information matrix:

$$ds^2 = \mathrm{D}_{\mathrm{KL}}\big[P(y \,|\, \mathbf{x};\ \theta) \| P(y \,|\, \mathbf{x};\ \theta + \delta)\big] \approx \frac{1}{2}\delta^\top F(\theta)\delta, \tag{8}$$

$$F(\theta) = \mathop{\mathbb{E}}_{\mathbf{x}\sim P(\mathbf{x}), y \sim P(y \,|\, \mathbf{x})}\left[\frac{\partial \log P(y \,|\, \mathbf{x};\ \theta)}{\partial \theta}\frac{\partial \log P(y \,|\, \mathbf{x};\ \theta)}{\partial \theta}^\top\right], \tag{9}$$

where, δ is a small change to the parameters. The Riemannian metric above presents a geometric view of parameter spaces. The following analysis of the Riemannian metric provides some insight into how normalization methods could help in training neural networks.

5.2.2 The geometry of normalized generalized linear models

We focus our geometric analysis on the generalized linear model. The results from the following analysis can be easily applied to understand deep neural networks with block-diagonal approximation to the Fisher information matrix, where each block corresponds to the parameters for a single neuron.

A generalized linear model (GLM) can be regarded as parameterizing an output distribution from the exponential family using a weight vector w and bias scalar b. To be consistent with the previous sections, the log likelihood of the GLM can be written using the summed inputs a as the following:

$$\log P(y \,|\, \mathbf{x};\ w, b) = \frac{(a+b)y - \eta(a+b)}{\phi} + c(y, \phi), \tag{10}$$

$$\mathbb{E}[y \,|\, \mathbf{x}] = f(a+b) = f(w^\top \mathbf{x} + b), \ \ \mathrm{Var}[y \,|\, \mathbf{x}] = \phi f'(a+b), \tag{11}$$

where, $f(\cdot)$ is the transfer function that is the analog of the non-linearity in neural networks, $f'(\cdot)$ is the derivative of the transfer function, $\eta(\cdot)$ is a real valued function and $c(\cdot)$ is the log partition function. ϕ is a constant that scales the output variance. Assume a H-dimensional output vector $\mathbf{y} = [y_1, y_2, \cdots, y_H]$ is modeled using H independent GLMs and $\log P(\mathbf{y} \,|\, \mathbf{x};\ W, \mathbf{b}) = \sum_{i=1}^H \log P(y_i \,|\, \mathbf{x};\ w_i, b_i)$. Let W be the weight matrix whose rows are the weight vectors of the individual GLMs, $\mathbf{b}$ denote the bias vector of length H and $\mathrm{vec}(\cdot)$ denote the Kronecker vector operator. The Fisher information matrix for the multi-dimensional GLM with respect to its parameters $\theta = [w_1^\top, b_1, \cdots, w_H^\top, b_H]^\top = \mathrm{vec}([W, \mathbf{b}]^\top)$ is simply the expected Kronecker product of the data features and the output covariance matrix:

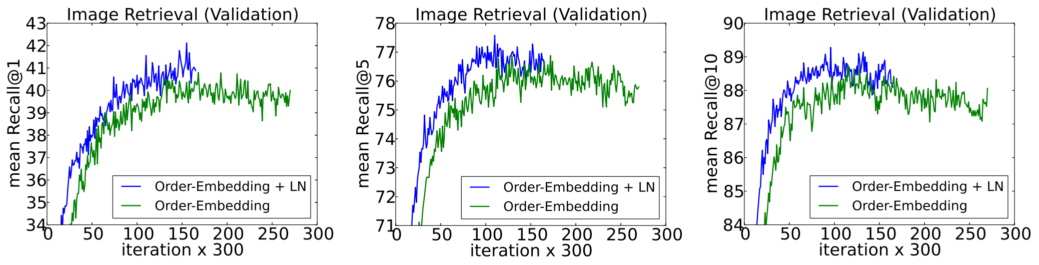
$$F(\theta) = \mathop{\mathbb{E}}_{\mathbf{x}\sim P(\mathbf{x})}\left[\frac{\mathrm{Cov}[\mathbf{y} \,|\, \mathbf{x}]}{\phi^2} \otimes \begin{bmatrix} \mathbf{x}\mathbf{x}^\top & \mathbf{x} \\ \mathbf{x}^\top & 1 \end{bmatrix}\right]. \tag{12}$$

We obtain normalized GLMs by applying the normalization methods to the summed inputs a in the original model through μ and σ. Without loss of generality, we denote $\bar{F}$ as the Fisher information matrix under the normalized multi-dimensional GLM with the additional gain parameters $\theta = \mathrm{vec}([W, \mathbf{b}, \mathbf{g}]^\top)$:

$$\bar{F}(\theta) = \begin{bmatrix} \bar{F}_{11} & \cdots & \bar{F}_{1H} \\ \vdots & \ddots & \vdots \\ \bar{F}_{H1} & \cdots & \bar{F}_{HH} \end{bmatrix}, \quad \bar{F}_{ij} = \mathop{\mathbb{E}}_{\mathbf{x}\sim P(\mathbf{x})}\left[\frac{\mathrm{Cov}[y_i,\ y_j \ |\, \mathbf{x}]}{\phi^2}\begin{bmatrix} \frac{g_i g_j}{\sigma_i \sigma_j}\chi_i \chi_j^\top & \chi_i \frac{g_i}{\sigma_i} & \chi_i \frac{g_i(a_j - \mu_j)}{\sigma_i \sigma_j} \\ \chi_j^\top \frac{g_j}{\sigma_j} & 1 & \frac{a_j - \mu_j}{\sigma_j} \\ \chi_j^\top \frac{g_j(a_i - \mu_i)}{\sigma_i \sigma_j} & \frac{a_i - \mu_i}{\sigma_i} & \frac{(a_i - \mu_i)(a_j - \mu_j)}{\sigma_i \sigma_j} \end{bmatrix}\right] \tag{13}$$

$$\chi_i = \mathbf{x} - \frac{\partial \mu_i}{\partial w_i} - \frac{a_i - \mu_i}{\sigma_i}\frac{\partial \sigma_i}{\partial w_i}. \tag{14}$$

Implicit learning rate reduction through the growth of the weight vector: Notice that, comparing to standard GLM, the block $\bar{F}_{ij}$ along the weight vector w_i direction is scaled by the gain parameters and the normalization scalar σ_i. If the norm of the weight vector w_i grows twice as large, even though the model's output remains the same, the Fisher information matrix will be different. The curvature along the w_i direction will change by a factor of $\frac{1}{2}$ because the σ_i will also be twice as large. As a result, for the same parameter update in the normalized model, the norm of the weight vector effectively controls the learning rate for the weight vector. During learning, it is harder to change the orientation of the weight vector with large norm. The normalization methods, therefore,



(a) Recall@1 (b) Recall@5 (c) Recall@10

Figure 1: Recall@K curves using order-embeddings with and without layer normalization.

MSCOCO								
	Caption Retrieval				Image Retrieval			
Model	**R@1**	**R@5**	**R@10**	**Mean** *r*	**R@1**	**R@5**	**R@10**	**Mean** *r*
Sym [Vendrov et al., 2016]	45.4		88.7	5.8	36.3		85.8	9.0
OE [Vendrov et al., 2016]	46.7		88.9	5.7	37.9		85.9	8.1
OE (ours)	46.6	79.3	89.1	5.2	37.8	73.6	85.7	7.9
OE + LN	**48.5**	**80.6**	**89.8**	**5.1**	**38.9**	**74.3**	**86.3**	**7.6**

Table 2: Average results across 5 test splits for caption and image retrieval. **R@K** is Recall@K (high is good). **Mean** *r* is the mean rank (low is good). Sym corresponds to the symmetric baseline while OE indicates order-embeddings.

have an implicit "early stopping" effect on the weight vectors and help to stabilize learning towards convergence.

Learning the magnitude of incoming weights: In normalized models, the magnitude of the incoming weights is explicitly parameterized by the gain parameters. We compare how the model output changes between updating the gain parameters in the normalized GLM and updating the magnitude of the equivalent weights under original parameterization during learning. The direction along the gain parameters in $\bar{F}$ captures the geometry for the magnitude of the incoming weights. We show that Riemannian metric along the magnitude of the incoming weights for the standard GLM is scaled by the norm of its input, whereas learning the gain parameters for the batch normalized and layer normalized models depends only on the magnitude of the prediction error. Learning the magnitude of incoming weights in the normalized model is therefore, more robust to the scaling of the input and its parameters than in the standard model. See Appendix for detailed derivations.

6 Experimental results

We perform experiments with layer normalization on 6 tasks, with a focus on recurrent neural networks: image-sentence ranking, question-answering, contextual language modelling, generative modelling, handwriting sequence generation and MNIST classification. Unless otherwise noted, the default initialization of layer normalization is to set the adaptive gains to 1 and the biases to 0 in the experiments.

6.1 Order embeddings of images and language

In this experiment, we apply layer normalization to the recently proposed order-embeddings model of Vendrov et al. [2016] for learning a joint embedding space of images and sentences. We follow the same experimental protocol as Vendrov et al. [2016] and modify their publicly available code to incorporate layer normalization [1] which utilizes Theano [Team et al., 2016]. Images and sentences from the Microsoft COCO dataset [Lin et al., 2014] are embedded into a common vector space, where a GRU [Cho et al., 2014] is used to encode sentences and the outputs of a pre-trained VGG ConvNet [Simonyan and Zisserman, 2015] (10-crop) are used to encode images. The order-embedding model represents images and sentences as a 2-level partial ordering and replaces the cosine similarity scoring function used in Kiros et al. [2014] with an asymmetric one.

[1] `https://github.com/ivendrov/order-embedding`