Diagnosing ML System

By Gaddisa Olani (Ph.D.)

gaddisaolex@gmail.com

Debugging a learning algorithm

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

- You have built you awesome linear regression model predicting price
- Work perfectly on you testing data

Debugging a learning algorithm

$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

- You have built you awesome linear regression model predicting price
- Work perfectly on you testing data
- Then it fails miserably when you test it on the new data you collected

Debugging a learning algorithm

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)^2$$

- You have built you awesome linear regression model predicting price
- Work perfectly on you testing data
- Then it fails miserably when you test it on the new data you collected
- What to do now?

Things You Can Try

- Get more data
- Try different features
- Try tuning your hyperparameter

Things You Can Try

- Get more data
- Try different features
- Try tuning your hyperparameter

- But which should I try first?

Diagnosing Machine Learning System

- Figure out what is wrong first
- Diagnosing your system takes time, but it can save your time as well
- Ultimate goal: low generalization error

Generalization

Training set (labels known)

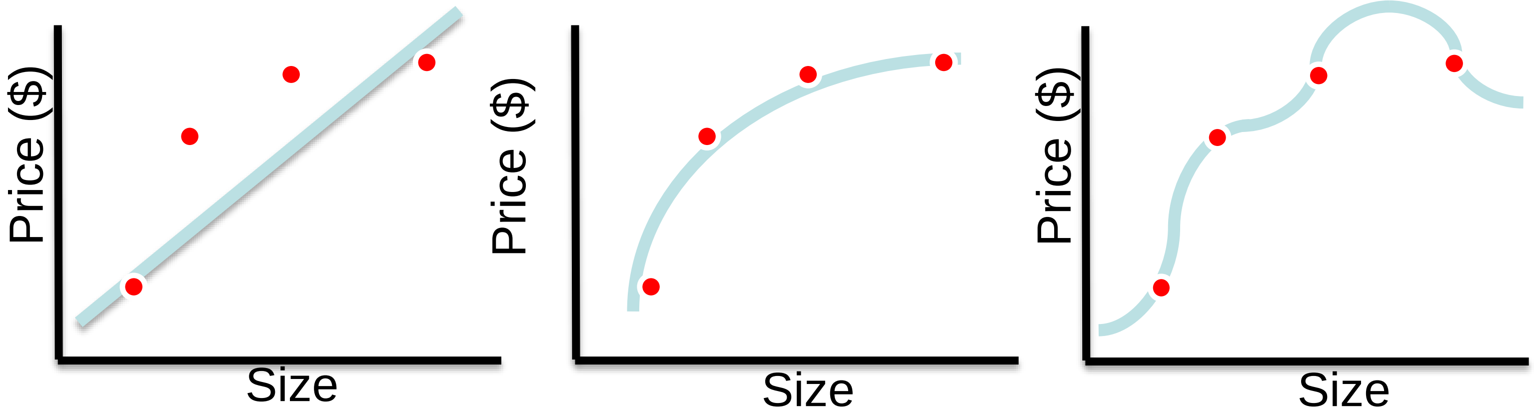
Test set (labels unknown)

- How well does a learned model generalize from the data it was trained on to a new test set?

Generalization

- Components of generalization error
  - **Bias:** how much the average model over all training sets differ from the true model?
    - Error due to inaccurate assumptions/simplifications made by the model
  - **Variance:** how much models estimated from different training sets differ from each other
- **Underfitting:** model is too “simple” to represent all the relevant class characteristics
  - High bias and low variance
  - High **training error** and **high test error**
- **Overfitting:** model is too “complex” and fits irrelevant characteristics (noise) in the data
  - Low bias and high variance
  - Low training error and high test error

Evaluate Your Hypothesis



$\theta_0 + \theta_1 x$

Underfit

$\theta_0 + \theta_1 x + \theta_2 x^2$

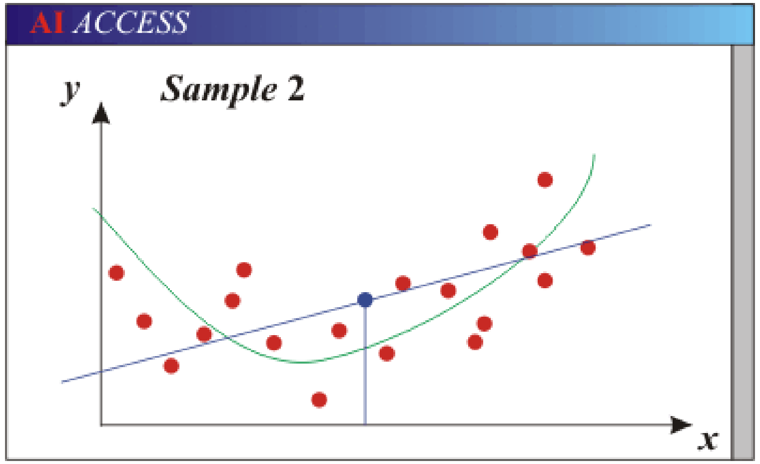
Just right

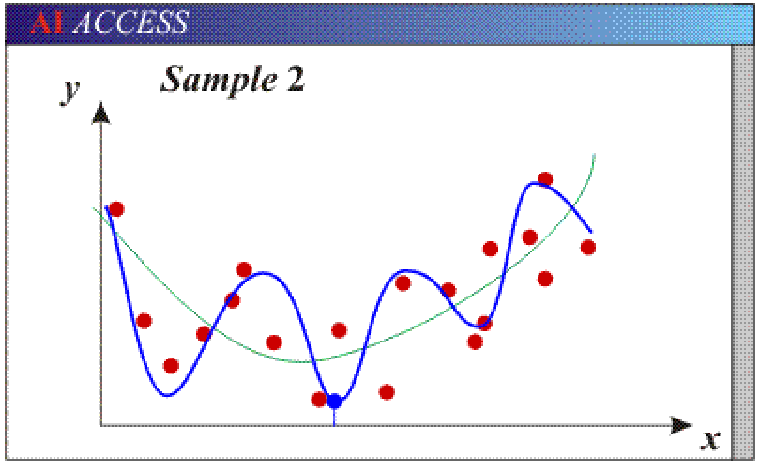
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Overfit

gaddisaolex@gmail.com

Bias-Variance Trade-off





- Models with too few parameters are inaccurate because of a large bias (not enough flexibility).

- Models with too many parameters are inaccurate because of a large variance (too much sensitivity to the sample).

Slide credit: D. Hoiem

Technique to Avoid Overfitting?

- Regularization
- Dropout
- Data Augmentation

Regularization

- Attempt to guide solution to *not overfit*

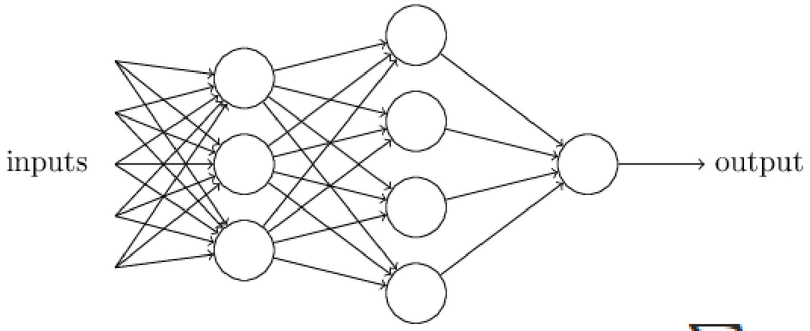
- But still give freedom with many parameters
  - Idea: add a cost to having high weights
  - λ = regularization parameter

$$C = C_0 + \frac{\lambda}{2n} \sum_w w^2,$$

Dropout

- Our networks typically start with random weights.
- Every time we train = slightly different outcome.

- Why random weights?
- If weights are all equal, response across filters will be equivalent.
  – Network doesn't train.



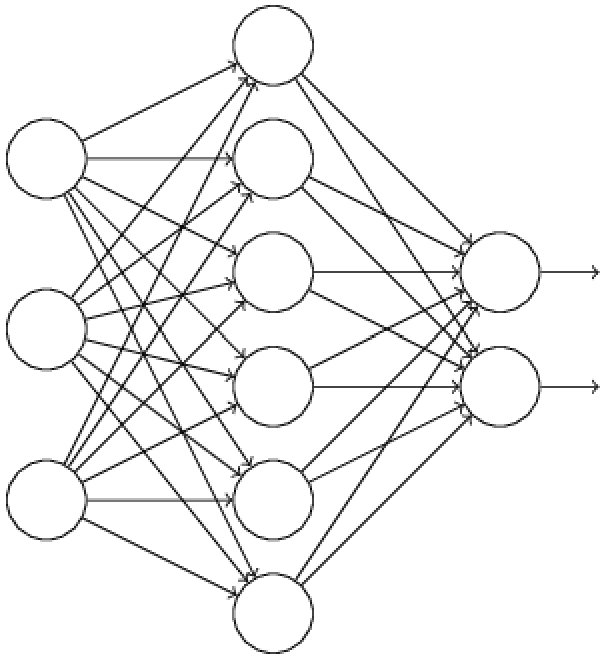
$$w \cdot x \equiv \sum_j w_j x_j,$$

Dropout cont…

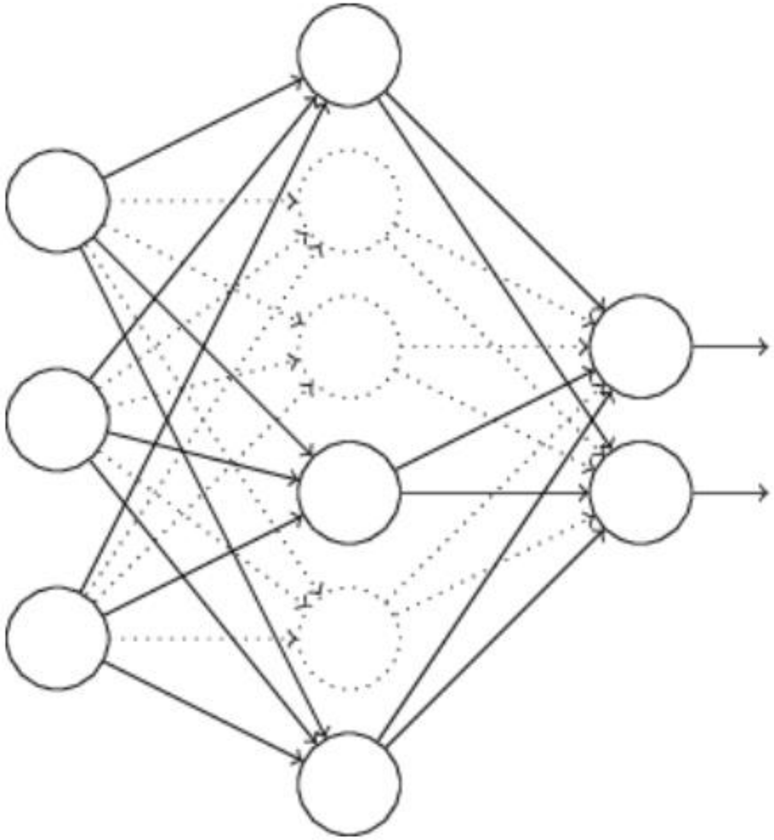
- Our networks typically start with random weights.
- Every time we train = slightly different outcome.

- Why not train 5 different networks with random starts and vote on their outcome?
  - Works fine!
  - Helps generalization because error is averaged.

Regularization: Dropout



Regularization: Dropout



At each mini-batch:
- Randomly select a subset of neurons.
- Ignore them.

On test: half weights outgoing to compensate for training on half neurons.

Effect:
- Neurons become less dependent on output of connected neurons.
- Forces network to learn more robust features that are useful to more subsets of neurons.
- Like averaging over many different trained networks with different random initializations.
- Except cheaper to train.

Data Augmentation

Data augmentation in data analysis are techniques used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. It acts as a regularize and helps reduce Overfitting when training a machine learning model.

Technique to Avoid Underfitting?

- Increase model complexity.

- Increase the number of features, performing feature engineering.

- Remove noise from the data.

- Increase the number of epochs or increase the duration of training to get better results.

How to reduce variance?

- Choose a simpler classifier

- Regularize the parameters

- Get more training data

gaddisaolex@gmail.com