TNM112 – Deep Learning for Media Technology

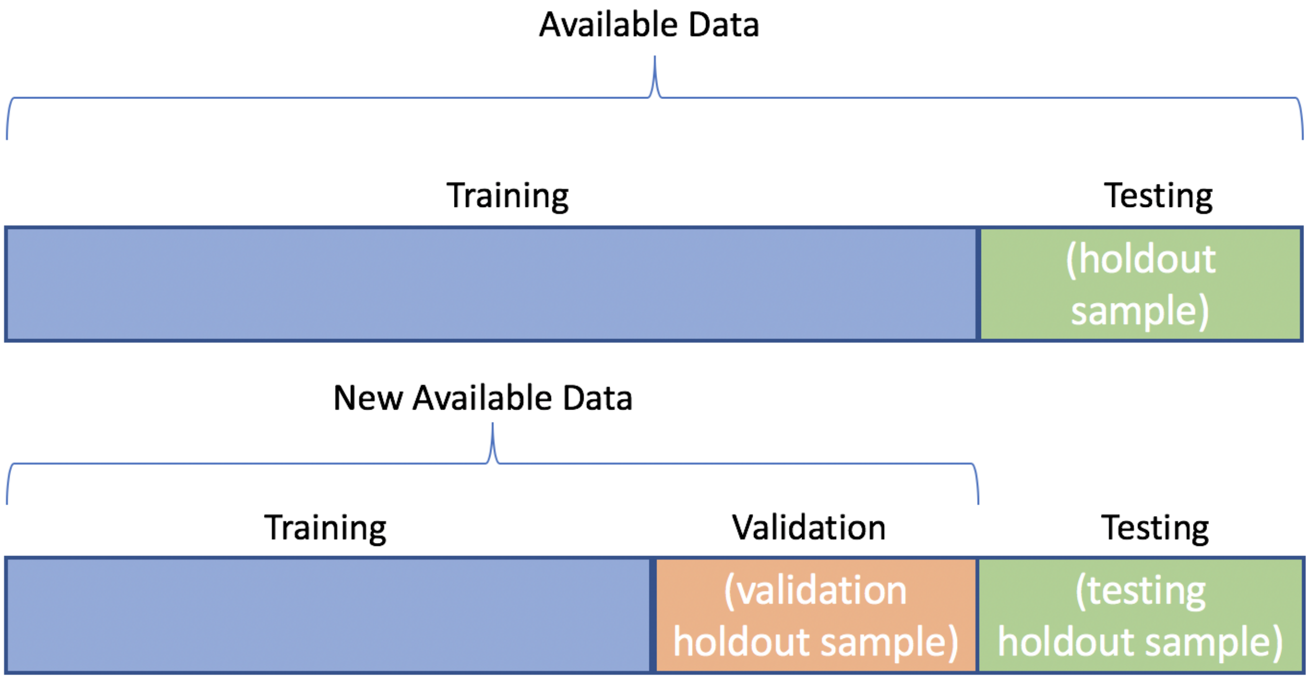
Nithesh Chandher Karthikeyan

LINKÖPING UNIVERSITY

Overfitting

- When the machine learning model performs well in the training set but fails on the test set.
- How to measure Overfitting?
  - Using an unseen validation set
  - If the difference in validation and training performance is high
- High Variance (error due to variations in f(x))

Data Setup

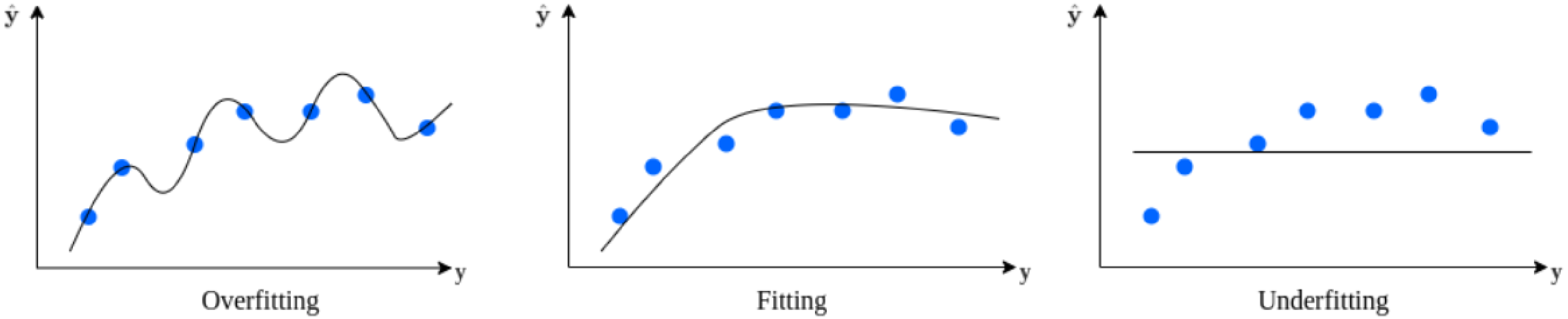


Available Data
Training
Testing
(holdout sample)
New Available Data
Training
Validation
Testing
(validation holdout sample)
(testing holdout sample)

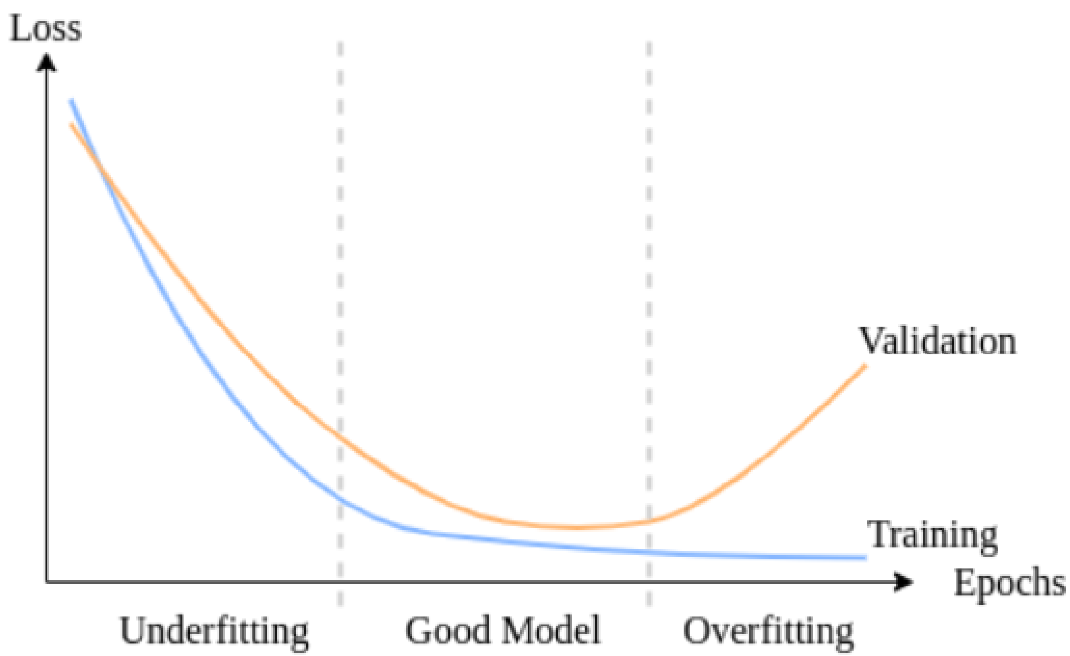
Underfitting

- When the machine learning model is too simple to capture the relationship between X and Y.
- Reasons:
  - Model is too simple
  - Size of the training data is not enough
- How to overcome underfitting?
  - Deeper network
  - More training epochs

Underfit vs Overfit vs Good fit



Underfit vs Overfit vs Good fit



How to avoid Overfitting?

Some methods to avoid Overfitting:

- Regularization
- Early Stopping
- Dropout
- Data Augmentation
- Pruning, Ensembling, etc..,

Regularization

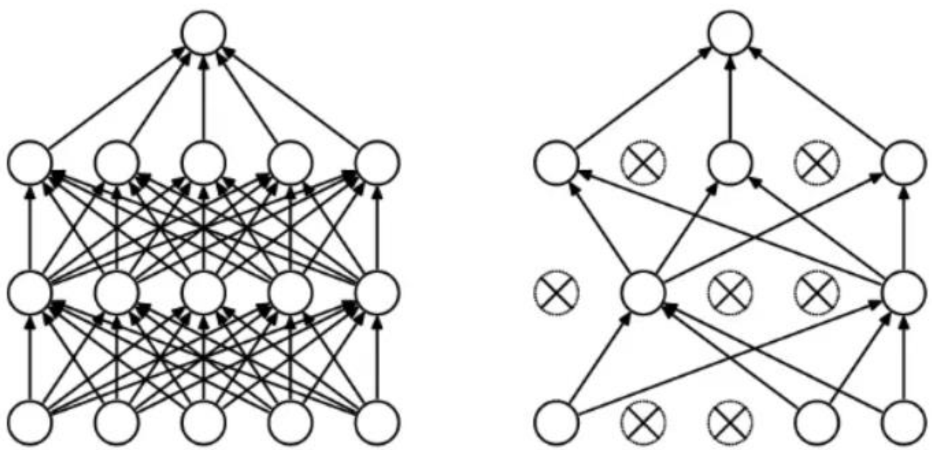
- Prevents overfitting by penalizing the model parameters
- Adds penalty term to the loss function
- Two types: Lasso and Ridge Regularization
- Keras: kernel_regularizer = regularizers.L2(0.01)

Early Stopping

- Stop the model training before it overfits
- Two ways:
  - Complete the entire training and choose the parameters when the validation loss is the lowest
  - Define a stopping criteria
- Keras: keras.callbacks.EarlyStopping(patience=3)

Dropout

- Randomly turns off neurons in the network
- Makes the network more robust
- Keras: keras.layers.Dropout(0.3)



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

Data Augmentation

- Artificially increasing the training data
- For Image datasets, make geometric and color space transformations.
- Geometric Transformations: flipping, rotation, cropping, translation, zoom etc.,
- Color Space Transformations: contrast, brightness, blurring etc.,
- Eg: tf.keras.layers.RandomRotation(), RandomFlip(), RandomCrop(), RandomContrast(), etc.,

Data Augmentation (Eg: STL10 Data)

Evaluation Metrics - Accuracy

- Classification Accuracy is the standard metric
- Ratio of correct predictions to the total number of predictions made
- Consider a classifier that predicted 91 cases correctly out of 100 test cases, what is the accuracy of the model?
- Ans: 91%

Accuracy can be misleading

Consider a test set that has 90 images of cat and 10 images of dog.

- Classifier A was trained on CatvsDog dataset, achieves 84% accuracy.
- Classifier B is a fake classifier, that always predicts any image as Cat.

Which classifier got the better accuracy?

Evaluation Metrics

Given a classifier that predicts two classes (Positive and Negative).

- True Positive (TP): Number of data correctly predicted as Positive
- True Negative (TN): Number of data correctly predicted as Negative
- False Positive (FP): Number of data predicted as Positive but belongs to Negative class
- False Negative (FN): Number of data predicted as Negative but belongs to Positive Class

Evaluation Metrics

- Accuracy = (TP + TN)/ (TP + TN + FP + FN)
- Precision = TP / TP + FP
- Recall = TP / TP + FN
- F1 Score =

  2 x (Precision*Recall) / (Precision + Recall)

Feedback

Kindly send us your feedback