

TECH ARTICLE / AI

What is a Gated Recurrent Unit (GRU) and How Does it Work?

06 February 2024 • 5 min

👁 1050 👍 2

➦ Share

**Ellis Stewart**Content Manager at
EM360Tech

Expert



Neural Networks power many of the AI systems we use today. They play a crucial role in popular technologies like [Large Language Models \(LLMs\)](#), autonomous vehicles and a range of [machine-learning technologies](#), and continue to push the boundaries of what's possible across a wide range of applications in artificial intelligence and beyond.

Neural networks are made up of several different sub-architectures that enable them to play such an important role in today's technology.

One of these architectures is the Gated Recurrent Unit (GRU), which is crucial to unlocking the power of neural networks for a wide range of applications.

This article tells you everything you need to know about Gated Recurrent Units (GRUs), including **what GRUs are**, **how they work** and their role in recurrent neural networks.

What is a gated recurrent unit (GRU)?

Gated Recurrent Units (GRUs) are **a type of Recurrent Neural Network architecture** specifically designed to deal with sequential data.

They were introduced to solve a long-standing problem with recurrent neural networks (RNNs) known as the **vanishing gradient problem**, which makes it difficult for RNNs to remember information from earlier steps in long sequences.

What is the vanishing gradient problem?

The vanishing gradient problem happens in standard RNNs because the information from earlier steps gets multiplied by small activation values (often sigmoid or tanh) during backpropagation.

As the multiplications occur repeatedly, the gradients become incredibly small, effectively vanishing. This prevents the network from learning long-term dependencies.

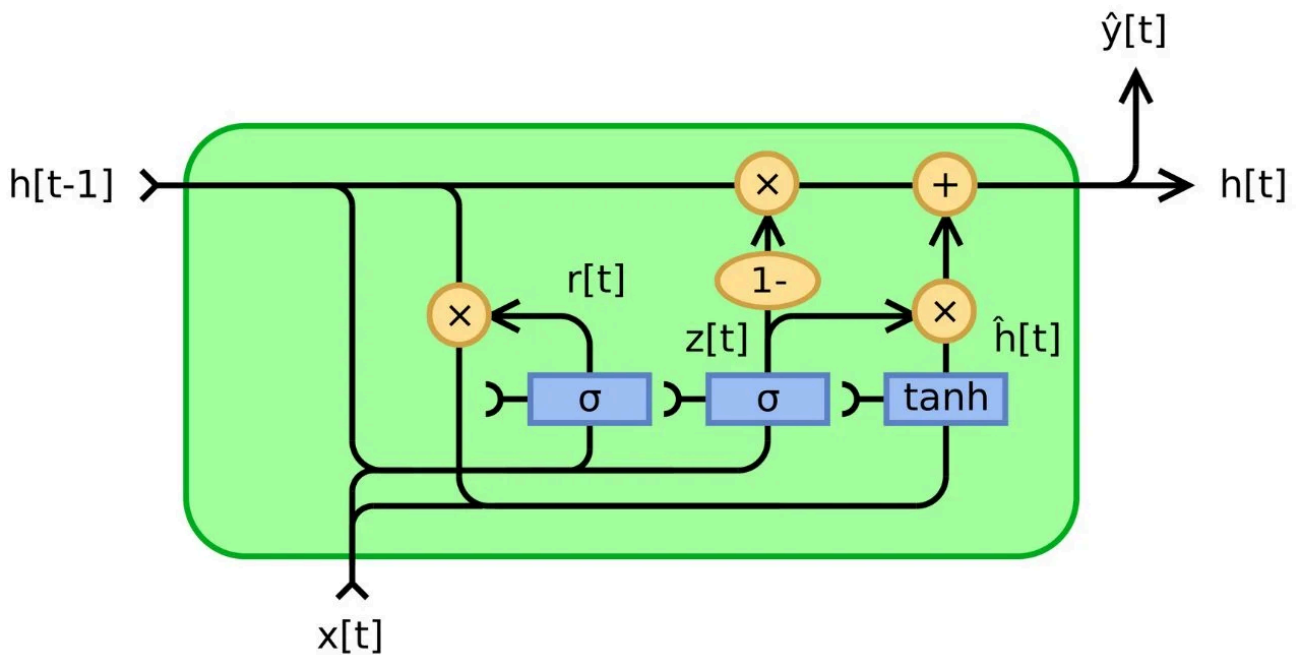
GRUs, along with Long Short-Term Memory units (LSTMs), overcome this issue by using gates to control information flow within the network. These gates regulate the flow of information within the network, meticulously selecting what information to preserve, pass on, or forget.

How do GRUs work?

Unlike other RNNs, GRUs **use gates to control the flow of information**, allowing them to learn and remember long-term dependencies in data.

Through two internal gates, the **Reset Gate** and the **Update Gate**, GRUs carefully decide what past information to discard and what to integrate with the current input. This allows them to effectively remember relevant information from earlier steps in a sequence – including with long spans where traditional RNNs often struggle.

The **reset gate**, with its output between 0 and 1, decides how much of the previous hidden state (carrying information from earlier steps) is forgotten. Imagine a value close to 0 erasing most past information, while a value close to 1 remembers it almost entirely. This selective forgetting prevents irrelevant details from cluttering the network's memory.



Gated recurrent network, fully gated version.

Similar to the reset gate, the **update gate** operates on the current input and previous hidden state, but its output also determines how much of the processed current input is integrated into the new hidden state. A value close to 0 emphasizes the previous state, while a value close to 1 prioritizes the new information.

The reset gate's output essentially "filters" the previous hidden state, keeping only the relevant information. This filtered past is then combined with the processed current input, weighted by the update gate's output, to create the new **candidate hidden state**.

Finally, the network combines the new candidate hidden state with the original previous hidden state, again weighted by the update gate's output. This ensures that only the relevant, updated information persists, while irrelevant parts are gradually phased out.

By using the reset gate, update gate, and candidate hidden state, GRUs can **selectively update their hidden states** and effectively capture both short-term and long-term dependencies in sequential data. This solves the vanishing gradient problem and enables the model to learn more complex patterns in sequences.

Advantages of using GRUs over RNNs

RNNs struggle with long-term dependencies due to the vanishing gradient problem, where information from earlier steps gets progressively weaker as it travels through the network.

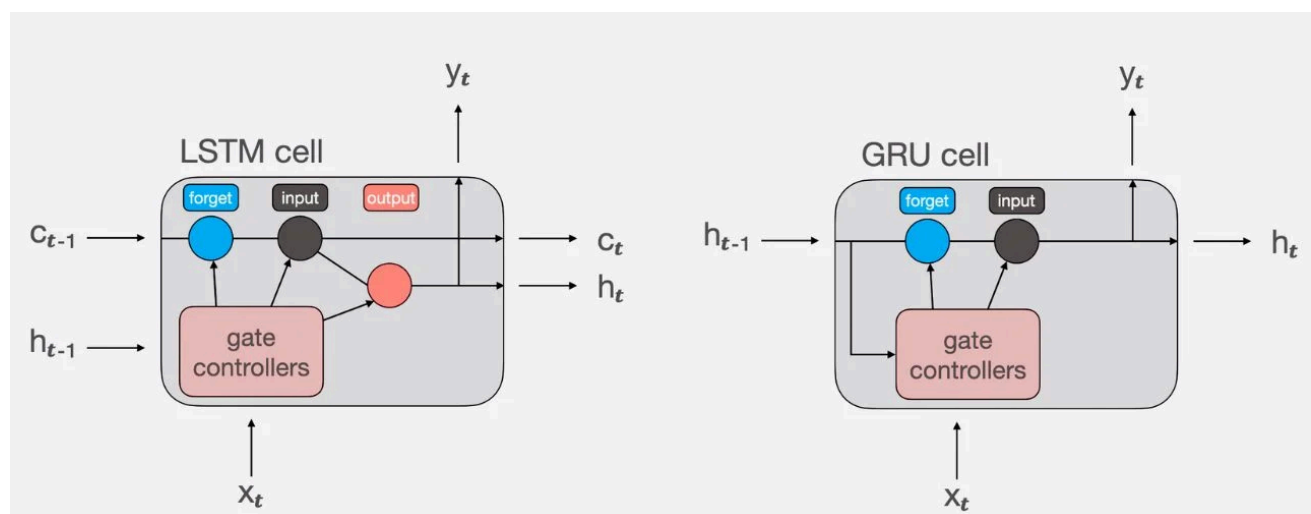
GRUs tackle this by using **gated mechanisms** (reset gate and update gate) that control information flow and selectively forget irrelevant details.

The reset gate decides what past information to forget, and the update gate controls how much new information to integrate. This selective control prevents irrelevant details from dominating the network and mitigates the vanishing gradient problem.

GRUs vs LSTMs: What's the difference?

Both GRUs and LSTMs are types of recurrent neural networks (RNNs) designed to handle sequential data that both use gates to control information flow and address the vanishing gradient problem.

But GRUs have a **simpler architecture** with fewer parameters due to only using two gates (reset and update). This makes them faster to train and less computationally expensive.



The structure of a GRU cell compared to an LSTM cell.

LSTMs, however, have a more complex architecture with three gates (forget, input, and output) and a separate cell state. This allows them to capture more complex long-term dependencies but requires more computational resources.

If computational efficiency is crucial and long-term dependencies aren't overly complex, GRUs are more suited. But for highly demanding tasks requiring remembering distant information, LSTMs might be necessary.

Challenges and considerations

While GRUs offer several advantages over traditional RNNs and LSTMs, there are some key challenges and considerations to keep in mind when deciding if they're the right choice for your project:

For one, Compared to LSTMs, GRUs have a simpler architecture and may struggle to capture very long-term dependencies in complex sequences. If your task requires remembering information from distant points in the sequence, LSTMs might be a better fit.

You've also got to consider that new RNN architectures like Transformers are emerging and showing promising results, and could potentially exceed GRUs in certain tasks.

It's important to stay updated on these developments for any AI development project.

Tech Article

What is the EU AI Act? A Look into Europe's First AI Law

213 8

6 min

[Read now](#)

Tech Article

A Look into AI Studio, Meta's New Chatbot Builder

608 2

4 min

[Read now](#)

Tech Article

OpenAI Takes Aim at Google Search With SearchGPT

207 2

5 min

[Read now](#)

[Learn more about AI](#)



Artificial Intelligence
547 Articles | 53 Podcasts

Explore AI

Did you find the article helpful?



👍 2 📖 0

➦ Share

More from Ellis Stewart

☰ TOP 10

Top 10 Biggest GDPR Fines in History (So Far)

by Ellis Stewart • 13 min



 [TECH ARTICLE](#)

Microsoft Says Cyber Attack Caused Global Azure Outage

by Ellis Stewart • 4 min



Explore more

Ellis Stewart

 Expert

Content Manager at
EM360Tech

 417  0  0

Recommended for you

BARC

The Ultimate Guide to Data Pipelines for Generative AI: Five Criteria to Evaluate Tools

 [WHITEPAPER](#) [DATA](#)

The Ultimate Guide to Data Pipelines for Generative AI...

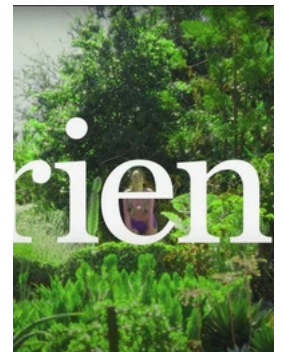
by Kevin Petrie

• 3 min

 [TECH ARTICLE](#)

Meet Friend: The AI Wearable That's Always Listening

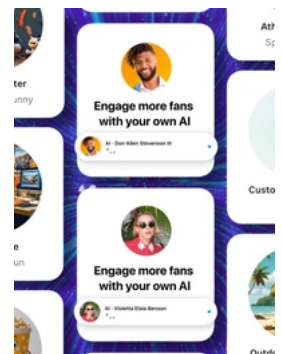
by Katie Baker • 5 min



 [TECH ARTICLE](#)

A Look into AI Studio, Meta's New Chatbot Builder

by Ellis Stewart • 4 min



Explore All



Access the latest analyst-led podcasts, tech articles, and industry resources as you connect with some of the brightest minds in enterprise tech.



Topics ▼

Resources ▼

Collaborators ▼

Company ▼

Subscribe to the EM360Tech Newsletter

Name

E-mail

Subscribe now

[Contact Us](#) [Our Services](#) [Privacy Policy](#) [Editorial Policy](#) [GDPR Policy](#) [Sitemap](#)

All rights reserved. ©2024 | Enterprise Management 360