03 Why Is Prediction the Same as Learning?

BY NILUSHANAN KULASINGHAM

16 MIN READ · INTERACTIVE

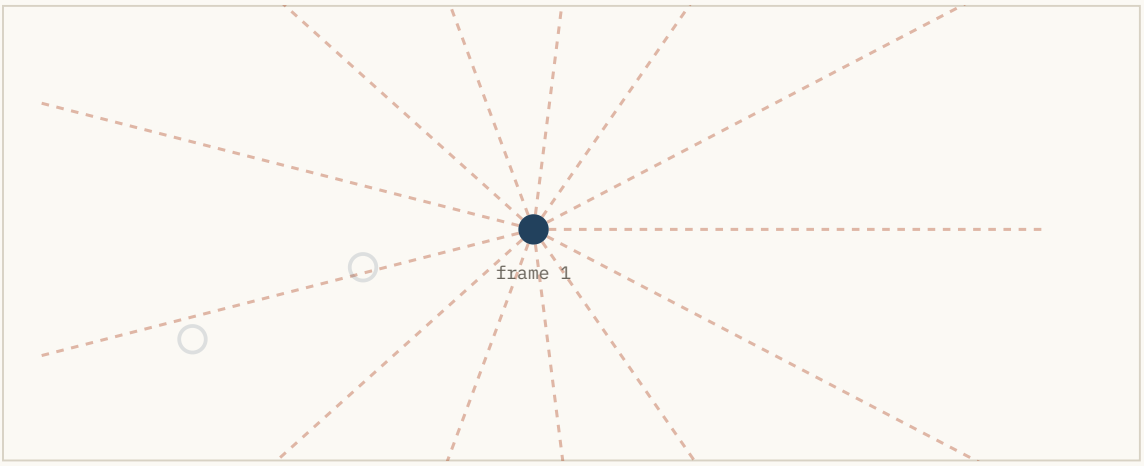
> Ask something to predict what comes next and it has no choice but to build whatever the next moment depends on. Jeffrey Elman showed it with words in 1990. Claude Shannon had already shown that the same loop, read the other way, is compression.

The figures in this chapter are interactive. Press and drag them online at worldmodels101.com/chapters/prediction.

Take a photograph of a ball in mid-air and ask where it will be a moment later. You cannot say. Up, down, sideways, or barely moving at all, the picture looks the same. Everything the answer depends on is missing from a single frame.

Now take two photographs a fraction of a second apart. This time the answer arrives before you have finished asking. Both pictures were equally sharp, so detail is not what changed.

Two frames hold something one frame cannot: a direction and a speed. Neither is drawn on either picture. They live in the relationship between the two, and you pulled them out without being told to.



one ball, photographed a few times

WHAT YOU CAN WORK OUT

Where it is. That is all a single picture can tell you.

WHAT IS STILL OPEN

Every direction, and every speed. The picture rules out nothing.

FIG. 3.1 Drag the slider from one frame to three and watch the fan. The fan is every future still consistent with what has been seen. One frame rules out nothing. A second fixes the heading and the speed, and a third settles whether the path is bending. Each extra frame is one more quantity you can work out, and none of them was in any single picture.

Ask something to predict what comes next, and it has no choice but to build whatever the next moment depends on. It was never asked for those things, and it cannot be right without them. I call them stowaways: the quantities a predictor is forced to carry that nobody put on the manifest. Direction and speed are the first two.

People have been finding larger stowaways ever since, with better instruments each time. Claude Shannon went first, in 1948, with *A Mathematical Theory of Communication*. He showed that how well you can predict a message sets the price of sending it.

Jeffrey Elman followed in 1990 with *Finding Structure in Time*. He found grammar inside a network asked only to guess the next word. Then in 2022 Kenneth Li and colleagues wrote *Emergent World Representations*, and found a whole game board

inside a network asked only to guess moves.

YEAR AND WHO	ASKED TO	INSTRUMENT	HAD TO CARRY
Opening: the ball	say where the ball will be	two photographs	IN THE HOLD
1948, Claude Shannon	send a message cheaply	a pencil and probabilities (the maths of information)	IN THE HOLD
1990, Jeffrey Elman	guess the next word	look at the network's internal states and group them	IN THE HOLD
2022, Kenneth Li and colleagues	guess the next legal Othello move	a probe trained to read one fact out of the activations	IN THE HOLD

ROWS OPEN

0 of 4

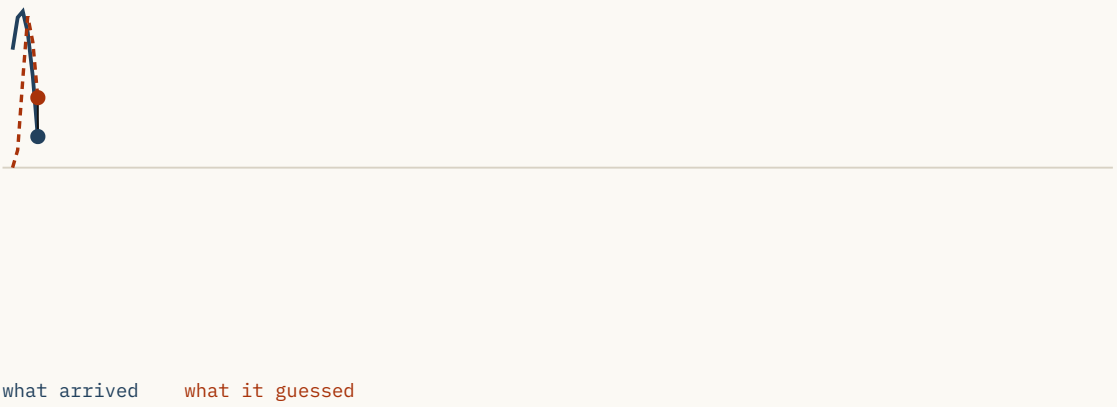
FIG. 3.2 The three finds we'll walk through, laid out as a ledger. Each row says what the system was asked to do and what instrument was used to look. Press Open the hold and the third column fills in with what it had to carry to do the job. None of those was on the manifest.

"Prediction is learning" gets said a lot, usually as a slogan, and rarely with any account of what the predictor learned or how anybody checked.

Guess, check, adjust

The loop is as small as it sounds. Look at what you have and say what you think is coming. Wait, then compare what arrived to what you said. Change yourself a little in the direction that would have been less wrong, and go again.

None of the steps is clever. The power is in what the loop does not need: nobody has to label anything, because the target is the next piece of the recording. Let's watch it run.



RECENTLY WRONG BY	WHAT IT HAS LEARNED	HOW IT IS DOING
0.426	0.52, 0.17	Guessing. It has seen almost nothing yet.

FIG. 3.3 Press Run. The predictor guesses the next number from the previous two, and is corrected a little after every guess. It starts at zero and knows nothing. Watch the dashed line climb onto the solid one, then look under What it has learned: two numbers that nobody supplied, reached by being wrong and adjusting.

Nothing in that figure was told the rule. It was told, two hundred and twenty times, by how much it had just missed. Each time it nudged its own two numbers (the *weights*: the adjustable numbers inside a model, and the only thing that learning ever changes) a little way towards being right, and that was enough. Splendid.

> This is all that *self-supervised* means. Supervised, because the model is told how wrong it was after every guess. Self, because the data did the telling rather than a person with a spreadsheet.

For most of the field's history a model learned from examples that somebody had labelled. A million images meant a person writing down what was in each one. The labels were the costly part, and the size of a dataset was the size of somebody's

budget. Prediction skips the bill, because any recording of anything is already a training set.

The argument was still open in 2013, when Yoshua Bengio, Aaron Courville and Pascal Vincent wrote *Representation Learning: A Review and New Perspectives*. It is a survey, and it reads as a case still being made. That the data was free settled it, and that convenience, more than any single result, is why prediction took over the field.

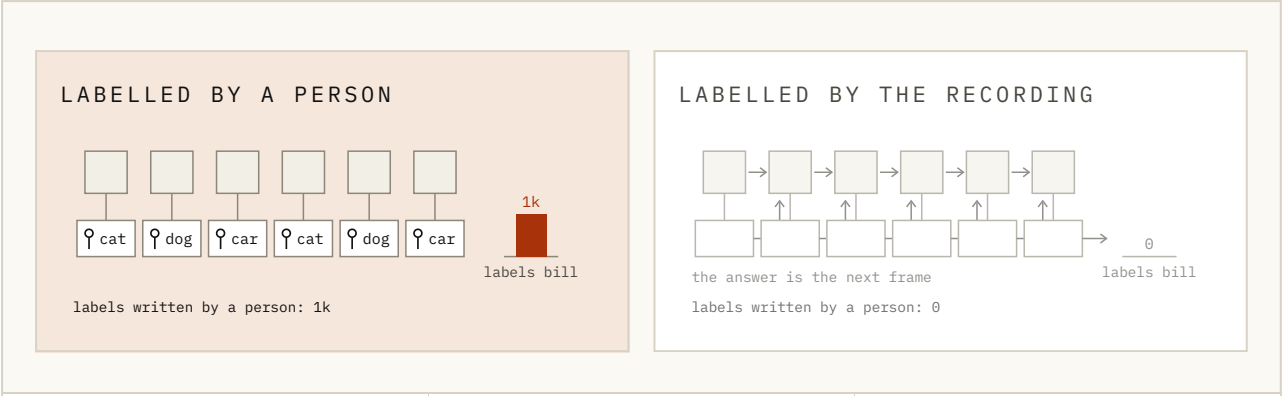


FIG. 3.4 Flip the switch between the two ways of training. On the left, every example needs a person to write the answer beside it, and the labels bill climbs with the data. On the right, the answer is the next frame of the same recording, and nobody is paid for anything. Drag the amount of data up and watch which bill grows. The numbers are illustrative.

What guessing forces you to build

Elman was a cognitive scientist at the University of California, San Diego, who came to networks from the study of language. In 1990 he trained a small network on one job: read a word and predict the next word.

Then he looked inside. The internal states had sorted themselves into groups. The groups were nouns and verbs, animate and inanimate, things that can eat and things that can be eaten.

Nobody had given the network a single category. It reached them because a word's category is what its next word depends on. The title, *Finding Structure in Time*, is literal: the structure was found rather than installed.

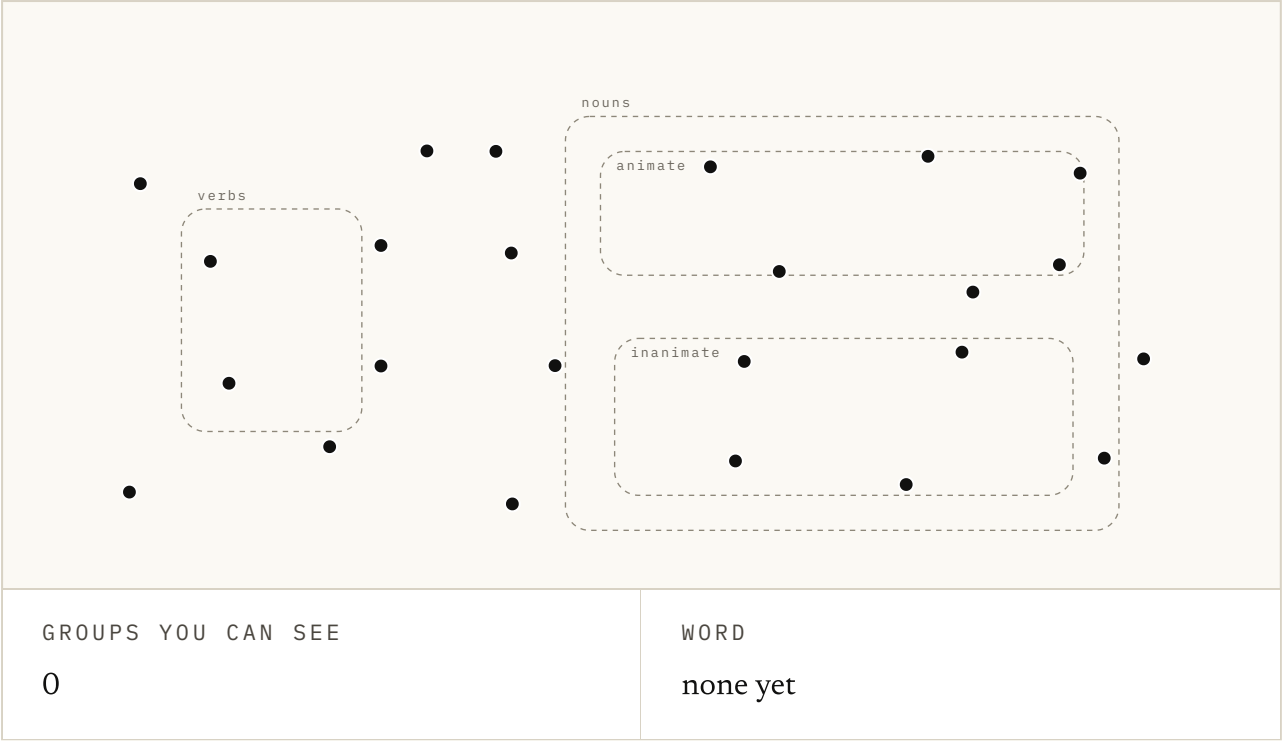


FIG. 3.5 Each dot is one word, placed by the network's internal state for it. Leave the switch on Before training and the words are scattered. Flip it to After training and watch them sort: nouns to one side, verbs to the other, and inside the nouns the animate ones apart from the inanimate. Hover or tap a dot to read the word. Nothing here was labelled, and the positions are illustrative.

For a long time that was where it stopped. The network was small and the grammar was a toy, and a toy is not a claim about the world. The finding waited thirty-two years for instruments that could make it one.

In 2022 Li and colleagues, interpretability researchers (people who open networks up rather than train them), took a network since nicknamed Othello-GPT. It was trained only to predict legal moves in Othello (a board game on an eight-by-eight grid. You flip your opponent's discs by trapping them between two of yours). It was never shown a board and never told the rules.

They probed its activations (train a small separate classifier to read one specific fact out of a network's internal numbers). The state of the board was in there, which on its own is a curiosity.

The intervention is what made it a finding. Change that internal board, and the moves the network goes on to make change with it. So the board is there because the network is using it. The two steps are worth keeping apart, because only the second one settles anything.

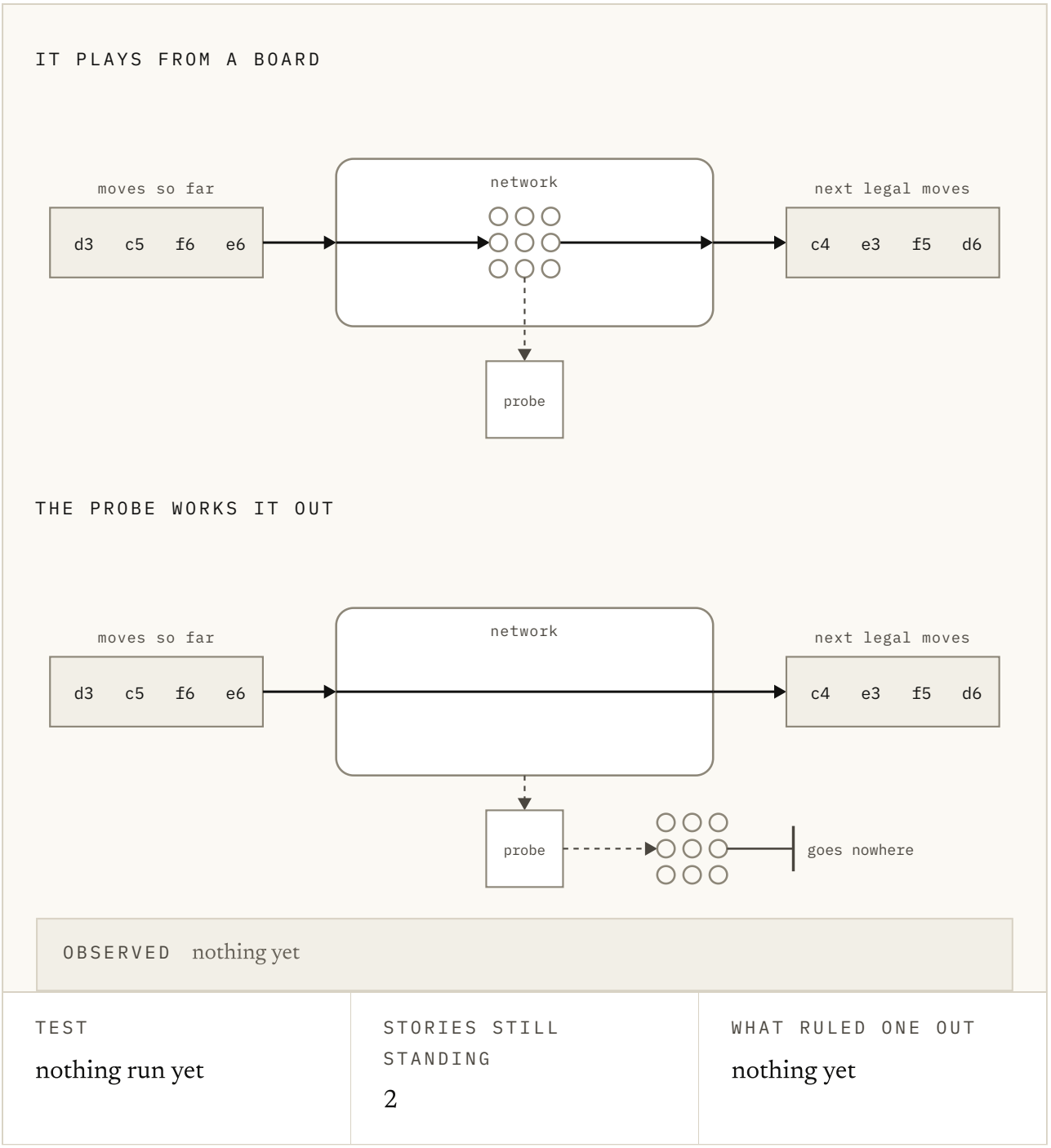


FIG. 3.6 Two stories about the same network, and they differ by one wire. Press Read the board and both survive, because both say a board can be read out. Now press Change one square and watch the next legal moves: in one story they change with it, and in the other there is nothing downstream to change. The move lists are illustrative.

Neel Nanda and colleagues studied the same network again in 2023, in *Othello-GPT has a linear emergent world representation*. Li's probe had been a small network of its own, because a straight line had failed. Nanda showed a straight line works once you

ask the right question. The network stored each square as mine or theirs rather than black or white, and flipped its view every turn.

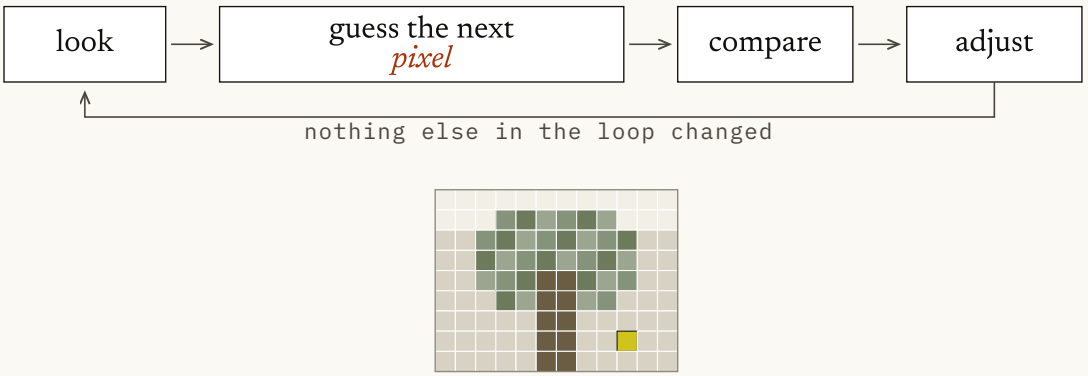
The same rule sits under both results, thirty-two years apart. Prediction rewards one thing only: making the next moment less surprising. Sometimes the cheapest way to be less surprised is to build the very thing that is making the data.

> A predictor will settle for any shortcut that works on the data it saw. Telling the shortcut from the process it stands in for is most of the work.

Predict the next what?

The loop says predict the next thing. It never says what counts as a thing, and that choice is the design decision. Let's take three choices in turn.

Predict the next pixel and you get a system that is very good at leaves. Leaves are hard to predict and they are also most of the pixels, so that is where the capacity goes (*capacity* is how much a model can hold. Spend it on one thing and it is not there for another). I call that the leaf problem, and chapter 4 is built around it.



where the capacity goes

leaves
sky, trunk, ball

most of the pixels, hardest to call

what comes out
a renderer

TARGET	WHAT COMES OUT	WHERE THE CAPACITY WENT
pixel	a renderer	the leaves

FIG. 3.7 Same loop, three targets. Pick what goes on the right-hand side of the guess. Start on pixels and watch where the capacity bar goes: to the leaves, which are most of the picture and the hardest part to call. Switch to the next word and the bar moves onto grammar and facts. Switch to a compact state and you get something a planner can search. The bars are illustrative. Nothing else in the loop changed.

Predict the next word and you get grammar, plus a surprising amount of knowledge about the world. That is what the next word turns on. In 2019 Alec Radford and colleagues at OpenAI showed how far it goes at scale. Their paper was *Language Models are Unsupervised Multitask Learners*.

Their model was GPT-2. It was trained on nothing but next-word prediction over a large pile of web text. It came out able to answer questions, summarise, and translate a little, and nobody had trained it for any of those. OpenAI released it in stages, uneasy about what a fluent text machine might be used for.

The model learned those tasks because the next word sometimes depends on them. That is the clearest evidence we have that the target decides what gets built.

FIG. 3.8 Step through the sentences and try to fill the blank yourself before you press Reveal. Each one is chosen so that the next word turns on something else: a fact, a translation, a summary of what came before. Nobody has to teach those as tasks. They are what the next word sometimes depends on.

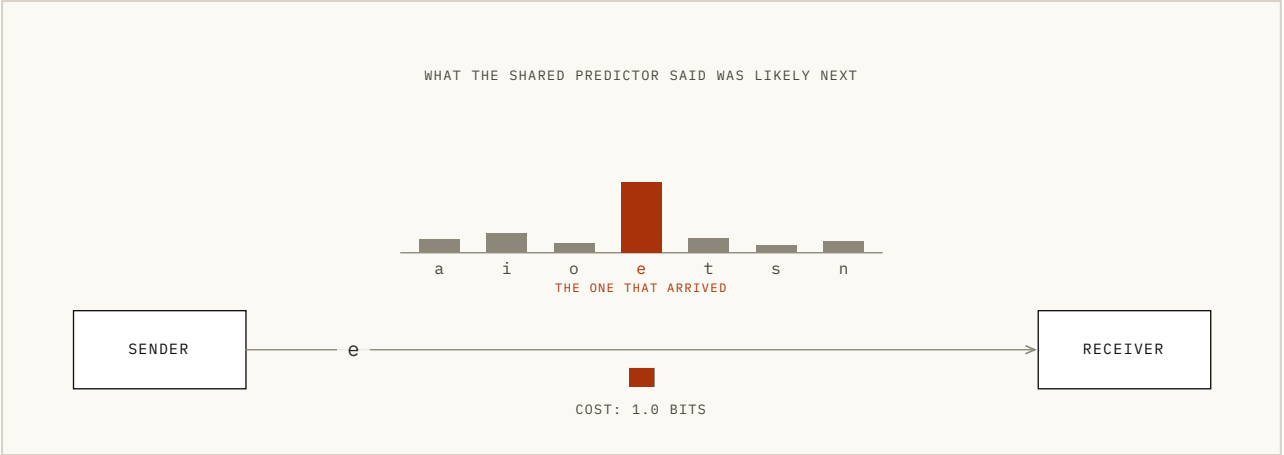
Predict the next state of some compact description and you get something a planner can use. Compact means a short list of numbers rather than a picture.

So the same loop gives three different machines, and all that changed is what was put on the right-hand side of the guess. *Predict the next thing* is a family of methods rather than one. Picking the target is the design decision that matters most, so an argument about prediction should say what is being predicted.

Being right is the same as being brief

There is a second way to look at this, four decades older than Elman's result, and it comes to the same thing. Shannon was a mathematician at Bell Labs, the research arm of the American telephone company, whose business was pushing more calls down the same wire. Suppose you have to send a message down a wire and you are charged by the symbol.

You and the receiver share a predictor. Before each symbol both of you run it and get a list of what is likely next, so you need only send enough to pick the right one. Confident and correct costs almost nothing, and confident and wrong costs a lot.



PROBABILITY GIVEN

1 in 2

COST TO SEND

1.0 bits

FIG. 3.9 Drag the slider to set how likely the shared predictor said the symbol that actually arrived was, and read off what that symbol costs to send. Half way along, one in two, costs one bit. Push it towards the left end, where the predictor was confident and wrong, and watch the price climb. Push it right and it is nearly free.

Shannon made this exact in 1948. A symbol you gave a probability of one in two costs one bit (one bit is one yes or no answer, so eight bits can pick one option out of 256). One in four costs two, and one in a thousand costs about ten. That paper started information theory, and the word "bit" first appeared in print there.

Three years later he turned the argument on English itself. In *Prediction and Entropy of Printed English* he had people guess a passage one letter at a time. A language people can guess is cheap to send, and the experiment measured how cheap. That put a human being on the bottom row of Figure 3.10.

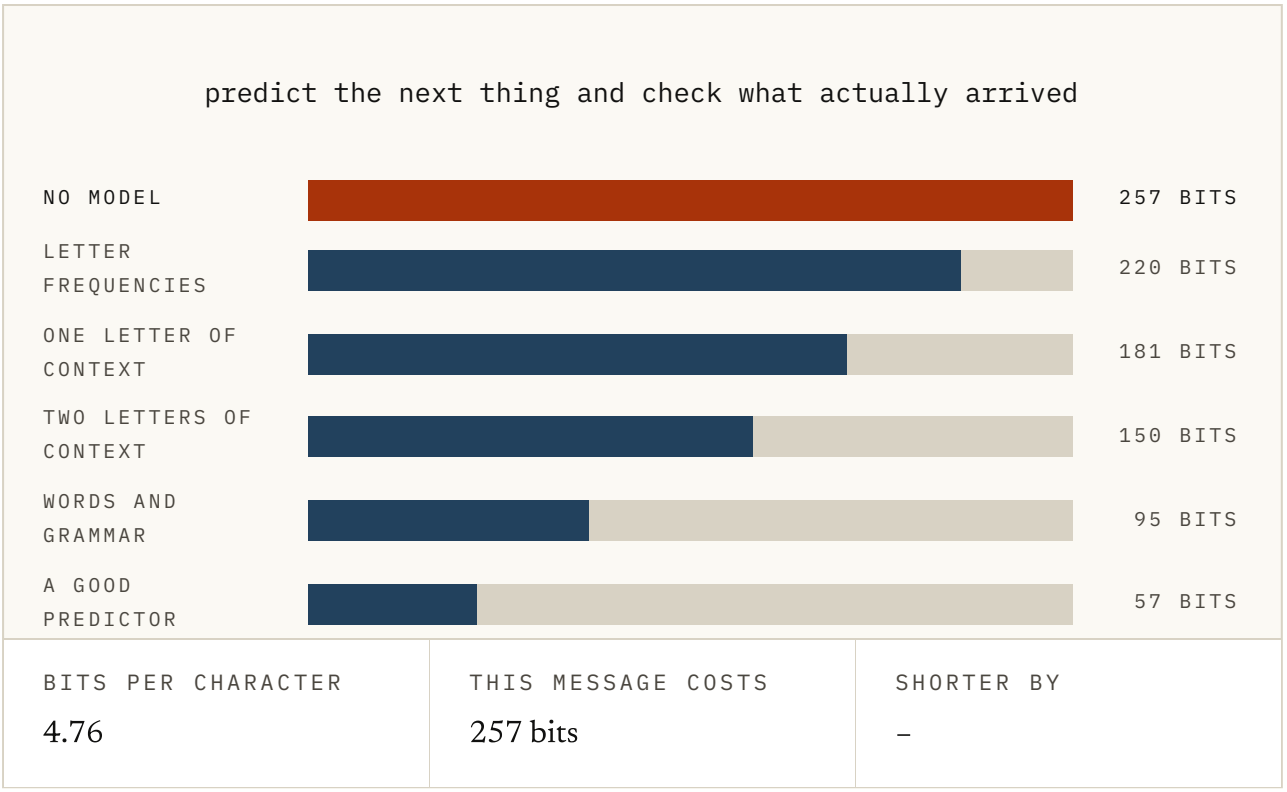


FIG. 3.10 Step the predictor up from No model to A good predictor and watch the price of the same sentence fall. The bits per character are published estimates for English rather than anything this figure computed. The bottom row is roughly where Shannon's 1951 experiment put a person guessing letter by letter, and it is also where good language models sit today.

So the length of the shortest message you can write measures how well you predicted. That makes a better predictor a better compressor. Read backwards, whatever compresses your data the hardest has understood the most about it.

Marcus Hutter, a theorist of perfect learners in the abstract, took the reading literally. The Hutter Prize is a long-running cash prize for compressing a snapshot of Wikipedia. It runs on the argument that you cannot squeeze text much further without knowing what it means. Every winner to date has been, in effect, a language model.

Jürgen Schmidhuber went a step further in 2010. Schmidhuber ran a lab in Switzerland and had co-invented the LSTM, a memory network that was the workhorse of speech recognition for years. In 2018 he would co-write the paper this course is named after. His 2010 paper was *Formal theory of creativity, fun, and intrinsic motivation*.

He proposed rewarding a learner for compression progress. That is the discovery that it can now squeeze its data further than it could a moment ago. Getting better at compressing becomes a drive in its own right, a reason to go and look at new things.

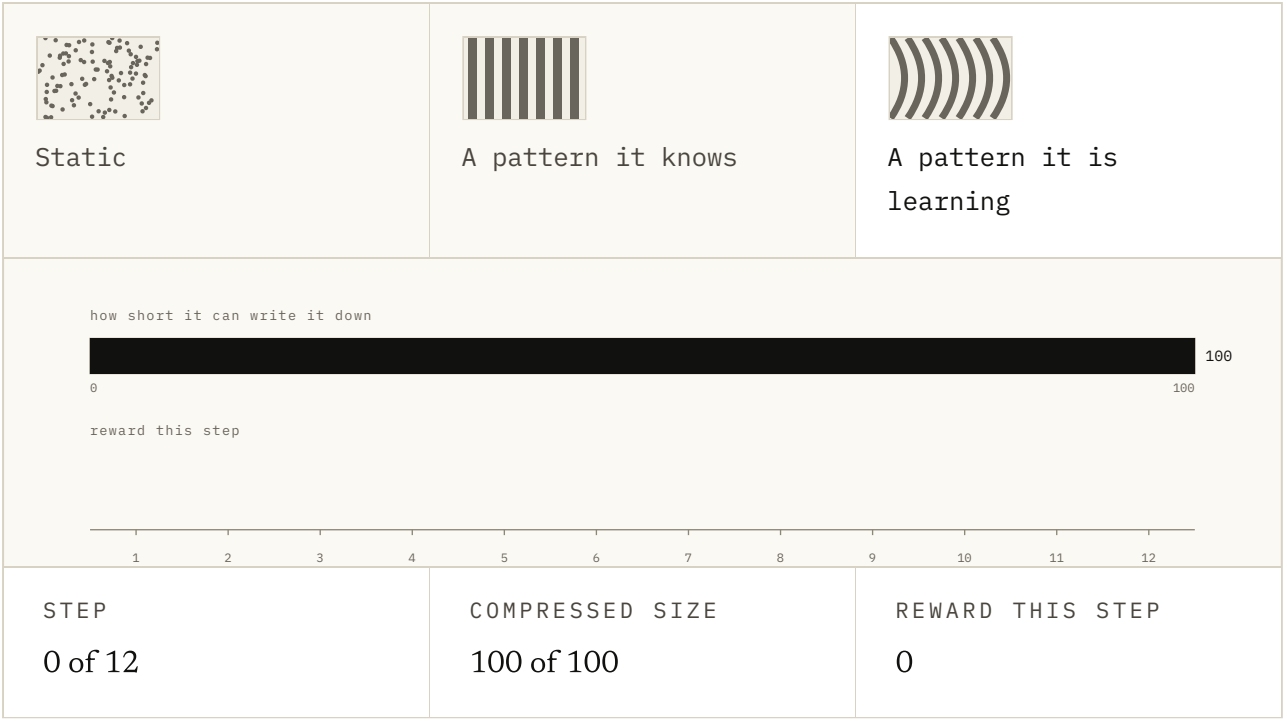
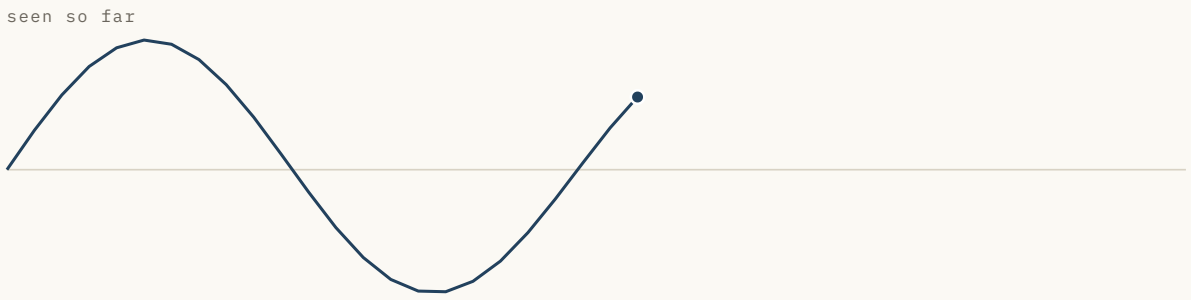


FIG. 3.11 Three things a learner could look at: static, a pattern it already knows, and a pattern it is still learning. Pick one and press Watch. The bar is how short the learner can write down what it has seen so far, and the reward at each step is how much shorter it just got. Static never gets shorter, and the known pattern is already short, so neither pays anything. All the reward is in the middle one. Illustrative.

The same reading says why the stowaways were never optional. Memorising is the costly way. A lookup table of everything that ever happened is huge, and useless on anything new.

The cheap option, the one that shortens the message, is to work out the rule that produced the data and keep that instead. The rule is the stowaway.



THE TABLE

1	0.00
2	0.31
3	0.59
4	0.81
5	0.96
6	1.02
7	0.99
8	0.87
9	0.67
10	0.41
11	0.12
12	-0.18

... and 12 more

STORES: 24 VALUES

THE RULE

the rule: two numbers (illustrative)

a 1.90 b -0.99

next = a × last + b × the one before

STORES: 2 NUMBERS

VALUES SEEN	THE TABLE STORES
24	24 values
THE RULE STORES	ON NEW DATA
2 numbers	nothing new yet

FIG. 3.12 Drag the amount of data up and watch the two boxes. The table stores every value it has seen, and grows with the data. The rule is two numbers, the same two the predictor in Figure 3.3 found, and it does not grow at all. Then press New data and see which of the two has anything to say about it.

Where it stops being enough

There are two limits.

Predicting well is not the same as being useful. A system can be excellent at continuing a recording and still hand you nothing you can act on. Saying what comes next is a different question from saying what would come next if you did something different. Only the second supports a decision. Li's network carries a board and was still only ever asked which moves were legal.

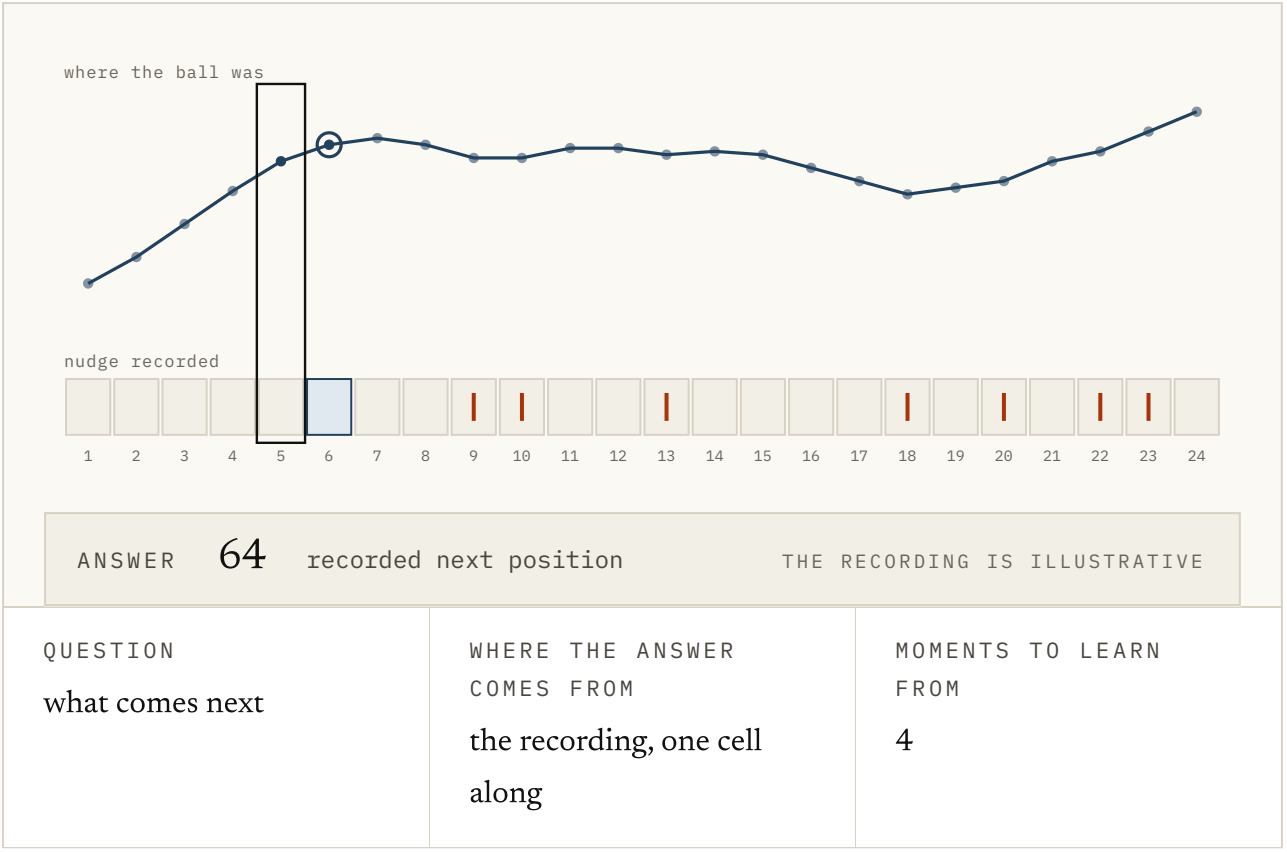
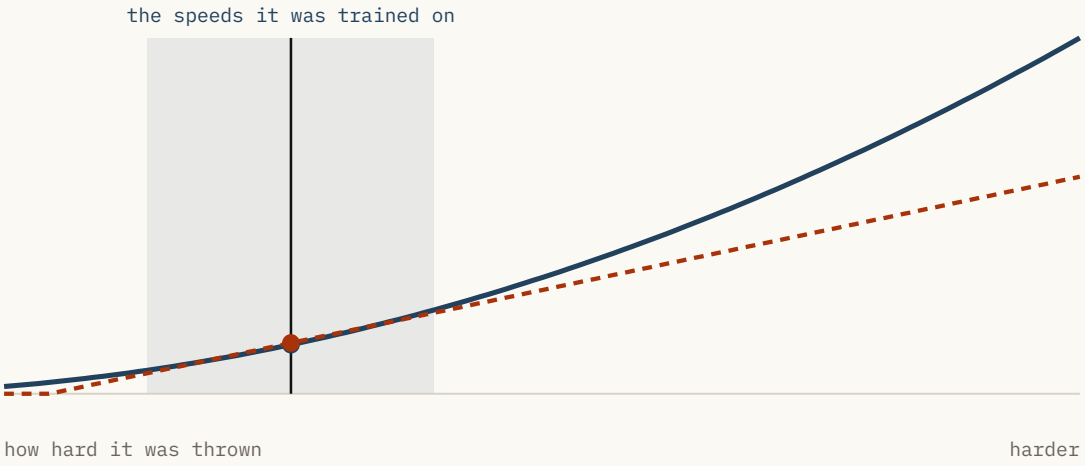


FIG. 3.13 Drag along the recording and pick a moment. Press What comes next and the answer is the very next cell, already there. Press What if I nudge it and, unless somebody happened to nudge at that moment, there is no cell to read: the answer has to be borrowed from other moments where the ball was in about the same place and somebody did, and sometimes there are none. The recording is illustrative.

Being unsurprised is not the same as being right. A predictor that has only ever seen calm weather is not surprised by calm weather. That tells you nothing about what it would do in a storm.

Low error on what you happened to record is a weaker claim than it sounds, and it is the claim most headline numbers make. Let's see it with a ball instead of weather.



WHERE IT REALLY LANDS	WHERE THE MODEL SAYS	OFF BY
69 m	71 m	4%

FIG. 3.14 Drag the launch speed across the band the model was trained on and it lands close to the truth, so anyone testing it there would say it has the rule. Keep dragging past the band and watch the gap open. The calm days were the band, and it never saw a storm.

Neither limit undoes the argument. Prediction is enough to pull structure out of raw experience, and it is the cheapest signal anyone has found. The line from Shannon through Elman to Othello-GPT is people proving that with better instruments.

Everything here has assumed that what is being predicted is small: two numbers for a ball, and one word at a time for language. A camera hands you two million numbers, thirty times a second, and almost all of them are the leaves. That is the leaf problem arriving at full size, and it is where chapter 4 picks up.

Hopefully this chapter helped. There is a short quiz below if you want to check that the chain held together for you. If I have got something wrong, or you know a better source than the ones I used, the About page has a corrections section, and I would be glad to hear from you.

Try it

1 One photograph of a ball in flight. What can you not work out from it?

a. Where the ball is
b. How big the ball is
c. Which way it is going and how fast
d. What colour it is

> ANSWER c. Which way it is going and how fast. Direction and speed are not in any single frame. They exist in the relationship between frames, which is the kind of thing a predictor has to build for itself.

2 Why is next-thing prediction so much cheaper to train on than labelled data?

a. The models are smaller
b. The answer is already in the recording, so nobody has to write labels
c. It needs less computing time
d. It converges in fewer steps

> ANSWER b. The answer is already in the recording, so nobody has to write labels. Every moment is the answer to the moment before. That turns any recording of anything into a training set, and removes the step where a person has to annotate a million examples.

3 Jeffrey Elman's network was trained only to predict the next word, and its internal states sorted themselves into nouns, verbs, animate and inanimate. Why?

a. Those categories were in the training labels
b. The architecture had one unit per category
c. A word's category is what its next word depends on, so the categories were the cheapest way to be less wrong
d. It memorised the corpus

> ANSWER c. A word's category is what its next word depends on, so the categories were the cheapest way to be less wrong. Nothing supplied the categories. Prediction rewards whatever makes the next thing less surprising, and for words that is grammatical category.

4 A probe finds board state inside a network trained only on legal Othello moves. What turns that from a curiosity into a finding?

a. The probe is very accurate
b. The network was large
c. Changing that internal board changes the moves the network then makes
d. The board can be drawn as a picture

> ANSWER c. Changing that internal board changes the moves the network then makes. Accuracy alone could be a coincidence in the numbers. Intervening on the representation and watching behaviour follow is what shows the network is using it.

5 Under a good predictor, a symbol you were confident about and got right costs almost nothing to send. Why?

a. It can be left out of the message
b. The cost of a symbol is set by the probability you gave it
c. Common symbols are stored in a table
d. The receiver guesses it and does not need the message

> ANSWER b. The cost of a symbol is set by the probability you gave it. Claude Shannon made the price exact. High probability means a short code, which is why a better predictor is a better compressor.

6 What does the compression view say about memorising the training data?

a. It is the best available strategy
b. It is the expensive option: a lookup table of everything is enormous and useless on anything new
c. It compresses better than any rule
d. It is what all learning does

> ANSWER b. It is the expensive option: a lookup table of everything is enormous and useless on anything new. The short message comes from finding the rule that generated the data and keeping that instead. Memorising is what compression penalises.

7 Same loop, different target: predict the next pixel, or the next word, or the next compact state. What does that choice change?

a. Nothing, the loop is what matters
b. Only how long training takes
c. What the system ends up good at, and what its capacity gets spent on
d. Whether the method counts as self-supervised

> ANSWER c. What the system ends up good at, and what its capacity gets spent on. Predict every pixel and most of the capacity goes to leaves, because that is where most of the pixels are. Picking the target is the design decision that matters most.

8 A model has very low error predicting the next frame of a recording. What does that alone not tell you?

a. That it saw the recording
b. What it would predict if you acted differently, and how it behaves outside what it recorded
c. That its error was measured correctly
d. That the recording was long enough

> ANSWER b. What it would predict if you acted differently, and how it behaves outside what it recorded. Being unsurprised by what you happened to record is a weaker claim than it sounds, and saying what comes next is not the same as saying what would come next under a different action.

FIG. 3.15 Eight questions on the loop and what it produces. The last two are the ones that separate a good predictor from a useful one, and they are the ones I'd most like you to get right.

Sources

A Mathematical Theory of Communication SHANNON, 1948 ↗

Where the price of a symbol becomes the probability you gave it. Everything about prediction and compression being one job starts here.

https://people.math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf

Prediction and Entropy of Printed English SHANNON, 1951 ↗

Claude Shannon sat people down and had them guess English one letter at a time. The bottom row of Figure 3.10 is roughly what he measured.

https://doi.org/10.1002/j.1538-7305.1951.tb01366.x

Finding Structure in Time ELMAN, 1990 ↗

Train a small network to predict the next word, then look inside: nouns, verbs, animate and inanimate, none of it asked for.

https://doi.org/10.1207/s15516709cog1402_1

Emergent World Representations LI ET AL., 2022 ↗

A network given nothing but legal Othello moves, with the board found inside it and causally manipulated.

https://arxiv.org/abs/2210.13382

Othello-GPT has a linear emergent world representation NANDA ET AL., 2023 ↗

The follow-up that sharpened what the probe was actually reading.

https://arxiv.org/abs/2309.00941

Language Models are Unsupervised Multitask Learners RADFORD ET AL., 2019 ↗

Next-word prediction, scaled, turning into capabilities nobody trained for. The strongest evidence that the target choice is the design decision.

https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

The Hutter Prize HUTTER, ONGOING ↗

A cash prize for compressing a snapshot of Wikipedia, run on the argument that you cannot squeeze text further without modelling what it means.

http://prize.hutter1.net/

Representation Learning: A Review and New Perspectives

BENGIO, COURVILLE & VINCENT, 2013 ↗

The survey that laid out what a good learned representation is for, written before prediction had finished winning the argument.

https://arxiv.org/abs/1206.5538

Formal theory of creativity, fun, and intrinsic motivation SCHMIDHUBER, 2010 ↗

Compression progress as a drive in its own right: not just a way to measure a model, but a reason to go and look at something.

https://people.idsia.ch/~juergen/creativity.html

FIG. 3.16 I've ordered these for someone building on this rather than for historical completeness.