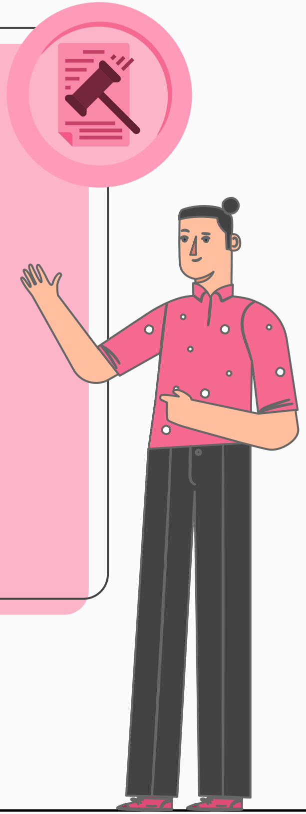
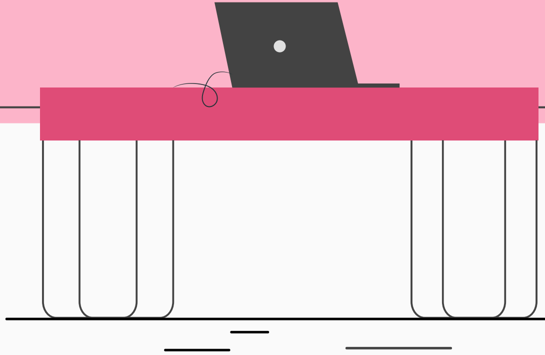
REGULARIZATION





MACHINE LEARNING MODEL TRAINING METRICS: GENERALIZATION

In machine learning algorithm we **learn** from known data and **generalize** to unseen data.

Goal of a good model is maximizing classification accuracy, minimizing *loss or error*.

A model that has high classification accuracy on seen data but poor accuracy on unseen data is a model that has poor generalizability.

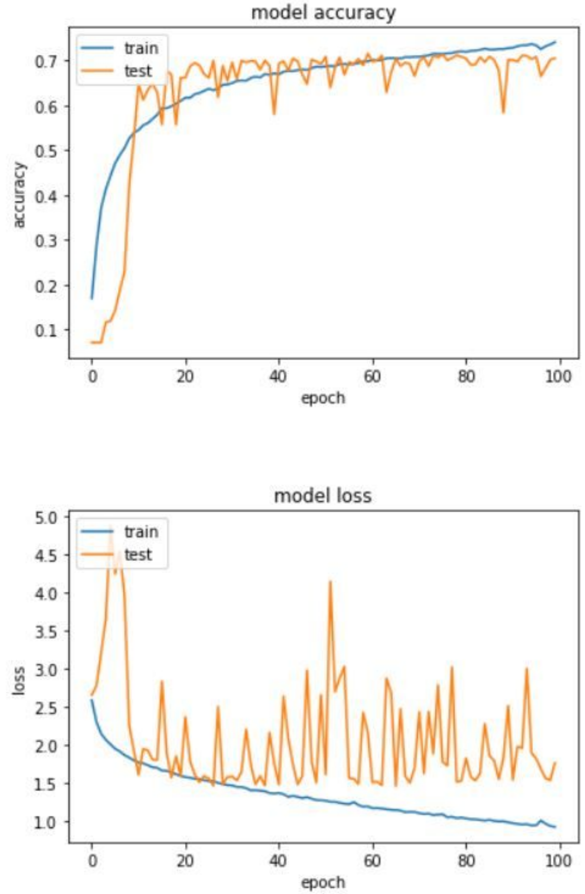
The generalization error is the estimated error on a new input from the test set.

- Ideally, we want the generalization error to be small.
- We want to make the gap between training and test error small.

MACHINE LEARNING MODEL TRAINING METRICS: LEARNING CURVES

Learning curves help to visualize how the model is “learning” through the training.

- **Training curve** tells us how well the model is learning.
- **Validation curve** tells us how well the model is generalizing.
- *Optimization learning curves* calculate the metric by which the parameters of the model are being optimized, e.g. loss.
- *Performance learning curves* calculate the metric by which the model will be evaluated and selected, e.g. accuracy.
- **Overfitting & underfitting** can be spotted through the learning curves.



SureStart

OVERFITTING AND UNDERFITTING

A model might have learned the training set perfectly, but perform poorly during prediction or be unable to generalize to new(unseen) samples.

Two kinds of model fitting issues might occur: overfitting and underfitting.

Overfitting is not able to narrow the gap between the training and the test error.

- It occurs when a neural network is trained to the point that it begins to memorize rather than generalize.
- A model can overfit if the model doesn't have sufficient training examples, then a small group of neurons might become responsible for doing most of the processing and other neurons become redundant.
- If the neural network is too complex, it will start memorizing the training data instead of having a general understanding of the data, hence causing overfitting.

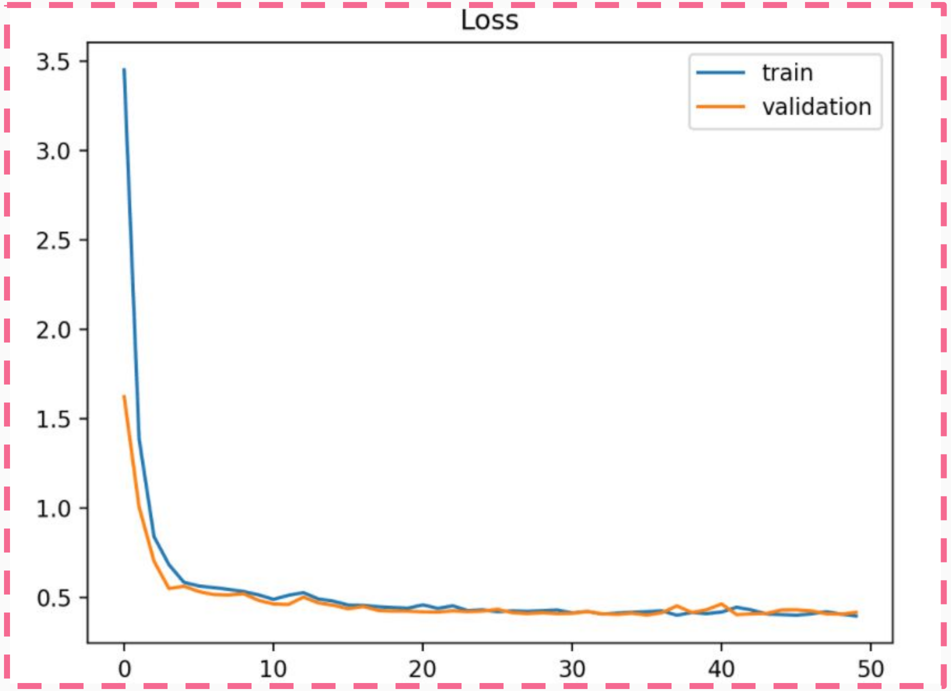
Underfitting is not able to have sufficiently low error on the training set.

- Underfitting occurs when the model is not able to obtain a sufficiently low error value on the training set.

GOOD FIT LEARNING CURVES

A plot of learning curves shows a good fit if:

- The plot of training loss decreases to a point of stability.
- The plot of validation loss decreases to a point of stability and has a small gap with the training loss.

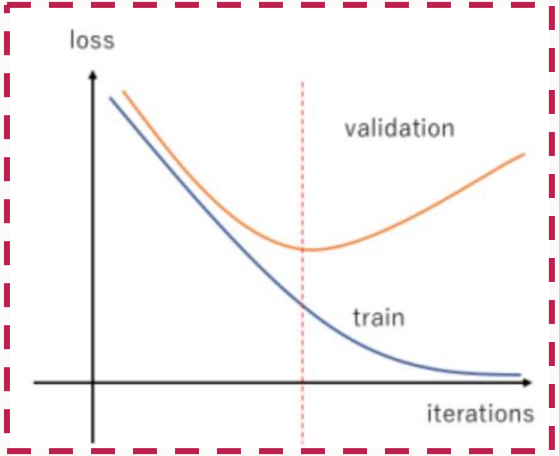


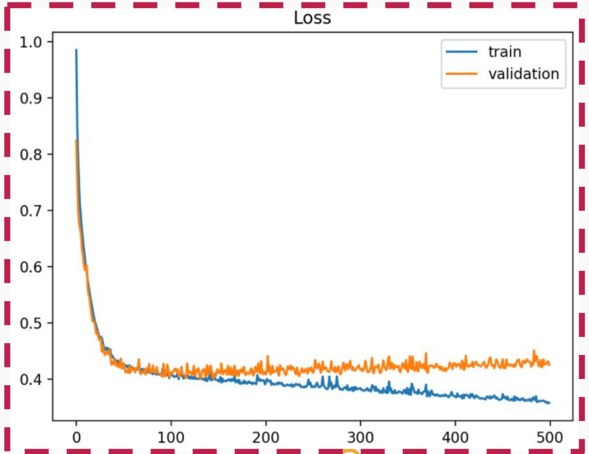
SureStart

OVERFIT MODEL

A plot of learning curves shows overfitting if:

- The plot of training loss continues to decrease with experience.
- The plot of validation loss decreases to a point and begins increasing again.





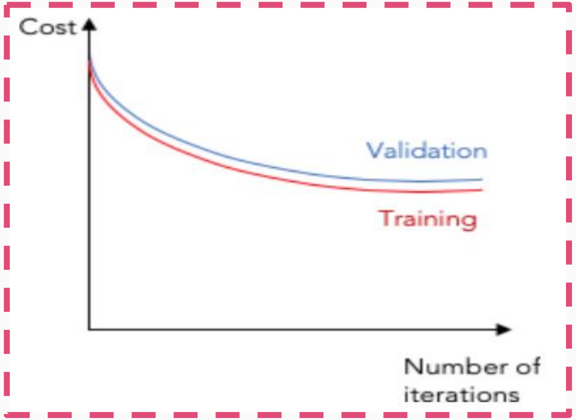
SureStart

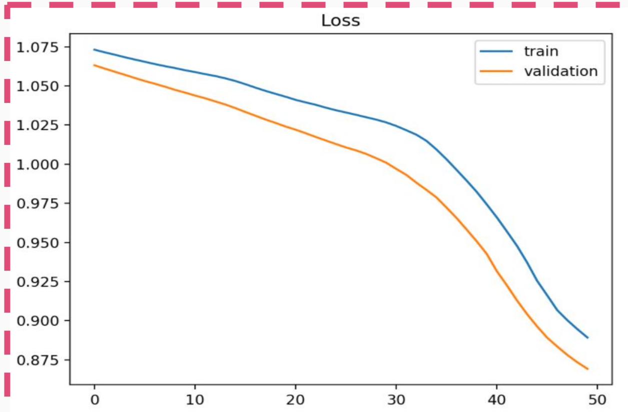
UNDERFIT MODEL

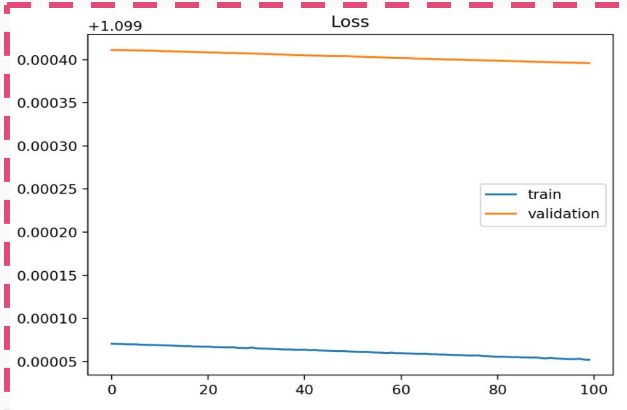
- A flat line or noisy values of relatively high loss indicating that the model was unable to learn the training dataset at all.

<u>A plot of learning curves shows underfitting if:</u>

- The cost (loss) function is high and doesn't decrease with the number of iterations, both for the validation and training curves
- The training loss remains flat regardless of training.
- The training loss continues to decrease until the end of training.



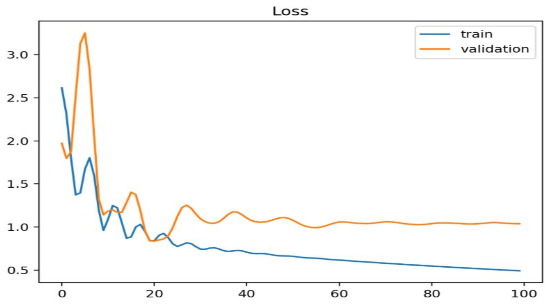


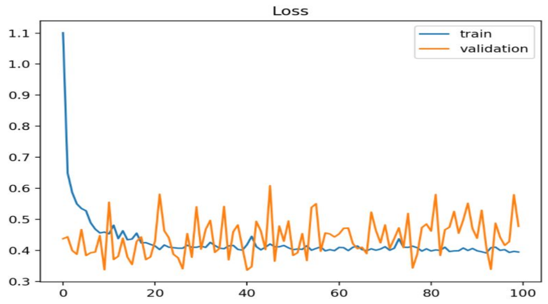


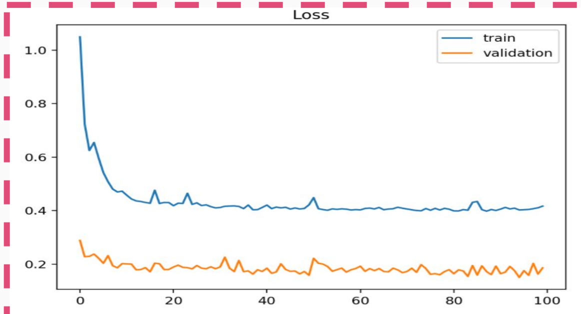
SureStart

UNREPRESENTATIVE VALIDATION DATASET

- This may occur if the validation dataset has too few examples as compared to the training dataset.
- The validation dataset does not provide sufficient information to evaluate the ability of the model to generalize.
- As we can see, the training curve looks ok, but the validation function moves noisily around the training curve.







SureStart

REDUCING OVERFITTING

1. Increase more data in the dataset
2. Image augmentation (e.g., jittering, noise injection, etc)
3. Reduce the model's complexity. (e.g., reducing the number of parameters)
4. Applying Regularization
    a. L2 and L1 regularization
    b. Dropout layer
    c. Early stopping

L2 AND L1 REGULARIZATION

L2 REGULARIZER

encourages the weight values towards zero (but not exactly zero).

$$cost\,function = loss + \frac{\lambda}{2m} + \Sigma||w||^2$$

L1 REGULARIZER

encourages the weight values to be zero.

$$cost\,function = loss + \frac{\lambda}{2m} + \Sigma||w||$$

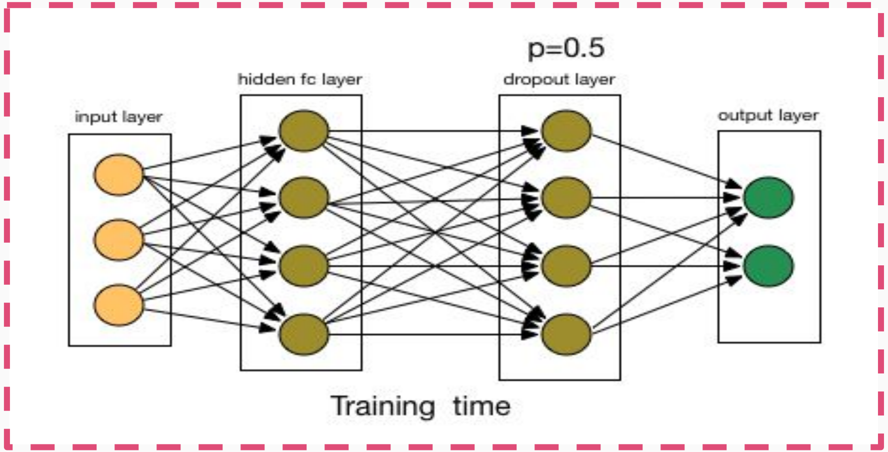
Here λ is the regularization parameter that we can alter and see the affect in the model.

- L2 regularization we add a component that will penalize large weights.
- Large weights will be driven down in order to minimize the cost function.
- A less complex model will fit to the data, effectively reducing overfitting.

DROP OUT

At every iteration, it randomly selects some nodes and removes them along with all of their incoming and outgoing connections.

- Randomly shutdown a subset of units in training. It is a sparse representation, reduced weight parameter and complexity.
- It is a different net each time, but all nets share the parameters.
- Dropout forces a neural network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons.



Drop out is implemented with a probability hyperparameter *p* (e.g., 0.5, 50% nodes are dropped out)

SureStart

EARLY STOPPING

Evaluate model's performance against a reserved validation set (a given set of operations or a given number of epochs).

This function halts the training process when the model stops improving its accuracy and restores the best weights after stopping the training.

Some important parameters of the early stopping callback:

1. monitor: quantity to be monitored. by default, it is validation loss
2. min_delta: minimum change in the monitored quantity to qualify as improvement
3. patience: number of epochs with no improvement after which training will be stopped
4. mode: one of {"auto", "min", "max"}. if it's a minimization problem, we maximize accuracy and minimize loss; else we minimize accuracy and maximize loss
5. restore_best_weights: whether to use best models weight or the last epoch weight