Matrix products in neural nets

In neural networks, we often come acorss formulations that involve matrix products. The most straighforward example is a fully connect layer which connects the previous the the next $W_\ell x_\ell$. Another example, which is arguably leading to the longest chain of product appears in Recurrent Neural Networks (RNNs), where the recursive application of a weight $x_{t+1} = x_t + Ax_{t-1}$ matrix to the input sequence can lead to arbitrary large matrix powers $x_{t+1} = \sum_{i=1}^{t} A^i x_{t-i}$, which are a particular form of matrix product. The most prominent example of matrix product appears in application of the chain rule, where each layer input/outputs can be vecotrized, leading to a chain of matrix Jaccobian products.

While matrix sums and particularly sums of random matrices have been subject of numerous extensive studies in variouas discliplines, matrix products are relatively much harder to analyze and understand. In the case of neural networks, understanding them becomes even more acute, due to various forms of non-linearity and inter-dependencies between units. In this thesis, we present several basic results on understanding and analyzing matrix products as it relates to various neural network modules and architectures.

We are broadly interested in how the depth of the matrix product chain affects signal forward and backward pass. This depth could correspond roughly to the number of layers in the case of non-recurrent networks, and lenght of the input sequence in the recurrent architectures. In plain words, we want to make sure that the forward and backward pass do not explode nor vanish at an exponential rate in depth, such that we can represent them accurately with floating point representation in CPU or GPU compute units. As we shall see, this objective of trying to keep forward and backward activations reasonably bounded is surprisingly challenging, which has a simple and straighforward description, can be very challenging or even impossible to obtain under certain conditions. As we will show later, this surprising challenges can be understood to a great extend by understanding matrix products.

While many of the observations and statements stated and derived here can be derived with alternative methods, such as Stielties transform and free probabiltiy, I am forcing myself to reproduce them via principles from basic probability, mostly to educate myself about the basic principles governing these problems and gain some basic insights into them.

I will use the example of of a linear MLP to illustrate the surprising complexities that emerge in

Case study: inear MLP

Let us start with a warmup example of a linear MLP: Let $x_\ell \in d_\ell$ be the activation for layer ℓ, with the layer update defined as

$$x_{\ell+1} = W_\ell x_\ell,$$

with W_ℓ a Gaussian matrix of size $d_{\ell+1} \times d_\ell$, with elements drawn iid from $N(0, s_\ell)$, where s_ℓ is a constant to be determined. We can also assume that $x_0 \in d_0$ is our input of size d_0. As mentioned before, we aim to ensure that our forward and backward activations are neither vanishing nor exploding.

Let us first analyze the forward activation. Since the activations are sign-symmetric, is clear that their expectation is zero $Ex_\ell = EW_\ell Ex_{\ell-1} = \mathbf{0}_{d_{\ell+1}}$. In order to make sure that the forward pass is stable, we can ensure the magnitude of each element of x_ℓ, such as its variance, is a constant. Equivalently, we can try to ensure $E\frac{1}{d}\|x_\ell\|^2 = 1$. If we exapnd the activation conditioned on the previous layer we have $E[\|x_\ell\|^2|x_{\ell-1}] = E\|W_\ell x_{\ell-1}\|^2$ which using linearity of expectation can be simplified to $x_{\ell-1}^\top (EW_\ell^\top W_\ell) x_{\ell-1} = d_{\ell-1} s_\ell \|x_{\ell-1}\|^2$. Thus, if we recursively calculate the norms of previous hidden activations we have $E\|x_\ell\|^2 = s_\ell d_{\ell-1} s_{\ell-1} d_{\ell-2} \dots s_1 d_0 \|x_0\|^2$. If the sequence $\{s_\ell d_{\ell-1}\}_{\ell=1,\dots,L}$ is even slightly larger or smaller than than 1, that will lead to explosion or vanishing values for norm of x_ℓ in ℓ. This observation immediately suggests setting variance of Gaussian layer to the inverse of its output dimension $s_\ell = 1/d_{\ell-1}$, which coincides with the Kaiming initialization(He et al. 2015).

Let us assume existence of a loss function $\mathcal{L} : R^{d_L} \to R$, which will map the L-th layer activations to a scalar value. The backward gradients with respect to a given layer $\partial\mathcal{L}(x_L)/\partial W_\ell$ can be computed via the chain rule as follows, which we can use to set the gradients of the weights

$$\frac{\partial \mathcal{L}(x_L)}{W_\ell} = W_{\ell+1}^\top \dots W_L^\top \frac{\partial \mathcal{L}(x_L)}{\partial x_L} x_\ell^\top$$

if we expand x_ℓ, we can set gradients backwards to

$$dW_\ell := \left(W_{\ell+1}^\top \dots W_L^\top \frac{\partial \mathcal{L}(x_L)}{\partial x_L} \right) \otimes \left(W_{\ell-1} \dots W_0 x_0 \right),$$

where $\otimes$ denotes Hadamard product. Owing to the fact that Jacobian of its layers are equal to the weight matrices, the matrix product chain is also appearing in the backward gradients, which is one of the distinctive features of linear MLP. Thus, with a similar argument as before, we can argue that by setting the variances s_ℓ's according the criteria that was mentioned before, we will also ensure that the backward gradients are stable (using the fact that $\|u \otimes v\|_F^2 = \|u\|^2 \cdot \|v\|^2$)

$$E\|dW_\ell\|_F^2 = E\|W_{\ell+1}^\top \dots W_L^\top \frac{\partial \mathcal{L}(x_L)}{\partial x_L}\|^2 \cdot E\|W_{\ell-1} \dots W_0 x_0\|^2 = E\|\frac{\partial \mathcal{L}(x_L)}{\partial x_L}\|^2 \|x_0\|^2,$$

If we assume inputs and the gradients with respect to the last hidden layer to be of some constant magnitude $\|x_0\|, \|\partial\mathcal{L}(x_L)/\partial x_L\| = O(1)$, we can similarly infer that the backward gradients are of also of constant magnitude $O(1)$. At this point, we might conclude that the problem of stabilising forward and backward pass for linear MLP has been essentially resolved. However, a simulation of this MLP reveals a strange and mysterious behavior.

Long-tailed distribution of activations in depth

As we argued before, setting variances of layer ℓ to $1/d_{\ell-1}$, ensures that this quantity $\|x_L\|/\|x_0\|$, is on average equal to 1. Let us delve a bit deeper into the distribution of forward activations by plotting $\|x_L\|$ for random choices of the Gaussian weights, when input vector is a uniformly drawn unit vector $\|x_0\| = 1$.

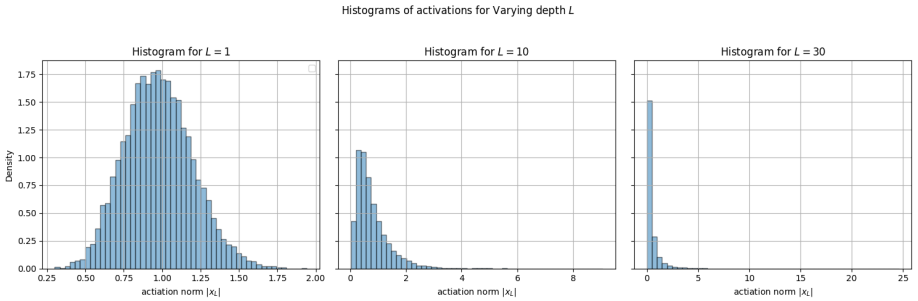

Figure 1: Histogram of forward pass activations $\|x_L\|$ for width $d = 10$, random choices of Gaussian weights, and a uniformly drawn unit vector as input $\|x_0\| = 1$. Depth L is varied to see the effect on the distribution,

As shown in fig. 1, when depth is small ($L = 1$,) the mean that we calculated before is predictive of the actual values of our activations. as the subfigure for $L = 20$ demonstrates, as the network becomes deeper, the distribution becomes more heavi-tailed, while the bulk of its values are very small. If we test an ever deeper network, such as $L = 100$, more than half of the values are below < 0.01, while in about 1 in every 1000 network, the value was higher than 10.! While the ean activation is the average of a few unlikely evenets that the norm is above large (> 1), and the majority of the events that the value is nearly zero.

In plain words, as the network depth grows, majority of the initialized networks exhibit vanishing activations, while a small number of these networks have exploding activations. Therefore, the average behavior for a randomly initilized network is the mean between these two extremes, which gives us the false impression that forward activations are stable! In statistical terms, the mean value is only a predictive statistics for light-tailed symmetric distributions, and when the depth grows, the distribution of activations becomes heavy-tailed and highly skewed.

Stabilizing forward pass via normalization

One remedy for stabilising forward activations is a simple normalization, such as Root Mean Squared (RMS) normalizatioin:

$$\|x\|_{rms} = \sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i}$$

This block can be applied on top of our MLP activations to ensure they have constant norms:

$$x_{\ell+1} = W_\ell x / \|W_\ell x\|_{rms}, \quad \ell = 1, \ldots, L$$

Thus, it will always hold trivially $E\|x_L\|^2 = d_L$. While this might seem like a

A the distance between ℓ and L grows, the matrix product chain $W_L \ldots W_{\ell+1}$ will also grow. Thus, spectral properties of this Gaussian product chain will be highly impactful on our gradients. It is a well-known mathematical fact the product of a chain of Gaussian matrices converges to a rank-1 matrix as the number of these matrices increases (Latouche and Ramaswami 1999). This is clearly poses a problem for the backward gradients. Let v denote the singular vector corresponding to this non-zero non-zero singular value, which is randomly oriented in space. Thus, the cosine between the top left eigenvecotr of $\partial\mathcal{L}(x_L)/\partial x_L$ will be of the order $O(1/\sqrt{d_L})$, and the cosine between the top right singular vector and x_ℓ will be $O(1/\sqrt{d_\ell})$. In other words, as the width of different layers grows $d_\ell, d_L \to \infty$, the gradient of loss will converge towards zero with $O(1/\sqrt{d_\ell d_L})$.

While normally gradients are taken for loss with respect to the parameters, layer-to-layer Jaccobians form the bulk of the matrix chain and determine its properties. Thus, to understand the asymptotic behavior, we can focus on the input-output Jaccobian. reasons that will become clear layer, let us consider n inputs as columns of minibatch $X_\ell \in R^{d_\ell \times n}$, being passed through the network with the same update $X_\ell = W_\ell X_{\ell-1}$.

Let's try citing a paper (Braun and McAuliffe 2010) What do neural nets and financial time se

Braun, Michael, and Jon McAuliffe. 2010. "Variational Inference for Large-Scale Models of Discrete Choice." *Journal of the American Statistical Association* 105 (489): 324–35.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on Imagenet Classification." In *Proceedings of the IEEE International Conference on Computer Vision*, 1026–34.

Latouche, Guy, and Vaidyanathan Ramaswami. 1999. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. SIAM.