Using Pretrained Models to Generalize Different Tasks

Christian DerManuelian
cdermanuelian@ucsd.edu

March 24, 2024

Abstract

AI is popular because of its ability to learn patterns without explicitly being told how. To take it a step further, what if it could learn to solve a variety of problems without being told how? This is the motivation behind the sub field called AGI, or Artificial General Intelligence. Having our machine learning algorithms able to abstract their findings to a general variety of tasks would be extremely useful for training time, development effort, and could push us closer to modeling the human mind's complexity using computers. But does our current framework of AI even possess the possibility of achieving this?

1 Introduction

What are the first steps to achieving AGI? We must analyze whether the current framework of AI is even suited for generalizable tasks. For example, take AI designed to play a certain category/genre of video games. Say we train a model on one game, and it gets very good at it. If we take that same model and run it on a different game, will it outperform the same model but in a random state? This is the core of understanding a necessary skill AGI systems should have. This is a complex example, so how can we test this with much simpler datasets?

2 Experimental Design

2.1 Data

To answer the above question, we will use an example dataset acquired from here.

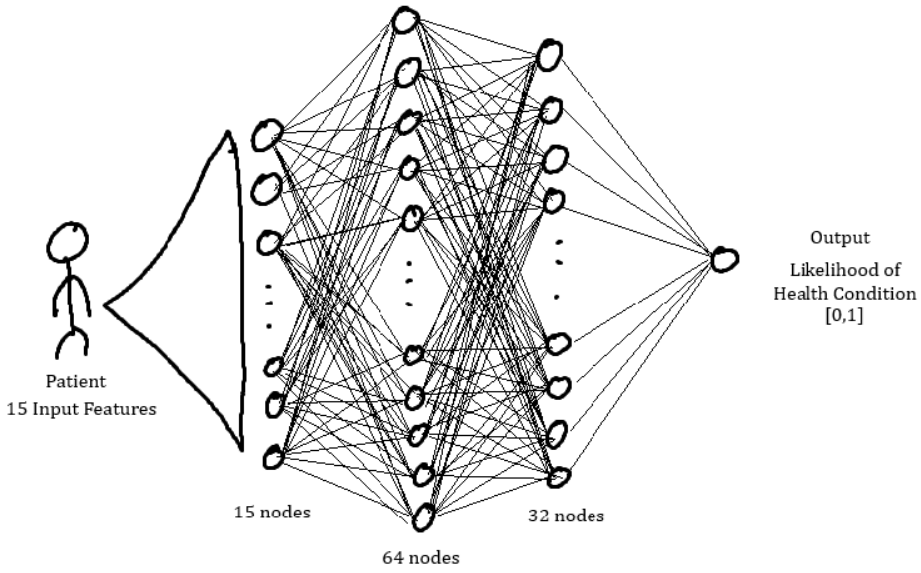
The Osteoporosis Risk Prediction dataset contains health-related data for 2000 individual patients. The data contains features such as patient age, do they smoke, gender, hormonal changes, etc. All of which are meant to predict whether a patient will have Osteoporosis. Though, the topic of this paper ignores Osteoporosis entirely.

But that's the point of the data? Yes it is, and it is a good dataset when suited for that task. In order to get valuable insights on AGI, we need something generalizable. The dataset also includes a feature for the patients medical conditions. More specifically, each patient either has Rheumatoid Arthritis, Hyperthyroidism, or is healthy. Now we turn this into a tri-prediction problem. How can we use the same set of input data to make this three-way prediction. We could just use one model, but discussed much later is why we do not. Instead, we would have to train three separate "regression" models. The reason for not simply making one large model is that a larger model demands more training time. Moreover, our goal is to gain insights into generalization, which is nullified by using one model.

So not only does this paper aim to explore AGI possibilities, but it also acts a test to see what methods in Machine Learning could vastly reduce training time. But, how do we test this with such a simple example?

2.2 Network A

Assume we have network architecture A that can predict any of the three target health conditions.



We could train three separate networks, or the much simpler and efficient method of combining all three outputs into one layer, but we should note something interesting. If we have three separate networks, they have in common the fact that the hidden layers are learned latent features. We will focus on the second hidden layer with 32 nodes.

Even if the parameters are different when trained separately, if the data we are interested are similar, then the latent representation of each patient will hold useful information between the three cases. Since the target outputs are all health conditions, and our features are health-related data, then each data point exists in some high-dimensional abstract space suited for predicting health-related information.

2.3 Network B

Then what if we only train one model? We train one model on one target output (Rheumatoid Arthritis) to obtain a network. Then we use this network up until the last hidden layer. More specifically, we take our initial

data, and pass each observation into the model and save the 32-dimensional representation as our new latent features. Then we train models using that latent representation as the input. Now, the new models can be much simpler and easier to train. But will the accuracy hold?

We will call this architecture B. This network takes in 32 input features and has no hidden layers. It is simply a linear combination of those features into one output node for the health condition we want to predict.

3 Methods

Let's put this to the test. Step 1 is to train model A to see how it performs. The architecture was selected through various trials of different number of layers, different number of nodes per layer, and testing ReLU vs. Sigmoid activations. Since most of the data is in One-Hot-Encoded style, it happened that Sigmoid performed better. Of course, making a binary prediction, Binary CrossEntropy was the ideal loss function. After various trials of fine tuning, these were the results.

- Predicting Rheumatoid Arthritis (Network A)
  - Training Time: 2.56 seconds
  - Testing Accuracy: 65%
- Predicting Hyperthyroidism (Network A)
  - Training Time: 2.68 seconds
  - Testing Accuracy: 70%
- Predicting Healthy (Network A)
  - Training Time: 2.59 seconds
  - Testing Accuracy: 62%

Given a lot of optimization time and fine-tuning of hyperparameters, this was the best recorded testing accuracy found for this dataset.

As for the interesting part, we will explore how training just one of these models allows for accurate generalizations for the others. We will take the model predicting Rheumatoid Arthritis as our principal model. We saved

the model state, and then loaded it in to transform our original data from 15-dimensional feature space to 32-dimensional latent feature space. I.e., ignoring the last layer and executing it to produce the last hidden layer. Since this is just an execution, this part of the process ran almost instantly.

Time to introduce model B. Model B will take our data in 32-dimensional latent space and use one layer to predict each of the three health conditions. The results are shown below.

- Predicting Rheumatoid Arthritis (Network B)
  - Training Time: 0.77 seconds
  - Testing Accuracy: 63.27%
- Predicting Hyperthyroidism (Network B)
  - Training Time: 0.79 seconds
  - Testing Accuracy: 62.76%
- Predicting Healthy (Network B)
  - Training Time: 0.78 seconds
  - Testing Accuracy: 63.01%

There are multiple insights to unpack here. First of all, relevant to AGI, the accuracy is very nearly as good as when we trained 3 separate models. This trial was run multiple times with different random states and the results were consistent. The accuracy is extremely similar, and in fact more consistent for all three output targets.

Moreover, the training time drastically decreased. A few seconds does not seem like much, but when we think of this problem scaling up in terms of data size and problem complexity, this is huge.

4 Interpretation

Take the beginner approach to a simple Machine Learning problem. Take some features and fine-tune a network to predict some output. Instead, what we have found is that we should take our features, and create some latent-space abstract representation for our observations. Then, we can use that

representation directly to approach different prediction tasks as they come up. For example, we could use this representation for any health condition we want to predict. We could use it to predict what medications we should prescribe to the patient. Again, this is a small scope dataset and we should not be frightened by the low accuracy. This experiment was intended to be simple for the express purpose of singling out this phenomena that allows to make generalizations later.

Of course, this does not prove that our framework of machine learning is perfectly capable of solving AGI tasks, but it points to the idea that it is possible to generalize our networks to similar problems. So long as the problems we want to generalize involve the same input-space scope, we know now that it is worth trying the latent-space-first approach, and then apply that to any future problems we encounter.

4.1 Potential Issues

Of course this was a very small problem, so we cannot imply the findings are universal. We would have to test this on more complex problems to ensure the results hold across a variety of applications.

Moreover, you would argue that this set of networks could have just been one big network with 3 output nodes. This is true, and it is a much more efficient approach. What we found by splitting up the models was that certain models do have the ability to generalize to new problems. Imagine we only had one target output to begin with. Then we later had the need to predict more models. Instead of retraining one big model, we use the fact that the pretrained latent-space representation is still useful.

4.2 Key Takeaway

A key finding to this approach to revealing potential AGI capabilities involves how we created the latent space. We did not just "build" the representation. Instead, we picked a specific output target and trained the model using that. This means that in order to convert raw feature data into an abstract representation, there needs to be some motivation behind the representation. Is this how our minds represent things in that exist in the world? How does our mind determine what features are important to something? If you were a doctor, and your goal was to diagnose people, you would spend years studying about the human health condition to determine what features about a

patient are useful. If you want to predict someone's salary, you would view society through a lens of a completely different set of features. The point is that our minds do not simply hold representations of things in the world arbitrarily. Rather, we collect stimuli through all of our senses that are relevant to whatever our goals are. This is what gives us the ability to make informed decisions on general tasks. We start by learning whatever we can, and then using that ever-building pool of knowledge to generalize to different tasks. This is why you are always bad at things you are new to, yet still exhibit patterns in your decisions that are consistent with what you have previously learned.

If we find a systematic and well-defined way to have our AI models apply their findings to new tasks, we will have taken a major step towards AGI. The problem in this paper only dealt with a closed-scope problem of healthcare. But what if we have one model that can be tweaked to suit any future task? For example, if we took network A and applied it salaray prediction, it would probably suck. However, it would still hold some insights. Most of the features it has are useless to salary prediction, but columns like age and smoking could potentially aid that new problem.

4.3 What is Next?

The thing humans do that AI doesn't is the input scope. We only ever suit our models to a hand-picked set of features for a problem. Humans however, take in every aspect of our sensory stimuli and past knowledge to attempt to generalize past decision-making skills to new problems. What we have found here is that when the same input scope is applied to different, but similar, problems, AI shows generalizing capabilities. The next step is to apply this on an even larger scale.

Recall that the latent representation of a patient was described to be in a high-dimensional space that stores information relevant to health condition predictions. This ended up being true, as the representation from one health condition was extremely useful to predicting the other two conditions. So is there some even larger space that contains representations for things that can answer any prediction problem? Is that patient-space just a subset for information regarding all possible features of a human? Before this gets overly philosophical, we note that the generalization worked in the small scope feature space. So instead of leaping further, we could steadily scale this problem to see if the findings are still consistent.

Overall, the insight on AGI is to first define the problem space to generalize to. Then work backwards to find what level in the logic must be abstracted. We wanted to predict health conditions, so we need an abstract representation of a patient. Then create an abstraction with one example from the initial problem space. Then use that abstraction to generalize any problem. This is not a well-defined framework, rather a new avenue for reserching this deep field of Artifical Intelligence.