Spherical Bayesian Learning

The Hyperspherical Uniform Prior

Leif-Martin Sæther Sunde

UiO : Department of Mathematics
University of Oslo

Table of contents

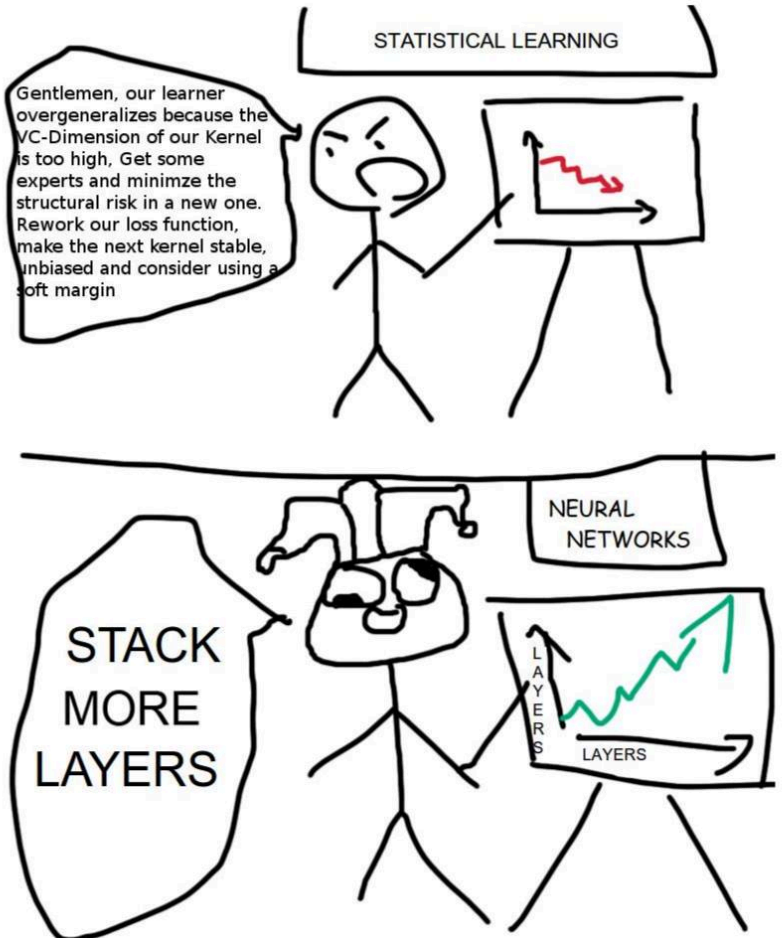
Intro

Deep Learning Framework

- Massive Datasets
- Massive Models

Deep Learning Framework

- Massive Datasets
- Massive Models

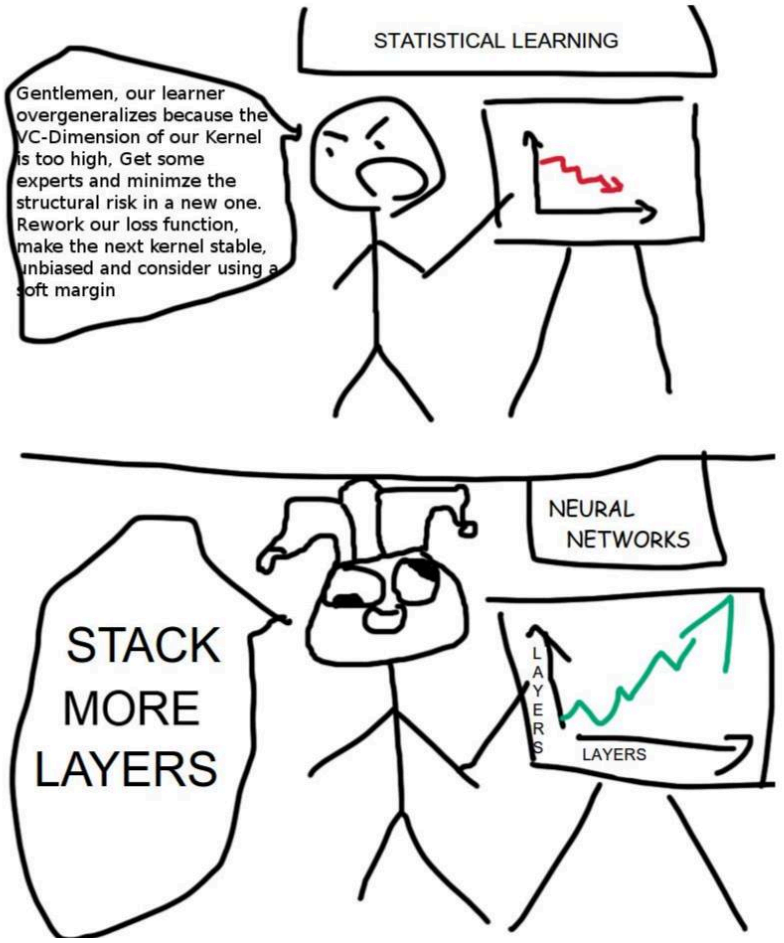


Presentation slide from CS282A at UC Berkeley, Spring 2022.

Deep Learning Framework

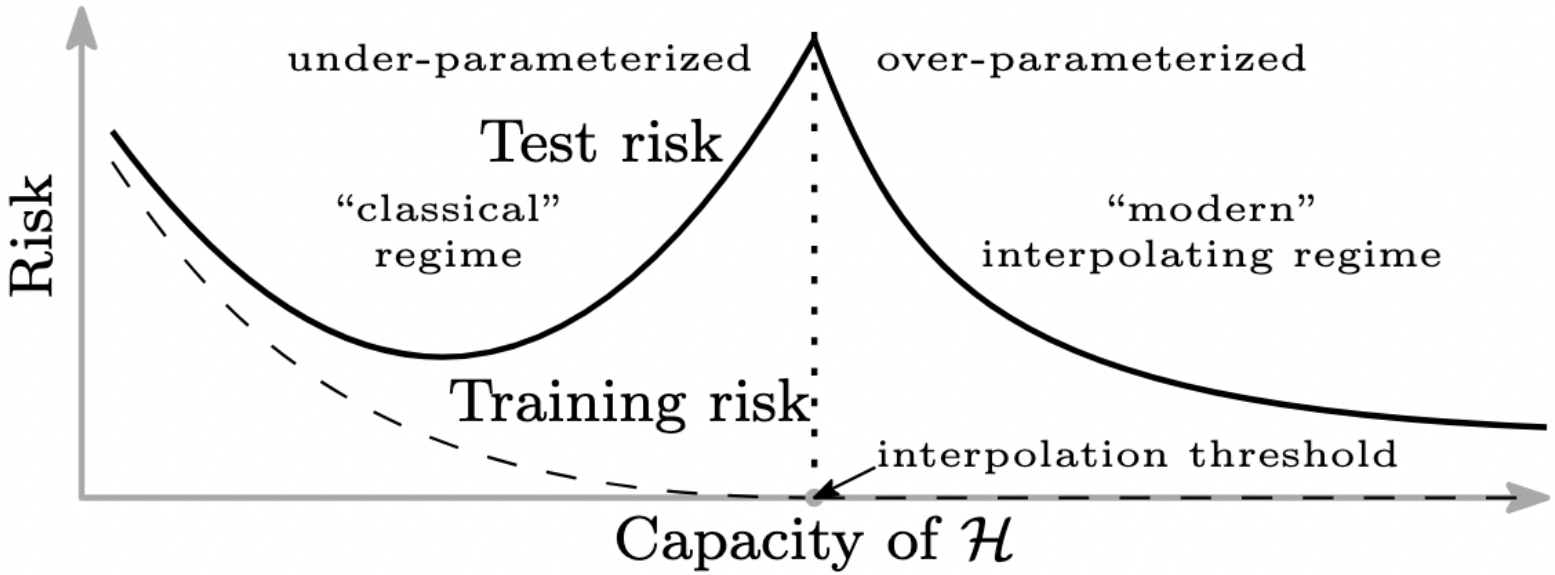
- Massive Datasets
- Massive Models

- Belkin et al. (2019); Sejnowski (2020); Sevilla et al. (2022); Belkin et al. (2020); Nakkiran et al. (2019); Mei and Montanari (2021)

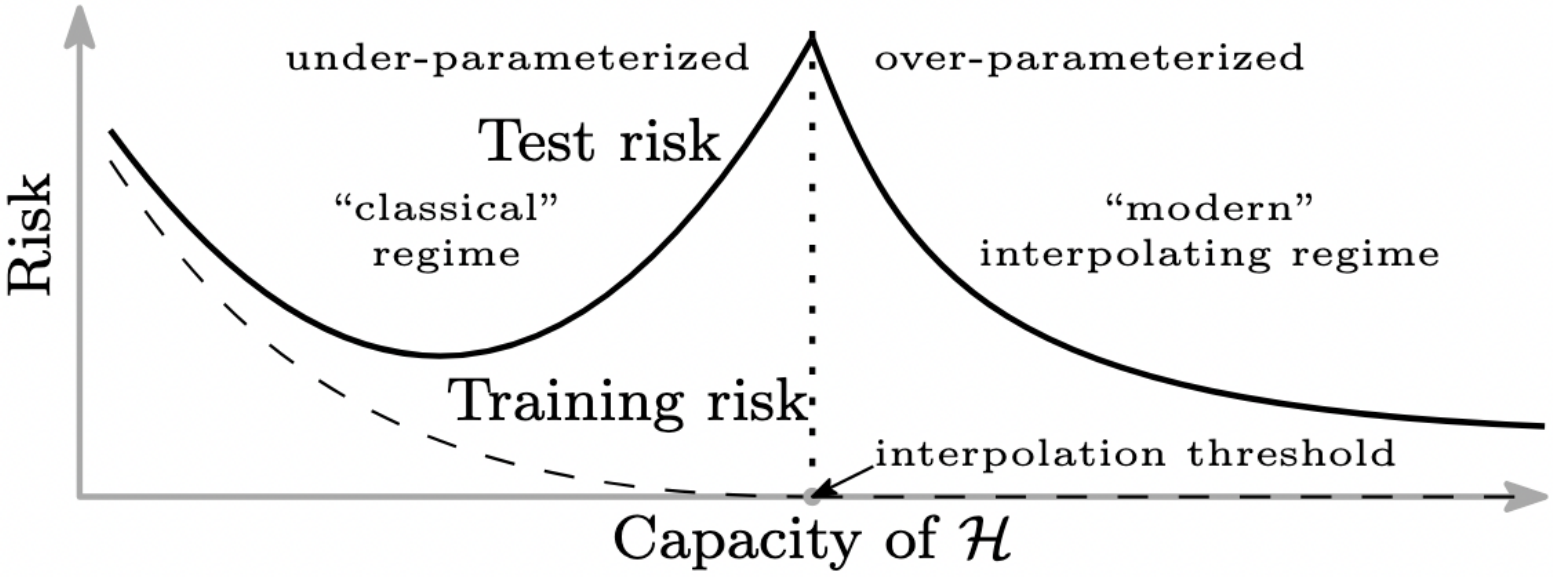


Presentation slide from CS282A at UC Berkeley, Spring 2022.

Massive Models Perform Well for Massive Data



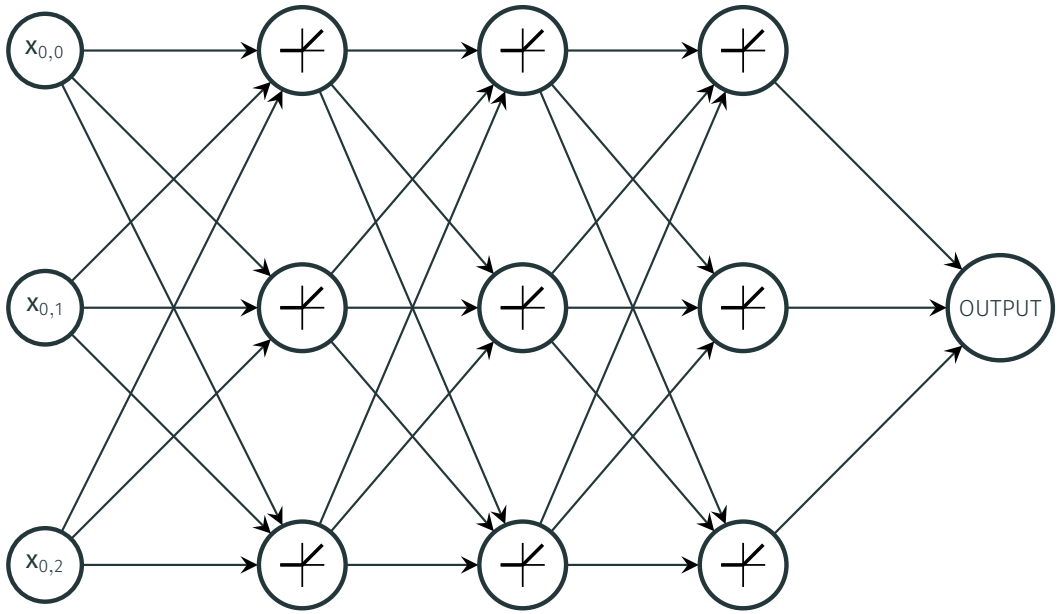
Massive Models Perform Well for Massive Data



Models with massive capacity are fruitful!

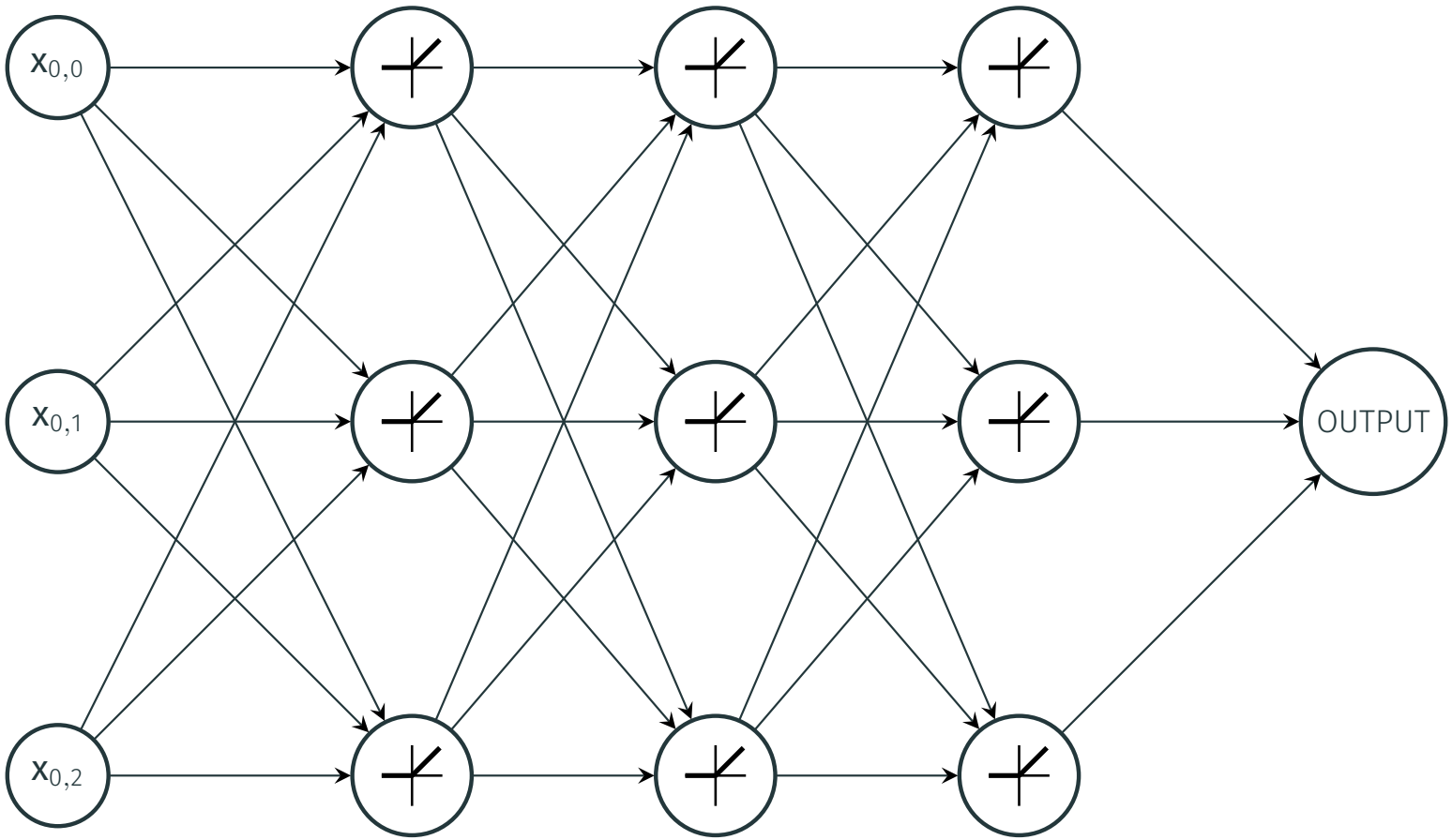
Neural Networks

- Forward Pass for making prediction.
- Backward Pass to find gradient of loss.

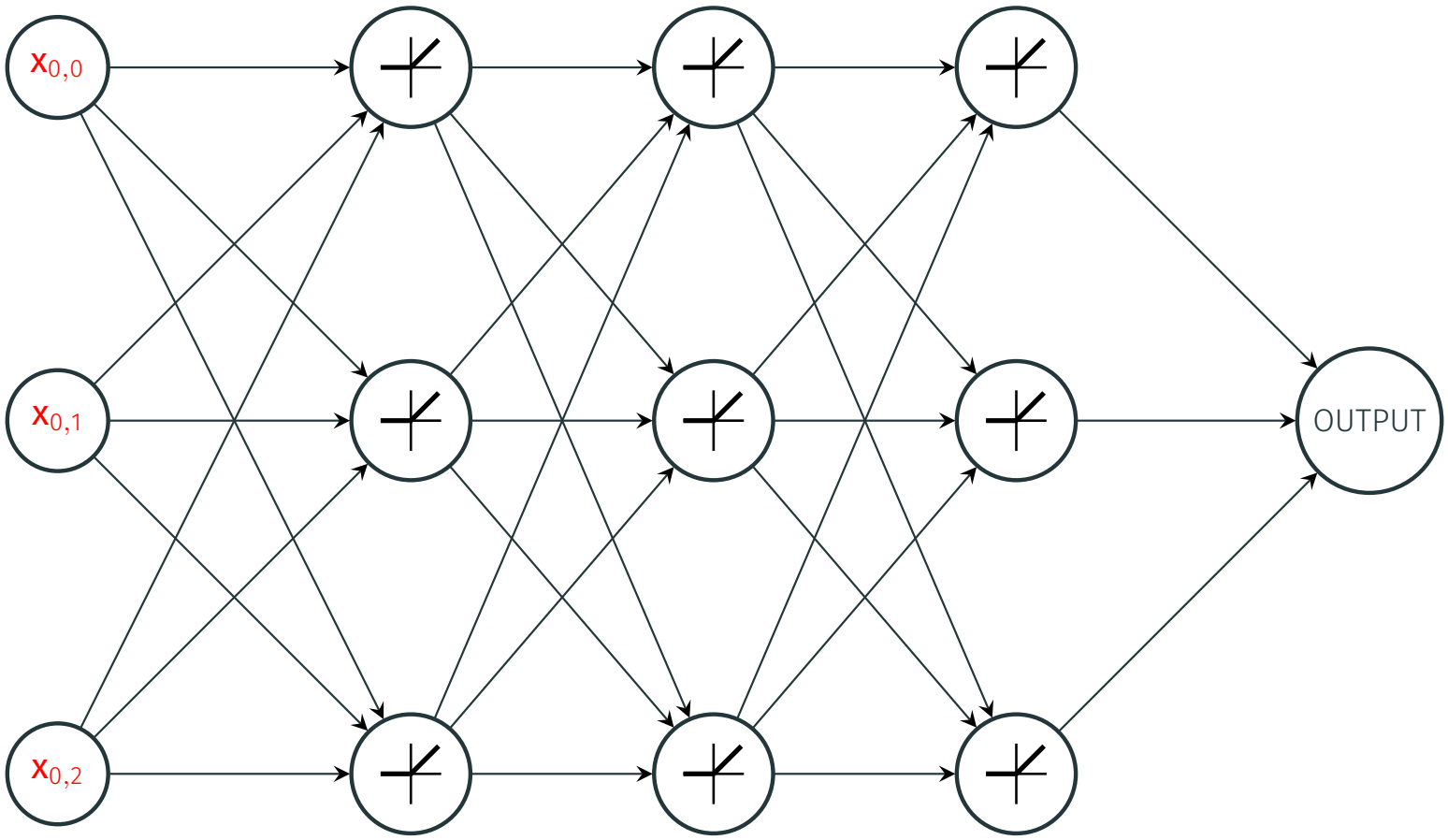


Forward Pass

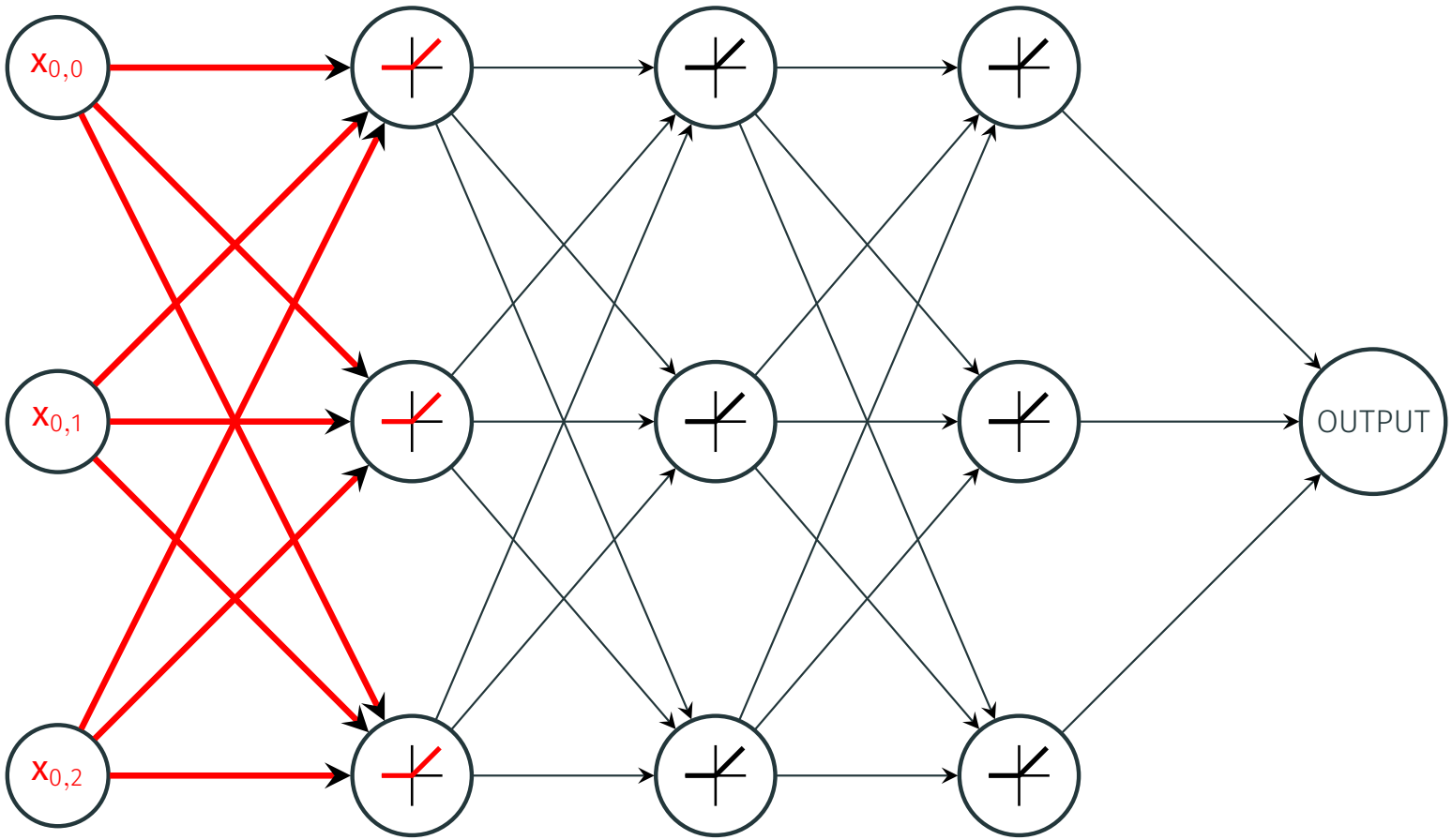
Forward Pass



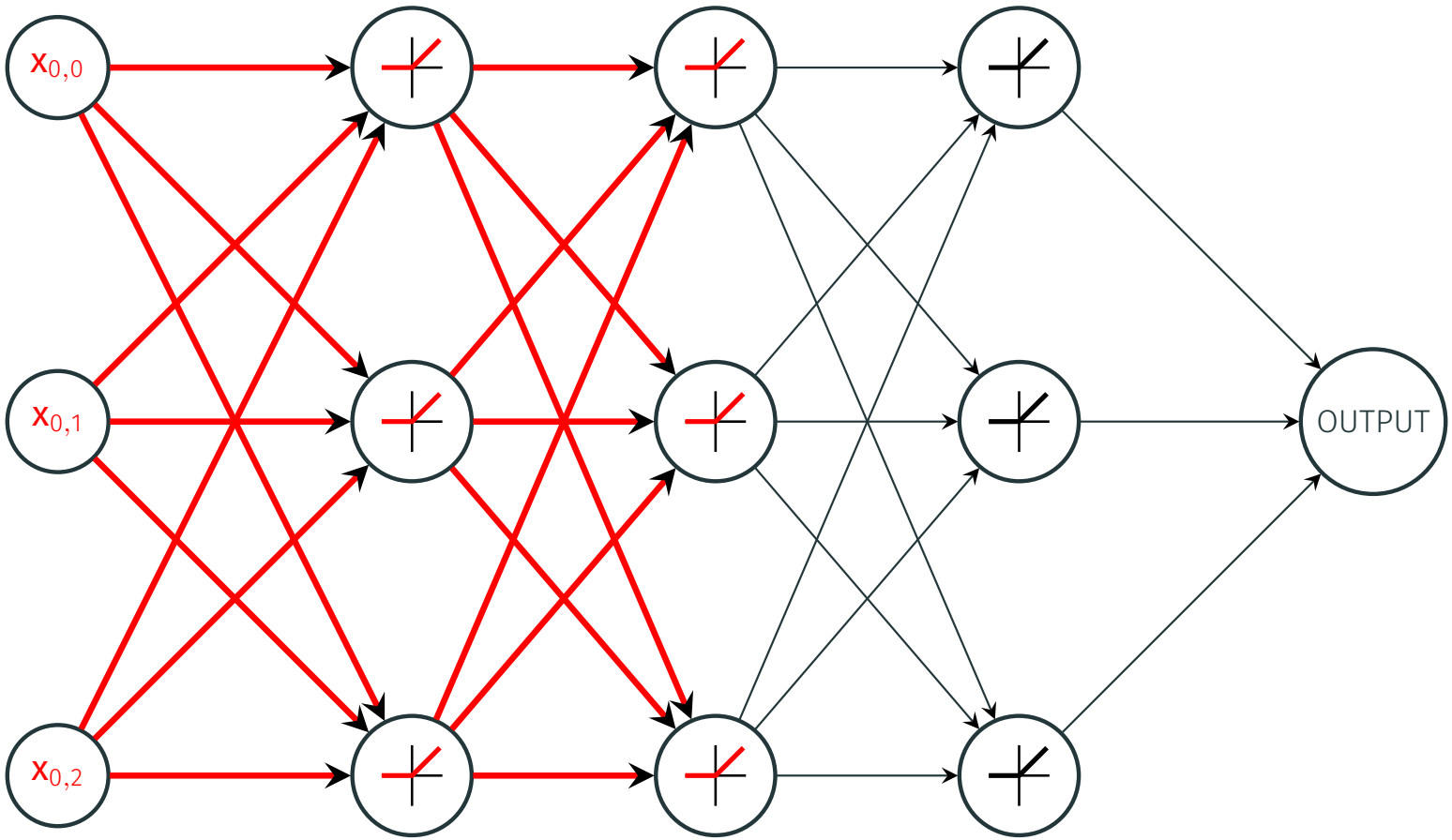
Forward Pass



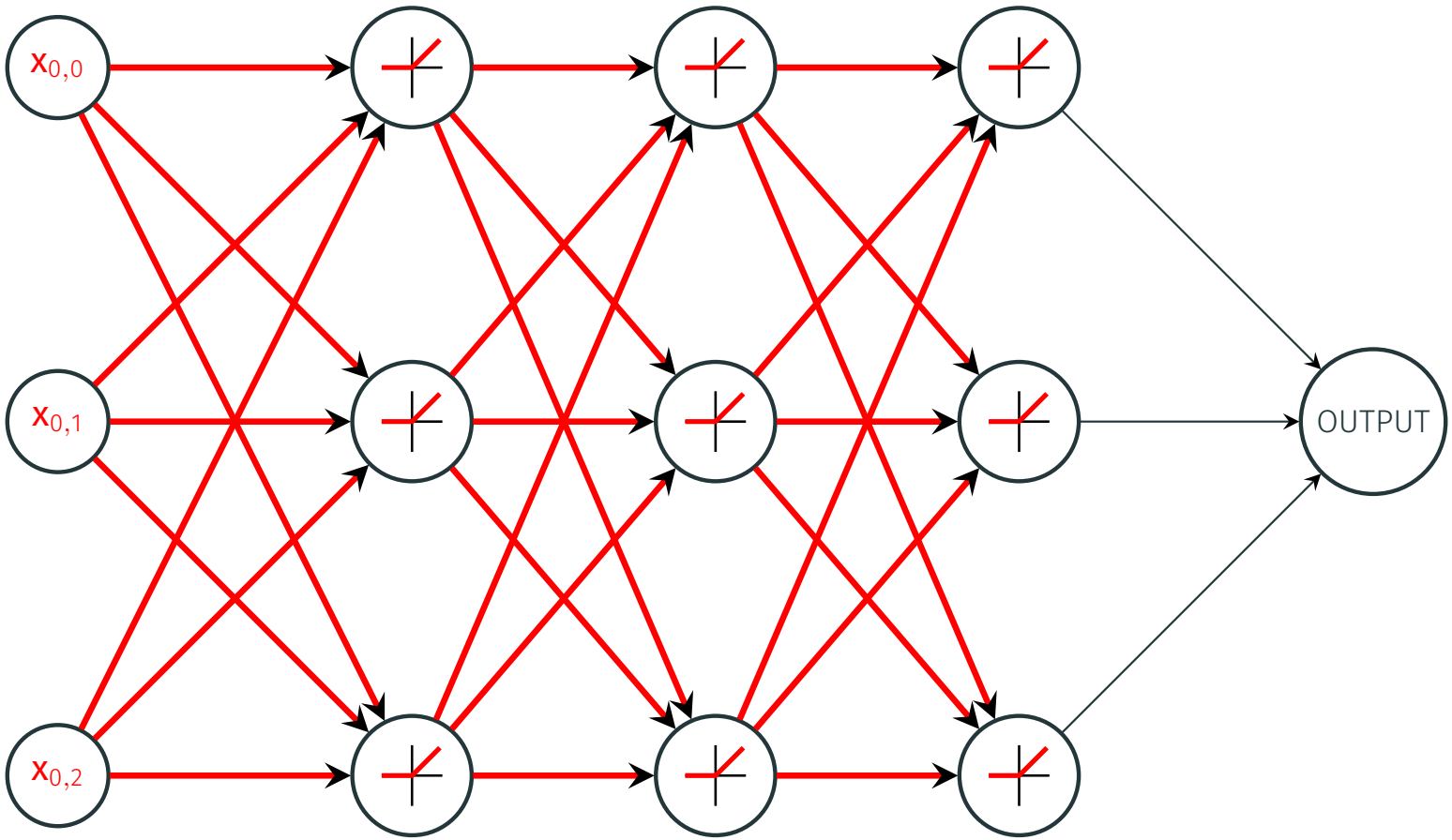
Forward Pass



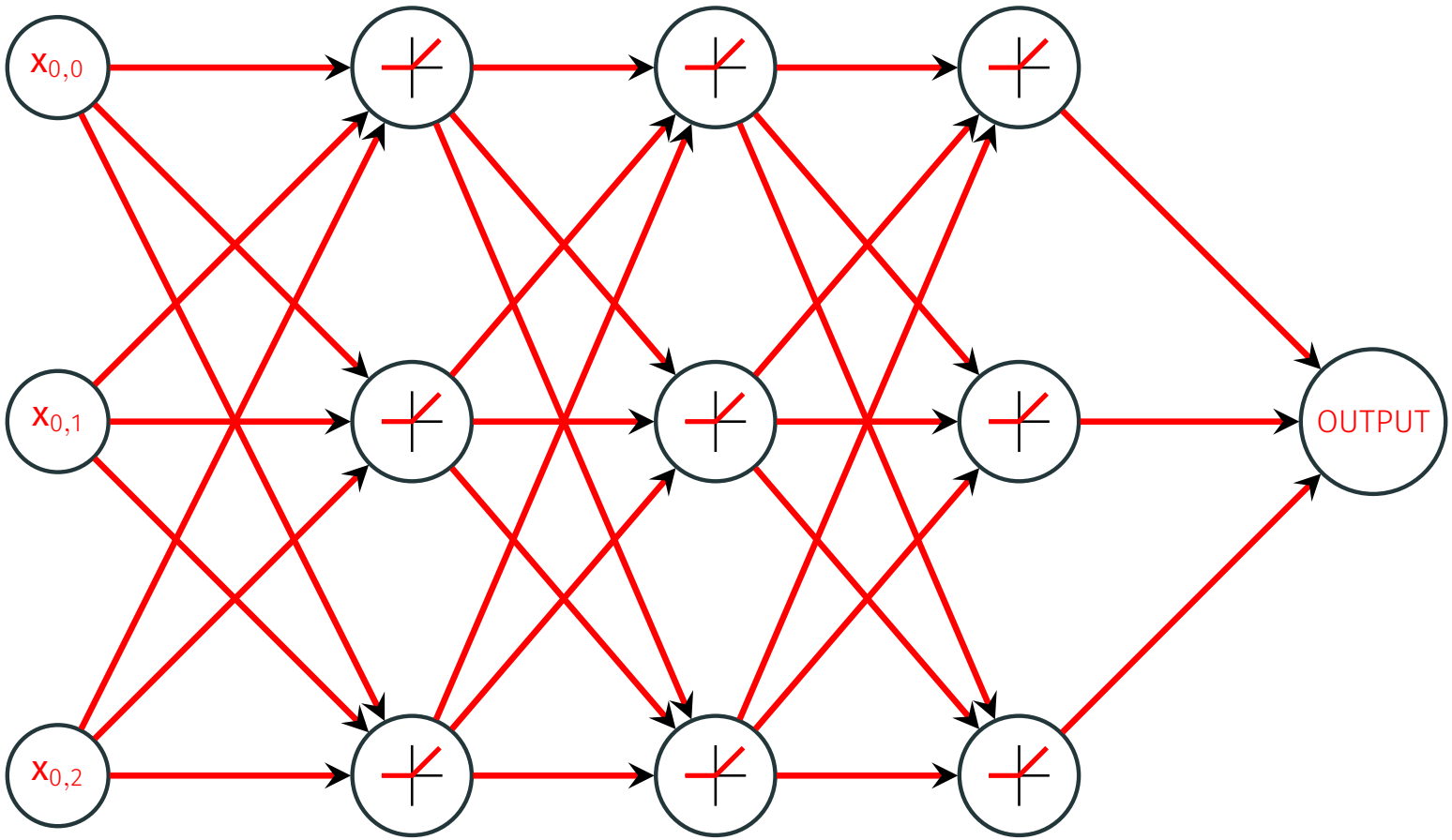
Forward Pass



Forward Pass

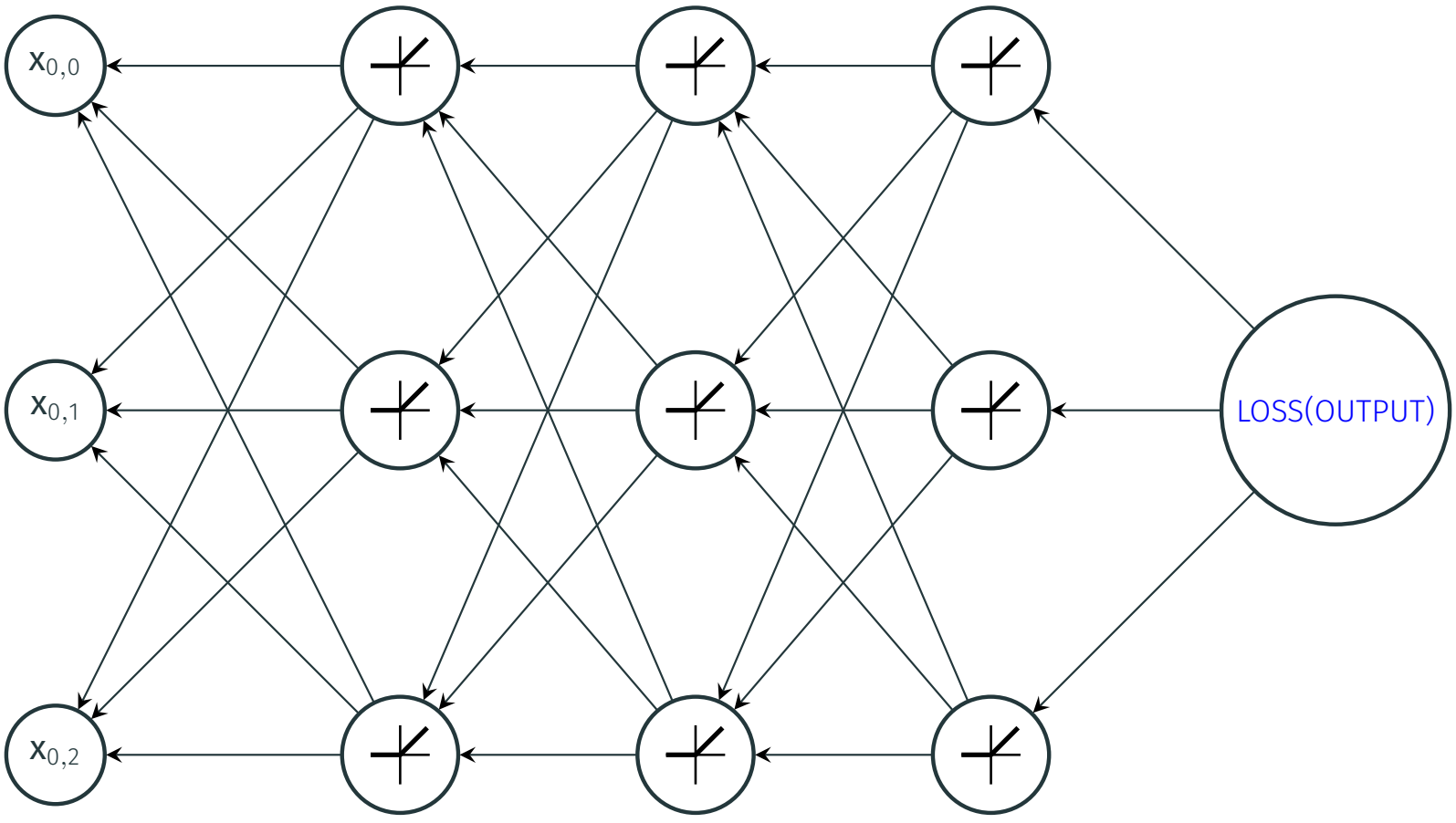


Forward Pass

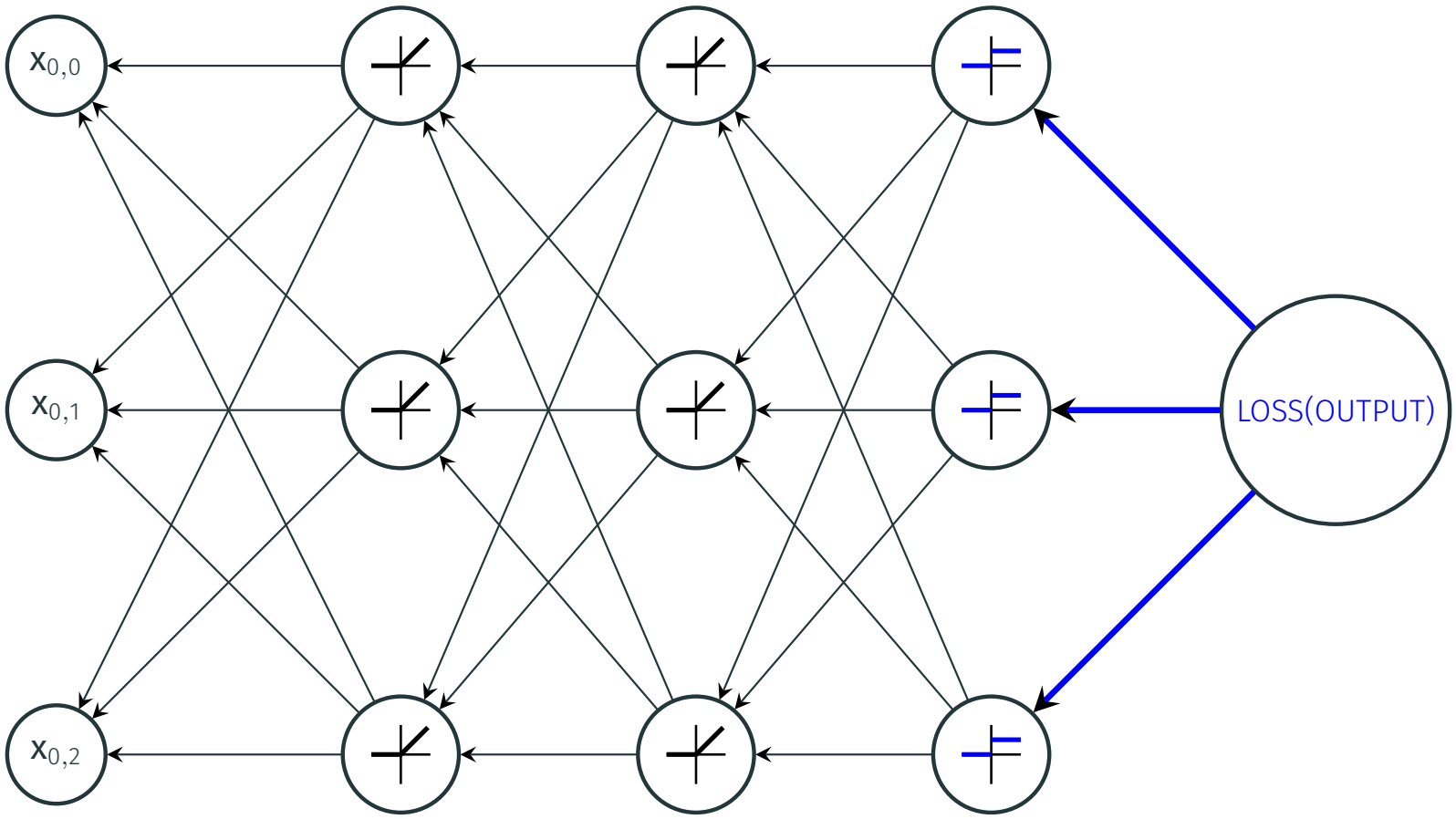


Backward Pass

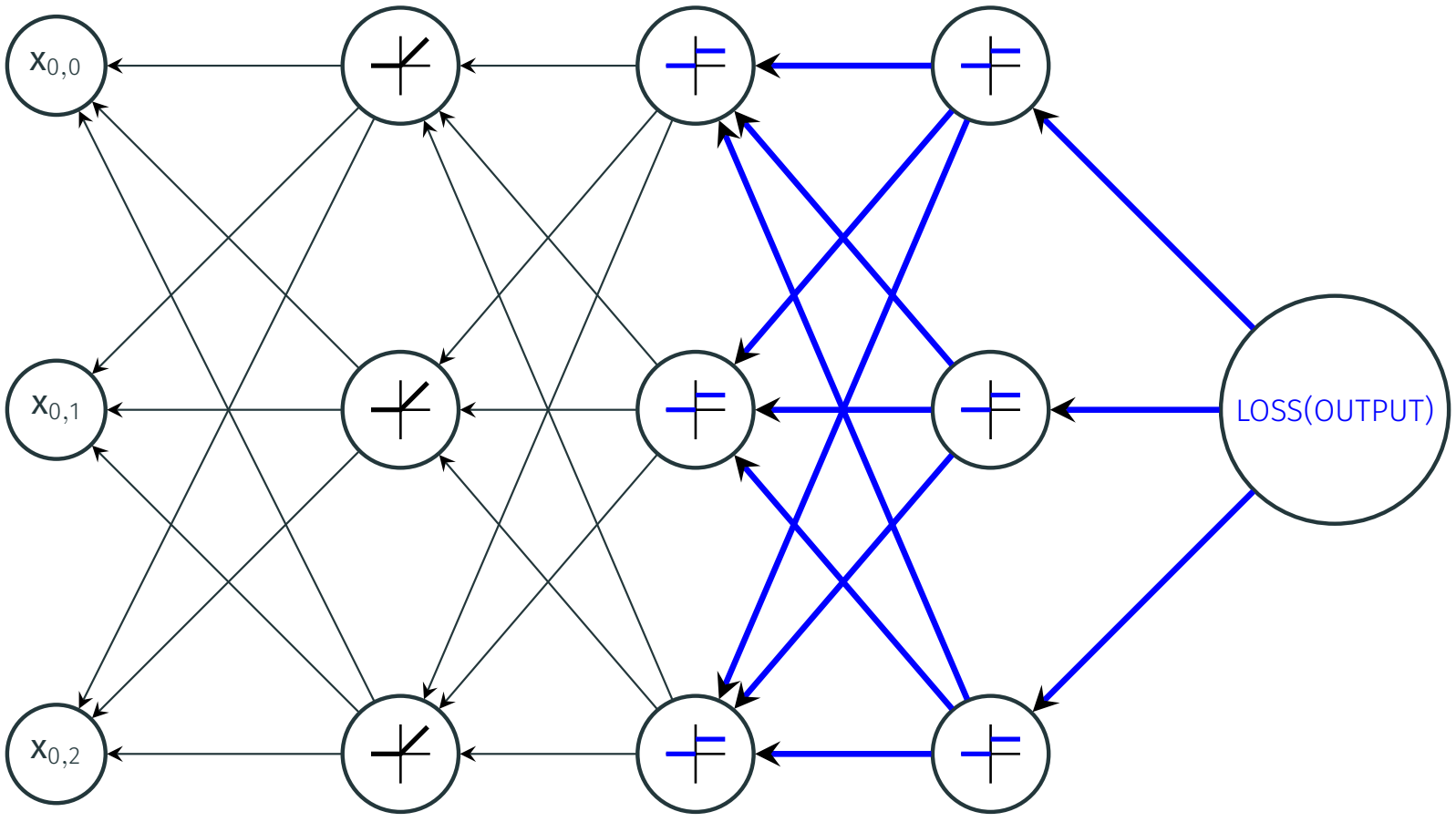
Backward Pass



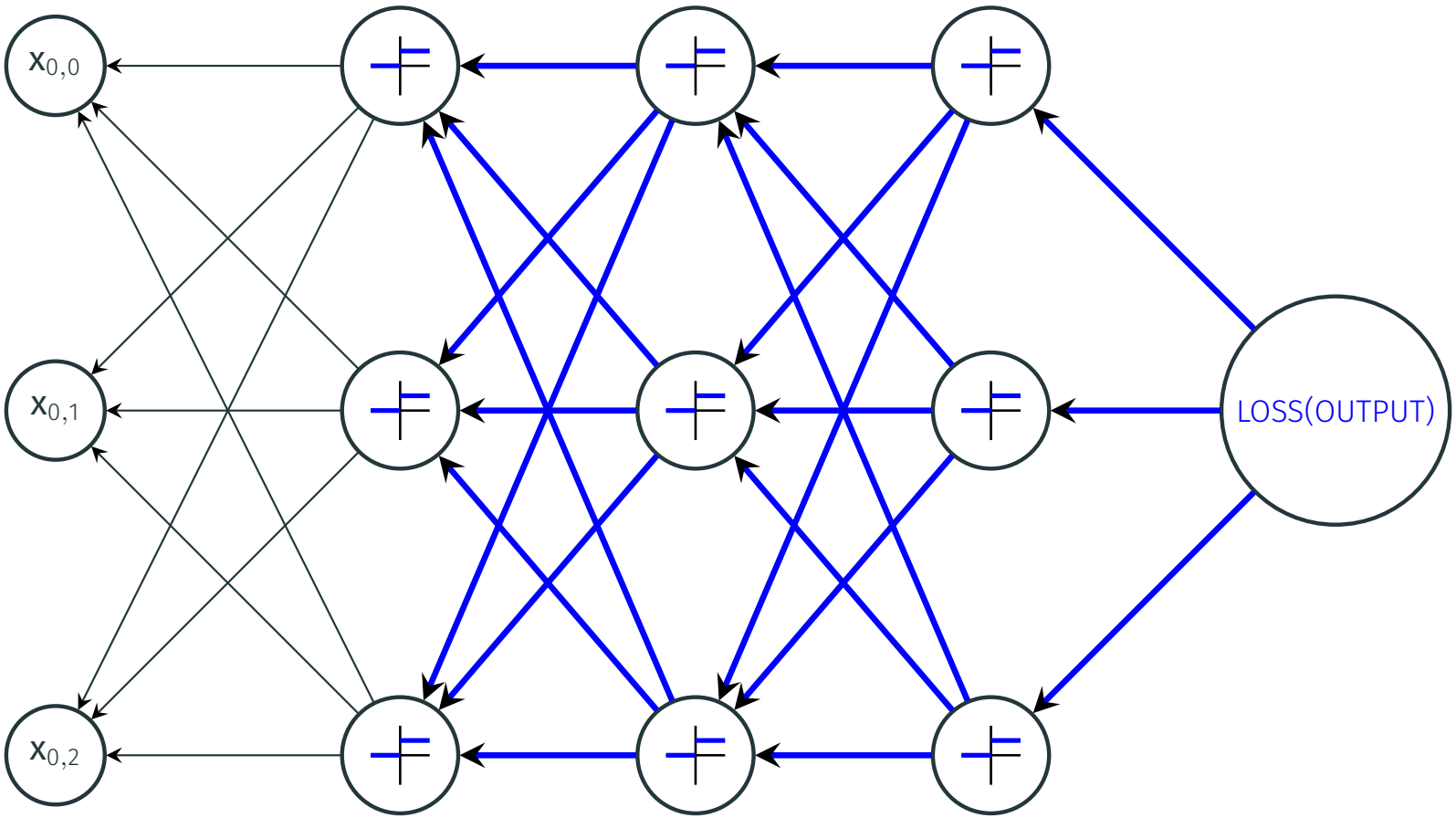
Backward Pass



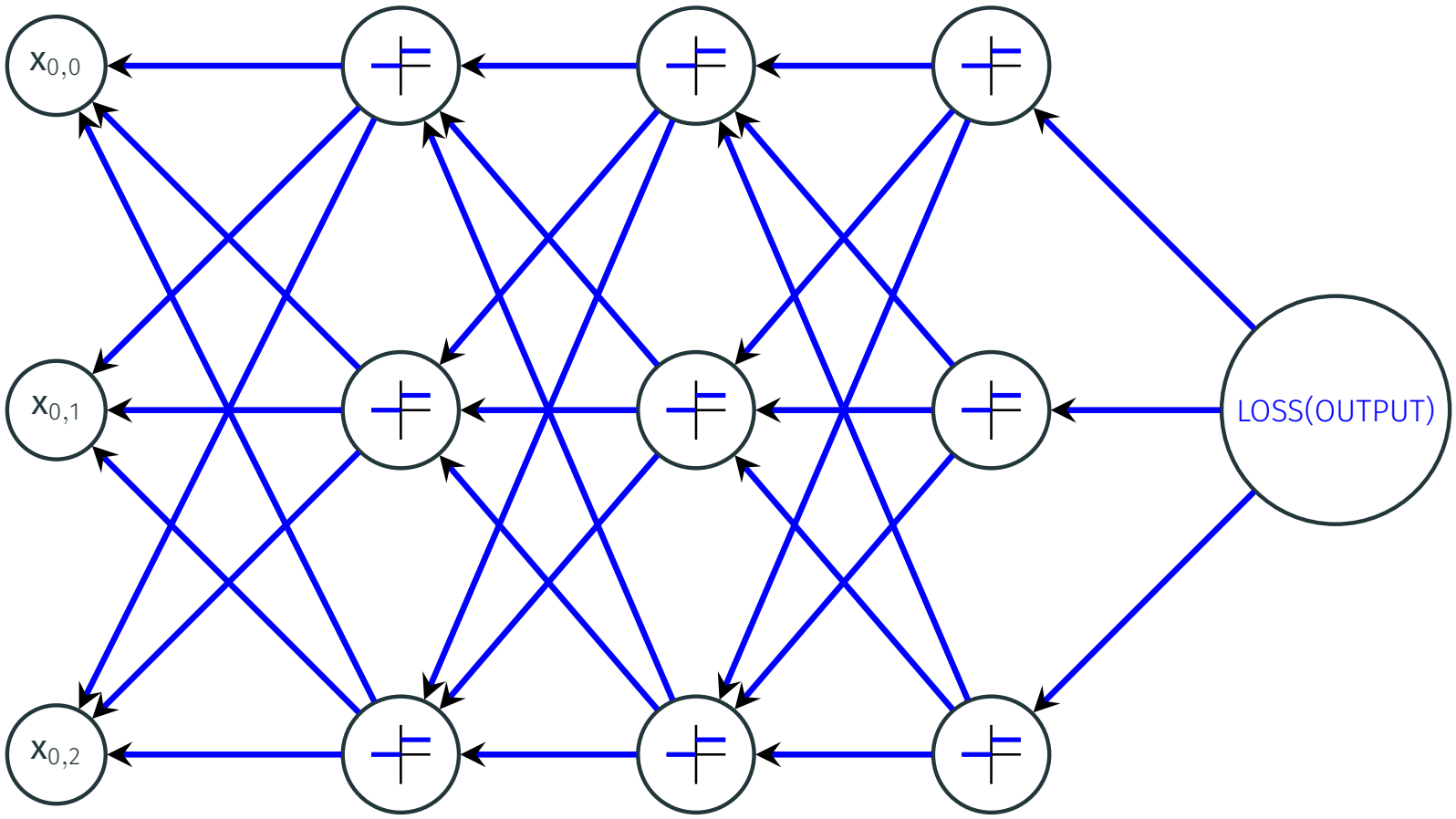
Backward Pass



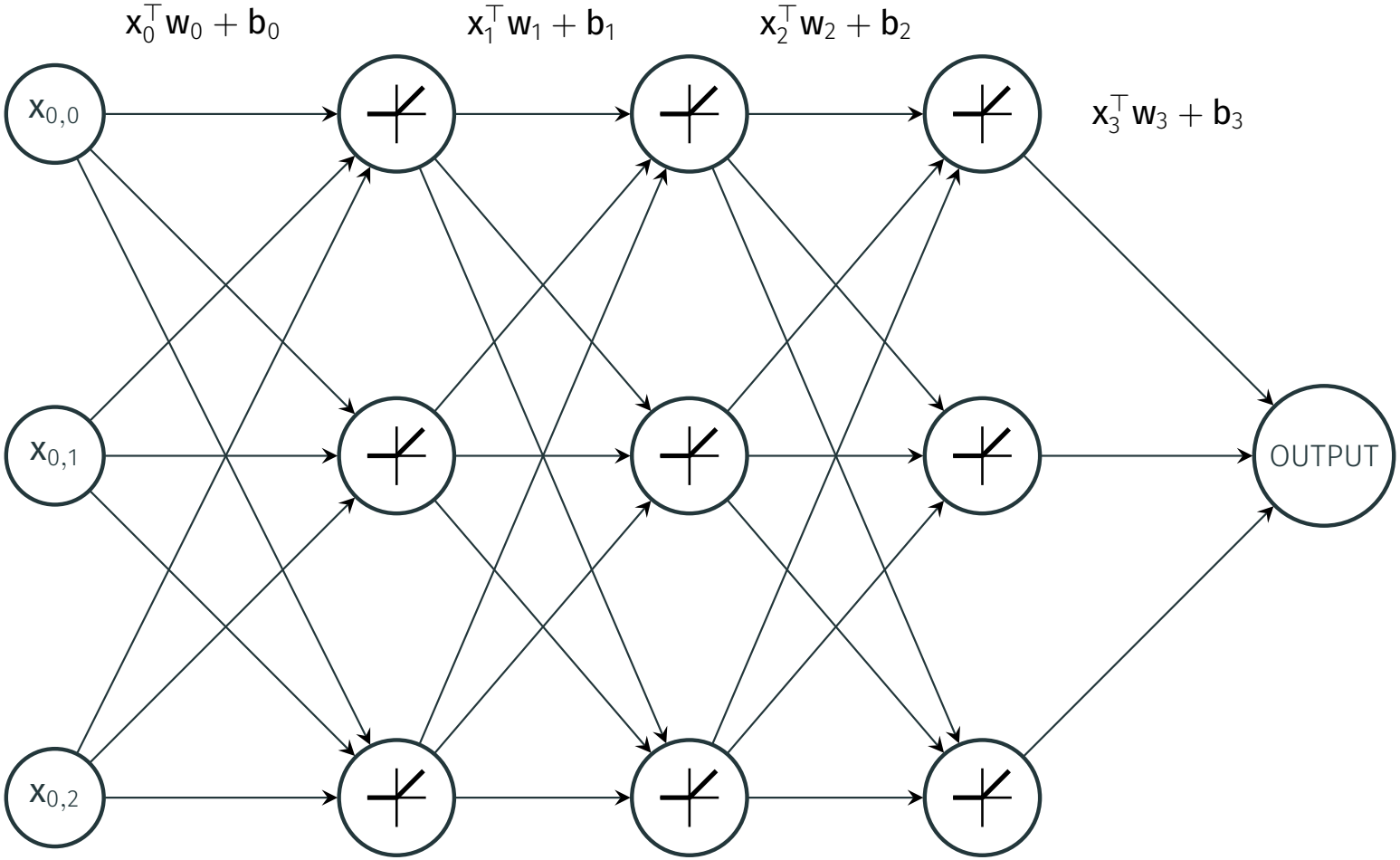
Backward Pass



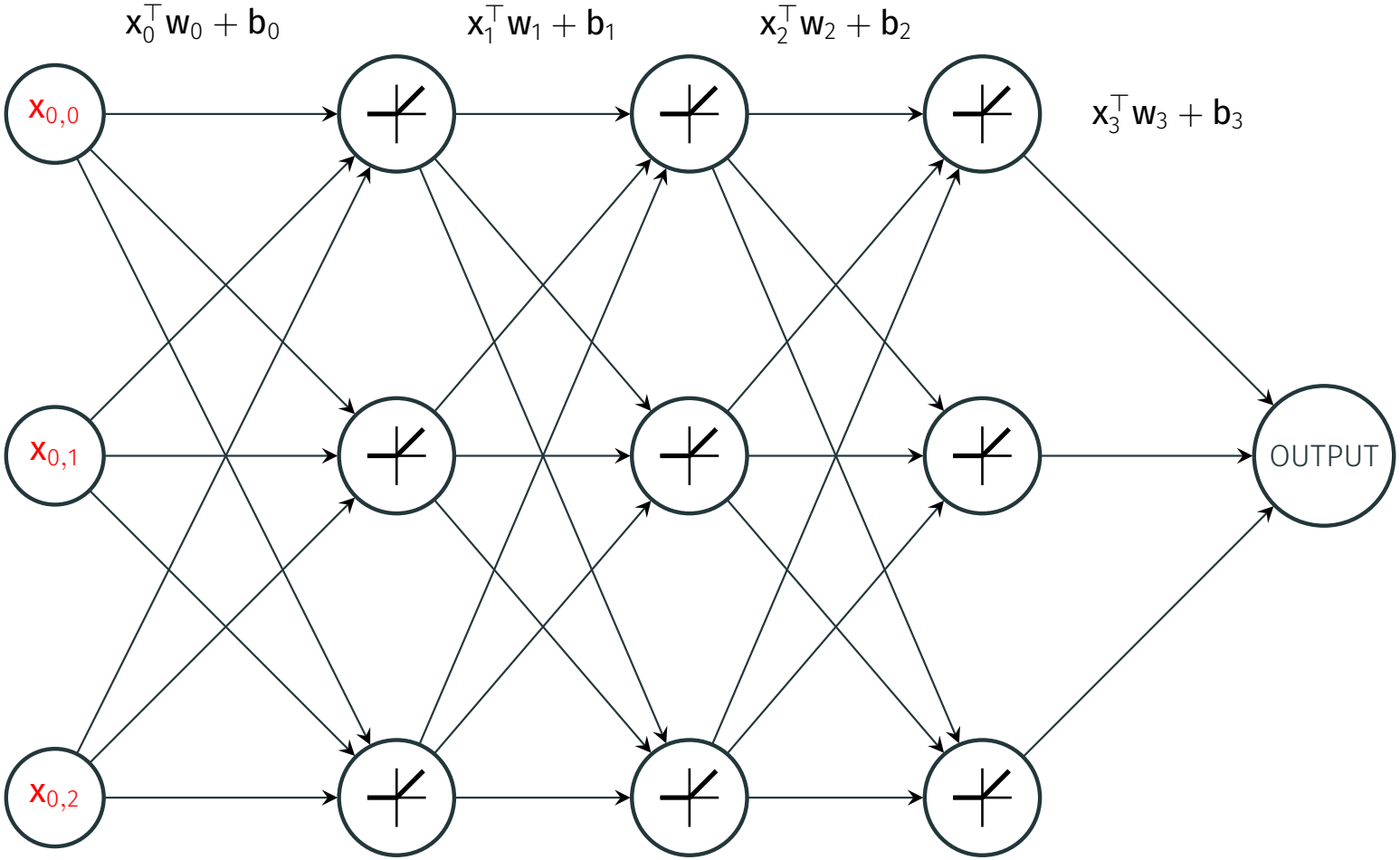
Backward Pass



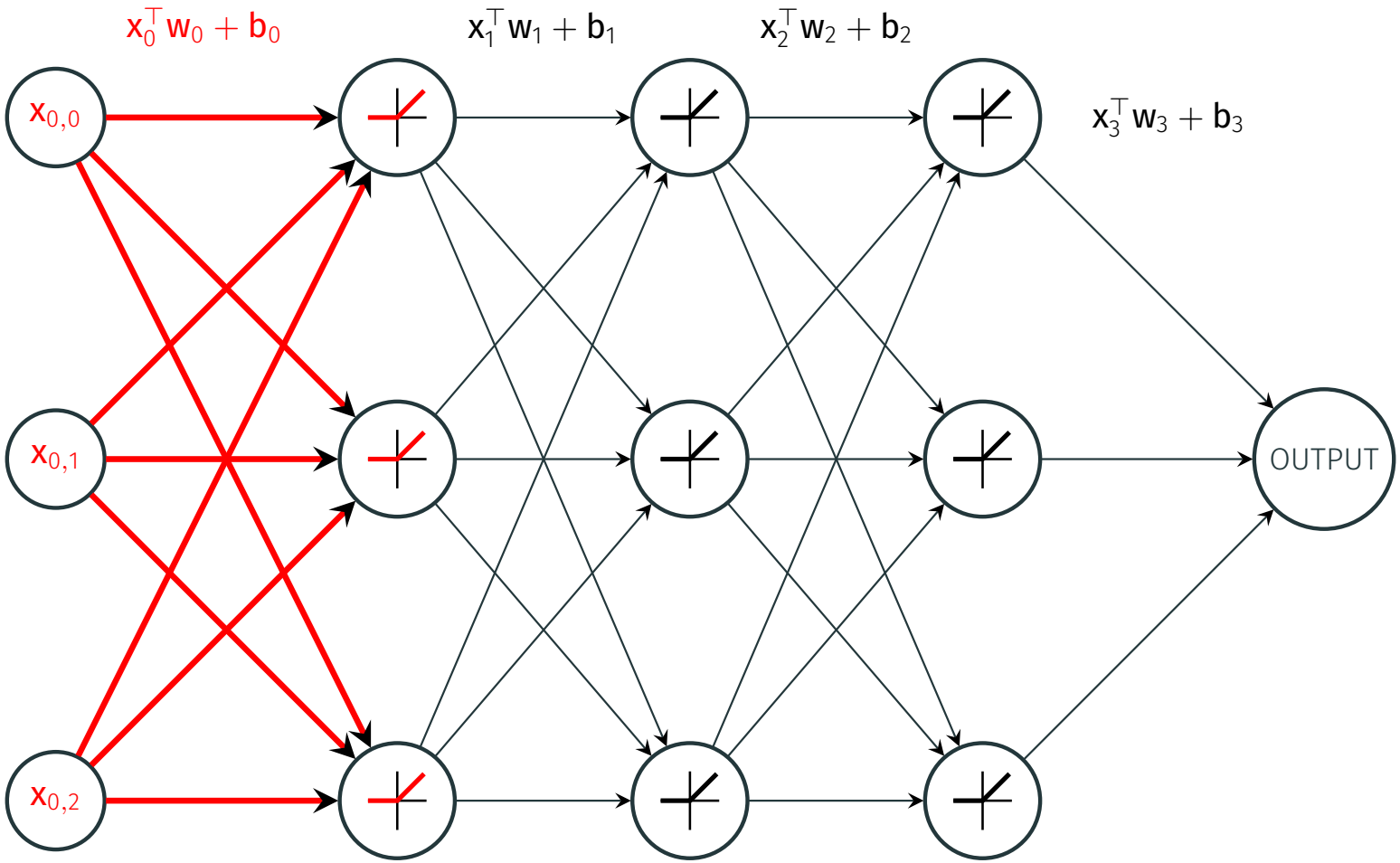
Frequentist Forward Pass



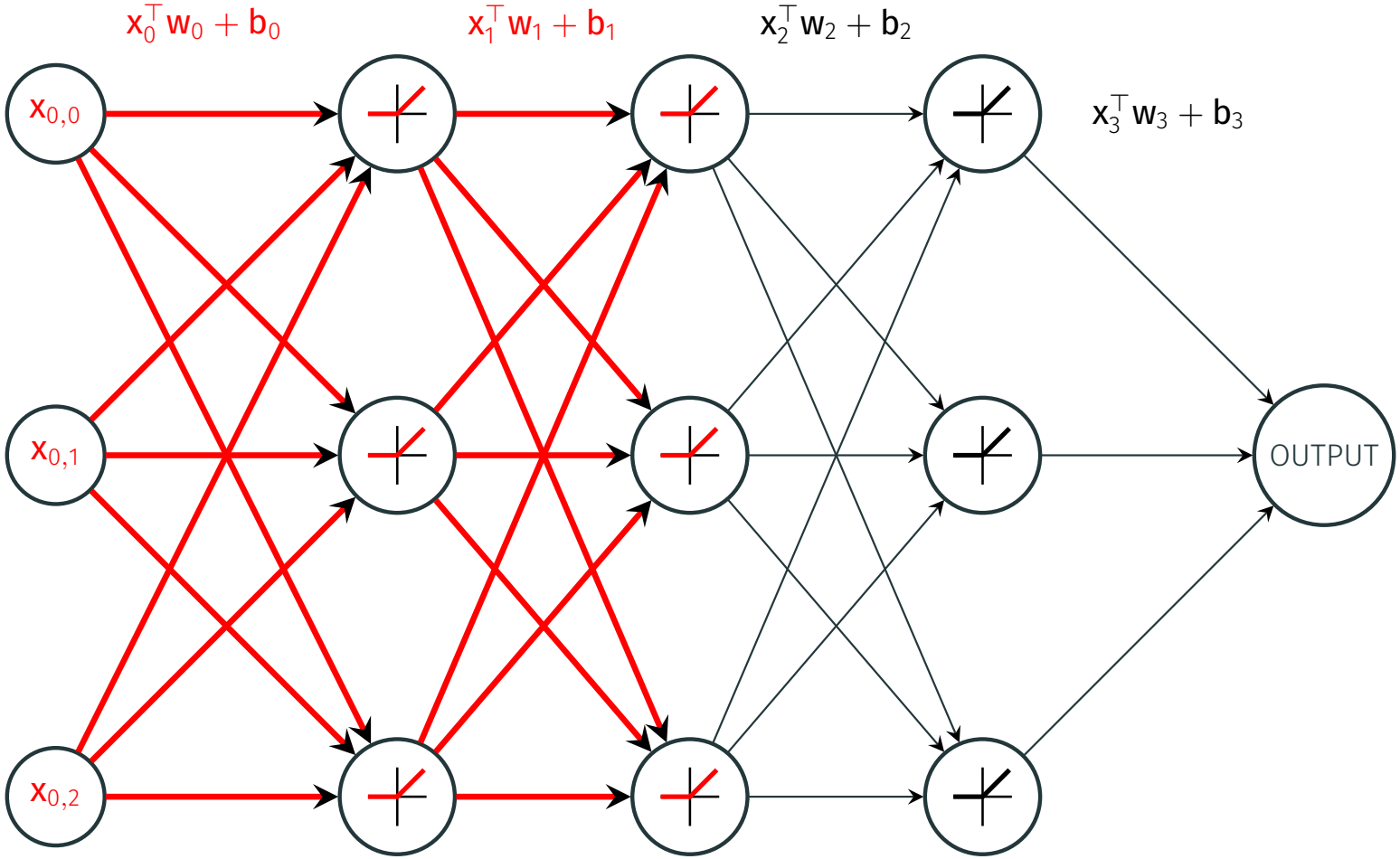
Frequentist Forward Pass



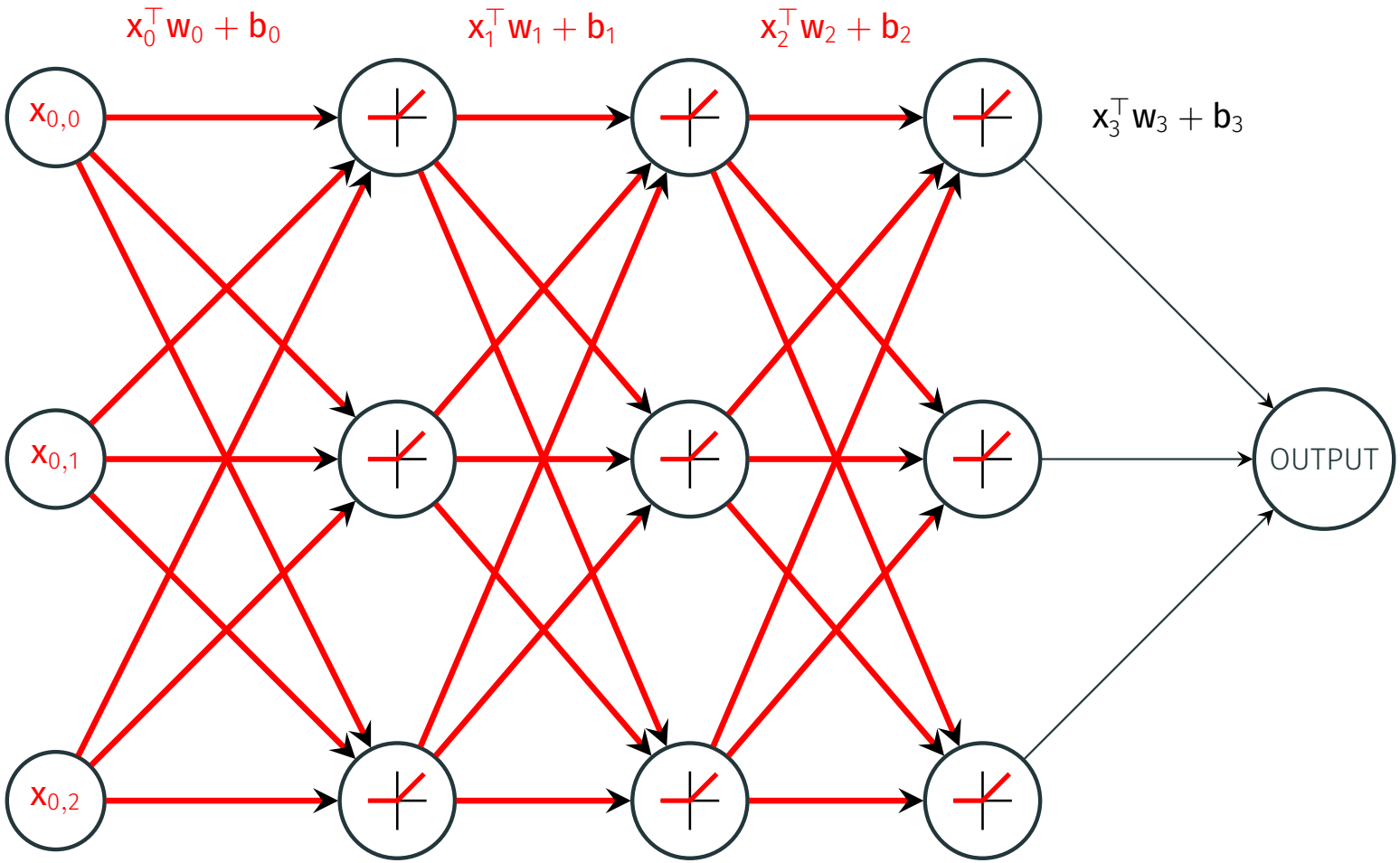
Frequentist Forward Pass



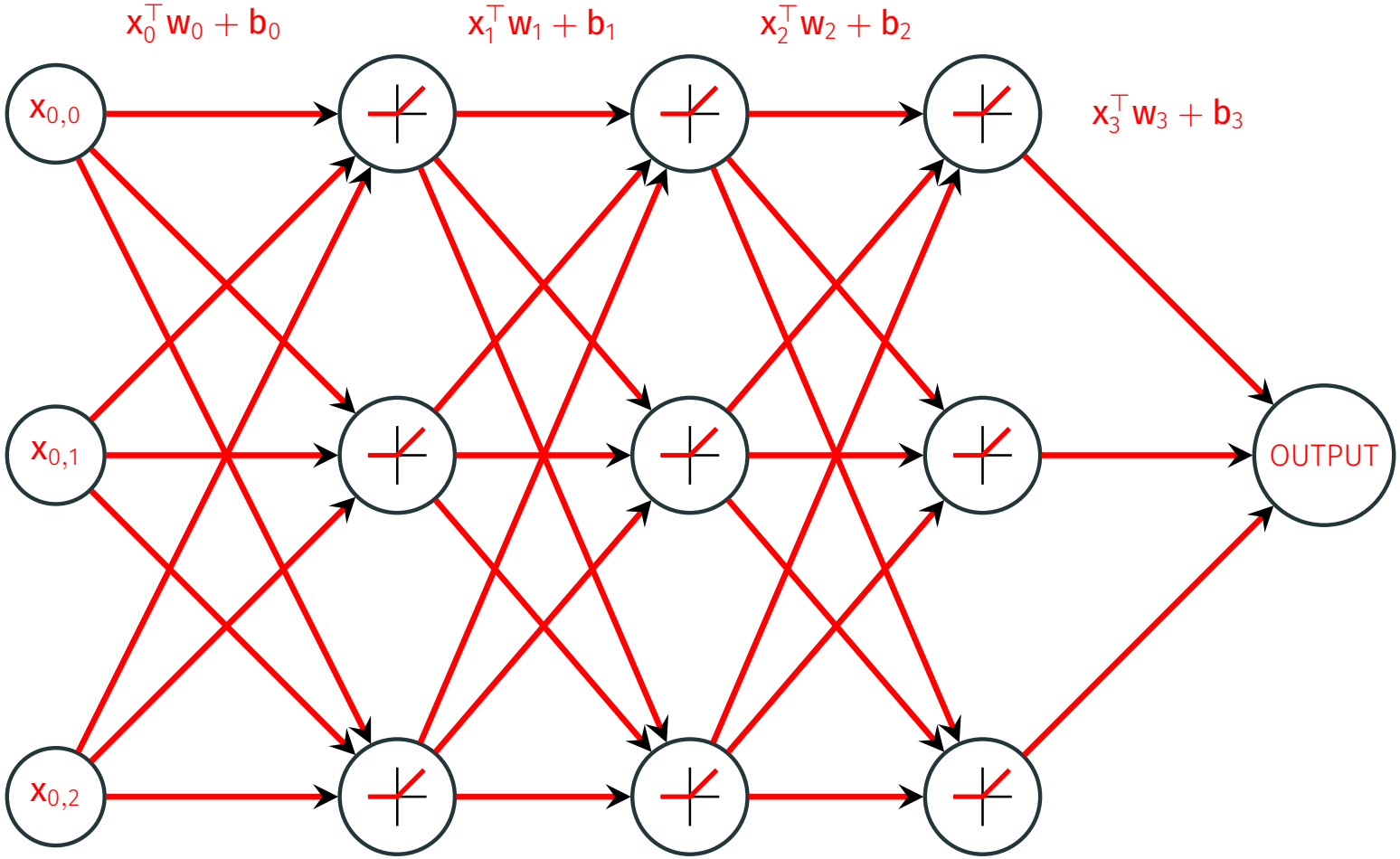
Frequentist Forward Pass



Frequentist Forward Pass

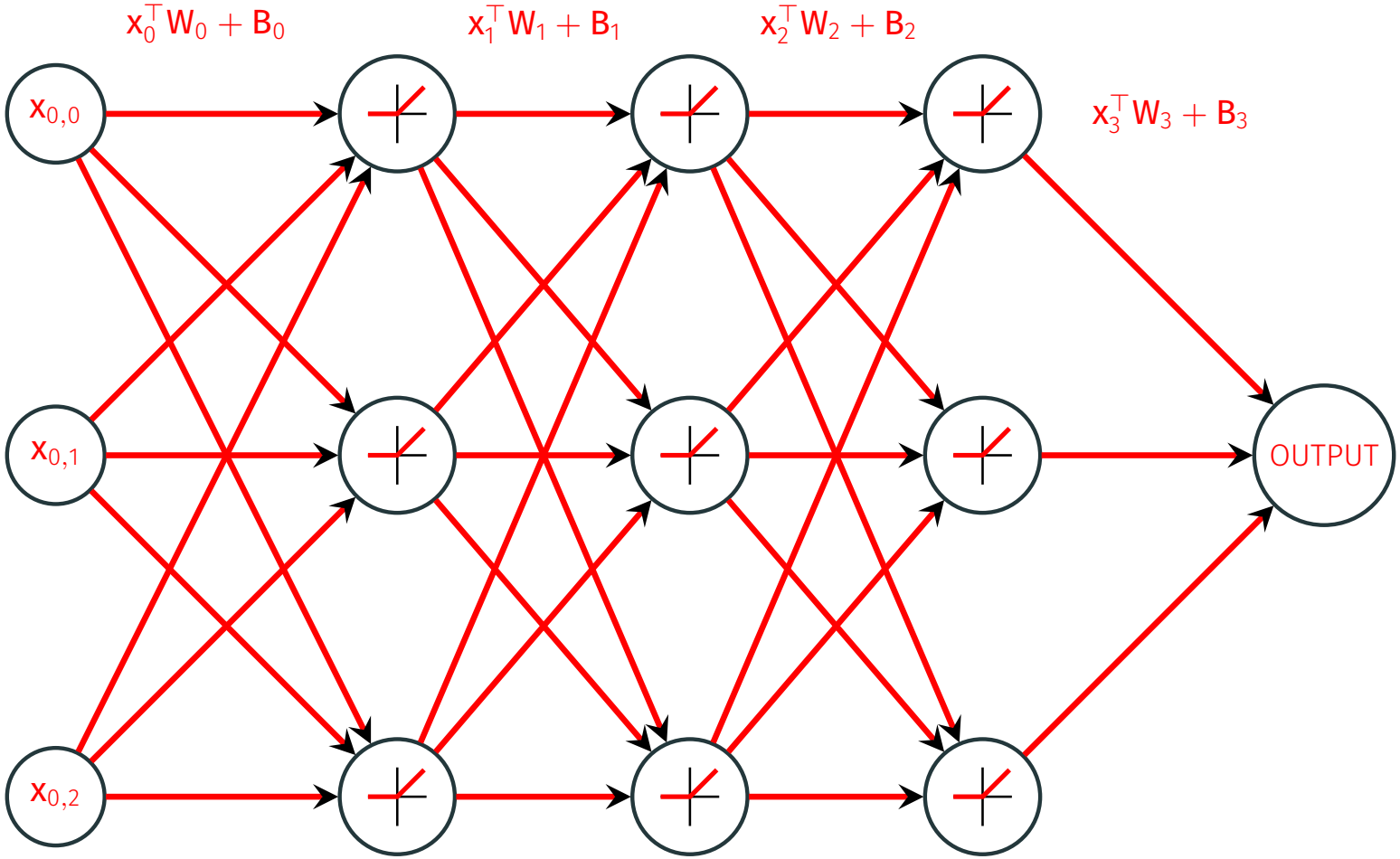


Frequentist Forward Pass

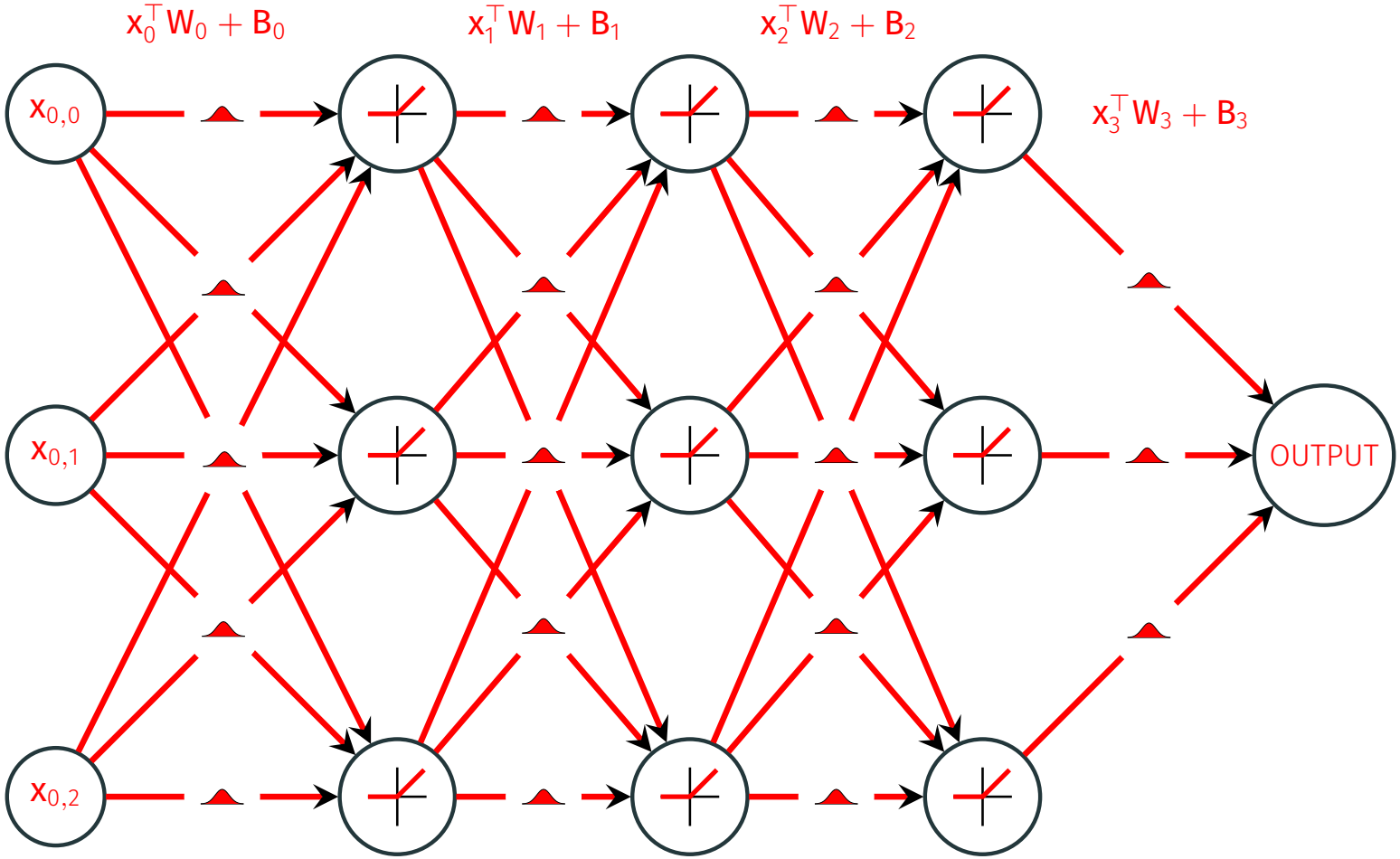


Bayesian Neural Networks

Bayesian Forward Pass

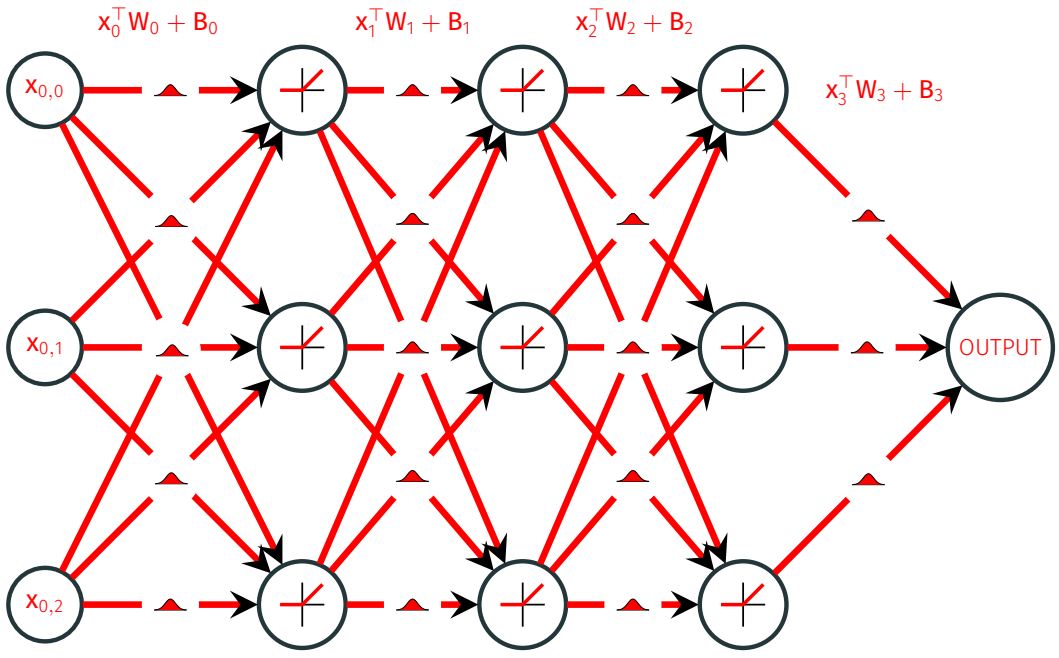


Bayesian Forward Pass

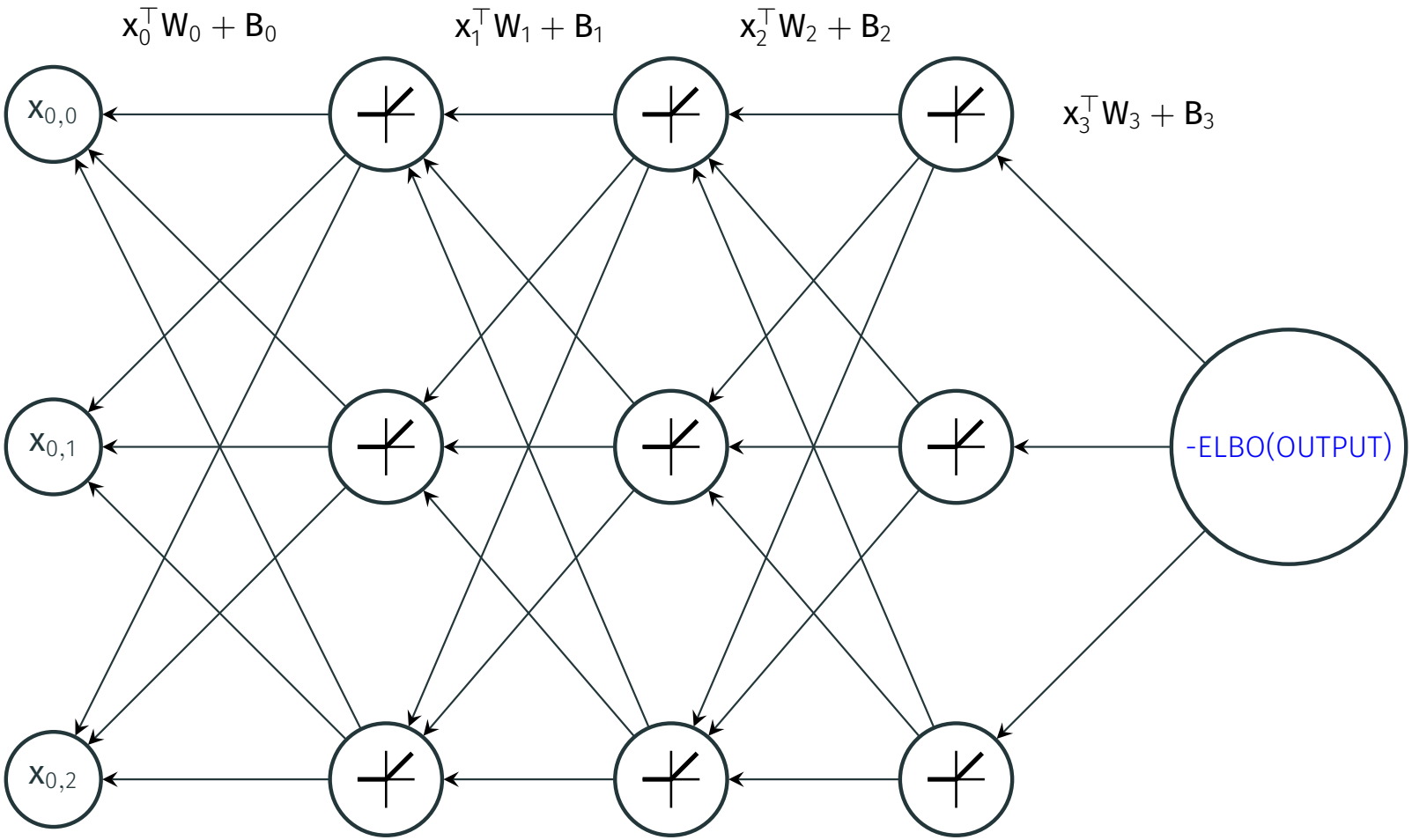


Bayesian Neural networks

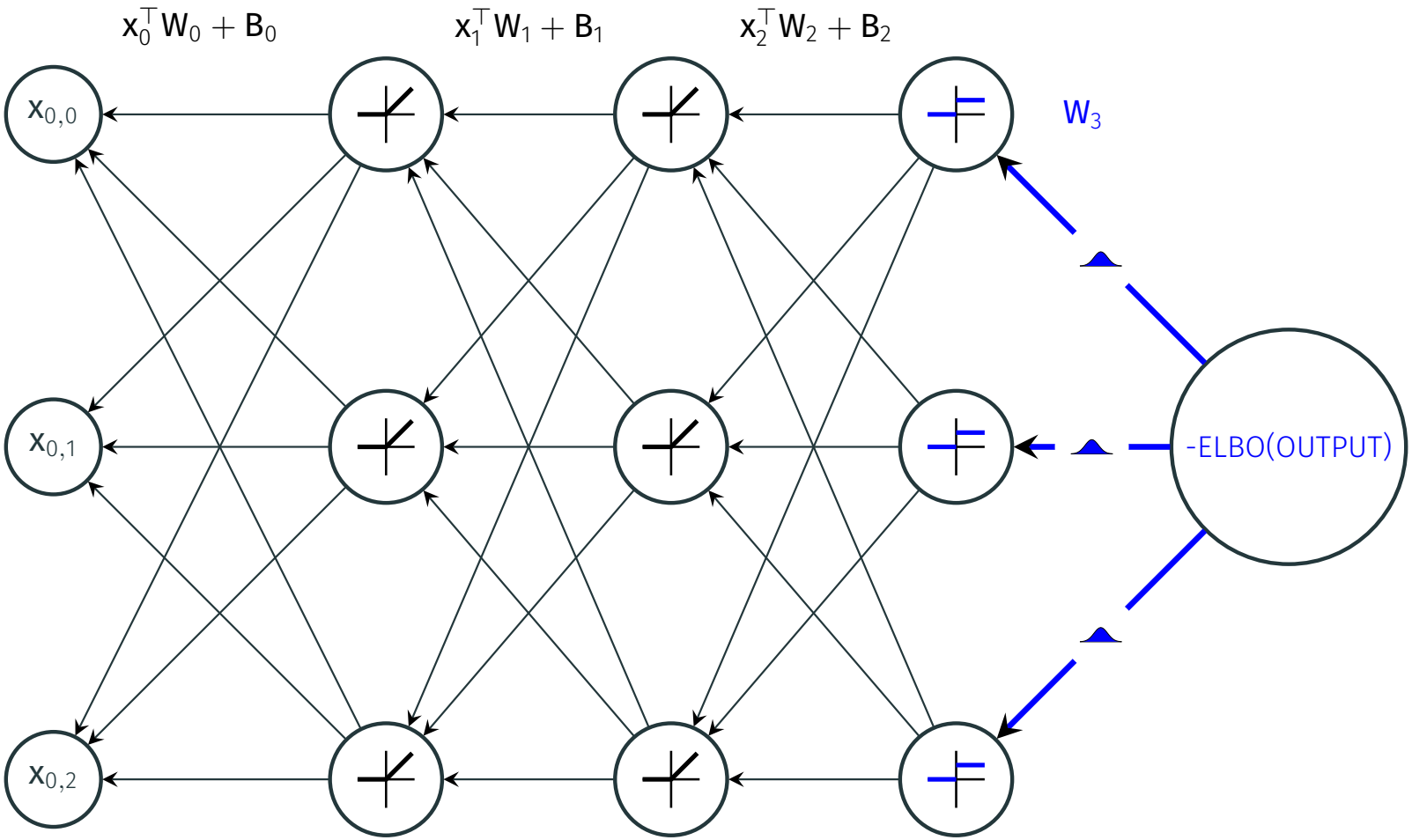
- Better Uncertainty Estimates.
- Better Robustness.
- Interpretability?
- Conventionally use Gaussian Prior.



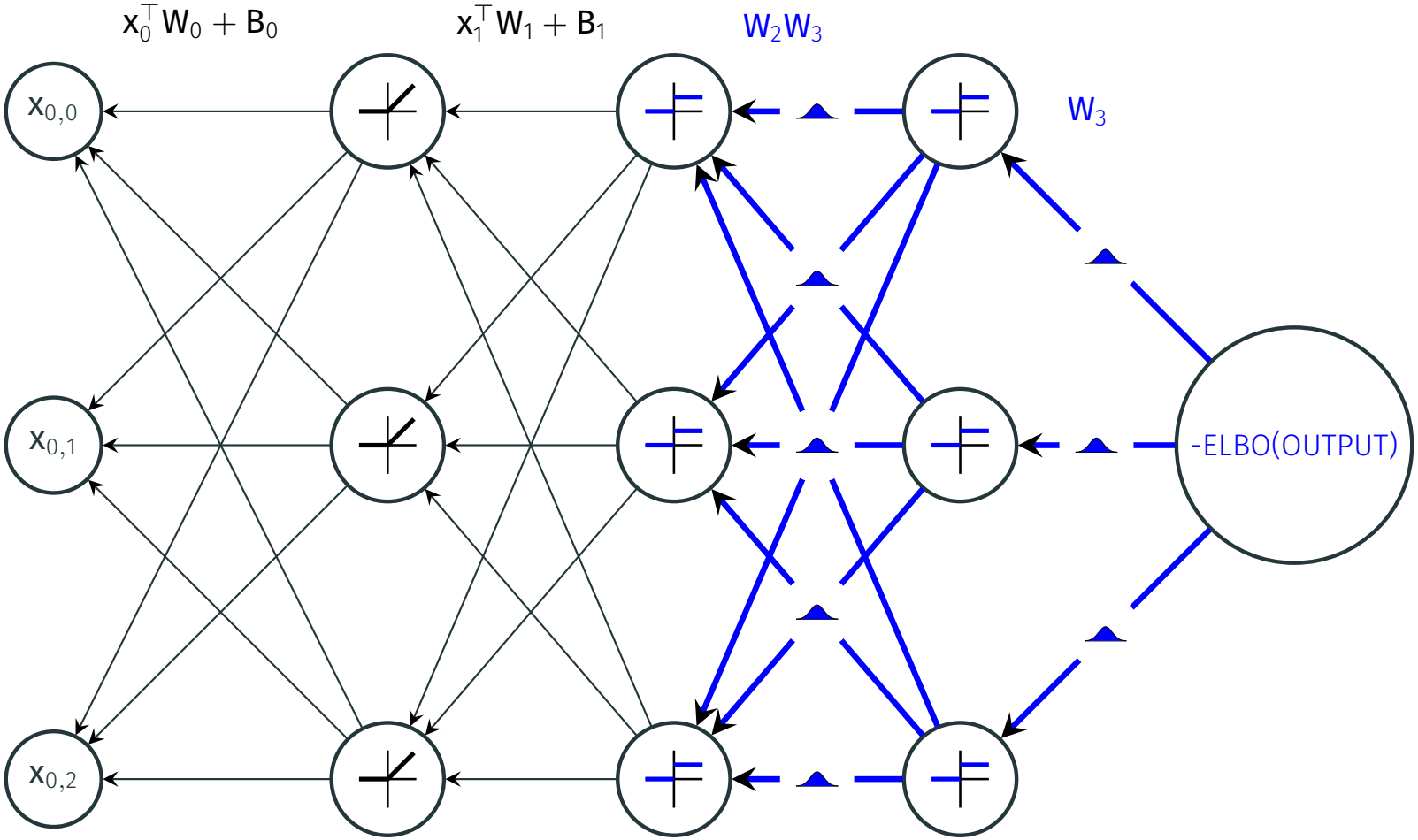
Bayesian Backward Pass



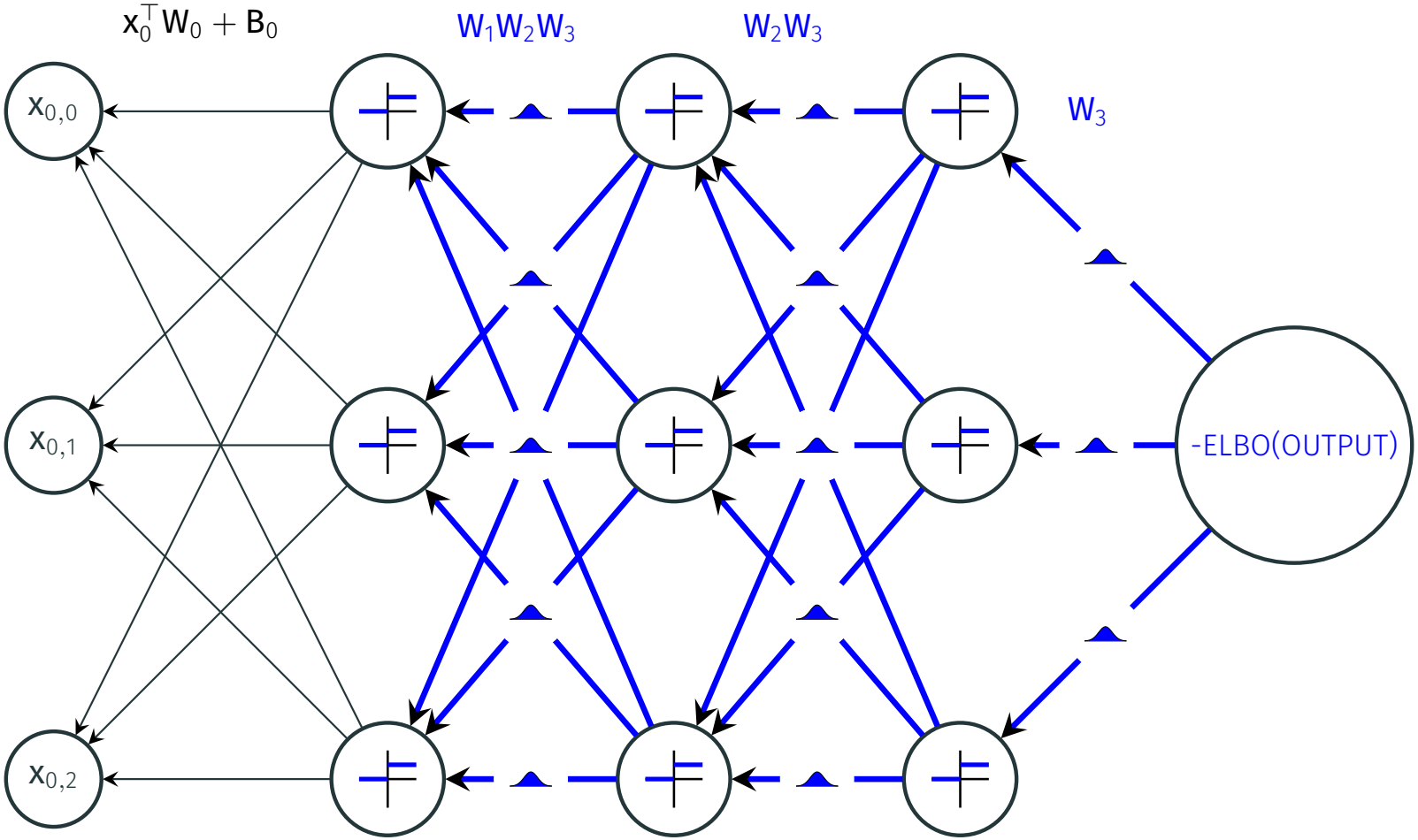
Bayesian Backward Pass



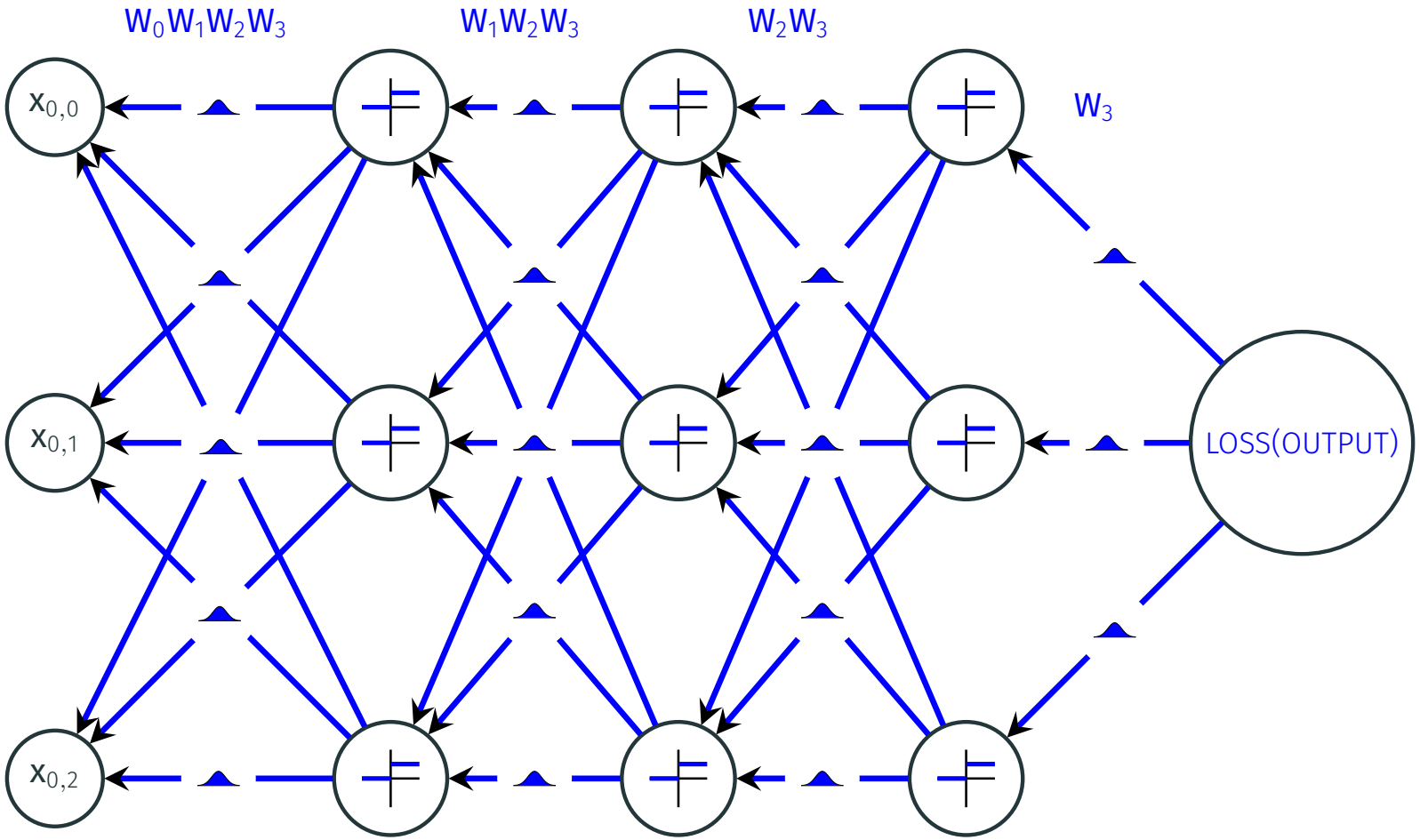
Bayesian Backward Pass



Bayesian Backward Pass



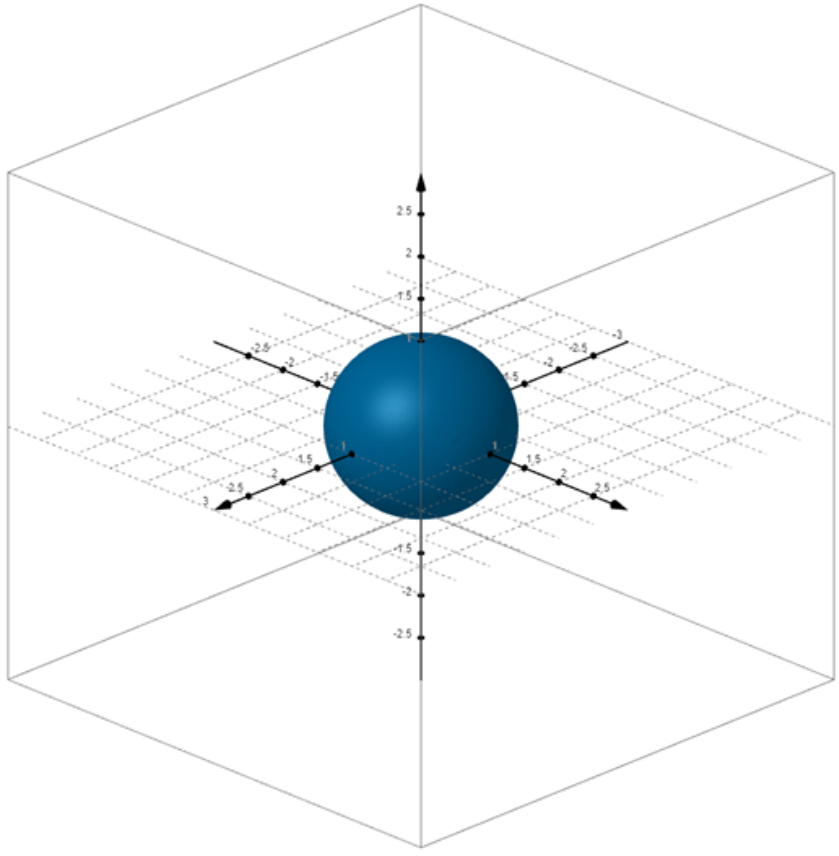
Bayesian Backward Pass



Hyperspherical Uniform Prior

Hyperspherical Uniform Prior

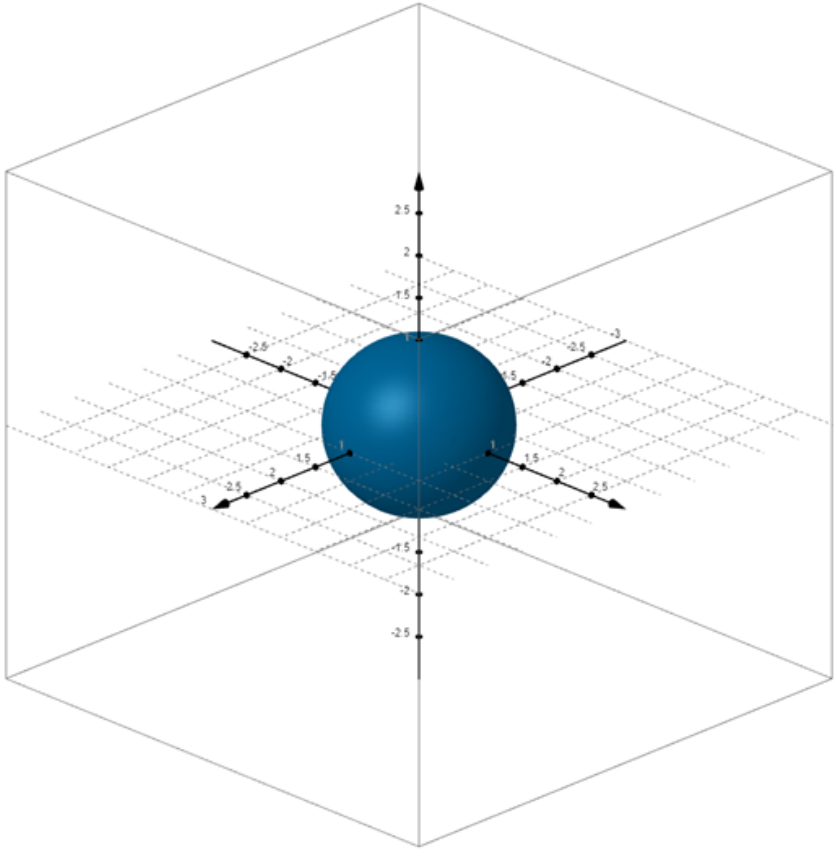
$$p_{HU_M}(\mathbf{w}) = \begin{cases} \frac{2(\pi^{m/2})}{\Gamma(m/2)} & ||\mathbf{w}|| = 1 \\ 0 & ||\mathbf{w}|| \neq 1 \end{cases}$$



Hyperspherical Uniform Prior

$$p_{HU_M}(\mathbf{w}) = \begin{cases} \frac{2(\pi^{m/2})}{\Gamma(m/2)} & ||\mathbf{w}|| = 1 \\ 0 & ||\mathbf{w}|| \neq 1 \end{cases}$$

- Supplants Batch Normalization.
- Keeps Parameters Closer in Scale.
- More Formally Bayesian



Finding the Posterior is Hard

- Analytically Intractable.
- MCMC Computationally Intractable.
-
-

$$p_{\Psi|\mathcal{D}}(\Psi, \mathcal{D}) = \frac{p_{\Psi}(\Psi) p_{\mathcal{D}|\Psi}(\mathcal{D}, \Psi)}{p_{\mathcal{D}}(\mathcal{D})}$$

Finding the Posterior is Hard

- Analytically Intractable.
- MCMC Computationally Intractable.
- Variational Inference.
- Posit a variational distribution.

$$p_{\Psi|\mathcal{D}}(\Psi, \mathcal{D}) = \frac{p_{\Psi}(\Psi) p_{\mathcal{D}|\Psi}(\mathcal{D}, \Psi)}{p_{\mathcal{D}}(\mathcal{D})}$$

Approximate with Variational Inference

$$KL(q_{\Psi}(\psi) \,||\, p_{\Psi|\mathcal{D}}(\psi, \mathcal{D})) \stackrel{def}{=} \int q_{\Psi}(\psi) \log \left(\frac{q_{\Psi}(\psi)}{p_{\Psi|\mathcal{D}}(\psi, \mathcal{D})} \right) d\psi$$

Approximate with Variational Inference

$$KL(q_\Psi(\psi) \,||\, p_{\Psi|\mathcal{D}}(\psi, \mathcal{D})) \stackrel{def}{=} \int q_\Psi(\psi) \log\left(\frac{q_\Psi(\psi)}{p_{\Psi|\mathcal{D}}(\psi, \mathcal{D})}\right) d\psi$$

$$= \int q_\Psi(\psi) \log\left(\frac{q_\Psi(\psi)}{p_\Psi(\psi)}\right) d\psi - \int q_\Psi(\psi) \log\left(p_{\mathcal{D}|\Psi}(\psi, \mathcal{D})\right) d\psi$$

The Loss is the Negative ELBO

$$= \int q_{\Psi}(\psi) \log \left(\frac{q_{\Psi}(\psi)}{p_{\Psi}(\psi)} \right) \, d\psi - \int q_{\Psi}(\psi) \log \left(p_{\mathcal{D}|\Psi}(\psi, \mathcal{D}) \right) \, d\psi$$

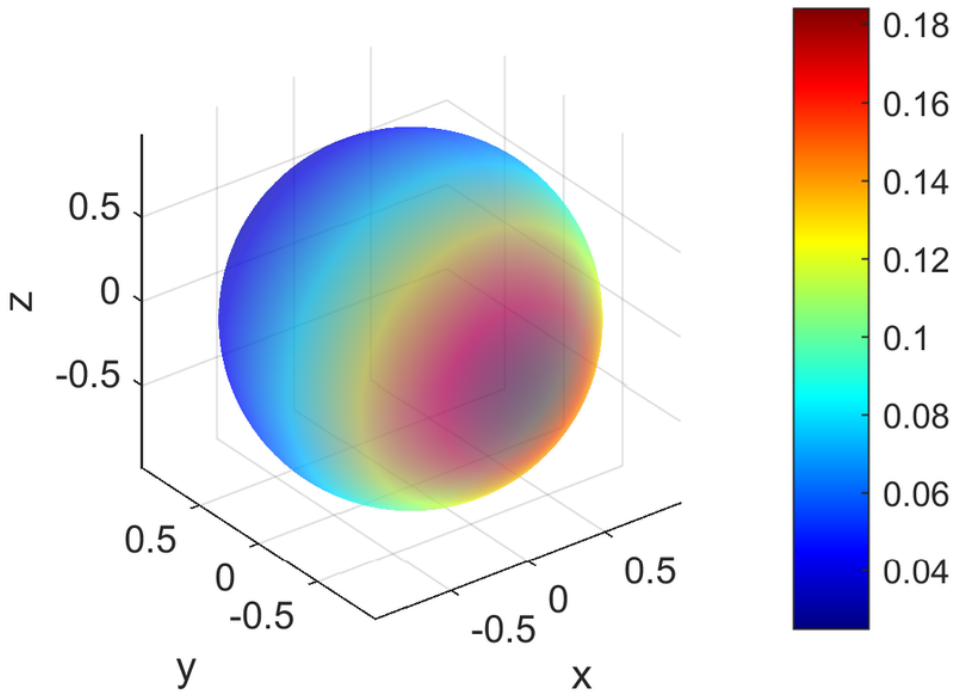
Selection of Variational Distribution

- Follows Prior Restriction.
- Reparametrizable.
- Arbitrary dimensionality.

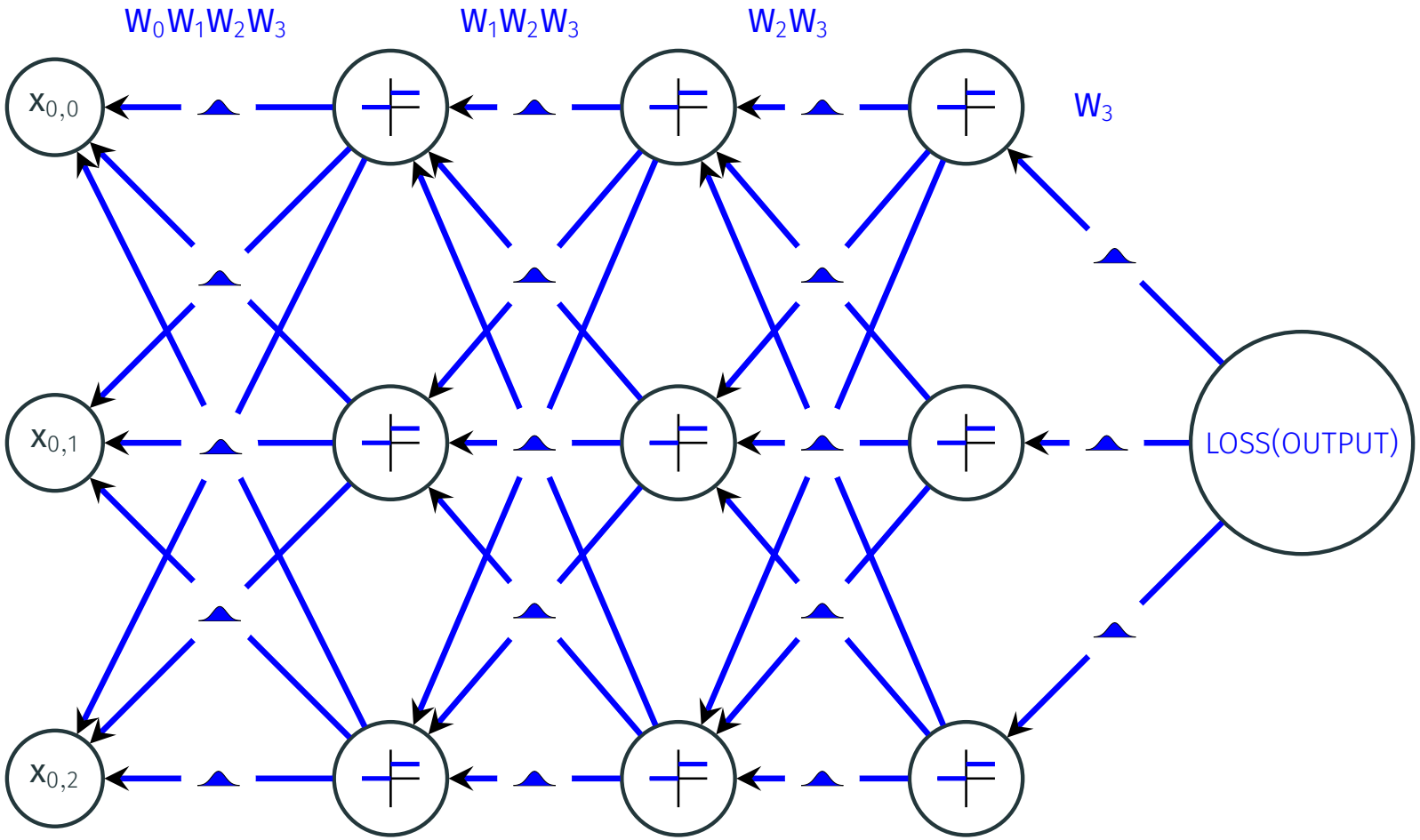
We Select the von Mises-Fisher Variational Distribution

$$p_{vMF_m}(\mathbf{w}; \boldsymbol{\mu}, \kappa) = C_m(\kappa) e^{(\kappa \boldsymbol{\mu}^\top \mathbf{w})}$$

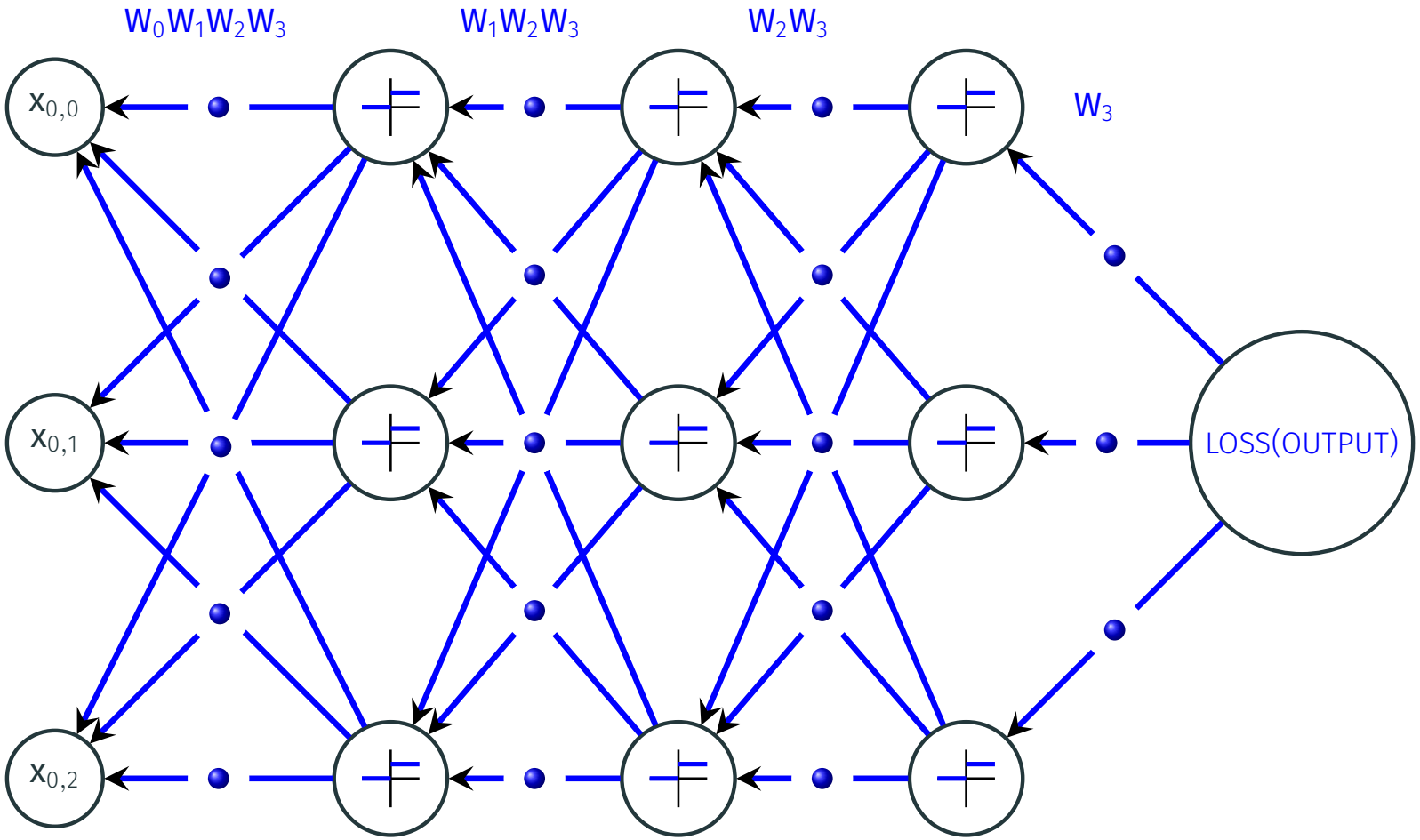
$$C_m(\kappa) = \frac{\kappa^{m/2-1}}{(2\pi)^{m/2} I_{m/2-1}(\kappa)}$$



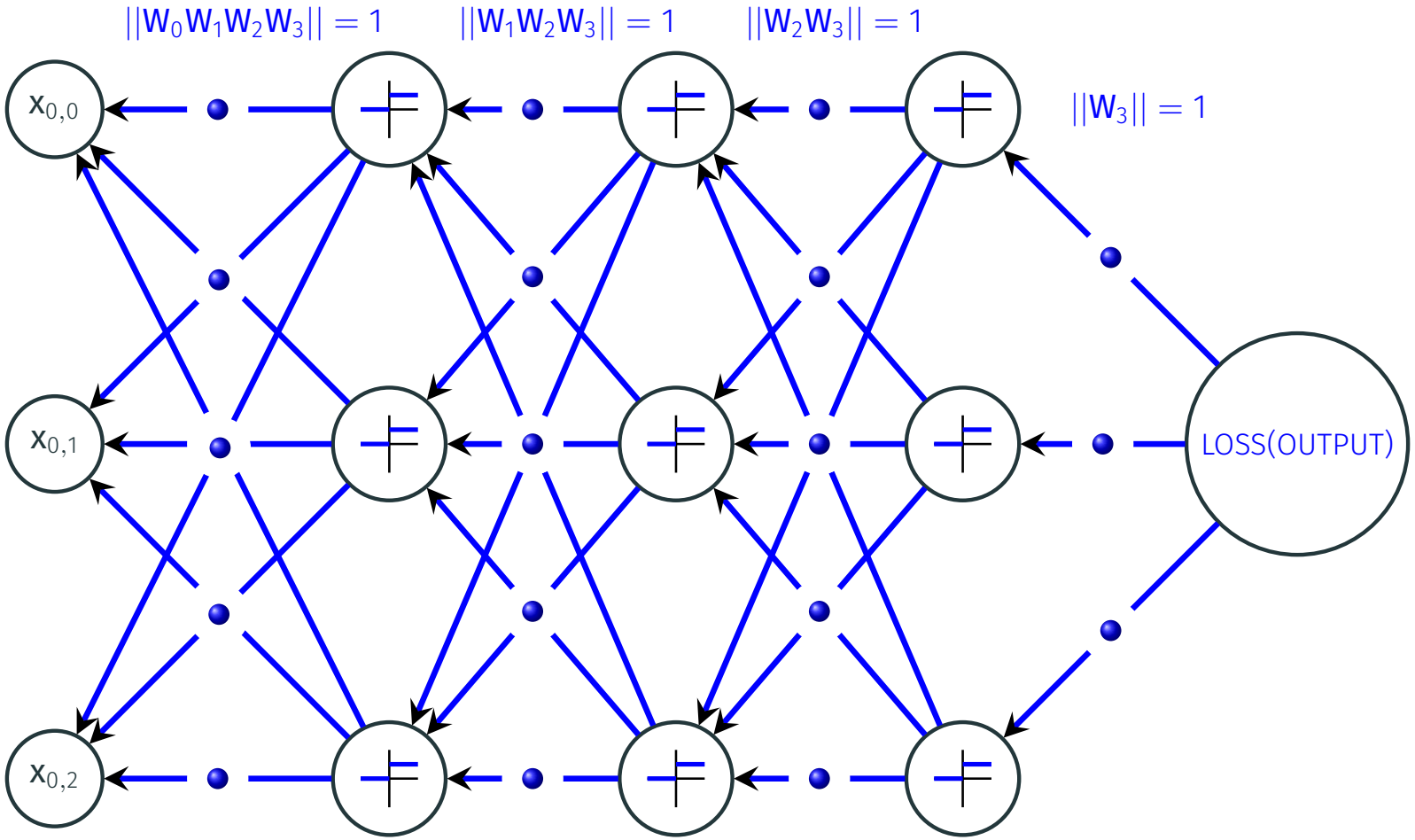
Norm one Backward Pass



Norm one Backward Pass



Norm one Backward Pass

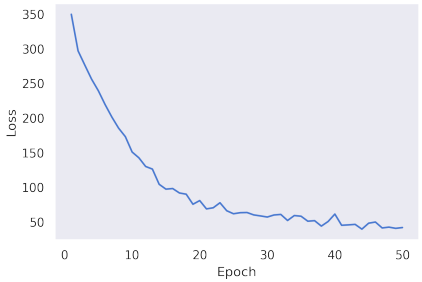


Results

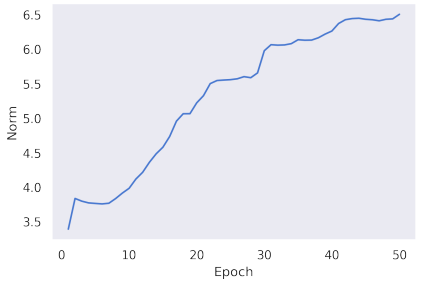
Hyperspherical Uniform compared to the state-of-the-art

Phoneme data

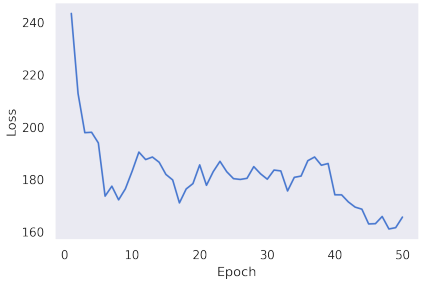
Gaussian Loss Curve



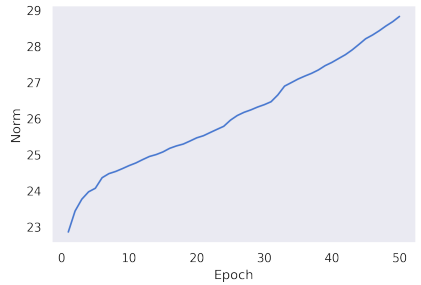
Gaussian Norm Curve



vMF Loss Curve

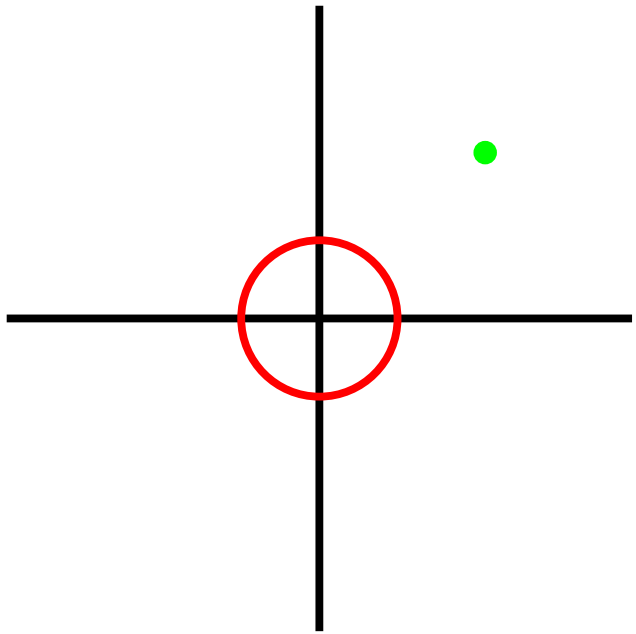


vMF Ghost Curve



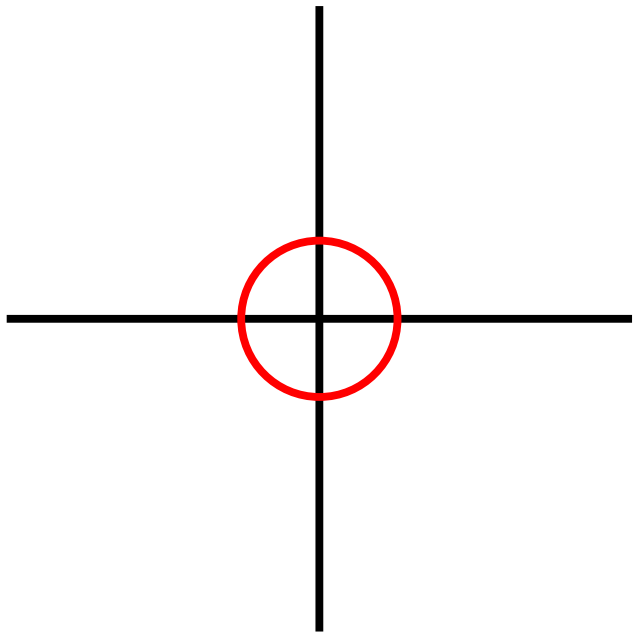
Restricted Optimization with PyTorch

- Parameters in PyTorch live on the real line.
- We have to work-around by projecting the parameters.
- This probably induces errors



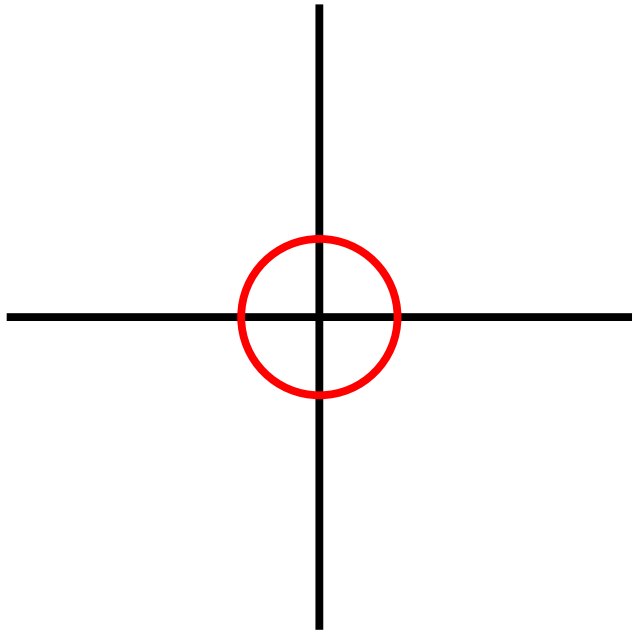
Restricted Optimization with PyTorch

- Parameters in PyTorch live on the real line.
- We have to work-around by projecting the parameters.
- This probably induces errors



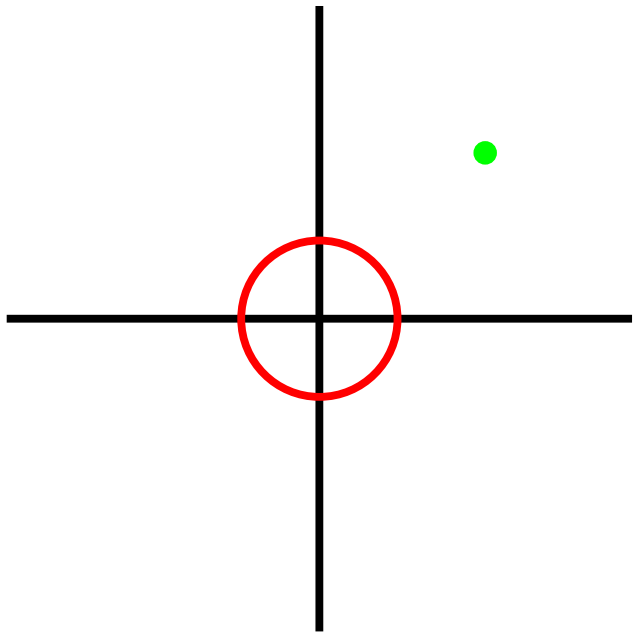
Restricted Optimization with PyTorch

- Parameters in PyTorch live on the real line.
- We have to work-around by projecting the parameters.
- This probably induces errors



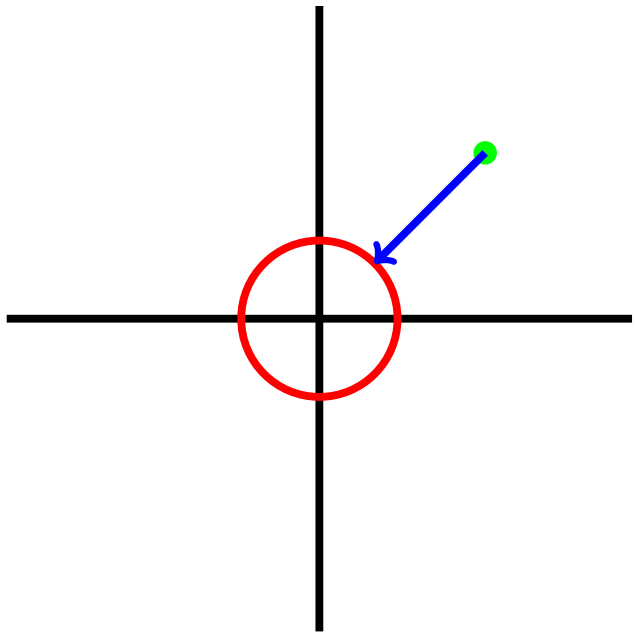
Normalized Initialization Inheritance Redresses the Problem

- Projects Registered Mu.
- Seems to work well in practice.



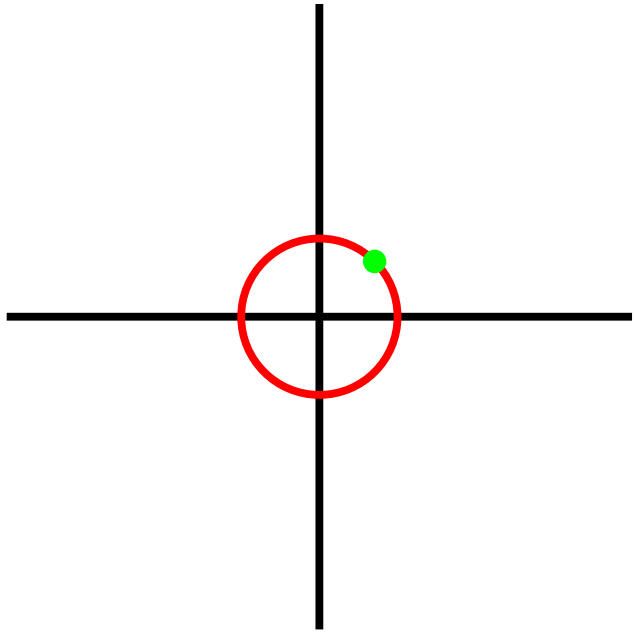
Normalized Initialization Inheritance Redresses the Problem

- Projects Registered Mu.
- Seems to work well in practice.



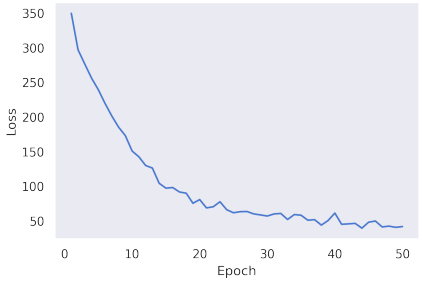
Normalized Initialization Inheritance Redresses the Problem

- Projects Registered Mu.
- Seems to work well in practice.

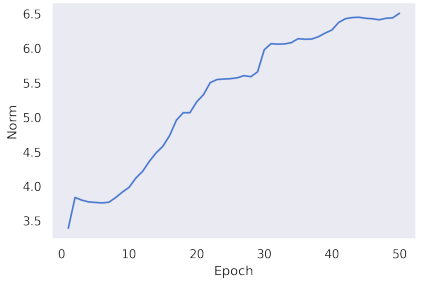


Phoneme data

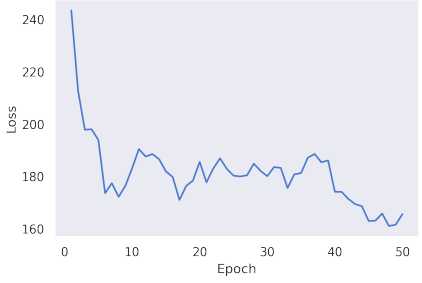
Gaussian Loss Curve



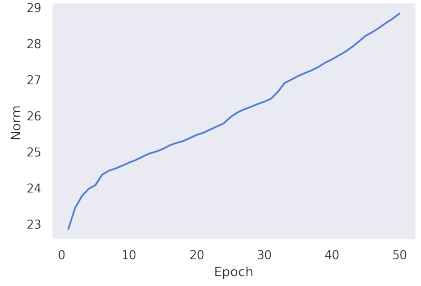
Gaussian Norm Curve



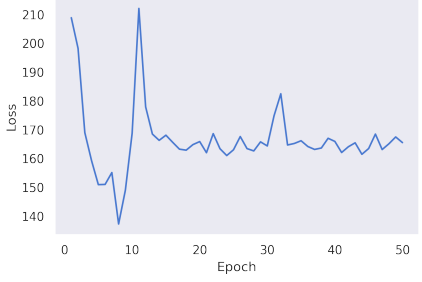
vMF Loss Curve



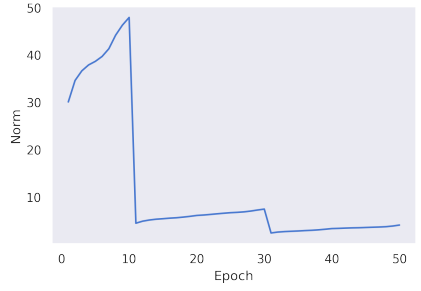
vMF Ghost Curve



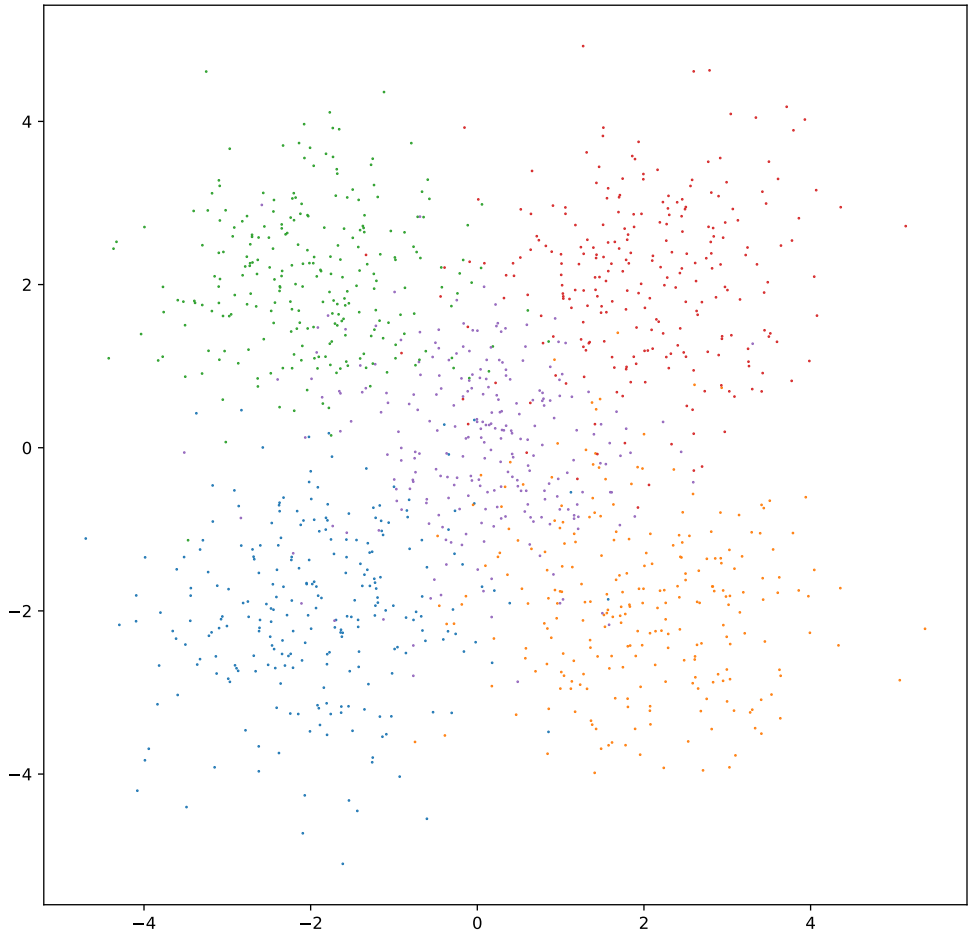
NIIvMF Loss Curve



NIIvMF Ghost Curve



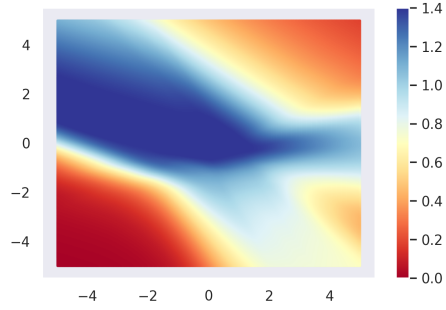
Simulated Data



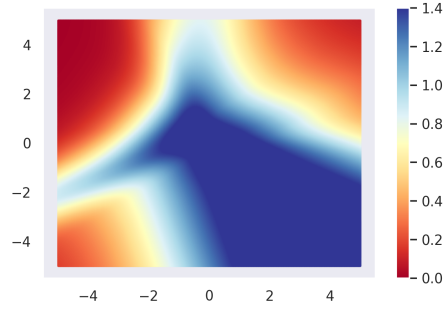
Simulated Data

Bivariate Gaussian Testset Accuracy			
N	Gaussian	vMF	NII vMF
50	137/200	129/200	138/200
150	484/600	497/600	499/600
250	862/1000	822/1000	853/1000
Hyperparameters			
Prior	$\mathcal{N}(0, e^{-0.1})$	HU	HU
Learning rate	0.04	0.07	0.14
Hidden Layers	(5, 5)	(5, 5)	(5, 5)

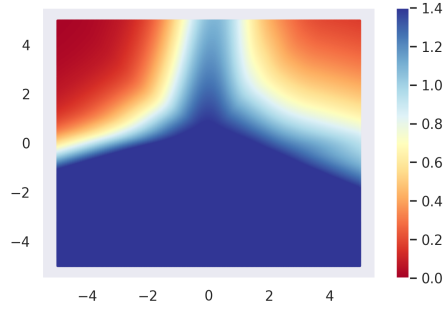
Gaussian Cross-Entropy N=50



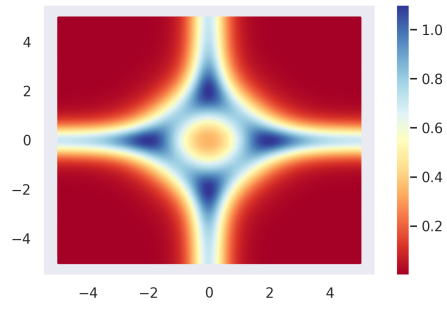
NII vMF Cross-Entropy N=50



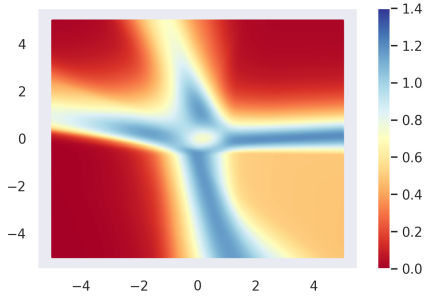
vMF Cross-Entropy N=50



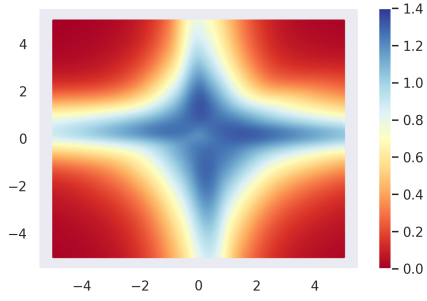
True Cross-Entropy



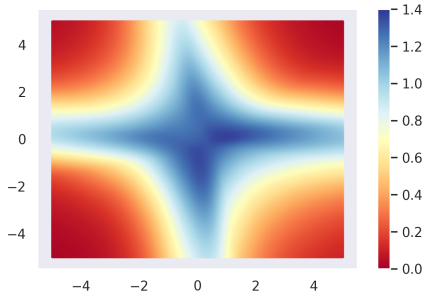
Gaussian Cross-Entropy N=150



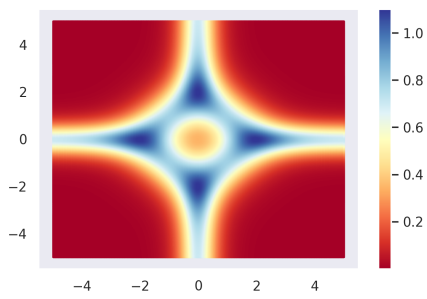
NII vMF Cross-Entropy N=150



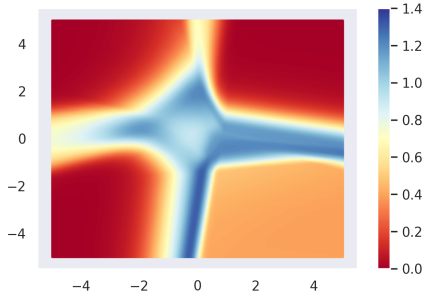
vMF Cross-Entropy N=150



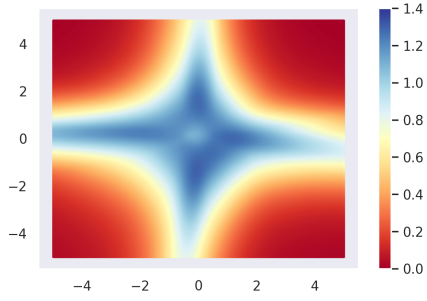
True Cross-Entropy



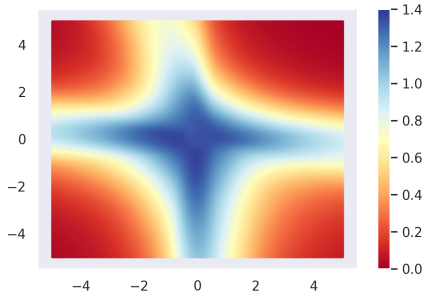
Gaussian Cross-Entropy N=250



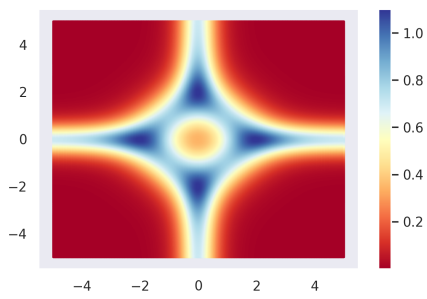
NII vMF Cross-Entropy N=250



vMF Cross-Entropy N=250



True Cross-Entropy



Relevance for Future Work

Limitations

- Isotropic Covariance
- Computational Intensity
- PyTorch (”Ghost Mu”)

Limitations

- Isotropic Covariance $\rightarrow$ PGN
- Computational Intensity
- PyTorch (”Ghost Mu”)

Limitations

- Isotropic Covariance → PGN
- Computational Intensity → PGN
- PyTorch (”Ghost Mu”)

Limitations

- Isotropic Covariance → PGN
- Computational Intensity → PGN
- PyTorch (”Ghost Mu”) → NII

Projected Gaussian Normal (PGN)

Reparametrize Ψ by $\Psi = (\cos\Theta, \sin\Theta)^T$:

$$p(\theta \mid \boldsymbol{\mu}, \Sigma) = \left(\frac{1}{2\pi A(\theta)}\right) |\Sigma|^{-\frac{1}{2}} \exp(C) \left\{ 1 + \frac{B(\theta)}{\sqrt{A(\theta)}} \frac{\Phi\left(\frac{B(\theta)}{\sqrt{A(\theta)}}\right)}{\varphi\left(\frac{B(\theta)}{\sqrt{A(\theta)}}\right)} \right\} I_{[0,2\pi)}(\theta)$$

$\boldsymbol{u}^T = (\cos\theta, \sin\theta)$

$A(\theta) = \boldsymbol{u}^T \Sigma^{-1} \boldsymbol{u}$

$B(\theta) = \boldsymbol{u}^T \Sigma^{-1} \boldsymbol{\mu}$

$C = -\frac{1}{2} \boldsymbol{\mu}^T \Sigma^{-1} \boldsymbol{\mu}$. $I_{(0,2\pi]}(\cdot)$ is an indicator function, and $\Phi(\cdot), \varphi(\cdot)$ are the standard normal distribution and density functions, respectively.

Conclusion

Conclusion

- Overall similar perfomance for ”small” networks.
- Interpretability.
- Superior Performance for Deep Networks?

Conclusion

- Overall similar perfomance for ”small” networks.
- Interpretability.
- Superior Performance for Deep Networks? $\rightarrow$ Batch Normalization is more important for Deeper networks.

References

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off, jul 2019. URL *https://doi.org/10.1073/pnas.1903070116*.

Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *SIAM Journal on Mathematics of Data Science*, 2 (4):1167–1180, 2020. doi: 10.1137/20M1336072. URL *https://doi.org/10.1137/20M1336072*.

References ii

Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75(4): 667–766, June 2021. doi: 10.1002/cpa.22008. URL *https://doi.org/10.1002/cpa.22008*.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt, 2019. URL *https://doi.org/10.48550/arXiv.1912.02292*.

Terrence J. Sejnowski. The unreasonable effectiveness of deep learning in artificial intelligence. *Proceedings of the National Academy of Sciences*, 117(48):30033–30038, 2020. doi: 10.1073/pnas.1907373117. URL *https://doi.org/10.1073/pnas.1907373117*.

References iii

Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. Compute trends across three eras of machine learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, July 2022. doi: 10.1109/ijcnn55064.2022.9891914. URL *https://doi.org/10.1109/ijcnn55064.2022.9891914*.