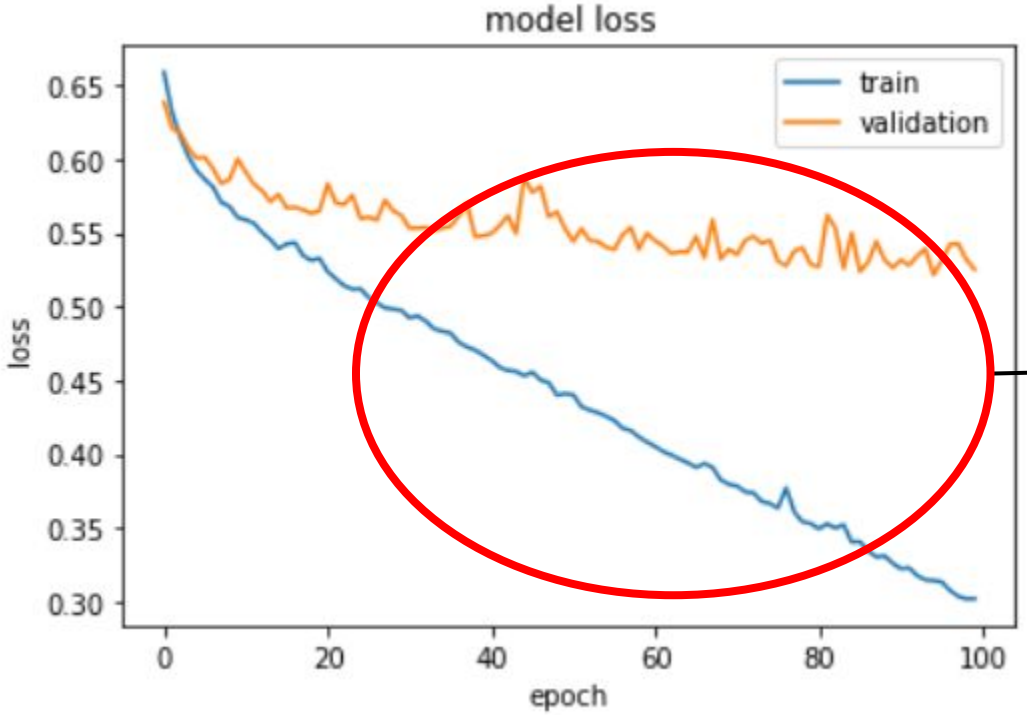
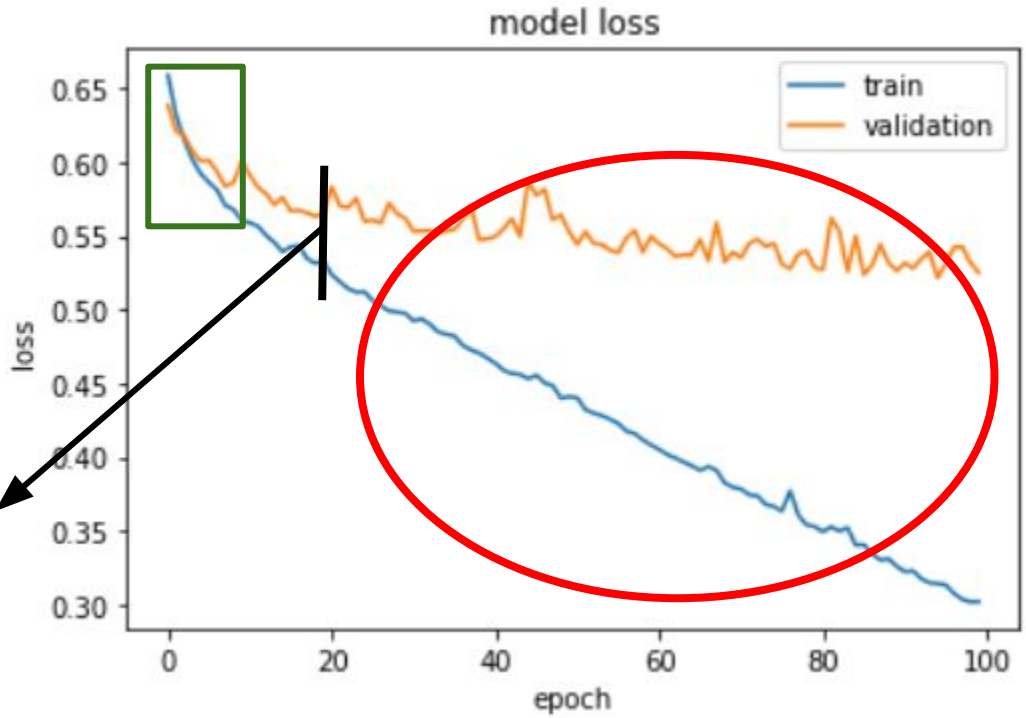
Improving your Neural Network

Overfitting



Model is overfitting

Solution 1: Early Stopping



Stop the training here

Why our model is overfitting?

Why our model is overfitting?

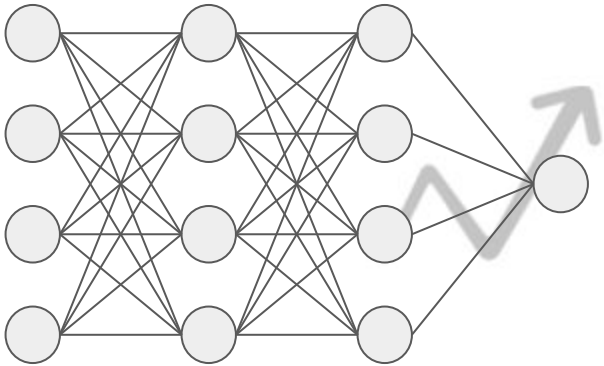
- Does not learn patterns / signals from training data

Why our model is overfitting?

- Does not learn patterns / signals from training data
- Model starts memorizing the training data

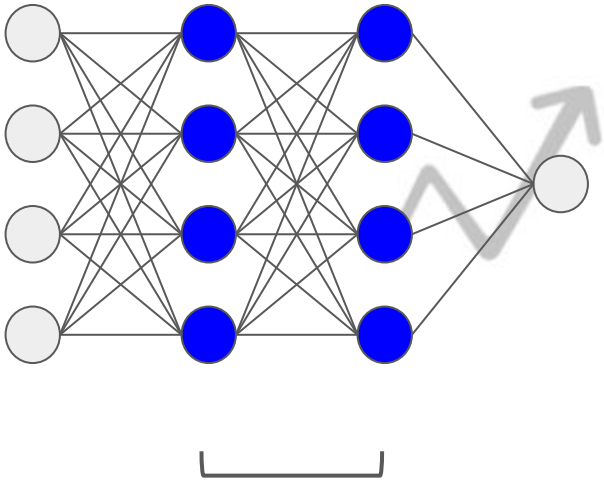
Solution 2: Dropout Regularization

Solution 2: Dropout Regularization



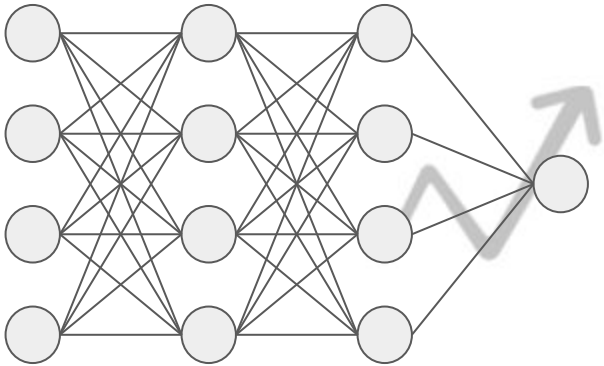
Input Layer Hidden Layer Output Layer

Solution 2: Dropout Regularization

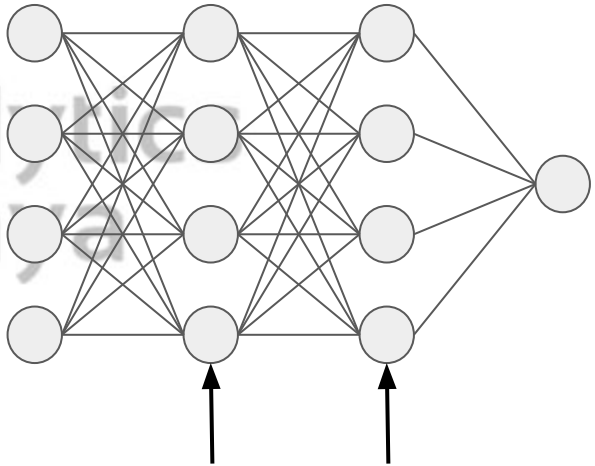


Hidden Layer

Solution 2: Dropout Regularization



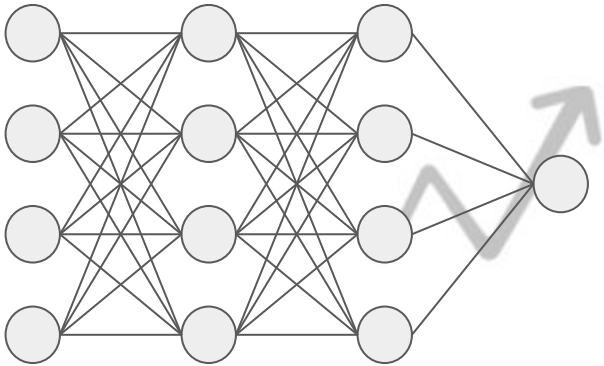
Without Dropout



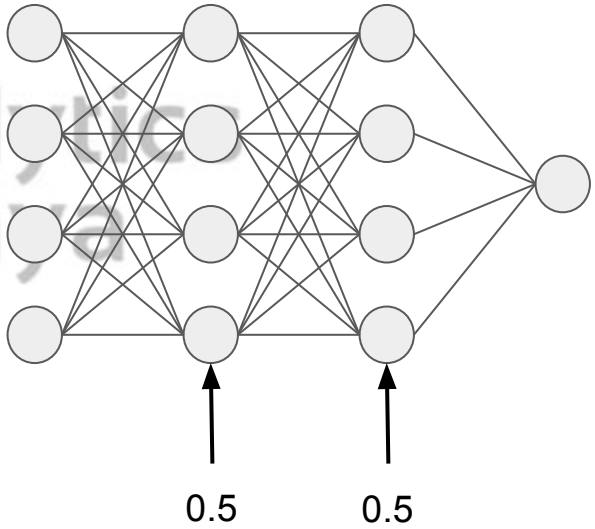
Probability

With Dropout

Solution 2: Dropout Regularization

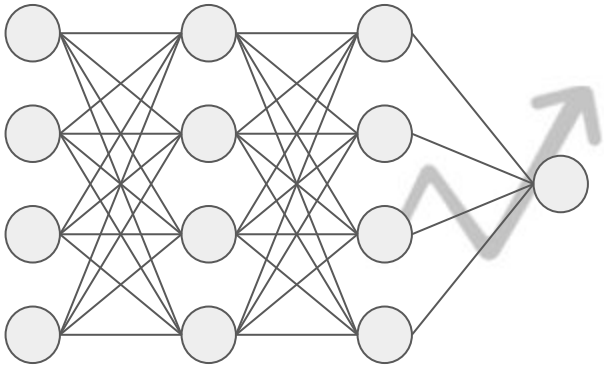


Without Dropout

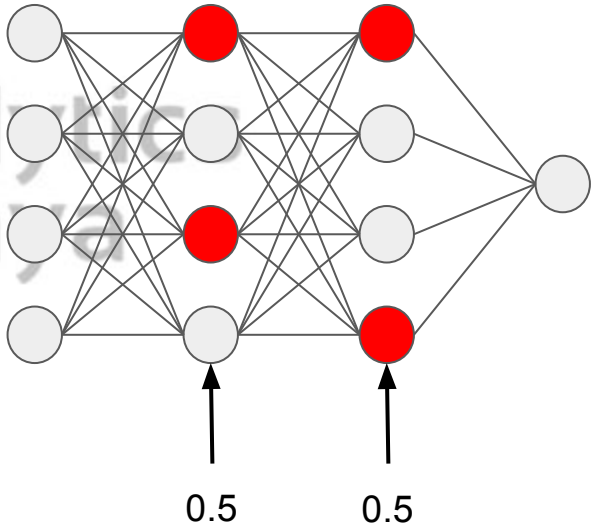


With Dropout

Solution 2: Dropout Regularization

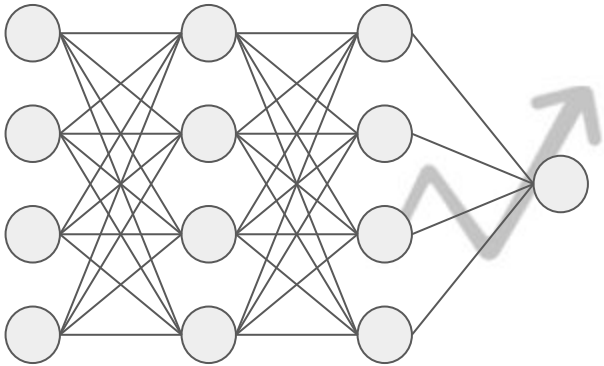


Without Dropout

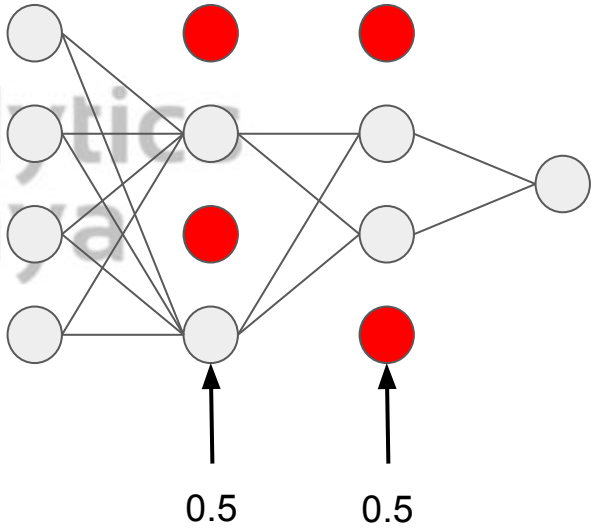


With Dropout

Solution 2: Dropout Regularization

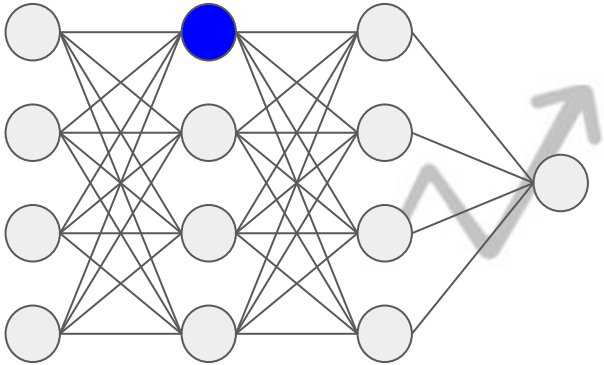


Without Dropout



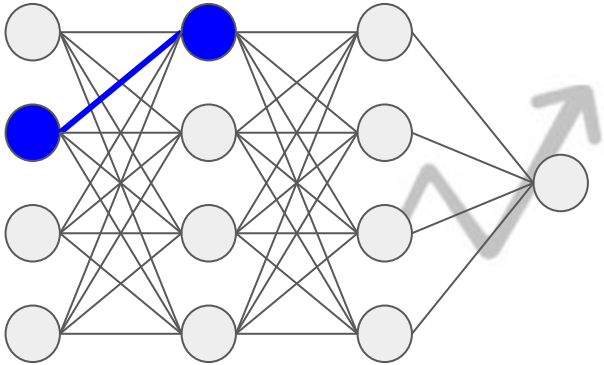
With Dropout

Solution 2: Dropout Regularization



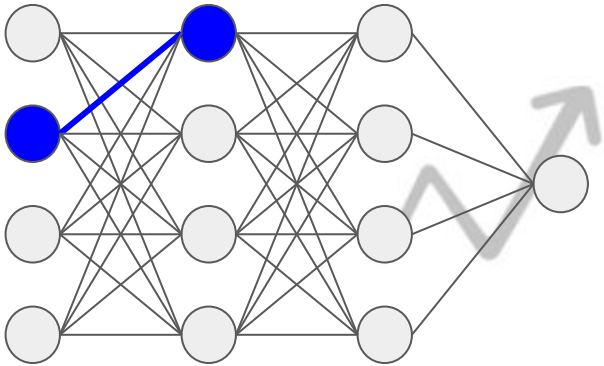
Without Dropout

Solution 2: Dropout Regularization

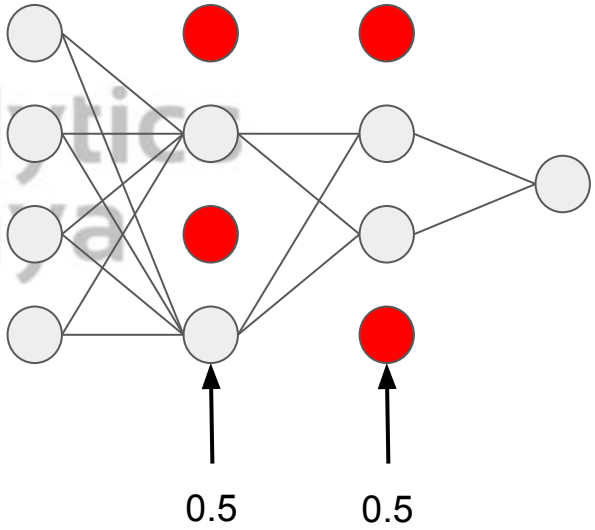


Without Dropout

Solution 2: Dropout Regularization



Without Dropout



With Dropout

Notes: Dropout Regularization

- At each iteration, neurons are dropped randomly

Notes: Dropout Regularization

- At each iteration, neurons are dropped randomly
- Generally, we don’t apply dropout on the input layer

Notes: Dropout Regularization

- At each iteration, neurons are dropped randomly
- Generally, we don’t apply dropout on the input layer
- No dropout during test time

Steps to solve emergency vs non-emergency vehicle classification problem

1. Loading the dataset
2. Pre-processing the data
3. Creating training and validation set
4. Defining the model architecture
5. Compiling the model
6. Training the model
7. Evaluating model performance

Steps to solve emergency vs non-emergency vehicle classification problem using Dropout

1. Loading the dataset
2. Pre-processing the data
3. Creating training and validation set
4. Defining the model architecture
    - Defining dropout layer(s)
5. Compiling the model
6. Training the model
7. Evaluating model performance

Thank You!