Bias Variance Tradeoff

Learning Rate Schedulers

The learning rate schedulers are algorithms that allow you to control your model's learning rate according to some schedule.

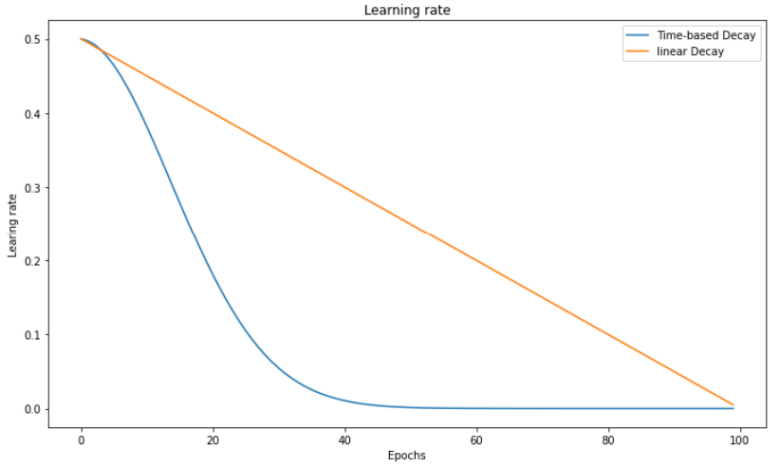
$$w^{+} = w - \eta * \frac{\partial Etotal}{\partial w}$$

Difficult to set the learning rate value

Time based decay

The learning rate is scheduled to decrease after each epoch of training

lr = lr*(1 / (1 + decay * iterations))

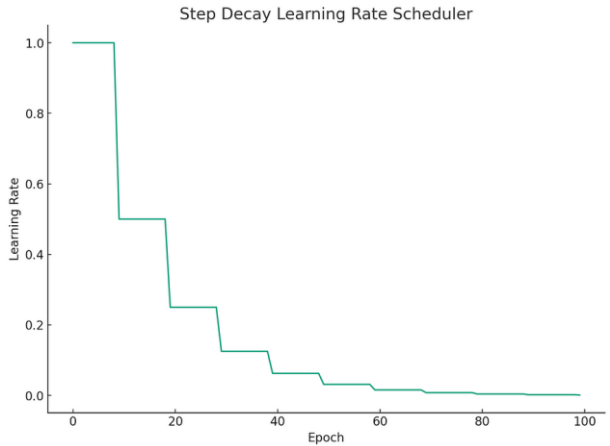


Step Decay

reduces the learning rate by a constant factor every few epochs

$$lr = lr_0 \cdot d^{(\text{floor}(1+\text{epoch})/s)}$$

Lr0= initial LR, d= decay rate, s=step size

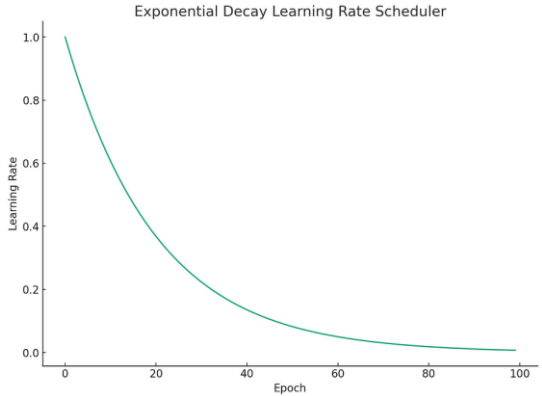


Exponential Decay

To make the decay more visible, a larger initial learning rate and a larger decay rate is used.

$$lr = lr_0 \cdot e^{-k \cdot \text{epoch}}$$

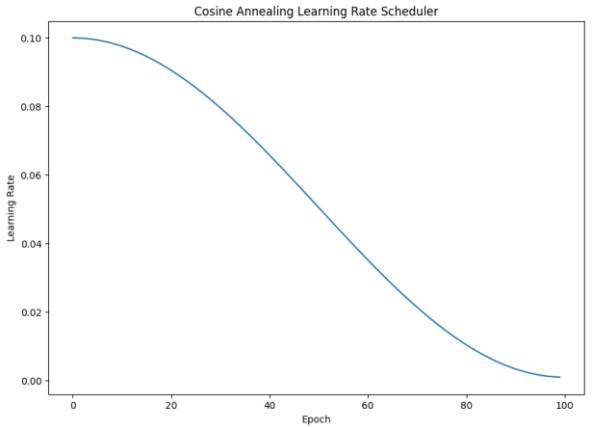
k=decay rate



Cosine Annealing

Cosine annealing reduces the learning rate using a cosine-based schedule

$$lr = lr_{\min} + 0.5 \cdot (lr_{\max} - lr_{\min}) \cdot \left(1 + \cos\left(\frac{\text{epoch}}{\text{max\_epochs}} \cdot \pi\right)\right)$$



Bias Variance Tradeoff

1. Bias error
2. Variance error
3. The noise

While the noise is the irreducible error that we cannot eliminate, the other two i.e. Bias and Variance are reducible errors that we can attempt to minimize as much as possible.

Bias Variance Tradeoff

Bias:

- Error between average model prediction and ground-truth
- Depicts the capacity of underlying model to predict the values

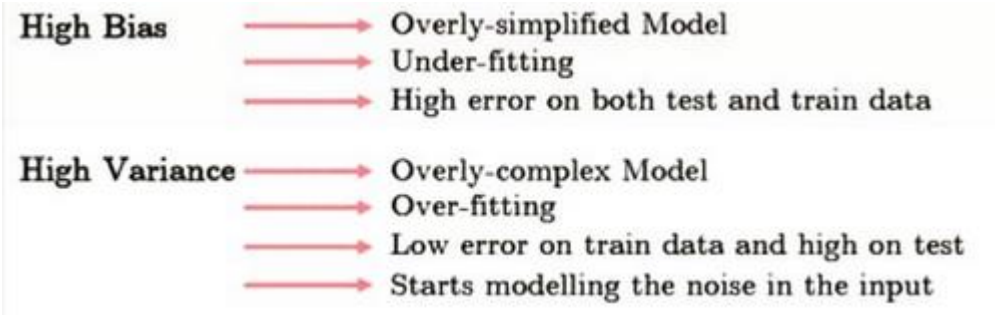
$$\text{Bias} = E[f`(x)-f(x)]$$

Variance:

- Average variability in the model prediction
- Depicts how much function can adjust the change

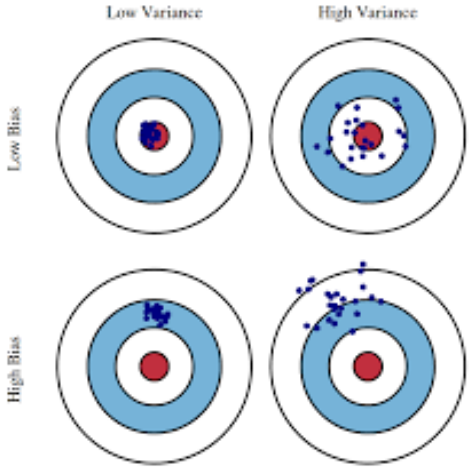
$$\text{Variance} = E[f`(x)-E|f(x)|^2]$$

Bias Variance Tradeoff



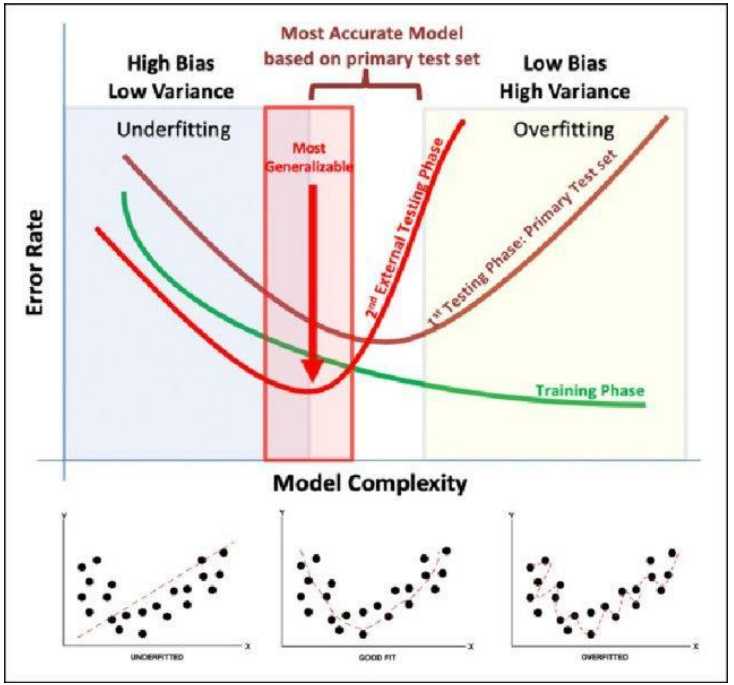
Bias Variance Tradeoff

- **Top left**: The points are close to the target, which indicates low bias. They are tightly knit as well, which indicates low variance.
- **Top right**: The points are far away from the target, which indicates high bias. But they are tightly knit, which indicates low variance.
- **Bottom left**: The points are close to the target, which indicates low bias. But they are not tightly knit, which indicates high variance.
- **Bottom right**: The points are far away from the target, which indicates high bias. And they are not tightly knit either, which indicates high variance.

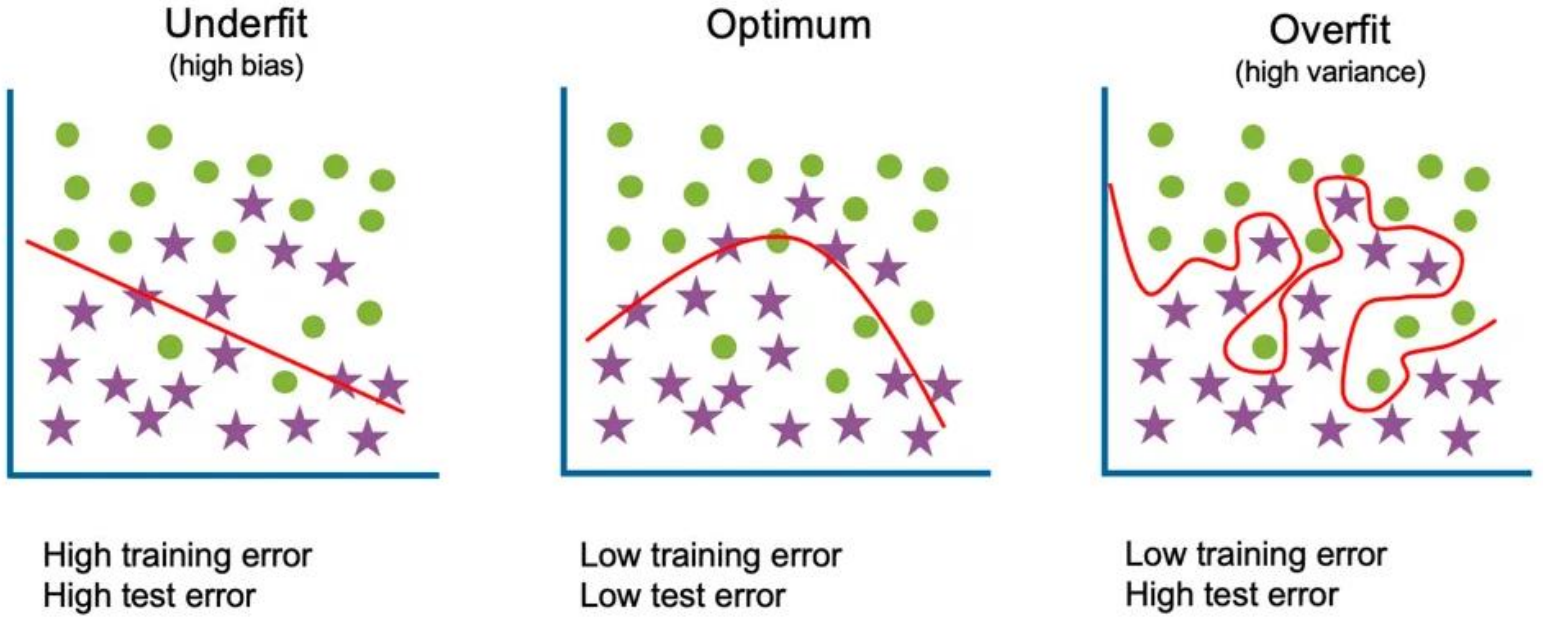


Bias Variance Tradeoff

- Increasing bias reduces variance and vice versa
- Error = bias^2 +variance irreducible error
- Compromise between bias and variance



Bias Variance Tradeoff



Bias Variance Tradeoff

- **Underfitting (High Bias):**

Issue: The model is too simplistic and fails to capture the underlying patterns in the data.

Solution: Increase model complexity, use more features, or choose a more sophisticated algorithm.

- **Overfitting (High Variance):**

Issue: The model is too complex and captures noise in the training data.

Solution: Reduce model complexity, use regularization, or collect more training data.

- **Balancing Bias and Variance:**

Goal: Find the right level of model complexity that minimizes the total error on new, unseen data.

Approach: Regularization, cross-validation, feature engineering, and ensemble methods can be used to strike a balance.

Strategies to Handle the Bias-Variance Tradeoff:

1. **Regularization:**

- **Purpose**: Regularization techniques, such as L1 or L2 regularization, penalize overly complex models by adding a regularization term to the loss function.
- **Effect:** This discourages the model from fitting the training data too closely and helps control variance.

2. **Cross-Validation:**

- **Purpose:** Cross-validation techniques, like k-fold cross-validation, help assess a model's performance on different subsets of the data.
- **Effect:** It provides a more reliable estimate of a model's ability to generalize to new data and helps diagnose whether a model has high bias or high variance.

Strategies to Handle the Bias-Variance Tradeoff:

3. **Feature Engineering:**

- **Purpose:** Carefully selecting and engineering features can impact a model's complexity.
- **Effect:** It helps create more relevant and informative features, reducing the risk of underfitting or overfitting.

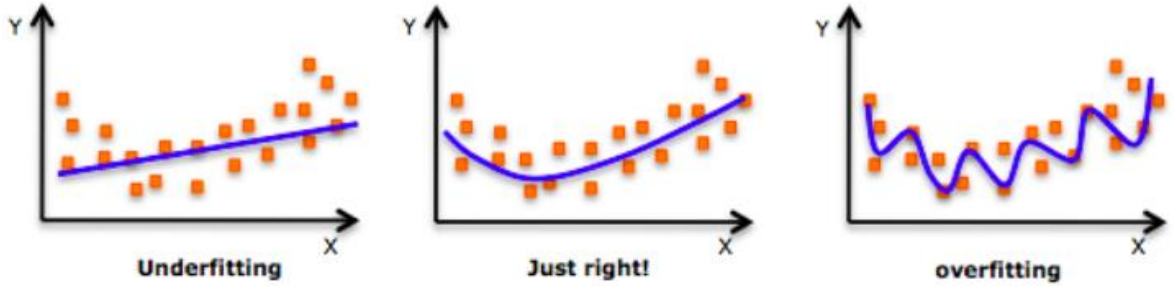
4. Ensemble Methods:

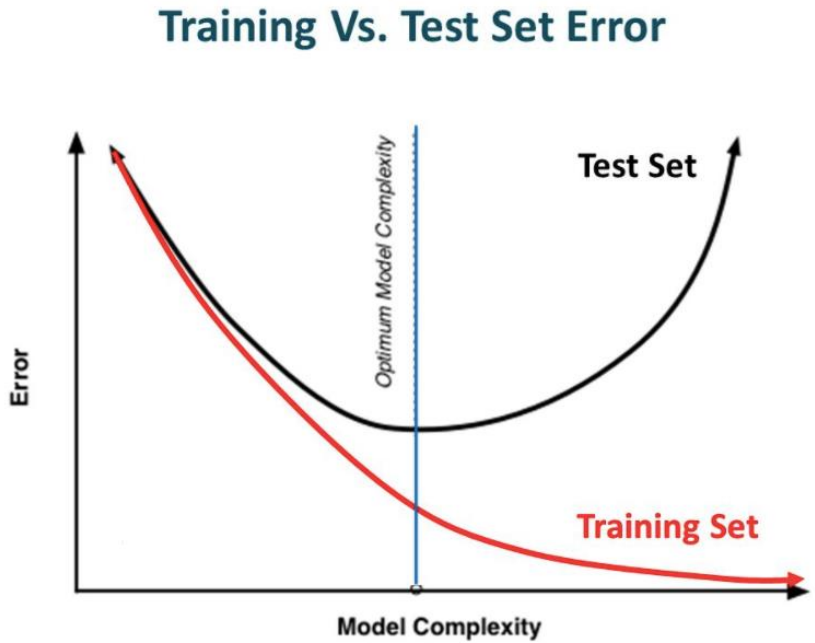
- **Purpose:** Ensemble methods, such as bagging and boosting, combine multiple models to improve overall performance.
- **Effect:** By averaging predictions or combining weak models, ensemble methods can often reduce variance and improve generalization.

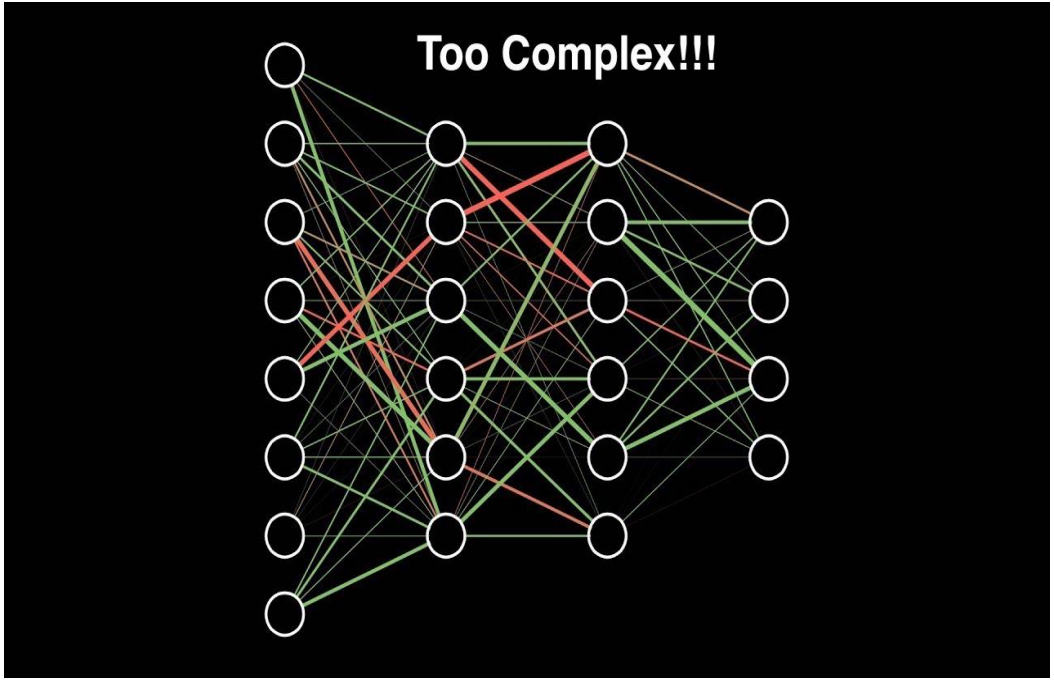
Regularisation

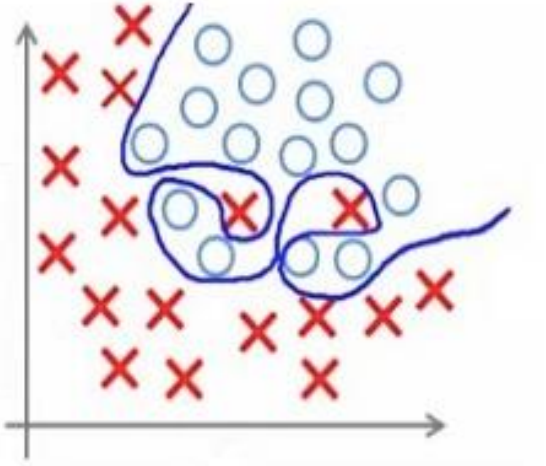
- Regularization is a technique used in machine learning and deep learning to prevent overfitting and improve the generalization performance of a model.
- It involves adding a penalty term to the loss function during training.
- Regularization in deep learning methods include L1 and L2 regularization, dropout, early stopping, and more.
- By applying regularization, models become more robust and better at making accurate predictions on unseen data.

Regularizations









Over-fitting

Parameter Norm Penalties

- limiting the capacity of models by adding a parameter norm penalty Ω(θ)
- Regularized objective function:

$$\tilde{J}(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) = J(\boldsymbol{\theta}; \boldsymbol{X}, \boldsymbol{y}) + \alpha\Omega(\boldsymbol{\theta})$$

- where α ∈ [0, ∞) is a hyperparameter. α =0 means no regularisation
- it will decrease both the original objective J on the training data and some measure of the size of the parameters

Parameter Norm Penalties

- Norm penalty Ω penalizes only weights at each layer and leaves biases unregularized
- Each bias controls only a single variable

LASSO

- L1 Regularisation or Lasso (Least absolute shrinkage and selection operator)
- the absolute value of the magnitude of coefficients or weights multiplied with a regularizer term is added to the loss or cost function

$$\underbrace{\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij} b_j\right)^2}_{Loss\ Term} + \underbrace{\lambda \cdot \sum_{j=1}^{p} |b_j|}_{Regularizer\ term}$$

y_i – *labels*
x_{ij} – *features*
b_j – *weights*
λ – *regularizer term (lambda)*
i – *no of training samples*

Ridge regularization

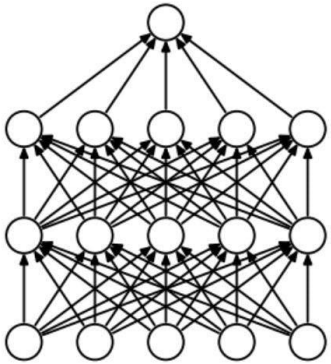
- Ridge regression or Tikhonov regularization or L2 regularisation.
- Parameter norm penalty
- Drives weights closer to the origin by adding a regularization term to the objective function

$$\underbrace{\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p} x_{ij} b_j\right)^2}_{Loss\ term} + \underbrace{\lambda \cdot \left(\sum_{j=1}^{p} b_j\right)^2}_{Regularizer\ term}$$

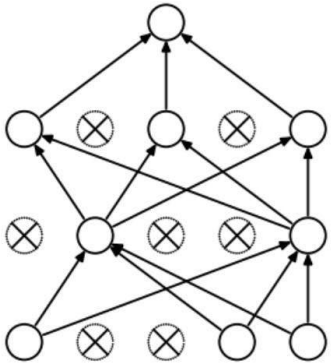
y_i – $labels$
x_{ij} – $features$
b_j – $weights$
λ – $regularizer\ term\ (lambda)$
i – $no\ of\ training\ samples$

Dropout

- Dropping out the nodes (input and hidden layer) in a neural network
- All the forward and backwards connections with a dropped node are temporarily removed, thus creating a new network architecture out of the parent network.
- The nodes are dropped by a dropout probability of p.

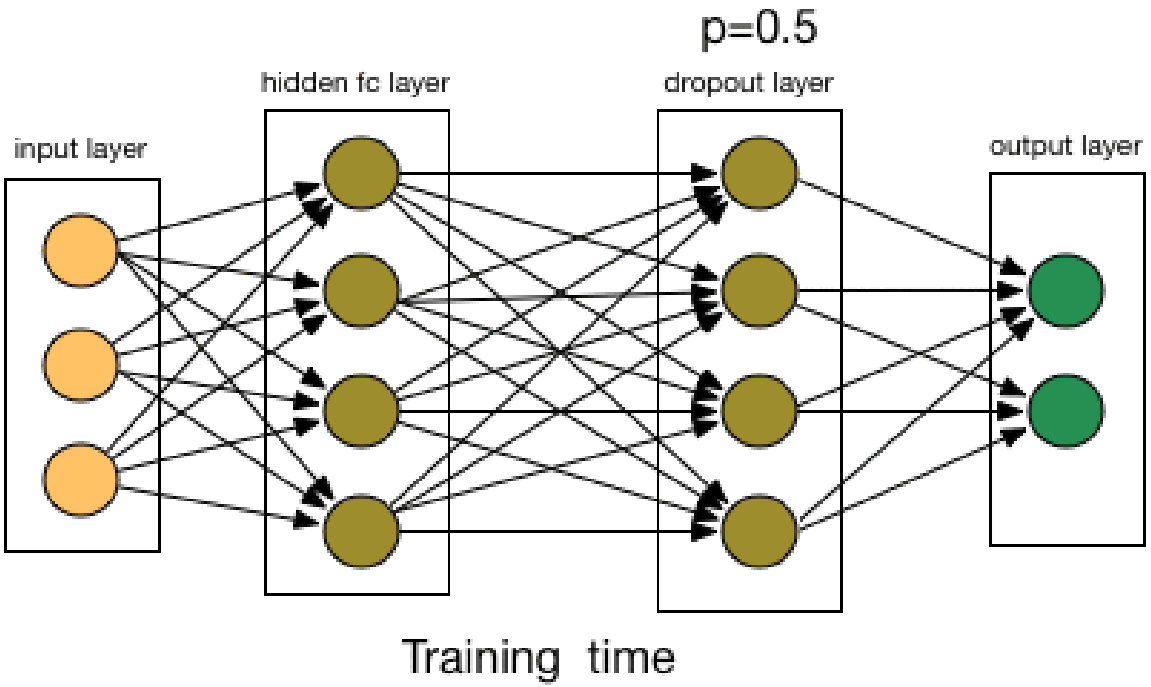


(a) Standard Neural Net



(b) After applying dropout.

Dropout


p=0.5
hidden fc layer
dropout layer
input layer
output layer
Training time

Dropout

x: {1, 2, 3, 4, 5} is given to a fully connected layer

Dropout layer p=0.2

20% of the nodes would be dropped, i.e. the x could become {1, 0, 3, 4, 5} or {1, 2, 0, 4, 5} and so on.

Criteria: Random

The greater the drop probability more sparse the model!

Data Augmentation

- The process of artificially generating new data from existing data, primarily to train new machine learning (ML) models
- Useful when the number of samples are less in training set

Data Augmentation

Methods: Depends on the type of the data there

1. Image Data:

- Geometric transformations: random flip, crop, rotate or translate
- Color space transformations: RGB to other spaces
- Kernel filters: sharpen or blur an image
- Random Erasing: parts of images are masked
- Mixing images: mix images with one another

Data Augmentation

2. Text

- Word/sentence shuffling
- Word replacement – replace words with synonyms
- Syntax-tree manipulation – paraphrase the sentence to be grammatically correct using the same words

3. Audio

- Noise injection
- Shifting
- Changing the speed of the tape

Early Stopping

Early stopping is a kind of cross-validation strategy where we keep one part of the training set as the validation set.

When we see that the performance on the validation set is getting worse, we immediately stop the training on the model.

In the above image, we will stop training at the dotted line since after that our model will start overfitting on the training data.

Early Stopping

Pros:

- Helps in reducing overfitting
- It improves generalisation
- It requires less amount of training data
- Takes less time compared to other regularisation models
- It is simple to implement

Cons:

- If the model stops too early, there might be risk of underfitting
- It may not be beneficial for all types of models
- If validation set is not chosen properly, it may not lead to the most optimal stopping

Cross Validation

- A technique used to test a model’s ability to predict unseen data, data not used to train the model
- Useful if we have limited data when our test set is not large enough

K-Fold Cross Validation

- Split training data into K equal parts
- Fit the model on k-1 parts and calculate test error using the fitted model on the kth part
- Repeat k times, using each data subset as the test set once. (usually k= 5~20)

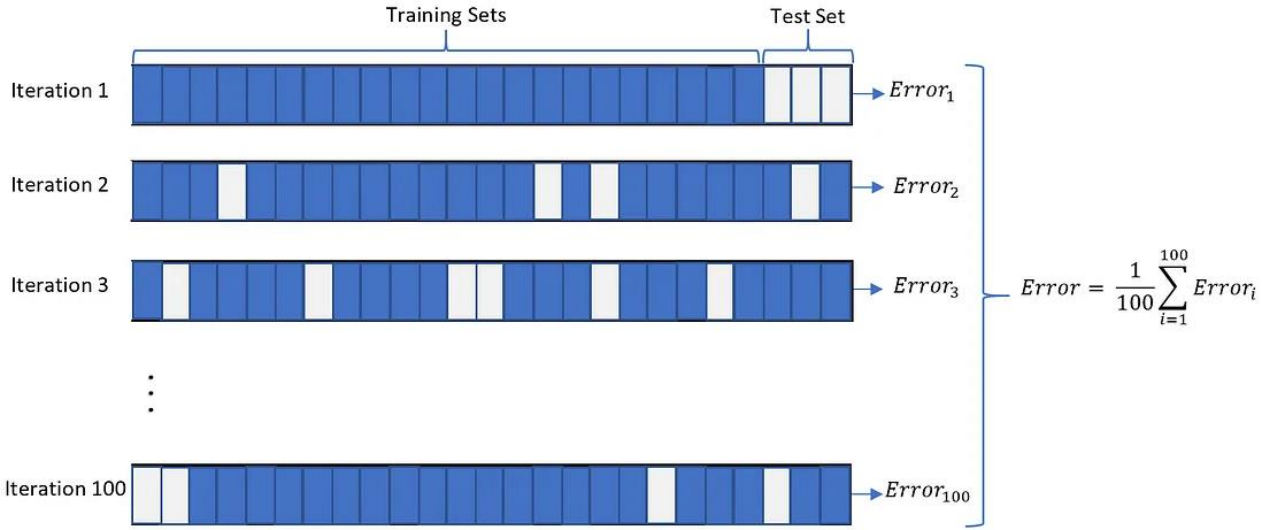
K-Fold Cross Validation

Error is averaged

Monte-Carlo Cross Validation

- creates multiple random splits of the dataset into training and validation data
- For each iteration, the train-test split percentage is different.
- Fit the model on train data set for that iteration and calculate test error using the fitted model on test data
- Result is averaged over splits
- Repeat many iterations (say 100 or 500 or even 1000 iterations)

Monte-Carlo Cross Validation



Ensemeble Learning

Set of classifiers ; criteria of voting

Improves learning by:

- By reducing the variance of weak learners
- By reducing the bias of weak learners,
- By improving the overall accuracy of strong learners.

Bagging

Bootstrap Aggregating is known as Bagging

Bootstrapping

Involves resampling subsets of data called bootstrap with replacement from an initial dataset. Each bootstrap dataset is used to train a weak learner.

Aggregating

The individual weak learners are trained independently from each other. Each learner makes independent predictions. The results of those predictions are aggregated at the end to get the overall prediction.

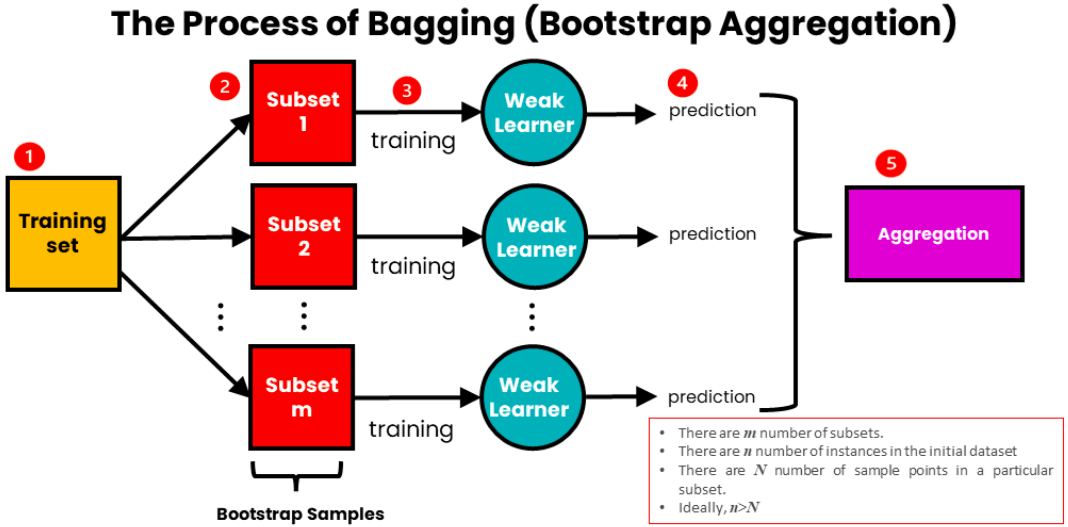
Bagging

Max Voting

A prediction from each model counts as a single ‘vote’. The most occurring ‘vote’ is chosen as the representative for the combined model.

Averaging

The resulting average is used as the overall prediction for the combined model.

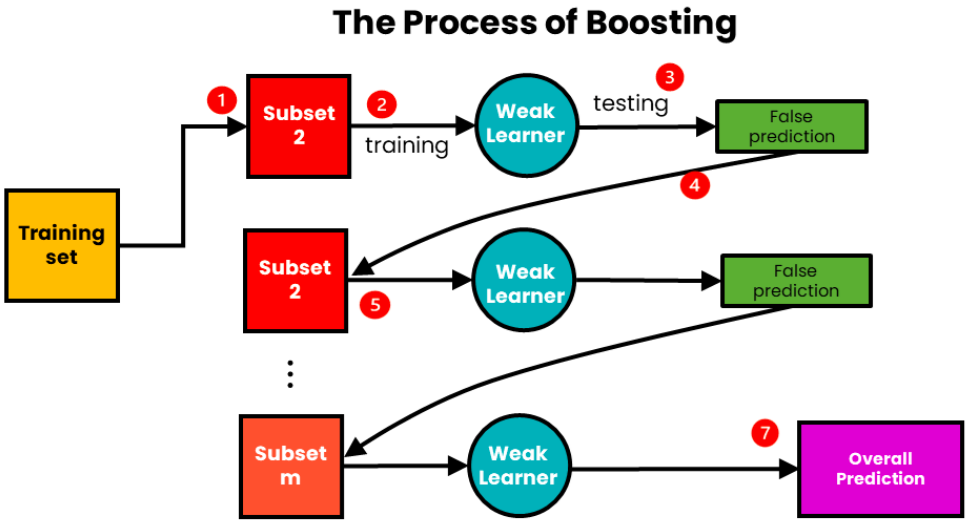


Steps of Bagging

- Create a m-number of subsets of data from the training set. Take a subset of N sample points from the initial dataset for each subset. Each subset is taken with replacement. This means that a specific data point can be sampled more than once.
- For each subset of data, train the corresponding weak learners independently. These models are homogeneous, meaning that they are of the same type.
- Each model makes a prediction.
- The predictions are aggregated into a single prediction. For this, either max voting or averaging is used.

Boosting

- Boosting aims to produce a model with a lower bias than that of the individual models
- sequentially training weak learners.



Steps of Boosting

- Sample m-number of subsets from an initial training dataset.
- Using the first subset, train the first weak learner.
- Test the trained weak learner using the training data. As a result of the testing, some data points will be incorrectly predicted.
- Each data point with the wrong prediction is sent into the second subset of data, and this subset is updated.
- Using this updated subset, train and test the second weak learner.
- Continue with the following subset until the total number of subsets is reached.
- The overall prediction has already been aggregated at each step, so there is no need to calculate it.