Example: Random Projections of World Models

The Llama-2 model has ingested huge amounts of publicly available data from the internet including Wikipedia dumps from the June-August 2022 period (Touvron et al. 2023). It is therefore highly likely that the training data contains geographical coordinates, either directly or indirectly. At the very least, we should expect that the model has seen features during training that are highly correlated with geographical coordinates. The model itself is essentially a very large latent space to which all features are randomly projected in the very first instance, before being passed through a series of layers which are gradually trained for downstream tasks.

In our first example in this section, we simulate this scenario, stop short of training the model. In particular, we

- Take the world_place.csv that was used in Gurnee and Tegmark (2023), which maps locations/areas to their latitude and longitude. For each place, it also indicates the corresponding country.
- Take the subset that contains countries that are currently part of the top 10 FIFA world ranking.
- Assign the current rank to each country, i.e. Argentina gets 1, France gets 2, ...
- Transform the longitude and latitude data, respectively, as follows: $\rho \cdot \text{coord} + (1-\rho) \cdot \epsilon$ where $\rho = 0.5$ and $\epsilon \sim \mathcal{N}(0,5)$. This is to ensure that the training data only involves a noisy version of the coordinates.
- Next, encode all features except the FIFA world rank indicator as continuous variables: $X^{(n \times m)}$ where n is the number of samples and m is the number of resulting features.
- Additionally, add a large number of random features to X to simulate the fact that not all features ingested by Llama-2 are necessarily correlated with geographical coordinates. Let d denote the final number of features, i.e. $d = m+k$ where k is the number of random features.
- Initialize a small neural network, let's call it a *projector*, that maps from X to a single hidden layer with $h < d$ hidden units and sigmoid activation and then from there to a lower-dimensional output space.
- Without performing any training on the *projector*, we simply compute a forward pass of X through the *projector* and retrieve the activations $\mathbf{Z}^{(n \times h)}$.
- Next, we perform the linear probe on $\mathbf{Z}$ through Ridge regression: $\mathbf{W} = (\mathbf{Z}'\mathbf{Z} + \lambda \mathbf{I})(\mathbf{Z}'\mathbf{Y})^{-1}$ where $\mathbf{Y}$ is the $(n \times 2)$ matrix containing the longitude and latitude for each sample.
- Finally, we compute the predicted coordinates for each samples as $\widehat{\mathbf{Y}} = \mathbf{Z}\mathbf{W}$ and plot the results on a world map (see Figure 1).

While this is admittedly a somewhat contrived example, it serves to illustrate the point that if we randomly project a small number of features that are highly correlated or predictive of the target into a large enough space, we should expect to be able to recover meaningful representations of the target from the latent space. Similarly, Alain and Bengio (2018) observe that

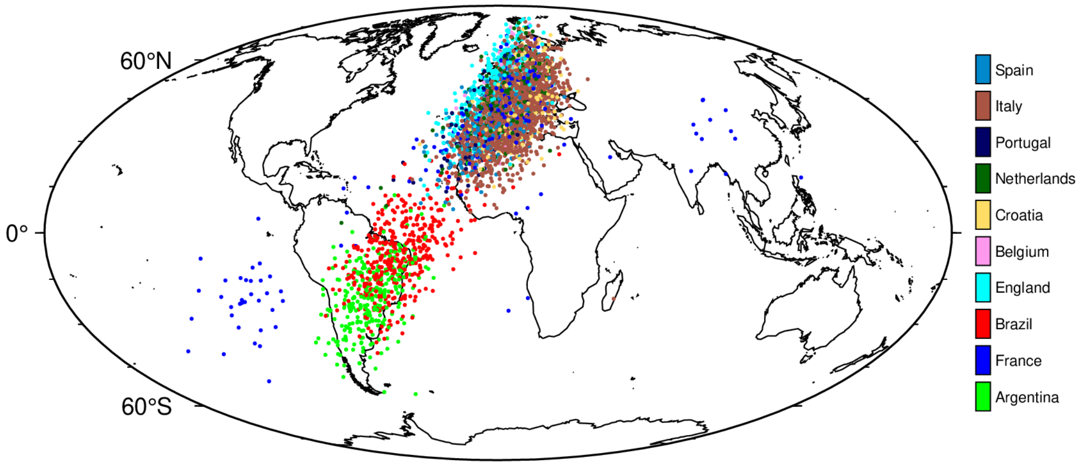

Figure 1: Probe predictions for a random projection.

even before training a convolutional neural network on MNIST data, the layer-wise activations can already be used to perform binary classification. In fact, it is well-known that random projections can be used for prediction tasks (Dasgupta 2013).

Example: PCA

Example: Autoencoder

Example: LLM

References

Alain, Guillaume, and Yoshua Bengio. 2018. "Understanding Intermediate Layers Using Linear Classifier Probes." https://arxiv.org/abs/1610.01644.

Dasgupta, Sanjoy. 2013. "Experiments with Random Projection." https://arxiv.org/abs/1301.3849.

Gurnee, Wes, and Max Tegmark. 2023. "Language Models Represent Space and Time." https://arxiv.org/abs/2310.02207.

Touvron, Hugo, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, et al. 2023. "LLaMA: Open and Efficient Foundation Language Models." https://arxiv.org/abs/2302.13971.