The Cash-Flow Game — how it works

A 2,929-parameter network that learnt to run a payments desk from nothing but the rules.

What you just watched. A network of 2,929 numbers ran a small Dutch company — *Overvloed & Fabel B.V.*, a €600,000-a-year party-goods wholesaler — for eleven thousand simulated years, and gradually learnt to manage its cash better than any of the three rules of thumb a real bookkeeper might use. It was never told what a good decision looks like. It was told the rules of the game and the goal — the end-of-year net position — and worked the rest out itself.

WHAT IT SEES

Every week, for each open supplier invoice, it looks at **ten numbers** and decides *pay now, or wait*. It isn't judging invoices in isolation — it's judging *this invoice, given how the company is doing this week*.

ABOUT THE INVOICE

1. how large it is
2. how many weeks until it is due
3. whether the 2% early-payment discount is still available
4. what that discount would be worth in euros

ABOUT THE COMPANY THIS WEEK

5. cash in the bank
6. how much of the €50,000 credit line is already drawn
7. weeks until the next VAT quarter
8. the size of that VAT bill
9. how overdue this invoice is already
10. customer money due to arrive in the next fortnight

HOW IT LEARNS — THE INTERESTING PART

It plays the year against itself. At every training step it runs the *same* year twice. Once **greedily**, doing whatever it currently thinks best. Once **sampling**, rolling weighted dice at each decision — a slightly improvised version of itself. Then it compares the two end-of-year results. If the improviser did better, the network shifts to make those improvised choices more likely. If the improviser did worse, it shifts away from them.

Its disciplined self plays its improvising self, and whichever wins teaches the other. Over 11,482 simulated years, small correct nudges accumulate; noise cancels; a strategy emerges.

WHY IT GETS WORSE BEFORE IT GETS BETTER

Two reasons. The sampling run deliberately does random things, and some are catastrophic — stop paying and late penalties compound at 1.5% *a week*, which pushes you onto the credit line at 12% a year, and past the limit at 26%. And a single unlucky pair of years can push the network the wrong way and knock it out of a decent strategy. Some seeds never recover. This one does.

WHAT IT DISCOVERED

It rediscovered the expert human rule. Take the 2% discount when it's offered, because 2% over one week beats 12%-a-year credit interest by miles. Otherwise hold the money until the due date, because holding is free. But never hold *past* due — 1.5% a week compounding is far worse than borrowing. That is precisely the Discount Hunter rule, and the network worked it out from nothing but the rules and a score. The extra €763 it earns on top is **situational judgement**: paying differently when cash is tight, or when a VAT quarter is looming.

Is this like ChatGPT? No — different family entirely. No language, no pre-reading, 2,929 numbers against hundreds of billions. This learns from a score, not from text.

Is it just memorising? No, and it is tested for. It trains on randomly generated years and is scored on *one fixed year it never trains on*. Memorising the training years wouldn't help.

Could I run this on my real books? The technique transfers; this model does not. The simulation has no disputes, no negotiation, no relationships, and suppliers never react to being paid late.

Why not use it for everything, then? It needed to play the year eleven thousand times. You can only do that in a simulation. Reinforcement learning works where you have a simulator or a game — routing, scheduling, board games, protein folding — and doesn't work where each attempt costs real money and real time. This is why the same technique that beat the world at Go can now manage a payments desk, and why it can't yet decide who to hire.