Regularizations: weight decay, dropout, normalizations

Seongok Ryu

KAIST Chemistry

Contents

- Importance of regularizations
- Weight decay
- Bayesian interpretation of weight decay
- Dropout
- Batch normalization
- Other normalizations

Importance of regularizations

- Empricial risk minimization aims to obtain a model (parameter) by minimizing the risk of our hypothesis on *empirical data*.

$$\hat{h} = \underset{h \in \mathcal{H}}{\operatorname{argmin}} R_{emp}(h) \quad s.t.\ R_{emp}(h) = \frac{1}{n}\sum_{i=1}^{n} L(h(x_i), y_i)$$

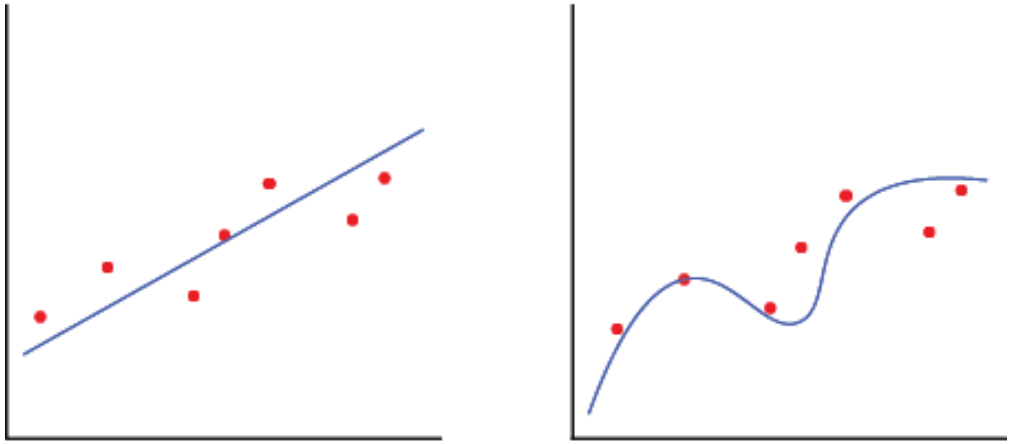
- However, minimizing the empirical risk can results over-fitting problems, where the hypothesis can give completely wrong results on outlier samples.
- Therefore, a variety of regularization techniques has been invented to decrease the complexity of a resulting hypothesis.

Weight decay

- Weight decay is a regularization approach that minimize the L_p-norm of weight parameters $\|w\|^p$ as well as a loss function.

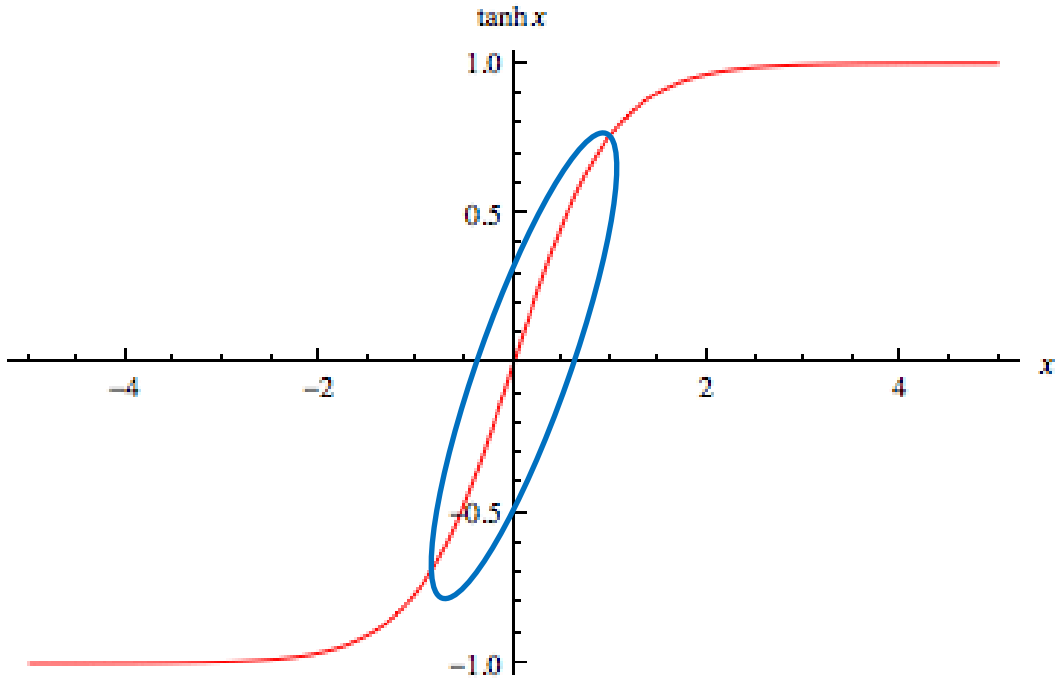
$$\hat{h} = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} L(h(x_i), y_i) + \lambda \cdot \|w\|^p$$

- If weight parameters have high values, the output will be changed as a small change of input values.
- Therefore, penalizing the norm of weight parameters can results less complex model.



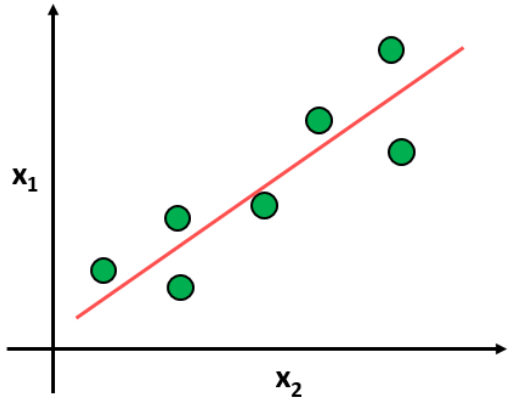
Weight decay

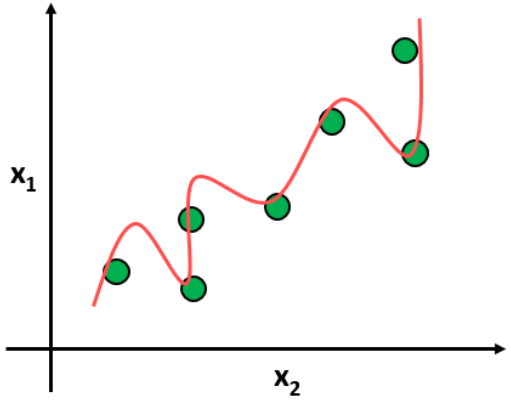
- Outputs from the activation function with small weight values are nearly linear, which allows roughly linear models



$$h = \sigma(w \cdot x + b)$$

Small w ➔ active function becomes linear➔ NN becomes roughly linear regression

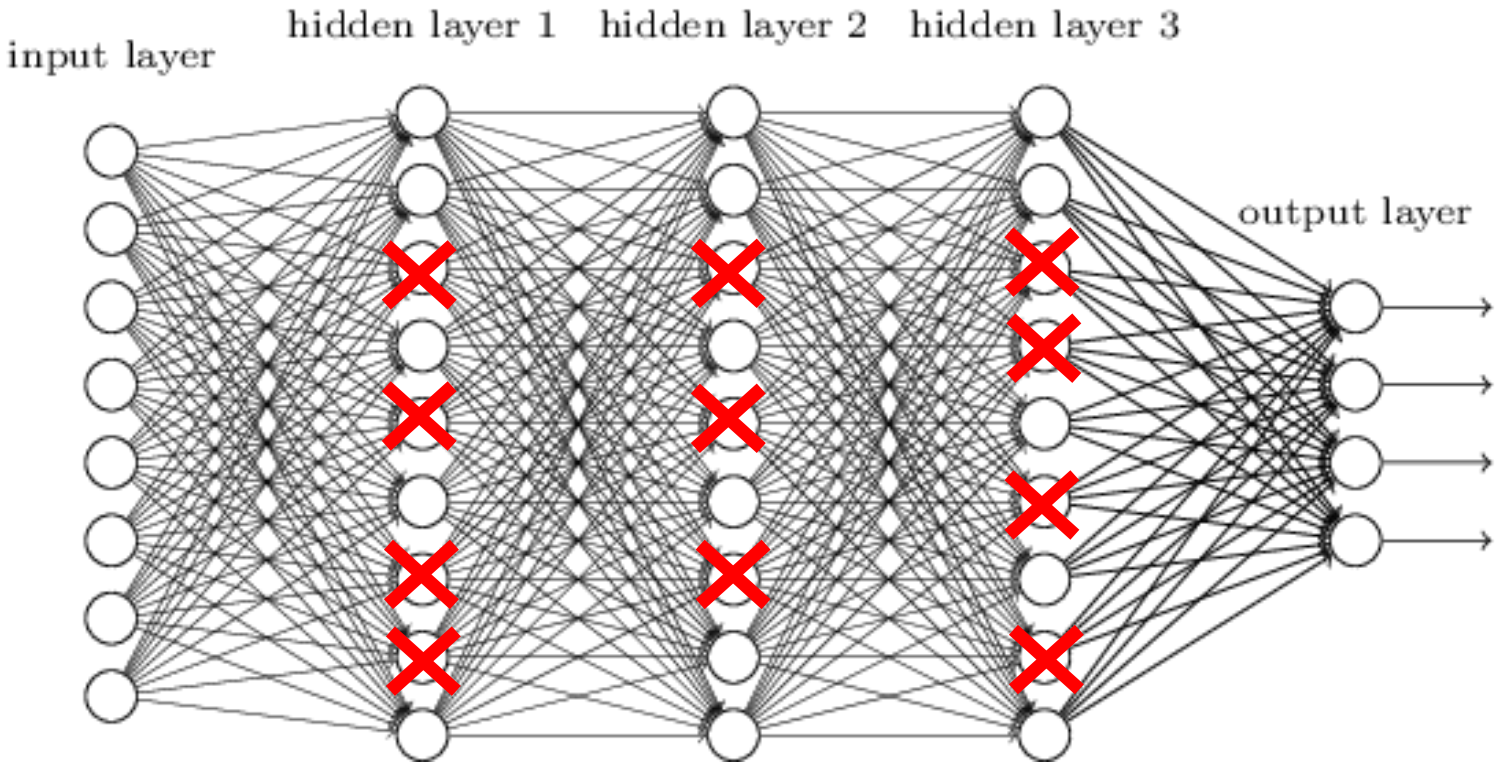




https://www.youtube.com/watch?v=NyG-7nRpsW8

Weight decay

- Also, small weights make some of hidden neurons have almost zero values, which reduce a model complexity.



➔ much simpler DNN

https://www.youtube.com/watch?v=NyG-7nRpsW8

Weight decay

- Let's investigate the mathematical description of L_p regularizations.
- First, L_p norm is given by

$$\|w\|^p = \left(\sum_i |w_i|^p\right)^{1/p}$$

- L_0 norm: the number of non-zero elements, ex) the L_0 norm of the vector (0,2) is 1 since there is only one non-zero element.
- L_1 norm: the sum of the magnitudes of the elements, also known as Manhattan distance or Taxicab norm, ex) the L1 norm of the vector (3,4) is $|3| + |4| = 7$
- L_∞ norm: the largest magnitude along each element of a vector, ex) the L_∞ norm of the vector (-6, 4, 2) is 6.

Weight decay

- Penalizing the L_0 norm is also named as a *sparsity regularization*, which attempts to reduce the number of non-zero weight parameters.
- *L_1-regularization* usually makes small parameters values be zero and gives large parameters. We can interpret that L1-regularization gives important parameters with large values and zero for otherwise. It also helps us to obtain sparse models.
- *L_2-regularization* is most widely used approach in current machine learning systems. It reduce the value of parameters more for whose value is large and vice versa. Therefore, resulting parameters have an even value and a model predicts its output by taking more number of input features.
- Comparison between *L1*- and *L2- regularizations*: https://shwksl101.github.io/ml/dl/2019/01/27/L1_L2_regularization.html

Bayesian interpretation of weight decay

- Recall that the expression of posterior and maximum-a-posteriori estimation of model parameters.

$$p(w|X,Y) \propto p(Y|X,w)p(w)$$

$$\hat{w}_{map} = \underset{w}{\operatorname{argmax}}\, p(w|X,Y)$$

where the first and second term in R.H.S. of the first line are a likelihood and a prior distribution, respectively.

- We can obtain the same solution by minimizing the negative log of the posterior:

$$\hat{w}_{map} = \underset{w}{\operatorname{argmin}} -\log p(Y|X,w) - \log p(w)$$

where the first term is the negative likelihood term which corresponds to a loss function.

Bayesian interpretation of weight decay

- If we let the prior distribution as a Gaussian distribution with zero mean and finite variance $p(w) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\|w\|^2}{2\sigma^2}\right)$, then the negative log of posterior is given by

$$\hat{w}_{map} = \underset{w}{\operatorname{argmin}} -\log p(Y|X,w) + \lambda \cdot \|w\|^2$$

  which becomes equivalent to L_2-regularization approach.

- On the other hand, if the prior distribution follows a Laplace distribution with zero mean and finite diversity, $p(w) = \frac{1}{2b} \exp\left(-\frac{\|w\|}{b}\right)$, then the objective function is given by

$$\hat{w}_{map} = \underset{w}{\operatorname{argmin}} -\log p(Y|X,w) + \lambda \cdot \|w\|$$

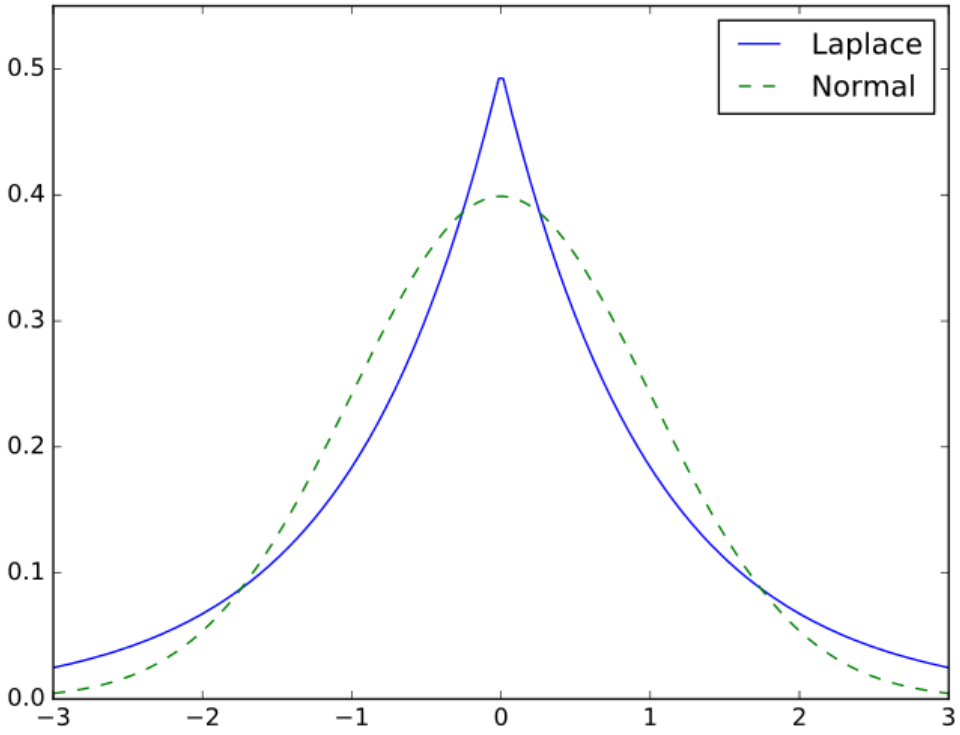
  which is equivalent to L_1-regularization approach.

Bayesian interpretation of weight decay

- As we can see from the right figure, the weight parameters sampled from a Laplace distribution tend to have zero values more than that follow a Gaussian distribution.
- Thus, we can understand that

  * L_1-regularization allows more sparse model and gives large values for important weight parameters.

  * L_2-regularization results wider range of parameter values than L1-regularization.



Dropout

- Dropout is one of widely used regularization method in modern deep neural networks, which regularizes models by masking hidden neurons randomly.

- Feed-forward operations in standard neural networks

$$z_i^{(l+1)} = \sigma\left(\mathbf{w}_i^{(l+1)}\mathbf{y}^{(l)} + b_i^{(l+1)}\right)$$

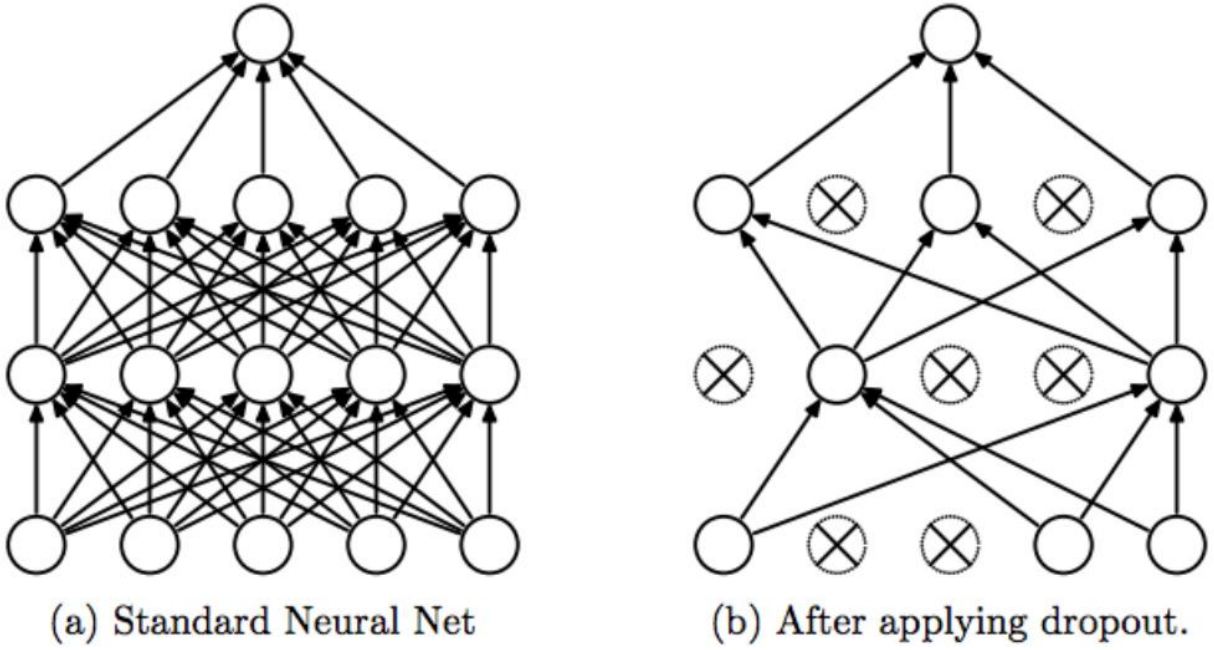
- Feed-forward operations after applying dropout

$$r_j^{(l)} \sim \text{Bernoulli}(p)$$

$$\tilde{\mathbf{y}}^{(l)} = \mathbf{r}^{(l)} * \mathbf{y}^{(l)}$$

$$z_i^{(l+1)} = \sigma(\mathbf{w}_i^{(l+1)}\tilde{\mathbf{y}}^{(l)} + b_i^{(l+1)})$$

- It is very important that the dropout mask is stochastically sampled from a Bernoulli distribution.

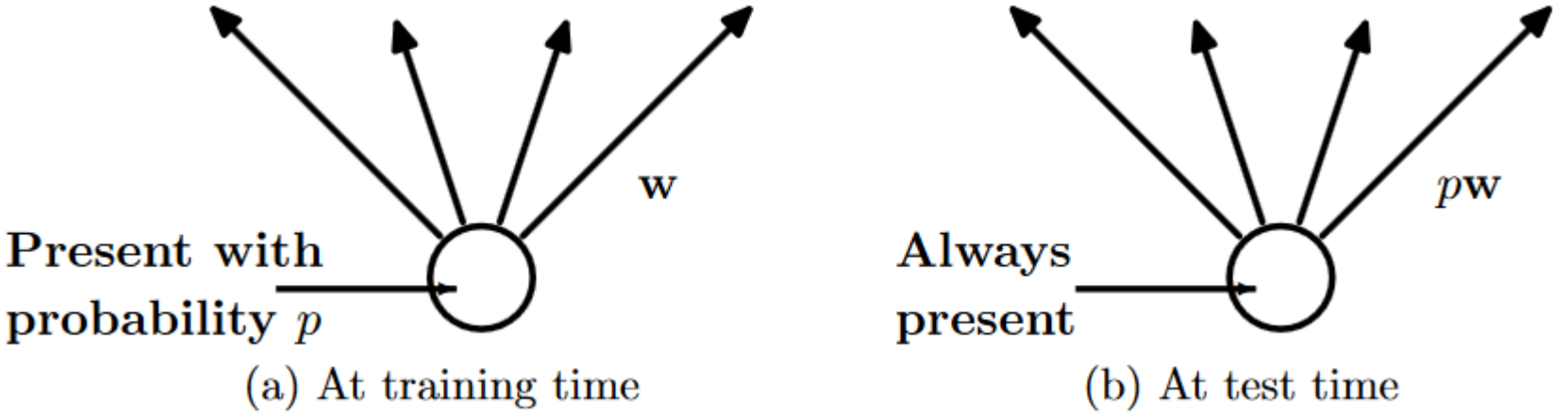


(a) Standard Neural Net (b) After applying dropout.

Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15.1 (2014): 1929-1958.

Dropout

- At training phase, dropout masks are kept turning on with probability p, which allows using only a fractional number of hidden features.
- At inference time, dropout masks are turned off and weight parameter values are divided by p.
- MC-dropout network keep turning on dropout masks but sampling its predictive output and averaged them to infer final predictions. This is one of the famous approximation of neural networks with dropout variational distribution. We will investigate more detail in the chapter “Bayesian neural network”.

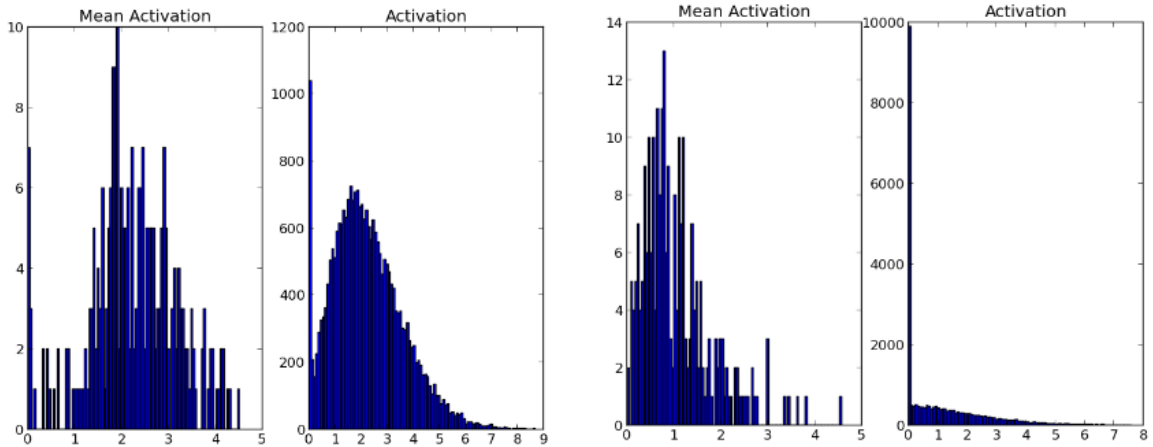


(a) At training time (b) At test time

Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15.1 (2014): 1929-1958.

Dropout

- Why dropout works?
- If some of hidden neurons were masked, then the output is predicted by using the fractional number of information. This procedure prevents a model depends on specific features too much for predicting outputs, helps us avoid over-fitting problem.
- Bayesian interpretation of dropout tells us that a set of randomly masked weight parameters corresponds to the variational approximation of posterior distribution, so-called the MC-dropout network. We will see this in next chapter.



(a) Without dropout (b) Dropout with $p = 0.5$.

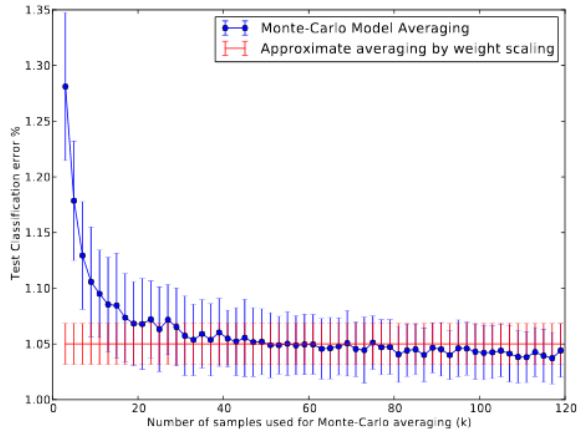
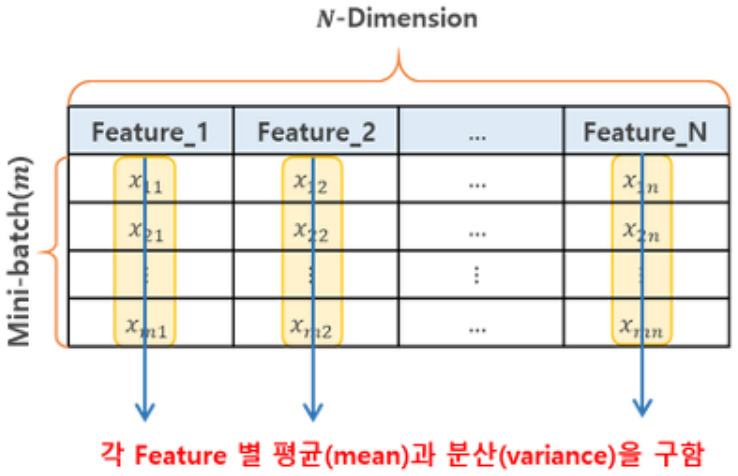


Figure 11: Monte-Carlo model averaging vs. weight scaling.

Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *The journal of machine learning research* 15.1 (2014): 1929-1958.

Batch normalization

- Batch normalization is a technique that normalizes features with mean and standard deviation of samples in given mini-batch.



Normalize

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

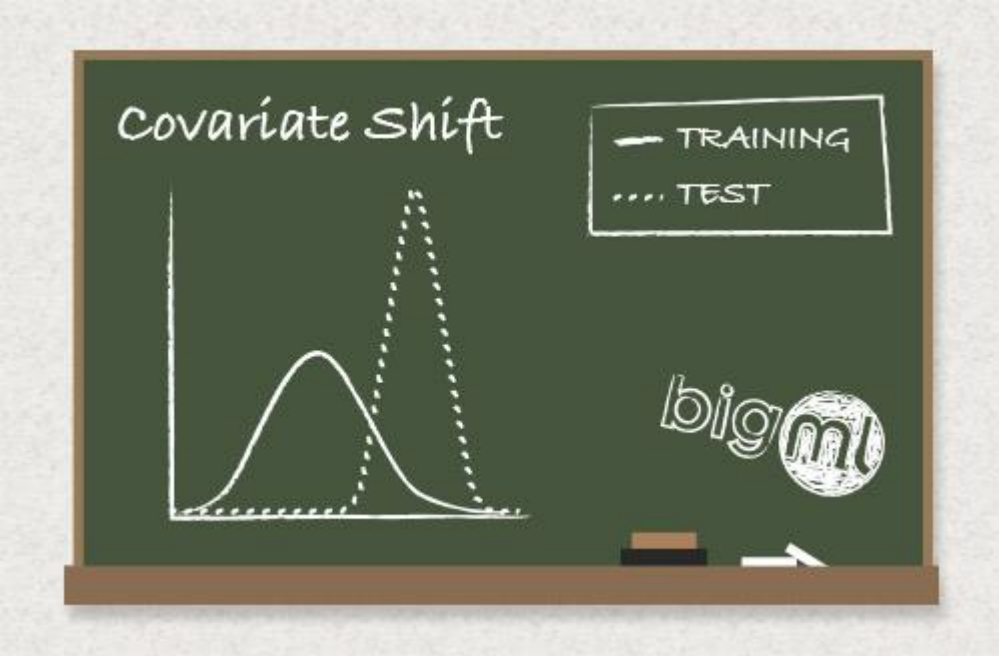
$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

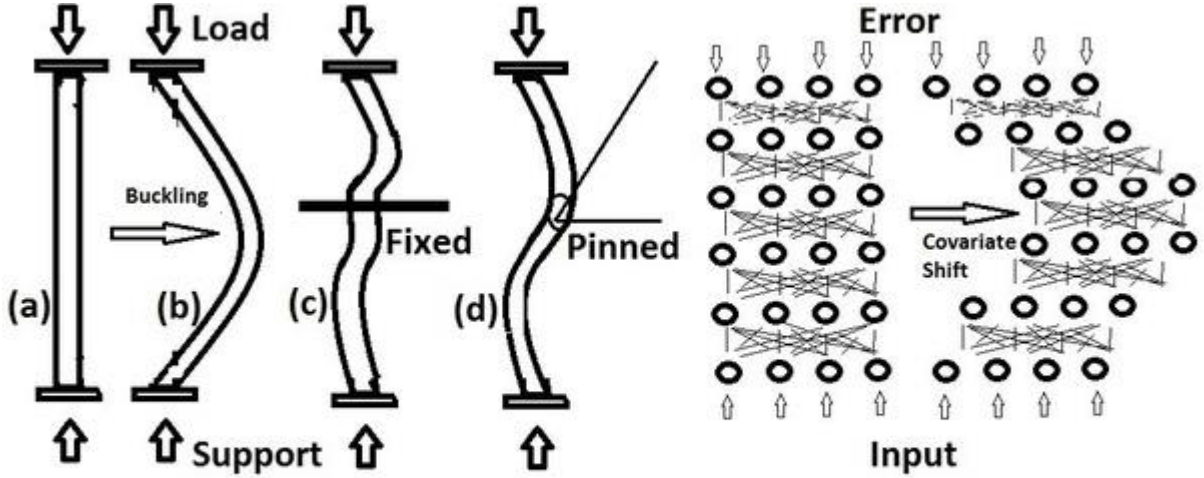
$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *arXiv preprint arXiv:1502.03167* (2015).

Batch normalization

- Why it works?
- In the original paper (ICML 2015), authors argued that batch normalization helps to avoid “internal covariate shift” of features in deep neural networks.





For both, Buckling or Co-Variate Shift a small perturbation leads to a large change in the later.

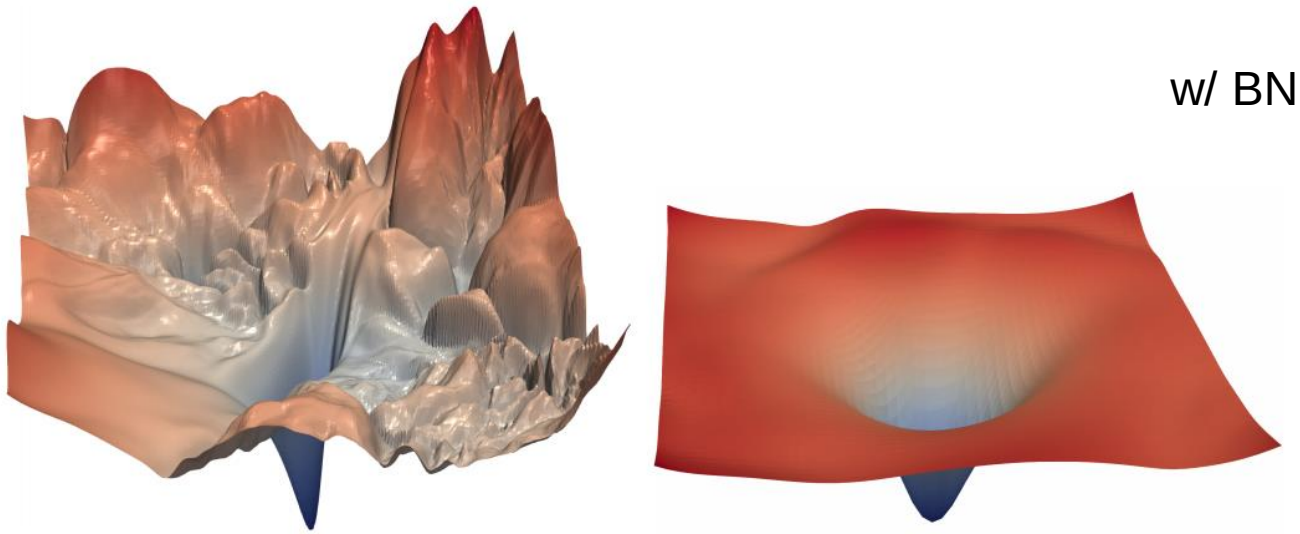
Debiprasad Ghosh, PhD, Uses AI in Mechanics

Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *arXiv preprint arXiv:1502.03167* (2015).

Batch normalization

- Why it works?
- MIT researchers presented their understanding on batch normalization (“How does batch normalization help optimization?”) in NIPS2018.
- They show that batch normalization does not help to avoid internal covariate shifts but make model’s loss landscape smoother.
- That is the loss changes at a smaller rate and the magnitudes of the gradients are smaller too, which enables us to use large learning rate and models to be converged much faster.

w/o BN



w/ BN

Santurkar, Shibani, et al. "How does batch normalization help optimization?." *Advances in Neural Information Processing Systems*. 2018.

Batch normalization

- Results
- BN allows using larger learning rate for training, leading to much faster convergence of models.

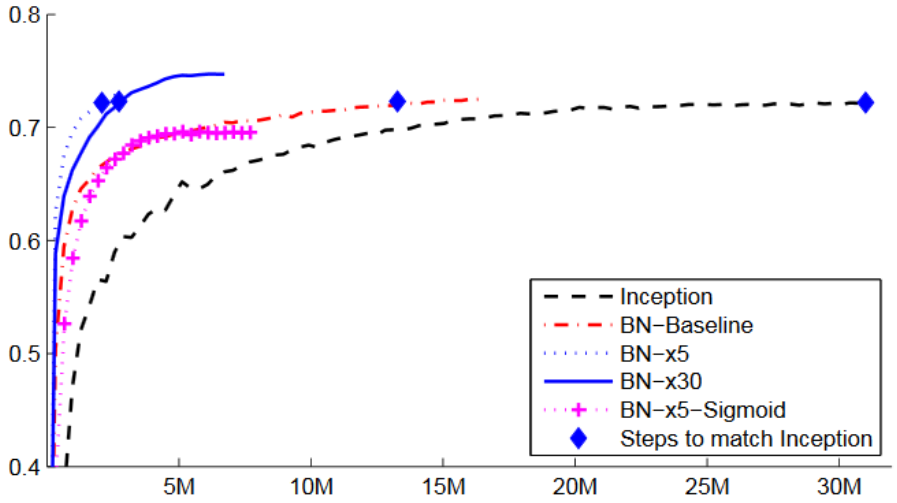
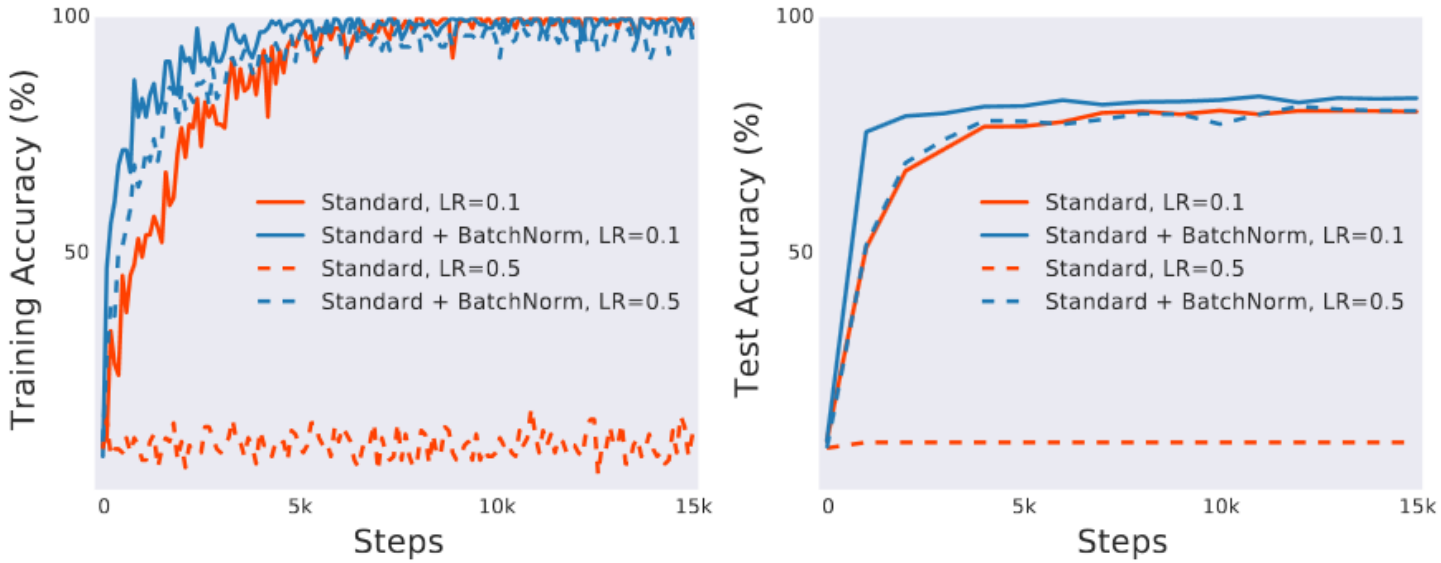


Figure 2: *Single crop validation accuracy of Inception and its batch-normalized variants, vs. the number of training steps.*

Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *arXiv preprint arXiv:1502.03167* (2015).

VGG network on CIFAR-10



Santurkar, Shibani, et al. "How does batch normalization help optimization?." *Advances in Neural Information Processing Systems*. 2018.

Other normalizations

- There are various ways to normalize hidden features with respect to the normalization axis.
- Since the performance of batch normalization is strongly depends on the mini-batch size, other techniques have been invented.
- Layer normalization is especially considered as a standard technique for natural language processing.
- Instance normalization is frequently used in style transfer models.

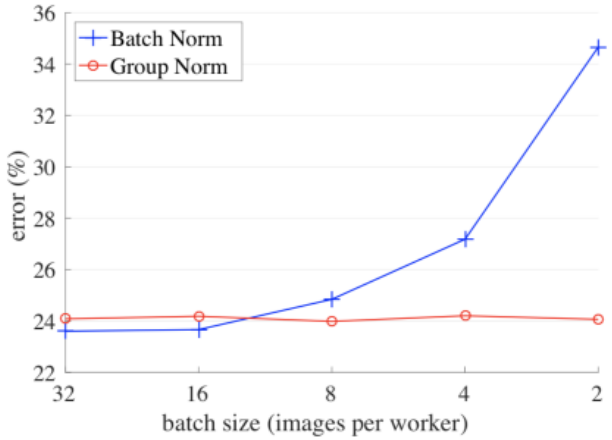


Figure 1. **ImageNet classification error *vs*. batch sizes**. This is a ResNet-50 model trained in the ImageNet training set using 8 workers (GPUs), evaluated in the validation set.