



Regularizations

Deep Neural Networks
Session 10
Pramod Sharma
pramod.sharma@prasami.com

2 Agenda

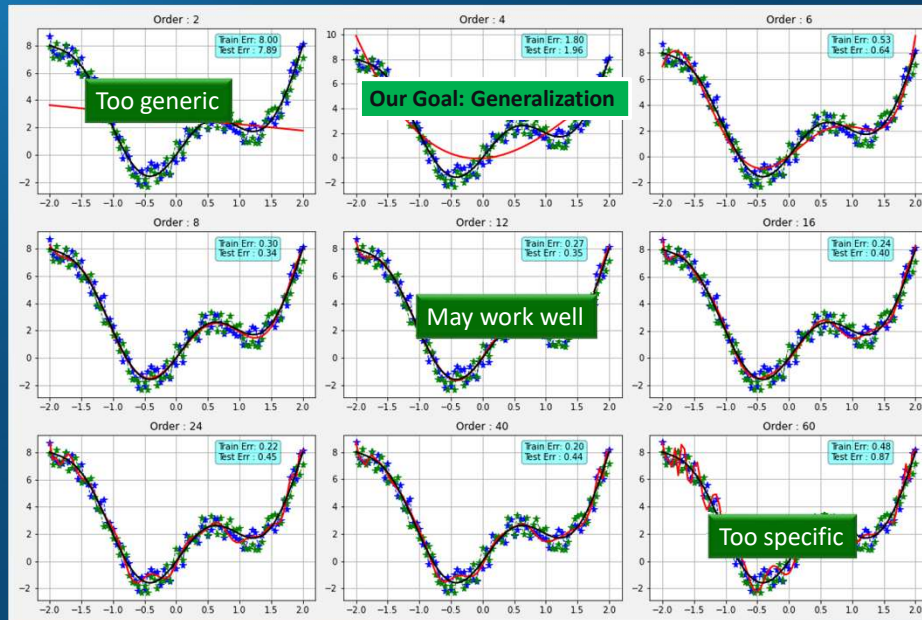
- L-1, L-2
- Dropout
- Early Stopping
- Augmentation

11/14/2025

pra-sami

3

Under-fitting vs. Over-fitting



11/14/2025

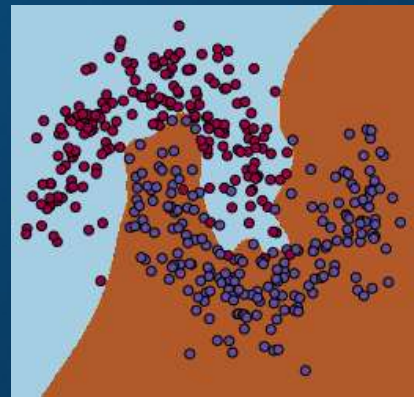
pra-sami

4

Regularization

- ❑ Regularization is like putting a penalty on a overly complex model.
- ❑ Regularization adds a *penalty* to the loss function.
- ❑ This penalty discourages large weights, leading to a simpler, smoother model.
- ❑ Different Regularization Techniques in Deep Learning
 - ❖ L1 regularization
 - ❖ L2 regularizations
 - ❖ Dropouts
 - ❖ Early stopping
 - ❖ Data Augmentation

Most Libraries have tunable hyper-parameters!



11/14/2025

pra-sami

5

Weights vs. Bias

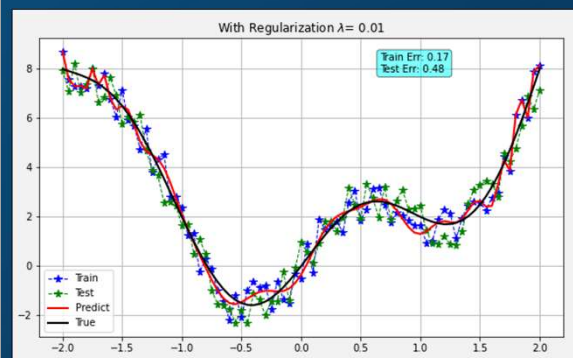
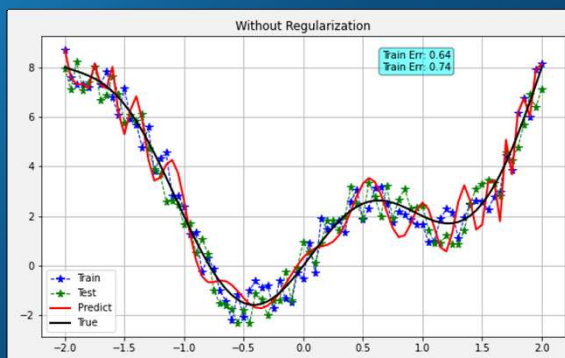
- ❑ For neural networks, we typically choose to use a parameter norm penalty Ω that penalizes only the weights at each layer and leaves the biases un-regularized.
- ❑ The biases typically less in numbers than the weights
- ❑ Weights control both (fan_in and fan_out) layers
- ❑ Each bias controls only a single layer.
- ❑ This means that we do not lose too much variance by leaving the biases un-regularized.

11/14/2025

pra-sami

6

Effect of L2 Regularization



11/14/2025

pra-sami

7

Regularization L1 and L2

□ L1 (Lasso):

- ❖ Like a tax collector who takes \$10 from everyone, regardless of income
- ❖ Poor people (small weights) go to \$0
- ❖ Rich people (large weights) keep most of their money

□ L2 (Ridge):

- ❖ Like an income tax - takes 10% from everyone
- ❖ Poor people keep most of their money
- ❖ Rich people pay more but remain rich

11/14/2025

pra-sâmi

8

Theory – Logistic Regression – L1 & L2

□ L1 (Lasso) tends to drive weights to zero, causing sparsity. L2 (Ridge) shrinks weights evenly

□ Idea is to minimize Cost Function

$$\begin{aligned} \diamond J(W, b) &= \frac{1}{m} * \sum \ell(a, y) \\ \diamond &= -\frac{1}{m} \{y * \log(a) + (1-y) * \log(1-a)\} \end{aligned}$$

□ L2: A term is added to Cost function

$$L1 \rightarrow J(W, b) = \frac{1}{m} * \sum \ell(a, y) + \frac{\lambda}{m} * \|W\|_1$$

$$L2 \rightarrow J(W, b) = \frac{1}{m} * \sum \ell(a, y) + \frac{\lambda}{2 * m} * \|W\|_2^2$$

11/14/2025

pra-sâmi

9

Theory – Logistic Regression – L2

- $J(W, b) = \frac{1}{m} * \sum \ell(a, y) + \frac{\lambda}{2 * m} * \|W\|_2^2 + \frac{\lambda}{2 * m} * b^2$
 - ❖ This is referred as L2 regularization
 - ❖ **Regularization hyperparameter λ :** It is another parameter we tune...
- $\|W\|_2^2 = \sum_{j=1}^n w_j^2 = W^T \cdot W$
- Here, we are using Euclidean Norm or L2 Norm
- Compared to W , bias b has fewer dimensions, hence, it is generally not considered
- If you add for b , $(\frac{\lambda}{2 * m} * b^2)$... that's ok too
 - ❖ Although its effect will be minimal,
 - ❖ Better to leave it alone.

11/14/2025

pra-sâmi

10

Theory – Logistic Regression – L1

- Sometimes L1 too is used
- $J(W, b) = \frac{1}{m} * (\sum \ell(a, y)) + \frac{\lambda}{m} * \|W\|_1$
- Differentiation of $\frac{\lambda}{m} * \|W\|_1 = \frac{\lambda}{m} \text{sign}(W)$
 - ❖ Keeps moving towards zero at a constant rate

11/14/2025

pra-sâmi

11

Neural Network – Frobenius Norm

- In neural network, we have different layers with different weights
- So we look at its cumulative effect over all layers
- Hence the Cost function
 - ❖ $J(W, b) = J(W[1], b[1], W[2], b[2], W[3], b[3] \dots)$
 - ❖ $J(W, b) = \frac{1}{m} * (\sum \ell(a, y)) + \frac{\lambda}{2 * m} * \sum_{l=1}^L \sum (w_{i,j})^2$
 - ❖ $J(W, b) = \frac{1}{m} * \sum \{y * \log(a) + (1-y) * \log(1-a)\} + \frac{\lambda}{2 * m} * \sum_{l=1}^L \|W[l]\|^2$
 - ❖ Where $\|W[l]\|_F^2 = \sum_{i=1}^{n^{[l-1]}} \sum_{j=1}^{n^{[l]}} (w_{ij}^{[l]})^2$
 - W is $(n^{[l-1]}, n^{[l]})$ dimensional matrix
- It is called *Frobenius norm* of a matrix
- Also the *Frobenius norm* defined as the square root of the sum of the absolute squares of its elements

11/14/2025

pra-sâmi

12

Frobenius Norm of a Vector

$$\square \|A\|_F = \sqrt{\sum (a_{ij})^2}$$

i.e.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \sqrt{(1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + 7^2 + 8^2 + 9^2)}$$

11/14/2025

pra-sâmi

13

Updates to weights

□ Earlier

- ❖ $\partial W^{[l]} = X \cdot \partial z$
- ❖ And $W^{[l]} = W^{[l]} - \alpha \cdot \partial W^{[l]}$
- ❖ For Regularization we add an extra term at the end

$$\partial W^{[l]} = X \cdot \partial z + \frac{\lambda}{m} \cdot W^{[l]}$$

Mathematically, we can show that it is still a valid definition of $\partial W^{[l]}$

$$\partial W^{[l]} = W^{[l]} - \alpha \cdot [X \cdot \partial z + \frac{\lambda}{m} \cdot W^{[l]}]$$

$$\partial W^{[l]} = (1 - \frac{\alpha \cdot \lambda}{m}) \cdot W^{[l]} - \alpha \cdot X \cdot \partial z$$

Weight Decay

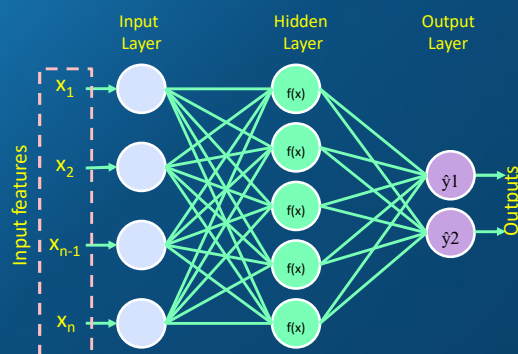
11/14/2025

pra-sâmi

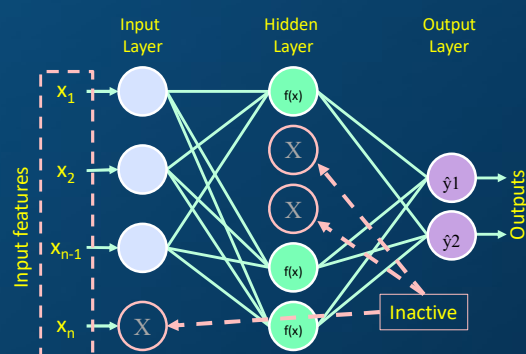
14

Regularization : Dropout

□ Original



□ With Dropout



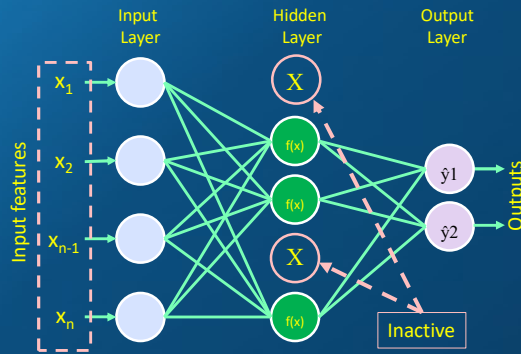
11/14/2025

pra-sâmi

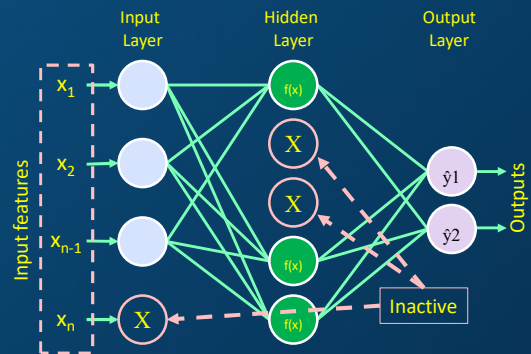
15

Regularization : Dropout

Iteration 1



Iteration 2



11/14/2025

pra-sami

16

Regularization : Early Stopping

- ❑ How long to train the model?
- ❑ Duration of training → under – fit or over – fit
- ❑ Train the model to the point where its performance on test set is best!
- ❑ Very simple and very effective

How:

- ❑ Train the model and monitor performance
- ❑ Save weight every time the performance improves
- ❑ Stop training if performance has not improved for N epochs
- ❑ It's the last parameter to tune
 - ❖ Repeated early stopping may lead to over-fitting the validation set
 - ❖ Example : K-fold

11/14/2025

pra-sami

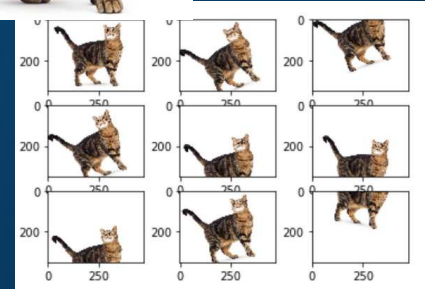
17

Regularization : Data Augmentation

- ❑ Where limited data is available for training the model (when is it not!)
- ❑ Very effective in image identification
- ❑ Most libraries have Image Generators (parameter driven)
 - ❖ Horizontal and Vertical Shift
 - ❖ Horizontal and Vertical Flip
 - ❖ Random Rotation
 - ❖ Random Brightness / Contrast
 - ❖ Random Zoom
 - ❖ Random Noise



<https://towardsdatascience.com/image-augmentation-for-deep-learning-histogram-equalization-a71387f609b2>



11/14/2025

pra-samī

18

Reflect...

- ❑ What is the purpose of dropout in deep neural networks?
 - ❖ A) To add noise to the input data
 - ❖ B) To randomly drop neurons during training to prevent overfitting
 - ❖ C) To increase the learning rate
 - ❖ D) To increase the model complexity
- ❑ Answer: B
- ❑ What is the primary purpose of regularization in deep neural networks?
 - ❖ A) To increase computational efficiency
 - ❖ B) To prevent overfitting
 - ❖ C) To speed up convergence during training
 - ❖ D) To increase the model's capacity
- ❑ Answer: B) To prevent overfitting

- ❑ Which type of regularization adds a penalty term to the loss function based on the absolute values of the weights?

- ❖ A) L1 Regularization
- ❖ B) L2 Regularization
- ❖ C) Dropout
- ❖ D) Batch Normalization

- ❑ Answer: A) L1 Regularization

- ❑ How does dropout regularization work?

- ❖ A) It penalizes large weights in the network
- ❖ B) It introduces noise to the input data during training
- ❖ C) It randomly removes neurons during training
- ❖ D) It normalizes the input features

- ❑ Answer: C) It randomly removes neurons during training

11/14/2025

pra-samī

19

Reflect...

- ❑ Which regularization technique is commonly applied to prevent exploding gradients during training?
 - ❖ A) Dropout
 - ❖ B) Batch Normalization
 - ❖ C) L2 Regularization
 - ❖ D) Data Augmentation
- ❑ Answer: B) Batch Normalization
- ❑ What is the role of early stopping as a form of regularization?
- ❑ Answer Choices:
 - ❖ A) To speed up the training process
 - ❖ B) To prevent the model from fitting the training data too closely
 - ❖ C) To add noise to the input data
 - ❖ D) To stop the training process when the model performance on a validation set plateaus or degrades
- ❑ Answer: D) To stop the training process when the model performance on a validation set plateaus or degrades

11/14/2025

pra-sâmi

- ❑ Which regularization method penalizes the squared values of the weights in the network?
 - ❖ A) Dropout
 - ❖ B) L1 Regularization
 - ❖ C) L2 Regularization
 - ❖ D) Batch Normalization
- ❑ Answer: C) L2 Regularization
- ❑ What is the trade-off associated with increasing the strength of regularization in a deep neural network?
 - ❖ A) Increased risk of overfitting
 - ❖ B) Increased risk of underfitting
 - ❖ C) Slower convergence during training
 - ❖ D) Improved model generalization
- ❑ Answer: B) Increased risk of underfitting

20

Reflect...


- ❑ In the context of regularization, what does the term "lambda" typically represent?
 - ❖ A) Learning rate
 - ❖ B) Regularization strength
 - ❖ C) Number of hidden layers
 - ❖ D) Batch size
- ❑ Answer: B) Regularization strength
- ❑ Which regularization technique is particularly useful for handling sequences and time-series data in deep learning?
 - ❖ A) L1 Regularization
 - ❖ B) Data Augmentation
 - ❖ C) Recurrent Dropout
 - ❖ D) Batch Normalization
- ❑ Correct Answer: C) Recurrent Dropout

11/14/2025

pra-sâmi

- ❑ What is the purpose of data augmentation as a regularization technique?
 - ❖ A) To add noise to the input data
 - ❖ B) To increase the model's capacity
 - ❖ C) To generate more training samples by applying random transformations to the existing data
 - ❖ D) To decrease the learning rate during training
- ❑ Answer: C) To generate more training samples by applying random transformations to the existing data

21 **Next Session...**



- Vanishing Gradients
- Exploding Gradients
- Gradient Check

11/14/2025 **pra-samí**

22



11/14/2025 **pra-samí**