1 The implicit regularization penalty of PT+FT in ReLU networks

1.1 Derivation of the regularization penalty

We consider a shallow neural network with the ReLU nonlinearity

$$f_w(x) := \sum_{h=1}^{H} w_h^{(2)} (\langle \boldsymbol{w}_h^{(1)}, x \rangle)_+, \quad w_h^{(2)} \in \mathbb{R}, \boldsymbol{w}_h^{(1)} \in \mathbb{R}^d. \tag{1}$$

We assume that its first-layer hidden weights after pretraining are given by $\boldsymbol{w}_{0h}^{(1)} \in \mathbb{R}^d$, whereas its readout weights are initialized by a small value $\gamma \in \mathbb{R}$. We then define norm of the deviation from the parameters at initialization as

$$R(w|w_0^{(1)}, \gamma) := \sum_{h=1}^{H} (w_h^{(2)} - \gamma)^2 + \|w_h^{(1)} - w_{0h}^{(1)}\|_2^2, \tag{2}$$

where $w = (\boldsymbol{w}_h^{(1)}, w_h^{(2)})_h$ represents the parameters of the network and $w_0 = (\boldsymbol{w}_{0h}^{(1)})_h$ represents the first-layer parameters of the network at (post-pretraining) initialization. Note that w is parameterized in a redundant manner, as we can multiply $w_h^{(2)}$ by a constant $a > 0$ and obtain the same function as long as we divide $\boldsymbol{w}_h^{(1)}$ by the same constant. To parameterize a function in a unique fashion, we describe it in terms of the normalized hidden weight $\hat{\boldsymbol{w}}_h := \boldsymbol{w}_h^{(1)}/\|\boldsymbol{w}_h^{(1)}\|_2$ and the (signed) magnitude $m_h := \|\boldsymbol{w}_h^{(1)}\|_2 w_h^{(2)}$. We wish to compute the norm of the deviation from the initialization in terms of this parameterization. Specifically, we compute

$$\tilde{R}(\hat{w}, m|w_0^{(1)}, \gamma) := \min_w R(w|w_0, \gamma) \text{ s.t. } \forall_h \hat{\boldsymbol{w}}_h = \boldsymbol{w}_h^{(1)}/\|\boldsymbol{w}_h^{(1)}\|_2, m_h = \|\boldsymbol{w}_h^{(1)}\|_2 w_h^{(2)}. \tag{3}$$

Note that Woodworth *et al.* [1] solve a version of this optimization problem for diagonal linear networks (though only for the case in which corresponding first and second layer weights are initialized with equal magnitude).

As this optimization problem decomposes over hidden units, we can solve it for each hidden unit individually:

$$\tilde{R}(\hat{w}, m|w_0^{(1)}, \gamma) = \sum_{h=1}^{H} \tilde{r}(\hat{\boldsymbol{w}}_h, m_h|\boldsymbol{w}_{0h}, \gamma), \tag{4}$$

$$\tilde{r}(\hat{\boldsymbol{w}}_h, m_h|\boldsymbol{w}_{0h}^{(1)}, \gamma) := \min_{\boldsymbol{w}_h^{(1)}, w_h^{(2)}} (w_h^{(2)} - \gamma)^2 + \|\boldsymbol{w}_h^{(1)} - \boldsymbol{w}_{0h}^{(1)}\|_2^2 \text{ s.t. } \hat{\boldsymbol{w}}_h = \boldsymbol{w}_h^{(1)}/\|\boldsymbol{w}_h^{(1)}\|_2, m_h = \|\boldsymbol{w}_h^{(1)}\|_2 w_h^{(2)}. \tag{5}$$

We can express this as

$$\tilde{r}(\hat{w}_h, m_h|w_{0h}^{(1)}, \gamma) = \min_{u>0} p(u), \quad p(u) := (m_h/u - \gamma)^2 + \left\| u\hat{\boldsymbol{w}}_h - \boldsymbol{w}_{0h}^{(1)} \right\|_2^2. \tag{6}$$

Note that this function is smooth and diverges to ∞ both for $u \to \infty$ and $u \to 0$. As long as we have a unique positive stationary point u^*, this stationary point must therefore be a minimum. We simplify

$$p(u) = (m_h/u - \gamma)^2 + u^2 + \|\boldsymbol{w}_{0h}^{(1)}\|_2^2 - 2u\langle \hat{\boldsymbol{w}}_h, \boldsymbol{w}_{0h}^{(1)} \rangle, \tag{7}$$

and, defining $m_{0h}^{(1)} := \|\boldsymbol{w}_{0h}^{(1)}\|_2$, and the cosine similarity between corresponding pretraining and finetuning task weights $\rho_h := \frac{\langle \boldsymbol{w}_h^{(1)}, \boldsymbol{w}_{0h}^{(1)} \rangle}{\|\boldsymbol{w}_h^{(1)}\|_2 \|\boldsymbol{w}_{0h}^{(1)}\|_2}$, write this as

$$p(u) = (m_h/u - \gamma)^2 + u^2 + (m_{0h}^{(1)})^2 - 2um_{0h}^{(1)}\rho_h. \tag{8}$$

We then determine the stationary points by computing the derivative

$$p'(u) = -2(m_h/u - \gamma)m_h/u^2 + 2u - 2m_{0h}^{(1)}\rho_h, \tag{9}$$

and setting it to zero. Simplifying this equation (and multiplying by $u^3/2$) yields

$$-m_h^2 + \gamma m_h u - m_{0h}^{(1)} \rho_h u^3 + u^4 = 0. \tag{10}$$

We can compute the solution to this quartic equation, e.g. using `np.roots` and select the (unique, in all empirical cases we explored) positive real root, u^*. We can then compute the original norm by plugging in u^*:

$$\tilde{r}(\hat{\boldsymbol{w}}_h, m_h | \boldsymbol{w}_{0h}^{(1)}, \gamma) = (m_h/u^* - \gamma)^2 + u^{*2} + (m_{0h}^{(1)})^2 - 2u^* m_{0h}^{(1)} \rho_h. \tag{11}$$

1.2 Analysis of the regularization penalty

We assume that the coefficient of feature h in the learned solution of the pretraining task is m^{aux} (without loss of generality we will assume this quantity is positive). Following pretraining, the first-layer weights will have the magnitude $m_{0h}^{(1)} = \sqrt{m^{\text{aux}}}$ (this is guaranteed if the pretrained solution also minimizes the parameter norm, as shown in Appendix B.2 in our manuscript, also see e.g. [2]). We plot the resulting penalty $\tilde{r}(\hat{\boldsymbol{w}}_h, m_h)$ for different correlations ρ between $\hat{\boldsymbol{w}}_h$ and $\boldsymbol{w}_{0h}$, and different initial magnitudes m^{aux} (Fig. 1a). Further, to determine whether the norm locally behaves more like an ℓ_1-norm (encouraging sparsity) or an ℓ_2-norm (not encouraging sparsity), we plot the derivative of the log norm,

$$\frac{\partial \log(\tilde{r}(\hat{\boldsymbol{w}}_h, m_h))}{\partial m}.$$

If this derivative is closer to 2, the norm behaves more like an ℓ_2-norm, if the derivative is closer to 1, the norm behaves more like an ℓ_1-norm (Fig. 1b).

If the finetuning task relies on features that are highly correlated with those of the pretraining task, the solution with minimum parameter norm deviation will recruit the units representing the corresponding pretraining feature to represent the highly correlated finetuning task feature, i.e. ρ will be close to 1. Notably, for ρ near 1, we find that the norm behaves qualitatively similar to the diagonal linear network norm (compare to Fig. 1a in the original submission). In particular even in the regime where it behaves more ℓ_1-like (i.e. the derivative is close to one, Fig. 1b, left), there remains a gap between $m^{\text{aux}} = 1$ (indicating a feature used in the pretraining task) and $m^{\text{aux}} = 0.01$ (indicating a feature not used in the pretraining task). This provides a mathematical justification for why we observe a nested sparsity effect for ReLU networks, which is essentially the same as the justification in the diagonal linear network case.

Second, we find that for $\rho < 1$ (even relatively large ρ, e.g. $\rho = 0.99$), the norm behaves similarly to an ℓ_1-norm in the $m \approx 1$ regime (i.e. if the finetuning task uses a feature with the same weight as the pretraining task does, which is the case in our teacher-student setup). This suggests an explanation for why in our experiments (unlike for the diagonal linear networks), the ReLU networks exhibit nested feature selection behavior without any weight rescaling.

Our new result also makes novel predictions. In particular, we observe that if the finetuning task feature is only moderately correlated with the pretraining task feature (e.g. $\rho = 0.9$), then if the weight of that pretraining task feature m^{aux} is much larger than the weight of the corresponding finetuning task feature (e.g. $m^{\text{aux}} = 100$, $m \approx 1$), it actually yields a worse penalty than a if the pretrainingn task feature has smaller weight ($m^{\text{aux}} = 0.01$). This is the opposite of the behavior in the $\rho = 1$ case! This theoretical result predicts that if the finetuning task employs features with smaller weights than the pretraining task, the network will not benefit from pretraining task features that are only moderately correlated with the finetuning task features (whereas if the weights are comparable in magnitude it will benefit). To test this, we examine the behavior of pretrained networks finetuned on tasks with varying the correlations between pretraining / finetuning task features and their magnitudes (Fig. 1c). For perfectly overlapping features (i.e. $\rho = 1$), the pretrained network is indeed much more sample efficient even when the finetuning task feature weight is small relative to the pretrainng task feature weight. However, as the correlation decreases this benefit vanishes and reverses rapidly (but for small finetuning task feature weights only). This is a novel and unintuitive phenomenon that we would not have hypothesized without the theoretical analysis above.

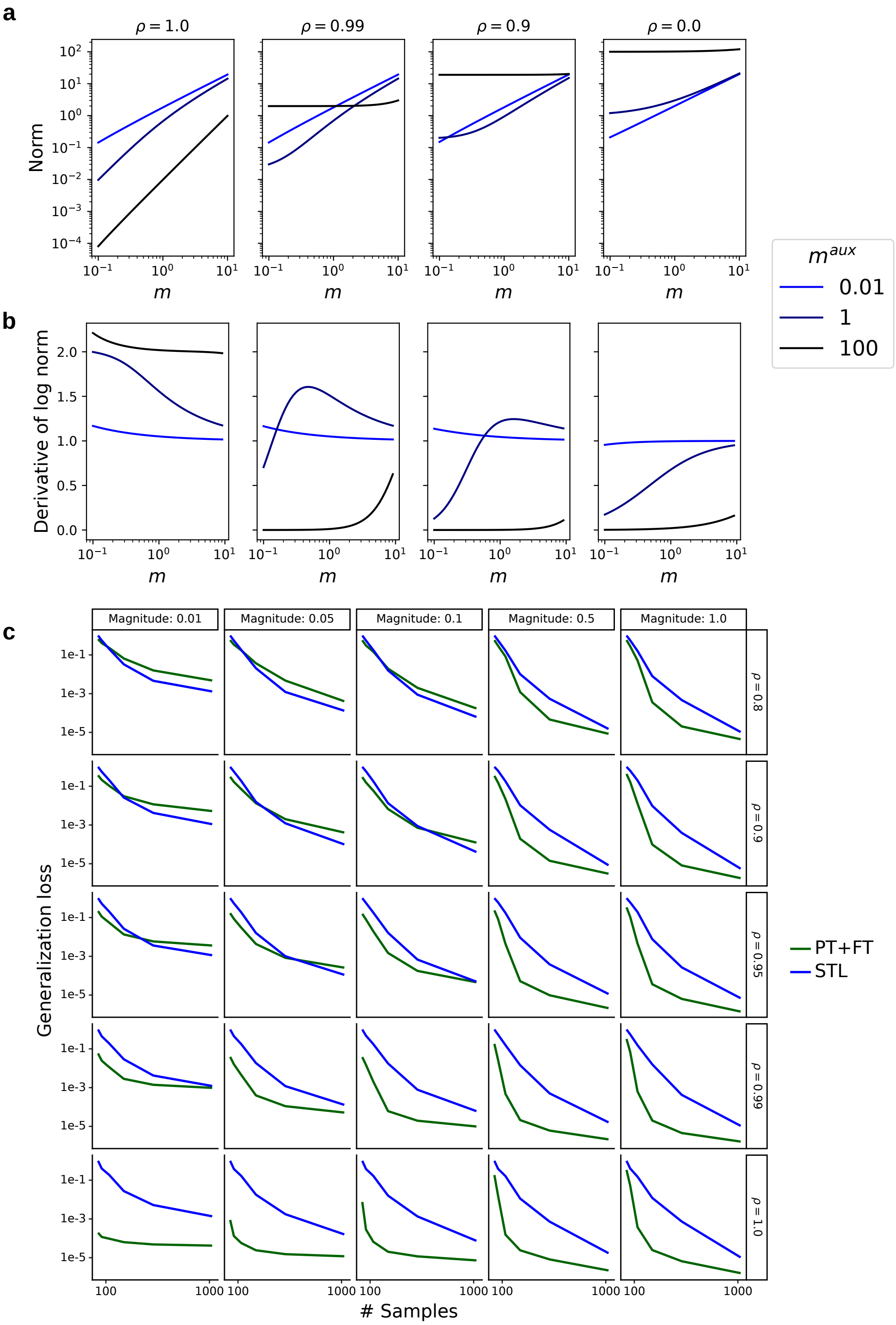

Figure 1: Regularization norm derived above for ReLU networks, and associated empirical predictions. **a**, Norm $\tilde{r}$ for different values of ρ and different initial magnitudes m^{aux}. $m^{\mathrm{aux}} = 0.01$ illustrates the case where the unit is essentially newly initialized, $m^{\mathrm{aux}} = 1$ illustrates the case where the feature represented by the neuron has comparable weight for the auxiliary and finetuning tasks, and $m^{\mathrm{aux}} = 100$ illustrates the case where the feature represented by the neuron is used with much larger weight in the auxiliary than the finetuning task. **b**, Derivative of $\log(\tilde{r})$, computed empirically. **c**, Sample efficiency for single-task learning (STL) and pretraining, then finetuning (PT+FT). We consider six teacher units and different possible correlations ρ between the pretraining and finetuning task (different rows) as well as different possible magnitudes of the finetuning task feature weights (different magnitudes). A lower magnitude of the finetuning task feature weight corresponds to a larger relative magnitude of the auxiliary task (i.e. equivalent to increasing m^{aux} while holding m constant).

References

1. Woodworth, B. *et al. Kernel and Rich Regimes in Overparametrized Models* en. in *Proceedings of Thirty Third Conference on Learning Theory* ISSN: 2640-3498 (PMLR, July 2020), 3635–3673. https://proceedings.mlr.press/v125/woodworth20a.html (2023).
2. Savarese, P., Evron, I., Soudry, D. & Srebro, N. How do infinite width bounded norm networks look in function space?: 32nd Conference on Learning Theory, COLT 2019. *Proceedings of Machine Learning Research* **99,** 2667–2690. http://www.scopus.com/inward/record.url?scp=85132757852&partnerID=8YFLogxK (2023) (2019).