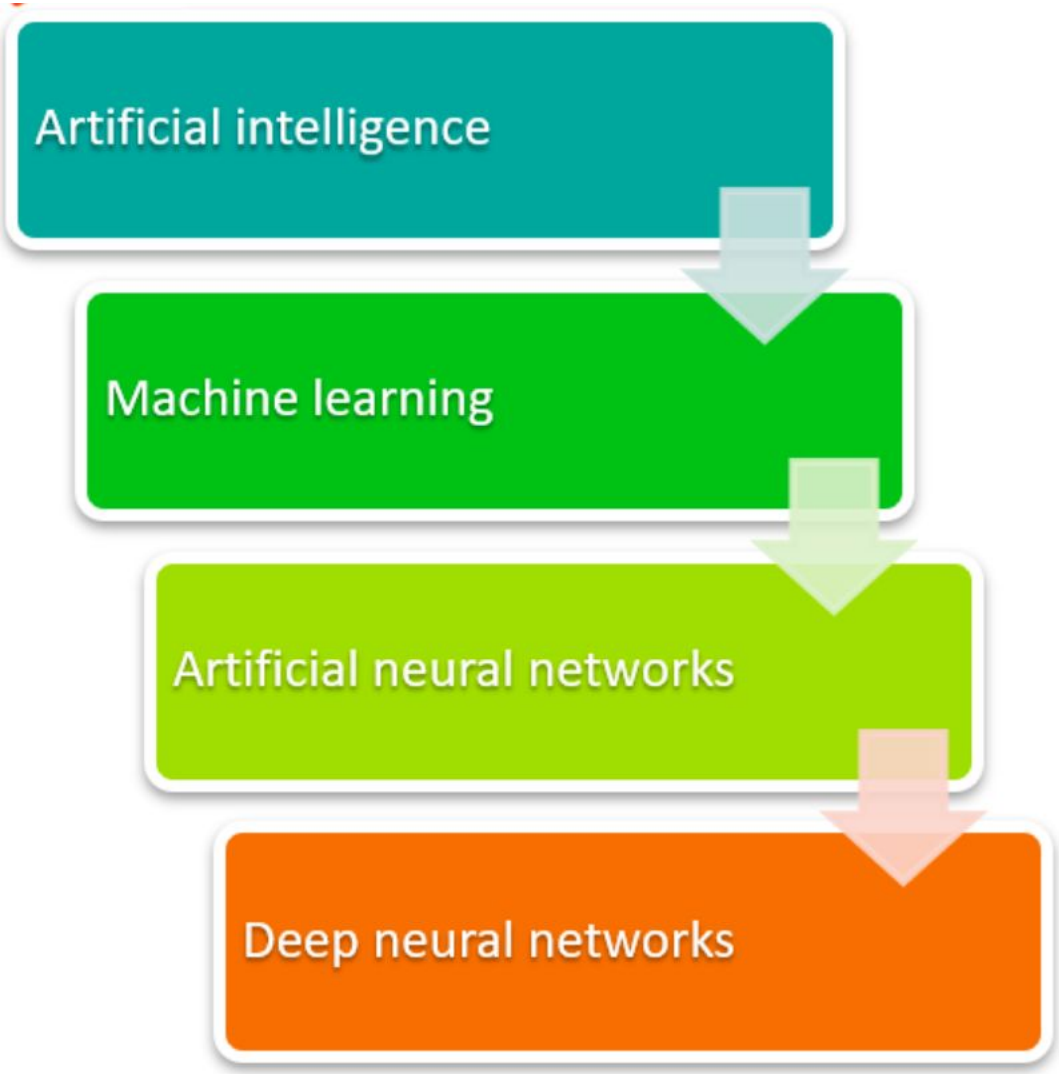
Berkeley Engineering | BerkeleyHaas

PROFESSIONAL CERTIFICATE IN MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

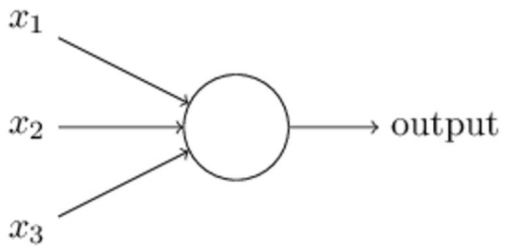
Office Hour #22 with Matilde D'Amelio

August 25, 2022 at 9 pm UTC

Deep Neural Network

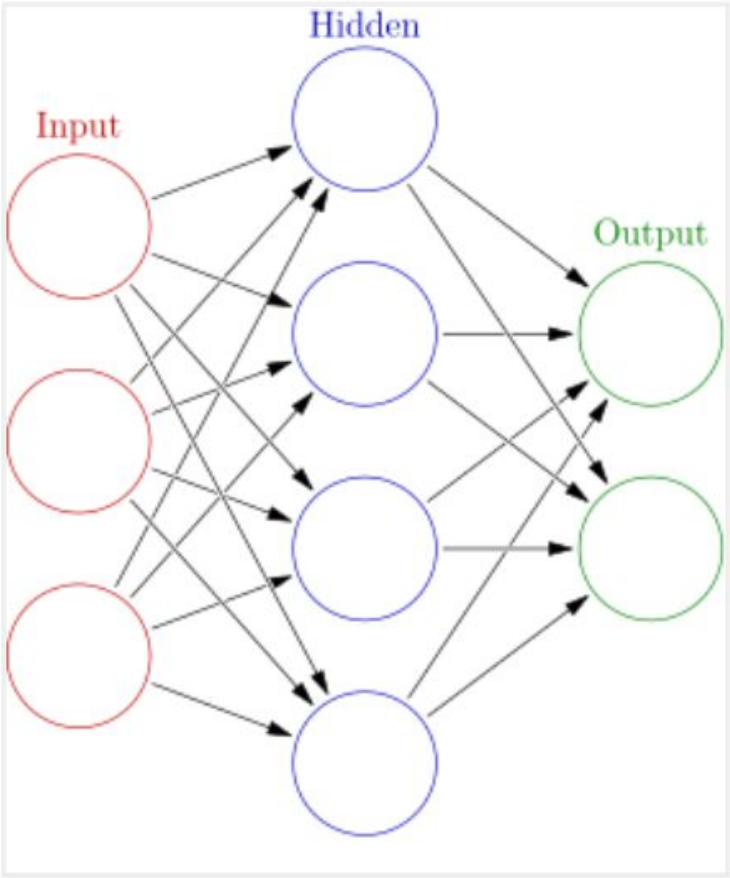


First, machine learning had to get developed. ML is a framework to automate (through algorithms) statistical models to get better at making predictions. A model is a single model that makes predictions about something. Those predictions are made with some accuracy.



Artificial Neural Network

Second it comes Artificial Neural Networks (ANNs), that utilize the **hidden layer** as a place to store and evaluate how significant one of the inputs is to the output. The hidden layer stores information regarding the input's importance, and it also makes associations between the importance of combinations of inputs.

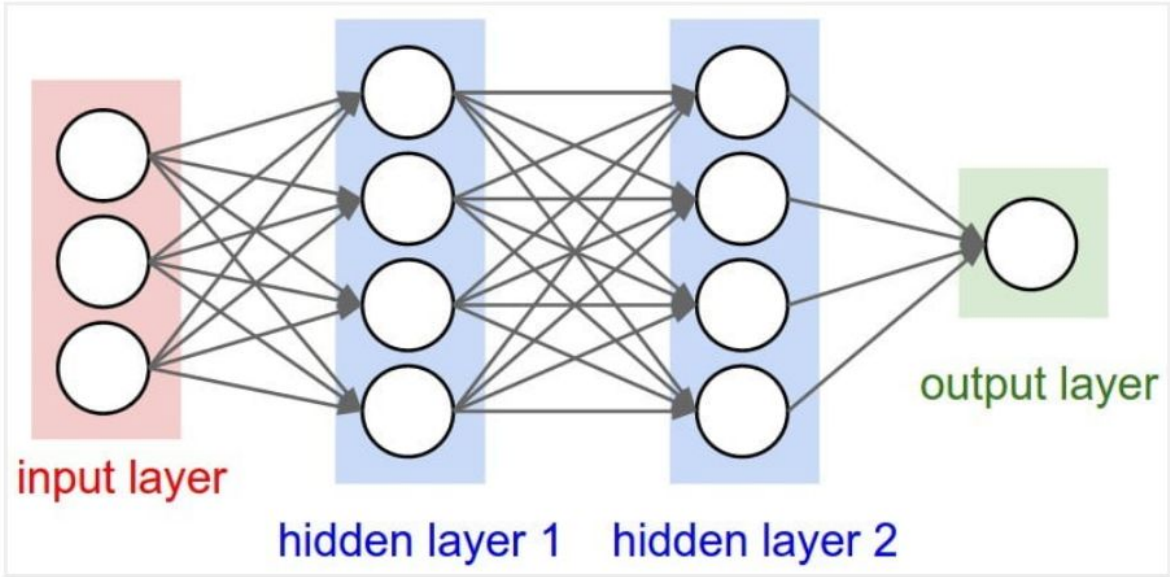


One hidden layer is considered an Artificial Neural Network (ANN)

Deep Neural Network

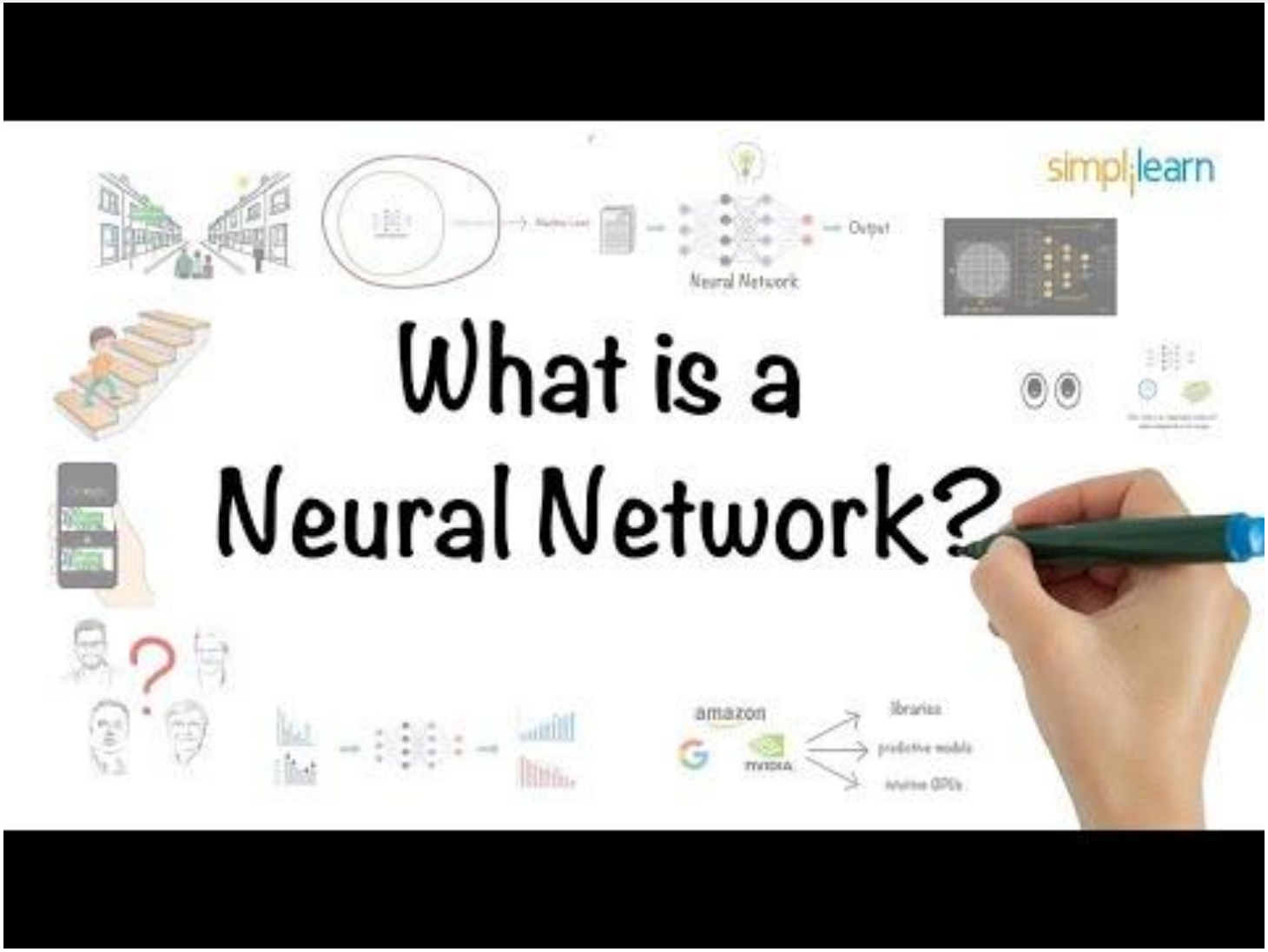
Deep neural nets, then, capitalize on the ANN component. They say, if that works so well at improving a model—because each node in the hidden layer makes both associations and grades importance of the input to determining the output—then why not stack more and more of these upon each other and benefit even more from the hidden layer?

So, the deep net has multiple hidden layers. ‘Deep’ refers to a model’s layers being multiple layers deep.



Two or more hidden layers comprise a Deep Neural Network

Deep Neural Network



Deep Neural Network - Applications

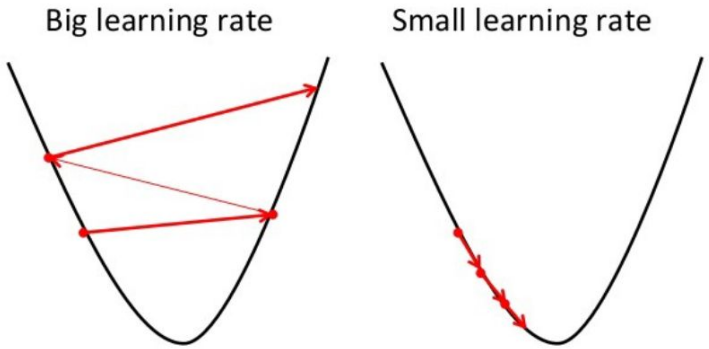
1. Facial Recognition (e.g., surveillance, security)
2. Stock market prediction
3. Behavior on Social Media
4. Aerospace (e.g., autopilot, control systems, dynamic simulations)
5. Defence Strategy
6. Voice Recognition
7. Healthcare (e.g., e.g., x ray, ultrasounds, scan)
8. Signature Verification and Handwriting Analysis (e.g., online banks)
9. Weather Forecast (e.g., likelihood of natural disasters-earthquakes)

Hyperparameter Optimisation

Hyperparameters are the **variables which determines the network structure** and the **variables which determine how the network is trained**. They are **set before training** (before optimizing the weights and bias)

Optimization techniques:

- **Number of hidden layers:** The idea is that you want to keep your DNN as simple as possible (you want it to be fast and well generalized), but at the same time, you want it to well classify your input data. In this case you should proceed with **manual attempts**.
- **Learning rate**: this hyperparameter refers to the step of backpropagation, when parameters are updated according to an optimization function.



Hyperparameter Optimisation

Optimization techniques:

- **Momentum** helps to know the direction of the next step with the knowledge of the previous steps. It helps to prevent oscillations. A typical choice of momentum is between 0.5 to 0.9.
- **Number of epochs**: number of epochs is the number of times the whole training data is shown to the network while training. Increase the number of epochs until the validation accuracy starts decreasing even when training accuracy is increasing (overfitting).
- **Batch size:** Mini batch size is the number of sub samples given to the network after which parameter update happens. A good default for batch size might be 32. Also try 32, 64, 128, 256, and so on.

Batch Normalisation

In **deep learning**, preparing a **deep neural network** with many layers as they can be delicate to the underlying i**nitial random weights** and design of the learning algorithm.

One potential purpose behind this trouble is the distribution of the inputs to layers somewhere down in the network may change after each mini-batch when the weights are refreshed. This can make the learning algorithm always pursue a moving target. This adjustment in the distribution of inputs to layers in the network has alluded to the specialized name **internal covariate shift**.

The challenge is that the model is refreshed layer-by-layer in reverse from the output to the input utilizing an estimate of error that accept the weights in the layers preceding the current layer are fixed.

Batch normalization gives a rich method of parametrizing practically any deep neural network. The reparameterization fundamentally decreases the issue of planning updates across numerous layers.

QUESTIONS?