REGULARIZATIONS

Introduction to ML, DL, AI and OpenVino

Session 11

Pramod Sharma
pramod.sharma@prasami.com

2

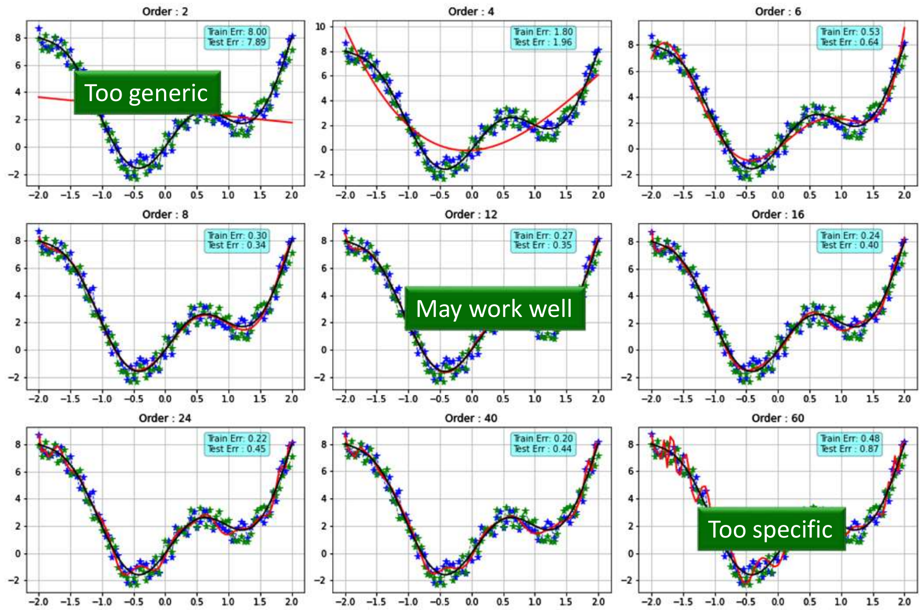
Agenda

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

pra-sâmi

3

Under-fitting vs. Over-fitting

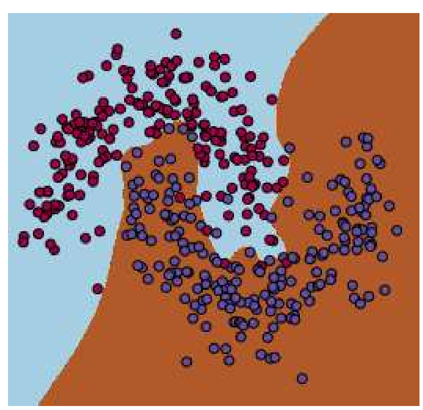


pra-sâmi

4

Regularization

- Regularization helps in avoiding over fitting by penalizing the coefficients
- In deep learning, it actually penalizes the weight matrices of the nodes
- Different Regularization Techniques in Deep Learning
  - L1 regularization
  - L2 regularizations
  - Dropouts
  - Early stopping
  - Data Augmentation



Most Libraries have tunable hyper-parameters!

pra-sâmi

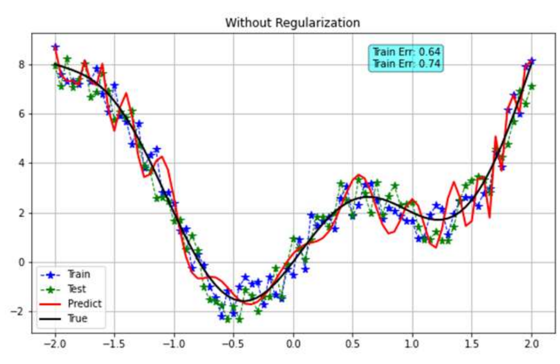
5

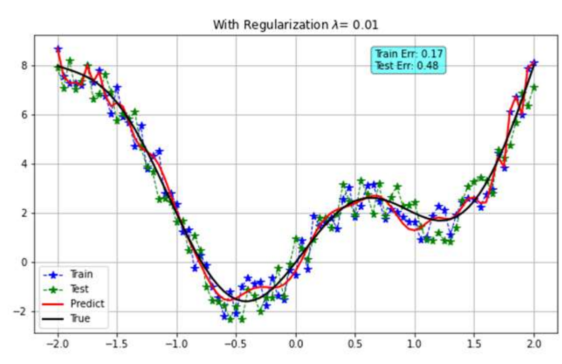
Weights vs. Bias

- 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 require less data to fit accurately than the weights.
- Fitting the weight will requires observing both layer in a variety of conditions.
- Each bias controls only a single layer.
- This means that we do not induce too much variance by leaving the biases un-regularized.
- Also, regularizing the bias parameters can introduce a significant amount of under fitting.

6

Effect of L2 Regularization





Theory – Logistic Regression – L1 & L2

- Idea is to minimize Cost Function
  - J (W, b) = $\frac{1}{m}$ * Σ ℓ (a, y)
  - = - $\frac{1}{m}$ {y * log(a) + (1-y) * log(1-a)}

- A term is added to Cost function $\frac{\lambda}{2*m} . \|W\|_2^2$
- J (W, b) = $\frac{1}{m}$ * Σ ℓ (a, y) + $\frac{\lambda}{2*m} . \|W\|_2^2$

pra-sâmi

Theory – Logistic Regression – L1 & L2

- J (W, b) = $\frac{1}{m}$ * Σ ℓ (a, y) + $\frac{\lambda}{2*m} . \|W\|_2^2 + \cancel{\frac{\lambda}{2*m} . b^2}$
  - This is referred as L2 regularization
  - **Regularization hyperparameter λ**: It is another parameter we tune…

- $\|W\|_2^2 = \sum_{j=1}^{n} w_j^2 = W^T$ .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.

pra-sâmi

9 Theory – Logistic Regression – L1 & L2

- Sometimes L1 too is used
- J (W, b) = $\frac{1}{m}$ * (Σ ℓ (a, y)) + $\frac{\lambda}{2*m} . \|W\|_1$
- Differentation of $\frac{\lambda}{2*m} . \|W\|_1 = \frac{\lambda}{2*m}$ sign(W)
  - Will be infinitely small and will have insignificant impact on gradient descent

10 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 ($W^{[1]}$, $b^{[1]}$, $W^{[2]}$, $b^{[2]}$, $W^{[3]}$, $b^{[3]}$ ...)
  - J (W, b) = $\frac{1}{m}$ * (Σ ℓ (a, y)) + $\frac{\lambda}{2*m} . \sum_{l=1}^{L} \sum (w_{i,j})^2$
  - J (W, b) = $\frac{1}{m}$ * Σ{y * log(a) + (1-y) * log(1-a)} + $\frac{\lambda}{2*m} . \sum_{l=1}^{L} \|W^{[l]}\|^2$
  - Where $\|W^{[l]}\|^2_F = \sum_{i=1}^{n[l-1]} \sum_{j=1}^{n[l]} (w_{ij}^{[l]})^2$
    - W is ($n^{[l-1]}$, $n^{[l]}$) dimensional matrix
- It is called $Frobenius\ norm$ of a matrix
- Also the Euclidean norm defined as the square root of the sum of the absolute squares of its elements

11

Frobenius Norm of a Vector

- $\|A\|_F = \sqrt{\Sigma(a_{ij})^2}$

i.e.

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

1/4/2024

pra-sâmi

12

Updates to weights

- Earlier
  - $\partial W^{[l]}$ = X . ∂z
  - And $W^{[l]}$ = $W^{[l]}$ - α . $\partial W^{[l]}$
- For Regularization we add an extra term at the end
  - $\partial W^{[l]}$ = X . ∂z + $\frac{\lambda}{m}$. $W^{[l]}$

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

  - $W^{[l]}$ = $W^{[l]}$ - α .[X . ∂z + $\frac{\lambda}{m}$. $W^{[l]}$]
  - $W^{[l]}$ = $(1 - \frac{\alpha . \lambda}{m})$. $W^{[l]}$ - α . X . ∂z

Weight Decay

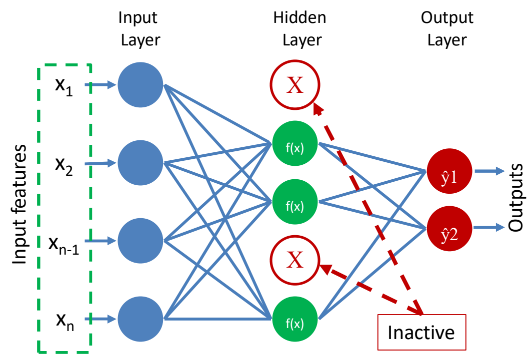
1/4/2024

pra-sâmi

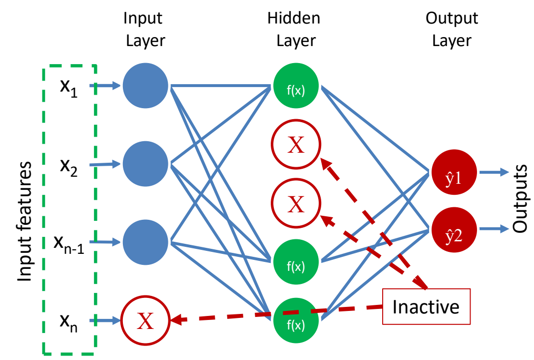
13

Regularization : Dropout

- Iteration 1



- Iteration 2

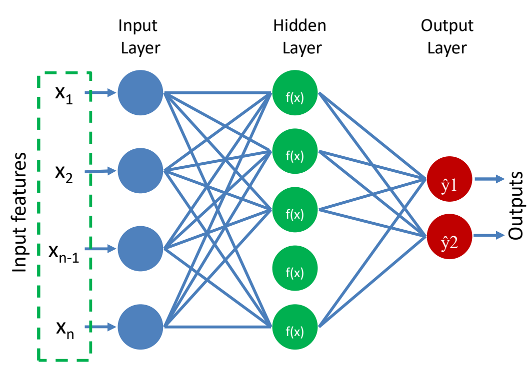


1/4/2024

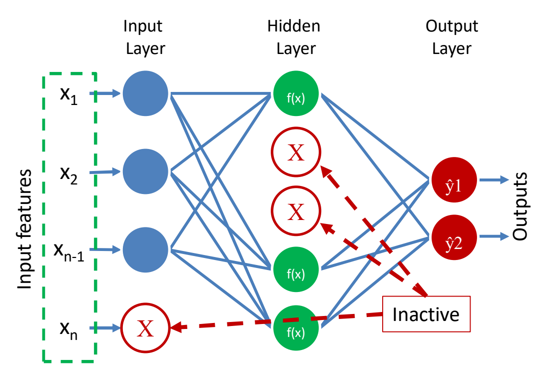
14

Regularization : Dropout

- Original



- With Dropout



1/4/2024

15

Regularization : Early Stopping

- How long to train the model?
- Duration of training ➔ under – fit or over – fit
- Train the model to the point where it 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

1/4/2024

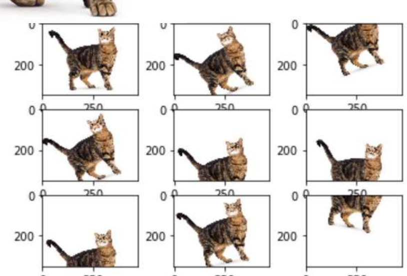
pra-sâmi

16

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



1/4/2024

pra-sâmi

17

1/4/2024

pra-sâmi