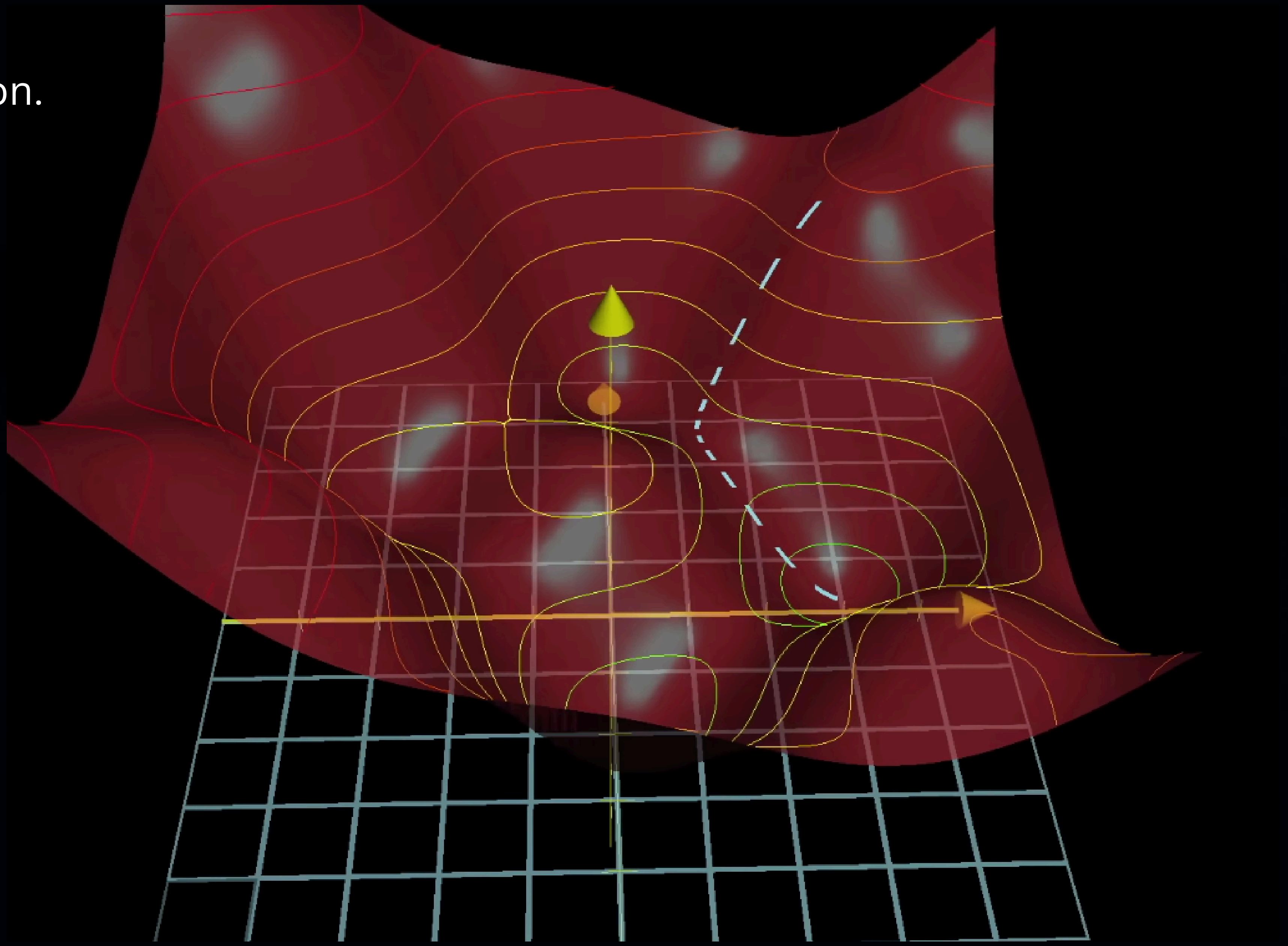
Layers in our network

Convolution Layer → Pooling Layer → Full Connection Layer



Minimizing Loss and Optimizing

The model is then trained on the training data several times. Each epoch the data has to back propagate and minimize the cost function. Minimizing the loss function is handled by an optimizer.

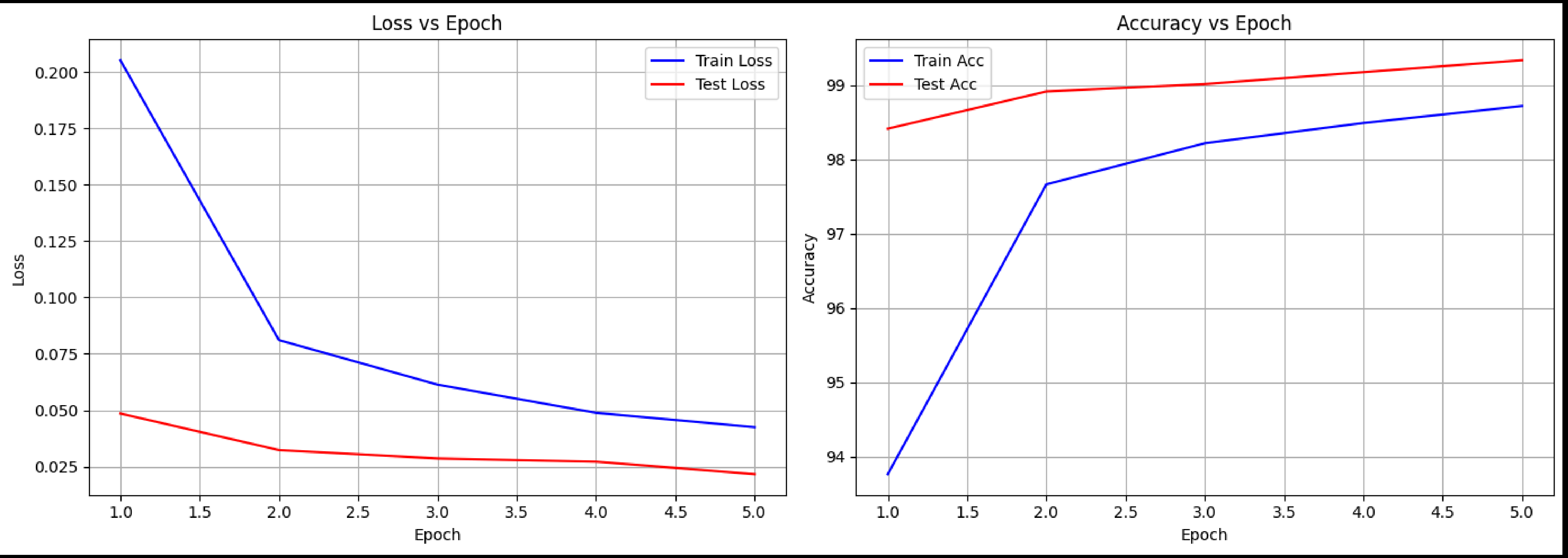


Criteria for Loss Function: CrossEntropy loss. Why?
Optimizer: Adam optimizer.

This increases our model's accuracy every epoch and we obtain better predictions

Type something or '/' for commands

Hyperparameter Tuning

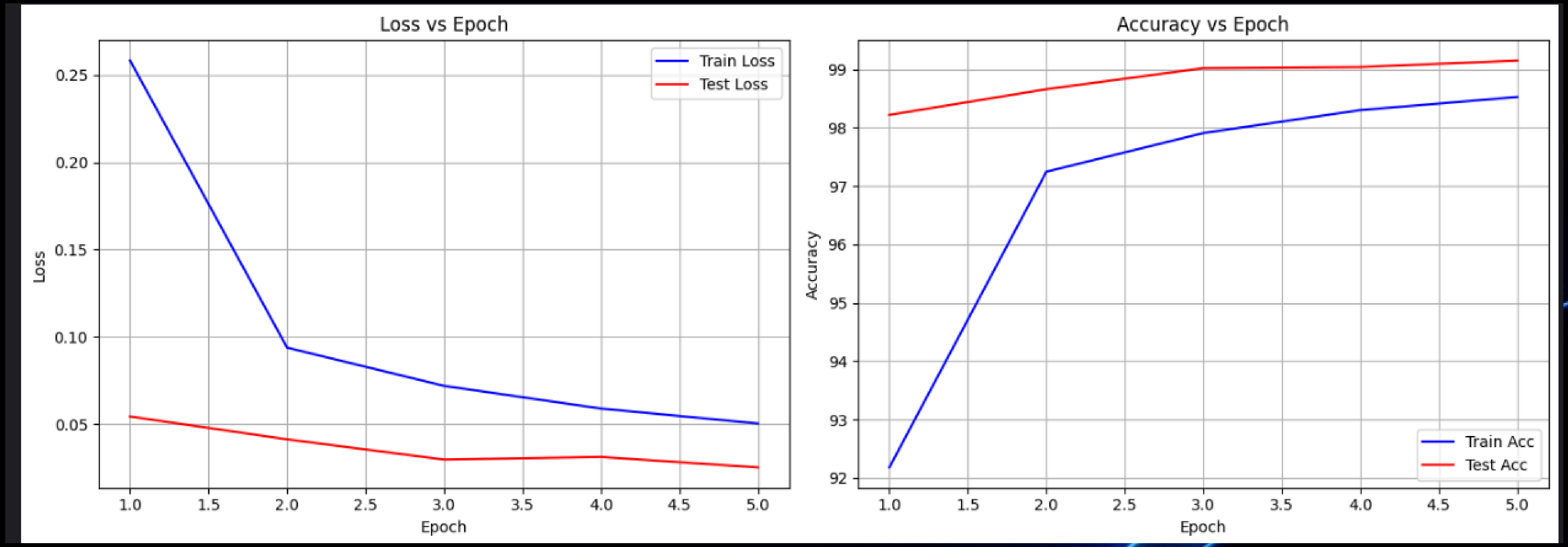


Max Pooling

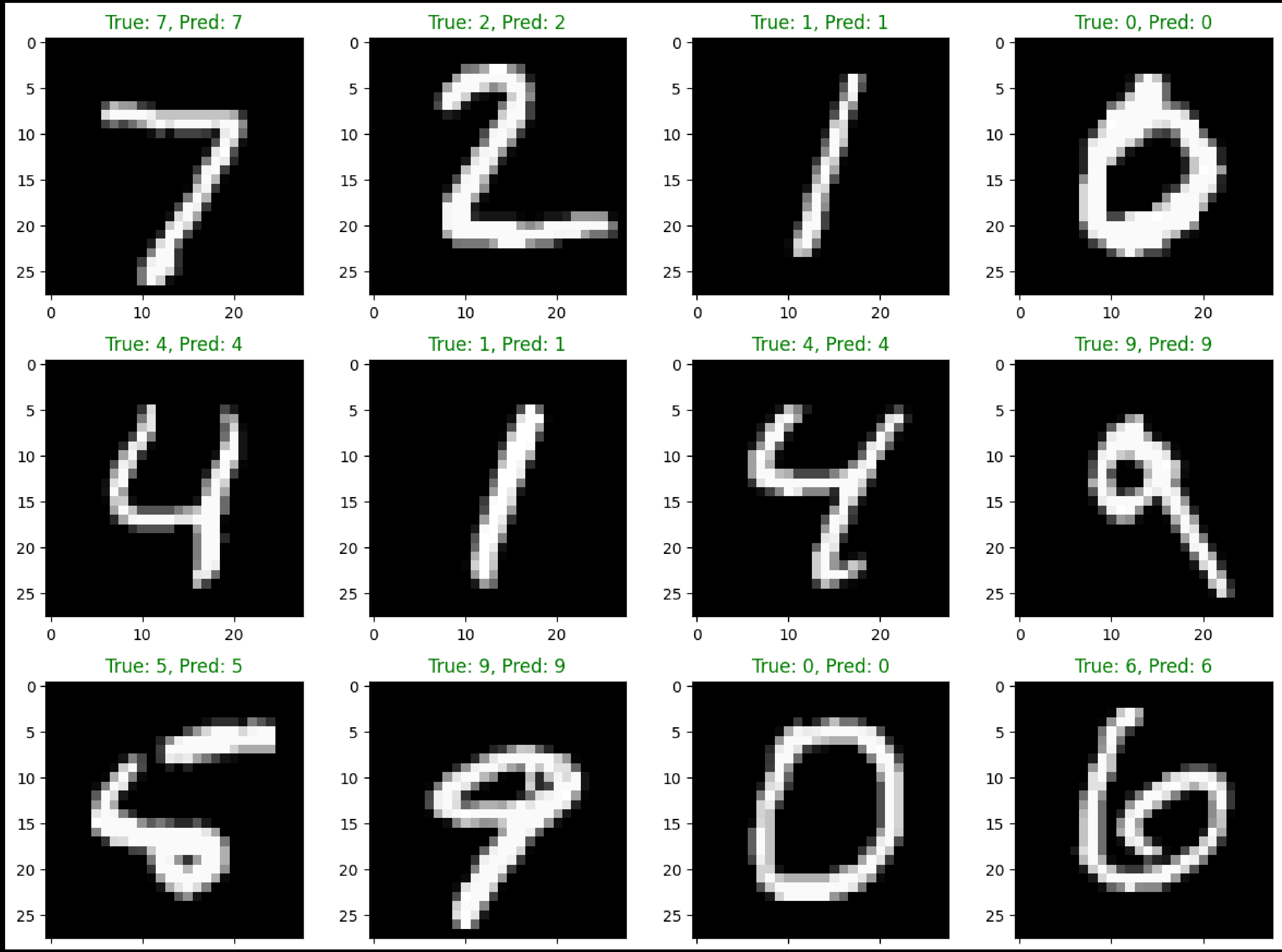
The graph shows amazing results. An overall accuracy of around 99.33% is observed.

Average Pooling

The graph shows good results but not as good as max pooling. An overall accuracy of around 99.15% is observed.



Results



We can see that the model gives true positive results for almost all the test data.

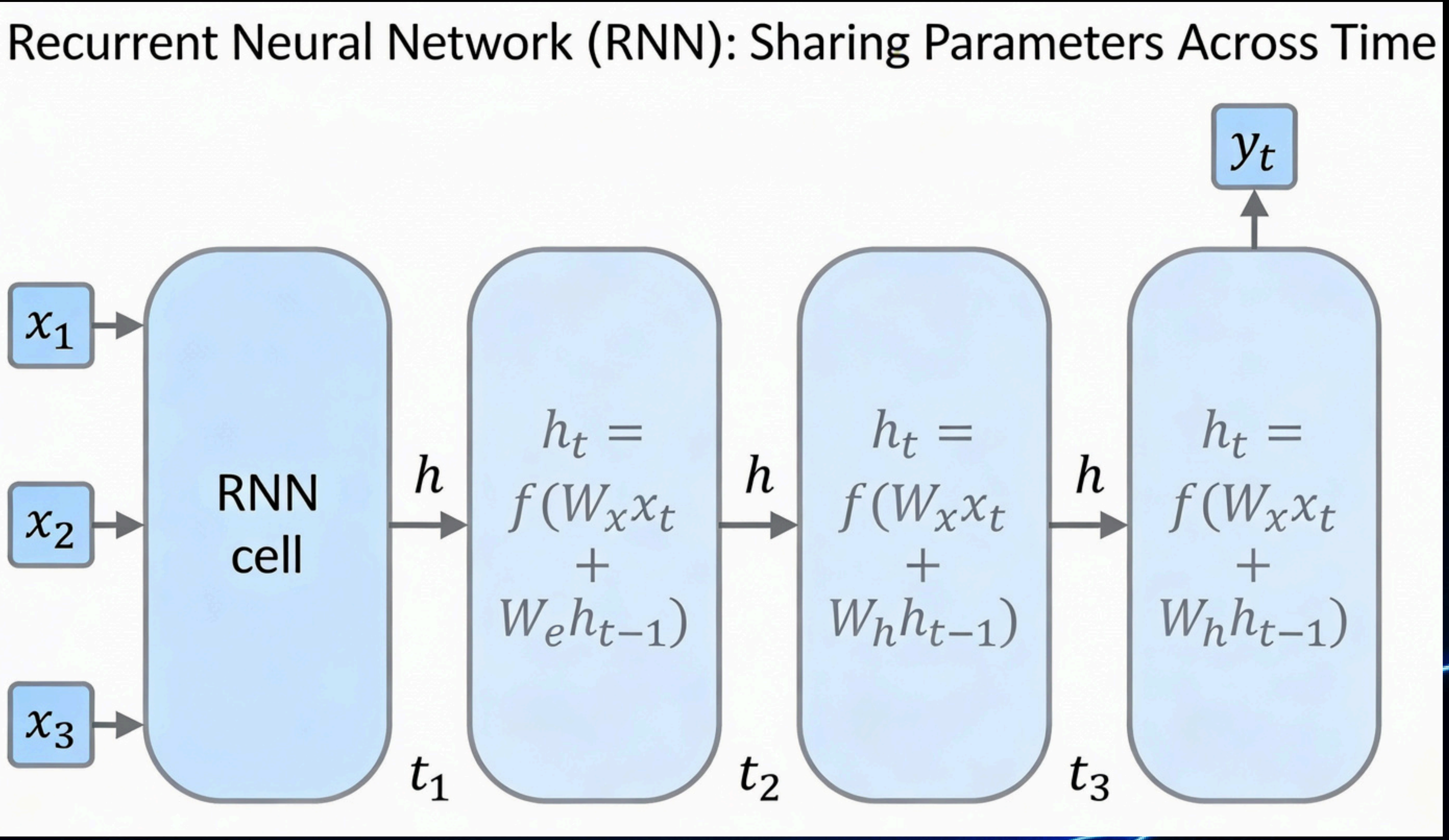
Overall Accuracy: 99.33%

06
Memory and Sequence Learning in Neural Networks

RNNs: Learning from Sequences

Designed specifically for sequential data

RNNs are built to handle sequences like sentences or time series by remembering past info(which could not be done by ANNs), helping the network learn patterns over time and make smarter predictions.



Maintaining memory of past inputs

RNNs keep track of past inputs by passing information through hidden states, helping them remember context over time—kind of like having a short-term memory for sequences.

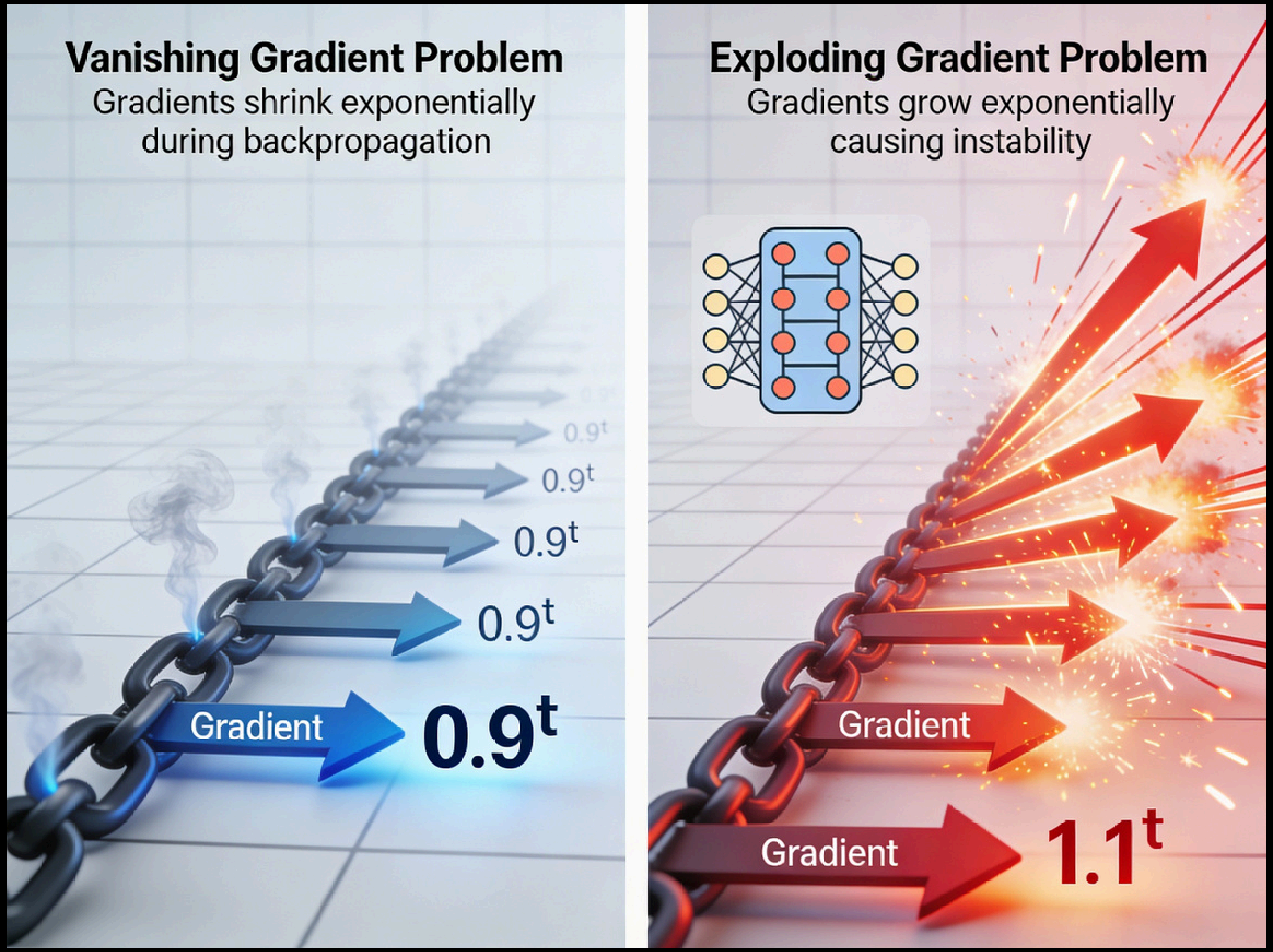
Applications in speech and language

RNNs shine in speech and language tasks by remembering context in words and sounds, making chatbots sound natural and helping apps understand your voice better. Cool, right?

LIMITATIONS IN RNN

Vanishing and Exploding Gradients

During backpropagation through time (BPTT), gradients can become extremely small (vanish) or extremely large (explode).



- Vanishing gradients → The network struggles to learn long-term dependencies because earlier layers receive almost no gradient updates.
- Exploding gradients → Training becomes unstable, causing large weight updates.

LSTM: Memory That Actually Works

Special gating mechanisms: input, forget, output

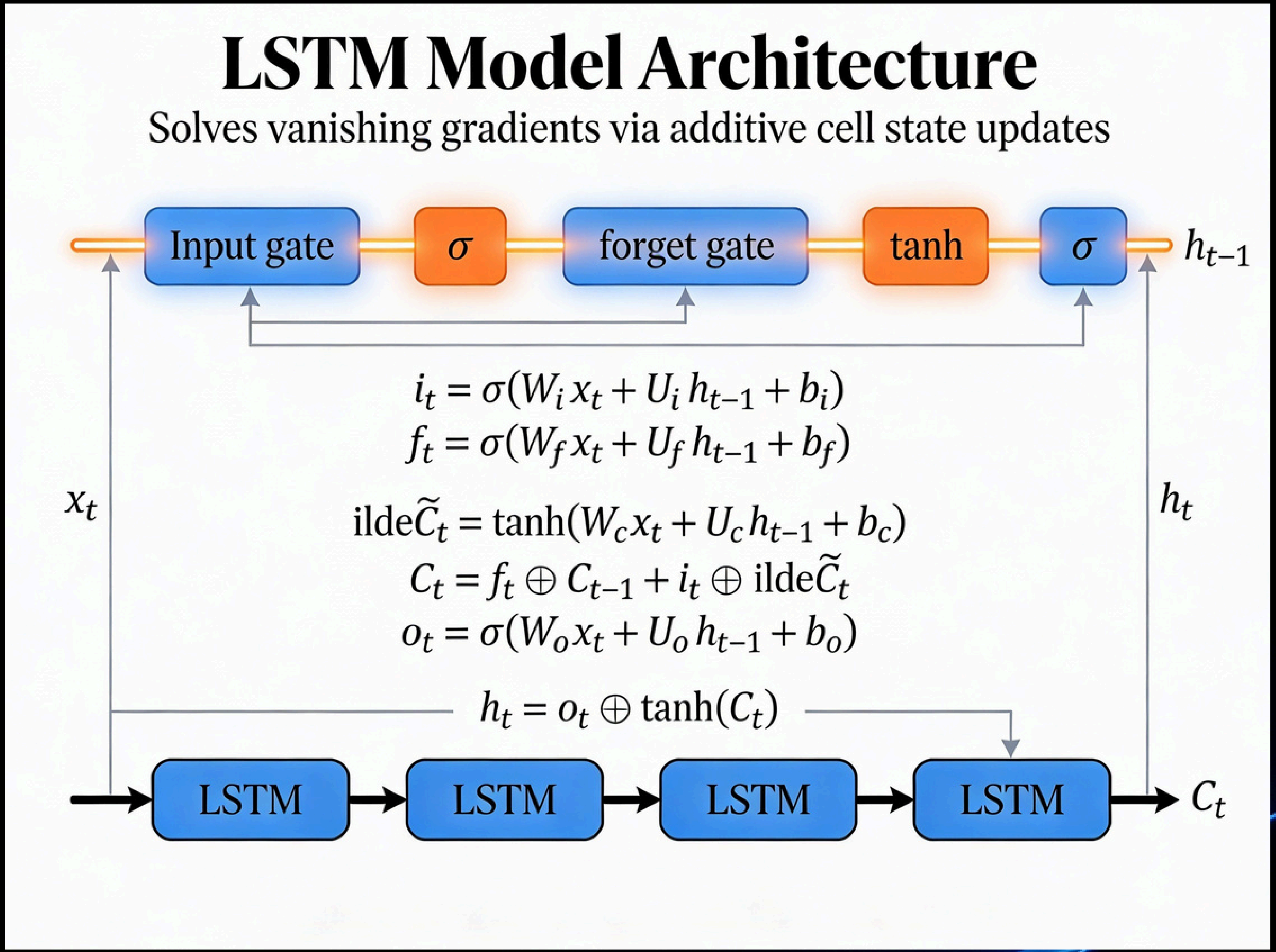
LSTMs use special gates input, forget, and output that control what info to keep, update, or share, making memory in neural networks actually work for sequence learning.

Solving the vanishing gradient problem

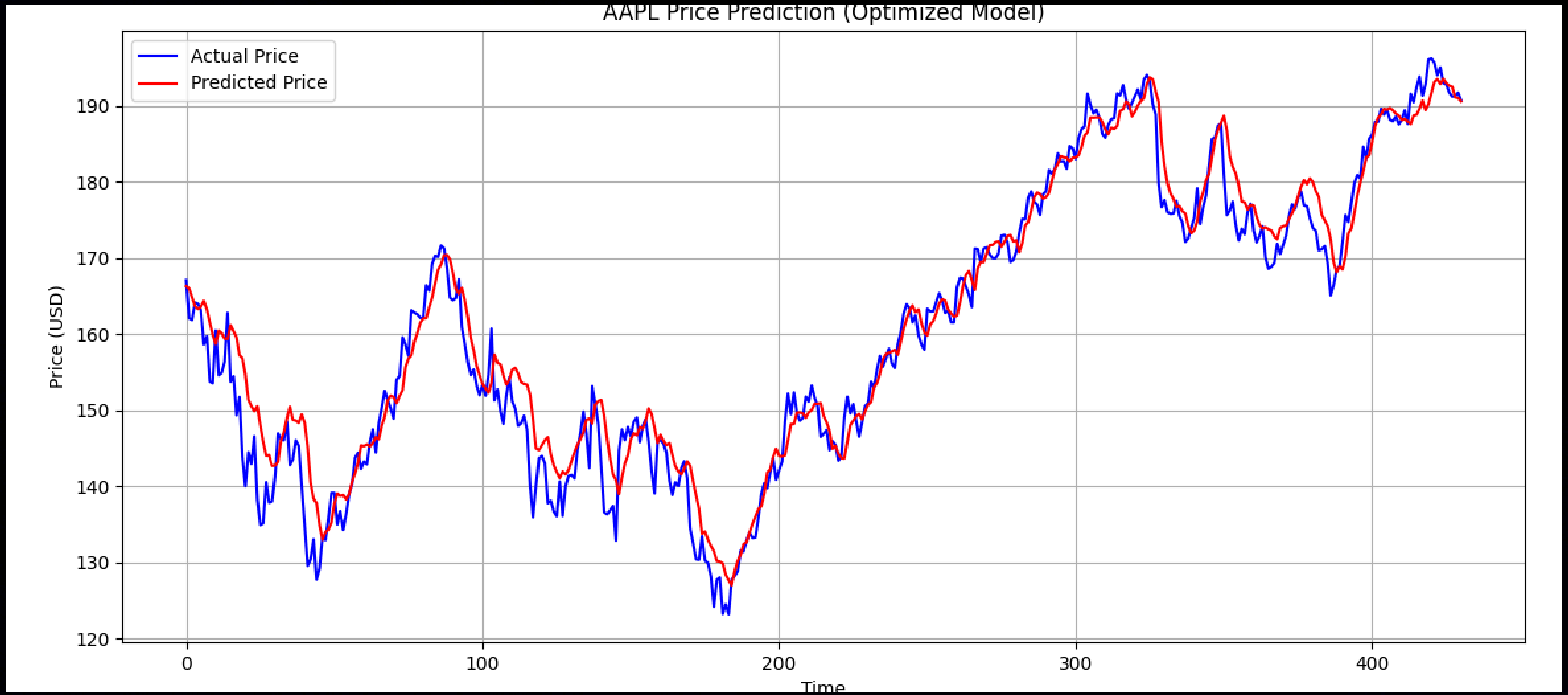
LSTMs tackle the vanishing gradient problem by using gates that control information flow, helping the network remember important stuff over long sequences making training way more stable and effective.

Selective memory as intelligence

Selective Selective memory means LSTMs focus on important info while ignoring the noise, helping neural networks learn sequences better kind of like how we remember key moments, making AI smarter and more efficient.
memory as intelligence



Instantances from Task

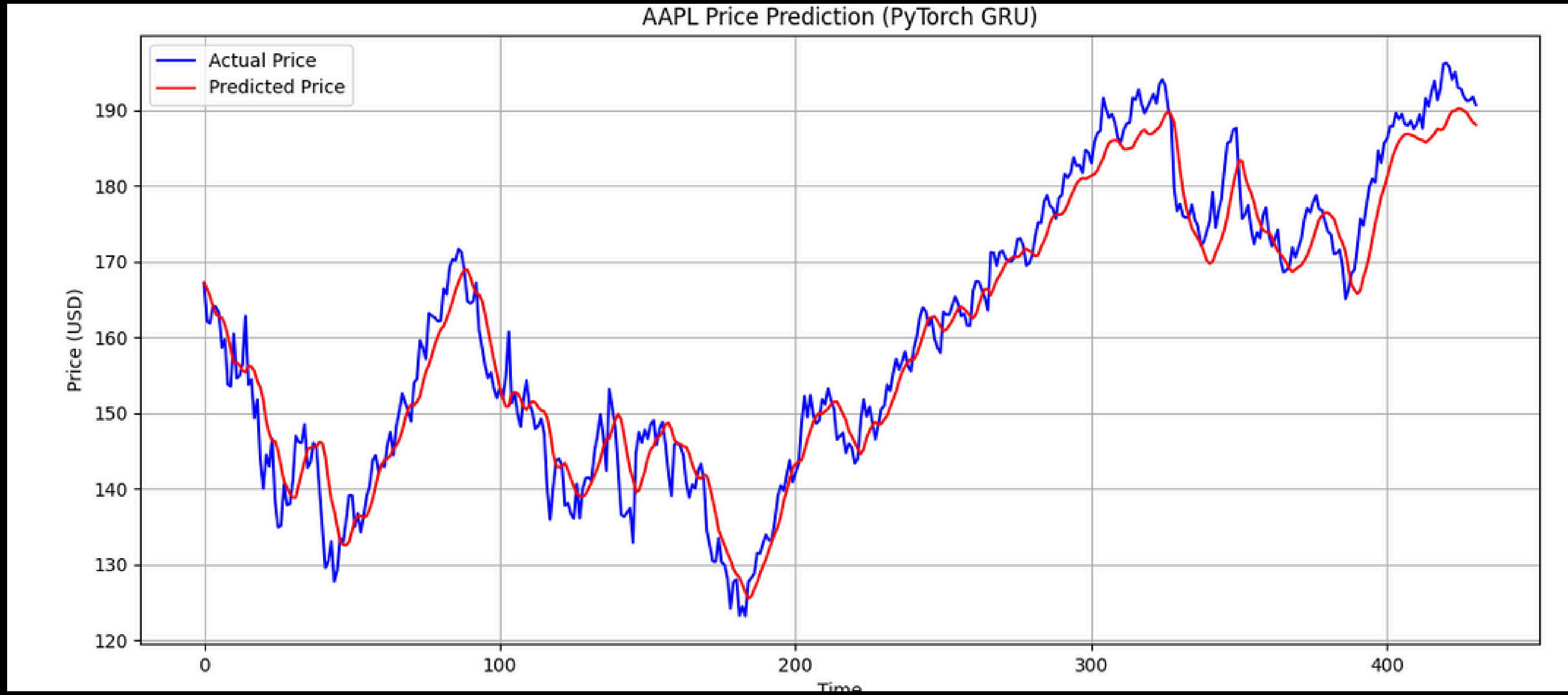


Why not ANNs or CNNs?

WHY RNN?

Instances of our task

We tested with GRU(gated recurrent units)the precdiction of aapl stocks



NEVER

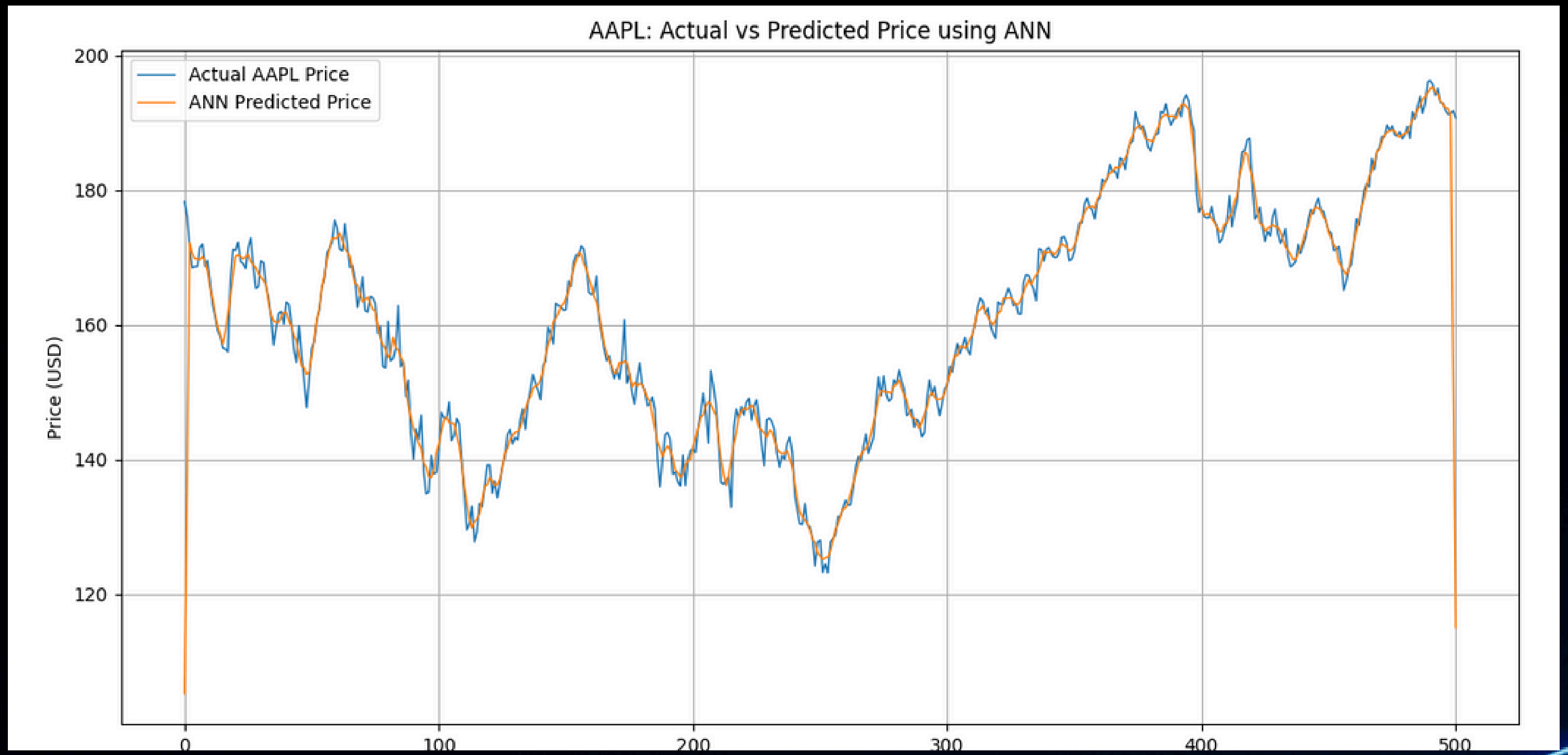
Can ANN models be used?

In the ANN plot:

- The prediction is noisy
- It follows the price but with lag
- Sudden changes are poorly handled
- Endpoints behave badly (sharp drop at edges)

Why this happens ?

- ANN sees only a fixed window of past values
- It has no memory
- Each prediction is independent



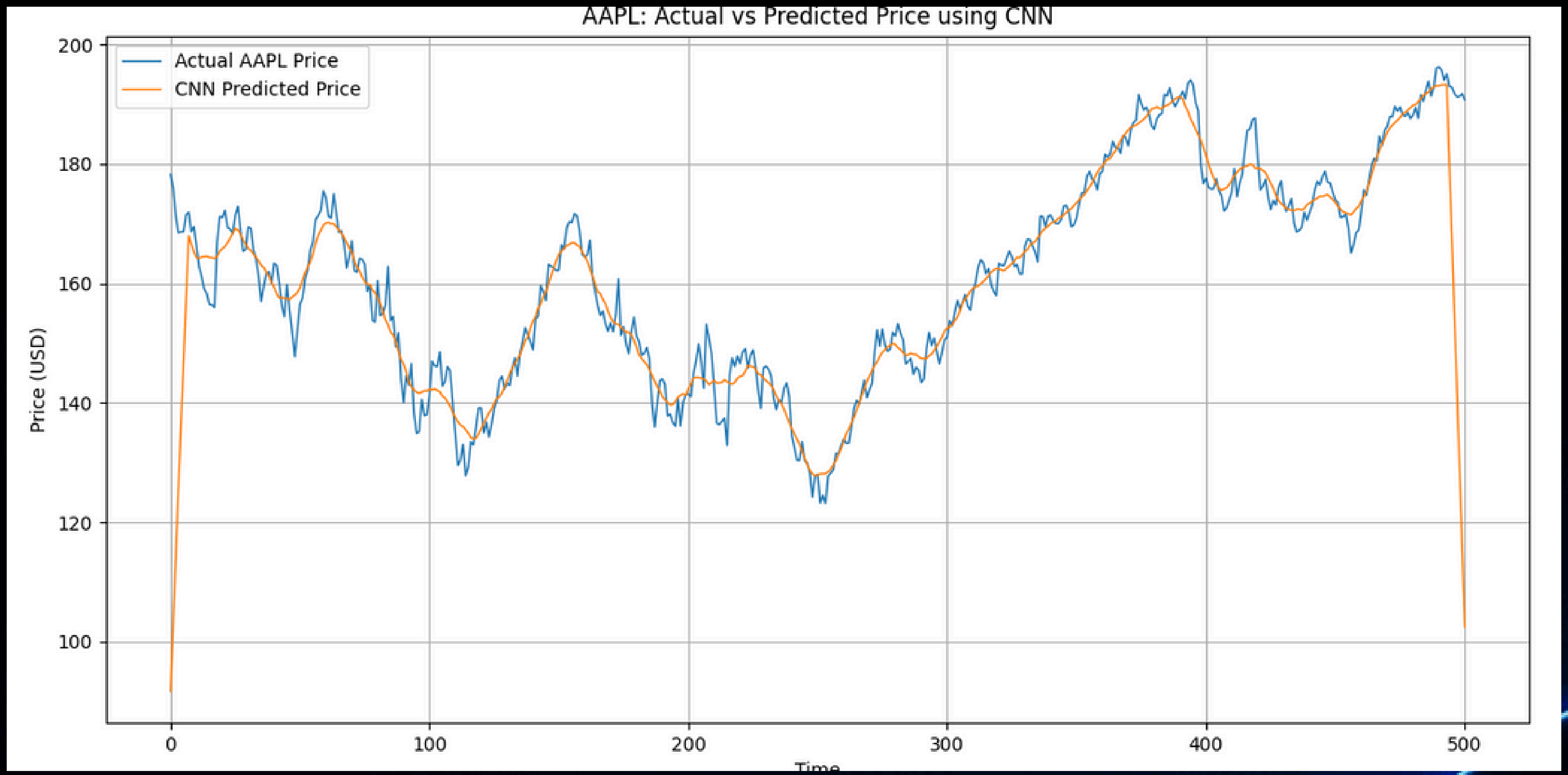
Can CNN models be used?

In the CNN plot:

- Prediction is smoother than ANN
- Local patterns are captured better
- Still lags behind major trend changes
- Long-term trend is underestimated

Why this happens?

- CNN captures local temporal patterns
- Convolution kernel has a fixed receptive field
- No mechanism to remember distant past



06

Modern Architectures and Closing Thoughts

Handling long dependencies with LSTMs

LSTMs help tackle long dependencies by remembering info over time, avoiding the forgetfulness problem regular RNNs face—making them great for sequences like language or time series.

Attention mechanisms learning relationships

Attention mechanisms help neural networks focus on important parts of data, making it easier to learn relationships and dependencies, especially in complex sequences like language or time series.

Transformers: stacking attention and MLPs

Transformers stack attention and MLP layers to capture complex patterns in data, enabling models to focus on important information while processing sequences efficiently and flexibly.

Parallel architectures and scalability

LSTMs handle sequences step-by-step, which can slow things down. Transformers, on the other hand, process data in parallel, making them way faster and more scalable for big tasks.

Backbone of modern AI

LSTMs kickstarted handling sequences, but Transformers blew things up—making AI faster, smarter, and more versatile. Together, they're the real backbone powering today's intelligent systems.

From Neurons to Intelligence

Teaching machines to learn, not program

Today's AI rocks specific tasks, but moving toward AGI means building systems that learn, reason, and adapt like humans—big challenges, but super exciting times ahead!

From Biological to Artificial Intelligence

Biological neurons inspire artificial ones, but while our brains use complex signals, AI simplifies them into math—making smart models that keep learning and evolving every day.

Evolution from perceptrons to deep networks

Starting from simple perceptrons, neural networks evolved into deep architectures, unlocking layers of complex patterns and powering today's AI with incredible learning abilities. Cool, right?

The road from narrow AI toward AGI

Today's AI rocks specific tasks, but moving toward AGI means building systems that learn, reason, and adapt like humans—big challenges, but super exciting times ahead!

From pattern recognition to reasoning

Neural networks started with pattern recognition, like spotting images, but now they're evolving to reason and make decisions—bringing us closer to true AI intelligence every day!

Thanks

Expressing Gratitude and Appreciation in Life