transformers for time series.

=> traditional deep learning methods for time series forecasting rely heavily on recurrent structures.

RNN → reads sequence step by step
LSTM → remembers longer forget noise
GRU → Simplified memory unit

The above all share the same limitation.
they all process data sequentially.

issue-1 Long range dependency collapse
if something in the sequence from 100 steps ago impacts today.
=> An LSTM must carry that information through 100 recurrent updates.
→ Noise, randomness, small fluctuations slowly erode the memory.
→ Gradients shrink → older information disappear.

This makes LSTM short-sighted especially in financial markets.

issue-2 parallelism bottleneck

RNN force pytorch to process time steps one by one which is slow and inefficient.

transformers remove recurrence completely and replace it with.

Self Attention.

Any time step can directly connect to any other time step
this makes model

=> Global
=> Parallel
-> Faster
-> Better at long seq
=> Better learning patterns.

This is why transformed daminate NLP, audio, visial, TSF

Let take a example of 50 time steps.

instead of passing the sequency to an LSTM

the transformes say Let me Look at every time step and see how important all the others are.

this comparision happen through three Learned vectors.

Query (Q) → what this position wants to find
key (K) → what information this position contains
Value (V) → content to be shared

So here every time step becomes the weighted combination of All other time steps.
This gives deep global understanding of the sequence

Multi-head attention:

Instead of Single attention mechanism transformer create multiple heads.

each head.

=> has its own Q, K, V projections.

=> looks at the sequence differently

=> capture different kind of relationship

one head might track.

=> Short term spikes

=> long term cycles

=> Volatility clusters.

=> momentum shifts

=> Cross-feature interactions.

transformer Encoder Block.

Each Encoder block has two major components.

A) Multi-head self-Attention.
this part responsible for understanding relationships
it takes the entire sequence and directly builds
global dependancies.

B) feed forward Network (FFN)

After the attention has mixed information across time.
=> Each time step is passed through a small neural
network.

=> This expands the representation.
-> Applies nonlinearity.
-> Compresses back to model dimension.

=> Why this is better for financial time Series.

transformer encoders shine because they capture long range interactions.

the impact of a federal Reserve announcement 30 days ago on today's stock move.

=> they can handle multivarite inputs naturally.
=> they process the entire window in parel
=> they are less prone to forgetting.

Because LSTM forgets distant events
Attention directly connects to them.

D-Linear Model

Despite of power of deep models like transfo
a simple linear model often performs better.
Use pure linear layers, Do not overfit the noise.