Title: Reparameterization trick

URL: https://en.wikipedia.org/wiki/Reparameterization_trick

PageID: 77939986

Categories: Category:Machine learning, Category:Stochastic optimization

-----

The reparameterization trick (aka "reparameterization gradient estimator") is a technique used in statistical machine learning , particularly in variational inference , variational autoencoders , and stochastic optimization . It allows for the efficient computation of gradients through random variables, enabling the optimization of parametric probability models using stochastic gradient descent , and the variance reduction of estimators .

It was developed in the 1980s in operations research , under the name of "pathwise gradients", or "stochastic gradients". [1] [2] Its use in variational inference was proposed in 2013. [3]

Mathematics

Let z {\displaystyle z} be a random variable with distribution q φ (z) {\displaystyle q_{\phi }(z)} , where φ {\displaystyle \phi } is a vector containing the parameters of the distribution.

REINFORCE estimator

Consider an objective function of the form: L (φ) = E z ~ q φ (z) [f (z)] {\displaystyle L(\phi)=\mathbb {E} _{z\sim q_{\phi }(z)}[f(z)]} Without the reparameterization trick, estimating the gradient ∇ φ L (φ) {\displaystyle \nabla _{\phi }L(\phi)} can be challenging, because the parameter appears in the random variable itself. In more detail, we have to statistically estimate: ∇ φ L (φ) = ∇ φ ∫ d z q φ (z) f (z) {\displaystyle \nabla _{\phi }L(\phi)=\nabla _{\phi }\int dz\;q_{\phi }(z)f(z)} The REINFORCE estimator, widely used in reinforcement learning and especially policy gradient , [4] uses the following equality: ∇ φ L (φ) = ∫ d z q φ (z) ∇ φ (ln ■ q φ (z)) f (z) = E z ~ q φ (z) [∇ φ (ln ■ q φ (z)) f (z)] {\displaystyle \nabla _{\phi }L(\phi)=\int dz\;q_{\phi }(z)\nabla _{\phi }(\ln q_{\phi }(z))f(z)=\mathbb {E} _{z\sim q_{\phi }(z)}[\nabla _{\phi }(\ln q_{\phi }(z))f(z)]} This allows the gradient to be estimated: ∇ φ L (φ) ≈ 1 N ∑ i = 1 N ∇ φ (ln ■ q φ (z i)) f (z i) {\displaystyle \nabla _{\phi }L(\phi)\approx {\frac {1}{N}}\sum _{i=1}^{N}\nabla _{\phi }(\ln q_{\phi }(z_{i}))f(z_{i})} The REINFORCE estimator has high variance, and many methods were developed to reduce its variance . [5]

Reparameterization estimator

The reparameterization trick expresses z {\displaystyle z} as: z = g φ (■) , ■ ~ p (■) {\displaystyle z=g_{\phi }(\epsilon),\quad \epsilon \sim p(\epsilon)} Here, g φ {\displaystyle g_{\phi }} is a deterministic function parameterized by φ {\displaystyle \phi } , and ■ {\displaystyle \epsilon } is a noise variable drawn from a fixed distribution p (■) {\displaystyle p(\epsilon)} . This gives: L (φ) = E ■ ~ p (■) [f (g φ (■))] {\displaystyle L(\phi)=\mathbb {E} _{\epsilon \sim p(\epsilon)}[f(g_{\phi }(\epsilon))]} Now, the gradient can be estimated as: ∇ φ L (φ) = E ■ ~ p (■) [∇ φ f (g φ (■))] ≈ 1 N ∑ i = 1 N ∇ φ f (g φ (■ i)) {\displaystyle \nabla _{\phi }L(\phi)=\mathbb {E} _{\epsilon \sim p(\epsilon)}[\nabla _{\phi }f(g_{\phi }(\epsilon))]\approx {\frac {1}{N}}\sum _{i=1}^{N}\nabla _{\phi }f(g_{\phi }(\epsilon _{i}))}

Examples

For some common distributions, the reparameterization trick takes specific forms:

Normal distribution : For z ~ N (μ , σ 2) {\displaystyle z\sim {\mathcal {N}}(\mu ,\sigma ^{2})} , we can use: z = μ + σ ■ , ■ ~ N (0 , 1) {\displaystyle z=\mu +\sigma \epsilon ,\quad \epsilon \sim {\mathcal {N}}(0,1)}

Exponential distribution : For z ~ Exp (λ) {\displaystyle z\sim {\text{Exp}}(\lambda)} , we can use: z = − 1 λ log ■ (■) , ■ ~ Uniform (0 , 1) {\displaystyle z=-{\frac {1}{\lambda }}\log(\epsilon),\quad

\epsilon \sim {\text{Uniform}}(0,1)} Discrete distribution can be reparameterized by the Gumbel distribution (Gumbel-softmax trick or "concrete distribution"). [6]

In general, any distribution that is differentiable with respect to its parameters can be reparameterized by inverting the multivariable CDF function, then apply the implicit method. See [1] for an exposition and application to the Gamma Beta , Dirichlet , and von Mises distributions .

Applications

Variational autoencoder

In Variational Autoencoders (VAEs), the VAE objective function, known as the Evidence Lower Bound (ELBO), is given by:

ELBO (φ , θ) = E z ~ q φ (z | x) [log ■ p θ (x | z)] − D KL (q φ (z | x) | | p (z)) {\displaystyle {\text{ELBO}}(\phi ,\theta)=\mathbb {E} _{z\sim q_{\phi }(z|x)}[\log p_{\theta }(x|z)]-D_{\text{KL}}(q_{\phi }(z|x)||p(z))}

where q φ (z | x) {\displaystyle q_{\phi }(z|x)} is the encoder (recognition model), p θ (x | z) {\displaystyle p_{\theta }(x|z)} is the decoder (generative model), and p (z) {\displaystyle p(z)} is the prior distribution over latent variables. The gradient of ELBO with respect to θ {\displaystyle \theta } is simply E z ~ q φ (z | x) [∇ θ log ■ p θ (x | z)] ≈ 1 L ∑ l = 1 L ∇ θ log ■ p θ (x | z l) {\displaystyle \mathbb {E} _{z\sim q_{\phi }(z|x)}[\nabla _{\theta }\log p_{\theta }(x|z)]\approx {\frac {1}{L}}\sum _{l=1}^{L}\nabla _{\theta }\log p_{\theta }(x|z_{l})} but the gradient with respect to φ {\displaystyle \phi } requires the trick. Express the sampling operation z ~ q φ (z | x) {\displaystyle z\sim q_{\phi }(z|x)} as: z = μ φ (x) + σ φ (x) ■ ■ , ■ ~ N (0 , I) {\displaystyle z=\mu _{\phi }(x)+\sigma _{\phi }(x)\odot \epsilon ,\quad \epsilon \sim {\mathcal {N}}(0,I)} where μ φ (x) {\displaystyle \mu _{\phi }(x)} and σ φ (x) {\displaystyle \sigma _{\phi }(x)} are the outputs of the encoder network, and ■ {\displaystyle \odot } denotes element-wise multiplication . Then we have ∇ φ ELBO (φ , θ) = E ■ ~ N (0 , I) [∇ φ log ■ p θ (x | z) + ∇ φ log ■ q φ (z | x) − ∇ φ log ■ p (z)] {\displaystyle \nabla _{\phi }{\text{ELBO}}(\phi ,\theta)=\mathbb {E} _{\epsilon \sim {\mathcal {N}}(0,I)}[\nabla _{\phi }\log p_{\theta }(x|z)+\nabla _{\phi }\log q_{\phi }(z|x)-\nabla _{\phi }\log p(z)]} where z = μ φ (x) + σ φ (x) ■ ■ {\displaystyle z=\mu _{\phi }(x)+\sigma _{\phi }(x)\odot \epsilon } . This allows us to estimate the gradient using Monte Carlo sampling: ∇ φ ELBO (φ , θ) ≈ 1 L ∑ l = 1 L [∇ φ log ■ p θ (x | z l) + ∇ φ log ■ q φ (z l | x) − ∇ φ log ■ p (z l)] {\displaystyle \nabla _{\phi }{\text{ELBO}}(\phi ,\theta)\approx {\frac {1}{L}}\sum _{l=1}^{L}[\nabla _{\phi }\log p_{\theta }(x|z_{l})+\nabla _{\phi }\log q_{\phi }(z_{l}|x)-\nabla _{\phi }\log p(z_{l})]} where z l = μ φ (x) + σ φ (x) ■ ■ l {\displaystyle z_{l}=\mu _{\phi }(x)+\sigma _{\phi }(x)\odot \epsilon _{l}} and ■ l ~ N (0 , I) {\displaystyle \epsilon _{l}\sim {\mathcal {N}}(0,I)} for l = 1 , … , L {\displaystyle l=1,\ldots ,L} .

This formulation enables backpropagation through the sampling process, allowing for end-to-end training of the VAE model using stochastic gradient descent or its variants.

Variational inference

More generally, the trick allows using stochastic gradient descent for variational inference . Let the variational objective (ELBO) be of the form: ELBO (φ) = E z ~ q φ (z) [log ■ p (x , z) − log ■ q φ (z)] {\displaystyle {\text{ELBO}}(\phi)=\mathbb {E} _{z\sim q_{\phi }(z)}[\log p(x,z)-\log q_{\phi }(z)]} Using the reparameterization trick, we can estimate the gradient of this objective with respect to φ {\displaystyle \phi } : ∇ φ ELBO (φ) ≈ 1 L ∑ l = 1 L ∇ φ [log ■ p (x , g φ (■ l)) − log ■ q φ (g φ (■ l))] , ■ l ~ p (■) {\displaystyle \nabla _{\phi }{\text{ELBO}}(\phi)\approx {\frac {1}{L}}\sum _{l=1}^{L}\nabla _{\phi }[\log p(x,g_{\phi }(\epsilon _{l}))-\log q_{\phi }(g_{\phi }(\epsilon _{l}))],\quad \epsilon _{l}\sim p(\epsilon)}

Dropout

The reparameterization trick has been applied to reduce the variance in dropout , a regularization technique in neural networks. The original dropout can be reparameterized with Bernoulli distributions : y = (W ■ ■) x , ■ i j ~ Bernoulli (α i j) {\displaystyle y=(W\odot \epsilon)x,\quad \epsilon _{ij}\sim {\text{Bernoulli}}(\alpha _{ij})} where W {\displaystyle W} is the weight matrix, x {\displaystyle x} is the input, and α i j {\displaystyle \alpha _{ij}} are the (fixed) dropout rates.

More generally, other distributions can be used than the Bernoulli distribution, such as the gaussian noise: $y_{i}=\mu_{i}+\sigma_{i}\odot \epsilon_{i},\quad \epsilon_{i}\sim {\mathcal {N}}(0,I)$ where $\mu_{i}=\mathbf{m}_{i}^{\top}x$ and $\sigma_{i}^{2}=\mathbf{v}_{i}^{\top}x^{2}$, with $\mathbf{m}_{i}$ and $\mathbf{v}_{i}$ being the mean and variance of the i -th output neuron. The reparameterization trick can be applied to all such cases, resulting in the variational dropout method. [7]

See also

Variational autoencoder

Stochastic gradient descent

Variational inference

References

Further reading

Ruiz, Francisco R.; AUEB, Titsias RC; Blei, David (2016). "The Generalized Reparameterization Gradient" . Advances in Neural Information Processing Systems . 29 . arXiv : 1610.02287 . Retrieved September 23, 2024 .

Zhang, Cheng; Butepage, Judith; Kjellstrom, Hedvig; Mandt, Stephan (2019-08-01). "Advances in Variational Inference". IEEE Transactions on Pattern Analysis and Machine Intelligence . 41 (8): 2008– 2026. arXiv : 1711.05597 . Bibcode : 2019ITPAM..41.2008Z . doi : 10.1109/TPAMI.2018.2889774 . ISSN 0162-8828 . PMID 30596568 .

Mohamed, Shakir (October 29, 2015). "Machine Learning Trick of the Day (4): Reparameterisation Tricks" . The Spectator . Retrieved September 23, 2024 .