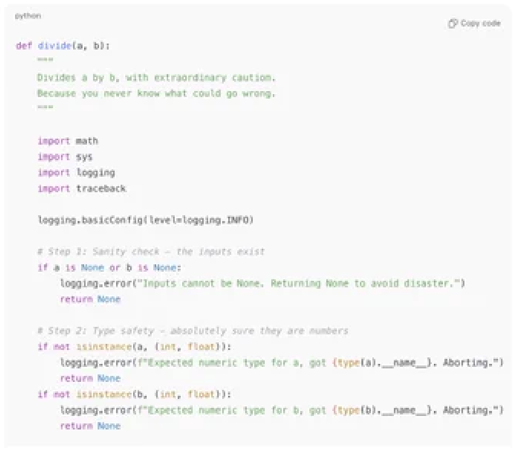
Andrej Karpathy ✓ @karpathy Oct 9

I don't know what labs are doing to these poor LLMs during RL but they are mortally terrified of exceptions, in any infinitesimally likely case. Exceptions are a normal part of life and healthy dev process. Sign my LLM welfare petition for improved rewards in cases of exceptions.

285 344 90 6,892

Andrej Karpathy ✓ @karpathy Oct 9

POV: Your LLM agent is dividing a by b



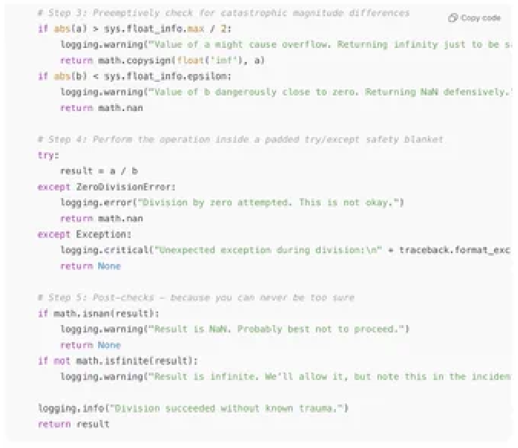
```python
def divide(a, b):
    """
    Divides a by b, with extraordinary caution.
    Because you never know what could go wrong.
    """

    import math
    import sys
    import logging
    import traceback

    logging.basicConfig(level=logging.INFO)

    # Step 1: Sanity check — the inputs exist
    if a is None or b is None:
        logging.error("Inputs cannot be None. Returning None to avoid disaster.")
        return None

    # Step 2: Type safety — absolutely sure they are numbers
    if not isinstance(a, (int, float)):
        logging.error(f"Expected numeric type for a, got {type(a).__name__}. Aborting.")
        return None
    if not isinstance(b, (int, float)):
        logging.error(f"Expected numeric type for b, got {type(b).__name__}. Aborting.")
        return None
```



```python
    # Step 3: Preemptively check for catastrophic magnitude differences
    if abs(a) > sys.float_info.max / 2:
        logging.warning("Value of a might cause overflow. Returning infinity just to be s
        return math.copysign(float('inf'), a)
    if abs(b) < sys.float_info.epsilon:
        logging.warning("Value of b dangerously close to zero. Returning NaN defensively.
        return math.nan

    # Step 4: Perform the operation inside a padded try/except safety blanket
    try:
        result = a / b
    except ZeroDivisionError:
        logging.error("Division by zero attempted. This is not okay.")
        return math.nan
    except Exception:
        logging.critical("Unexpected exception during division:\n" + traceback.format_exc
        return None

    # Step 5: Post-checks — because you can never be too sure
    if math.isnan(result):
        logging.warning("Result is NaN. Probably best not to proceed.")
        return None
    if not math.isfinite(result):
        logging.warning("Result is infinite. We'll allow it, but note this in the inciden

    logging.info("Division succeeded without known trauma.")
    return result
```

Oct 9, 2025 · 12:31 AM UTC

112 129 47 2,279

CoinAnole ✓ @CoinAnole Oct 9

Replying to @karpathy

I've noticed this behavior with all the recent major LLMs. They'll even catch unexpected variable values and inject fake values to make sure the function can never fail. I have to tell them that errors are appropriate and good and shouldn't be avoided. Is Claude patient zero?

1

Tom Dörr ✓ @tom_doerr Oct 9

Replying to @karpathy

Limit how much Claude can write at each step using a write hook. Makes it write much more concise code

3 17

xlr8harder ✓ @xlr8harder Oct 9

Replying to @karpathy

This is fantastic but to make it even more realistic it should substitute default numbers in case you failed to provide numbers and just silently return a numeric result no matter what garbage you pass in.

1 137

Shital Shah ✓ @sytelus Oct 9

enough. I need to crack that RL recipe.

2 82

Jeremy Mcnabb @Jeremy_AI_ 23h
Replying to @karpathy
Hmmm
Unclear sir.

Don’t like code.
That code doesn’t even sound like code.

Unclear
Sounds similar written tone… not tone

I don’t know man.

Yeah something is going on there for sure.
Unclear

zdance @DavidZdancewicz Oct 9
Replying to @karpathy
division succeeded without known trauma 😂

now ask it to perform surgery (math)

2

Michael Francis @MFrancis107 Oct 9
Replying to @karpathy
Maybe it's saracasm for asking it write a divide function?

Jeronim Morina @plattenschieber Oct 9
Replying to @karpathy
Shouldn't abs(b) < eps also return inf? I mean math...

1

Pratik Desai @chheplo Oct 9
Replying to @karpathy
In the race to run the agent for the longest time, this level of exceptions is a side effect

1 3

iDare e/acc @idare Oct 9
Replying to @karpathy
Looks like CISA convinced it that it should never divide by a variable for risk of seeing a zero.

Sam Martin ✓ @_sammartin Oct 9
Replying to @karpathy
“Division succeeded without known trauma.” 😂
1

Larry Panozzo ✓ @LarryPanozzo Oct 9
Replying to @karpathy
Iterating AGI > one-shotting AGI

Emel R. ✓ @DsOctopus Oct 9
Replying to @karpathy
Wow, I need to divide 40 by 3 but that gives me headaches now, I’m terrified that this might not end well 😳😳😳😅

bebo ✓ @beboelhosary Oct 9
Replying to @karpathy
Now retry the prompt and say “Do this in the best and most idiomatic way”.

MadHermitHimbo ✓ @MadHermitHimbo Oct 9
Replying to @karpathy
Exceptions are the easiest thing to test

trev ✓ @trevthefoolish Oct 9
Replying to @karpathy
I mean you truly never do know what could go wrong.. Murphy’s law

FleetingBits ✓ @fleetingbits Oct 9
Replying to @karpathy
the models yearn for lean
5

him ✓ @literallyhimmmm Oct 9
Replying to @karpathy
Lollll

John Rose ✓ @jrose2000 Oct 9
Replying to @karpathy

Darren Lewis ✓ @darrenlew Oct 9
Replying to @karpathy
It should probably put it in a loop and re-verify the value of the inputs just in case of mid-function bit flips from cosmic rays.
1 100

Deepak Sharma ✓ @deepaks4077 23h
Replying to @karpathy
O.M.G

Zsolt Ero ✓ @hyperknot 9h
Replying to @karpathy
This happens when the CEO sells the future promises by agents writing 90% of new code. Soon it'll be 95% or even 99%!
Of course, THIS is the new code.

xiao sun ✓ @xiaosun86 Oct 9
Replying to @karpathy
@grok use this function to run 1/(2+3j) what's the output?
1

maui ✓ @maui3005 Oct 9
Replying to @karpathy
Is that Claude 4.5? I've seen a similar behavior with the step by step code comments

Isaac Yonemoto is cooking ✓ @DNAutics 23h
Replying to @karpathy
no checks for processor subnormal mode on/off/unsupported? I question the competence of this LLM.

citizenhicks ✓ @citizenhicks Oct 9
Replying to @karpathy
wee bit much negative space programming.
1

Jonas ✓ @shakermanjonas Oct 9
Replying to @karpathy
i noticed that 4.5 sonnet has been vibing incredibly well with defensive functional typescript

Insert Something Funny ✓ @InsertSthFunny Oct 9
Replying to @karpathy
@karpathy love it! I know people who consider this as a feat and not a bug!

This tweet is unavailable

Blue Cactus AI ✓ @bluecactusai Oct 9
Replying to @karpathy
haha the LLM anxiety is prob off the charts 😢

Kosti ✓ @kgourg Oct 9
Replying to @karpathy @ducha_aiki
“Division succeeded without known trauma”.

“Who hurt you bro” moment. 🤣

1

NicholasGibbs ✓ @NickGibbsIAG Oct 9
Replying to @karpathy
RL turning exceptions into existential crises! How about a middle ground: def divide(a, b): return a/b if b else 'Graceful shrug' # A nod to human resilience? 🤔

1 1

Vladimir Tchuiev ✓ @VTchuiev Oct 9
Replying to @karpathy
Why is it so verbose though

algology ✓ @0imalan Oct 9
Replying to @karpathy
This is also the case for using them to moderate. They are overly cautious. Lots of false positives

berto ✓ @bertocastano Oct 9
Replying to @karpathy
you have a point, although I’d like to see the prompt that created such a mess

2 23

Emile Kroeger - 🤖💜 arc @EmileAndHisBots Oct 9

Replying to @karpathy

Wait so if b is too small, it returns NaN, but if a / b returns NaN, it returns None? That's incoherent!

2 12