Peirce in the Machine Supplementary Material

Appendix 1: Proof of Proposition 4.3

Suppose $\mathscr{H}$ is a top-expert mixture of experts model of n homogenous binary classifier experts drawn from the set $\mathscr{E}$ that is closed under translation with $VCD(\mathscr{E}) = m$ defined over instance space $\mathbb{R}$. We need to show that there exists a set of points $X = \{x^{(i)} \in \mathbb{R} : x^{(1)}, \ldots, x^{(nm)}\}$ such that $\mathscr{H}$ shatters X. Recall that each $h \in \mathscr{H}$ is a piece-wise function defined as follows where $Z_1, Z_2, \ldots, Z_n$ is a partition of $\mathbb{R}$:

$$h(x; n) = \begin{cases} e_1(x) & x \in Z_1 \\ e_2(x) & x \in Z_2 \\ \vdots & \vdots \\ e_n(x) & x \in Z_n \end{cases}$$

where each $e_i \in \mathscr{E}$ is a binary classifier, $e_i : \mathbb{R} \to \{1, 0\}$ for $i = 1, 2, \ldots, n$. Since $VCD(\mathscr{E}) = m$, there exists a set of points $X_m^1 = \{x^{(i)} \in \mathbb{R} : x^{(1)}, \ldots, x^{(m)}\}$ of cardinality m that $\mathscr{E}$ shatters, i.e. for any labels $Y_m^1 = \{y^{(1)}, \ldots, y^{(m)}\}$, there exists $e \in \mathscr{E}$ such that $e(x^{(i)}) = y^{(i)}$ for $i = 1, \ldots, m$.

We can construct pairwise disjoint sets $X_m^1, \ldots, X_m^n$ such that for any labels Y there exists some h that correctly assigns the labels. First, we build these pairwise disjoint sets. The idea is that we use our initial set X_m^1 and shift it around. We construct this set as follows:

$$X_m^i = \{x \in \mathbb{R} : \exists x^{(j)} \in X_m^{i-1} x = x^{(j)} + c_i\}$$

where each c_i is chosen such that $x^{(j)} + c_i \notin \bigcup_{k=1}^{i-1} X_m^k$ for all $j = 1, \ldots, m$ and $c_1 = 0$. We pick the constant that changes every member of the previous set to a new number.

We claim these sets are pairwise disjoint. Let X_m^i and X_m^k such that $i \neq k$. Suppose $i > k$. By definition, we have chosen c_i such that each member $x \in X_m^i$ is not equal to $x^{(j)} + c_k$ for all $j = 1, \ldots, m$. Similarly, suppose $k > i$, we have chosen c_k such that each member $x \in X_m^k$ is not equal to $x^{(j)} + c_i$ for all $j = 1, \ldots, m$.

Next we define classifiers $e_j^{Y^k} \in \mathscr{E}$ for $j = 1, \ldots, n$ with respect to the set of labels $Y^k = \{y_k^{(1)}, \ldots, y_k^{(m)}\}$ as simply removing all the translations c_l for $l = 1, \ldots, j$ and applying the classifier e^{Y^k} that correctly labels our points in the set $x^{(i)} \in X_m^1$, $e^{Y^k}(x^{(i)}) = y_k^{(i)}$ for $i = 1, \ldots, m$. We define for our new set of points $x^{*(i)} \in X_m^j$ the classifier $e_j^{Y^k}$:

$$e_j^{Y^k}(x^{*(i)}) = e^{Y^k}(x^{*(i)} - \sum_{l=1}^{j} c_l)$$

Each classifier $e_j^{Y^k}$ labels using e^{Y^k} by removing the translations that built the members of X_m^j. Note that $e_1^{Y^k} = e^{Y^k}$ since $c_1 = 0$. We know by assumption that e^{Y^k} must exist for Y^k for the set X_m^1. For any arbitrary labels Y^k on set X_m^j, the classifier $e_j^{Y^k}$ will correctly label the members of X_m^j because those labels applied to the original set $x^{(i)} \in X_m^1$ will

be correctly classified by e^{Y^k} and note that $x^{(i)} = x^{*(i)} - \sum_{l=1}^{j} c_l$. Since this new classifier is a translation of the older classier e^{Y^k}, it is in set $\mathscr{E}$.

We claim that for the set $\bigcup_{j=1}^{n} X_m^j$, for any labels $Y = \{y^{(1)}, \ldots, y^{(nm)}\}$, there exists $h(x^{(i)}; n) = y^{(i)}$ for $i = 1, \ldots, nm$. Partition Y into subsets Y_m^j for $j = 1, \ldots, n$ corresponding to each X_m^j where each label in Y_m^j applies to an element in X_m^j for $j = 1, \ldots, n$. We define our mixture of experts hypothesis h_Y as follows:

$$h_Y(x; n) = \begin{cases} e_1^{Y^1}(x) & x \in X_m^1 \\ e_2^{Y^2}(x) & x \in X_m^2 \\ \vdots & \vdots \\ e_{n-1}^{Y^{n-1}}(x) & x \in X_m^{n-1} \\ e_n^{Y^n}(x) & x \in C \end{cases}$$

where $C = X_m^n \cup \mathbb{R} - \bigcup_{j=1}^{n-1} X_m^j$ is the catch-all for everything outside of our disjoint set. Since each $e_j^{Y^j}$ can correctly label each $y^{(i)} \in Y_m^j$, we can label the union of Y_m^j corresponding to the union of X_m^j for $j = 1, \ldots, n$. Thus h_Y will correctly label Y and so there exists a $h \in \mathscr{H}$ that shatters $\bigcup_{j=1}^{n} X_m^j$, which by the fundamental theorem of counting has a cardinality of nm.

Appendix 2: Experiment Details

All code can be found at https://github.com/brushing-git/PeirceMachine.

Our linear regression experiments were conducted on datasets generated by univariate polynomials of degrees $[1, 2, 3, 4, 5]$. Feature data was drawn from a uniform distribution $[-2.0, 1.0]$ and the targets via polynomials with coefficients drawn from $[2, 3, -1, -1, 1, 1]$, e.g. the degree one polynomial was $p(x) = 2x + 3$ and so on. The data was rescaled using the Standard Scaling method. We added noise from a normal distribution $\mathscr{N}(0, 0.1^2)$ to the targets. Finally, an identity feature was added. The training and test data were 10000 and 2000 respectively for each degree.

The models trained in the linear regression experiments include a Bayesian linear regression and MOE of linear regressions. The former assumed a conjugate prior, with a posterior predictive distribution:

$$\begin{aligned} p(y|\mathbf{x}, \mathscr{D}) &= \int \mathscr{N}(y|\mathbf{x}^\intercal \theta, \sigma^2)\mathscr{N}(\theta|\mu, \Sigma)d\theta \\ &= \mathscr{N}(y|\mathbf{x}^\intercal \mu, \mathbf{x}^\intercal \Sigma \mathbf{x} + \sigma^2) \end{aligned}$$

where μ and Σ are the posterior mean and covariance and are calculated as usual (see Murphy (2022) section 11.7), and σ is the known likelihood standard deviation. Our prior was the standard normal $\mathscr{N}(0, \mathbb{I})$. For the MOEs, we trained sparse, top-2 expert MOEs using stochastic gradient descent and the Adam optimizer. In training, the expert

linear regressions learned to fit their means to the training data, and we used the likelihoods at inference time. Our learning rate was set via an exponential decay schedule, where the learning rate for each epoch $\eta(x) = \eta_0 \exp(-\gamma x)$, with a $\gamma = 0.75$ and an initial learning rate $\eta_0 = 0.2$. MOEs were trained for 30 epochs.

The MOE models in all experiments were sparse, top-k MOEs (Shazeer et al., 2017). Sparse, top-k MOEs apply noise to the gating function and only use the top-k experts. Here the gating function, G, is:

$$G(\mathbf{x}) = Softmax(KeepTopK(H(\mathbf{x}), k))$$

where $KeepTopK$ and H are:

$$KeepTopK(v, k)_i = \begin{cases} v_i & \text{if } v_i \text{ is in the top } k \text{ elements of } v \\ -\infty & \text{otherwise} \end{cases}$$

$$H(\mathbf{x})_i = (W_g \mathbf{x})_i + \varepsilon \cdot \log(1 + \exp((W_{noise}\mathbf{x})_i))$$

where $\varepsilon \sim \mathcal{N}(0, \mathbb{I})$, W_g is a matrix of gate weights, and W_{noise} is a matrix of noise weights. The MOEs were trained with the Adam optimizer and a fixed learning rate of 0.001.

In the VC dimension experiments, we generated the data as described in section 5. Our feature data was constructed by sampling x_1 from a uniform distribution $[-3.0, 3.0]$ and computing x_2 by polynomials $p(x; m)$ of degree $m \in [1, 2, 3, 4, 5, 6, 7, 8]$ with coefficients sampled from $\mathcal{N}(0, 10)$ and adding noise to x_2 from $\mathcal{N}(0, \sigma)$, where σ is the standard deviation of x_2. We rescaled the features identically to the linear experiments and added an identity feature. The targets were computed from the non-identity features by thresholding the feature x_2 based on the polynomial $p(x; m)$. The total training data and test data size was 10000 and 2000 for each degree of polynomial.

For the VC dimension experiments, we trained two Bayesian logistic regressions via stochastic gradient Hamiltonian Monte Carlo (SGHMC) (Chen et al., 2014) and variational inference (VI) (Blei et al., 2017). SGHMC is a method that approximates the posterior by taking noisy samples from a mini-batch of $n' \ll n$ to approximate the posterior energy function:

$$\widetilde{U}(\theta) = \frac{n'}{|n|} \sum_{i=1}^{n'} \log p(x_i|\theta) - \log p(\theta)$$

We update our parameters θ_k at each epoch k by adding v_{k-1} $\theta_k = \theta_{k-1} + v_{k-1}$, where v_k is updated at each epoch through the gradient of the approximate posterior energy function $v_k = v_{k-1} - \alpha_k \nabla \widetilde{U}(\theta_k) - \eta v_{k-1} + \sqrt{2(\eta - \hat{\gamma})\alpha_k}\varepsilon_k$ where α_k is the learning rate, η is a friction parameter, $\hat{\gamma}$ is an estimate of the data noise, and $\varepsilon \sim \mathcal{N}(0, \mathbb{I})$ is noise sampled from a normal distribution. In our experiments, $\eta = 0.9$, the learning rate α_k had an exponential decay schedule, $\hat{\gamma}$ was set to 10000^{-1}, we trained the models for 84 burn-in epochs, and collected 16 samples for inference. VI is an approximation scheme utilizing a variational distribution q with parameters ϕ. We train our model by minimizing

the Kullback-Leibler-divergence between our true posterior and the variational distribution, which can be done by minimizing $\mathscr{L}(x, \phi) = \mathbb{E}_{z\sim q(z;\phi)}[\log p(x, z) - T \log q(z; \phi)]$, where T is the temperature parameter. We used the Bayes By Backprop (Blundell et al., 2015) method, where we use the reparameterization trick to compute the gradients for the loss via the backpropagation algorithm. Our prior was the standard normal prior $\mathscr{N}(0, \mathbb{I})$ and a temperature of 0.1. Models were trained for 100 epochs at a learning rate of 0.01, and during inference, we drew 16 samples.

References

Blei, D. M., Kucukelbir, A. and McAuliffe, J. D. (2017) Variational inference: A review for statisticians. *Journal of the American statistical Association*. *112*(518), 859–877. 10.1080/01621459.2017.1285773.

Blundell, C., Cornebise, J., Kavukcuoglu, K. and Wierstra, D. (2015) Weight uncertainty in neural network. International conference on machine learning. PMLR. pp. 1613–1622.

Chen, T., Fox, E. and Guestrin, C. (2014) Stochastic gradient hamiltonian monte carlo. International conference on machine learning. PMLR. pp. 1683–1691.

Murphy, K. P. (2022) *Probabilistic machine learning: an introduction*. MIT press.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G. and Dean, J. (2017) Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.