

3

The Regression Theory of Everything

3.1 Let's Avoid Knowledge Representation!	12
3.2 A Simple Neural Network is also a Linear Regression . . .	13
3.3 Dummy Variables for Dummies (Wonkish)	13
3.4 Try That Again With A Few Billion Parameters	15
3.5 Multicollinearity and the End of Science	16
3.6 Let's Test Some Random Inputs! Feature Importance and Explainability	17
3.7 The Universal Machine Learning Workflow	17
3.8 A Machine Learning Engineer is a Data Janitor	19
3.9 Key Takeaways	20



"mdjrny-v4 a disembodied computer being force-fed data like a duck looking somewhat like a sci-fi character 8k" made with Mann-E

"AI Scientists disagree as to whether these language networks possess true knowledge or are just mimicking humans by remembering the statistics of millions of words. I don't believe any kind of deep learning network will achieve the goal of AGI [Artificial General Intelligence] if the network doesn't model the world the way the brain does. Deep learning networks work well, but not because they solved the knowledge representation problem. They work well because they avoided it completely, relying on statistics and lots of data instead. How deep learning networks work is clever, their performance impressive, and they are commercially valuable. I am only pointing out that they don't possess knowledge and, therefore, are not on the path to having the ability of a five-year-old child." Jeff Hawkins, 2022 [15]

3.1 Let's Avoid Knowledge Representation!

The knowledge representation problem in AI is the challenge of how to formally represent knowledge in a way that a computer can understand and reason about. This typically involves creating a set of symbols, rules, and structures that can be used to represent concepts, relationships, and other types of information. The goal is to create a representation that is both expressive enough to capture all relevant aspects of the domain, and computationally tractable enough to allow for efficient reasoning and inference. There are many different approaches to knowledge representation, including logic-based, semantic networks, frames, and ontologies, each with their own strengths and weaknesses.

Deep learning techniques handle knowledge representation differently than traditional symbolic AI methods. Unlike symbolic AI which relies on explicit and hand-coded representations of knowledge, deep learning techniques learn to represent knowledge implicitly through the use of neural networks.

In deep learning, knowledge is represented in the form of the weights of the neural network. These weights are learned through training on a large dataset and they capture the underlying relationships and patterns in the data. The neural network can then use these learned weights to make predictions, classifications, or generate new data.

Deep learning models can handle large and complex datasets, and can automatically extract features from the data without the need for manual feature engineering. This makes them particularly well-suited for tasks such as image and speech recognition, natural language processing, and other areas where large amounts of data are available. However, they are not as good at explicating how they arrived at a decision, which can be a disadvantage.

In summary, deep learning techniques handle knowledge representation by learning the underlying patterns and relationships in the data through the use of neural networks, which can then be used for prediction, classification, and generation tasks. In GOFAL, knowledge is held by the programmer and explicitly coded into rules, while deep learning methods instead use data to guess the best rules from the relationships present in the dataset.

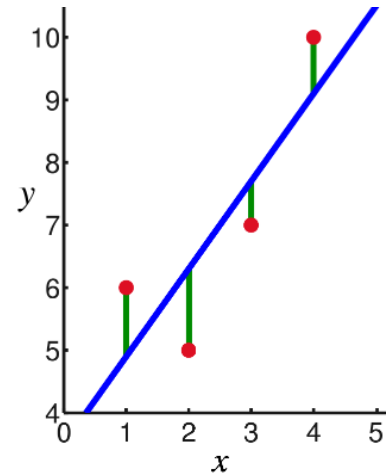
3.2 A Simple Neural Network is also a Linear Regression

A neural network can be mathematically equivalent to a regression or a decision tree under certain conditions.

A neural network is a machine learning model composed of layers of interconnected artificial neurons, which are designed to process and analyze data. They can be used for a wide range of tasks, such as image and speech recognition, natural language processing, and prediction.

A regression is a statistical method used to predict a continuous variable based on one or more input features. A linear regression, for example, is a simple neural network with one input layer, one output layer, and no hidden layers. In this case, the weights of the network are the coefficients of the linear equation and the network is equivalent to a linear regression model.

A decision tree is a tree-based model used for classification and prediction tasks. It consists of a series of if-then rules that are used to make decisions based on the input data. A neural network with one input layer, one output layer, and one hidden layer¹ is equivalent to a decision tree. The network implements a piecewise linear function which can represent the decision boundaries of a decision tree. So, under certain conditions, a neural network can be mathematically equivalent to a linear regression or a decision tree. These conditions include having one input and one output layers, and having a specific activation function in the case of a decision tree[16].



A simple linear regression, the red points are the training data, and the blue line is the regression line. If you don't understand this please read https://en.wikipedia.org/wiki/Regression_analysis

1: One with a ReLU activation, Neurons in a neural network can have different activation functions, if you don't know what it is and don't want to read about it on wikipedia, that's OK. https://en.wikipedia.org/wiki/Activation_function

[16]: Aytekin (2022), *Neural Networks are Decision Trees*

3.3 Dummy Variables for Dummies (Wonkish)

Chapter Summary: It's all numbers, man. Machine learning techniques require that we turn everything (images, text, sound) into numbers and shove them into the model in the same way we use dummy variables in a simple regression. If you are satisfied with this, please skip this section. If you would like to learn a bit about the details and see some code examples, please keep reading. This section is necessarily technical, but should be approachable for anyone who has taken a college statistics class.

Dummy variables are used in regression analysis to include categorical variables in a model. Categorical variables are variables that take on a finite number of distinct values, such as "red", "green", "blue", "yes", or "no". Since

these variables cannot be directly included in a regression model as they are not numerical, they need to be transformed into numerical variables.

The process of creating dummy variables is also known as one-hot encoding. It involves creating a new binary variable for each category of the original variable. For example, if you have a categorical variable "color" with three categories: "red", "green", "blue", you would create three binary variables: "color_red", "color_green", "color_blue". Each binary variable would take a value of 1 if the original variable is equal to the category; otherwise, it would be a value of 0.

When using dummy variables in a regression, it is important to remember to include only $n-1$ binary variables where n is the number of categories in the original variable. This is because including all n binary variables would result in perfect multicollinearity, which is when two or more independent variables are perfectly correlated. One of the binary variables can be dropped to avoid this problem.

Dummy variables are used in regression analysis to include categorical variables in a model. The process of creating dummy variables involves creating a new binary variable for each category of the original variable and one-hot encoding it. It is important to remember to include only $n-1$ binary variables to avoid perfect multicollinearity.

The creation of dummy variables in a regression is analogous to preprocessing image, text, and other data for a neural network for deep learning. This preprocessing is important as it ensures that the data is in a format that can be easily understood and processed by the network. The preprocessing steps for numbers, text, and images are slightly different.

For numbers:

- Normalization: It is common to normalize the input data by scaling it to have a mean of 0 and a standard deviation of 1. This helps to ensure that all input features have similar scales and prevents any one feature from dominating the network's computations.
- Imputation: Handling missing data is important, as it can negatively impact the model's performance. Common imputation techniques include replacing missing values with the mean, median, or mode of the feature.

For text:

- Tokenization: Text data must first be converted into a numerical format that can be understood by the network. This is typically done by tokenizing the text into individual words or n -grams and then encoding them as integers or real-valued vectors. A one-hot encoding exactly like the dummy variable method used in regression is also frequently used.^{2 3}
- Stop-words removal: The most common words in any language like "a", "an", "the", etc. that do not contain much meaning are called stop-words, they are often removed to reduce the dimensionality of the data.

2: Sometimes text is just mapped to a number! Shocking, but it works. See how it is taught in the TensorFlow tutorials https://www.tensorflow.org/text/guide/word_embeddings

3: GPT-3 uses byte-level Byte Pair Encoding (BPE) tokenization and has a vocabulary size of 50,257.

- ▶ **Stemming/Lemmatization:** Words that have the same meaning can be stemmed or lemmatized to reduce the vocabulary size and increase the chances of generalization.
- ▶ **Vocabulary Size:** Each model must choose a vocabulary size or the maximum number of tokens that it will analyze. This may cause misspellings, slang or typos to be discarded in analysis.

For images:

- ▶ **Converting to RGB or Greyscale:** Each image is analyzed by its pixel color value, every point on an image will either have 3 color values (red, green, blue) or one single value (on a white/black scale) if the image is analyzed in greyscale.
- ▶ **Convolutions⁴:** Pixel values are analyzed in groups that are defined by the model, since individual pixel values are only colors (or greyness) they must be combined together by the model to detect patterns like faces and stop signs. The method of convolution is defined by the model itself.
- ▶ **Resizing:** neural network can only accept images of a fixed size, so resizing the image to match the network's requirements is important.
- ▶ **Normalization:** It is common to normalize the pixel values to be in the range of 0-1 or -1 to 1. This will help the model converge faster.
- ▶ **Data Augmentation:** To increase the amount of data and prevent overfitting, common data augmentation techniques such as flipping, rotation, and cropping can be applied to the images.

In summary, preprocessing is an important step in training a neural network as it ensures that the data is in a format that can be easily understood and processed by the network. The preprocessing steps for numbers, text, and images involve normalization, imputation, tokenization, stop-words removal, stemming/lemmatization, resizing, and data augmentation.

3.4 Try That Again With A Few Billion Parameters

In the first section of this chapter, we introduced the idea that a simple neural network is mathematically equivalent to a regression. This is true and a useful way to think about neural networks and deep learning⁵. In practice, a neural network trained on a gaming PC from 2023 can easily have millions of trainable parameters. A simple image classifier used to classify handwritten digits might have 60,000 parameters, which is about the number that was used in the model used by Yann LeCun and company in the 1990s. Bleeding-edge deep neural networks are trained on large GPU farms and have billions of trainable parameters. More parameters gives the models tremendous power to recognize complex patterns, but it also makes any practical attempt at explaining their inner-workings impossible.

When a neural network has a million (or more) trainable parameters and deep layers of neurons⁶, it can be difficult to explain to a human what each of those parameters represents or how they contribute to the overall function

4: If you would prefer a visualization of a neural network check out this excellent video by Dennis Dmitriev <https://www.youtube.com/watch?v=3JQ3hYko51Y>.

5:

$$y = Wx + b \quad (3.1)$$

There's your simple formula for a single-cell neural network and regression, in practice we'll change W , x and b to matrices and introduce nonlinear activation function f so,

$$y = f(Wx + b) \quad (3.2)$$

if you please. If equations scare you, don't worry about it for now.

6: In case you are wondering, the human brain has about 86 billion neurons that do the "thinking". Physical neurons are more complicated than the neurons used in deep learning and have supporting structures like astrocytes that do some computation as well. <https://en.wikipedia.org/wiki/Astrocyte>

7: Mathematical chaos is a real thing, it refers to the behavior of certain dynamic systems that are highly sensitive to initial conditions. This sensitivity leads to seemingly random and unpredictable behavior, even though the underlying equations governing the system are deterministic (meaning they are transparent and known to us). https://en.wikipedia.org/wiki/Chaos_theory

8: Chaotic outputs in a large deterministic system are OK in many domains, and there are controls that can be put in place to keep AI "safe" but deep learning by itself will not produce those controls.

9: "Multicollinearity is when two or more variables in a study are closely related to each other. Think of it like dating: imagine you are trying to figure out what makes someone a good partner. You might look at things like how much they make, how attractive they are, and how kind they are. But, these things are all related to each other. For example, attractive people might make more money, and kind people might be more attractive. So, it's hard to know which one is really important for being a good partner, because they are all related. That's like multicollinearity in a study." By ChatGPT given the prompt "explain multicollinearity to a teenager using a dating example"

of the network. This is because the interactions between the different layers and neurons can be complex and nonlinear, thereby making it challenging to understand the specific role of each parameter. The parameters model's complex interactions between the inputs makes it difficult to understand their specific function. Furthermore, in deep neural networks, the high number of layers can lead to a high level of abstraction, meaning that the individual neurons and their weights have little interpretability.

When a neural network has many interacting parameters, it can lead to mathematical chaos⁷, which is a phenomenon where small changes in the initial conditions of the network can lead to vastly different outputs. This is because the interactions between the large number of parameters can create non-linear relationships that are sensitive to small changes. This can make it difficult to predict the behavior of the network, as small changes in the input or the parameters can lead to unexpected and seemingly random outputs⁸.

3.5 Multicollinearity and the End of Science

Data science is a horrible term because it implies that data scientists are scientists and that the work they do is scientific, when in reality data scientists are not scientists and the work they do is not scientific. Data scientists use mathematics, statistics, and computer science to analyze data, but it is not scientific in the traditional sense. Data scientists do not use the scientific method and do not conduct experiments or develop theories. Data science is more akin to engineering or data analytics than actual scientific research.

The main goal of a scientist is to gain knowledge and understanding of the natural world through research, experimentation, and data analysis. Scientists make models of the world to test and explain. So-called data scientists make models too, but a model with layers of interacting parameters trained under chaotic conditions is almost impossible to explain. Multicollinearity⁹ is just one issue, but deep learning techniques are practically incompatible with the scientific method. With deep learning methods, it can be difficult to determine which variable had the most impact on the outcome of the model. Coefficients of the model can be unstable and unreliable, which can make it difficult to explain the model in a meaningful way.

Explaining the inner-workings of a neural network with millions of interacting parameters is difficult for many reasons. Most of the complexity is due to the number of parameters, which can make it difficult to understand how the model works and why it makes certain decisions. The interactions between the parameters are often non-linear and can be difficult to understand as the model can be sensitive to small changes in the parameters, which is the definition of mathematical chaos. This can make it difficult to explain why the model is making certain decisions as the interactions between the parameters are not always clear.

3.6 Let's Test Some Random Inputs! Feature Importance and Explainability

With a huge complex system of neurons that combine inputs in novel ways, it becomes very hard to understand which inputs are the most important for the system as a whole. To better understand deep learning models, data scientists use randomized or averaged inputs to a model to test the feature importance of a deep learning neural network. This type of testing is done to determine which features are most influential in the overall output of the network. This is done by randomly or averaging the input values for each feature and then running the model to see how the output is affected. For example, if a neural network is used to detect objects in an image, the data scientist may randomize the color of the objects to see how the model's accuracy is affected. If the accuracy drops significantly, they can infer that color is an important feature in the network.

Randomized or averaged inputs to a model can also be used to determine if a particular feature is necessary for the network to function correctly. For example, if the output of the network is not as accurate when a particular feature is randomized or averaged, then the data scientist can infer that this feature is important for the network's performance. By using this method, data scientists can gain insights into which features are important and which can be removed from the model to improve performance.

3.7 The Universal Machine Learning Workflow

The Universal Machine Learning Workflow is an important chapter in a technical guide for data scientists by the co-author of Tensorflow, the most popular machine learning framework in the world as of 2023, that outlines the *Universal* workflow for machine learning projects. I think this chapter should be understood by everyone using, investing in, and creating machine learning models.

Before Chollet gets into the details of model building, he chooses to begin with a note on ethics:

"You may sometimes be offered ethically dubious projects, such as "building an AI that rates the trustworthiness of someone from a picture of their face." First of all, the validity of the project is in doubt: it isn't clear why trustworthiness would be reflected on someone's face. Second, such a task opens the door to all kinds of ethical problems. Collecting a dataset for this task would amount to recording the biases and prejudices of the people who label the pictures. The models you would train on such data would merely encode these same biases into a black-box algorithm that would give them a thin veneer of legitimacy. In a largely tech-illiterate society like ours, "the AI algorithm said this person cannot be trusted" strangely appears to carry more weight and objectivity than "John Smith said this person cannot be trusted," despite the former being a learned approximation of the latter. Your model would be laundering and operationalizing at scale the worst aspects of human judgement, with negative effects on the lives of real people. [17]



"the face of a trustworthy person" made with Stable Diffusion 2.1. (Hey, it's a white lady!)

Technology is never neutral. If your work has any impact on the world, this impact has a moral direction: technical choices are also ethical choices. Always be deliberate about the values you want your work to support.”[17]

Chollet uses the outline below to explain the *Universal* workflow. I’ll summarize the workflow and my own notes for a nontechnical audience here:

1. Define the Task

- a) Collect a Dataset ¹⁰
- b) Understand Your Data
- c) Choose a Measure of Success ¹¹

2. Develop a Model

- a) Prepare the Data ¹²
- b) Choose an Evaluation Protocol
- c) Beat a Baseline ¹³
- d) Develop a model that overfits
- e) Regularize and Tune Your Model

3. Deploy the Model

- a) Explain Your Work to Your Stakeholders and Set Expectations ¹⁴
- b) Ship an Inference Model
- c) Monitor Your Model in the Wild
- d) Maintain Your Model

10: This is often the hardest part. Data is often labeled by hand or stolen from another domain. See https://en.wikipedia.org/wiki/Transfer_learning

11: Accuracy isn’t everything! We might prefer a less accurate model that avoids false-positives in some scenarios.

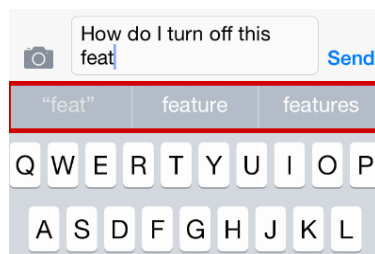
12: This is discussed in section 3.3 above, everything gets turned into numbers.

13: Does your model predict better than a totally random model and/or coin-flip?

14: This is almost never done. And as Chollet explains this is both difficult and requires mutual understanding and domain knowledge from the part of “the business” and the Machine Learning Engineer.

Of all of the steps in the workflow, “Defining The Task” and “Explaining The Work To Shareholders and Setting Expectations” are where the most miscommunication occurs.

In defining the task, machine learning engineers are often given impossible problems to solve, and because they want to keep paying their mortgage, they solve another related problem instead and allow stakeholders to jump to their own conclusions. By clearly understanding what a model is predicting and how the data is collected, some of this misunderstanding can be avoided.



Fundamentally LLMs are using the same techniques as a predictive keyboard on your phone. This is why Yann Lecun says even though they are impressive some like ChatGPT are not particularly innovative. <https://www.youtube.com/watch?v=ULbpPHjiSBg>

I will discuss Large Language Models later, but they all do the same thing. They predict the next words in a sentence, just like the keyboard on your iPhone. They come up with amazing text, but by fundamentally understanding what they are predicting, you gain insight into their limitations. They are also all trained on the text publicly available on the internet; this includes scientific sources, but also fan-fiction and anime, so the idea that a model trained on this data could be relied on to predict anything truthful is preposterous.

Many models work like magic and many users assume models are predicting the future, when they are really correlating based on their past data. Even simple models of creditworthiness might have a similar problem. While building a creditworthiness model for a bank, due to lack of data, a machine learning engineer might (on purpose or inadvertently) create a model that predicts whether someone is a smoker instead of whether they are creditworthy. Because smoking and poverty are correlated, maybe the model “works”, but it doesn’t do what stakeholders think it does.

Setting expectations is another hellscape of misaligned incentives. Good machine learning engineers and data scientists are supposed to educate and advise stakeholders about the limitations of their own models. Read Chollet's advice on this topic below:

The expectations of non-specialists towards AI systems are often unrealistic. For example, they might expect that the system "understands" its task and is capable of exercising human-like common sense in the context of the task. To address this, you should consider showing some examples of the failure modes of your model (for instance, show what incorrectly classified samples look like, especially those for which the misclassification seems surprising).

They might also expect human-level performance, especially for processes that were previously handled by people. Most machine learning models, because they are (imperfectly) trained to approximate human-generated labels, do not nearly get there. You should clearly convey model performance expectations. Avoid using abstract statements like "The model has 98 percent accuracy" (which most people mentally round up to 100 percent), and prefer talking, for instance, about false negative rates and false positive rates. You could say, "With these settings, the fraud detection model would have a 5 percent false negative rate and a 2.5 percent false positive rate. Every day, an average of 200 valid transactions would be flagged as fraudulent and sent for manual review, and an average of 14 fraudulent transactions would be missed. An average of 266 fraudulent transactions would be correctly caught." Clearly relate the model's performance metrics to business goals.

You should also make sure to discuss with stakeholders the choice of key launch parameters- for instance, the probability threshold at which a transaction should be flagged (different thresholds will produce different false negative and false positive rates). Such decisions involve trade-offs that can only be handled with a deep understanding of the business context.[17]

One does not need to be a psychologist to understand that these conversations almost never happen. The workflow of machine learning is universal; we are all doing fancy regressions and we all deal with the same wild expectations, bad data and misalignment with business.

3.8 A Machine Learning Engineer is a Data Janitor

Since the machine learning workflow is *universal*, it is fairly straightforward to automate. There are a plethora of offerings from companies big and small who offer AutoML tools that anyone with basic Excel skills can use¹⁵. Since modern AI is "all a regression", these tools do the regression for you. Users just need to feed them the data and the AutoML tool finds out the relationship. On the surface, it seems like the job of the machine learning engineer has become obsolete before it even became a proper discipline.

Despite the existence of AutoML tools, machine learning engineers (and data scientists) do some actual work. A 2016¹⁶ CrowdFlower survey of



"mdjrny-v4 a handsome businessperson explaining the business context to a scientist wearing a white coat over coffee 8k" made with Mann-E

15: See this awesome-AutoML github project for an up-to-date list of commercial AutoML tools <https://github.com/windmaple/awesome-AutoML#commercial-products>

16: I know this study is old, but I have found that the findings still hold when talking to data scientists and/or machine learning engineers that I train and employ.

16,000 data scientists showed that on average we spend our time doing the following:

- ▶ **60 percent of a machine learning engineer's time is spent cleaning and organizing data** - we are data janitors! If you had great data ready in an Excel file, you would have been able to fit your own model without us. We need to organize the mess of data that exists in the world so it is in a nice format to fit in a model. For example, it is well known that almost all large language models use the same common crawl dataset (<https://commoncrawl.org/>), but how it is organized and weighted can change the quality of the outputs dramatically. The organization of the same dataset can lead to the same dataset either producing a nice predictive keyboard or something approaching ChatGPT.
- ▶ **19 percent of a machine learning engineer's time is spent collecting data.** We often start making models without a huge dataset, so data needs to be collected in order to make the first model, and after that first model (to match people for our new dating app, let's say) is deployed, we rely on users to provide us with data for future models. Collecting new data is a creative endeavor sometimes, and often involves human effort. Amazon's Mechanical Turk¹⁷ service has been leveraged for this purpose for years, and OpenAI has successfully deployed outsourced talent to solve some of the trickiest problems facing large language models¹⁸.
- ▶ **9 percent of a machine learning engineer's time is spent fitting models.** We need to fit models; there is some art to choosing the right architecture and modeling techniques. An AutoML tool can do this reasonably well, too, but considering some fancy training servers cost \$28 per hour or more¹⁹, sometimes it is useful to have an expert guess the best model architecture and save time in training, even if that expert is getting paid \$249,000 per year²⁰.
- ▶ **4 percent of a machine learning engineer's time is spent refining algorithms.** Many machine learning engineers spend almost no time doing this. Other researchers (corporate-academic types who write scientific papers) spend a lot of time doing this. It averages out to four percent but doesn't represent the average day of a machine learning engineer.

Overall, the main job of most machine learning engineers is to clean up and collect a ton of data, and then feed that data into the same machine learning model that everyone else is using; and then rinse and repeat²¹.

3.9 Key Takeaways

- ▶ **Deep learning models are fundamentally large unscientific regressions.** They are trained to create a function that maps input data to output data.
- ▶ **Deep learning models are chaotic systems containing millions of interacting parameters.** They are not designed to be explained or

17: "Getting the right training data was another challenge. ImageNet was a collection of one hundred thousand labeled images that required significant human effort to generate, mostly by grad students and Amazon Mechanical Turk workers." <https://arstechnica.com>

18: Read more about how OpenAI used Kenyan workers to help label toxic content <https://time.com/6247678/openai-chatgpt-kenya-workers/>.

19: See the latest pricing from AWS here <https://aws.amazon.com/sagemaker/pricing/>

20: See the latest machine learning engineer salaries at levels.fyi <https://www.levels.fyi/t/software-engineer/focus/ml-ai>

21: Here is a discussion on reddit asking "When was the last time you wrote a custom neural net?" and supports my understanding of the world https://www.reddit.com/r/MachineLearning/comments/yto34q/d_when_was_the_last_time_you_wrote_a_custom/

created in such a way that their weights can be used for scientific analysis. They find reasonable answers and don't care how they get there. Multicollinearity (understanding the relationship of an input and output) and feature importance (understanding which inputs are most important) are only understandable with a high level of statistical error.

- ▶ **Small changes in inputs of a deep learning model may dramatically change the outputs.** Deep learning models are complex deterministic systems that can exhibit chaotic behavior. Their inner workings are functionally unknowable and practically impossible to test.
- ▶ **Machine learning engineers spend most of their time collecting and organizing data.** Because deep learning models often share common architecture, getting good data to train with is the best thing that an engineer can do to train good models. In practice, only a handful of corporate-academic types are experimenting with new and exciting architectures.
- ▶ **Deep learning models can have impressive and useful outputs, but the creators of models should be encouraged to highlight their failures and limitations.** Machine learning engineers might be more keen to highlight failures and limitations if they are encouraged to do so by their users, managers, and investors.