Welcome to Deep Learning Online Bootcamp

Day 9, 10 - Optimizing a Neural Network - Part 1

dφ

Democratizing Data Science Learning

Learning Objectives

Tensorboard

Validation Set

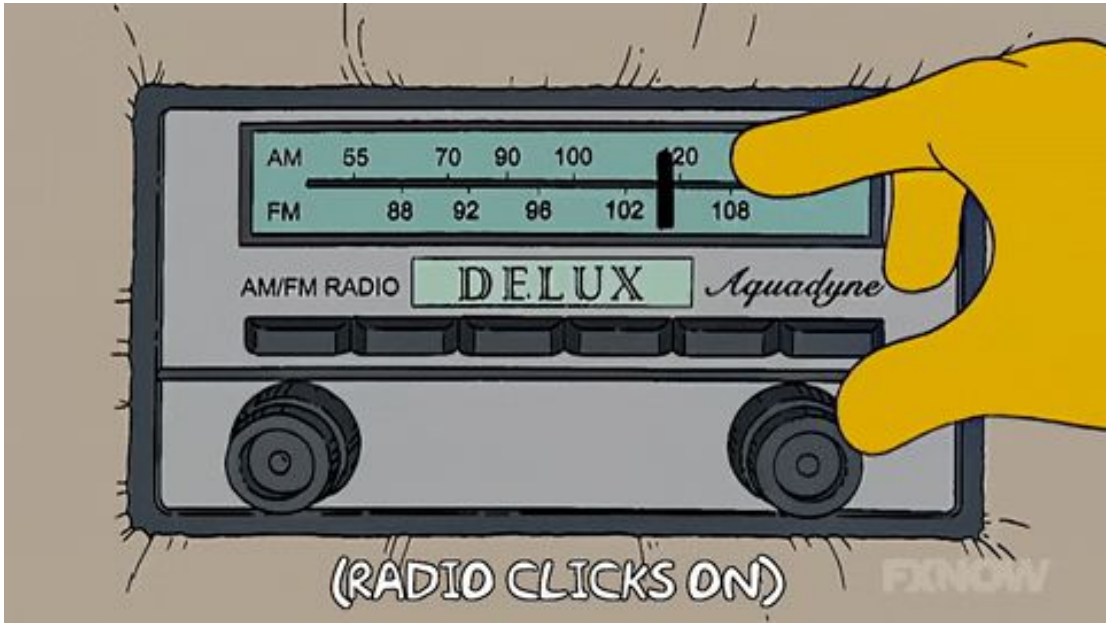
Overfitting and Underfitting

Bias and Variance

Techniques to avoid Overfitting - Early Stopping, Regularization, Dropout

Optimizing a Neural Network

Optimizing a Neural Network or any model for that matter is like tuning a radio. The difference is there might be a million knobs to tune simultaneously. Let's have a look at all the knobs one by one.



Tensorboard

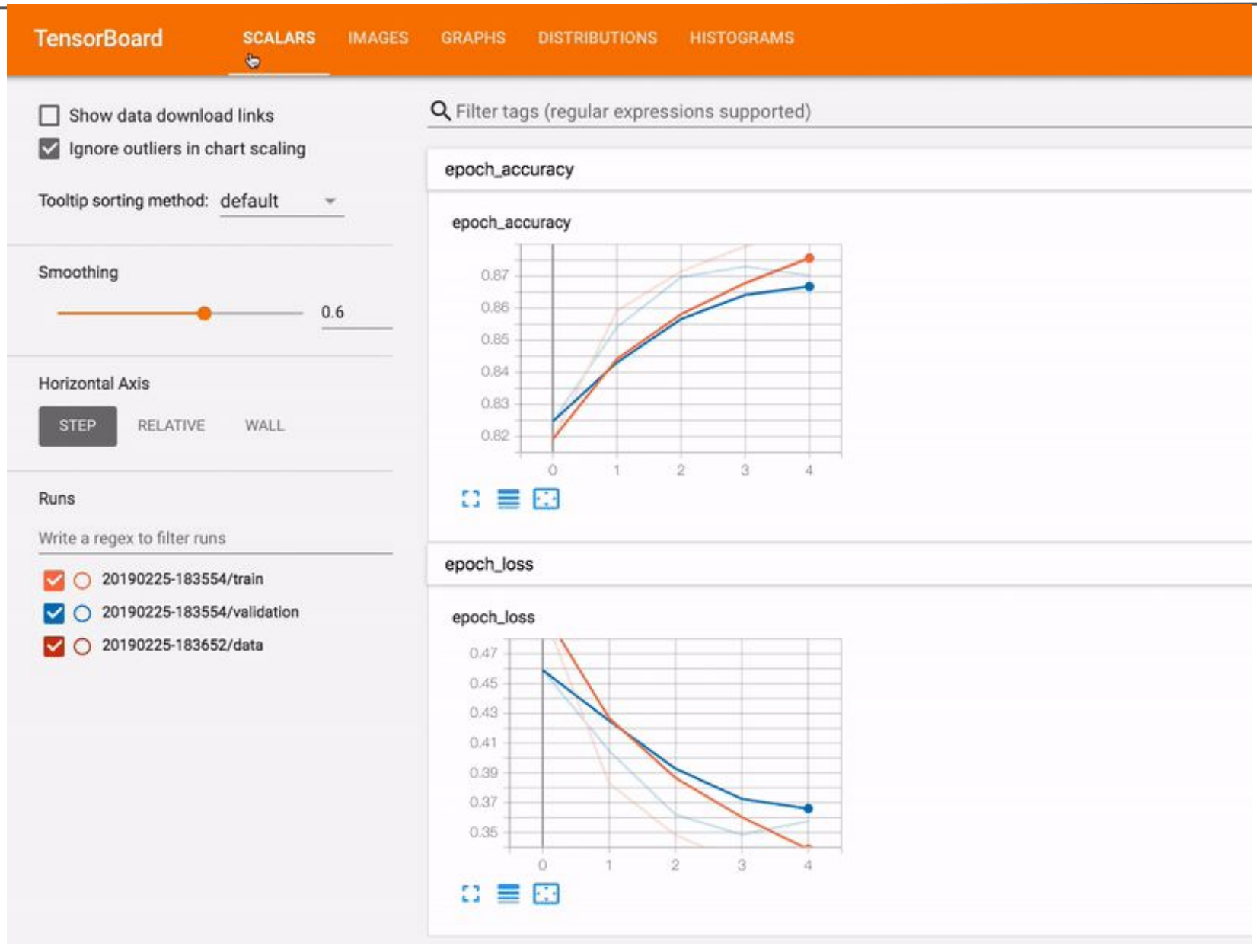
TensorBoard: TensorFlow's visualization toolkit

TensorBoard provides the visualization and tooling needed for machine learning experimentation:

- Tracking and visualizing metrics such as loss and accuracy
- Visualizing the model graph
- Viewing histograms of weights, biases, or other tensors as they change over time
- Displaying images, text, and audio data

And much more!

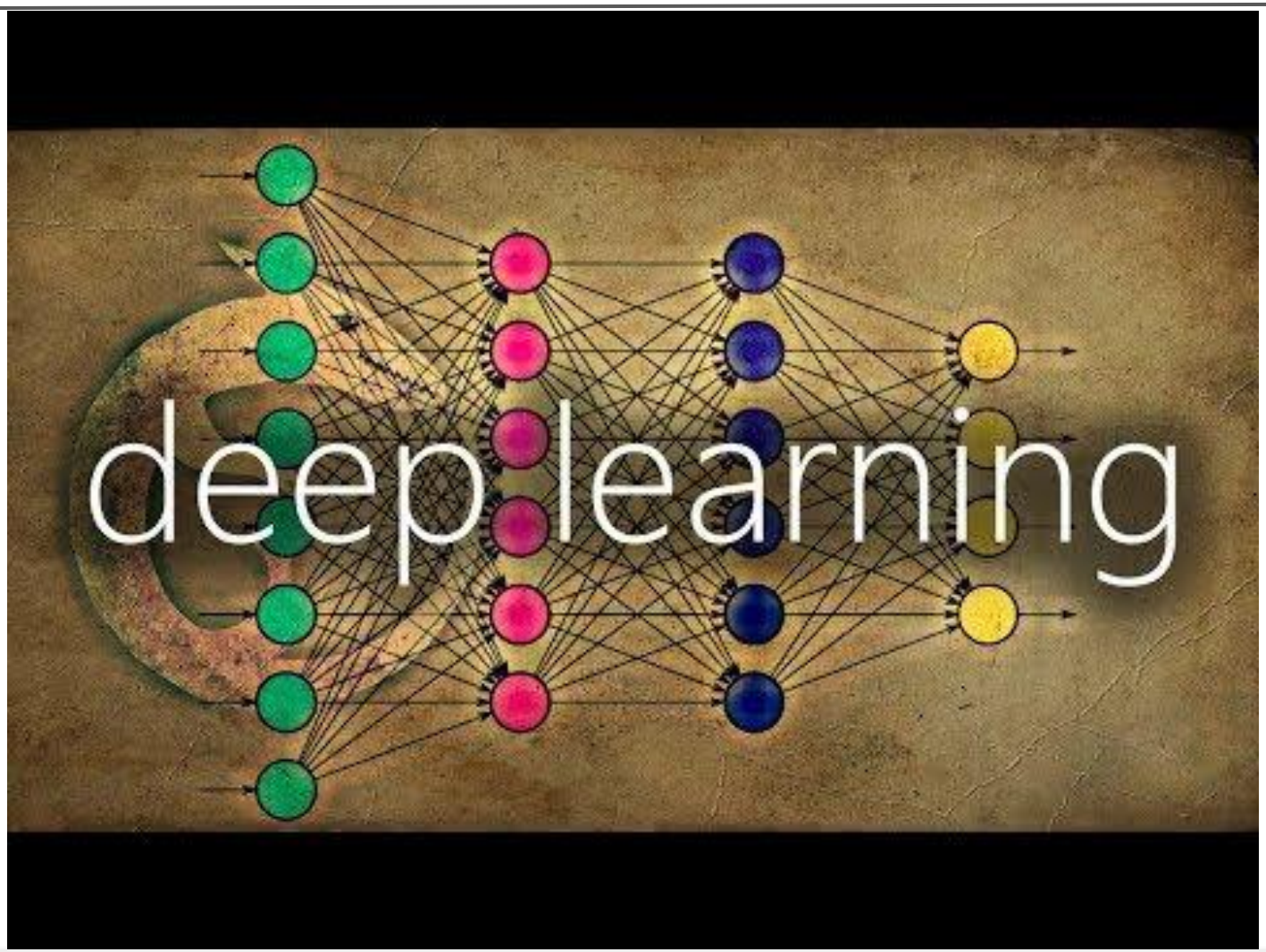
TensorBoard: TensorFlow's visualization toolkit



TensorBoard: TensorFlow's visualization toolkit

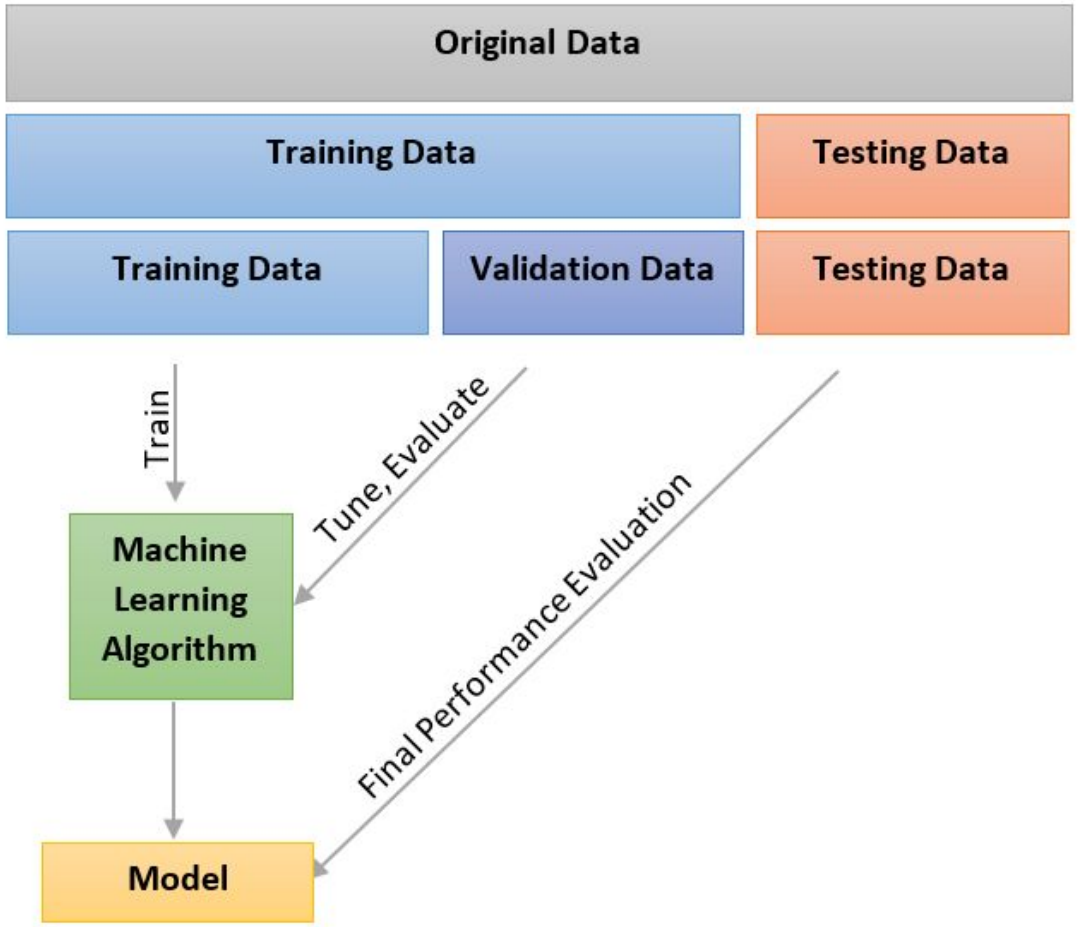
Train, Validation & Test Set

Train, Validation and Test Set



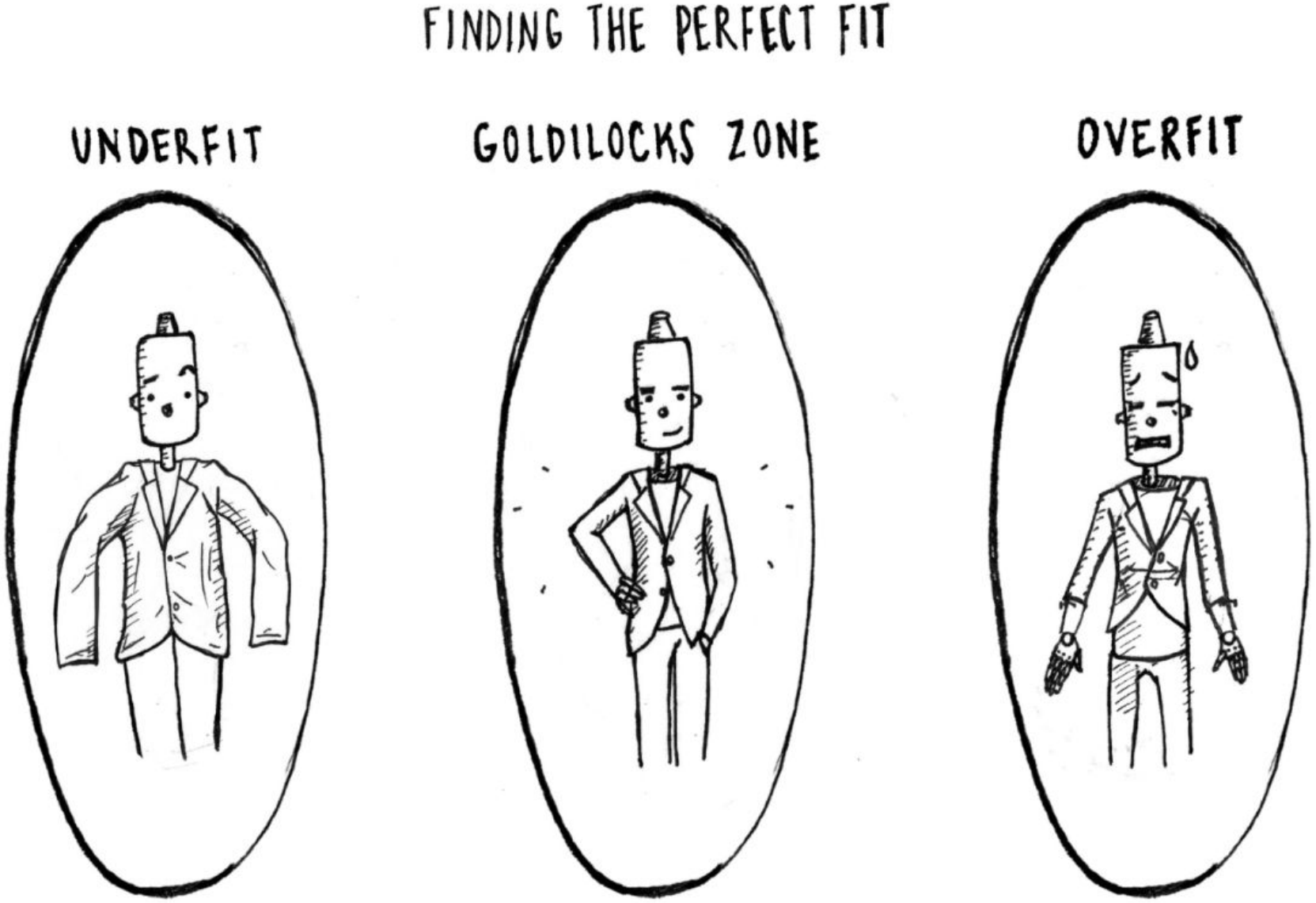
Train, Validation and Test Set

- **Training Dataset:** The sample of data used to fit (train) the model.
- **Validation Dataset:** The sample of data used to provide an evaluation of model's performance while tuning model hyperparameters.
- **Test Dataset:** The sample of data used for the final evaluation of model's performance



Overfitting and Underfitting

Overfitting - Underfitting



Overfitting

Overfitting is a scenario where your model **performs well on training data** but **performs poorly on data not seen during training**.

This basically means that your model has **memorized the training data** instead of learning the relationships between features and labels.

Imagine getting a **'too good to be true' accuracy on training data** but a **bad accuracy on test/validation data**. That's overfitting for you.

Underfitting

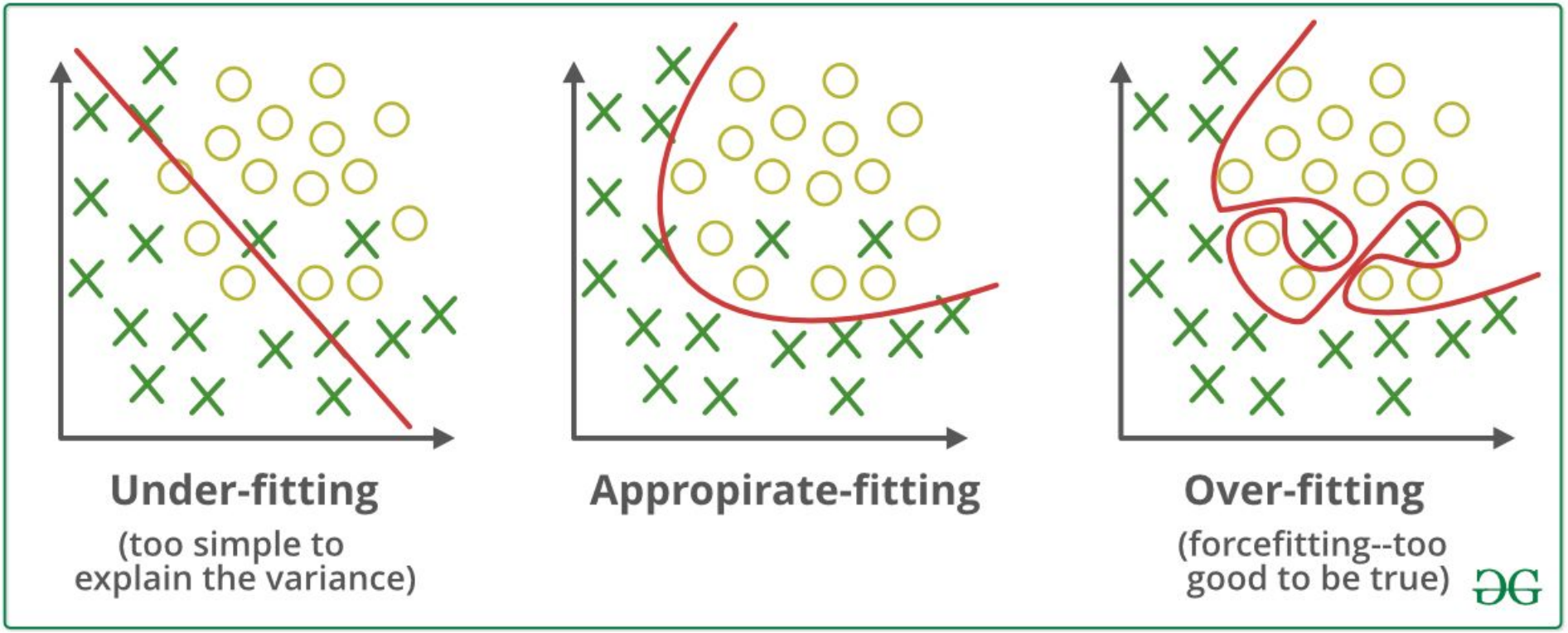
A machine learning algorithm is said to have underfitting when it **cannot capture the underlying trend of the data.**

You can think of underfitting to be the opposite of overfitting.

Overfitting learns the training data TOO well whereas Underfitting is not able to learn the data sufficiently well.

Imagine **neither getting a good accuracy in training data, nor in test/validation data**. That's underfitting for you.

Overfitting - Underfitting



Bias and Variance
And how they are related to underfitting and overfitting

Bias and Variance in real world

In dictionary terms :

Bias : Prejudice in favor of or against one thing, person, or group compared with another, usually in a way considered to be unfair.

Variance: The state or fact of disagreeing or quarreling.

In short, Bias represents how unfair is something towards others, and Variance represents how likely something changes with respect to others.

Confusing ? Worry not. The next example will clarify all your doubts.

Example

Let’s assume you have called two weather examiners, Mr. Bishop and Mr. Varian to test if it will rain or not.

Mr. Bishop loves rain a lot. And Mr. Varian is a bookworm.

Let us talk about the conditions for rain.

- **It rains only if it’s little humid.**

- **It does not rain if it's windy, hot or freezing.**

Mr. Bishop representing Bias

You ask Mr. Bishop (Despite of his training, he is too biased towards rain) :

Me :Sir, its extremely hot out here, will it rain ?
Mr. Bishop : Yup.

Me :Sir, its little windy, will it rain ?
Mr. Bishop : May be not.

Me :Sir, its freezing will it rain ?
Mr. Bishop : Yes of course.

Me :Sir, its little humid, will it rain ?
Mr. Bishop : Damn sure.

Did you notice, Mr. Bishop is highly Biased towards chances of having rain. During the test, he is unable to predict most of them correctly.

This condition is called **under fitting**.

Mr. Varian representing Variance

Now let us see your conversation with Mr. Varian (a bookworm who completely remembers the training he had):

Me :Sir, its extremely hot out here, will it rain ?
Mr. Varian: Nope.

Me :Sir, its little windy, will it rain ?
Mr. Varian: No way.

Me :Sir, its freezing, will it rain ?
Mr. Varian: No way.

Me :Sir, its little humid, will it rain ?
Mr. Varian: Yes it will.

Mr. Varian representing Variance

Mr. Varian successfully predicted whether it will rain or not. But being a bookworm, Mr. Varian is unknown to the conditions not described in the book during training.

Now, we ask Mr. Varian :

Me :Sir, there is a giant sitting on the cloud who lost his candy. Will it rain ?
Mr. Varian: Not sure, since the answer is “No” to most of the conditions, there is a high possibility that it will not rain .

Now, although the decision of Mr. Varian varies perfectly with the input conditions, he is not able to predict for the new and unseen condition (other general conditions apart from the given specific conditions while training).

This condition is called **over fitting.** And it offers **poor generalizability**.

High Bias or High Variance?

Then what is better, high bias (high generalizability) or high variance (high accuracy on training data) ?

Well, the answer is, “Best of both worlds”. We neither need high bias nor high variance. We would want our algorithm to perform better on training set and also offer best result on unseen data (the test set).

In general, **having high bias reduces the performance of the algorithm on training set while having high variance reduces performance on unseen data.**

This is known as **Bias Variance Trade off**.

Strategies to avoid overfitting

You can combat overfitting by reducing the complexity of your model (i.e. reducing the number of parameters).This is done by :

- Using fewer layers (shallower networks), fewer neurons per layer (narrow networks)
- Using **Dropouts**
- Using **Regularisation**
- **Early Stopping** in some cases

We'll learn about all above 'technical' terms in the upcoming units.

Strategies to avoid underfitting

Underfitting is a bit harder to diagnose.

Increasing the complexity of your model i.e. increasing layers and number of neurons can help combat underfitting.

With more layers, the network can learn more sophisticated relationships and perhaps perform well on difficult real-world tasks.

Tackling Overfitting using Early Stopping, Regularization and Dropouts

Early Stopping

When training a large network, there will be a point during training when the model will stop generalizing and start memorizing or learning the noise in the training dataset i.e it will start overfitting which results bad performance when the model is exposed to unseen new dataset.

In other words, this overfitting of the training dataset will result in an increase in generalization error (model unable to work well on validation/test data), making the model less useful at making predictions on new data.

The challenge is to train the network long enough that it is capable of learning the mapping from inputs to outputs, but training should not be for so long that it overfits the training data.

Early Stopping

Let's understand Overfitting, Underfitting and Early Stopping practically

https://github.com/dphi-official/Deep_Learning_Bootcamp/blob/master/Optimization_Techniques/OptimisingNeuralNetwork.ipynb

- Download
- Extract zip file
- Open in Jupyter Notebook or Upload on Google Colab

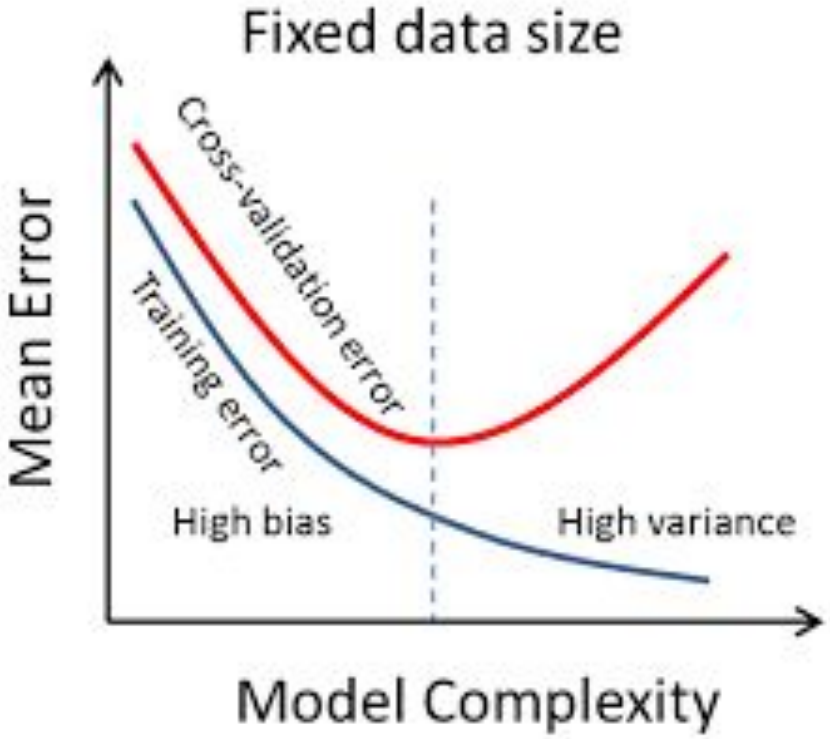
Regularization

Regularization is a technique which makes slight modifications to the learning algorithm such that the model generalizes better (Generalization refers to your model's ability to adapt properly to new, previously unseen data). This in turn improves the model’s performance on the unseen data as well.

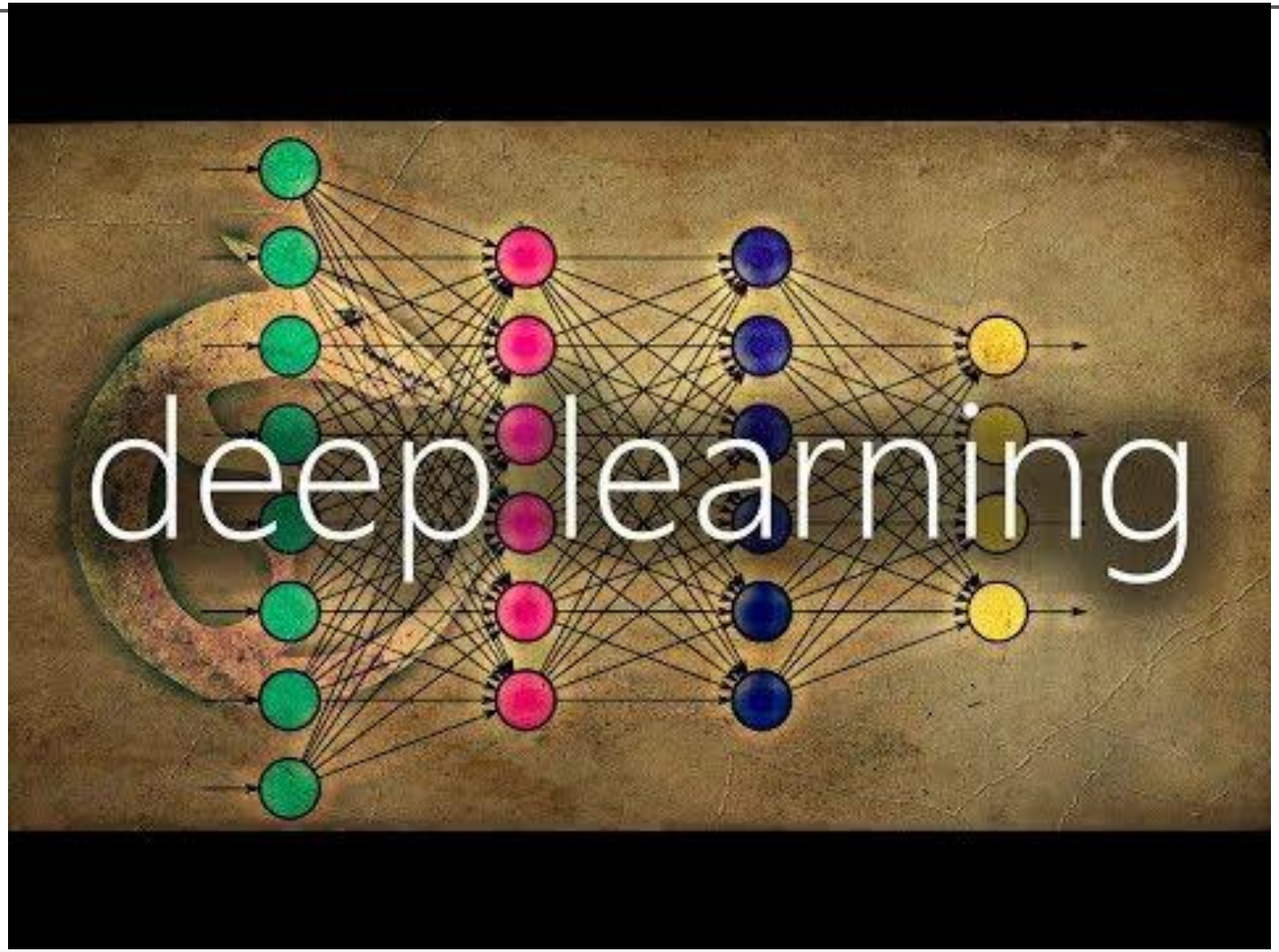
Remember when we were adding more layers to the model (making it more complex) ? Adding more than required layers might also lead to overfitting.

Regularization

By looking at the graph below can you guess, what should be an ideal model complexity?



Regularization



Dropout - An Analogy

Consider a TV show consisting of a participant, a group of audience, and a show host. The show works as follows:

- At the beginning of the game, the host selects a random unseen movie to be the main target of the event.
- At each stage, the host shows a short clip from the selected movie,
- then asks a question about the events of the movie so far.
- Each one of the audience will give an answer,
- the participant has to pick one answer from the audience.
- If the answer is correct, the participant and the chosen person from the audience will get 50$ each.
- If the answer is wrong, they both have to pay 100$.

Suppose that the participant notices that one of the audience is always giving the correct answer. With time, the participant will build trust with this person and will neglect the answers given by the others.

Dropout - An Analogy

There are some problems with this strategy.

- The trusted person may be good with the question categories in the early stages, but very bad with the ones in later stages of the game.
- If only one person (or a small group) are always selected, other people from the audience will feel left out and will not care to pay attention to the played video clip anymore.
- This way — at later stages — the trusted person will no longer be helpful, and other people from the audience who stopped paying attention will already lose the sequence of events to answer correctly.

Therefore, relying only on one person is bad, and it will definitely lead to losing a lot of money.

Dropout - An Analogy

How do you think we could manage to fix this problem?

One smart strategy is to always give chance to others.

- This way the participant will learn the strength of each one of the audience and knows which one to ask based on the question category.

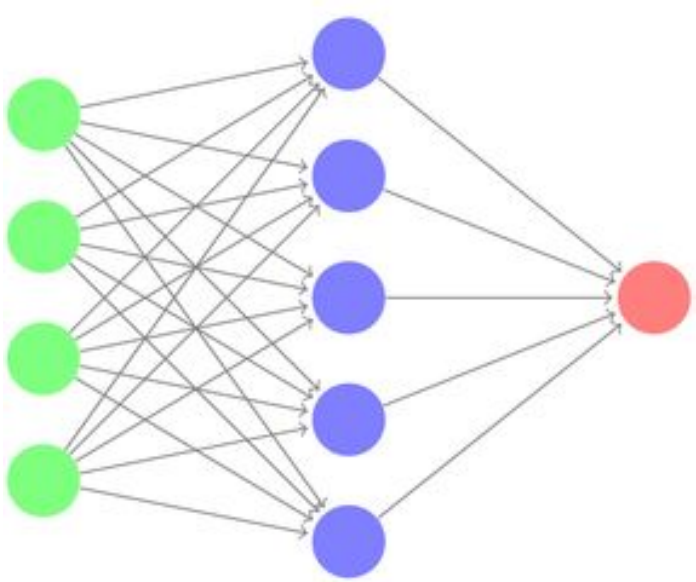
- In addition, this way everyone will always feel responsible and obliged to pay attention.

Dropout in Neural Network

You may ask what this has to do with neural network? Well, let us consider the following network...

We could think of

- the input layer (in green) as the question the host asking,
- each neuron in the hidden layer (in blue) as one person from the audience,
- and the output layer (in red) as the chosen answer from one selected audience.



If the output layer finds out that a specific neuron is always giving the best answer, it may neglect the others and give all the weight to this neuron.

Dropout in Neural Network

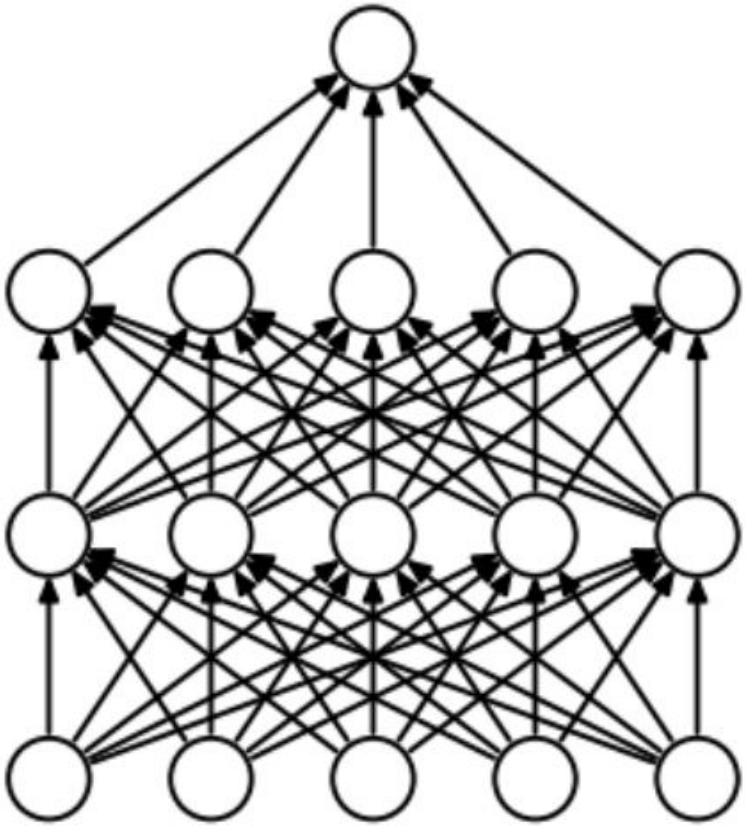
Based on our previous analysis, we chose to forbid some neurons of answering and give chance to others. This way we will achieve balance and force all neurons to learn.

This is the concept of dropout, and technically it works as follows:

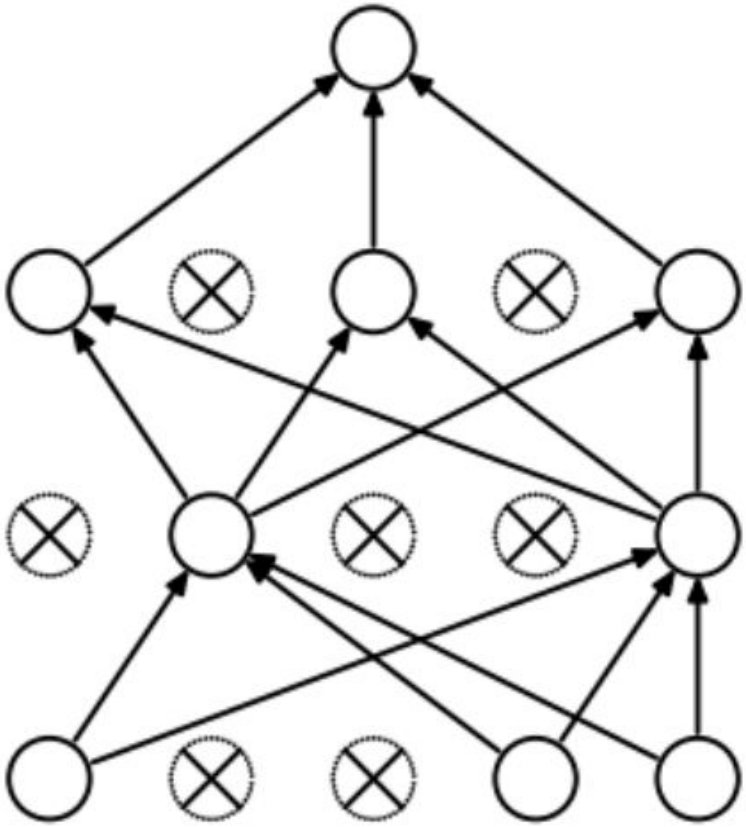
- We assign a dropout rate, which represents the percentage of neurons to drop (e.g. 20% of neurons)
- At each stage, we remove random neurons according to the predefined percentage.
- We calculate the final output according to the combination of results from the remaining neurons.

With this technique all neurons will have the chance to vote and will be obliged to answer correctly to decrease the model loss.

Dropout



(a) Standard Neural Net



(b) After applying dropout.

Dropout

In Deep Neural Networks, the chances of overfitting are very high. Therefore, Dropout acts as a regularization to the NN. It makes the model more robust.

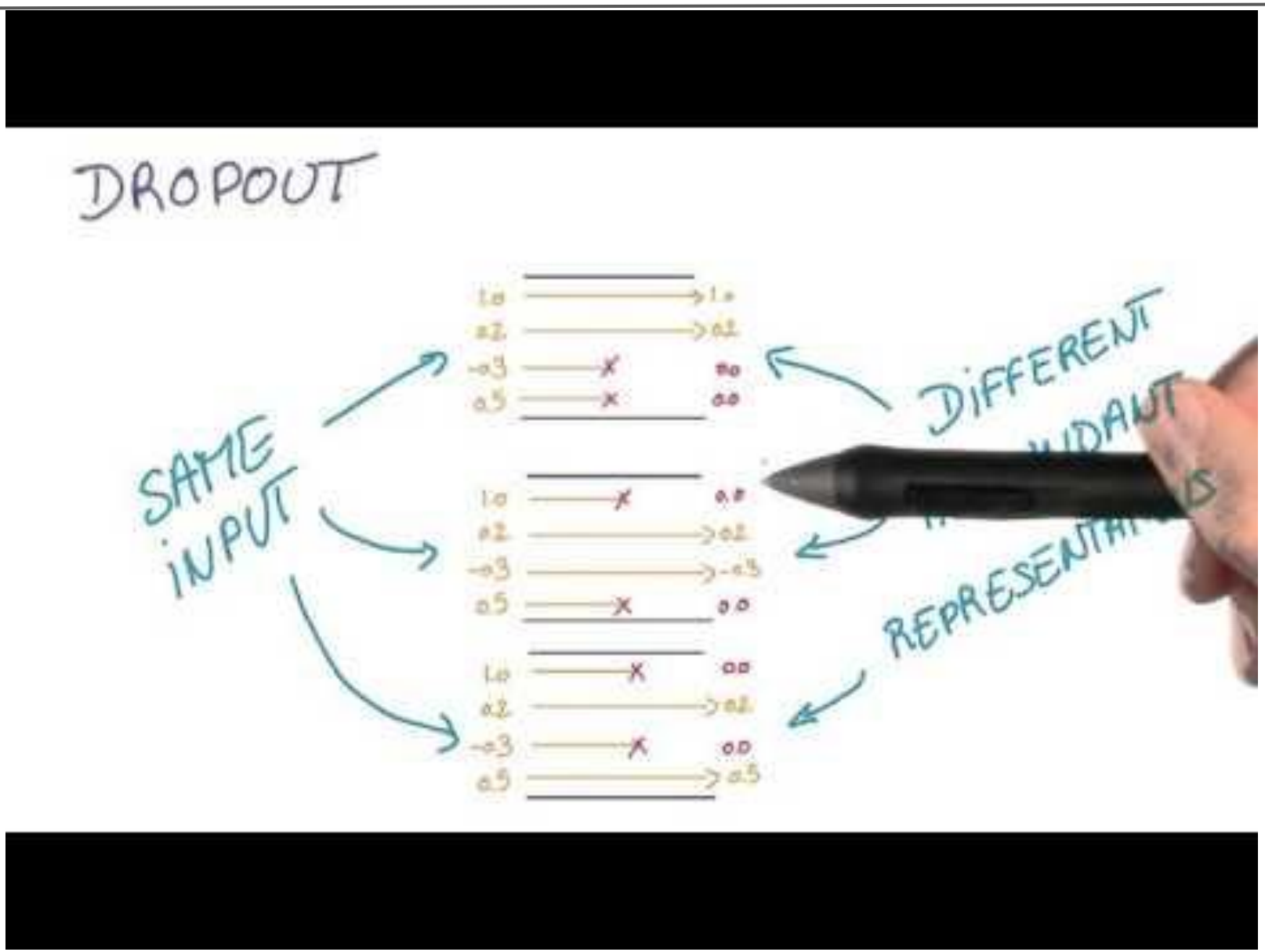
The percentage of neurons to be dropped is a hyperparameter that can be tuned based on the amount of overfitting on the data.

By dropping a unit out, we mean temporarily removing it from the network, along with all its incoming and outgoing connections.

It can be used with most types of layers, such as dense fully connected layers. Dropout may be implemented on any or all hidden layers in the network as well as the visible or input layer.

NOTE : It is not used on the output layer.

Dropout



Let's understand Regularization and Dropout practically

https://github.com/dphi-official/Deep_Learning_Bootcamp/blob/master/Optimization_Techniques/Regularization_and_Dropout.ipynb

- Download
- Extract zip file
- Open in Jupyter Notebook or Upload on Google Colab

Slide Download Link

- You can download the slides here: https://docs.google.com/presentation/d/13TTZ67amE1l80w4A-N2ro9DD9mNeR6RG-UbALpobzk0/edit?usp=sharing

References

https://towardsdatascience.com/an-intuitive-explanation-to-dropout-749c7fb5395c

That’s it for the day. Thank you!

Feel free to post any queries on the
Discuss forum or #help channel on Slack