



ARTIFICIAL INTELLIGENCE

# Training R<sup>2</sup> baselines

Motivation

Vincent Mueller

Feb 3, 2022 7 min read

## Recommended Articles

Close



### MCP in Practice



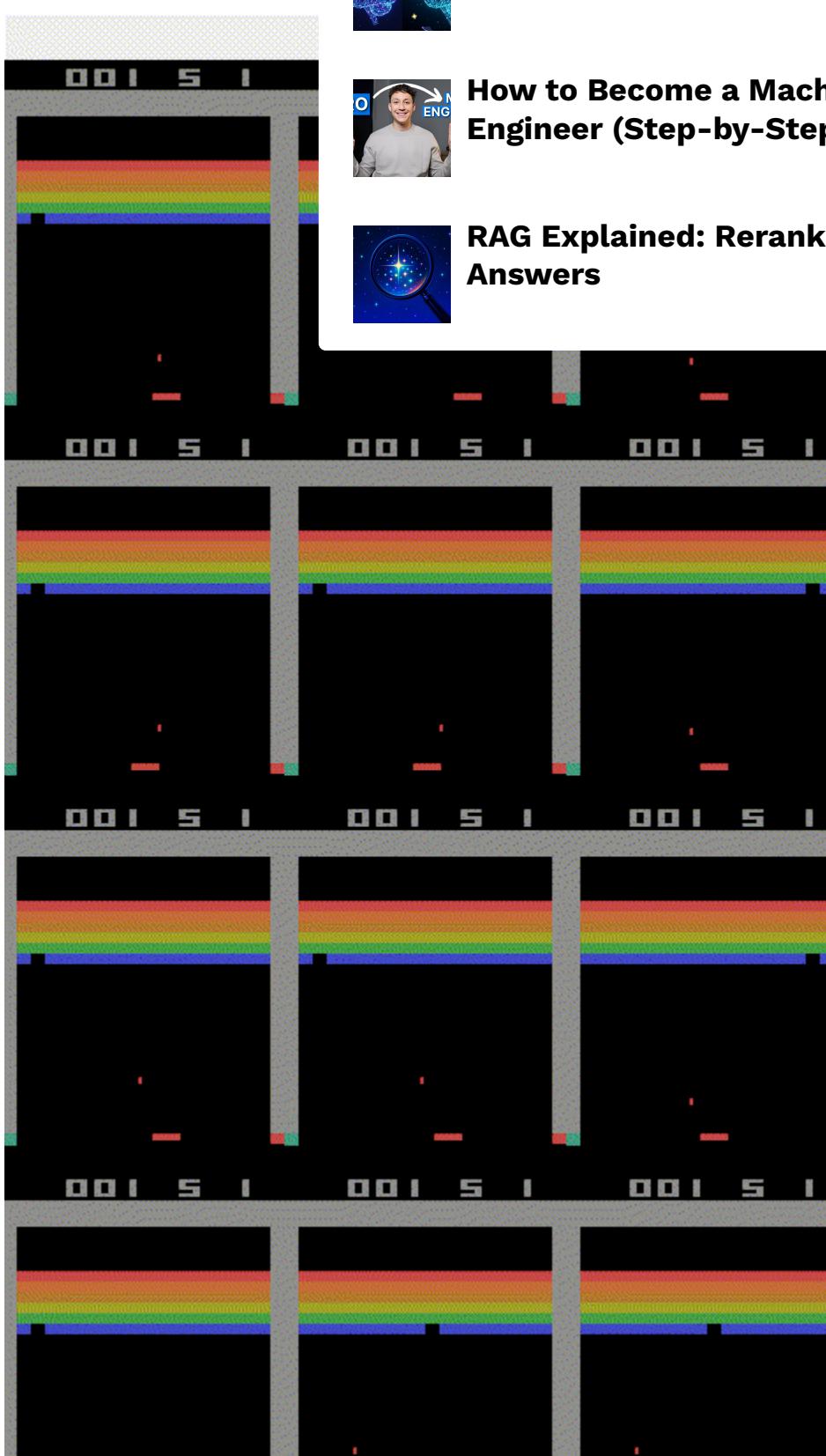
### Smarter, Not Harder: How AI's Self-Doubt Unlocks Peak Performance



### How to Become a Machine Learning Engineer (Step-by-Step)



### RAG Explained: Reranking for Better Answers



Since about 2 years, reinforcement learning has become a hobby for me. I especially enjoy training agents on games. A huge problem for me during these years, was the lack of a reliable reinforcement learning library for python and I had to either program the state of the art algorithms by myself or source on github. stable-baselines3

## Recommended Articles

[Close](#)



### MCP in Practice

## What you can

I will walk you through the stable-baselines3 library. You can train an agent on multiple environments. I will show you how to train an agent on the more complex LunarLander-v2 environment and an A2C agent on the atari breakout environment.



### Smarter, Not Harder: How AI's Self-Doubt Unlocks Peak Performance



### How to Become a Machine Learning Engineer (Step-by-Step)



### RAG Explained: Reranking for Better Answers

## Installation

The **stable-baselines3** library provides the most popular reinforcement learning algorithms. It can be installed via the python package manager "pip".

```
pip install stable-baselines3
```

I will demonstrate these algorithms using the openAI gym environment. Install it to follow along.

```
pip install gym
```

## Testing algorithms with cartpole environment

### Training a PPO agent

The stable-baselines library contains many different reinforcement learning algorithms. One of them is Proba

learning algorithms. In the following Code, I will show, how you can train an agent that can beat the openai cartpole environment using the proximal policy optimization algorithm.

```
1  from stable_baselines3 import PPO
2  import gym
3
4  # Parallel environments
5  env = gym.make("CartPole-v1")
6
7  model = PPO(policy =
8  model.learn(total_timesteps=1000)      Recommended Articles
```

Close

Cartpole\_PPO1.py hosted with [GitHub](#)

You can easily exchange the PPO policy with another one. For example, if you want to use a CNN policy, you can replace PPO with MlpPolicy in the code above.



## MCP in Practice



## Smarter, Not Harder: How AI's Self-Doubt Unlocks Peak Performance



## How to Become a Machine Learning Engineer (Step-by-Step)



## RAG Explained: Reranking for Better Answers

```
1  from stable_baselines3 import PPO
7  model = PPO(policy = "MlpPolicy",env =
```

Setting the policy to "**MlpPolicy**" means, that we use **state vector** as input to our model. There are other options here. Use "**CnnPolicy**" if you provide **image** inputs. There is "MultiInputPolicy" for handling multiple inputs. The cartpole environment cannot output images, I will use case of "CnnPolicy" later on with other gym environments.

## Saving and loading models

To save the model, use the following line of code:

```
model.save("ppo_cartpole") # saving the model to ppo_cartpole.pkl
```

You can load the saved model back into python

```
model = PPO.load("ppo_cartpole") # loading the model from ppo_cartpole.pkl
```

The following code shows the whole process of training and saving the model.

and loading a PPO model for the cartpole environment. Make sure, that you save your model only after training it.

```
1  from stable_baselines3 import PPO
2  import gym
3
4  env = gym.make("CartPole-v1")
5  model = PPO(policy = "MlpPolicy",env = env, verbose=1)
6  model.learn(total_timesteps=25000)
7
8  model.save("ppo_cartpole") # saving the model to ppo_cartpole.zip
9  model = PPO.load("ppo_cartpole.zip")
```

## Recommended Articles

Close

```
10
11 obs = env.reset()
12 for i in range(1000):
13     action, _state =
14     obs, reward, done
15     env.render()
16     if done:
17         obs = env.reset()
```

StableBaselinesSavingAndLoad



### MCP in Practice



### Smarter, Not Harder: How AI's Self-Doubt Unlocks Peak Performance



### How to Become a Machine Learning Engineer (Step-by-Step)

## Parallel Training on Multiple Environments

You can also very easily train multiple environments at the same time (parallel training) which speeds up the training process of the agent.



### RAG Explained: Reranking for Better Answers

We can create parallel environments using the `n_envs` function of the `stablebaselines3` library.

```
from stable_baselines3.common.env_util import make_vec_env
```

We use it in the same way, we used the `openai gym` to create a new environment. But we tell the function how many parallel environments we want to create.

```
env = make_vec_env("CartPole-v1", n_envs=4)
```

Since you train a single agent, you can save the model in the same manner as before.

But one important difference to the previous case is that we don't get a terminal state when we test our trained agent. This is because an episode ends in one of the environments, it is automatically reset. Before testing the agent looked like this:

```
obs = env.reset()
for i in range(1000):
```

```

101    + in range(1000):
102        action, _state = model.predict(obs, deterministic=True)
103        obs, reward, done, info = env.step(action)
104        env.render()
105
106    if done:
107
108        obs = env.reset()

```

Now, it get shorter:

```

obs = env.reset()
for i in range(1000):
    action, _state =
    obs, reward,
    env.render()

```

The following code do the same time:

```

1  from stable_baselines3 import PPO
2  from stable_baselines3 import VecEnv
3
4  # Parallel environment
5  env = make_vec_env("CartPole-v1", n_envs=4)
6
7  model = PPO("MlpPolicy", env, verbose=1)
8  model.learn(total_timesteps=25000)
9  model.save("ppo_cartpole")
10
11 obs = env.reset()
12 for i in range(1000):
13     action, _state = model.predict(obs, deterministic=True)
14     obs, reward, done, info = env.step(action)
15     env.render()

```

[ParallelTrainingSB3.py](#) hosted with ❤ by GitHub

## Recommended Articles

[Close](#)



### MCP in Practice



### Smarter, Not Harder: How AI's Self-Doubt Unlocks Peak Performance



### How to Become a Machine Learning Engineer (Step-by-Step)



### RAG Explained: Reranking for Better Answers

## Using other gym environments

In order to run most of the other gym environments you will need to install the Box2D library for python. This is pretty straight forward on mac and linux, but painstaking on windows.

### Installing Box2D

Box2D is an open-source physics engine for 2D games. Many of the gym environments use it for handling collisions.

## **Linux/OSX**

To the extend of my knowledge, there are no problems with directly installing Box2D on linux and mac PCs.

```
pip install box2d-py
```

## **Windows**

On windows there is a process of the Box2D installation separately using swig. To install swig using a command:

```
conda install swig
```

If you don't use an IDE like Microsoft Visual C++ have it installed, then download the new

### **Recommended Articles**

Close



#### **MCP in Practice**



#### **Smarter, Not Harder: How AI's Self-Doubt Unlocks Peak Performance**



#### **How to Become a Machine Learning Engineer (Step-by-Step)**



#### **RAG Explained: Reranking for Better Answers**

### **Microsoft C++ Build Tools – Visual Studio**

Here you can install the Buildtools.

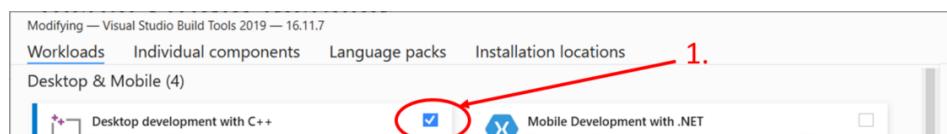
Visual Studio Installer

Installed Available

Visual Studio Build Tools 2019  
16.11.7  
The Visual Studio Build Tools allows you to build native and managed MSBuild-based applications without requiring the Visual Studio IDE. There are options to install the Visual C++ compilers and libraries, MFC, ATL, and C++/CLI support.  
[Release notes](#)

(Image by author)

When you have installed the "buildtools", open the Visual Studio installer (it is probably already open after installing the "buildtools").



Location: C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools

By continuing, you agree to the [license](#) for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the [3rd Party Notices](#) or in its accompanying license. By continuing, you also agree to those licenses.

Total space required: 6.66 GB

## Recommended Articles

[Close](#)

Then you can inst:

pip install box2d-py

**Beating LunarLander**

I will now show yo  
using the stable\_b  
landing module be



### MCP in Practice



### Smarter, Not Harder: How AI's Self-Doubt Unlocks Peak Performance



### How to Become a Machine Learning Engineer (Step-by-Step)



### RAG Explained: Reranking for Better Answers

(GIF by author)

It is a more complex task than the cartpole environment.  
The agent is given the following information in form of sensor readings:

- (Continuous): X distance from target site
- (Continuous): Y distance from target site
- (Continuous): X velocity

- (Continuous): Y velocity
- (Continuous): Angle of ship
- (Continuous): Angular velocity of ship
- (Binary): Left leg is grounded
- (Binary): Right leg is grounded

So we have to use the **MlpPolicy** as well.

I chose the PPO al  
be learning very fa  
agent 2 million tra  
The game is consi

## Recommended Articles

Close

### MCP in Practice



### Smarter, Not Harder: How AI's Self-Doubt Unlocks Peak Performance



### How to Become a Machine Learning Engineer (Step-by-Step)



### RAG Explained: Reranking for Better Answers

```

1 import gym
2 from stable_baselines3 import PPO
3
4 # Parallel environment
5 #env = make_vec_env('LunarLander-v2', n=4)
6
7 # Create environment
8 env = gym.make('LunarLander-v2')
9
10 # Instantiate the agent
11 model = PPO('MlpPolicy', env, verbose=1)
12 # Train the agent
13 model.learn(total_timesteps=int(2e6))
14 # Save the agent
15 model.save("ppo_lunar2")
16
17 # Load the trained agent
18 #model = PPO.load("ppo_lunar", env=env)
19
20 # Enjoy trained agent
21 obs = env.reset()
22 for i in range(10000):
23     action, _states = model.predict(obs, deterministic=True)
24     obs, rewards, dones, info = env.step(action)
25     env.render()
26     if dones:
27         obs = env.reset()

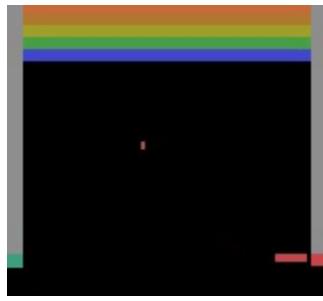
```

LunarLander-v2\_PPO.py hosted with ❤ by GitHub

## Atari breakout from pixels

Now it is time for our agent to tackle "atari break  
the pixels on the screen.





(GIF by author)

The breakout environment is not included in the standard installation of gym so you have to install a gym atari collection

## Recommended Articles

Close

```
pip install gym[atari]
```



### MCP in Practice

Given only a single and direction of the ball, how can we give the agent a navigation signal so he can learn the navigation?



### Smarter, Not Harder: How AI's Self-Doubt Unlocks Peak Performance



### How to Become a Machine Learning Engineer (Step-by-Step)

```
from stable_base
```



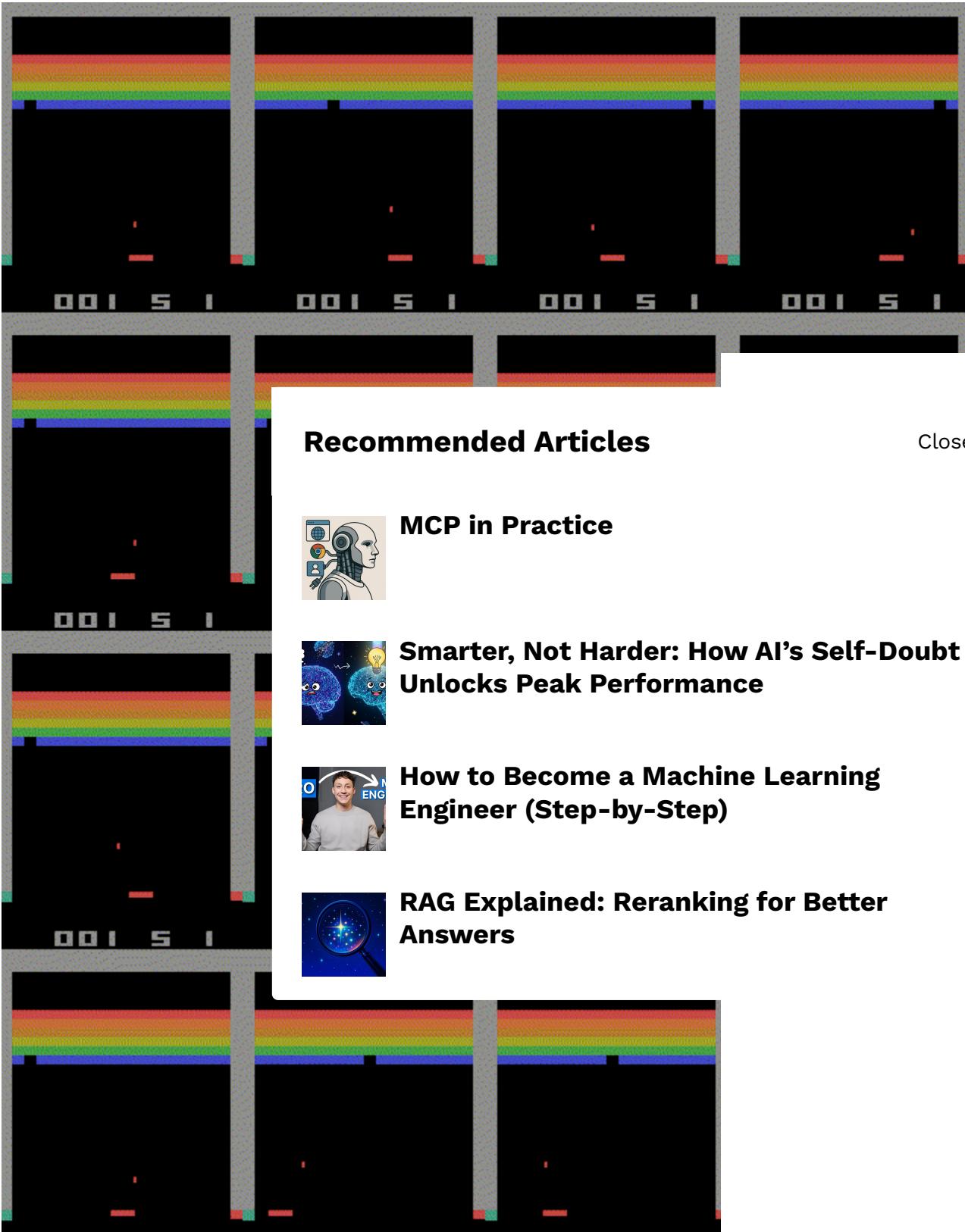
### RAG Explained: Reranking for Better Answers

Also be aware of the training time. I trained the agent for **5 million timesteps** \ on the **OpenAI Gym** algorithm and used **16 parallel environments**.

```
1  from stable_baselines3.common.env_util import make_atari_env
2  from stable_baselines3.common.vec_env import VecFrameStack
3  from stable_baselines3 import A2C
4
5  # There already exists an environment generator
6  # that will make and wrap atari environments correctly.
7  # Here we are also multi-worker training (n_envs=4 => 4 environments)
8  env = make_atari_env('BreakoutNoFrameskip-v4', n_envs=16)
9  # Frame-stacking with 4 frames
10 env = VecFrameStack(env, n_stack=4)
11
12 model = A2C("CnnPolicy", env, verbose=1)
13 model.learn(total_timesteps=int(5e6))
14 obs = env.reset()
15 #model = A2C.load("A2C_breakout") #uncomment to load saved model
16 model.save("A2C_breakout")
17 while True:
18     action, _states = model.predict(obs)
19     obs, rewards, dones, info = env.step(action)
20     env.render()
```

breakout\_A2C.py hosted with ❤ by GitHub





## Recommended Articles

Close

### MCP in Practice



### Smarter, Not Harder: How AI's Self-Doubt Unlocks Peak Performance



### How to Become a Machine Learning Engineer (Step-by-Step)



### RAG Explained: Reranking for Better Answers



As you can see, the agent has learned the trick of jumping in the bricks and shooting the ball behind the wall. It has difficulties with shooting the last few bricks and is not able to finish the game. This is most likely caused by the fact that the agent was not trained sufficiently on situations with no bricks. By increasing the training duration, the agent will be able to beat the environment.

## Conclusion

I want to take a brief moment to talk about the most important information about the stable-baselines3 library. The environment provides a vector with information about the state of the game, such as the position of the ball, the position of the paddle, and the positions of the bricks.

then use the **MlpPolicy**. If it instead gives whole images, then use the **CnnPolicy**. You can use multiple environments in parallel to speed up the training. But they all train the same **one agent**. The cartpole environment can be beaten easily with a few thousand time steps of data. The LunarLander-v2 environment is more complex required 2 million timesteps to beat with PPO. Atari breakout will be solved with pixels and this makes it an even harder task. With 5 million timesteps, I was almost there. I think the environment update is still not perfect.

## Recommended Articles

Close

### Want to connect?

Linkedin <https://www.linkedin.com/in/Vincent02770108/>  
Become medium reader  
membership fees :)



#### MCP in Practice



#### Smarter, Not Harder: How AI's Self-Doubt Unlocks Peak Performance



#### How to Become a Machine Learning Engineer (Step-by-Step)



#### RAG Explained: Reranking for Better Answers

[Join Medium with me](#)

### Related stories

[Deep Q learning is no rocket science](#)

[Snake with Policy Gradients Deep Reinforcement Learning](#)

[Backpropagation in Neural Networks](#)

### Other stories

[How you can use GPT-J](#)

[Eigenvalues and eigenvectors in PCA](#)

[Support Vector Machines, Illustrated](#)