Universidade de São Paulo
Escola Politécnica - Engenharia de Computação e Sistemas Digitais

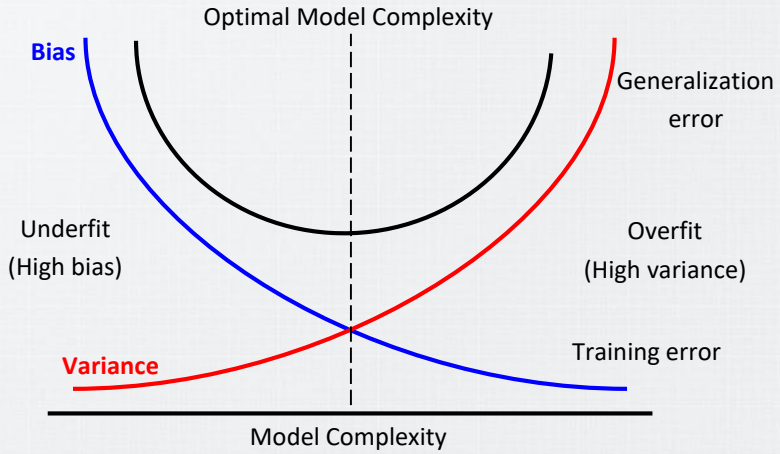
Regularization

Prof. Artur Jordão

Introduction

Regularization

- A central problem in deep learning (and other machine learning strategies) is how to make an algorithm that performs well not only on the training data but also on new (unseen) inputs
  - **Reducing the generalization gap between training and test data**



Definition

Regularization

- Any modification we make to a learning algorithm that aims **to reduce its generalization error without affecting its training error**

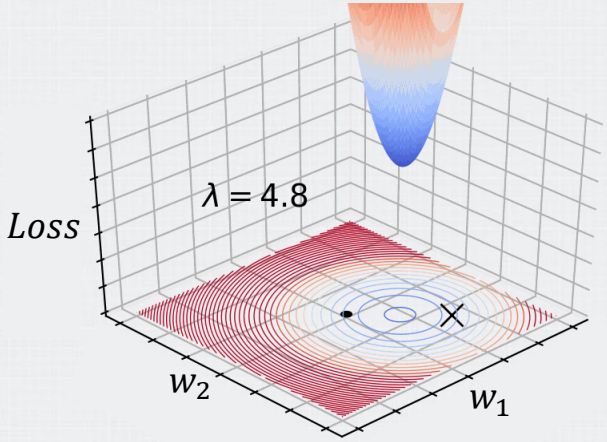
- Regularization strategies
  - Weight Decay
  - Dropout
  - Early stopping
  - Ensembling
  - Applying noise
  - Data augmentation

Weight Decay

Regularization

- Weight decay involves adding a penalty to the loss function used for training the neural network
  - It places a L_p-penalty on the parameters to encourage a minimum-norm solution

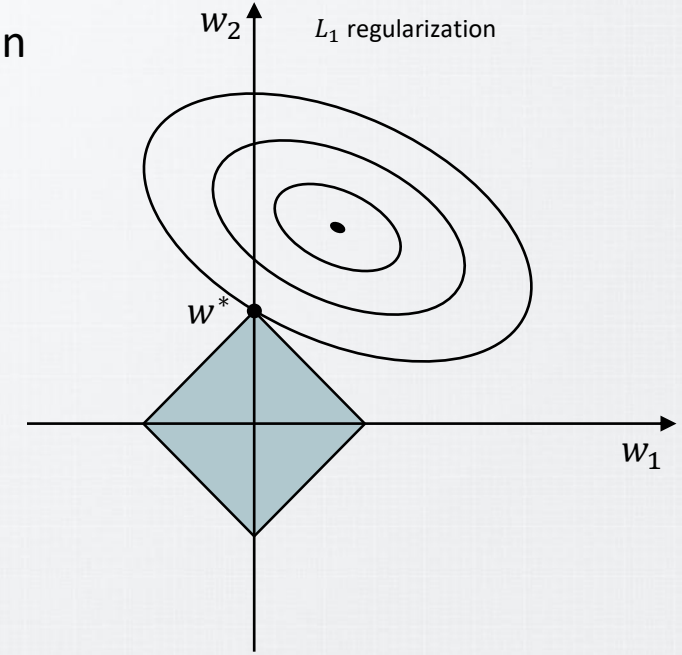
- Formally, with weight decay the loss function becomes
  - $Loss(W) + \lambda Complexity(W)$

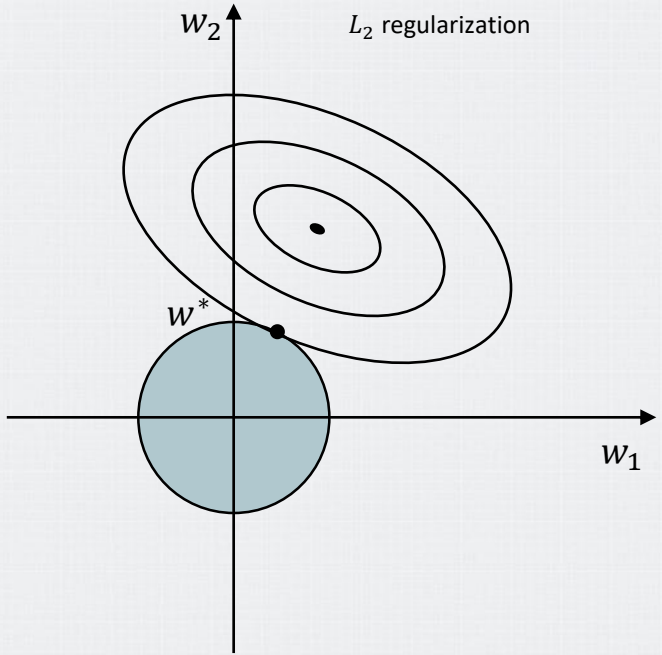


Weight Decay

Regularization

- L_1 regularization
  - It enables the model to learn sparse solutions and drop irrelevant features
- L_2 regularization
- L_p
  - $(\sum_i |w_i|^p)^{\frac{1}{p}}$





Early Stopping

Regularization

- This form of regularization refers to stopping the training procedure before it reaches full convergence
  - Here, convergence stands for the final training epoch

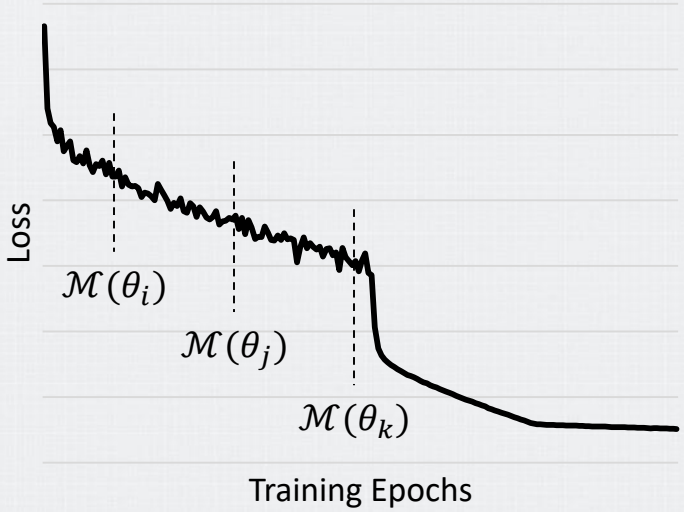
- Early stopping has a single hyperparameter: the number of steps after which training is terminated

- Early stopping can reduce overfitting if the model has **already captured the coarse shape of the underlying function** but has **not yet had time to overfit the noise**

Early Stopping

Regularization

- General idea
  - The model is trained once upon completion
  - The algorithm monitors the performance, $\mathcal{M}$, on a **validation set** every T iterations and stores the corresponding models
  - Select the best model according to the validation performance
  - $\theta = \max_{t \in T} \mathcal{M}(\theta_t)$



Ensembling

Regularization

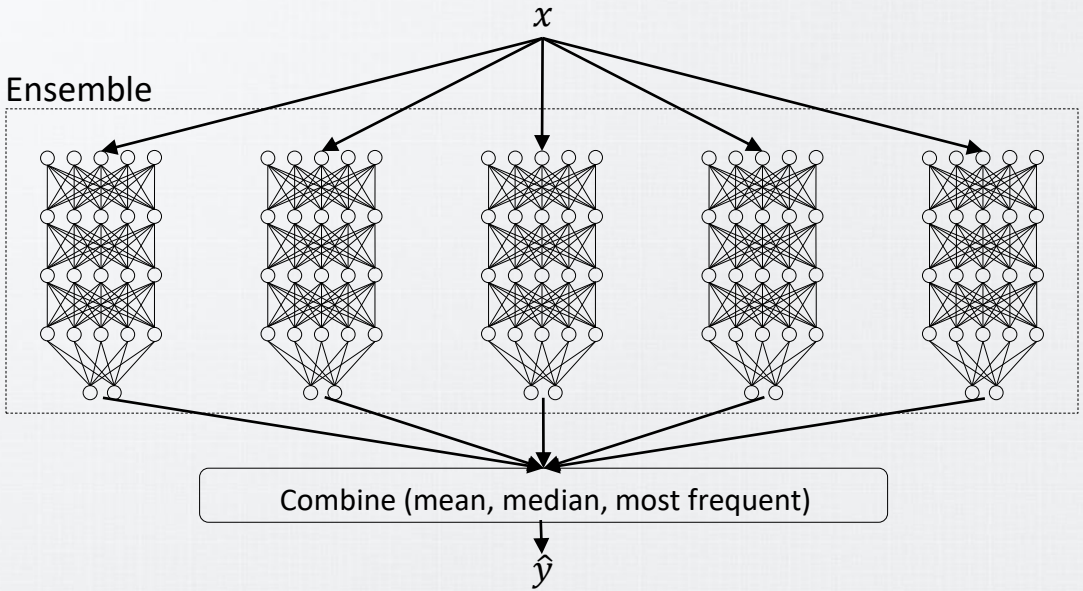
- The idea involves constructing multiple models and averaging their predictions
  - This group is referred to as an **Ensemble**

- Strategies for training the ensemble members (models)
  1. Train different models using different random initializations
  2. Generate several different datasets by resampling the original training data with replacement and train a different model from each (bootstrap aggregating – **bagging**)
  3. Training models with varying hyperparameters
  4. Training models different families of architectures

Ensembling

Regularization

- Strategies for building ensembles
  1. Take the mean of the outputs (for both regression and classification problems)
  2. Take the median of the outputs (for regression problems)
  3. Take the most frequently predicted class (for classification problems)

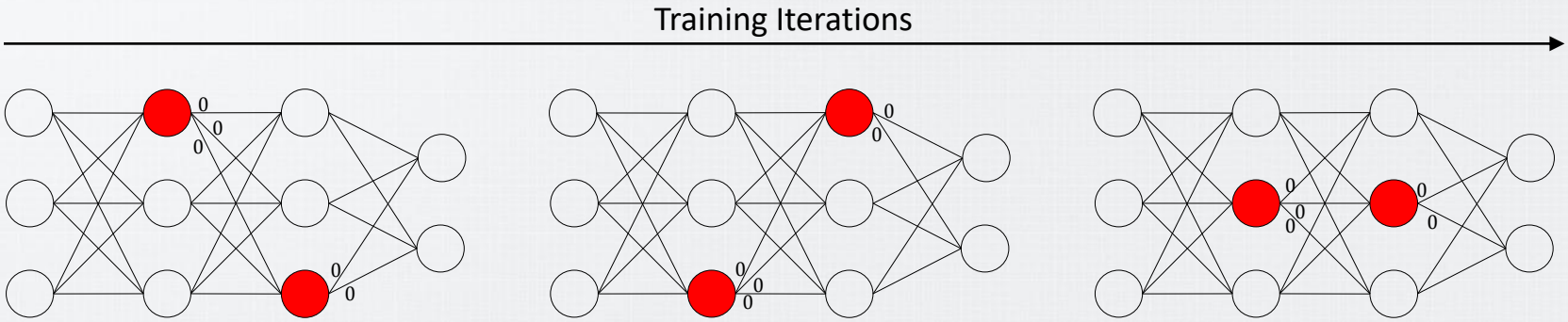


Dropout

Regularization

- Dropout randomly clamps a subset of neurons to zero at each training iteration
  - Involves removing structures during training
  - At the testing time, the structures remain unchanged

- Dropout is the equivalent of training several independent, smaller networks on the same task



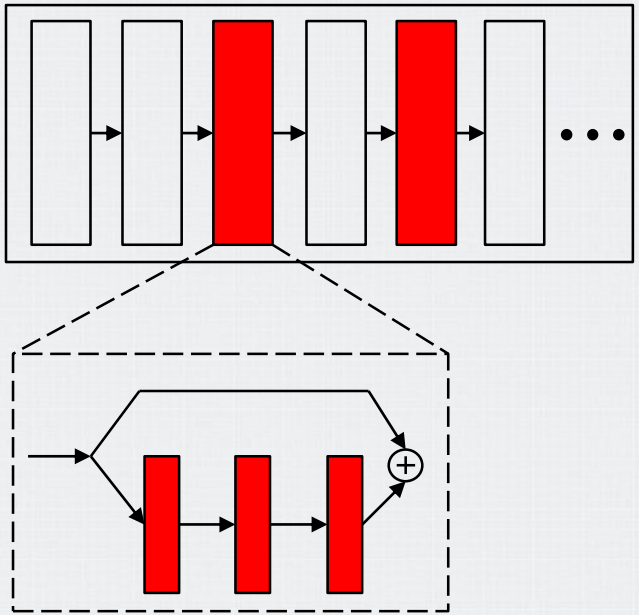
Alternative Forms of Dropout

Regularization

- Originally, dropout focused on removing individual neurons
  - More recently, studies have suggested dropping entire layers (Huang et al., 2016)

- Layer Dropout (Huang et al., 2016)

- Example-tied dropout (Maini et al., 2023)



Huang et al. *Deep networks with stochastic depth*. In European Conference on Computer Vision (ECCV), 2016

Maini et al. *Can Neural Network Memorization Be Localized?*. In International Conference on Machine Learning (ICML), 2023

Critical Periods for Regularization

Regularization

- Regularization at different phases of training produces varying outcomes

- There is a **critical period** during the initial epochs of training when regularization is more effective (Golatkar et al., 2019; Kleinman et al., 2023-2024)
  - After the critical period, regularization may not have any benefit in terms of regularizing the networks to generalize well

Golatkar et al., *Time Matters in Regularizing Deep Networks: Weight Decay and Data Augmentation Affect Early Learning Dynamics, Matter Little Near Convergence*. Neural Information Processing Systems (NeurIPS), 2019

Kleinman et al. *Critical Learning Periods for Multisensory Integration in Deep Networks*. Computer Vision and Pattern Recognition (CVPR), 2023

Kleinman et al. *Critical Learning Periods Emerge Even in Deep Linear Networks*. International Conference on Learning Representations (ICLR), 2024

Batch Normalization

Introduction

Batch Normalization

- Batch normalization (Ioffe et al., 2015) has made it possible for practitioners to routinely train networks with over 100 layers

- Batch normalization has been established as the default component of most modern neural network architectures
  - It improves training speed and enhances generalization

Ioffe et al. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In International Conference on Machine Learning (ICML), 2015

Introduction

Batch Normalization

- The widely known motivation for the Batch Normalization (BatchNorm or BN) technique is to normalize the shifts in input distribution caused by updates to successive layers
  - A phenomenon referred to as **internal covariate shift**

- BatchNorm **shifts** and **rescales** each activation so that its mean and variance across the batch become learned values during training
  - It has been designed to stabilize and accelerate the training process by normalizing intermediate representations during mini-batch processing

- BatchNorm learns affine linear transformations of neurons

Ioffe et al. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In International Conference on Machine Learning (ICML), 2015

Definition

Batch Normalization

- Let x be a data sample and β be the batch size
- Batch Normalization (BatchNorm) rescales each data sample in terms of

$$BatchNorm(x) = \left(\frac{x - \mu_\beta}{\sqrt{\sigma_\beta^2 + \epsilon}}\right) \odot \gamma + \delta$$

Shift by δ

Scale by γ

- μ_β and σ_β stands for the mean and variance **across the batch size**
- γ and δ are **learnable** parameters

Properties of Batch Normalization

- To ensure deterministic inference, we set the mean (μ) and variance (σ) to their exponential moving statistics (estimated during the training phase)
  - Hence, batch normalization behaves differently during training than at test time

- The widely known motivation of the Batch Normalization (BN) technique is to mitigate **internal covariate shift**

- Santurkar et al. (2018) observed that internal covariance shift does not exist, attributing the success of BN to its ability to smooth the loss landscape, thereby enhancing and accelerating convergence

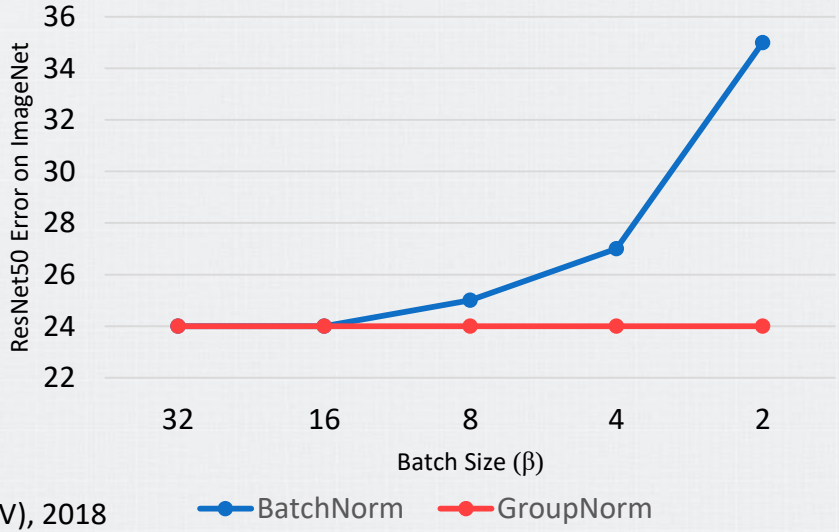
Santurkar et al. *How does batch normalization help optimization?* In Neural Information Processing Systems (NeurIPS), 2018

Alternative Forms of Normalization

Batch Normalization

- The major disadvantages of BatchNorm include its high computational and memory costs, as well as its disruption of the independence among minibatch samples
  - BatchNorm works best in combination with relatively **large batch sizes** (Wu et al., 2018)

- These drawbacks have inspired the search for alternative forms of normalization



Wu et al. *Group Normalization*. International Conference on Computer Vision (ICCV), 2018

Alternative Forms of Normalization

Batch Normalization

- Layer normalization (LayerNorm)
  - Contrary to BatchNorm, LayerNorm is applied to **one observation at a time**
  - Batch-agnostic

- Layer normalization estimates statistics over all the **hidden units H**
  - H refers to the same layer L

$$BatchNorm(x) = \left(\frac{x - \mu_\beta}{\sqrt{\sigma_\beta^2 + \epsilon}}\right) \odot \gamma + \delta,$$

$$\mu_\beta = \frac{1}{\beta}\sum_{i=1}^{\beta} x$$

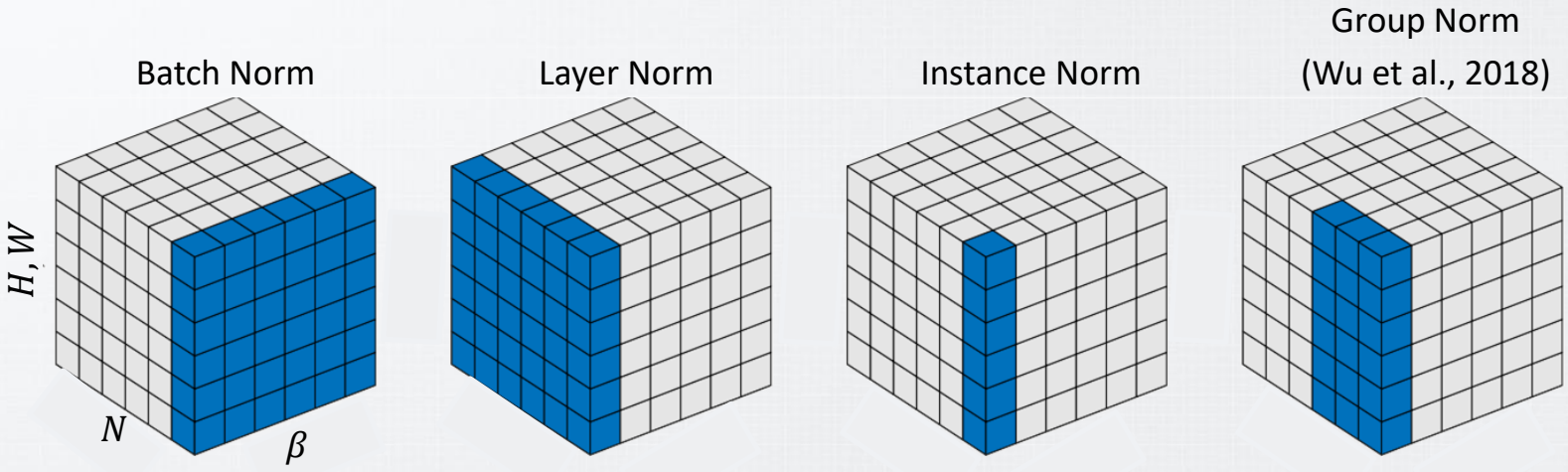
$$LayerNorm(x) = \left(\frac{x - \mu_L}{\sqrt{\sigma_L^2 + \epsilon}}\right),$$

$$\mu = \frac{1}{H}\sum_{i=1}^{H} x_i \, , \sigma = \frac{1}{H}\sum_{i=1}^{H} (x_i - \mu)^2$$

Alternative Forms of Normalization

Batch Normalization

- Definitions
  - β batch size
  - N number of neurons
  - H, W output dimension



Ulyanov et al. *Instance normalization: The missing ingredient for fast stylization*. ArXiv, 2016

Wu et al. *Group Normalization*. International Conference on Computer Vision (ICCV), 2018

Final Notes

Batch Normalization

- Normalization layers are integral to many deep learning architectures and significantly contribute to their success
  - In particular, they are crucial for Convolutional Neural Networks

- Surprisingly, previous works have confirmed that **training BN parameters alone** while **keeping the remaining neural network parameters frozen** to their initial values can achieve nontrivial performance
  - Especially for deeper and wider networks (Frankle et al., 2021; Burkholz, 2024)

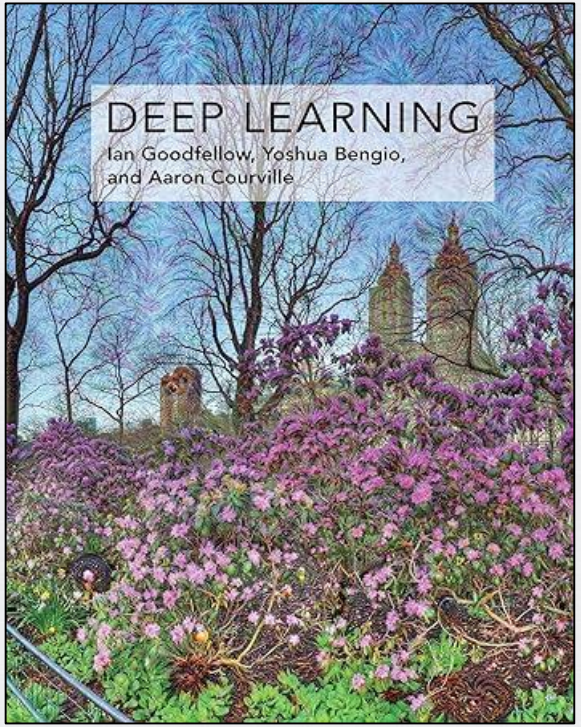
Frankle et al. *Training BatchNorm and only BatchNorm: On the expressive power of random features in CNNs*. In International Conference on Learning Representations (ICLR), 2021.

Burkholz. *Batch Normalization is Sufficient For Universal Function Approximation in CNNs*. International Conference on Learning Representations (ICLR), 2024

Bibliography

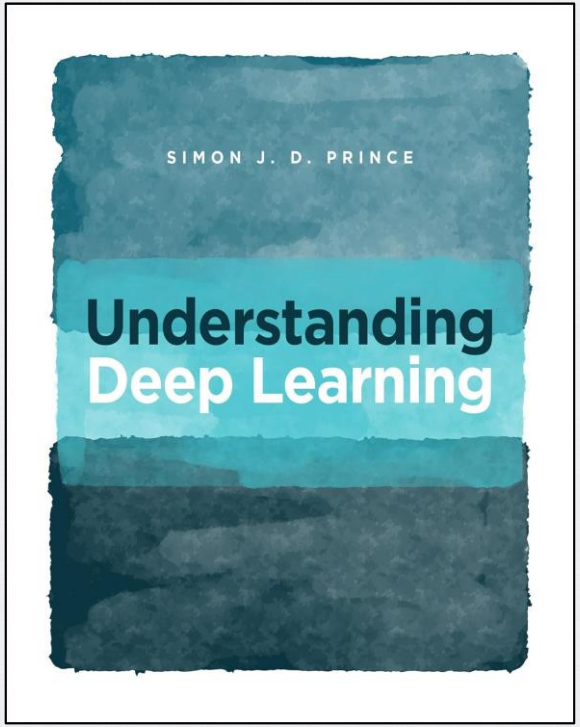
Bibliography

- Deep Learning
  - Chapter 7
    - 7.2 Dropout



Bibliography

- Understanding Deep Learning
  - Chapter 9
    - 9.3.1 Early stopping
    - 9.3.2 Ensembling
    - 9.3.3 Dropout
    - 9.4 Summary



Bibliography

- Burkholz. *Batch Normalization is Sufficient For Universal Function Approximation in CNNs*. International Conference on Learning Representations (ICLR), 2024

- Santurkar et al. *How does batch normalization help optimization?.* Neural Information Processing Systems (NeurIPS), 2018

ICLR
International Conference On Learning Representations

NEURAL INFORMATION PROCESSING SYSTEMS