Heuristic explanations for 2nd argmax models

Jacob Hilton

December 21, 2024

Notation

We consider a 1-layer RNN $M_{d,n} : \mathbb{R}^n \to \Delta\{0, \ldots, n-1\}$ with d hidden neurons that has been trained to predict the 2nd argmax of a sequence of length n. This is specifed using the weight matrices

$$\begin{aligned} W^{hi} &\in \mathbb{R}^{d \times 1} \\ W^{hh} &\in \mathbb{R}^{d \times d} \\ W^{oh} &\in \mathbb{R}^{n \times d} \end{aligned}$$

and its value on the input $(x_0, \ldots, x_{n-1})$ is defined by the equations

$$\begin{aligned} h_0 &= \mathbf{0} \in \mathbb{R}^d, \\ z_{t+1} &= W^{hh} h_t + W^{hi} x_t, && t = 0, \ldots, n-1, \\ h_{t+1} &= \mathrm{ReLU}(z_{t+1}), && t = 0, \ldots, n-1, \\ l &= W^{oh} h_n, \\ M_{d,n}(x_0, \ldots, x_{n-1}) &= \mathrm{softmax}(l). \end{aligned}$$

We may also write these variables as a function of the inputs, for example $h_t(x_0, \ldots, x_{t-1})$ or $l(x_0, \ldots, x_{n-1})$.

The case $d = 4$, $n = 3$

Original version

$M_{4,3}$ happens to have the property that $W_0^{hi}, W_1^{hi} > 0$ and $W_0^{hi}, W_1^{hi} < 0$. It follows that

$$h_1(x_0) = \begin{cases} \left(W_0^{hi} x_0, W_1^{hi} x_0, \quad 0, \quad 0\right), & \text{if } x_0 \geq 0, \\ \left(\quad 0, \quad 0, W_2^{hi} x_0, W_3^{hi} x_0\right), & \text{if } x_0 \leq 0, \end{cases}$$

and hence

$$z_2(x_0, x_1) = \begin{cases} (a_j x_0 + c_j x_1)_{j=0,1,2,3}, & \text{if } x_0 \geq 0, \\ (b_j x_0 + c_j x_1)_{j=0,1,2,3}, & \text{if } x_0 \leq 0, \end{cases}$$

where

$$
\begin{aligned}
a_j &= W_{j0}^{hh} W_0^{hi} + W_{j1}^{hh} W_1^{hi} \\
b_j &= W_{j2}^{hh} W_2^{hi} + W_{j3}^{hh} W_3^{hi} \\
c_j &= W_j^{hi}
\end{aligned}
$$

As it turns out, we have

$$
\begin{aligned}
a_3 &\approx -a_0 > 0, \\
a_2 &\approx -a_1 > 0, \\
b_3 &\approx -b_0 > 0, \\
b_2 &\approx -b_1 > 0, \\
c_0 &\approx -c_3 > 0, \\
c_1 &\approx -c_2 > 0,
\end{aligned}
$$

and so for the purposes of drawing decision boundaries, we approximate a_3 and $-a_0$ by $\alpha_3 := \frac{a_3 + (-a_0)}{2}$, and similarly for α_2, β_3, β_2, γ_0 and γ_1. Since it happens that $\frac{\gamma_1}{\alpha_2} > \frac{\gamma_0}{\alpha_3}$ and $\frac{\gamma_1}{\beta_2} > \frac{\gamma_0}{\beta_3}$, under this approximation we have

$$
h_2(x_0, x_1) \approx \begin{cases}
(a_0 x_0 + c_0 x_1, a_1 x_0 + c_1 x_1, \quad 0, \quad 0), & \text{if } 0 \le x_0 \le \frac{\gamma_0}{\alpha_3} x_1, \\
(b_0 x_0 + c_0 x_1, b_1 x_0 + c_1 x_1, \quad 0, \quad 0), & \text{if } x_0 \le \min\left(0, \frac{\gamma_1}{\beta_2} x_1\right), \\
(\quad 0, \quad 0, a_2 x_0 + c_2 x_1, a_3 x_0 + c_3 x_1), & \text{if } x_0 \ge \max\left(0, \frac{\gamma_1}{\alpha_2} x_1\right), \\
(\quad 0, \quad 0, b_2 x_0 + c_2 x_1, b_3 x_0 + c_3 x_1), & \text{if } \frac{\gamma_0}{\beta_3} x_1 \le x_0 \le 0, \\
(\quad 0, a_1 x_0 + c_1 x_1, \quad 0, a_3 x_0 + c_3 x_1), & \text{if } \frac{\gamma_0}{\alpha_3} x_1 \le x_0 \le \frac{\gamma_1}{\alpha_2} x_1, \\
(b_0 x_0 + c_0 x_1, \quad 0, b_2 x_0 + c_2 x_1, \quad 0), & \text{if } \frac{\gamma_1}{\beta_2} x_1 \le x_0 \le \frac{\gamma_0}{\beta_3} x_1.
\end{cases}
$$

It turns out that moreover $a_j \approx b_j$ for $j = 0, 1, 2, 3$. Since the second and third regions are larger than the first and fourth, we approximate a_0 by b_0 and a_1 by b_1 in the first region and b_2 by a_2 and b_3 by a_3 in the second region to obtain

$$
h_2(x_0, x_1) \approx \begin{cases}
(b_0 x_0 + c_0 x_1, b_1 x_0 + c_1 x_1, \quad 0, \quad 0), & \text{if } x_0 \le \min\left(\frac{\gamma_0}{\alpha_3} x_1, \frac{\gamma_1}{\beta_2} x_1\right), \\
(\quad 0, \quad 0, a_2 x_0 + c_2 x_1, a_3 x_0 + c_3 x_1), & \text{if } x_0 \ge \max\left(\frac{\gamma_0}{\beta_3} x_1, \frac{\gamma_1}{\alpha_2} x_1\right), \\
(\quad 0, a_1 x_0 + c_1 x_1, \quad 0, a_3 x_0 + c_3 x_1), & \text{if } \frac{\gamma_0}{\alpha_3} x_1 \le x_0 \le \frac{\gamma_1}{\alpha_2} x_1, \\
(b_0 x_0 + c_0 x_1, \quad 0, b_2 x_0 + c_2 x_1, \quad 0), & \text{if } \frac{\gamma_1}{\beta_2} x_1 \le x_0 \le \frac{\gamma_0}{\beta_3} x_1,
\end{cases}
$$

and hence we obtain $z_3(x_0, x_1, x_2)$ as linear function of (x_0, x_1, x_2) on each of these 4 regions.

Now we consider the 6 possible orderings of x_0, x_1 and x_2:

- For each of the first and second regions, there are 3 possible orderings, since it happens that $\frac{\gamma_0}{\alpha_3}, \frac{\gamma_0}{\beta_3} < 1$ and $\frac{\gamma_1}{\alpha_2}, \frac{\gamma_1}{\beta_2} > 1$, which implies that $x_0 \le x_1$ in the first region and $x_0 \ge x_1$ in the second region.
- For each of the third and fourth regions, all 6 orderings are possible, but only 4 are likely, since x_0 and x_1 are fairly close in these regions, because $\frac{\gamma_0}{\alpha_3}$, $\frac{\gamma_1}{\alpha_2}$, $\frac{\gamma_0}{\beta_3}$ and $\frac{\gamma_1}{\beta_2}$ all happen to be fairly close.

This gives 14 likely subregions plus 4 unlikely subregions to consider.

To finish, we apply a presumption of independence in each of the likely subregions to

$$h_3\left(x_0, x_1, x_2\right)_j = \mathrm{ReLU}\left(z_3\left(x_0, x_1, x_2\right)_j\right) = z_3\left(x_0, x_1, x_2\right)_j \mathbb{1}_{z_3(x_0,x_1,x_2)_j>0}$$

for $j = 0, 1, 2, 3$. We approximate each $\mathbb{1}_{z_3(x_0,x_1,x_2)_j>0}$ in each likely subregion as a random variable that is independent of $z_3\left(x_0, x_1, x_2\right)$ in the subregion, but dependent on other such random variables through some latent random variable $Z \sim \mathrm{Bernoulli}\left(q\right)$, where $q = 0.2$ is a hyperparameter. Writing p for the probability that $z_3\left(x_0, x_1, x_2\right)_j > 0$ in the subregion and $\sigma = \sqrt{p\left(1-p\right)}$, if $p \leq \frac{1}{2}$ then we approximate $\mathbb{1}_{z_3(x_0,x_1,x_2)_j>0}$ by a random variable that is $p - \sqrt{\frac{q}{1-q}}\sigma$ when $Z = 0$ and $p + \sqrt{\frac{1-q}{q}}\sigma$ when $Z = 1$, and if $p > \frac{1}{2}$ then we approximate $1 - \mathbb{1}_{z_3(x_0,x_1,x_2)_j>0}$ the same way, to match the mean and variance. This works well because p happens to usually to be close to either 0 or 1. We then apply W^{oh} to obtain the 3 logits in that subregion as a linear function of x_0, x_1 and x_2, and the model's accuracy in that subregion is then then probability that the appropriate logit is the largest. For each of the unlikely subregions, we simply approximate the model's accuracy in that subregion as $\frac{1}{3}$. The accuracies for the likely and unlikely subregions can then be combined to obtain an estimate for the overall accuracy of the model.

These calculations can be performed by using the polygonal generalization of Girard's theorem[1] to calculate the proportion of $\mathbb{R}^3$ taken up by the intersection of a set of half-spaces whose boundaries pass through the origin. To obtain our overall estimate, we need:

- the volume of all 18 subregions, each of which is the intersection of 4 half-spaces;
- the volume of the sub-subregion where $z_3\left(x_0, x_1, x_2\right)_j > 0$ for each of the 14 likely subregions and each $j = 0, 1, 2, 3$, for a total of 56 sub-subregions, each of which is the intersection of 5 half-spaces;
- the volume of the sub-subregion where the appropriate logit is largest for each of the 14 likely subregions and each of the cases $Z = 0$ and $Z = 1$, for a total of 28 sub-subregions, each of which is the intersection of 6 half-spaces.

Version based on George's approach

First read George's writeup *Symmetric RNNs*. Here I will discuss how to use George's approach to produce an estimate for the model's accuracy by tracking noise terms.

George observes that

$$\begin{aligned} W_{hi} &\approx U^{\mathsf{T}} W'_{hi} \\ W_{hh} &\approx U^{\mathsf{T}} W'_{hh} U \\ W_{oh} &\approx W'_{oh} V \end{aligned}$$

for some $W'_{hi} \in \mathbb{R}^{2\times 1}$, $W'_{hh} \in \mathbb{R}^{2\times 2}$ and $W'_{oh} \in \mathbb{R}^{3\times 2}$, where $U = \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{pmatrix}$ and $V = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$ (we have changed from superscripts to subscripts so we can match

[1] `https://en.wikipedia.org/wiki/Spherical_trigonometry#Area_and_spherical_excess`

George's "prime" notation). So let us define

$$\begin{aligned}\varepsilon_{hi} &= W_{hi} - U^\mathsf{T} W'_{hi} \\ \varepsilon_{hh} &= W_{hh} - U^\mathsf{T} W'_{hh} U \\ \varepsilon_{oh} &= W_{oh} - W'_{oh} V,\end{aligned}$$

where W'_{hi}, W'_{hh} and W'_{oh} are obtained by averaging the corresponding entries of W_{hi}, W_{hh} and W_{oh} so as to minimize the sum of squared entries of ε_{hi}, ε_{hh} and ε_{oh}. Let us also define h'_t (for $t = 0, 1, 2, 3$), z'_t (for $t = 1, 2, 3$) and l' to be the variables that result from using the above approximations, and ε_{h_t}, ε_{z_t} and ε_l to be the corresponding differences, which are all random variables since they are functions of x_0, x_1 and x_2.

Now we use a version of covariance propagation to estimate the mean and covariance matrix of ε_l. To spell this out, we start with $h'_0 = \varepsilon_{h_0} = \mathbf{0}$, and then, given h'_t and ε_{h_t} for some $t = 0, 1, 2$, we have

$$\begin{aligned}z'_{t+1} &= U^\mathsf{T} W'_{hh} U h'_t + U^\mathsf{T} W'_{hi} x_t \\ h'_{t+1} &= \mathrm{ReLU}\left(z'_{t+1}\right)\end{aligned}$$

and

$$\begin{aligned}\varepsilon_{z_{t+1}} &= W_{hh} h_t - U^\mathsf{T} W'_{hh} U h'_t + W_{hi} x_t - U^\mathsf{T} W'_{hi} x_t = W_{hh} \varepsilon_{h_t} + \varepsilon_{hh} h'_t + \varepsilon_{hi} x_t \\ \varepsilon_{h_{t+1}} &= \mathrm{ReLU}\left(z'_{t+1} + \varepsilon_{z_{t+1}}\right) - \mathrm{ReLU}\left(z'_{t+1}\right).\end{aligned}$$

The mean and covariance matrix of z'_{t+1} can be computed exactly from the mean and covariance matrix of h'_t. To estimate the mean and covariance matrix of h'_{t+1}, we simply multiply each entry of z'_{t+1} by the probability that it is positive, which is a poor approximation, but one that will end up being multiplied by ε_{hh} or ε_{oh}, both of which are small. To estimate the mean and covariance matrix of $\varepsilon_{z_{t+1}}$, we apply a presumption of independence to ε_{h_t} and h'_t, and the calculation can then be done exactly. To estimate the mean and covariance matrix of $\varepsilon_{h_{t+1}}$, we multiply each entry of $\varepsilon_{z_{t+1}}$ by the probability that corresponding entry of z'_{t+1} is positive, which is a better approximation since $\varepsilon_{z_{t+1}}$ is small. Finally, we have

$$\varepsilon_l = W_{oh} h_3 - W'_{oh} V h'_3 = W_{oh} \varepsilon_{h_3} + \varepsilon_{oh} h'_3,$$

and so we can compute the mean and covariance of ε_l by applying a presumption of independence to ε_{h_3} and h'_3.

Note that every random variable in this approximation is actually treated a zero-mean Gaussian, and so all of the probabilities of variables being positive are actually just $\frac{1}{2}$.

To finish, we have 4 regions corresponding to the possible combinations of signs of the 2 neurons in the reduced absolute-value network, and 6 subregions for each of those 4 regions corresponding to the possible orderings of x_0, x_1 and x_2. Even though only 20 of those 24 subregions are possible and only 10 of those are likely, it is not much more work to calculate the probability times the accuracy of a subregion than it is to calculate its probability, so we simply calculate the probability times the accuracy of for every subregion. Since the noise term ε_l is 3-dimensional, this requires us to calculate the volume of 24 subregions in 6 dimensions, each of which is the intersection of 4 half-spaces. We avoid the problem of figuring out how to do this exactly for now and just use sampling.

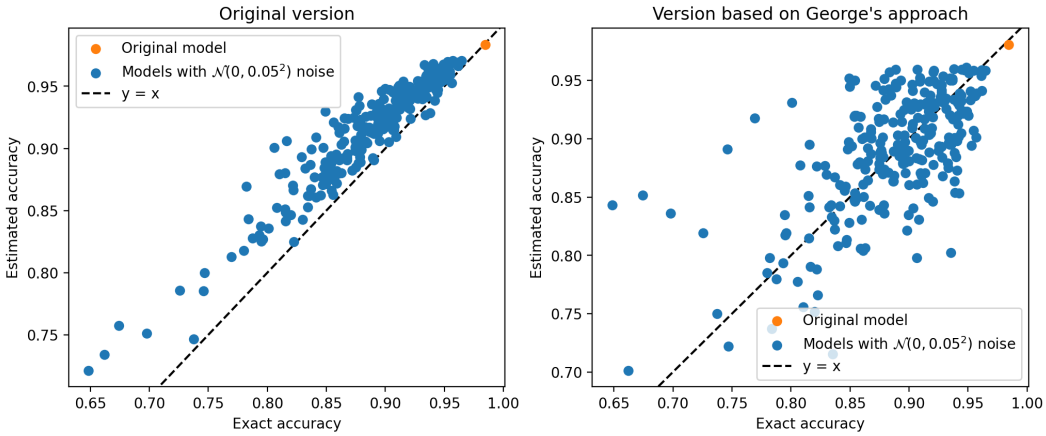

Figure 1: Comparison of the two accuracy estimators.

Quality versus compute

To test these methods of estimating the accuracy of $M_{4,3}$, we added zero-mean Gaussian noise with a standard deviation of 0.05 to the weights of network and ran the two estimators on 256 such models. The original version achieves a RMS error of around 3%, and the version based on George's approach achieves a RMS error of around 5%, although it appears to have lower bias. A plot of the individual estimates is given in Figure 1.

The computationally expensive part of both estimators is calculating the volumes of intersections of half-spaces. In 3 dimensions, let us crudely estimate that it takes around $5k^3$ operations to do this for k half-spaces, since we exhaustively find the intersection of each pair of hyperplanes and check whether each ray lies in all the other half-spaces. In 6 dimensions, let us just use the same estimate for simplicity. Then the original version takes around $5\left(18 \cdot 4^3 + 56 \cdot 5^3 + 28 \cdot 6^3\right) = 71000$ operations and the version based on George's approach takes around $5 \cdot 24 \cdot 4^3 = 7680$ operations.

For comparison, estimating the accuracy using k samples takes around $(3 \cdot 4 \cdot (4+1) + 4 \cdot 3)\, k = 72k$ operations and achieves a RMS error of $\sqrt{\frac{p(1-p)}{k}}$ for a model with accuracy p. The RMS of this expression for our 256 noisy models is around $\frac{24\%}{\sqrt{k}}$, and so with the same computational budget as our two methods, sampling should achive an RMS error of around 0.8% and 2.3% respectively. So we are underperforming sampling by around a factor of 2 in terms of error (i.e, a factor of 4 in terms of compute). But the half-space algorithm could almost certainly be done more efficiently.

Surprise accounting

For the original version, we incur roughly 16 bits of surprise by checking the inequalities $a_3, -a_0, a_2, -a_1, b_3 - b_0, b_2, -b_1, c_0, -c_3, c_1, -c_2 > 0$, $\frac{\gamma_0}{\alpha_3}, \frac{\gamma_0}{\beta_3} < 1$ and $\frac{\gamma_1}{\alpha_2}, \frac{\gamma_1}{\beta_2} > 1$, which imply the other inequalities claimed. We also incur surprise for each of the $18 + 56 + 28 = 102$ volume calculations, for which we should pay something like the log probability of each calculated volume, although it is unclear what prior we should use. We could perhaps use an intersection of random half-spaces for this prior, although this fails to capture how the

half-spaces were chosen. We should also perhaps incur some surprise for the details of the derivation in this write-up. Finally, we incur surprise about the true accuracy of the model given our estimate. This all seems confusing enough that we should perhaps just stick to quality versus compute for evaluating explanations, which surprise accounting seems to be approximating.