Deep Learning and Convolutional Neural Network (42028)

Convolutional Neural Network (CNN) - 2

Dataset preparation

- In case of small dataset (Range : 100 - <100k)

  - Train set: 60%
  - Validation set: 20%
  - Test set: 20%

  Or,

  - Train set: 70%
  - Test set: 30%

Popular dataset spit choice in non-DL era!
Or Small Data era!

Source and reference: https://cs230-stanford.github.io/train-dev-test-split.html

Dataset preparation

- In case of Large dataset (Range : 500K - 1M+)

Example: Total data sample : 1M+

Train: 98% !

Validation: 10,000 samples

Test: 10,000 samples

Popular dataset spit choice in DL era!
Or BIG Data era!

Source and reference: https://cs230-stanford.github.io/train-dev-test-split.html

Dataset preparation

Train, validation and test set distribution:

Rule of Thumb:

Validation and Test set should come from the same distribution

Source and reference: https://cs230-stanford.github.io/train-dev-test-split.html

Bias and Variance

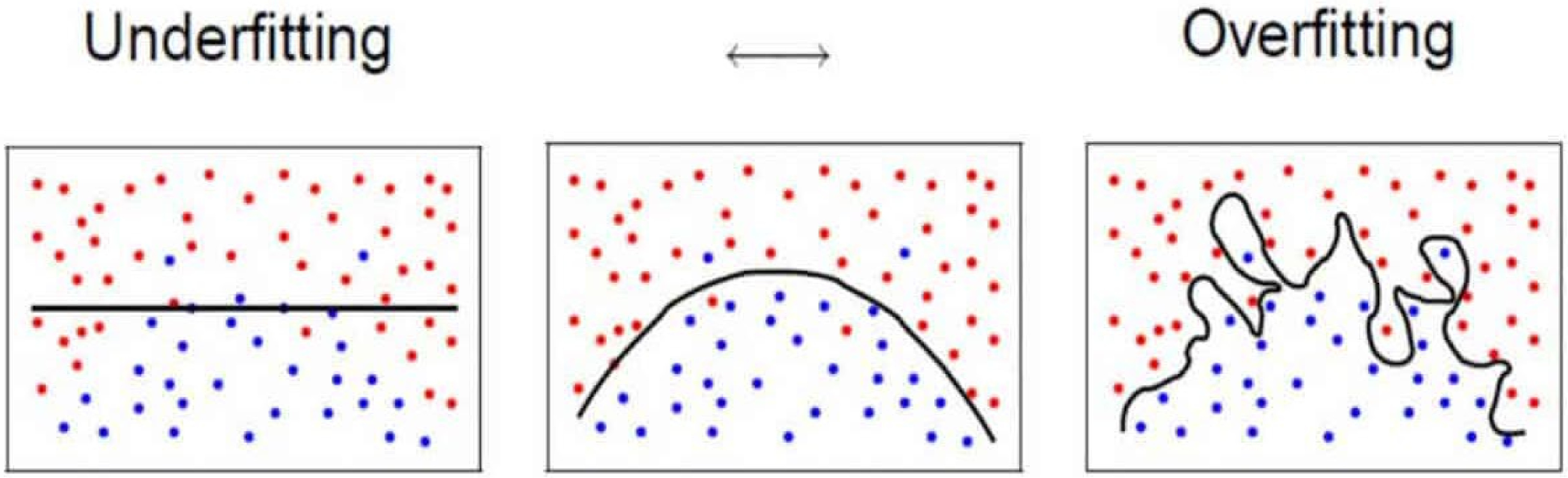
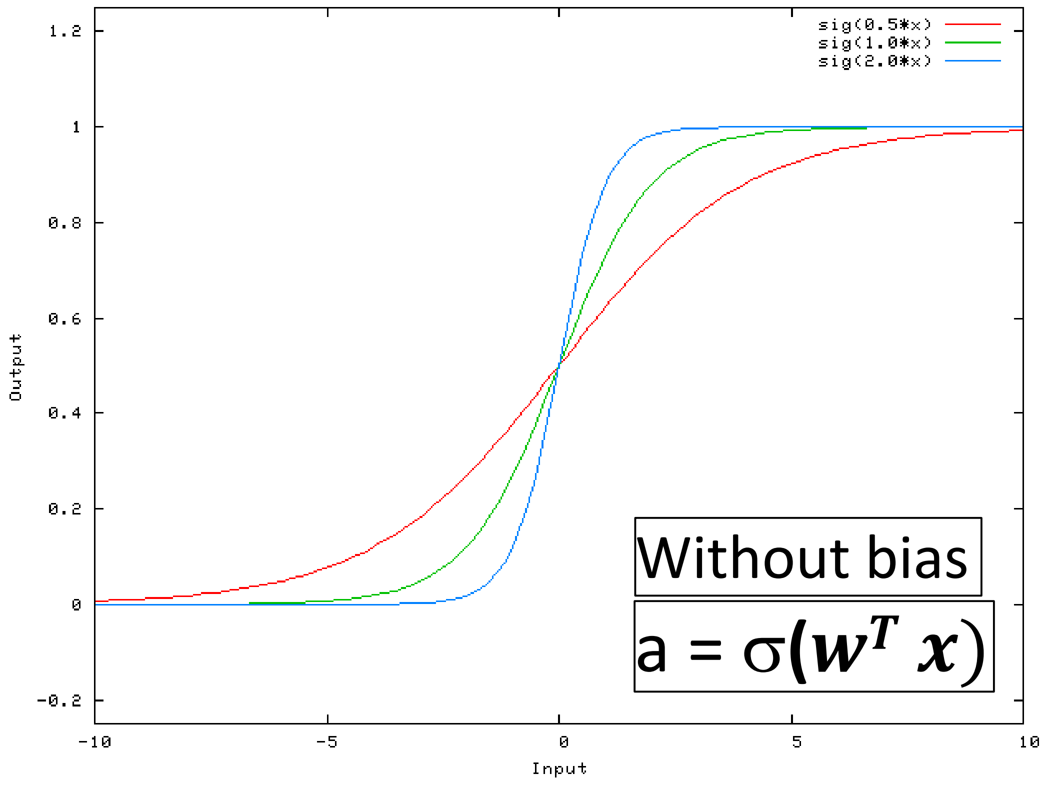
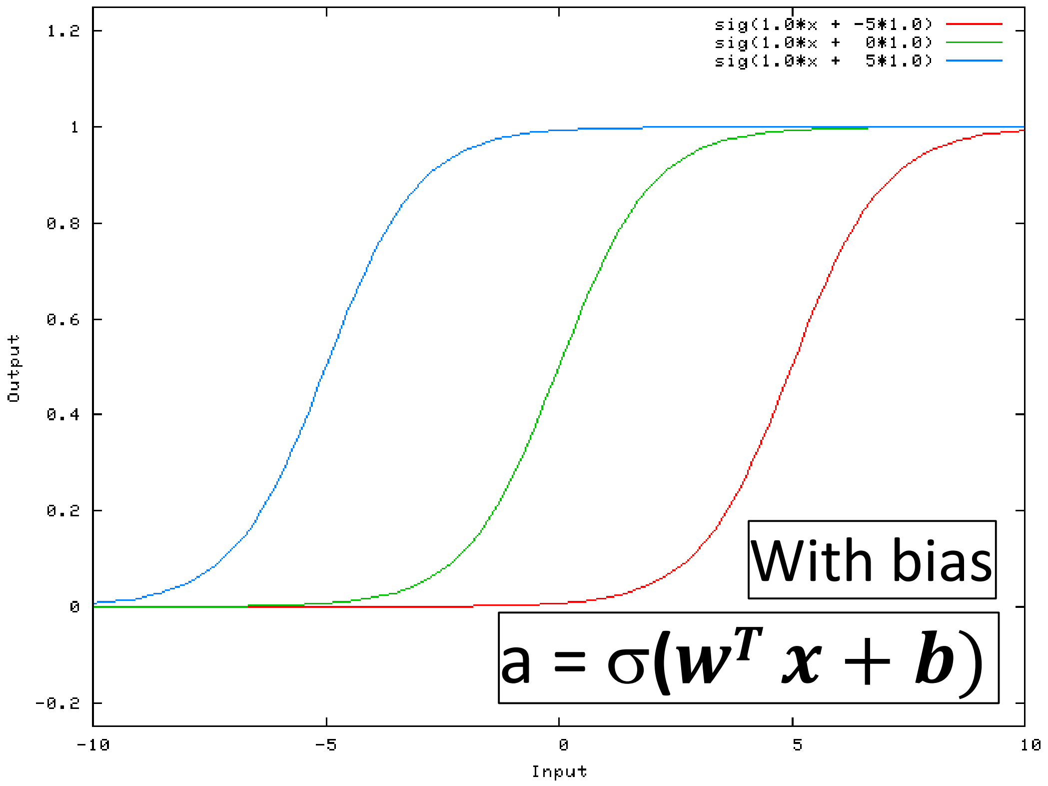


Image Source: https://cv-tricks.com/machine-learning/bias-variance-trade-off/

Bias

- It is a value that allows to shift the activation function to left or right, to better fit the data



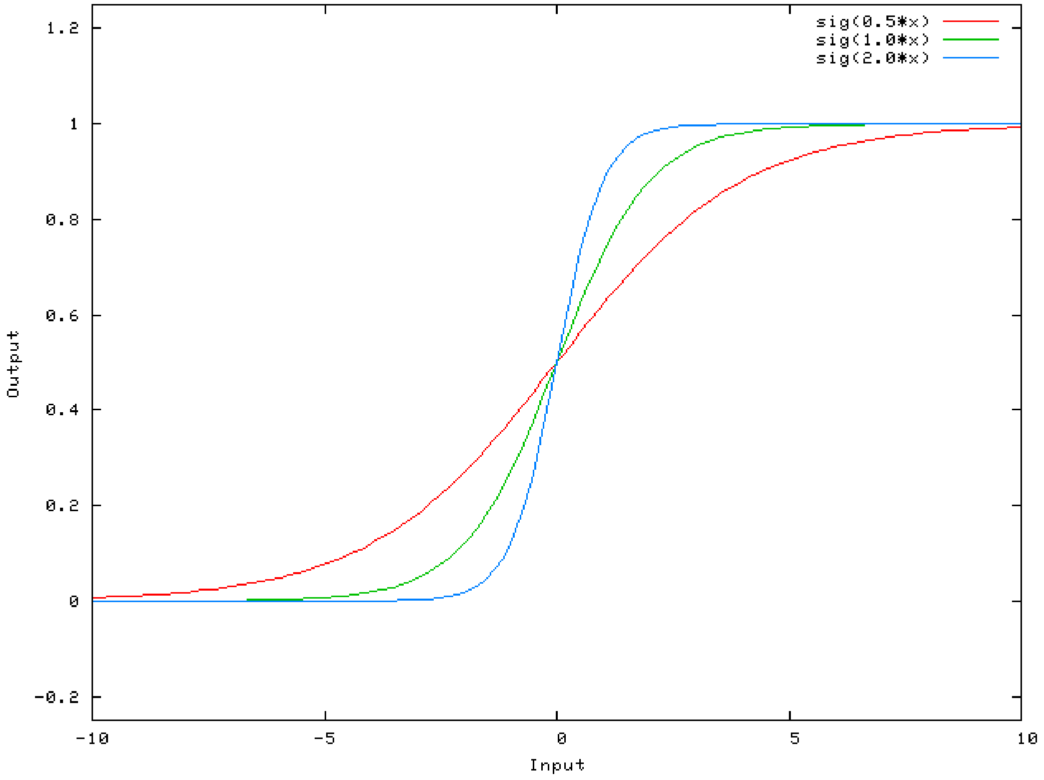


Reference and (image) Source: https://medium.com/deeper-learning/glossary-of-deep-learning-bias-cf49d9c895e2

Bias

- Changes in ‘**w**’ alters the steepness of the curve, keeping the origin at (0,0) or same/unchanged

- Without bias we may get a poor fit to training data



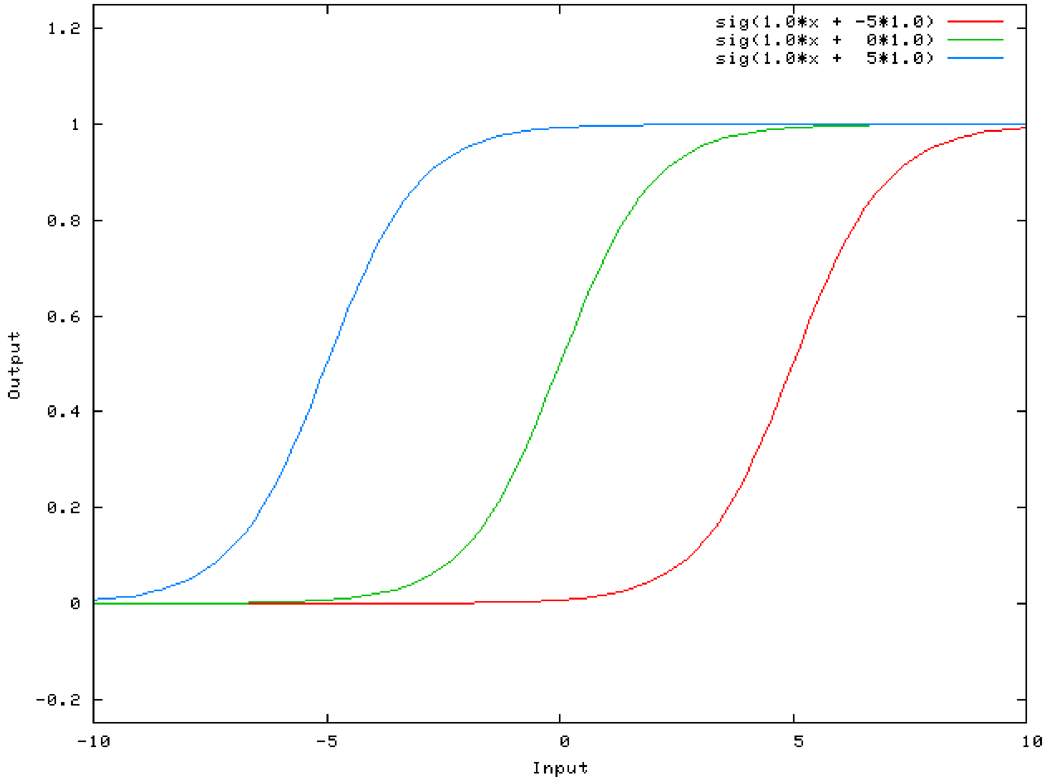
Without bias

$a = \sigma(\boldsymbol{w}^T \boldsymbol{x})$

Reference and (image) Source: https://medium.com/deeper-learning/glossary-of-deep-learning-bias-cf49d9c895e2

Bias

- Changes in ‘**b**’ shifts the curve to left or right
- With bias the curve/line will not always pass through origin
- We get a better fit to training data



With bias

a = $\sigma(\boldsymbol{w}^T \boldsymbol{x} + \boldsymbol{b})$

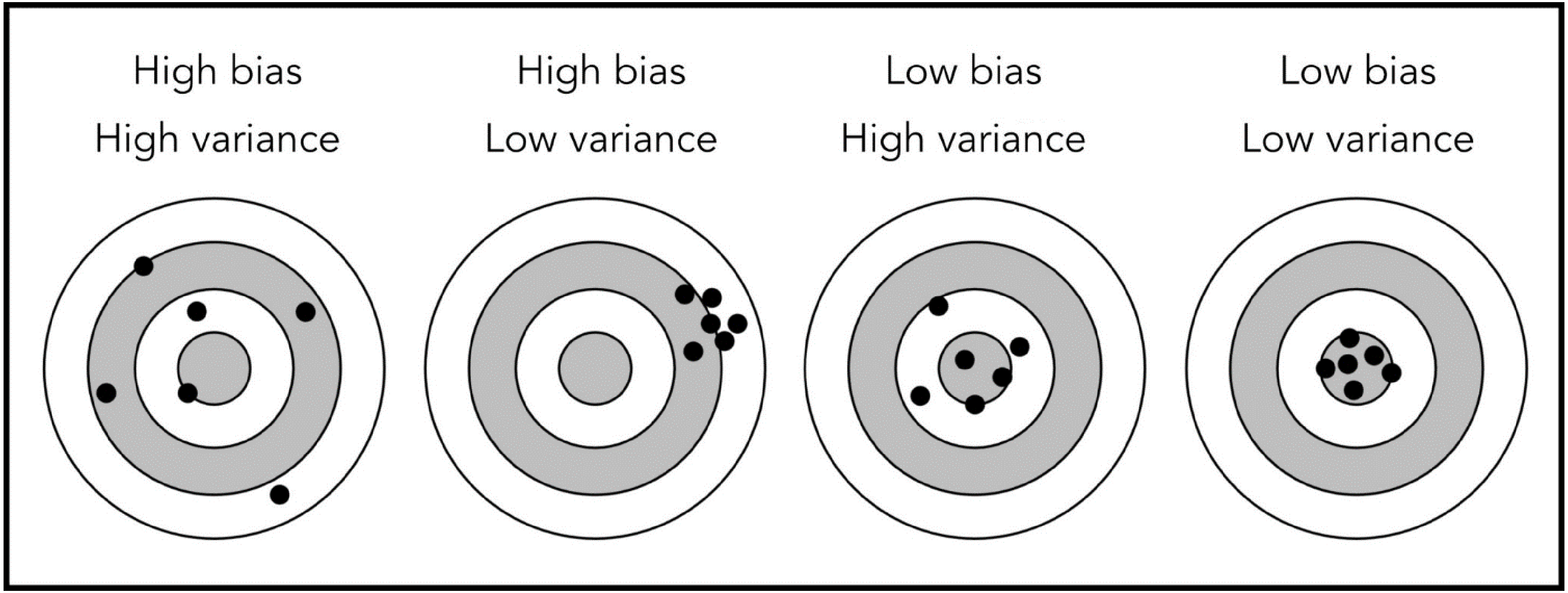
Reference and (image) Source: https://medium.com/deeper-learning/glossary-of-deep-learning-bias-cf49d9c895e2

Variance

- It is the change in prediction accuracy of Machine Learning model between training data and test data.
- Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before.
- With high variance, models perform very well on training data but has high error rates on test data.

Reference and Source: https://medium.com/datadriveninvestor/bias-and-variance-in-machine-learning-51fdd38d1f86

Bias and Variance effect



Reference and Source: http://snoek.ddns.net/~oliver/mysite/the-bias-variance-tradeoff.html

Bias and Variance effect

- **Identify High Bias:**
  - High training error
  - Validation/test error nearly same as train error
- **Identify High Variance:**
  - Low training error
  - High validation/test error

Reference and Source: https://medium.com/datadriveninvestor/bias-and-variance-in-machine-learning-51fdd38d1f86

Bias and Variance effect

- **High Bias Low Variance:** Models are consistent but inaccurate
- **High Bias High Variance:** Models are inconsistent and inaccurate
- **Low Bias and Low Variance**: Models are consistent and accurate
- **Low Bias and High Variance:** Models are somewhat accurate but inconsistent on average

Reference and Source: https://medium.com/datadriveninvestor/bias-and-variance-in-machine-learning-51fdd38d1f86

Fixing Bias and Variance issues

- **High Bias:** Due to simple ML model and high training error.
- **Potential things to try :**
  - Increase features: this will help in generalizing dataset
  - Make ML model more complicated
  - Decrease *Regularization* parameter

Reference and Source: https://medium.com/datadriveninvestor/bias-and-variance-in-machine-learning-51fdd38d1f86

Fixing Bias and Variance issues

- **High Variance:** Due to a ML model which is fitting most of the training dataset - overfitting.
- **Potential things to try :**
  - Increase dataset size
  - Reduce input features
  - Increasing *Regularization* parameter

Reference and Source: https://medium.com/datadriveninvestor/bias-and-variance-in-machine-learning-51fdd38d1f86

Regularization

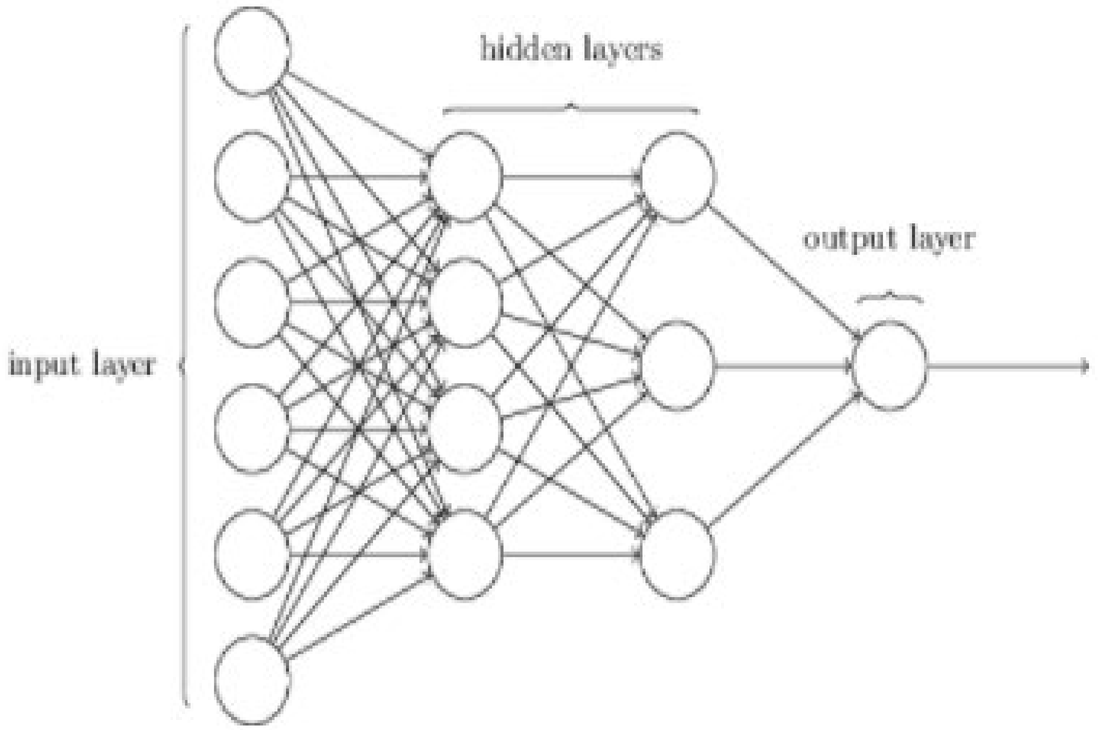
- Regularization is a technique which makes slight modifications to the learning algorithm such that the model generalizes better.

- Improves the model's performance on the unseen data as well.

- Popular techniques:
  - L2 and L1 regularization
  - Dropout

Source and reference: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

Regularization



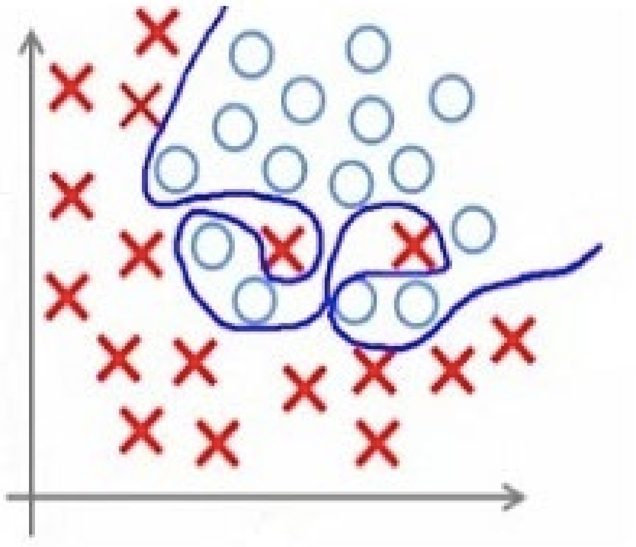
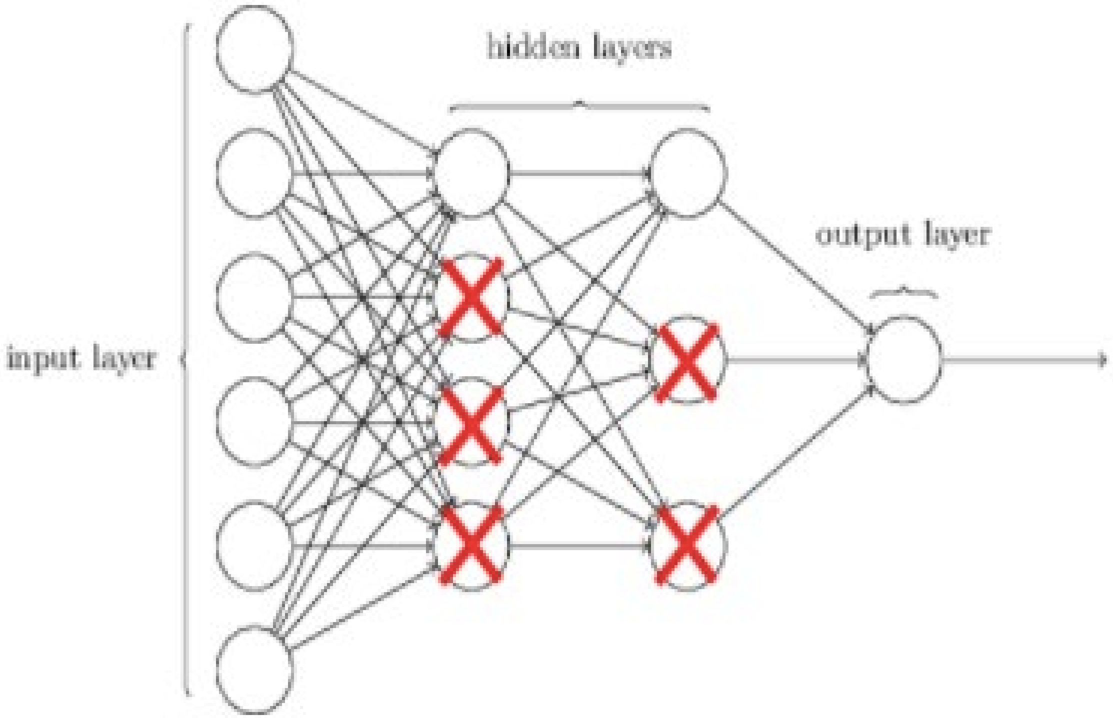


Image Source and reference: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

Regularization



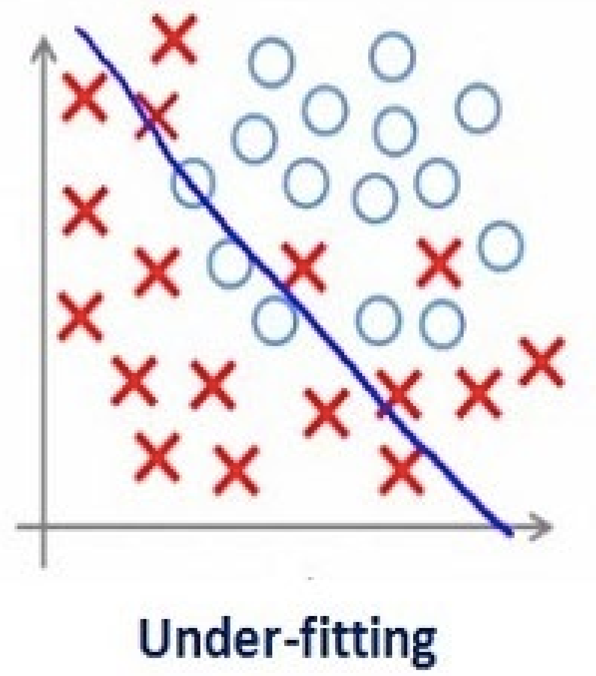


Image Source and reference: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

Regularization- L1 and L2

- L2 and L1 regularization are common types and help in reducing the overfitting issue
- Idea: Update the loss/cost function by adding a regularization term

Loss function = Loss + Regularization term (λ)

- Due to λ, the weight matrices will decrease, assuming a neural network with smaller weight matrices leads to simpler model
- In Deep Learning, Regularization penalizes the weight matrices of the nodes

Source and reference: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

Regularization- L1 and L2

- L2 regularization:

$$\text{Cost function} = \text{Loss} + \boxed{\frac{\lambda}{2m} * \sum ||w||^2}$$

λ is a hyper-parameter

Also know as weight decay, as it forces the weight to decay towards zero, but not exactly zero.

Source and reference: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

Regularization- L1 and L2

- L1 regularization:

$$\text{Cost function} = \text{Loss} + \boxed{\frac{\lambda}{2m} * \sum ||w||}$$

- Penalize the absolute value of the 'w'
- Weight may reduce to zero
- Useful in compress a model

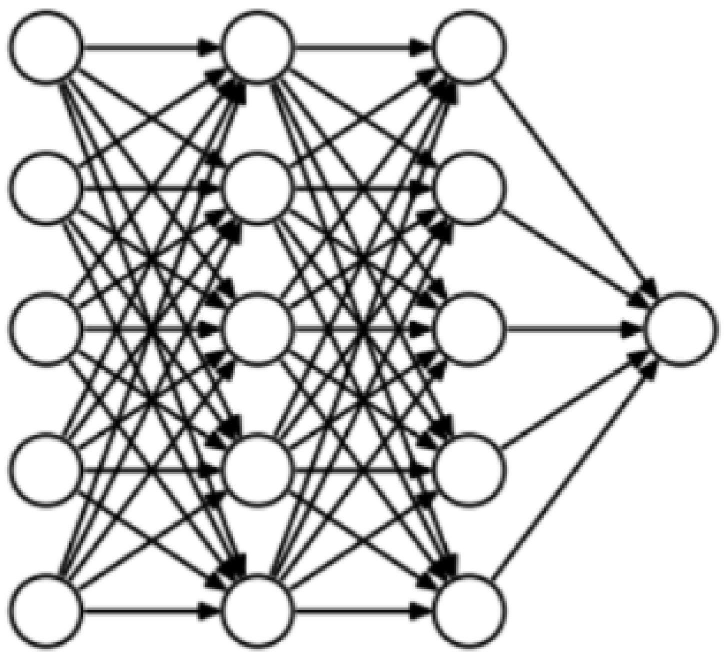
Source and reference: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

Regularization- Dropout

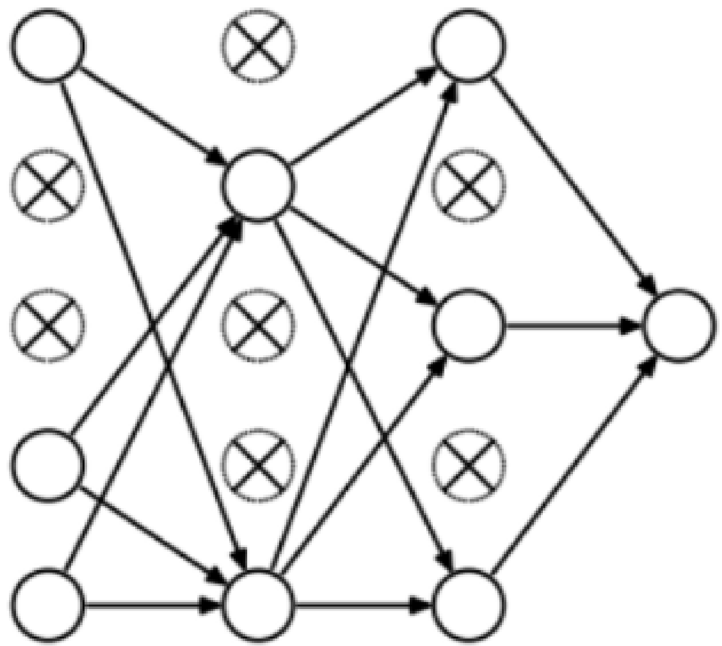
- It produces good results and most popular regularization technique
- At every iteration it randomly selects and drops some nodes and remove all the connections to and from them
- Each iteration has a different set of nodes

Source and reference: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

Regularization- Dropout



Example Deep NN

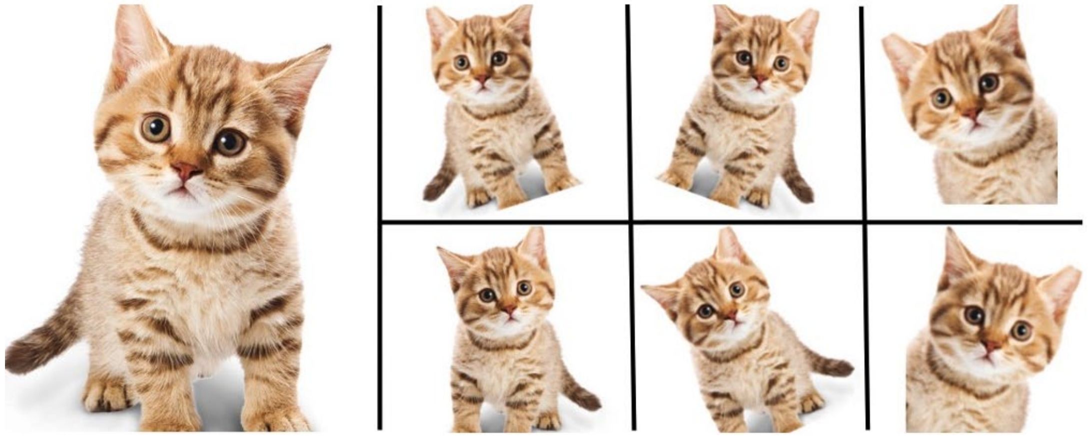


Example Deep NN with Dropout

Source and reference: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

Data Augmentation

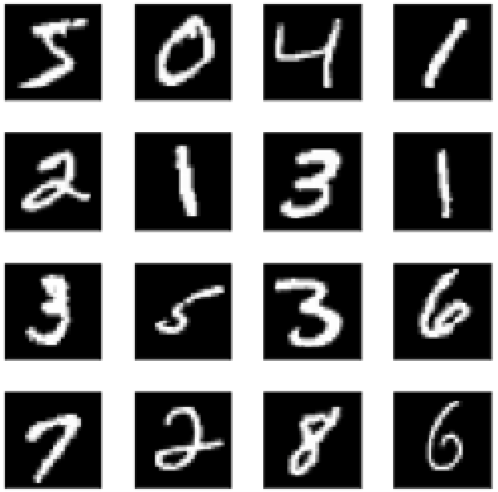
- Another simple way to reduce overfitting is to increase size of training dataset!
- Increase the size of training data by creating more sample using the existing training set and applying the following simple operations:
  - Flip
  - Rotate
  - Scale
  - Crop
  - Translate
  - Gaussian Noise



Source and reference: https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/

Data Augmentation

- Advanced data augmentation techniques:
  - Generative Adversarial Networks (GANs):
    - Among the hottest topic is DL
    - Able to generate images which look similar to the original ones
    - Proven to be very effective



Original image from MNIST

GAN generated

Source and reference: https://towardsdatascience.com/advanced-data-augmentation-strategies-383226cd11ba
Image Source: https://towardsdatascience.com/having-fun-with-deep-convolutional-gans-f4f8393686ed

Data Augmentation

- Advanced data augmentation techniques:
  - Neural Style transfer:
    - Using CNN to separate style
    - transfer style to different image

Source and reference: https://towardsdatascience.com/advanced-data-augmentation-strategies-383226cd11ba